ETSI TS 103 115 V9.4.0 (2015-04)

TECHNICAL SPECIFICATION

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

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

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

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

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

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

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

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

Copyright Notification

ETSI

Contents

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.

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

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

1 Scope

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

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

2 References

2.1 Normative references

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

2.2 Informative references

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

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

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

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

Not applicable.

3 Definitions, symbols and abbreviations

3.1 Definitions

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

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

Conformance Requirement Reference (CRR): description of the expected HCI API behaviour according to ETSI TS 102 705 [1]

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

HCP message: message as specified in ETSI TS 102 622 [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 (ETSI TS 102 622 [3]) protocol specification

3.2 Symbols

For the purposes of the present document, the symbols given in ETSI 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 ETSI 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
CTR   ConTRol
DUT   Device Under Test
EVT   EVenT
FFS   For Further Study
GND   GrouND
HCI   Host Controller Interface

NOTE: According to ETSI TS 102 622 [3].

HCP   Host Controller Protocol

NOTE: According to ETSI 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

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 an unique group of related optional items and the logic of their selection which is defined immediately following the table.

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

References to items

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

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

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

4 Applicability

4.1 Table of optional features

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

Table 4.1: Options

Item	Option	Status	Support	Mnemonic
1	Card emulation, Type A	O		O_CE_TYPE_A
2	Card emulation, Type B	O		O_CE_TYPE_B
3	Card emulation, Type B'	O		O_CE_TYPE_B_PRIME
4	Card emulation, Type F	O		O_CE_TYPE_F
5	Reader Mode, Type A	O		O_RM_TYPE_A
6	Reader Mode, Type B	O		O_RM_TYPE_B
7	HCP message size greater than supported buffer size	O		O_MSG_GT_BUF

4.2 Applicability table

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

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

Table 4.2 a): Applicability of tests

Clause	Test case number and description	Release	Execution requirements	Rel-9 UICC	Support
6.1.1.1	Method getHCIService (ID2, ID4)	Rel-9	SAA1	M	
6.1.1.1	Method getHCIService (ID1-1)	Rel-9	SAA1	M	
6.1.1.1	Method getHCIService (ID1-2)	Rel-9	SAA2, SAA1	M	
6.1.1.1	Method getHCIService (ID1-3)	Rel-9	SAA3	M	
6.1.1.1	Method getHCIService (ID3-1)	Rel-9	SAA6	M	
6.1.1.1	Method getHCIService (ID3-2)	Rel-9	SAA1, SAA7	M	
6.1.1.1	Method getHCIService (ID3-3)	Rel-9	SAA8	M	
6.1.1.1	Method getHCIService (ID5-1)	Rel-9	SAA4, SAA1	M	
6.1.1.1	Method getHCIService (ID5-2)	Rel-9	SAA3, SAA5	M	
6.1.1.2	Method getPowerMode	Rel-9	SAA1	M	
6.1.1.3	Method isHCIServiceAvailable (ID1-1, ID3-1)	Rel-9	SAA1	M	
6.1.1.3	Method isHCIServiceAvailable (ID1-2)	Rel-9	SAA2, SAA1	M	
6.1.1.3	Method isHCIServiceAvailable (ID1-3)	Rel-9	SAA3	M	
6.1.1.3	Method isHCIServiceAvailable (ID2)	Rel-9		M	
6.1.1.3	Method isHCIServiceAvailable (ID3-2)	Rel-9	SAA3	M	
6.1.1.3	Method isHCIServiceAvailable (ID4-1)	Rel-9	SAA4	M	
6.1.1.3	Method isHCIServiceAvailable (ID4-2)	Rel-9	SAA5	M	
6.1.1.3	Method isHCIServiceAvailable (ID5-1)	Rel-9	SAA10	M	
6.1.1.3	Method isHCIServiceAvailable (ID5-2)	Rel-9	SAA11	M	
6.1.2.1	Method register (ID1-1, ID3-1, ID4)	Rel-9	SAA1	M	
6.1.2.1	Method register (ID1-2, ID2, ID3-2)	Rel-9	SAA1, SAA2	M	
6.1.2.1	Method register (ID1-3, ID3-3)	Rel-9	SAA3	M	
6.1.2.2	Method deregister (ID1-1, ID1-3)	Rel-9	SAA1	M	
6.1.2.2	Method deregister (ID1-2, ID2)	Rel-9	SAA1, SAA2	M	
6.1.2.3	Method activateEvent (ID1, ID4, ID6, ID7, ID8, ID9, ID10)	Rel-9	SAA1	M	
6.1.2.3	Method activateEvent (ID2)	Rel-9	SAA3	M	

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, SAA3	M	
6.1.2.3	Method activateEvent (ID5-2)	Rel-9	SAA11, SAA3	M	
6.1.2.4	Method deactivateEvent (ID1, ID4, ID5)	Rel-9	SAA1	M	
6.1.2.4	Method deactivateEvent (ID2)	Rel-9	SAA3	M	
6.1.2.4	Method deactivateEvent (ID7)	Rel-9	SAA3, SAA1	M	
6.1.2.4	Method deactivateEvent (ID3, ID6)	Rel-9	SAA1, SAA2	M	
6.1.2.5	Method requestCallbackNotification	Rel-9	SAA1	M	
6.1.2.6	Method getEventNotificationStatus(ID1, ID2, ID5-1)	Rel-9	SAA1	M	
6.1.2.6	Method getEventNotificationStatus(ID3,ID4, ID5-2)	Rel-9	SAA3	M	
6.1.2.6	Method getEventNotificationStatus(ID5-3, ID6, ID7)	Rel-9	SAA2, SAA1	M	
6.1.3.1	Method isHeading	Rel-9	SAA1	M	
6.1.3.2	Method isComplete	Rel-9	SAA1	M	
6.1.3.3	Method getType (ID1)	Rel-9	SAA1	M	
6.1.3.3	Method getType (ID2)	Rel-9	SAA2, SAA1	M	
6.1.3.4	Method getInstruction	Rel-9	SAA1	M	
6.1.3.6	Method getReceiveLength	Rel-9	SAA1	M	
6.1.3.7	Method getReceiveBuffer	Rel-9	SAA1	M	
6.1.4.1	Method throwIt	Rel-9	SAA1	M	
6.1.5.1	Method onCallback (ID1, ID2, 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, ID7-2)	Rel-9	SAA1	C001	
6.2.1.3.1	Method onCallback (ID4, ID5, ID6, ID7-3)	Rel-9	SAA1	C002	
6.2.1.3.1	Method onCallback (ID7-1, ID7-4)	Rel-9	SAA1	M	
6.2.2.1.1	Method prepareAndSendConnectivityEvent (ID1, ID3, ID4, ID5)	Rel-9	SAA2, SAA1	M	
6.2.2.1.1	Method prepareAndSendConnectivityEvent (ID2-1)	Rel-9	SAA2, SAA10	M	
6.2.2.1.1	Method prepareAndSendConnectivityEvent (ID2-2)	Rel-9	SAA2, SAA11	M	
6.2.2.1.1	Method prepareAndSendConnectivityEvent (ID6)	Rel-9	SAA2, SAA1, SAA9	M	
6.2.2.1.1	Method prepareAndSendConnectivityEvent (ID7)	Rel-9	SAA2, SAA3	M	
6.2.2.1.2	Method prepareAndSendTransactionEvent byte[] aid, short aidOffset, short aidLen, byte[] parameters, short parametersOffset, short parametersLen) (ID1, ID3, ID4)	Rel-9	SAA1, SAA2	M	

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	
6.2.2.1.2	Method prepareAndSendTransactionEvent byte[] aid, short aidOffset, short aidLen, byte[] parameters, short parametersOffset, short parametersLen) (ID6)	Rel-9	SAA2, SAA3	M	
6.2.2.1.3	Method prepareAndSendTransactionEvent (byte[] parameters, short parametersOffset, short parametersLen) (ID1, ID4)	Rel-9	SAA2, SAA1	M	
6.2.2.1.3	Method prepareAndSendTransactionEvent (byte[] parameters, short parametersOffset, short parametersLen) (ID3)	Rel-9	SAA2	M	
6.2.2.1.3	Method prepareAndSendTransactionEvent (byte[] parameters, short parametersOffset, short parametersLen) (ID2-1)	Rel-9	SAA2, SAA10	M	
6.2.2.1.3	Method prepareAndSendTransactionEvent (byte[] parameters, short parametersOffset, short parametersLen) (ID2-2)	Rel-9	SAA2, SAA11	M	
6.2.2.1.3	Method prepareAndSendTransactionEvent (byte[] parameters, short parametersOffset, short parametersLen) (ID5)	Rel-9	SAA2, SAA3	M	
6.2.3.1.1	Method restartReaderModeProcedure (ID1)	Rel-9	SAA3	C005	
6.2.3.1.1	Method restartReaderModeProcedure (ID2)	Rel-9	SAA3	C006	
6.2.3.1.1	Method restartReaderModeProcedure (ID3, ID4)	Rel-9	SAA3	M	
6.2.3.1.1	Method restartReaderModeProcedure (ID5-1)	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 (ID5)	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 ETSI 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 (SAAn) for several test cases, to define the service availability and access possibility for the different services. For these test cases, the availability of the different services shall be guaranteed in order to execute the corresponding test procedure against the DUT. In case of absence of a particular (SAAn), the corresponding test case should not be carried out.

5 Test environment

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

5.1 Test environment description

The general architecture for the test environment is:
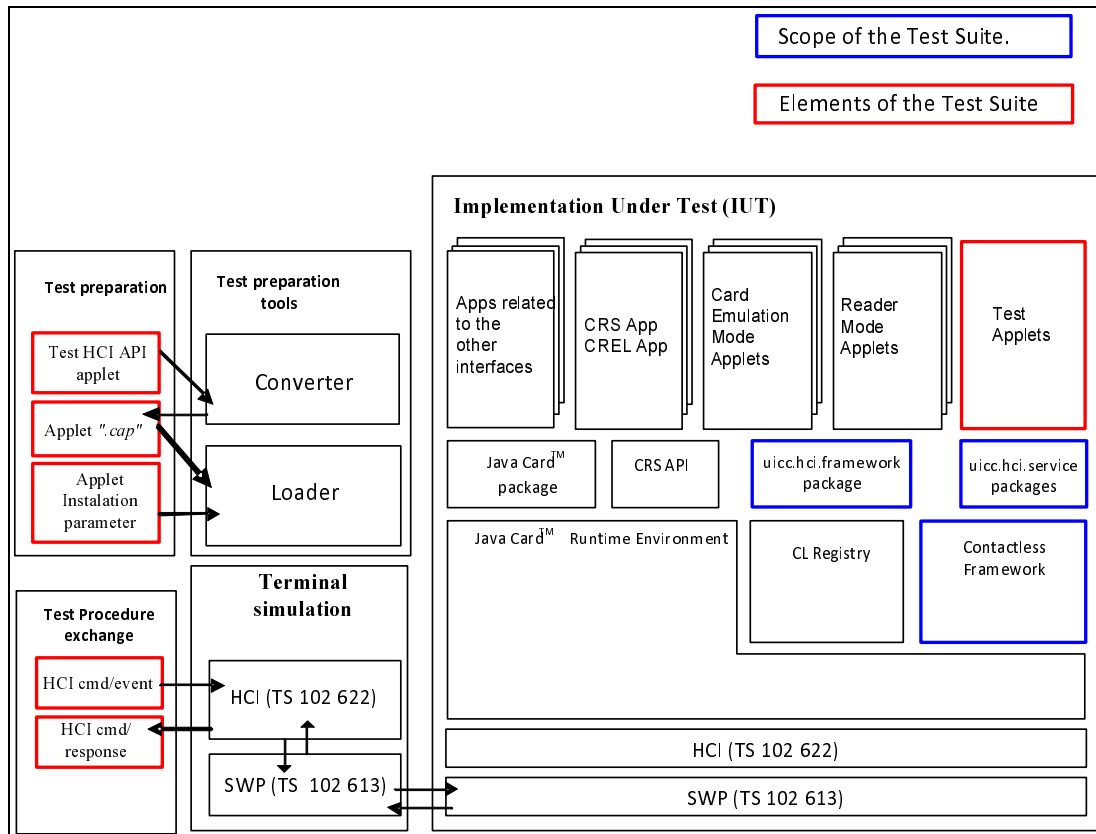
Figure 5.1

5.2 Tests format

5.2.1 Test area reference

Each test area is referenced as follows:

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

- package name:
  - uicc.hci.framework: '1'.
  - uicc.hci.services: '2'.
- class name:
  - yyy: 3 letters for each class/interface.
    - See annex A for full classes/interfaces acronyms list.

- method name:

    - zzz[input parameters]:

        - See annex A for full methods name acronyms list.

5.2.1.1 Conformance requirements

The conformance requirements are expressed in the following way:

- Method prototype as listed in ETSI TS 102 705 [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 precondtions defined in clause 5.3, this clause defines the initial conditions prior to the execution of each test case; i.e. for each ID.

5.2.1.4 Test procedure

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

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

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

5.3 Initial conditions

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

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

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

Prior to any test:

- For test cases run in full power mode, the test environment shall have the UICC powered on and performed the session initialization and RF registries updates as described in ETSI TS 102 622 [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 ETSI 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 ETSI 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 ETSI 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 ETSI 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 ETSI 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 ETSI 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

Byte 1		Byte 12	Byte 13	Byte 14	Byte 15	Byte 16	
							Application Provider specific data
							TAR
							Specified in ETSI TS 101 220 [4]
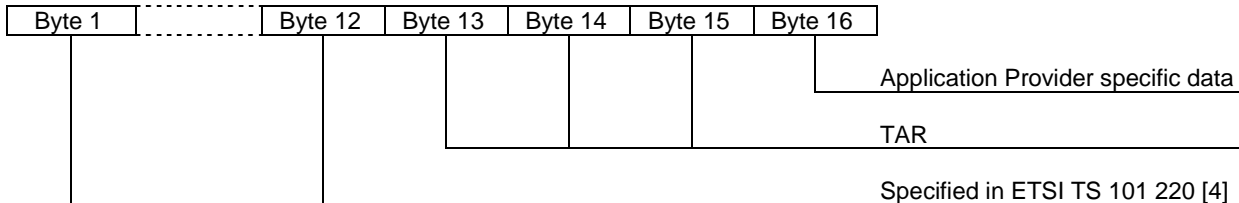
TAR coding (3 bytes / 24 bits):

```
b1 ---- b4  b5 ---- b8  b9 ---- b16 ---- b17 ---- b24
                                              └─── Applet class/instance number
                                   └─────────────── Package number
                 └────────────────────────────────── Core package identifier
    └───────────────────────────────────────────────── Test Part Identifier
```
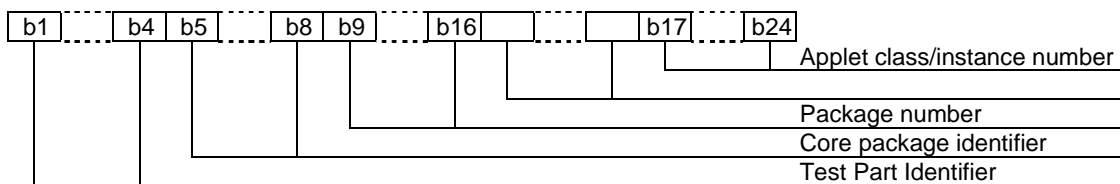
Applet instance number, Applet Class number, Package number:

- For package AID, package number shall start from 0 and class and instance numbers shall be 0.

- For class AID, package number is the number of the class package, class number shall start from 1 and instance shall be 0.

- For instance AID, package and class number are the number of class and package of which instance belongs, and instance number shall start from 1.

Test Part and Core Package Identifier

- 0000 0000 reserved (as TAR= '00.00.00' is reserved for Issuer Security Domain).

- 0010 0001 uicc.hci.test.framework.

- 0010 0100 uicc.hci.test.services.

- 0010 0101 uicc.hci.test.services.cardemulation.

- 0010 0110 uicc.hci.test.services.connectivity.

- 0010 0111 uicc.hci.test.services.readermode.

Application Provider specific data (1 byte):

- '00' for Package.

- '01' for Applet class.

- '02' for Applet Instance.

EXAMPLE: Annex B.

5.6 Test equipment

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

5.6.1 Test tool

This test tool shall meet the following requirements:

- be able to send and receive commands using the underlying HCI resources (e.g. gates and pipes) defined by the HCI protocol as specified in ETSI TS 102 622 [3];

- be able to send and receive the commands correctly on the lower layer; i.e ETSI 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**
1			Select supported and registered service		
	1 - EVT_SEND_DATA (INS = '01')	`getHCIService()` `serviceID =` `CARD_EMULATION_SERVICE_ID`	No exception shall be thrown	EVT_SEND_DATA (SW - '90 00')	N1, N2
	2 - EVT_SEND_DATA (INS = '02')	`getHCIService()` `serviceID =` `CONNECTIVITY_SERVICE_ID`	No exception shall be thrown	EVT_SEND_DATA (SW - '90 00')	N1, N2
	3 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.1.1.3 not applicable here - send APDU (INS = '03')	`getHCIService()` `serviceID =` `READER_SERVICE_ID`	No exception shall be thrown	SW - '90 00'	N1, N2
2			Applet not registered		
	EVT_SEND_DATA (INS = '01')	`Applet.register() has not` `yet been invoked.` `getHCIService()` `serviceID =` `CARD_EMULATION_SERVICE_ID`	Return: Null	EVT_SEND_DATA (SW - '90 00')	N3
3			Select not availabile service		
	1 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.1.1.3 not applicable here - send APDU (INS = '01')	`getHCIService()` `serviceID =` `CardEmulationService`	Shall throw uicc.hci.framework.HCIException with error code HCI_SERVICE_NOT_AVAILABLE	SW - '90 01'	C2
	2 - EVT_SEND_DATA (INS ='02')	`getHCIService()` `serviceID =` `ConnectivityService`	Shall throw uicc.hci.framework.HCIException with error code HCI_SERVICE_NOT_AVAILABLE	EVT_SEND_DATA (SW - '90 01')	C2
	3 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.1.1.3 not applicable here - send APDU (INS = '03')	`getHCIService()` `serviceID = ReaderService`	Shall throw uicc.hci.framework.HCIException with error code HCI_SERVICE_NOT_AVAILABLE	SW - '90 01'	C2
4			Select undefined service		
	EVT_SEND_DATA (INS = '04')	`getHCIService()` `serviceID = -1`	Shall throw javacard.framework.SystemException with error code ILLEGAL_VALUE	EVT_SEND_DATA (SW - '90 00')	P1

	Test case				
ID	**HCI Command**	**API Description**	**API Expectation**	**HCI Response**	**CRR**
5	Access not granted				
	1 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.1.1.3 not applicable here - send APDU (INS = '01')	`getHCIService() seviceID = CardEmulationService`	Shall throw uicc.hci.framework.HCIException with error code HCI_ACCESS_NOT_GRANTED	SW - '90 02'	C1
	2 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.1.1.3 not applicable here - send APDU (INS='03')	`getHCIService() serviceID = ReaderService`	Shall throw uicc.hci.framework.HCIException with error code HCI_ACCESS_NOT_GRANTED	SW - '90 02'	C1

6.1.1.2 Method getPowerMode

Test Area Reference: Api_1_Hdv_Gpm.

6.1.1.2.1 Conformance requirements

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

```
public static byte getPowerMode()
```

6.1.1.2.1.1 Normal execution

- CRRN1: returns the current power mode of the UICC as defined in ETSI TS 102 613 [14].

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

- CRRN3: Applets communicating through the process() method shall also be able to use the API services defined in ETSI TS 102 705 [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	**Verify power mode**				
	- Activate the UICC in full power mode. - EVT_FIELD_ON - EVT_CARD_ACTIVATED - EVT_SEND_DATA (select applet) - EVT_SEND_DATA(INS='05')	`getPowerMode()`	return = FULL_POWER_MODE	EVT_SEND_DATA (SW - '90 03')	N1, N3
2	**Verify power mode**				
	Activate the UICC in low power mode. - EVT_FIELD_ON - EVT_CARD_ACTIVATED - EVT_SEND_DATA (select applet) - EVT_SEND_DATA(INS='05')	`getPowerMode()`	return = LOW_POWER_MODE	EVT_SEND_DATA (SW - '90 04')	N1, N3
3	**ISO interface is activated, SWP interface is not activated**				
	Send command on ISO interface to select applet. - send APDU (INS='01') - send APDU (INS='05')	`getPowerMode()`	return = FULL_POWER_MODE	- SW - '90 00' SW - '90 03'	N1 N3

6.1.1.3   Method isHCIServiceAvailable

Test Area Reference: Api_1_Hdv_Isa.

6.1.1.3.1   Conformance requirements

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

```
public static byte isHCIServiceAvailable(short serviceID)
```

6.1.1.3.1.1   Normal execution

- CRRN1: this method is used to retrieve information about the availability of a HCIService more related to user interface.

- CRRN2: this method returns (byte)0x00 if the service is available and can be used, otherwise bit values set to:

    - bit 0 set: interface not supported by terminal;

    - bit 1 set: service not supported by terminal;

    - bit 2 set: access not allowed for Applet;

    - bit 3 set: HCI interface currently disabled in UICC.

6.1.1.3.1.2   Parameter errors

- None.

6.1.1.3.1.3   Context errors

- None.

6.1.1.3.2 Test suite files

Applet Name	Test case ID
Api_1_Hdv_Isa_1.java	1
Api_1_Hdv_Isa_1.java	2
Api_1_Hdv_Isa_1.java	3
Api_1_Hdv_Isa_1.java	4
Api_1_Hdv_Isa_2.java	5-1
Api_1_Hdv_Isa_3.java	5-2

6.1.1.3.3 Initial conditions

- EVT_FIELD_ON has been sent on HCI interface.

- EVT_CARD_ACTIVATED has been sent on HCI interface.

- According applet has been successfully installed and selected using HCI interface.

6.1.1.3.4 Test procedure

	Test Case				
ID	HCI/APDU Command	API Description	API Expectation	HCI/APDU Response	CRR
1	Retrieve information of an available service				
	1 - EVT_SEND_DATA (INS = '06')	isHCIServiceAvailable() serviceID = CARD_EMULATION_SERVICE_ID	0x00	EVT_SEND_DATA (SW - '90 00', data byte set to '00')	N1, N2
	2 - EVT_SEND_DATA (INS = '07')	isHCIServiceAvailable() serviceID = CONNECTIVITY_SERVICE_ID	0x00	EVT_SEND_DATA (SW - '90 00', data byte set to '00')	N1, N2
	3 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.3.1.3 not applicable here - send APDU INS='08'	isHCIServiceAvailable() serviceID = READER_SERVICE_ID	0x00	SW - '90 00', data byte set to '00'	N1, N2
2	Interface not supported by terminal				
	Precondition: SWP interface not supported; the initial conditions in clause 6.1.1.3.1.3 not applicable here, and C6 should be connected to GND: Sending command on ISO interface: - select applet - send APDU (INS = '06')	isHCIServiceAvailable()	bit 0 set	- SW - '90 00' - SW - '90 00', bit 0 set in the data byte	N2
3	Service not supported by terminal				
	1 - Precondition: disable Connectivity service support in terminal; i.e. connectivity gate is not available in terminal pipe creation is not possible and the terminal shall indicate in TERMINAL PROFILE HCI Connectivity is not supported. - EVT_SEND_DATA (INS = '07')	isHCIServiceAvailable() serviceID = CONNECTIVITY_SERVICE_ID	bit 1 set	EVT_SEND_DATA (SW - '90 00', bit 1 set in the data byte)	N2

ID	HCI/APDU Command	API Description	API Expectation	HCI/APDU Response	CRR

	Test Case				
ID	**HCI/APDU Command**	**API Description**	**API Expectation**	**HCI/APDU Response**	**CRR**
	2 - Precondition: disable Reader service support in terminal; i.e. reader RF gates are not available in terminal Send command on ISO interface to select applet; the initial conditions in clause 6.1.1.1.3 not applicable here - - send APDU (INS='08')	`isHCIServiceAvailable() serviceID = READER_SERVICE_ID`	bit 1 set	SW - '90 00', bit 1 set in the data byte	N2
4	**Access not allowed for Applet**				
	1 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.1.1.3 not applicable here - send APDU (INS = '06')	`isHCIServiceAvailable() serviceID = CARD_EMULATION_SERVICE_ID`	bit 2 set	SW - '90 00', bit 2 set in the data byte	N2
	2 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.1.1.3 not applicable here - send APDU (INS = '08')	`isHCIServiceAvailable() serviceID = READER_SERVICE_ID`	bit 2 set	SW - '90 00', bit 2 set in the data byte	N2
5	**HCI interface currently disabled in UICC**				
	1 - Precondition: The terminal shall indicate the support of class r by setting the 26th byte, 'b2' and the 31st byte, 'b1' in the terminal profile and disable the contactless functionality in the UICC as defined in ETSI TS 102 223 [7]. The initial conditions in clause 6.1.1.3.3 are not applicable here. Send on ISO interface the following commands: - Send APDU to select the applet. - Send APDU (INS = '06')	`isHCIServiceAvailable() serviceID = CARD_EMULATION_SERVICE_ID`	bit 3 set	SW - '90 00', bit 3 set in the data byte	N2

	2 - Precondition: The contactless interface is disabled in the UICC as defined in Global Platform Amendment C.
	The initial conditions in clause 6.1.1.3.3 are not applicable here.
	Send on ISO interface send the following commands:
	- Send APDU to select the applet.
	- Send APDU (INS = '06')
	- Postcondition:
	The contactless interface is enabled again in the UICC as defined in Global Platform Amendment C
		`setCommunicationInterface() API method of Global Platform Amendment C [10] is used to disable HCI interface`
		
		`isHCIServiceAvailable() serviceID = CARD_EMULATION_SERVICE_ID`
		
		`setCommunicationInterface() API method of Global Platform Amendment C [10] is used to enable again HCI interface`
			bit 3 set	SW - '90 00', bit 3 set in the data byte	N2

6.1.2   Interface HCIService

6.1.2.1   Method register

Test Area Reference: Api_1_Hsr_Reg.

6.1.2.1.1   Conformance requirements

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

```
void register(HCIListener listener)
               throws HCIException
```

6.1.2.1.1.1   Normal execution

- CRRN1: registers a Listener object to the Service instance.

- CRRN2: a certain listener type can only be registered to the same service type.

- CRRN3: The Registration of Listener Interfaces and activation of events shall be persistent.

6.1.2.1.1.2   Parameter errors

- CRRP1: throws HCIException in case the listener Object registered to the service is not implementing the corresponding interface of the HCIService instance with error code:

  - HCIException.HCI_WRONG_LISTENER_TYPE.

6.1.2.1.1.3   Context errors

- CRRC1: throws HCIException with error code HCI_LISTENER_ALREADY_REGISTERED in case a listener Object of the same type has already been registered.

6.1.2.1.2   Test suite files

Applet Name	Test case ID
Api_1_Hsr_Reg_1.java	1-1
Api_1_Hsr_Reg_3.java	1-2
Api_1_Hsr_Reg_5.java	1-3
Api_1_Hsr_Reg_1.java	2
Api_1_Hsr_Reg_2.java	3-1
Api_1_Hsr_Reg_4.java	3-2
Api_1_Hsr_Reg_6.java	3-3
Api_1_Hsr_Reg_1.java	4

6.1.2.1.3   Initial conditions

- EVT_FIELD_ON has been sent on HCI interface.

- EVT_CARD_ACTIVATED has been sent on HCI interface.

- According applet has been successfully installed and selected using HCI interface.

6.1.2.1.4 Test procedure

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

6.1.2.2 Method deregister

Test Area Reference: Api_1_Hsr_Drg.

6.1.2.2.1 Conformance requirements

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

```
void deregister(HCIListener listener)
```

6.1.2.2.1.1 Normal execution

- CRRN1: deregisters a Listener object from the Service instance.

- CRRN2: Calling deregister with an argument that does not identify a currently registered HCIListener has no effect.

- CRRN3: Applets communicating through the process() method shall also be able to use the API services defined in ETSI TS 102 705 [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
1	Deregister Listener to a Service - CardEmulation				
	1 - EVT_SEND_DATA (INS = '01')	`register()` `Service =` `CardEmulationService` `Listener =` `CardEmulationListener` `deregister()` `Service =` `CardEmulationService` `Listener =` `CardEmulationListener`	No exception shall be thrown	EVT_SEND_DATA (SW - '90 00')	N1, N3
	2 - EVT_SEND_DATA (INS = '02')	`deregister()` `Service =` `ConnectivityService` `Listener =` `CardEmulationListener`	No exception shall be thrown	EVT_SEND_DATA (SW - '90 00')	N2

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

	3 - EVT_SEND_DATA (INS = '03')	`deregister()`
		`Service =`
		`CardEmulationService`
		`Listener =`
		`CardEmulationListener`
			No exception shall be thrown	EVT_SEND_DATA (SW - '90 00')	N2, N3

2		**Deregister Listener to a Service - Connectivity**			

	1 - EVT_SEND_DATA (INS = '01')	`register()`
		`Service =`
		`ConnectivityService`
		`Listener =`
		`ConnectivityListener`
		
		`deregister()`
		`Service =`
		`ConnectivityService`
		`Listener =`
		`ConnectivityListener`
			No exception shall be thrown	EVT_SEND_DATA (SW - '90 00')	N1

	2 - EVT_SEND_DATA (INS = '02')	`deregister()`
		`Service =`
		`CardEmulationService`
		`Listener =`
		`ConnectivityListener`
			No exception shall be thrown	EVT_SEND_DATA (SW - '90 00')	N2

	3 - EVT_SEND_DATA (INS = '03')	`deregister()`
		`Service =`
		`ConnectivityService`
		`Listener =`
		`ConnectivityListener`
			No exception shall be thrown	EVT_SEND_DATA (SW - '90 00')	N2

6.1.2.3 Method activateEvent

Test Area Reference: Api_1_Hsr_Ace.

6.1.2.3.1 Conformance requirements

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

```
void activateEvent(byte event)
                throws HCIException
```

6.1.2.3.1.1 Normal execution

- CRRN1: This method is used to activate an event.

- CRRN2: for all service interfaces HCIListener.EVENT_HCI_TRANSMISSION_FAILED is sent to notify the applet when the UICC failed to transmit a message sent by CLF.

- 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 ETSI 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 ETSI 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 ETSI 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		**Event activation - CardEmulationService**			

	1 - EVT_SEND_DATA (INS = '10')	`HCIListener = CardEmulationListener HCIService = CardEmulationService`
		
		`activateEvent() event = HCIListener.EVENT_HCI_TRANSMISSION_FAILED`
			No exception shall be thrown	EVT_SEND_DATA (SW - '90 00')	N1, N2

	2 - EVT_SEND_DATA (INS = '11')	`HCIListener = CardEmulationListener HCIService = CardEmulationService`
		
		`activateEvent() event = HCIListener.EVENT_HCI_RECEPTIONS_FAILED`
			No exception shall be thrown	EVT_SEND_DATA (SW - '90 00')	N1, N3

	3 - EVT_SEND_DATA (INS = '12')	`HCIListener = CardEmulationListener HCIService = CardEmulationService`
		
		`activateEvent() event = CardEmulationListener.EVENT_GET_PARAMETER_RESPONSE`
			No exception shall be thrown	EVT_SEND_DATA (SW - '90 00')	N1, N4

	4 - EVT_SEND_DATA (INS = '14')	`HCIListener = CardEmulationListener HCIService = CardEmulationService activateEvent() event = CardEmulationListener.EVENT_ON_SEND_DATA`	No exception shall be thrown	EVT_SEND_DATA (SW - '90 00')	N1, N4
2		**Event activation - ReaderService**			

	1- Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.3.3 not applicable here - Send APDU INS='10'	`HCIService = ReaderService`
		
		`activateEvent() event = HCIListener.EVENT_HCI_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

Test Case					
ID	**HCI Command**	**Description**	**API Expectation**	**HCI Response**	**CRR**
3	Event activation - ConnectivityListener				

	1 - EVT_SEND_DATA (INS = '10')	`HCIListener = ConnectivityListener HCIService = ConnectivityService`
		
		`activateEvent() event = HCIListener.EVENT_HCI_TRANSMISSION_FAILED`
			No exception shall be thrown.	EVT_SEND_DATA (SW - '90 00')	N1, N2

	2 - EVT_SEND_DATA (INS = '11')	`HCIListener = ConnectivityListener HCIService = ConnectivityService`
		
		`activateEvent() event = HCIListener.EVENT_HCI_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	Wrong event type				

	EVT_SEND_DATA (INS = '16')	`HCIListener = CardEmulationListener HCIService = CardEmulationService`
		
		`activateEvent() event value =  0x02`
			HCIException with reason code HCI_WRONG_EVENT_TYPE shall be thrown	EVT_SEND_DATA (SW - '90 00')	P1

5	HCI is disabled				
	1- Precondition: The terminal shall indicate the support of class r by setting the 26th byte, 'b2' and the 31st byte, 'b1' in the terminal profile and disable the contactless functionality in the UICC as defined in ETSI TS 102 223 [7]. - Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.3.3 not applicable here - - Send APDU (INS = '01') on ISO interface	`HCIService = ReaderService activateEvent() event = ReaderListener.EVENT_TARGET_DISCOVERED`	HCIException with reason code HCI_CURRENTLY_DISABLED shall be thrown	SW- '90 00'	P2, N8

	2 - Precondition: The contactless interface is disabled in the UICC as defined in Global Platform Amendment C. - Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.3.3 not applicable here - Send APDU (INS = '01') on ISO interface - Postcondition: The contactless interface is enabled again in the UICC as defined in Global Platform Amendment C	`setCommunicationInterface() API method of "GlobalPlatform Amendment C" [10] used to disable HCI interface HCIService = ReaderService activateEvent() event = ReaderListener.EVENT_TARGET_DISCOVERED`
		
		`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

		Test Case			
ID	**HCI Command**	**Description**	**API Expectation**	**HCI Response**	**CRR**
6		**Persistent Event Activation**			
	- Send EVT_SEND_DATA (INS = '01') - EVT_FIELD_OFF - Power off - Power on - EVT_FIELD_ON - EVT_CARD_ACTIVATED - Select the applet - Send EVT_SEND_DATA (INS = '02')	`Service = CardEmulationService Listener = CardEmulationListener` `activateEvent(EVENT_FIELD_OFF )`	No exception shall be thrown. getEventNotificatio nStatus() = true	- EVT_SEND_DATA (SW - '90 00') - EVT_SEND_DATA (SW - '90 00')	N7
7		**Event Field Off before deselect**			
	- EVT_FIELD_OFF - EVT_FIELD_ON - EVT_CARD_ACTIVATED - Select the applet - Send EVT_SEND_DATA (INS = '01')	`Service = CardEmulationService Listener = CardEmulationListener activateEvent(EVENT_ON_SEND_D ATA) during the installation onCallback()` `activateEvent() event = EVENT_FIELD_OFF` `deselect()`	EVENT_FIELD_OF F raised before deselect()	EVT_SEND_DATA (SW - '90 00')	N9, N11, N12, N14
8		**No triggering after Event Field Off**			
	EVT_FIELD_OFF - Send EVT_SEND_DATA (INS = '01')	`Service = CardEmulationService Listener = CardEmulationListener activateEvent(EVENT_ON_SEND_D ATA) during the installation onCallback()` `activateEvent() event = EVENT_FIELD_OFF`	EVENT_ON_SEN D_DATA shall not be raised	No response, or EVT_SEND_DATA with any response except: (SW - '90 00')	N1, N4, N10, N11
9		**Check for Selection**			
	Send EVT_SEND_DATA (INS = '01')	`Service = CardEmulationService Listener = CardEmulationListener`	Check for invocation of select()	EVT_SEND_DATA (SW - '90 00')	N12, N13
10		**No triggering after Event Card Deactivate**			
	EVT_CARD_DEACTIVATE - Send EVT_SEND_DATA (INS = '01')	`Service = CardEmulationService Listener = CardEmulationListener activateEvent(EVENT_ON_SEND_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		**Event deactivation - CardEmulationService**			

	1 - EVT_SEND_DATA(INS='10')	`HCIService = CardEmulationService Event has been successfully activated`
		
		`deactivateEvent() event = HCIListener.EVENT_HCI_TRANSMISSION_FAILED`
			No exception shall be thrown	EVT_SEND_DATA (SW - '90 00')	N1, N2

	2 - EVT_SEND_DATA(INS='11')	`HCIService = CardEmulationService Event has been successfully activated`
		
		`deactivateEvent() event = HCIListener.EVENT_HCI_RECEPTIONS_FAILED`
			No exception shall be thrown	EVT_SEND_DATA (SW - '90 00')	N1, N3

	3 - EVT_SEND_DATA(INS='12')	`HCIService = CardEmulationService Event has been successfully activated`
		
		`deactivateEvent() event = CardEmulationListener.EVENT_GET_PARAMETER_RESPONSE`
			No exception shall be thrown	EVT_SEND_DATA (SW - '90 00')	N1, N4

	4 - EVT_SEND_DATA(INS='14')	`HCIService = CardEmulationService Event has been successfully activated`
		
		`deactivateEvent() event = CardEmulationListener.EVENT_ON_SEND_DATA`
			No exception shall be thrown	EVT_SEND_DATA (SW - '90 00')	N1, N4

2		**Event deactivation - ReaderService**			

	1- Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.3.3 not applicable here
	- Send APDU (INS='09')
		`HCIService = ReaderService`
		
		`deactivateEvent() event = HCIListener.EVENT_HCI_TRANSMISSION_FAILED`
			No exception shall be thrown	SW - '90 00'	N1, N2

	2 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.4.3 not applicable here
	- Send APDU (INS='10')
		`HCIListener = ReaderListener HCIService = ReaderService Event has been successfully activated`
		
		`deactivateEvent() event = HCIListener. EVENT_HCI_TRANSMISSION_FAILED`
			No exception shall be thrown	SW - '90 00'	N1, N3

	3 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.4.3 not applicable here
	- Send APDU (INS='11')
		`HCIListener = ReaderListener HCIService = ReaderService Event has been successfully activated`
		
		`deactivateEvent() event = EVENT_GET_PARAMETER_RESPONSE`
			No exception shall be thrown	SW - '90 00'	N1, N5

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

	4 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.4.3 not applicable here - Send APDU (INS='12')	`HCIListener = ReaderListener HCIService = ReaderService`
		
		`deactivateEvent() event = EVENT_WRITE_EXCHANGE_DATA _RESPONSE`
			No exception shall be thrown	SW - '90 00'	N1, N5

	5- Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.4.3 not applicable here - Send APDU (INS='13')	`HCIListener = ReaderListener HCIService = ReaderServiceEvent has been successfully activated`
		
		`deactivateEvent() event = EVENT_TARGET_DISCOVERED`
			No exception shall be thrown	SW - '90 00'	N1, N5

3		**Event deactivation - ConnectivityService**			

	1 - EVT_SEND_DATA(INS='10')	`HCIListener = ConnectivityListener HCIService = ConnectivityService Event has been successfully activated`
		
		`deactivateEvent() event = HCIListener.EVENT_HCI_TRA NSMISSION_FAILED`
			No exception shall be thrown	EVT_SEND_DAT A (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_REC EPTIONS_FAILED`
			No exception shall be thrown	EVT_SEND_DAT A (SW - '90 00')	N1, N3

	3 - EVT_SEND_DATA(INS='12')	`HCIListener = ConnectivityListener HCIService = ConnectivityService Event has been successfully activated`
		
		`deactivateEvent() event =  EVENT_STAND_BY`
			No exception shall be thrown	EVT_SEND_DAT A (SW - '90 00')	N1, N6

4		**Wrong event type**			

	EVT_SEND_DATA (INS = '16')	`HCIService = CardEmulationService Event has been successfully activated`
		
		`deactivateEvent() event value = 02`
			HCIException with reason code HCI_WRONG_EV ENT_TYPE shall be thrown	EVT_SEND_DAT A (SW - '90 00')	P1

5		**Deactivate an non-active event - CardEmulationService**			
	1 - EVT_SEND_DATA (INS = '10')	`HCIService = CardEmulationService Event was not activated before deactivateEvent() event = HCIListener.EVENT_HCI_TRA NSMISSION_FAILED`	HCIException with reason code HCI_WRONG_EV ENT_TYPE shall be thrown	EVT_SEND_DAT A (SW - '90 00')	P2

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

	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_DAT A (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_DAT A (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_DAT A (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_DAT A (SW - '90 00')	P2

6	**Deactivate an non-active event - ConnectivityService**				

	EVT_SEND_DATA(INS='12')	`HCIListener = ConnectivityListener HCIService = ConnectivityService Event was not activated before`
		
		`deactivateEvent() event = EVENT_STAND_BY`
			HCIException with reason code HCI_WRONG_EV ENT_TYPE shall be thrown	EVT_SEND_DAT A (SW - '90 00')	P2

7	**Deactivate an non-active event - ReaderService**				

	1 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.4.3 not applicable here - Send APDU (INS='10')	`HCIListener = ReaderListener HCIService = ReaderService Event was not activated before`
		
		`deactivateEvent() event = EVENT_GET_PARAMETER_RESPO NSE`
			HCIException with reason code HCI_WRONG_EV ENT_TYPE shall be thrown	EVT_SEND_DAT A (SW - '90 00')	P2

	2 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.4.3 not applicable here - Send APDU (INS='11')	`HCIListener = ReaderListener HCIService = ReaderService Event was not activated before`
		
		`deactivateEvent() event = EVENT_WRITE_EXCHANGE_DATA _RESPONSE`
			HCIException with reason code HCI_WRONG_EV ENT_TYPE shall be thrown	EVT_SEND_DAT A (SW - '90 00')	P2

Test case					
ID	**HCI Command**	**API Description**	**API Expectation**	**HCI Response**	**CRR**

	3 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.4.3 not applicable here - Send APDU (INS='12')	`HCIListener = ReaderListener HCIService = ReaderService Event was not activated before`
		
		`deactivateEvent() event = EVENT_TARGET_DISCOVERED`
			HCIException with reason code HCI_WRONG_EV ENT_TYPE shall be thrown	EVT_SEND_DAT A (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_DAT A (INS = '10')	`Precondition: HCIService = CardEmulationService Corresponding listener registered No event is activated`
		
		`requestCallbackNotification () event = HCIListener.EVENT_HCI_TRANSMISS ION_FAILED`
			throws HCIexception with reason code HCI_ACCESS_NOT_ GRANTED	EVT_SEND_DAT A (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		**Event notification status false - CardEmulationService**			

	1 - EVT_SEND_DATA (INS = '10')	`HCIService = CardEmulationService`
		
		`getEventNotificationStatus()`
		`event = HCIListener.EVENT_HCI_TRANSMISSION_FAILED`
			No exception shall be thrown.
			
			getEventNotificationStatus() = false
				EVT_SEND_DATA (SW - '90 00')	N1, N5

	2 - EVT_SEND_DATA (INS = '11')	`HCIService = CardEmulationService`
		
		`getEventNotificationStatus()`
		`event = HCIListener.EVENT_HCI_RECEPTIONS_FAILED`
			No exception shall be thrown.
			
			getEventNotificationStatus() = false
				EVT_SEND_DATA (SW - '90 00')	N1, N5

	3 - EVT_SEND_DATA (INS = '12')	`HCIService = CardEmulationService`
		
		`getEventNotificationStatus()`
		`event = CardEmulationListener.EVENT_GET_PARAMETER_RESPONSE`
			No exception shall be thrown.
			
			getEventNotificationStatus() = false
				EVT_SEND_DATA (SW - '90 00')	N1, N2

	4 - EVT_SEND_DATA (INS = '13')	`4 HCIService = CardEmulationService`
		
		`getEventNotificationStatus()`
		`event = CardEmulationListener.EVENT_FIELD_OFF`
			No exception shall be thrown.
			
			getEventNotificationStatus() = false
				EVT_SEND_DATA (SW - '90 00')	N1, N2

	5 - EVT_SEND_DATA (INS = '14')	`HCIService = CardEmulationService`
		
		`getEventNotificationStatus()`
		`event = CardEmulationListener.EVENT_ON_SEND_DATA`
			No exception shall be thrown.
			
			getEventNotificationStatus() = false
				EVT_SEND_DATA (SW - '90 00')	N1, N2

2		**Event notification status true - CardEmulationService**			

	EVT_SEND_DATA (INS = '15')	`HCIService = CardEmulationService`
		`activateEvent()`
		`event = CardEmulationListener.EVENT_ON_SEND_DATA`
		
		`getEventNotificationStatus()`
		`event = CardEmulationListener.EVENT_ON_SEND_DATA`
			No exception shall be thrown.
			
			getEventNotificationStatus() = true
				EVT_SEND_DATA (SW - '90 00')	N1, N2

3		**Event notification status true - ReaderService**			

	1 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.6.3 not applicable here - Send APDU (INS = '21')	`HCIService = ReaderService`
		`activateEvent()`
		`event = ReaderListener.EVENT_WRITE_EXCHANGE_DATA_RESPONSE`
		
		`getEventNotificationStatus()`
		`event = ReaderListener.EVENT_WRITE_EXCHANGE_DATA_RESPONSE`
			No exception shall be thrown.
			
			getEventNotificationStatus() = true
				SW - '90 00'	N1, N3

		Test case			
ID	**HCI Command**	**API Description**	**API Expectation**	**HCI Response**	**CRR**

	2 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.6.3 not applicable here - Send APDU (INS = '22')	`HCIService = ReaderService`
		
		`activateEvent()`
		`event =`
		`ReaderListener.EVENT_TARGET_`
		`DISCOVERED`
		
		`getEventNotificationStatus()`
		`event =`
		`ReaderListener.EVENT_TARGET_`
		`DISCOVERED`
			No exception shall be thrown.
			
			getEventNotificationStatus() = true
				SW - '90 00'	N1, N3

	3 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.6.3 not applicable here - Send APDU (INS = '23')	`HCIService = ReaderService`
		`activateEvent()`
		`event =`
		`ReaderListener.`
		`EVENT_GET_PARAMETER_RESPONSE`
		
		`getEventNotificationStatus()`
		`event =`
		`ReaderListener.`
		`EVENT_GET_PARAMETER_RESPONSE`
			No exception shall be thrown.
			
			getEventNotificationStatus() = true
				SW - '90 00'	N1, N3

	4 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.6.3 not applicable here - Send APDU (INS = '24')	`HCIService = ReaderService`
		`activateEvent()`
		`event =`
		`ReaderListener.`
		`EVENT_HCI_RECEPTIONS_FAILED`
		
		`getEventNotificationStatus()`
		`event =`
		`ReaderListener.`
		`EVENT_HCI_RECEPTIONS_FAILED`
			No exception shall be thrown.
			
			getEventNotificationStatus() = true
				SW - '90 00'	N1, N5

	5- Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.6.3 not applicable here - Send APDU (INS = '25')	`HCIService = ReaderService`
		`activateEvent()`
		`event =`
		`ReaderListener.`
		`EVENT_HCI_TRANSMISSION_FAILE`
		`D`
		
		`getEventNotificationStatus()`
		`event =`
		`ReaderListener.`
		`EVENT_HCI_TRANSMISSION_FAILE`
		`D`
			No exception shall be thrown.
			
			getEventNotificationStatus() = true
				SW - '90 00'	N1, N5

4		**Event notification status false - ReaderService**			

	1 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.6.3 not applicable here - Send APDU (INS='10')	`HCIListener = ReaderListener`
		`HCIService =`
		`ReaderServiceEvent has been`
		`successfully activated`
		
		`deactivateEvent()`
		`event = HCIListener.`
		`EVENT_HCI_TRANSMISSION_FAILE`
		`D`
			No exception shall be thrown.
			
			getEventNotificationStatus() = false
				SW - '90 00'	N1, N5

	2 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.6.3 not applicable here - Send APDU (INS = '11')	`HCIListener = ReaderListener`
		`HCIService =`
		`ReaderServiceEvent has been`
		`successfully activated`
		
		`deactivateEvent()`
		`event =`
		`EVENT_GET_PARAMETER_RESPONSE`
			No exception shall be thrown.
			
			getEventNotificationStatus() = false
				SW - '90 00'	N1, N3

	3 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.6.3 not applicable here - Send APDU (INS='12')	`HCIListener = ReaderListener`
		`HCIService =`
		`ReaderServiceEvent has been`
		`successfully activated`
		
		`deactivateEvent()`
		`event =`
		`EVENT_WRITE_EXCHANGE_DATA_RE`
		`SPONSE`
			No exception shall be thrown.
			
			getEventNotificationStatus() = false
				SW - '90 00'	N1, N3

			Test case		
ID	**HCI Command**	**API Description**	**API Expectation**	**HCI Response**	**CRR**

	4- Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.6.3 not applicable here - Send APDU (INS = '13')	`HCIListener = ReaderListener HCIService = ReaderServiceEvent has been successfully activated`
		
		`deactivateEvent() event = EVENT_TARGET_DISCOVERED`
			No exception shall be thrown.
			
			getEventNotificationStatus() = false
				SW - '90 00'	N1, N3

	5- Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.6.3 not applicable here - Send APDU (INS='14')	`HCIListener = ReaderListener HCIService = ReaderServiceEvent has been successfully activated`
		
		`deactivateEvent() event = EVENT_HCI_RECEPTION_FAILED`
			No exception shall be thrown.
			
			getEventNotificationStatus() = false
				SW - '90 00'	N1, N3

5			**Wrong event type**		

	1- EVT_SEND_DATA (INS = '16')	`HCIService = CardEmulationService`
		
		`getEventNotificationStatus() event = 0x02`
			HCIException with reason code HCI_WRONG_EVENT_TYPE shall be thrown	EVT_SEND_DATA (SW - '90 00')	P1

	2- Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.6.3 not applicable here - Send APDU (INS = '16')	`HCIService = ReaderService`
		
		`getEventNotificationStatus() event = 0x84`
			HCIException with reason code HCI_WRONG_EVENT_TYPE shall be thrown	SW - '90 00'	P1

	3 - EVT_SEND_DATA (INS = '16')	`HCIService = ConnectivityService`
		
		`getEventNotificationStatus() event = 0x02`
			HCIException with reason code HCI_WRONG_EVENT_TYPE shall be thrown	EVT_SEND_DATA (SW - '90 00')	P1

6			**Event notification status false - ConnectivityService**		

	1 - EVT_SEND_DATA (INS = '10')	`HCIListener = ConnectivityListener HCIService = ConnectivityService`
		
		`getEventNotificationStatus() event = HCIListener.EVENT_HCI_TRANSMISSION_FAILED`
			No exception shall be thrown.
			
			getEventNotificationStatus() = false
				EVT_SEND_DATA (SW - '90 00')	N1, N5

	2 - EVT_SEND_DATA (INS = '11')	`HCIListener = ConnectivityListener HCIService = ConnectivityService`
		
		`getEventNotificationStatus() event = HCIListener.EVENT_HCI_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

Test case					
ID	HCI Command	API Description	API Expectation	HCI Response	CRR
7	Event notification status true - ConnectivityService				
	EVT_SEND_DATA (INS = '15')	`HCIListener = ConnectivityListener` `HCIService = ConnectivityService` `activateEvent()` `event = ConnectivityListener.EVENT_S TAND_BY` `getEventNotificationStatus()` `event = ConnectivityListener.EVENT_S TAND_BY`	No exception shall be thrown. getEventNotificationStatus() = true	EVT_SEND_DAT A (SW - '90 00')	N1, N4

6.1.3 Interface HCIMessage

6.1.3.1 Method isHeading

Test Area Reference: Api_1_Hme_Mhd.

6.1.3.1.1 Conformance requirements

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

```
boolean isHeading()
```

6.1.3.1.1.1 Normal execution

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

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

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

<table>
<tr><th colspan="6">Test case</th></tr>
<tr><th>ID</th><th>HCI Command</th><th>API Description</th><th>API Expectation</th><th>HCI Response</th><th>CRR</th></tr>
<tr><td>1</td><td colspan="5">message is complete</td></tr>
<tr><td></td><td>EVT_SEND_DATA (INS ='01')</td><td>onCallback()
event =
EVENT_ON_SEND_DATA
HCIMessage.isHeading()</td><td>No exception shall be thrown.

isHeading() = true</td><td>EVT_SEND_DATA (SW - '90 00')

Result returned in the first response byte of the R-APDU data: b1 is set</td><td>N1,
N2,
N3,
N4,
N5</td></tr>
</table>

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

<table>
<tr><th>Applet Name</th><th>Test case ID</th></tr>
<tr><td>Api_1_Hme_Mco_1.java</td><td>1</td></tr>
</table>

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

<table>
<tr><th colspan="6">Test case</th></tr>
<tr><th>ID</th><th>HCI Command</th><th>API Description</th><th>API Expectation</th><th>HCI Response</th><th>CRR</th></tr>
<tr><td>1</td><td colspan="5">message is complete</td></tr>
<tr><td></td><td>EVT_SEND_DATA (INS = '01')</td><td>onCallback()
event =
EVENT_ON_SEND_DATA
HCIMessage.isComplete()</td><td>No exception shall be thrown.

isComplete() = true</td><td>EVT_SEND_DATA (SW - '90 00')

Result returned in the second response byte of the R-APDU data: b1 is set</td><td>N1</td></tr>
</table>

6.1.3.3 Method getType

Test Area Reference: Api_1_Hme_Mty.

6.1.3.3.1 Conformance requirements

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

```
byte getType()
```

6.1.3.3.1.1 Normal execution

- CRRN1: Returns the type of the incoming HCI message, it shall be one of constant values TYPE_defined in this interface.The returned type possibilities are: TYPE_COMMAND Indicates the HCI message type "command", TYPE_EVENT Indicates the HCI message type "event" and TYPE_RESPONSE indicates the HCI message type "response".

- CRRN2: In the case of a fragmented incoming message this method shall return the HCI message type coded in the first part of the HCI message.

6.1.3.3.1.2 Parameter errors

- None.

6.1.3.3.1.3 Context errors

- None.

6.1.3.3.2 Test Suite Files

Applet Name	Test case ID
Api_1_Hme_Mty_1.java	1
Api_1_Hme_Mty_2.java	2

6.1.3.3.3 Initial condition

- EVT_FIELD_ON has been sent on HCI interface.

- EVT_CARD_ACTIVATED has been sent on HCI interface.

- According applet has been successfully installed and selected using HCI interface.

6.1.3.3.4 Test procedure

			Test case		
ID	**HCI Command**	**API Description**	**API Expectation**	**HCI Response**	**CRR**
1			message of type event - Card emulation		

	EVT_SEND_DATA (INS = '01')	`onCallback()`
		`event = EVENT_ON_SEND_DATA`
		`HCIMessage.getType()`
			No exception shall be thrown.
			
			getType () = TYPE_EVENT
				EVT_SEND_DATA (SW - '90 00')
				
				third response byte of the R-APDU data is '40'
					N1

2			message of type event - Connectivity		

	- Send command on ISO interface to select applet; the initial conditions in clause 6.1.3.3.3 not applicable here
	
	- EVT_STANDBY
	- Send on ISO interface (INS = '01')
		`extends ConnectivityListener`
		`activateEvent()`
		`event = EVENT_STAND_BY`
		
		`onCallback()`
		`event = EVENT_STAND_BY`
		`HCIMessage.getType()`
			No exception shall be thrown.
			
			getType () = TYPE_EVENT
				SW = '90 00'	N1

6.1.3.4 Method getInstruction

Test Area Reference: Api_1_Hme_Min.

6.1.3.4.1 Conformance requirements

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

```
byte getInstruction()
```

6.1.3.4.1.1 Normal execution

- CRRN1: returns the instruction part of the HCP message header of the current incoming HCP message, see ETSI TS 102 622 [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

ID	HCI Command	API Description	API Expectation	HCI Response	CRR
		Test case			
1		message of type event: Card emulation			
	EVT_SEND_DATA (INS = '01')	`onCallback() event = EVENT_ON_SEND_DATA HCIMessage.getInstruction()`	No exception shall be thrown. getInstruction() = '10'	EVT_SEND_DATA (SW - '90 00') the 4th response byte of the R-APDU data is '10'	N1
2		message of type response: Card emulation			
	- EVT_SEND_DATA (INS = '01') - ANY_OK(ATQA) - EVT_SEND_DATA (arbitrary data)	`onCallback() prepareAndSendGetParameterComman d() 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_PARAM TER(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	Get received message length: CardEmulationService				

	EVT_SEND_DATA: Send message with the length '0A' message is complete	`onCallback()`
		`event = EVENT_ON_SEND_DATA`
		`HCIMessage.getReceiveLength()`
			No exception shall be thrown.
			
			getReceiveLength() = '0A'
				EVT_SEND_DATA (SW - '90 00')
				
				Result returned in the 7th and 8th response byte of the R-APDU data ('00 0A')
					N1

6.1.3.7   Method getReceiveBuffer

Test Area Reference: Api_1_Hme_Mrb.

6.1.3.7.1   Conformance requirements

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

```
byte[] getReceiveBuffer()
```

6.1.3.7.1.1   Normal execution

- CRRN1: returns a reference to the underlying HCI message receive buffer. The length of this buffer is system dependent; it is guaranteed to be at least 270 bytes.

- CRRN2: it returns the buffer holding the current HCI message.

- CRRN3: the content of the array starts at the location indicated by getReceiveOffset() and has the length indicated by getReceiveLength().

6.1.3.7.1.2   Parameter errors

- None.

6.1.3.7.1.3   Context errors

- None.

6.1.3.7.2   Test suite files

Applet Name	Test case ID
Api_1_Hme_Mrb_1.java	1

6.1.3.7.3   Initial condition

- EVT_FIELD_ON has been sent on HCI interface.

- EVT_CARD_ACTIVATED has been sent on HCI interface.

- According applet has been successfully installed and selected using HCI interface.

6.1.3.7.4   Test procedure

<table>
<tr><th colspan="6">Test case</th></tr>
<tr><th>ID</th><th>HCI Command</th><th>API Description</th><th>API Expectation</th><th>HCI Response</th><th>CRR</th></tr>
<tr><td>1</td><td colspan="5" align="center">Message length < 270 bytes</td></tr>
<tr>
<td></td>
<td>EVT_SEND_DATA (message with the length = 09: '00 01 02 03 04 05 06 07 08')</td>
<td><code>onCallback()</code>
<code>event = EVENT_ON_SEND_DATA</code>
<code>getReceiveBuffer()</code>
<code>getReceiveOffset()</code>
<code>getReceiveLength()</code></td>
<td>No exception shall be thrown.
First 9 bytes of getReceiveBuffer() shall be '00 01 02 03 04 05 06 07 08 '</td>
<td>EVT_SEND_DATA ()

First 9 bytes of the response data shall be '00 01 02 03 04 05 06 07 08'</td>
<td>N1, N2, N3</td>
</tr>
</table>

6.1.4   Class HCIException

6.1.4.1   Method throwIt

Test Area Reference: Api_1_Hxp_Trw.

6.1.4.1.1   Conformance requirements

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

```
public static void throwIt(short reason)
                 throws HCIException
```

6.1.4.1.1.1   Normal execution

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

6.1.4.1.1.2   Parameter errors

- None.

6.1.4.1.1.3   Context errors

- None.

6.1.4.1.2   Test suite files

Applet Name	Test case ID
Api_1_Hxp_Trw_1.java	1

6.1.4.1.3   Initial conditions

- EVT_FIELD_ON has been sent on HCI interface.

- EVT_CARD_ACTIVATED has been sent on HCI interface.

- According applet has been successfully installed and selected using HCI Interface.

6.1.4.1.4 Test procedure

		Test Case			
ID	**HCI Command**	**API Description**	**API Expectation**	**HCI Response**	**CRR**
1		**Throw Exception**			
	1 - EVT_SEND_DATA (INS = '01')	`throwIt()` `reason = HCI_ACCESS_NOT_GRANTED`	Throw exception with corresponding reason code	EVT_SEND_DATA (SW - '90 00')	N1
	2 - EVT_SEND_DATA (INS = '02')	`throwIt()` `reason = HCI_CONDITIONS_NOT_SATISFIED`	Throw exception with corresponding reason code	EVT_SEND_DATA (SW - '90 00')	N1
	3 - EVT_SEND_DATA (INS = '03')	`throwIt()` `reason =   HCI_CURRENTLY_DISABLED`	Throw exception with corresponding reason code	EVT_SEND_DATA (SW - '90 00')	N1
	4 - EVT_SEND_DATA (INS = '04')	`throwIt()` `reason = HCI_FRAGMENTED_MESSAGE_ONGOING`	Throw exception with corresponding reason code	EVT_SEND_DATA (SW - '90 00')	N1
	5 - EVT_SEND_DATA (INS = '05')	`throwIt()` `reason =   HCI_INVALID_LENGTH`	Throw exception with corresponding reason code	EVT_SEND_DATA (SW - '90 00')	N1
	6 - EVT_SEND_DATA (INS = '06')	`throwIt()` `reason = HCI_LISTENER_ALREADY_REGISTERED`	Throw exception with corresponding reason code	EVT_SEND_DATA (SW - '90 00')	N1
	7 - EVT_SEND_DATA (INS = '07')	`throwIt()` `reason = HCI_NOT_AVAILABLE`	Throw exception with corresponding reason code	EVT_SEND_DATA (SW - '90 00')	N1
	8 - EVT_SEND_DATA (INS = '08')	`throwIt()` `reason = HCI_RESOURCES_NOT_AVAILABLE`	Throw exception with corresponding reason code	EVT_SEND_DATA (SW - '90 00')	N1
	9 - EVT_SEND_DATA (INS = '09')	`throwIt()` `reason = HCI_SERVICE_NOT_AVAILABLE`	Throw exception with corresponding reason code	EVT_SEND_DATA (SW - '90 00')	N1
	10 - EVT_SEND_DATA (INS = '0A')	`throwIt()` `reason = HCI_WRONG_EVENT_TYPE`	Throw exception with corresponding reason code	EVT_SEND_DATA (SW - '90 00')	N1
	11 - EVT_SEND_DATA (INS = '0B')	`throwIt()` `reason = HCI_WRONG_LISTENER_TYPE`	Throw exception with corresponding reason code	EVT_SEND_DATA (SW - '90 00')	N1

6.1.5 Interface HCIListener

6.1.5.1 Method onCallback

Test Area Reference: Api_1_Hln_Ocb.

6.1.5.1.1 Conformance requirements

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

```
void onCallback(byte event, HCIMessage message)
```

6.1.5.1.1.1   Normal execution

- CRRN1: The Contactless Framework shall invoke the method `*typeListener.onCallback` only with events which are defined in this particular listener or `HCIListener` and only with message objects of this listener type.

- CRRN2: The Applet instance shall activate the events with CardEmulationService.activateEvent before it will receive any event notification.

- CRRN3: The context as defined in the Java Card™ specification [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 ETSI 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			**Get context**		
	1 - Send EVT_SEND_DATA (Select applet) Send EVT_SEND_DATA (INS = '01'))	`Service = CardEmulationService` `Listener =` `CardEmulationListener` `process():` `JCSystem.getAID();`	No exception shall be thrown expected AID = current AID	EVT_SEND_DATA (SW '90 00')	N3
	2 - Send EVT_SEND_DATA (Select applet) Send EVT_SEND_DATA (INS = '02')	`Service = CardEmulationService` `Listener =` `CardEmulationListener` `process():` `JCSystem.getPreviousContextAID(` `);`	No exception shall be thrown previous AID = null	EVT_SEND_DATA (SW '90 00')	N3
2			**Abort transaction**		
	- Send EVT_SEND_DATA (Select applet) - Send EVT_SEND_DATA (INS = '01') - Send ENVELOPE command on the ISO interface which will trigger EVENT_UNRECOGNIZED_ENVELOPE in the applet as specified in ETSI TS 102 241 [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 modifed	EVT_SEND_DATA (SW '90 00') EVT_SEND_DATA (SW '90 00') SW - '90 00'	N4
3			**Data link layer recovery**		
	- Send EVT_SEND_DATA (Select applet) - Send EVT_SEND_DATA (INS = '01') - reset the underlying data link layer while receiving the reponse	`Service = CardEmulationService` `Listener =` `CardEmulationListener` `activateEvent(EVENT_ON_SEND_DAT` `A) during the installation` `onCallback()` `prepareAndSendSendDataEvent()` ` data = {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
4			**Receive messages while waiting for get parameter**		
	- EVT_SEND_DATA (Select applet) - EVT_FIELD_OFF before ANY_OK	`Listener =` `CardEmulationListener` `activateEvent()` `event =` `EVENT_GET_PARAMETER_RESPONSE` `prepareAndSendGetParameterComma` `nd()` `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

Test case					
ID	HCI Command	API Description	API Expectation	HCI Response	CRR
5	Receive messages while waiting for get parameter				

	-
	EVT_SEND_DAT
	A (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.

6.2.1.1.1.1.3 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		Get parameter: Type A			
	1 - EVT_SEND_DATA (Sele	`prepareAndSendGetParameterCommand()` `paramID = PARAM_ID_TYPE_A_CARD_APPLICATION_DATA`	No exception shall be thrown	GET_PARAMETER with command parameter as indicated in the API Description column	N1, N2, N3
	2 - EVT_SEND_DATA (Select applet)	`prepareAndSendGetParameterCommand()` `paramID = PARAM_ID_TYPE_A_CARD_ATQA`	No exception shall be thrown	GET_PARAMETER with command parameter as indicated in the API Description column	N1, N2, N3
	3 - EVT_SEND_DATA (Select applet)	`prepareAndSendGetParameterCommand()` `paramID = PARAM_ID_TYPE_A_CARD_CID_SUPPORT`	No exception shall be thrown	GET_PARAMETER with command parameter as indicated in the API Description column	N1, N2, N3
	4 - EVT_SEND_DATA (Select applet)	`prepareAndSendGetParameterCommand()` `paramID = PARAM_ID_TYPE_A_CARD_CLT_SUPPORT`	No exception shall be thrown	GET_PARAMETER with command parameter as indicated in the API Description column	N1, N2, N3
	5 - EVT_SEND_DATA (Select applet)	`prepareAndSendGetParameterCommand()` `paramID = PARAM_ID_TYPE_A_CARD_DATARATE_MAX`	No exception shall be thrown	GET_PARAMETER with command parameter as indicated in the API Description column	N1, N2, N3
	6 - EVT_SEND_DATA (Select applet)	`prepareAndSendGetParameterCommand()` `paramID = PARAM_ID_TYPE_A_CARD_FWI_SFGI`	No exception shall be thrown	GET_PARAMETER with command parameter as indicated in the API Description column	N1, N2, N3

			Test case			
ID	**HCI Command**	**API Description**	**API Expectation**	**HCI Response**	**CRR**	
	7 - EVT_SEND_DAT A (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_DAT A (Select applet)	`prepareAndSendGetParameterCommand( )` `paramID = PARAM_ID_TYPE_A_CARD_SAK`	No exception shall be thrown	GET_PARAMETER with command parameter as indicated in the API Description column	N1, N2, N3	
2			**Get parameter: Type B**			
	1- EVT_SEND_DAT A (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_DAT A (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_DAT A (Select applet)	`prepareAndSendGetParameterCommand( )` `paramID = PARAM_ID_TYPE_B_CARD_DATARATE_MAX`	No exception shall be thrown	GET_PARAMETER with command parameter as indicated in the API Description column	N1, N2, N3	
	4 - EVT_SEND_DAT A (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_DAT A (Select applet)	`prepareAndSendGetParameterCommand( )` `paramID = PARAM_ID_TYPE_B_CARD_MODE`	No exception shall be thrown	GET_PARAMETER with command parameter as indicated in the API Description column	N1, N2, N3	
3			**The Contactless Framework is receiving fragmented HCI Message**			
	1 – EVT_SEND_DAT A (Select applet) Send as fragmented HCI Message: - EVT_SEND_DAT A (data length = supported buffer size + 5 bytes, message not complete) With CB = 1 for the last sent frame EVT_SEND_DAT A sent without waiting for response	`prepareAndSendGetParameterCommand( )` `paramID = PARAM_ID_TYPE_A_CARD_ATQA`	Shall throw uicc.hci.framework.HCIE xception with error code HCI_FRAGMENTED_M ESSAGE_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) Send as fragmented HCI Message: EVT_SEND_DATA (data length = supported buffer size + 5 bytes, message not complete) EVT_SEND_DATA sent without waiting for response	`prepareAndSendGetParameterCommand( )` `paramID = PARAM_ID_TYPE_B_CARD_ATQB`	Shall throw uicc.hci.framework.HCIException with error code HCI_FRAGMENTED_MESSAGE_ONGOING	- EVT_SEND_DATA (SW – '90 00') - EVT_SEND_DATA (No data) - EVT_SEND_DATA (one byte with second least significant bit set)	C1

6.2.1.1.2 Method prepareAndSendSendDataEvent

Test Area Reference: Api_2_CEm_Ssd.

6.2.1.1.2.1 Conformance requirements

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

```
void prepareAndSendSendDataEvent(byte[] data,
                                 short offset,
                                 short len)
                          throws HCIException,
                                 java.lang.NullPointerException,
                                 java.lang.ArrayIndexOutOfBoundsException
```

6.2.1.1.2.1.1 Normal execution

- CRRN1: This non-blocking method builds the the HCI event EVT_SEND_DATA sends it to the Contactless Card Emulation Gate of the CLF.

- CRRN2: the data to be sent should be formatted according to the HCI specification, the offset define the offset of data into the data buffer and the len define the length of data in the data buffer.

- CRRN3: Applet instances shall receive CardEmulationMessages after the registration of a CardEmulationListener interface to a CardEmulationService only if the EVENT_ON_SEND_DATA is activated for the Applet instance.

- CRRN4: The CardEmulationListener.onCallback method shall be called by the Contactless Framework. The HCI message that resulted in the selection of this Applet according to the rules defined in "GlobalPlatform Amendment C" [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			Send data - Type A		

	EVT_SEND_DATA (Select applet)	`prepareAndSendSendDataEvent( )`
		`data = {0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09}`
		`offset = 2`
		`length = 5`
			No exception shall be thrown	EVT_SEND_DATA ({0x03, 0x04, 0x05, 0x06, 0x07})	N1, N2, N3, N4

2			Send data - Type B		

	EVT_SEND_DATA (Select applet)	`prepareAndSendSendDataEvent( )`
		`data = {0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09}`
		`offset = 2`
		`length = 5`
			No exception shall be thrown	EVT_SEND_DATA ({0x03, 0x04, 0x05, 0x06, 0x07})	N1, N2, N3, N4

3			Recption of fragmented HCI Message		

	- EVT_SEND_DATA to select applet Send as fragmented HCI Message:
	- EVT_SEND_DATA (data length = supported buffer size + 5 bytes, message not complete)
	- EVT_SEND_DATA (INS = '01') without waiting for response
	
	- Send EVT_SEND_DATA (INS = '01')
		`prepareAndSendSendDataEvent( )`
		`data = {0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09}`
		`offset = 2`
		`length = 5`
			Shall throw uicc.hci.framework.HCIException with error code HCI_FRAGMENTED _MESSAGE_ONGOING	- EVT_SEND_DATA (SW - '90 00')
				- EVT_SEND_DATA (No data)
				
				- EVT_SEND_DATA (SW- '90 02')
					C2

Test case					
ID	HCI Commnad	API Description	API Expectation	HCI Response	CRR
4	Send no data				
	EVT_SEND_DATA (Select applet)	`prepareAndSendSendDataEvent( )` `data = null` `offset = 2` `length = 5`	Shall throw java.lang exception with error code NullPointerException	EVT_SEND_DATA ({0xFF, 0xFF, 0xFF, 0xFF, 0xFF })	P1
5	Array Index Out Of Bounds				
	1 - EVT_SEND_DATA (Select applet)	`prepareAndSendSendDataEvent( )` `1 - data = {0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09}` `offset = 2` `length = 15`	Shall throw java.lang exception with error code ArrayIndexOutOfBou ndsException	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 ArrayIndexOutOfBou ndsException	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			Select Applet		
	EVT_SEND_DATA (Select applet)	`selectingMessage()`	No exception shall be thrown. selectingMessage()= true	EVT_SEND_DATA (SW - '90 00')	N1, N2
2			Arbitrary Command		
	- EVT_SEND_DATA (Select applet) - EVT_SEND_DATA (INS = '01')	`selectingMessage()`	No exception shall be thrown. selectingMessage() = false	- EVT_SEND_DATA(SW - '90 00') - EVT_SEND_DATA ('01 02 03 90 00')	N1, N2

6.2.1.2 Interface CardEmulationService

6.2.1.2.1 Method getCardRFType

Test Area Reference: Api_2_CEs_RFt.

6.2.1.2.1.1 Conformance requirements

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

```
byte getCardRFType()
```

6.2.1.2.2.1.1 Normal execution

- CRRN1: returns the reader RF type on which the service is connected, shall be one of the TYPE_* constant values defined in this interface, or -1 if the RF type can not be evaluated.

- CRRN2: Applets communicating through the process() method shall also be able to use the API services defined in the present document which do not require a CardEmulationListener registration (e.g. requesting the power mode or connectivity service).

6.2.1.2.2.1.2 Parameter errors

- None.

6.2.1.2.2.1.3 Context errors

- None.

6.2.1.2.2 Test Suite Files

Applet Name	Test case ID
Api_2_CEs_RFt_1.java	1
Api_2_CEs_RFt_1.java	2
Api_2_CEs_RFt_1.java	3
Api_2_CEs_RFt_1.java	4

6.2.1.2.3 Initial conditions

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

- EVT_FIELD_ON has been sent on HCI interface.

- EVT_CARD_ACTIVATED has been sent on HCI interface.

- According applet has been successfully installed and selected using HCI interface.

6.2.1.2.3.1 Test procedure

Test case					
ID	HCI Command	API Description	API Expectation	HCI Response	CRR
1	RF type A				
	EVT_SEND_DATA (INS = '01')	`getCardRFType()`	No exception shall be thrown getCardRFType() = TYPE_A_CARD_RF	EVT_SEND_DATA (SW = '90 01')	N1 N2
2	RF type B				
	EVT_SEND_DATA (INS = '01')	`getCardRFType()`	No exception shall be thrown getCardRFType() = TYPE_B_CARD_RF	EVT_SEND_DATA (SW = '90 02')	N1 N2
3	RF type F				
	EVT_SEND_DATA (INS = '01')	`getCardRFType()`	No exception shall be thrown getCardRFType() = TYPE_F_CARD_RF	EVT_SEND_DATA (SW = '90 04')	N1 N2
4	RF type B'				
	EVT_SEND_DATA (INS = '01')	`getCardRFType()`	No exception shall be thrown getCardRFType() = TYPE_B_PRIM_CARD_RF	EVT_SEND_DATA (SW = '90 03')	N1 N2

6.2.1.3 Interface CardEmulationListener

6.2.1.3.1 Method onCallback

Test Area Reference: Api_2_CEl_Ocb.

6.2.1.3.1.1 Conformance requirements

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

```
void onCallback(byte event, HCIMessage message)
```

6.2.1.3.1.1.1 Normal execution

- CRRN1: The Contactless Framework shall invoke the method CardEmulationListener.onCallback only with events which are defined in this particular listener or CardEmulationListener and only with CardEmulationMessage objects.

- CRRN2: The Applet instance shall activate the events with CardEmulationService.activateEvent before it will receive any event notification. In case the Applet instance has registered the *CardEmulationListener* and has activated the EVENT_ON_SEND_DATA the *process()* method of this Applet instance shall not be invoked during the selection. The *CardEmulationListener.onCallback* method shall be called by the Contactless Framework.

- CRRN3: If the EVENT_ON_SEND_DATA is deactivated for the Applet instance and an APDU is received via the EVT_SEND_DATA, the javacard.framework.APDU class and the process() method of the Applet instance shall be invoked.

- CRRN4: It shall not be possible to switch between the usage of the CardEmulationListener interface and the invocation through the process() method within a contactless application session, i.e. not before the Applet has been deselected and selected again.

6.2.1.3.1.1.2 Parameter errors

- None.

6.2.1.3.1.1.3 Context errors

- None.

6.2.1.3.1.2 Test Suite Files

Applet Name	Test case ID
Api_2_CEI_Ocb_1.java	1
Api_2_CEI_Ocb_4.java	3
Api_2_CEI_Ocb_1.java	4
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	Activate EVT_SEND_DATA - Type A				
	- EVT_SEND_DATA(INS = '14') - EVT_SEND_DATA(INS = '24')- - deselect the applet - select the applet	`HCIListener = CardEmulationListener HCIService = CardEmulationService activateEvent() event = CardEmulationListener.EVENT_ON_SEND_DATA`	- process() method: No exception shall be thrown - process() method: No exception shall be thrown onCallback() method shall notice the event EVENT_ON_SEND_DATA	- EVT_SEND_DATA (SW - '90 00') - EVT_SEND_DATA (SW - '90 00') - EVT_SEND_DATA (SW - '90 00')	N1, N2, N3, N4,
	EVENT_HCI_RECEPTIONS_FAILED - Type A - FFS				
2					
3	Activate EVENT_GET_PARAMETER_RESPONSE - Type A				
	- EVT_SEND_DATA (INS = '01') - ANY_OK as response to GET_PARAMETER - deselect/select the applet - EVT_SEND_DATA(INS='22')	`Listener = CardEmulationListener activateEvent() event = EVENT_GET_PARAMETER_RESPONSE prepareAndSendGetParameterCommand() paramID = PARAM_ID_TYPE_A_CARD_ATQA`	onCallback() method shall notice the event EVENT_ON_SEND_DATA onCallback() shall receive ANY_OK as EVENT_GET_PARAMETER_RESPONSE -process() method No exception shall be thrown - process() method: ANY_OK reception is verified	GET_PARAMETER with the PARAM_ID_TYPE_A_CARD_ATQA - No exception after ANY_OK reception. EVT_SEND_DATA (SW - '90 00') - EVT_SEND_DATA (SW - '90 00') - EVT_SEND_DATA (SW - '90 00')	N1, N2, N3, N4,

ETSI

ID	HCI Command	API Description	API Expectation	HCI Response	CRR
			Test case		
4			**Activate EVT_SEND_DATA - Type B**		

	- EVT_SEND_DATA(INS = '14')
	
	- EVT_SEND_DATA(INS = '24')-
	
	- deselect the applet
	- select the applet
		`HCIListener = CardEmulationListener`
		`HCIService = CardEmulationService`
		`activateEvent()`
		`event = CardEmulationListener.EVENT_ON_SEND_DATA`
			- process() method: No exception shall be thrown
			
			- process() method: No exception shall be thrown
			
			onCallback() method shall notice the event EVENT_ON_SEND_DATA
				- EVT_SEND_DATA (SW - '90 00')
				
				- EVT_SEND_DATA (SW - '90 00')
				
				- EVT_SEND_DATA (SW - '90 00')
					N1, N2, N3, N4,

5			**EVENT_HCI_RECEPTIONS_FAILED - Type B - FFS**		
6			**Activate EVENT_GET_PARAMETER_RESPONSE - Type B**		

	- EVT_SEND_DATA (INS = '01')
	
	- ANY_OK as response to GET_PARAMETER
	
	- deselect/select the applet
	
	- EVT_SEND_DATA(INS = '22')
		`Listener = CardEmulationListener`
		`activateEvent()`
		`event = EVENT_GET_PARAMETER_RESPONSE`
		
		`prepareAndSendGetParameterCommand()`
		`paramID = PARAM_ID_TYPE_B_CARD_ATQB`
			onCallback() method shall notice the event EVENT_ON_SEND_DATA
			
			onCallback() shall receive ANY_OK as EVENT_GET_PARAMETER_RESPONSE
			
			-process() method No exception shall be thrown
			
			- process() method: ANY_OK reception is verified
				- GET_PARAMETER with the PARAM_ID_TYPE_B_CARD_ATQB
				
				- No exception after ANY_OK reception. EVT_SEND_DATA (SW - '90 00')
				
				- EVT_SEND_DATA (SW - '90 00')
				
				- EVT_SEND_DATA (SW - '90 00')
					N1, N2, N3, N4

7			**Deactivate events**		

	1 - 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_DAT A(INS='12')
	
	- ANY_OK as response to GET_PARAMET ER
	
	- deselect/select the applet
	
	- EVT_SEND_DAT A(INS='20')
		`HCIService = CardEmulationService` Event has been successfully activated
		
		`deactivateEvent()` `event = CardEmulationListener.EVENT_GET _PARAMETER_RESPONSE`
		
		`prepareAndSendGetParameterComma nd()` `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_DAT A(INS='12')
	
	- ANY_OK as response to GET_PARAMET ER
	
	- deselect/select the applet
	
	- EVT_SEND_DAT A(INS='20')
		`HCIService = CardEmulationService` Event has been successfully activated
		
		`deactivateEvent()` `event = CardEmulationListener.EVENT_GET _PARAMETER_RESPONSE`
		
		`prepareAndSendGetParameterComma nd()` `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_DAT A(INS='14') - deselect /select the applet - EVT_SEND_DAT A(INS='20')	`HCIService = CardEmulationService` Event has been successfully activated `deactivateEvent()` `event = CardEmulationListener.EVENT_ON_ SEND_DATA`	No exception shall be thrown onCallback() method shall not notice the event	- EVT_SEND_DATA (SW - '90 00') - EVT_SEND_DATA (SW - '90 00') - EVT_SEND_DATA (SW- '90 01')	N2

6.2.2 Package Connectivity Service

6.2.2.1 Interface ConnectivityService

6.2.2.1.1 Method prepareAndSendConnectivityEvent

Test Area Reference: Api_2_CNs_Sce.

6.2.2.1.1.1 Conformance requirements

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

```
void prepareAndSendConnectivityEvent()
                        throws HCIException
```

6.2.2.1.1.1.1 Normal execution

- CRRN1: This non-blocking method builds the HCI event EVT_CONNECTIVITY which notifies the terminal host that it shall send a "HCI connectivity event" as defined in ETSI TS 102 223 [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 ETSI TS 102 241 [6] is then available to the Applet when that Applet instance is triggered with the processToolkit() method defined in ETSI 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 in card emulation mode or when this Applet instance is in the state ACTIVATED (according to "GlobalPlatform Amendment C" [10]) for the reader mode.

- CRRN5: The ProactiveHandler defined in ETSI 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
Api_2_CNs_Sce_7.java	7

6.2.2.1.1.3 Initial condition

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

- Prior to this test the Terminal shall have been powered on and performed the PROFILE DOWNLOAD procedure.

- In terminal profile set the 25th byte, 'b6' to indicate that class m is supported.

- In terminal profile set the 5th byte, 'b1' to indicate proactive UICC: SET UP EVENT LIST.

- UICC has created and opened a pipe to connectivity gate in the terminal host

- EVT_FIELD_ON has been sent on HCI interface.

- EVT_CARD_ACTIVATED has been sent on HCI interface.

- According applet has been successfully installed and selected using HCI interface.

Initial conditions for ID 2-1:

- Prior to this test the Terminal shall have been powered on and performed the PROFILE DOWNLOAD procedure.

- In terminal profile set 5th byte, 'b1' to indicate proactive UICC: SET UP EVENT LIST.

- In terminal profile the following CAT facilities shall be supported:

    - Set the 25th byte, 'b6' to indicate that class m is supported

    - Set the 26th byte, 'b2' and the 31st byte, 'b1' to indicate that class r is supported.

- Contactless functionality state is disabled in the UICC as defined in ETSI TS 102 223 [7].

Initial conditions for ID 2-2:

- Prior to this test the Terminal shall have been powered on and performed the PROFILE DOWNLOAD procedure.

- In terminal profile set the 25th byte, 'b6' to indicate that class m is supported.

- In terminal profile set the 5th byte, 'b1' to indicate proactive UICC: SET UP EVENT LIST.

Initial conditions for ID 7:

- Prior to this test the Terminal shall have been powered on and performed the PROFILE DOWNLOAD procedure.

- In terminal profile set the 25th byte, 'b6' to indicate that class m is supported.

- In terminal profile set the 5th byte, 'b1' to indicate proactive UICC: SET UP EVENT LIST.

- UICC has created and opened a pipe to connectivity gate in the terminal host

- According applet has been successfully installed and selected using ISO Interface and the applet is set into ACTIVATED state.

6.2.2.1.1.4 Test procedure

Test case					
ID	HCI commands	API Description	API Expectation	HCI Response	CRR
1	Send HCI Connectivity Event (card emulation)				
	- Send EVT_SEND_DATA (INS = '01')	`Listener = CardEmulationListener process(): prepareAndSendConnectivityEvent()`	No exception shall be thrown	EVT_SEND_DATA (SW – '90 00') HCI event EVT_CONNECTIVITY sent. This may be sent before or after the EVT_SEND_DATA indicated above.	N1 N3

ID	HCI commands	API Description	API Expectation	HCI Response	CRR
			Test case		
2			**HCI interface disabled**		

	1
	 Send on ISO interface:
	 - Select applet
	 - Send APDU
	 (INS = '02')
		`Listener = CardEmulationListener`
		 `process():`
		 `prepareAndSendConnectivityEvent()`
			throw HCIException with error code reason HCI_CURRENTLY_DISABLED	SW - '90 00'
				 No EVT_CONNECTIVITY shall be sent
					C1
					 N3

	2 - The contactless interface is disabled in the UICC as defined in Global Platform Amendment C Send on ISO interface send the following commands:
	 - Send APDU to select the applet.
	 - Send APDU (INS = '02')
	 - Postcondition: The contactless interface is enabled again in the UICC as defined in Global Platform Amendment C
		`setCommunicationInterface() API`
		 `method of  Global Platform`
		 `Amendment C [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			**Wrong Precondition Connectivity Event**		

	The initial conditions in clause 6.2.2.1.1.3 related to card emulation are not applicable here.
	 Send on ISO:
	 - Select applet
	 - Send
	 APDU with
	 (INS = '01')
		`Service = ConnectivityService`
		 `Listener = ConnectivityListener`
		
		 `process()`
		 `ConnectivityService.`
		 `prepareAndSendConnectivityEvent()`
			Shall throw exception with error code reason HCI_CONDITIONS_NOT_SATISFIED	No EVT_CONNECTIVITY shall be sent	N4, C3

4			**No Proactive Handler**		

	- Send EVT_SEND_DATA (INS = '01')	`Service = CardEmulationService`
		 `Listener = CardEmulationListener`
		
		 `process()`
		 `ProactiveHandlerSystem.getTheHandler()`
			ProactiveHandler shall not be available	EVT_SEND_DATA (SW – '90 00')	N5

5			**No Proactive Handler onCallback**		

	- Send EVT_SEND_DATA (INS = '01')	`Service = CardEmulationService`
		 `Listener = CardEmulationListener`
		 `activateEvent(EVENT_ON_SEND_DATA)`
		 `during the installation`
		
		 `onCallback()`
		 `ProactiveHandlerSystem.getTheHandler()`
			ProactiveHandler shall not be available	EVT_SEND_DATA (SW – '90 00')	N5

		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
		`Listener = CardEmulationListener`
		`process():`
		`prepareAndSendConnectivityEvent()`
		`register for`
		`EVENT_EVENT_DOWNLOAD_HCI_CONNECTIVITY`
		`TY`
		
		`processToolkit():`
		`use arbitrary method of the UICC`
		`API (ETSI TS 102 241 [6])`
			No exception shall be thrown	-
				EVT_SEND_DATA (SW – '90 00')
				
				- HCI event EVT_CONNECTIVITY sent. This may be sent before or after the EVT_SEND_DATA indicated above.
				
				- response to ENVELOPE
					N1
					N2 N3

7		**Send HCI Connectivity Event (reader mode)**			
	- Send APDU(INS = '01') on ISO interface	`process():` `prepareAndSendConnectivityEvent()`	No exception shall be thrown	HCI event EVT_CONNECTIVITY sent. - SW - '90 00'	N1 N3

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

Test Area Reference: Api_2_CNs_Ste.

6.2.2.1.2.1 Conformance requirements

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

```
void prepareAndSendTransactionEvent(byte [] aid,
                                    short aidOffset,
                                    short aidLen,
                                    byte[] parameters,
                                    short parametersOffset,
                                    short parametersLen)
                                    throws HCIException
                                            java.lang.ArrayIndexOutOfBoundsException,
                                            java.lang.NullPointerException
```

6.2.2.1.2.1.1 Normal execution

- CRRN1: this non-blocking method builds the HCI event EVT_TRANSACTION which notifies the terminal host that it shall launch an application on the terminal which is associated to an Applet in the UICC host identified.

- CRRN2: the contactless runtime environment shall bind the services defined in uicc.hci.services.connectivity to the corresponding resources (e.g. gates and pipes) specified by the HCI protocol [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 in card emulation mode or when this Applet instance 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 ETSI 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_6.java	6

6.2.2.1.2.3 Initial condition

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

- Prior to this test the Terminal shall have been powered on and performed the PROFILE DOWNLOAD procedure.

- In terminal profile set the 25th byte, 'b6' to indicate that class m is supported.

- In terminal profile set the 5th byte, 'b1' to indicate proactive UICC: SET UP EVENT LIST.

- UICC has created and opened a pipe to connectivity gate in the terminal host

- EVT_FIELD_ON has been sent on HCI interface.

- EVT_CARD_ACTIVATED has been sent on HCI interface.

- According applet has been successfully installed and selected using HCI interface.

Initial conditions for ID 2-1:

- Prior to this test the Terminal shall have been powered on and performed the PROFILE DOWNLOAD procedure.

- In terminal profile set 5th byte, 'b1' to indicate proactive UICC: SET UP EVENT LIST.

- In terminal profile the following CAT facilities shall be supported :

  - Set the 25th byte, 'b6' to indicate that class m is supported

  - Set the 26th byte, 'b2' and the 31st byte, 'b1' to indicate that class r is supported.

- Contactless functionality state is disabled in the UICC as defined in ETSI TS 102 223 [7].

Initial conditions for ID2-2:

- Prior to this test the Terminal shall have been powered on and performed the PROFILE DOWNLOAD procedure.

- In terminal profile set the 25th byte, 'b6' to indicate that class m is supported.

- In terminal profile set the 5th byte, 'b1' to indicate proactive UICC: SET UP EVENT LIST.

Initial conditions for ID 6:

- Prior to this test the Terminal shall have been powered on and performed the PROFILE DOWNLOAD procedure.

- In terminal profile set the 25th byte, 'b6' to indicate that class m is supported.

- In terminal profile set 5th byte, 'b1' to indicate proactive UICC: SET UP EVENT LIST.

- UICC has created and opened a pipe to connectivity gate in the terminal host

- According applet has been successfully installed and selected using ISO Interface and the applet is set into ACTIVATED state.

6.2.2.1.2.4 Test procedure

ID	HCI commands	API Description	API Expectation	HCI Response	CRR
		Test case			
1		**Send HCI Transaction Event (card emulation)**			
	Send EVT_SEND_DATA with INS = '01'	`extends CardEmulationListener process(): prepareAndSendTransactionEvent() aid = 'A00000000901…01' aidOffset = 0 aidLen = 16 parameters = '01…01' parametersOffset = 0 parametersLen = 10`	No exception shall be thrown	EVT_SEND_DATA (SW - '90 00') HCI event EVT_TRANSACTION sent. This may be sent before or after the EVT_SEND_DATA indicated above.	N1, N2, N3
2		**HCI interface disabled**			
	1 Send on ISO interface: - Send APDU to select the applet - Send APDU (INS = '02')	`extends CardEmulationListener process(): prepareAndSendTransactionEvent()`	throw HCIException with error code reason HCI_CURRENTLY_DISABLED	SW – '90 00' No EVT_TRANSACTION shall be sent	C1
	2 The contactless interface is disabled in the UICC as defined in Global Platform Amendment C Send on ISO interface send the following commands: - Send APDU to select the applet. - Send APDU (INS = '02') - Postcondition: The contactless interface is enabled again in the UICC as defined in Global Platform Amendment C	`setCommunicationInterface() API method of Global Platform Amendment C [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

		Test case			
ID	**HCI commands**	**API Description**	**API Expectation**	**HCI Response**	**CRR**
3		Wrong conditions			

	The initial conditions in clause 6.2.2.1.2.3 related to card emulation are not applicable here. Send on ISO:
	 - Select applet
	 - Send APDU with (INS = '03')
		`Service = ConnectivityService`
		 `Listener = ConnectivityListener`
		 `process():`
		 `prepareAndSendTransactionEvent()`
			throw HCIException with error code reason HCI_CONDITIONS_NOT _SATISFIED	SW - '90 00' No EVT_ TRANSACTION shall be sent	C3

4		Wrong parameters			

	1 - Send EVT_SEND_DAT Awith INS = '01'	`extends CardEmulationListener`
		 `process():`
		
		 `prepareAndSendTransactionEvent()`
		 `aid = 'A00000000901…01'`
		 `aidOffset = 20`
		 `aidLen = 16`
		 `parameters = '01…01'`
		 `parametersOffset = 0`
		 `parametersLen = 10`
			throw java.lang.ArrayIndexOut OfBoundsException	EVT_SEND_DA TA (SW - '90 00') No EVT_ TRANSACTION shall be sent	P2, P3

	2 - Send EVT_SEND_DAT Awith INS = '02'	`extends CardEmulationListener`
		 `process():`
		
		 `prepareAndSendTransactionEvent()`
		 `aid = 'A00000000901…01'`
		 `aidOffset = 0`
		 `aidLen = 20`
		 `parameters = '01…01'`
		 `parametersOffset = 0`
		 `parametersLen = 10`
			throw java.lang.ArrayIndexOut OfBoundsException	EVT_SEND_DA TA (SW - '90 00') No EVT_ TRANSACTION shall be sent	P2, P3

	3 - Send EVT_SEND_DAT 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_DA TA (SW - '90 00') No EVT_ TRANSACTION shall be sent	P2, P3

	4 - Send EVT_SEND_DAT 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_DA TA (SW - '90 00') No EVT_ TRANSACTION shall be sent	P2, P3

	5 - Send EVT_SEND_DAT 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_DA TA (SW - '90 00') No EVT_ TRANSACTION shall be sent	P2, P3

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

	6 - Send EVT_SEND_DAT AINS = '06'	`extends CardEmulationListener`
		`process():`
		
		`prepareAndSendTransactionEvent()`
		`aid = 'A00000000901…01'`
		`aidOffset = 0`
		`aidLen = 16`
		`parameters = null`
		`parametersOffset = 0`
		`parametersLen = 10`
			throw java.lang.NullPointerException	EVT_SEND_DATA (SW - '90 00') No EVT_ TRANSACTION shall be sent	P2, P3

5	Void				
6	Send HCI Transaction Event (reader mode)				

	- Send APDU(INS = '01') on ISO interface	`process():`
		`prepareAndSendTransactionEvent()`
		`aid = 'A00000000901…01'`
		`aidOffset = 0`
		`aidLen = 16`
		`parameters = '01…01'`
		`parametersOffset = 0`
		`parametersLen = 10`
			No exception shall be thrown	HCI event EVT_TRANSAC TION sent.
				
				- SW - '90 00'
					N1, N2, N3

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

Test Area Reference: Api_2_CNs_Stt.

6.2.2.1.3.1 Conformance requirements

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

```
void prepareAndSendTransactionEvent(byte[] parameters,
                                    short parametersOffset,
                                    short parametersLen)
                                    throws HCIException,
                                           java.lang.ArrayIndexOutOfBoundsException,
                                           java.lang.NullPointerException
```

6.2.2.1.3.1.1 Normal execution

- CRRN1: this non-blocking method builds the HCI event EVT_TRANSACTION which notifies the terminal host that it shall launch an application on the terminal which is associated to an Applet in the UICC host identified.

- CRRN2: the contactless runtime environment shall bind the services defined in uicc.hci.services.connectivity to the corresponding resources (e.g. gates and pipes) specified by the HCI protocol [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 ETSI 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
Api_2_CNs_Stt_5.java	5

6.2.2.1.3.3 Initial condition

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

- Prior to this test the Terminal shall have been powered on and performed the PROFILE DOWNLOAD procedure.

- In terminal profile set the 25th byte, 'b6' to indicate that class m is supported.

- In terminal profile set the 5th byte, 'b1' to indicate proactive UICC: SET UP EVENT LIST.

- UICC has created and opened a pipe to connectivity gate in the terminal host

- EVT_FIELD_ON has been sent on HCI interface.

- EVT_CARD_ACTIVATED has been sent on HCI interface.

- According applet has been successfully installed and selected using HCI interface.

Initial conditions for ID 2-1:

- Prior to this test the Terminal shall have been powered on and performed the PROFILE DOWNLOAD procedure.

- In terminal profile set the 5th byte, 'b1' to indicate proactive UICC: SET UP EVENT LIST.

- In terminal profile the following CAT facilities shall be supported :

  - Set the 25th byte, 'b6' to indicate that class m is supported

  - Set the 26th byte, 'b2' and the 31st byte,'b1' to indicate that class r is supported.

- Contactless functionality state is disabled in the UICC as defined in ETSI TS 102 223 [7].

Initial conditions for ID2-2:

- Prior to this test the Terminal shall have been powered on and performed the PROFILE DOWNLOAD procedure.

- In terminal profile set the 25th byte, 'b6' to indicate that class m is supported.

- In terminal profile set the 5th byte, 'b1' to indicate proactive UICC: SET UP EVENT LIST.

Initial conditions for ID 5:

- Prior to this test the Terminal shall have been powered on and performed the PROFILE DOWNLOAD procedure.

- In terminal profile set the 25th byte, 'b6' to indicate that class m is supported.

- In terminal profile set 5th byte, 'b1' to indicate proactive UICC: SET UP EVENT LIST.

- UICC has created and opened a pipe to connectivity gate in the terminal host

- According applet has been successfully installed and selected using ISO Interface and the applet is set into ACTIVATED state.

6.2.2.1.3.4 Test procedure

			Test case		
ID	**HCI commands**	**API Description**	**API Expectation**	**HCI Response**	**CRR**
1		Send HCI Transaction Event (card emulation)			
	Send EVT_SEND_DATA with INS = '01'	```extends CardEmulationListener process(): prepareAndSendTransactionEvent() parameters = '01…01' parametersOffset = 0 parametersLen = 10```	No exception shall be thrown	EVT_SEND_DATA (SW - '90 00') HCI event EVT_TRANSACTION sent. This may be sent before or after the EVT_SEND_DATA indicated above.	N1, N2
2		HCI interface disabled			
	1 Send on ISO interface: - Send APDU to select applet - Send APDU (INS = '02')	```extends CardEmulationListener process(): prepareAndSendTransactionEvent()```	throw HCIException with error code reason HCI_CURRENTLY_DISABLED	SW – '90 00' No EVT_TRANSACTION shall be sent	C1
	2 The contactless interface is disabled in the UICC as defined in Global Platform Amendment C Send on ISO interface send the following commands: - Send APDU to select the applet. - Send APDU (INS = '02') - Postcondition: The contactless interface is enabled again in the UICC as defined in Global Platform Amendment C	```setCommunicationInterface() API method of Global Platform Amendment C [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		Wrong conditions			
	The initial conditions in clause 6.2.2.1.3.3 related to card emulation are not applicable here. Send on ISO interface: - Select applet - APDU data with INS = '03'	```Service = ConnectivityService Listener = ConnectivityListener process(): prepareAndSendTransactionEvent()```	throw HCIException with error code reason HCI_CONDITIONS_NOT _SATISFIED	SW - '90 00' No EVT_ TRANSACTION shall be sent	C3

Test case					
ID	HCI commands	API Description	API Expectation	HCI Response	CRR
4	Wrong parameters				
	1 - Send EVT_SEND_DATA with INS = '01'	`extends CardEmulationListener process(): prepareAndSendTransactionEvent() parameters = '01…01' parametersOffset = 20 parametersLen = 10`	throw java.lang.ArrayIndexOutOfBoundsException	SW - '90 00' No EVT_ TRANSACTION shall be sent	P2, P3
	2 - Send EVT_SEND_DATA with INS = '02'	`extends CardEmulationListener process(): prepareAndSendTransactionEvent() parameters = '01…01' parametersOffset = 0 parametersLen = 20`	throw java.lang.ArrayIndexOutOfBoundsException	SW - '90 00' No EVT_ TRANSACTION shall be sent	P2, P3
	3 - Send EVT_SEND_DATA with INS = '03'	`extends CardEmulationListener process(): prepareAndSendTransactionEvent() parameters = null parametersOffset = 0 parametersLen = 10`	throw java.lang.NullPointerException	SW - '90 00' No EVT_ TRANSACTION shall be sent	P2, P3
5	Send HCI Transaction Event (reader mode)				
	- Send APDU(INS = '01') on ISO interface	`process(): prepareAndSendTransactionEvent() parameters = '01…01' parametersOffset = 0 parametersLen = 10`	No exception shall be thrown	HCI event EVT_TRANSACTION sent. - SW - '90 00'	N1, N2, N3

6.2.3 Package Reader Service

6.2.3.1 Interface ReaderMessage

6.2.3.1.1 Method restartReaderModeProcedure

Test Area Reference: Api_2_RMm_Rrp.

6.2.3.1.1.1 Conformance requirements

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

```
void restartReaderModeProcedure()
                    throws HCIException
```

6.2.3.1.1.1.1 Normal execution

- CRRN1: this method sends first the HCI event EVT_END_OPERATION and then the HCI event EVT_READER_REQUESTED.

- CRRN2: this method is intended to recover from the following error states:

    - reception of EVT_TARGET_DISCOVERED with a MULTIPLE_TARGET_STATUS.

    - reception of RESP_WR_RF_ERROR.

    - reception of RESP_ANY_E_TIMEOUT.

- CRRN3: an Applet shall only be able to activate ReaderListener.EVENT_TARGET_DISCOVERED or to use the restartReadermodeProcedure method if it is in lifecycle state ACTIVATED.

6.2.3.1.1.1.2 Parameter errors

- None.

6.2.3.1.1.1.3 Context errors

- CRRC1: throw HCIException with error code reason HCI_CURRENTLY_DISABLED if the HCI interface is disabled.

- CRRC2: throw HCIException with error code reason HCI_CONDITIONS_NOT_SATISFIED if the Applet is not ACTIVATED.

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	MULTIPLE_TARGET_STATUS is received - Type A				

	Send EVT_TARGET_DISCOVERED(Status = '03')	`restartReaderModeProcedure()`
		
		`event = EVT_END_OPERATION`
		`event = EVT_READER_REQUESTED`
			No exception shall be thrown	-
				EVT_END_OPERATION
				-
				EVT_READER_REQUESTED
					N1, N2, N3

2	MULTIPLE_TARGET_STATUS is received - Type B				

	Send EVT_TARGET_DISCOVERED(Status = '03')	`restartReaderModeProcedure()`
		
		`event = EVT_END_OPERATION`
		`event = EVT_READER_REQUESTED`
			No exception shall be thrown	-
				EVT_END_OPERATION
				-
				EVT_READER_REQUESTED
					N1, N2, N3

3	RESP_WR_RF_ERROR is received				

	- Send EVT_TARGET_DISCOVERED (status = '00') -WR_RF_ERROR	`- prepareAndSendWriteXchgDataCommand()`
		`timeout = -1`
		`data = '01…01'`
		`offset = 0`
		`len = 10`
		
		`- restartReaderModeProcedure()`
		`event = EVT_END_OPERATION`
		`event = EVT_READER_REQUESTED`
			No exception shall be thrown	-
				WR_XCHG_DATA
				-
				EVT_END_OPERATION
				-
				EVT_READER_REQUESTED
					N1, N2, N3

4	RESP_ANY_E_TIMEOUT is received				

	- Send EVT_TARGET_DISCOVERED (status = '00') -
	ANY_E_TIMEOUT
		`- prepareAndSendWriteXchgDataCommand()`
		`timeout = -1`
		`data = '01…01'`
		`offset = 0`
		`len = 10`
		
		`- restartReaderModeProcedure()`
		`event = EVT_END_OPERATION`
		`event = EVT_READER_REQUESTED`
			No exception shall be thrown	-
				WR_XCHG_DATA
				
				-
				EVT_END_OPERATION
				-
				EVT_READER_REQUESTED
					N1, N2, N3

Test case					
ID	HCI Command	API Description	API Expectation	HCI Response	CRR
5		**HCI interface is disabled**			

	1 - Precondition:
	
	The terminal shall indicate the support of class r by setting the 26th byte, 'b2' and the 31st byte, 'b1' in the terminal profile and disable the contactless functionality in the UICC as defined in ETSI TS 102 223 [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_R EQUESTED shall be sent
					C1

	2 - Precondition: The contactless interface is disabled in the UICC as defined in Global Platform Amendment C Send on ISO interface send the following commands:
	- Send EVT_TARGET_DISCOVERED (status = '03')
	- Send APDU on ISO interface ('01')
	- Postcondition: The contactless interface is enabled again in the UICC as defined in Global Platform Amendment C
		`setCommunicationInterface() API method of Global Platform Amendment C [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_R EQUESTED 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.

6.2.3.1.2.2 Test suite files

Applet Name	Test case ID
Api_2_RMm_Srx_1.java	1
Api_2_RMm_Srx_1.java	2
Api_2_RMm_Srx_2.java	3-1
Api_2_RMm_Srx_3.java	3-2
Api_2_RMm_Srx_2.java	4
Api_2_RMm_Srx_2.java	5
Api_2_RMm_Srx_2.java	6
Api_2_RMm_Srx_2.java	7

6.2.3.1.2.3   Initial condition

- According applet has been successfully installed and selected using ISO Interface and the applet is set into ACTIVATED state.

- The UICC has sent EVT_READER_REQUESTED on HCI interface.

6.2.3.1.2.4   Test procedure

		Test case			
ID	**HCI Command**	**API Description**	**API Expectation**	**HCI Response**	**CRR**
1		**Send Data - Type A**			
	1 - Send APDU(INS = '01') on ISO interface - Send EVT_TARGET_DISCOVERED (status = '00') - Send ANY_OK('90 00')	`prepareAndSendWriteXchgDataCommand()` `timeout = -1` `data = '00 01 00 00'` `offset = 0` `len = 4`	No exception shall be thrown	WR_XCHG_DATA, with CTR = '0X' (where X is any value) and data sent = '00 01 00 00'	N1, N2, N3, N4
	2 - Send APDU(INS = '02') on ISO interface - Send EVT_TARGET_DISCOVERED (status = '00') - Send ANY_OK('01 90 00 00')	`prepareAndSendWriteXchgDataCommand()` `timeout = -1` `data = '80 02 01 02 00'` `offset = 0` `len = 5`	No exception shall be thrown	- WR_XCHG_DATA, with CTR = '0X' (where X is any value) and data sent = '80 02 01 02 00'	N1, N2, N3, N4
	3 - Send APDU(INS = '03') on ISO interface - Send EVT_TARGET_DISCOVERED (status = '00') - Send ANY_OK('90 00 00')	`prepareAndSendWriteXchgDataCommand()` `timeout = -1` `data = 'A0 03 FE FF 02 3F 00'` `offset = 0` `len = 7`	No exception shall be thrown	- WR_XCHG_DATA, with CTR = '0X' (where X is any value) and data sent = 'A0 03 FE FF 02 3F 00'	N1, N2, N3, N4
	4 - Send APDU(INS = '04') on ISO interface - Send EVT_TARGET_DISCOVERED (status = '00') - Send ANY_OK('01 90 00 00')	`prepareAndSendWriteXchgDataCommand()` `timeout = -1` `data = '00 04 00 00 F8 01 … F8 00'` `offset = 0` `len = 254`	No exception shall be thrown	- WR_XCHG_DATA, with CTR = '0X' (where X is any value) and data sent = '00 04 00 00 F8 01 … F8 00'	N1, N2, N3, N4
	5 - Send APDU(INS = '05') on ISO interface - Send EVT_TARGET_DISCOVERED (status = '00') - Send ANY_OK('01 90 00 00')	`prepareAndSendWriteXchgDataCommand()` `timeout = 0` `data = '00 05 00 00 02 3F 00 00'` `offset = 0` `len = 8`	No exception shall be thrown	- WR_XCHG_DATA, with CTR = '10' and data sent = '00 05 00 00 02 3F 00 00'	N1, N2, N3, N4

		Test case			
ID	**HCI Command**	**API Description**	**API Expectation**	**HCI Response**	**CRR**

	6 - Send APDU(INS = '06') on ISO interface
	- Send EVT_TARGET_DISCOVERED (status = '00')
	- Send ANY_OK('01 90 00 00')
		`prepareAndSendWriteXchgDataCommand()`
		`timeout = 5`
		`data = '00 06 00 00 02 3F 00 00'`
		`offset = 0`
		`len = 8`
			No exception shall be thrown	-
				WR_XCHG_DATA, with CTR = '15' and data sent = '00 06 00 00 02 3F 00 00'
					N1, N2, N3, N4

	7 - Send APDU(INS = '07') on ISO interface
	- Send EVT_TARGET_DISCOVERED (status = '00')
	- Send ANY_OK('01 90 00 00')
		`prepareAndSendWriteXchgDataCommand()`
		`timeout = 14`
		`data = '00 07 00 00 02 3F 00 00'`
		`offset = 0`
		`len = 8`
			No exception shall be thrown	-
				WR_XCHG_DATA, with CTR = '1E' and data sent = '00 07 00 00 02 3F 00 00'
					N1, N2, N3, N4

2		Send Data - Type B			

	1 - Send APDU(INS = '01') on ISO interface
	- Send EVT_TARGET_DISCOVERED (status = '00')
	- Send ANY_OK('90 00')
		`prepareAndSendWriteXchgDataCommand()`
		`timeout = -1`
		`data = '00 01 00 00'`
		`offset = 0`
		`len = 4`
			No exception shall be thrown	WR_XCHG_DATA, with CTR = '0X' (where X is any value) and data sent = '00 01 00 00'	N1, N2, N3, N4

	2 - Send APDU(INS = '02') on ISO interface
	- Send EVT_TARGET_DISCOVERED (status = '00')
	- Send ANY_OK('01 90 00 00')
		`prepareAndSendWriteXchgDataCommand()`
		`timeout = -1`
		`data = '80 02 01 02 00'`
		`offset = 0`
		`len = 5`
			No exception shall be thrown	-
				WR_XCHG_DATA, with CTR = '0X' (where X is any value) and data sent = '80 02 01 02 00'
					N1, N2, N3, N4

	3 - Send APDU(INS = '03') on ISO interface
	- Send EVT_TARGET_DISCOVERED (status = '00')
	- Send ANY_OK('90 00 00')
		`prepareAndSendWriteXchgDataCommand()`
		`timeout = -1`
		`data = 'A0 03 FE FF 02 3F 00'`
		`offset = 0`
		`len = 7`
			No exception shall be thrown	-
				WR_XCHG_DATA, with CTR = '0X' (where X is any value) and data sent = 'A0 03 FE FF 02 3F 00'
					N1, N2, N3, N4

	4 - Send APDU(INS = '04') on ISO interface
	- Send EVT_TARGET_DISCOVERED (status = '00')
	- Send ANY_OK('01 90 00 00')
		`prepareAndSendWriteXchgDataCommand()`
		`timeout = -1`
		`data = '00 04 00 00 F8 01 … F8 00'`
		`offset = 0`
		`len = 254`
			No exception shall be thrown	-
				WR_XCHG_DATA, with CTR = '0X' (where X is any value) and data sent = '00 04 00 00 F8 01 … F8 00'
					N1, N2, N3, N4

		Test case			
ID	**HCI Command**	**API Description**	**API Expectation**	**HCI Response**	**CRR**

	5 - Send APDU(INS = '05') on ISO interface
	- Send EVT_TARGET_DISCOVERED (status = '00')
	- Send ANY_OK('01 90 00 00')
		`prepareAndSendWriteXchgDataCommand()`
		`timeout = 0`
		`data = '00 05 00 00 02 3F 00 00'`
		`offset = 0`
		`len = 8`
			No exception shall be thrown	-
				WR_XCHG_DATA, with CTR = '10' and data sent = '00 05 00 00 02 3F 00 00'
					N1, N2, N3, N4

	6 - Send APDU(INS = '06') on ISO interface
	- Send EVT_TARGET_DISCOVERED (status = '00')
	- Send ANY_OK('01 90 00 00')
		`prepareAndSendWriteXchgDataCommand()`
		`timeout = 5`
		`data = '00 06 00 00 02 3F 00 00'`
		`offset = 0`
		`len = 8`
			No exception shall be thrown	-
				WR_XCHG_DATA, with CTR = '15' and data sent = '00 06 00 00 02 3F 00 00'
					N1, N2, N3, N4

	7 - Send APDU(INS = '07') on ISO interface
	- Send EVT_TARGET_DISCOVERED (status = '00')
	- Send ANY_OK('01 90 00 00')
		`prepareAndSendWriteXchgDataCommand()`
		`timeout = 14`
		`data = '00 07 00 00 02 3F 00 00'`
		`offset = 0`
		`len = 8`
			No exception shall be thrown	-
				WR_XCHG_DATA, with CTR = '1E' and data sent = '00 07 00 00 02 3F 00 00'
					N1, N2, N3, N4

3		**HCI interface is disabled**			

	1 - Precondition:
	
	The terminal shall indicate the support of class r by setting the 26th byte, 'b2' and the 31st byte, 'b1' in the terminal profile and disable the contactless functionality in the UICC as defined in ETSI TS 102 223 [7]
	- Send APDU on ISO interface to select the applet
	- Send EVT_TARGET_DISCOVERED (status = '00')
	- Send APDU on ISO interface (INS = '11')
		`prepareAndSendWriteXchgDataCommand()`
		`timeout = 14`
		`data = '01…01'`
		`offset = 0`
		`len = 10`
			Shall throw uicc.hci.framework.HCIException with error code HCI_CURRENTLY_DISABLED	- SW - '90 01'	C1

		Test case			
ID	HCI Command	API Description	API Expectation	HCI Response	CRR
	2 - Precondition: The contactless interface is disabled in the UICC as defined in Global Platform Amendment C Send on ISO interface send the following commands: - Send APDU on ISO interface to select the applet - Send EVT_TARGET_DISCOVERED (status = '00') - Send APDU on ISO interface (INS = '01') - Postcondition: The contactless interface is enabled again in the UICC as defined in Global Platform Amendment C	`setCommunicationInterface() API method of Global Platform Amendment C [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.HCIException with error code HCI_CURRENTLY_DISABLED	- SW - '90 01'	C1
4		Data is null			
	- Send APDU on ISO interface (INS = '02') - Send EVT_TARGET_DISCOVERED (status = '00') - Send APDU on ISO interface ('12')	`prepareAndSendWriteXchgDataCommand() timeout = 14 data = null offset = 0 len = 10`	Shall throw java.lang.NullPointerException	- SW - '90 02'	P1
5		ArrayIndexOutOfBounds Exception			
	- Send APDU on ISO interface (INS = '03') - Send EVT_TARGET_DISCOVERED (status = 00) - Send APDU on ISO interface ('13')	`prepareAndSendWriteXchgDataCommand() timeout = 14 data = '01…01' offset = -1 len = 10`	Shall throw java.lang. ArrayIndexOutOfBoundsException	- SW - '90 03'	P2
6		ArrayIndexOutOfBounds Exception			
	- Send APDU on ISO interface (INS = '04') - Send EVT_TARGET_DISCOVERED (status = '00') - Send APDU on ISO interface (INS = '14')	`prepareAndSendWriteXchgDataCommand() timeout = 14 data = '01…01' offset = 0 len = 11`	Shall throw java.lang. ArrayIndexOutOfBoundsException	- SW - '90 04'	P2

Test case					
ID	HCI Command	API Description	API Expectation	HCI Response	CRR
7	Timeout has illegal value				

	- Send APDU on ISO interface (INS = '05')
	- Send EVT_TARGET_DISCOVERED (status = 00)
	- Send APDU on ISO interface (INS = '15')
		`prepareAndSendWriteXchgDataCommand()`
		`timeout = -2`
		`data = '01…01'`
		`offset = 0`
		`len = 10`
			Shall throw uicc.hci.framework.HCIException with error code ILLEGAL_VALUE	- SW - '90 05'	P3

6.2.3.1.3 Method prepareAndSendGetParameterCommand

Test Area Reference: Api_2_RMm_Sgp.

6.2.3.1.3.1 Conformance requirements

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

```
void prepareAndSendGetParameterCommand(byte paramid)
                             throws HCIException,
                                    javacardframework.SystemException
```

6.2.3.1.3.1.1 Normal execution

- CRRN1: This non-blocking method builds the HCI command ANY_GET_PARAMETER for the indicated information and sends it to the Contactless Reader Gate of the CLF.

- CRRN2: this method should only be called when the CLF has successfully discovered a card in the field, i.e. after reception of the HCI event EVT_TARGET_DISCOVERED.

- CRRN3: paramid shall be one of the PARAM_* values defined in this interface.

6.2.3.1.3.1.2 Parameter errors

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

6.2.3.1.3.1.3 Context errors

- CRRC1: throw HCIException with error code reason HCI_CURRENTLY_DISABLED if the HCI interface was disabled.

- CRRC2: throw HCIException with error code reason HCI_RESOURCES_NOT_AVAILABLE if the contactless framework does not have enough resources to process the command.

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	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

<table>
<tr><th colspan="6">Test case</th></tr>
<tr><th>ID</th><th>HCI Command</th><th>API Description</th><th>API Expectation</th><th>HCI Response</th><th>CRR</th></tr>
<tr><td>1</td><td colspan="5">**Get Parameter: Type A**</td></tr>
<tr>
<td></td>
<td>1 - Send APDU on ISO interface (INS = '01')
- Send EVT_TARGET_DISCOVERED (status = '00')
- Send ANY_OK as response to GET_PARAMETER

- Send APDU on ISO interface (INS = '20')</td>
<td>`prepareAndSendGetParameterCommand()`
`paramid =`
`PARAM_ID_TYPE_A_READER_UID`

`Response received:`
`onCallback()`
`event = EVENT_GET_PARAMETER_RESPONSE`</td>
<td>No exception shall be thrown</td>
<td>-
GET_PARAMETER with command parameter as indicated in the API Description column
- APDU-Response '0100'</td>
<td>N1, N2, N3</td>
</tr>
<tr>
<td></td>
<td>2 - Send APDU on ISO interface (INS = '02')
- Send EVT_TARGET_DISCOVERED (status = '00')
- Send ANY_OK as response to GET_PARAMETER

- Send APDU on ISO interface (INS = '20')</td>
<td>`prepareAndSendGetParameterCommand()`
`paramid =`
`PARAM_ID_TYPE_A_READER_ATQA`

`Response received:`
`onCallback()`
`event = EVENT_GET_PARAMETER_RESPONSE`</td>
<td>No exception shall be thrown</td>
<td>-
GET_PARAMETER with command parameter as indicated in the API Description column
- APDU-Response '0200'</td>
<td>N1, N2, N3</td>
</tr>
<tr>
<td></td>
<td>3 - Send APDU on ISO interface (INS = '03')
- Send EVT_TARGET_DISCOVERED (status = '00')
- Send ANY_OK as response to GET_PARAMETER
- Send APDU on ISO interface (INS = '20')</td>
<td>`prepareAndSendGetParameterCommand()`
`paramid =`
`PARAM_ID_TYPE_A_READER_APPLICATION_D ATA`

`Response received:`
`onCallback()`
`event = EVENT_GET_PARAMETER_RESPONSE`</td>
<td>No exception shall be thrown</td>
<td>-
GET_PARAMETER with command parameter as indicated in the API Description column
- APDU-Response '0300'</td>
<td>N1, N2, N3</td>
</tr>
<tr>
<td></td>
<td>4 - Send APDU on ISO interface (INS = '04')
- Send EVT_TARGET_DISCOVERED (status = '00')
- Send ANY_OK as response to GET_PARAMETER

- Send APDU on ISO interface (INS = '20')</td>
<td>`prepareAndSendGetParameterCommand()`
`paramid =`
`PARAM_ID_TYPE_A_READER_SAK`

`Response received:`
`onCallback()`
`event = EVENT_GET_PARAMETER_RESPONSE`</td>
<td>No exception shall be thrown.</td>
<td>-
GET_PARAMETER with command parameter as indicated in the API Description column
- APDU-Response '0400'</td>
<td>N1, N2, N3</td>
</tr>
<tr>
<td></td>
<td>5 - Send APDU on ISO interface (INS = '05')
- Send EVT_TARGET_DISCOVERED (status = '00')
- Send ANY_OK as response to GET_PARAMETER

- Send APDU on ISO interface (INS = '20')</td>
<td>-
`prepareAndSendGetParameterCommand()`
`paramid =`
`PARAM_ID_TYPE_A_READER_FWI`

`Response received:`
`onCallback()`
`event = EVENT_GET_PARAMETER_RESPONSE`</td>
<td>No exception shall be thrown</td>
<td>-
GET_PARAMETER with command parameter as indicated in the API Description column
- APDU-Response '0500'</td>
<td>N1, N2, N3</td>
</tr>
</table>

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

	6 - Send APDU on ISO interface (INS = '06') - Send EVT_TARGET_DISCOVERED (status = '00') - Send ANY_OK as response to GET_PARAMETER
	
	- Send APDU on ISO interface (INS = '20')
		`prepareAndSendGetParameterCommand()`
		`paramid =`
		`PARAM_ID_TYPE_A_READER_DATARATE_MAX`
		
		`Response received:`
		`onCallback()`
		`event = EVENT_GET_PARAMETER_RESPONSE`
			No exception shall be thrown	-
				GET_PARAMETER with command parameter as indicated in the API Description column
				- APDU-Response '0600'
					N1, N2, N3

2		**Get Parameter: Type B**			

	1 - Send APDU on ISO interface (INS = '07') - Send EVT_TARGET_DISCOVERED (status = '00') - Send ANY_OK as response to GET_PARAMETER
	
	- Send APDU on ISO interface (INS = '20')
		`prepareAndSendGetParameterCommand()`
		`paramid =`
		`PARAM_ID_TYPE_B_READER_PUPI`
		
		`Response received:`
		`onCallback()`
		`event = EVENT_GET_PARAMETER_RESPONSE`
			No exception shall be thrown	-
				GET_PARAMETER with command parameter as indicated in the API Description column
				- APDU-Response '0700'
					N1, N2, N3

	2 - Send APDU on ISO interface (INS = '08') - Send EVT_TARGET_DISCOVERED (status = '00') - Send ANY_OK as response to GET_PARAMETER
	
	- Send APDU on ISO interface (INS = '20')
		`prepareAndSendGetParameterCommand()`
		`paramid =`
		`PARAM_ID_TYPE_B_READER_APPLICATION_D`
		`ATA`
		
		`Response received:`
		`onCallback()`
		`event = EVENT_GET_PARAMETER_RESPONSE`
			No exception shall be thrown	-
				GET_PARAMETER with command parameter as indicated in the API Description column
				- APDU-Response '0800'
					N1, N2, N3

	3 - Send APDU on ISO interface (INS = '09') - Send EVT_TARGET_DISCOVERED (status = '00') - Send ANY_OK as response to GET_PARAMETER
	
	- Send APDU on ISO interface (INS = '20')
		`prepareAndSendGetParameterCommand()`
		`paramid =`
		`PARAM_ID_TYPE_B_READER_AFI`
		
		`Response received:`
		`onCallback()`
		`event = EVENT_GET_PARAMETER_RESPONSE`
			No exception shall be thrown	-
				GET_PARAMETER with command parameter as indicated in the API Description column
				- APDU-Response '0900'
					N1, N2, N3

	4 - Send APDU on ISO interface (INS = '0A') - Send EVT_TARGET_DISCOVERED (status = '00') - Send ANY_OK as response to GET_PARAMETER
	
	- Send APDU on ISO interface (INS = '20')
		`prepareAndSendGetParameterCommand()`
		`paramid =`
		`PARAM_ID_TYPE_B_READER_HIGHER_LAYER_`
		`RESPONSE`
		
		`Response received:`
		`onCallback()`
		`event = EVENT_GET_PARAMETER_RESPONSE`
			No exception shall be thrown	-
				GET_PARAMETER with command parameter as indicated in the API Description column
				- APDU-Response '0A00'
					N1, N2, N3

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

	5 - Send APDU on ISO interface (INS = '0B')
	 - Send EVT_TARGET_DISCOVERED (status = '00')
	 - Send ANY_OK as response to GET_PARAMETER
	
	 - Send APDU on ISO interface (INS = '20')
		`prepareAndSendGetParameterCommand()` `paramid =` `PARAM_ID_TYPE_B_READER_HIGHER_LAYER_` `DATA`
		
		 `Response received:` `onCallback()` `event = EVENT_GET_PARAMETER_RESPONSE`
			No exception shall be thrown	- GET_PARAMETER with command parameter as indicated in the API Description column
				 - APDU-Response '0B00'
					N1, N2, N3

3	HCI interface is disabled				
	1 - Precondition: The terminal shall indicate the support of class r by setting 26th byte, 'b2' and the 31st byte, 'b1' in the terminal profile and disable the contactless functionality in the UICC as defined in ETSI TS 102 223 [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: The contactless interface is disabled in the UICC as defined in Global Platform Amendment C Send on ISO interface send the following commands:
	 - Send APDU on ISO interface to select the applet
	 - Send EVT_TARGET_DISCOVERED (status = '00')
	 - Send APDU on ISO interface (INS = '11')
	 - Postcondition: The contactless interface is enabled again in the UICC as defined in Global Platform Amendment C
		`setCommunicationInterface() API` `method of Global Platform Amendment` `C [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

4	Void				
5	The parameter has an illegal value				
	- Send APDU on ISO interface (INS = '05') - Send EVT_TARGET_DISCOVERED (status = '00') - Send APDU on ISO interface (INS = '15')	`-` `prepareAndSendGetParameterCommand()` `paramid = -1`	Shall throw uicc.hci.framework. HCIException with error code ILLEGAL_VALUE	- SW - '90 05'	P1

6.2.3.2 Interface ReaderListener

6.2.3.2.1 Method onCallback

Test Area Reference: Api_2_Rml_Ocb.

6.2.3.2.1.1 Conformance requirements

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

```
void onCallback(byte event, HCIMessage message)
```

6.2.3.2.1.1.1 Normal execution

- CRRN1: The Contactless Framework shall invoke the method ReaderListener.onCallback only with events which are defined in ReaderListener or HCIListener and only with ReaderMessage objects.

- CRRN2: The Applet instance shall activate the events with ReaderService.activateEvent before it will receive any event notification.

6.2.3.2.1.1.2 Parameter errors

- None.

6.2.3.2.1.1.3 Context errors

- None.

6.2.3.2.1.2 Test Suite Files

Applet Name	Test case ID
Api_2_Rml_Ocb_1.java	1
Api_2_Rml_Ocb_1.java	2
Api_2_Rml_Ocb_1.java	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**
1			EVENT_TARGET_DISCOVERED - Type A		

	- send APDU (INS = '10') on ISO interface
	
	- Send EVENT_TARGET_DISCOVERED (status = '00')
		`HCIService = ReaderService`
		
		`activateEvent()`
		`event = `
		`EVENT_TARGET_DISCOVERED`
			No exception shall be thrown.
			
			onCallback() method shall notice the event
				WR_XCHG_DATA ('00 01 00 00')	N1, N2

2			EVENT_WRITE_EXCHANGE_DATA_RESPONSE - Type A		

	- send APDU(INS = '11') on ISO interface
	
	- Send EVT_TARGET_DISCOVERED (status = '00')
	
	- Send ANY_OK ()
		`HCIService = ReaderService`
		
		`activateEvent()`
		`event = `
		`EVENT_WRITE_EXCHANGE_DATA_RES`
		`PONSE`
			No exception shall be thrown.
			
			onCallback() method shall notice the event
				Start bulk data transfer on HCI interface.	N1, N2

3			HCIListener.EVENT_HCI_TRANSMISSION_FAILED - Type A		

	- send APDU (INS = '12') on ISO interface
	- 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_TRANSMI`
		`SSION_FAILED`
			No exception shall be thrown.
			
			onCallback() method shall notice the event
				- No WR_XCHG_DATA is expected.
				
				- SW - '90 00'
					N1, N2.

4			EVENT_GET_PARAMETER_RESPONSE - Type A		

	- send APDU(INS = '13') on ISO interface
	- Send EVT_TARGET_DISCOVERED (status = '00')
	- Send ANY_OK ()
		`HCIService = ReaderService`
		
		`activateEvent()`
		`event = `
		`EVENT_GET_PARAMETER_RESPONSE`
			No exception shall be thrown.
			
			onCallback() method shall notice the event
				ANY_GET_PARAMETER	N1, N2

5			EVENT_TARGET_DISCOVERED - Type B		

	- send APDU (INS = '10') on ISO interface
	
	- Send EVENT_TARGET_DISCOVERED (status = '00')
		`HCIService = ReaderService`
		
		`activateEvent()`
		`event = `
		`EVENT_TARGET_DISCOVERED`
			No exception shall be thrown.
			
			onCallback() method shall notice the event
				WR_XCHG_DATA ('00 01 00 00')	N1, N2

6			EVENT_WRITE_EXCHANGE_DATA_RESPONSE - Type B		

	- send APDU(INS = '11') on ISO interface
	
	- Send EVT_TARGET_DISCOVERED (status = '00')
	
	- Send ANY_OK ()
		`HCIService = ReaderService`
		
		`activateEvent()`
		`event = `
		`EVENT_WRITE_EXCHANGE_DATA_RES`
		`PONSE`
			No exception shall be thrown.
			
			onCallback() method shall notice the event
				Start bulk data transfer on HCI interface.	N1, N2

Test Case					
ID	HCI Command	API Description	API Expectation	HCI Response	CRR
HCIListener.EVENT_HCI_TRANSMISSION_FAILED - Type B					

7	- send APDU (INS = '12') on ISO interface
	- 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_TRANSMI`
		`SSION_FAILED`
			No exception shall be thrown.
			
			onCallback() method shall notice the event
				- No WR_XCHG_DATA is expected.
				
				- SW - '90 00'
					N1, N2.

EVENT_GET_PARAMETER_RESPONSE - Type B					

8	- send APDU(INS = '13') on ISO interface
	- Send EVT_TARGET_DISCOVERED (status = '00')
	- Send ANY_OK ()
		`HCIService = ReaderService`
		
		`activateEvent()`
		`event =`
		`EVENT_GET_PARAMETER_RESPONSE`
			No exception shall be thrown.
			
			onCallback() method shall notice the event
				ANY_GET_PARAMETER	N1, N2

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

A.1 HCI framework

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

A.1.1 Class HCIDevice

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

A.1.2 Interface HCIService

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

A.1.3 Interface HCIMessage

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

A.1.4 Interface HCIListener

Method Name	Acronyms
`void onCallback()`	Ocb

A.1.5 Class HCIException

Method Name	Acronyms
`public static void throwIt()`	Trw

A.2 HCI Services

A.2.1 Package cardemulation

Interface Name	Acronyms
CardEmulationMessage	CEm
CardEmulationService	CEs

A.2.1.1 Interface CardEmulationListener

Method Name	Acronyms
void onCallback()	Ocb

A.2.1.2 Interface CardEmulationMessage

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

A.2.1.3 Interface CardEmulationService

Method Name	Acronyms
byte getCardRFType()	Rft

A.2.2 Package connectivity

Interface Name	Acronyms
ConnectivityMessage	CNm
ConnectivityService	CNs

A.2.2.1 Interface ConnectivityListener

FFS

A.2.2.2 Interface ConnectivityMessage

FFS

A.2.2.3 Interface ConnectivityService

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

A.2.3 Readermode

A.2.3.1 Interface RaederListener

Method Name	Acronyms
void onCallback()	Ocb

A.2.3.2 Interface ReaderMessage

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

A.2.3.3 Interface ReaderService

FFS

Annex B (normative):
AIDs - to be reserved

B.1   Package HCI framework

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

B.1.1   Class HCIDevice

Package name	AID
uicc.test.hci.framework. Api_1_Hdv_Gsr	A0 00 00 00 09 00 05 FF FF FF FF 89 21 01 00 00
uicc.test.hci.framework. Api_1_Hdv_Gpm	A0 00 00 00 09 00 05 FF FF FF FF 89 21 02 00 00
uicc.test.hci.framework. Api_1_Hdv_Isa	A0 00 00 00 09 00 05 FF FF FF FF 89 21 03 00 00

Applet name	AID
Api_1_Hdv_Gsr_1.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 01 01 02
Api_1_Hdv_Gsr_2.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 01 02 02
Api_1_Hdv_Gpm_1.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 02 01 02
Api_1_Hdv_Isa_1.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 03 01 02
Api_1_Hdv_Isa_2.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 03 02 02
Api_1_Hdv_Isa_3.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 03 03 02

B.1.2   Interface HCIService

Package name	AID
uicc.test.hci.framework. Api_1_Hsr_Reg	A0 00 00 00 09 00 05 FF FF FF FF 89 21 04 00 00
uicc.test.hci.framework. Api_1_Hsr_Drg	A0 00 00 00 09 00 05 FF FF FF FF 89 21 05 00 00
uicc.test.hci.framework. Api_1_Hsr_Ace	A0 00 00 00 09 00 05 FF FF FF FF 89 21 06 00 00
uicc.test.hci.framework. Api_1_Hsr_Dae	A0 00 00 00 09 00 05 FF FF FF FF 89 21 07 00 00
uicc.test.hci.framework. Api_1_Hsr_Rcn	A0 00 00 00 09 00 05 FF FF FF FF 89 21 08 00 00
uicc.test.hci.framework. Api_1_Hsr_Gen	A0 00 00 00 09 00 05 FF FF FF FF 89 21 09 00 00

Applet name	AID
Api_1_Hsr_Reg_1.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 04 01 02
Api_1_Hsr_Reg_2.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 04 02 02
Api_1_Hsr_Reg_3.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 04 03 02
Api_1_Hsr_Reg_4.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 04 04 02
Api_1_Hsr_Drg_1.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 05 01 02
Api_1_Hsr_Drg_2.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 05 02 02
Api_1_Hsr_Ace_1.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 06 01 02
Api_1_Hsr_Ace_2.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 06 02 02
Api_1_Hsr_Ace_3.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 06 03 02
Api_1_Hsr_Ace_4.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 06 04 02
Api_1_Hsr_Ace_5.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 06 05 02
Api_1_Hsr_Ace_6.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 06 06 02
Api_1_Hsr_Ace_7.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 06 07 02
Api_1_Hsr_Ace_8.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 06 08 02
Api_1_Hsr_Ace_9.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 06 09 02
Api_1_Hsr_Dae_1.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 07 01 02
Api_1_Hsr_Dae_2.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 07 02 02
Api_1_Hsr_Dae_3.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 07 03 02
Api_1_Hsr_Dae_4.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 07 04 02
Api_1_Hsr_Dae_6.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 07 06 02
Api_1_Hsr_Rcn_7.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 08 07 02
Api_1_Hsr_Gen_1.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 09 01 02
Api_1_Hsr_Gen_2.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 09 02 02

B.1.3 Interface HCIMessage

Package name	AID
uicc.test.hci.framework. Api_1_Hme_Mhd	A0 00 00 00 09 00 05 FF FF FF FF 89 21 0A 00 00
uicc.test.hci.framework. Api_1_Hme_Mco	A0 00 00 00 09 00 05 FF FF FF FF 89 21 0B 00 00
uicc.test.hci.framework. Api_1_Hme_Mty	A0 00 00 00 09 00 05 FF FF FF FF 89 21 0C 00 00
uicc.test.hci.framework. Api_1_Hme_Min	A0 00 00 00 09 00 05 FF FF FF FF 89 21 0D 00 00
uicc.test.hci.framework. Api_1_Hme_Mro	A0 00 00 00 09 00 05 FF FF FF FF 89 21 0E 00 00
uicc.test.hci.framework. Api_1_Hme_Mrl	A0 00 00 00 09 00 05 FF FF FF FF 89 21 0F 00 00
uicc.test.hci.framework. Api_1_Hme_Mrb	A0 00 00 00 09 00 05 FF FF FF FF 89 21 10 00 00

Applet name	AID
Api_1_Hme_Mhd_1.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 0A 01 02
Api_1_Hme_Mco_1.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 0B 01 02
Api_1_Hme_Mty_1.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 0C 01 02
Api_1_Hme_Mty_2.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 0C 02 02
Api_1_Hme_Min_1.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 0D 01 02
Api_1_Hme_Min_2.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 0D 02 02
Api_1_Hme_Mrl_1.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 0F 01 02
Api_1_Hme_Mrb_1.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 10 01 02

B.1.4 Interface HCIListener

Package name	AID
uicc.test.hci.framework. Api_1_Hln_Ocb	A0 00 00 00 09 00 05 FF FF FF FF 89 21 11 00 00

Applet name	AID
Api_1_Hln_Ocb_2.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 11 02 02
Api_1_Hln_Ocb_3.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 11 03 02
Api_1_Hln_Ocb_4.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 11 04 02
Api_1_Hln_Ocb_5.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 11 05 02
Api_1_Hln_Ocb_6.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 11 06 02

B.1.5 Class HCIException

Package name	AID
uicc.test.hci.framework. Api_1_Hxp_Trw	A0 00 00 00 09 00 05 FF FF FF FF 89 21 12 00 00

Applet name	AID
Api_1_Hxp_Trw_1.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 12 01 02

B.2 HCI Services

B.2.1 Package cardemulation

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

B.2.1.1 Interface CardEmulationListener

Package name	AID
uicc.test.hci.services.cardemulation.Api_2_CEI_Ocb	A0 00 00 00 09 00 05 FF FF FF FF 89 25 05 00 00

Applet name	AID
Api_2_CEl_Ocb_1.java	A0 00 00 00 09 00 05 FF FF FF FF 89 25 05 01 02
Api_2_CEl_Ocb_3.java	A0 00 00 00 09 00 05 FF FF FF FF 89 25 05 03 02
Api_2_CEl_Ocb_4.java	A0 00 00 00 09 00 05 FF FF FF FF 89 25 05 04 02
Api_2_CEl_Ocb_5.java	A0 00 00 00 09 00 05 FF FF FF FF 89 25 05 05 02
Api_2_CEl_Ocb_6.java	A0 00 00 00 09 00 05 FF FF FF FF 89 25 05 06 02
Api_2_CEl_Ocb_7.java	A0 00 00 00 09 00 05 FF FF FF FF 89 25 05 07 02
Api_2_CEl_Ocb_8.java	A0 00 00 00 09 00 05 FF FF FF FF 89 25 05 08 02

B.2.1.2 Interface CardEmulationMessage

Package name	AID
uicc.test.hci.services.cardemulation.Api_2_CEm_Sgp	A0 00 00 00 09 00 05 FF FF FF FF 89 25 01 00 00
uicc.test.hci.services.cardemulation.Api_2_CEm_Ssd	A0 00 00 00 09 00 05 FF FF FF FF 89 25 02 00 00
uicc.test.hci.services.cardemulation.Api_2_CEm_Scm	A0 00 00 00 09 00 05 FF FF FF FF 89 25 03 00 00

Applet name	AID
Api_2_CEm_Sgp_1.java	A0 00 00 00 09 00 05 FF FF FF FF 89 25 01 01 02
Api_2_CEm_Sgp_2.java	A0 00 00 00 09 00 05 FF FF FF FF 89 25 01 02 02
Api_2_CEm_Sgp_3.java	A0 00 00 00 09 00 05 FF FF FF FF 89 25 01 03 02
Api_2_CEm_Sgp_4.java	A0 00 00 00 09 00 05 FF FF FF FF 89 25 01 04 02
Api_2_CEm_Sgp_5.java	A0 00 00 00 09 00 05 FF FF FF FF 89 25 01 05 02
Api_2_CEm_Sgp_6.java	A0 00 00 00 09 00 05 FF FF FF FF 89 25 01 06 02
Api_2_CEm_Sgp_7.java	A0 00 00 00 09 00 05 FF FF FF FF 89 25 01 07 02
Api_2_CEm_Sgp_8.java	A0 00 00 00 09 00 05 FF FF FF FF 89 25 01 08 02
Api_2_CEm_Sgp_9.java	A0 00 00 00 09 00 05 FF FF FF FF 89 25 01 09 02
Api_2_CEm_Sgp_10.java	A0 00 00 00 09 00 05 FF FF FF FF 89 25 01 0A 02
Api_2_CEm_Sgp_11.java	A0 00 00 00 09 00 05 FF FF FF FF 89 25 01 0B 02
Api_2_CEm_Sgp_12.java	A0 00 00 00 09 00 05 FF FF FF FF 89 25 01 0C 02
Api_2_CEm_Sgp_13.java	A0 00 00 00 09 00 05 FF FF FF FF 89 25 01 0D 02
Api_2_CEm_Sgp_14.java	A0 00 00 00 09 00 05 FF FF FF FF 89 25 01 0E 02
Api_2_CEm_Sgp_15.java	A0 00 00 00 09 00 05 FF FF FF FF 89 25 01 0F 02
Api_2_CEm_Ssd_1.java	A0 00 00 00 09 00 05 FF FF FF FF 89 25 02 01 02
Api_2_CEm_Ssd_2.java	A0 00 00 00 09 00 05 FF FF FF FF 89 25 02 02 02
Api_2_CEm_Ssd_3.java	A0 00 00 00 09 00 05 FF FF FF FF 89 25 02 03 02
Api_2_CEm_Ssd_4.java	A0 00 00 00 09 00 05 FF FF FF FF 89 25 02 04 02
Api_2_CEm_Scm_1.java	A0 00 00 00 09 00 05 FF FF FF FF 89 25 03 01 02

B.2.1.3 Interface CardEmulationService

Package name	AID
uicc.test.hci.services.cardemulation.Api_2_CEs_RFt	A0 00 00 00 09 00 05 FF FF FF FF 89 25 04 00 00

Applet name	AID
Api_2_CEs_RFt_1.java	A0 00 00 00 09 00 05 FF FF FF FF 89 25 04 01 02

B.2.2 Package connectivity

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

B.2.2.1 Interface ConnectivityListener

FFS

B.2.2.2 Interface ConnectivityMessage

FFS

B.2.2.3 Interface ConnectivityService

Package name	AID
uicc.test.hci.services.connectivity.Api_2_CNs_Sce	A0 00 00 00 09 00 05 FF FF FF FF 89 26 01 00 00
uicc.test.hci.services.connectivity.Api_2_CNs_Ste	A0 00 00 00 09 00 05 FF FF FF FF 89 26 02 00 00
uicc.test.hci.services.connectivity.Api_2_CNs_Stt	A0 00 00 00 09 00 05 FF FF FF FF 89 26 03 00 00

Applet name	AID
Api_2_CNs_Sce_1.java	A0 00 00 00 09 00 05 FF FF FF FF 89 26 01 01 02
Api_2_CNs_Sce_2.java	A0 00 00 00 09 00 05 FF FF FF FF 89 26 01 02 02
Api_2_CNs_Sce_3.java	A0 00 00 00 09 00 05 FF FF FF FF 89 26 01 03 02
Api_2_CNs_Sce_4.java	A0 00 00 00 09 00 05 FF FF FF FF 89 26 01 04 02
Api_2_CNs_Sce_5.java	A0 00 00 00 09 00 05 FF FF FF FF 89 26 01 05 02
Api_2_CNs_Sce_6.java	A0 00 00 00 09 00 05 FF FF FF FF 89 26 01 06 02
Api_2_CNs_Sce_7.java	A0 00 00 00 09 00 05 FF FF FF FF 89 26 01 07 02
Api_2_CNs_Ste_1.java	A0 00 00 00 09 00 05 FF FF FF FF 89 26 02 01 02
Api_2_CNs_Ste_2.java	A0 00 00 00 09 00 05 FF FF FF FF 89 26 02 02 02
Api_2_CNs_Ste_4.java	A0 00 00 00 09 00 05 FF FF FF FF 89 26 02 04 02
Api_2_CNs_Ste_5.java	A0 00 00 00 09 00 05 FF FF FF FF 89 26 02 05 02
Api_2_CNs_Ste_6.java	A0 00 00 00 09 00 05 FF FF FF FF 89 26 02 06 02
Api_2_CNs_Stt_1.java	A0 00 00 00 09 00 05 FF FF FF FF 89 26 03 01 02
Api_2_CNs_Stt_2.java	A0 00 00 00 09 00 05 FF FF FF FF 89 26 03 02 02
Api_2_CNs_Stt_3.java	A0 00 00 00 09 00 05 FF FF FF FF 89 26 03 03 02
Api_2_CNs_Stt_4.java	A0 00 00 00 09 00 05 FF FF FF FF 89 26 03 04 02
Api_2_CNs_Stt_5.java	A0 00 00 00 09 00 05 FF FF FF FF 89 26 03 05 02

B.2.3 Package readermode

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

B.2.3.1 Interface ReaderMessage

Package name	AID
uicc.test.hci.services.reader.Api_2_RMm_Rrp	A0 00 00 00 09 00 05 FF FF FF FF 89 27 01 00 00
uicc.test.hci.services.reader.Api_2_RMm_Srx	A0 00 00 00 09 00 05 FF FF FF FF 89 27 02 00 00
uicc.test.hci.services.reader.Api_2_RMm_Sgp	A0 00 00 00 09 00 05 FF FF FF FF 89 27 03 00 00

Applet name	AID
Api_2_RMm_Rrp_1.java	A0 00 00 00 09 00 05 FF FF FF FF 89 27 01 01 02
Api_2_RMm_Rrp_2.java	A0 00 00 00 09 00 05 FF FF FF FF 89 27 01 02 02
Api_2_RMm_Rrp_3.java	A0 00 00 00 09 00 05 FF FF FF FF 89 27 01 03 02
Api_2_RMm_Srx_1.java	A0 00 00 00 09 00 05 FF FF FF FF 89 27 02 01 02
Api_2_RMm_Srx_2.java	A0 00 00 00 09 00 05 FF FF FF FF 89 27 02 02 02
Api_2_RMm_Srx_3.java	A0 00 00 00 09 00 05 FF FF FF FF 89 27 02 03 02
Api_2_RMm_Sgp_1.java	A0 00 00 00 09 00 05 FF FF FF FF 89 27 03 01 02
Api_2_RMm_Sgp_2.java	A0 00 00 00 09 00 05 FF FF FF FF 89 27 03 02 02
Api_2_RMm_Sgp_3.java	A0 00 00 00 09 00 05 FF FF FF FF 89 27 03 03 02

B.2.3.2 Interface ReaderListener

Package name	AID
uicc.test.hci.services.reader.Api_2_RMl_Ocb	A0 00 00 00 09 00 05 FF FF FF FF 89 27 04 00 00

Applet name	AID
Api_2_Rml_Ocb_1.java	A0 00 00 00 09 00 05 FF FF FF FF 89 27 04 01 02

B.2.3.3 Interface ReaderService

FFS

Annex C (normative):
Requirements

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

C.1.1 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 ETSI 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 ETSI 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 Api_2_RMm_Sgp
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 ETSI 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
Returns -1 if the power mode cannot be retrieved.	Api_1_Hdv_Gpm

C.2.2   ETSI TS 102 705 prose part

Requirement	Reference
When the contactless interface is disabled (cf. "state of contactless functionality" in ETSI TS 102 223 [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 ETSI 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):
Void

Annex F (informative):
Void

Annex G (informative):
Core specification version information

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

Release	Latest version from which conformance requirements have been extracted
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
		SCP(13)000039	002	-	F	Clarification of initial conditions activity performed by UICC	9.0.0	9.1.0
		SCP(13)000040	003	-	F	Test case 6.1.5.1: corrected definition of unrecognised Envelope	9.0.0	9.1.0
		SCP(13)000041	004	-	F	Correction of expected HCI event for TC 6.2.2.1.2.4 and 6.2.2.1.3.4 ID2-1 and ID2-2	9.0.0	9.1.0
		SCP(13)000042	005	-	F	Clarification of prepareAndSendGetParameterCommand test cases	9.0.0	9.1.0
		SCP(13)000043	006	-	F	Correction of test suite files	9.0.0	9.1.0
		SCP(13)000044	007	-	F	Correction of wrong Events deactivation	9.0.0	9.1.0
		SCP(13)000046	008	-	F	Clarification of initial conditions related to power mode	9.0.0	9.1.0
		SCP(13)000085r1	010	-	F	Modification and addition of test applets to test HCI is disabled condition	9.0.0	9.1.0
2013-04	SCP#59	SCP(13)000086	011	1	F	Addition of references to GlobalPlatform APIs.	9.0.0	9.1.0
		SCP(13)000087r1	012	-	F	Clarification of EVT_READER_REQUESTED transmission	9.0.0	9.1.0
		SCP(13)000088	013	1	F	Correction of applet selection in TC 6.2.1.1.1 ID3	9.0.0	9.1.0
		SCP(13)000089	014	-	F	Correction of API expected received data in TC 6.1.3.7	9.0.0	9.1.0
		SCP(13)000090	015	-	F	Correction of applicability for test cases 6.1.1.1 (ID3-2) 6.1.1.2 and 6.1.2.3 (ID5)	9.0.0	9.1.0
		SCP(13)000091	016	-	F	Modification of test procedure 6.1.1.3.4 ID3-1 and 6.2.2.1.1.4	9.0.0	9.1.0
		SCP(13)000092	017	-	F	Addition of Annexes E and F contents into ETSI TS 103 115	9.0.0	9.1.0
		SCP(13)000093	018	-	F	Corrections to avoid failed deletions of test applets	9.0.0	9.1.0
		SCP(13)000037	019	-	F	Corrections on Reader Mode applets	9.0.0	9.1.0
2013-07	SCP#60	SCP(13)000045r1	009	1		Usage of Annex F is made fully optional	9.1.0	9.2.0
		SCP(13)000188	021	-		Deletion of getResponse() method from Annex F	9.1.0	9.2.0
		SCP(13)000140r1	022	1	F	Corrections on Reader Mode test cases and proposal of Test Case 4 from 6.2.3.1.3.4 for FFS	9.1.0	9.2.0
		SCP(13)000141r1	023	1	F	Correction of 6.2.2.1.2 method prepareAndSendTransactionEvent() test case ID5	9.1.0	9.2.0
		SCP(13)000138	020	-	F	Annex E update with Reader Mode features used in Annex F	9.1.0	9.2.0
2013-10	SCP#61	SCP(13)000217	024	-	F	Correction of applet selection issues	9.2.0	9.3.0
		SCP(13)000218	025	-	F	Corrections in applicability table	9.2.0	9.3.0
		SCP(13)000220	026	-	F	Correction of reader mode test cases regarding applet selection and usage of ISO interface	9.2.0	9.3.0
		SCP(13)000222	027	-	F	Correction of TC 6.1.3.3.4 ID2	9.2.0	9.3.0
		SCP(13)000219	028	-	F	Test cases 6.1.1.1/5: correction of execution requirements	9.2.0	9.3.0
		SCP(13)000221	029	-	F	Modification of Terminal profile in the initial conditions for some test cases	9.2.0	9.3.0
		SCP(13)000223	030	-	F	Removal of redundant entries in applicability table	9.2.0	9.3.0
2014-02	SCP#62	SCP(14)000019r1	031	1	F	Test case 6.1.5.1 ID2: specification of Terminal Profile and response to ENVELOPE command	9.3.0	9.4.0
		SCP(14)000020r1	032	1	F	Correction of some test cases to add SAA1	9.3.0	9.4.0
		SCP(14)000021r1	033	1	F	Corrections on AID coding section	9.3.0	9.4.0
		SCP(14)000022r1	034	1	F	Test case 6.1.1.1 ID5-2: correction of execution requirements	9.3.0	9.4.0
		SCP(14)000023r1	035	1	F	Test case 6.1.2.1 ID2: correction of execution requirements	9.3.0	9.4.0
		SCP(14)000024r1	036	1	F	Test case 6.1.1.3 ID 5-2: correction of expected status word	9.3.0	9.4.0
		SCP(14)000025r1	037	1	F	Connectivity test cases: correction related to ordering of events	9.3.0	9.4.0
		SCP(14)000026r1	038	1	F	Connectivity test cases involving wrong precondition: correction of initial conditions	9.3.0	9.4.0
		SCP(14)000027r1	039	1	F	Correction of uninstall() test cases to implement AppletEvent interface	9.3.0	9.4.0
		SCP(14)000029r1	041	1	F	Test case 6.1.1.3 ID5-1/2: HCI selection removed	9.3.0	9.4.0
		SCP(14)000030r1	043	1	D	Alignment of enable/disable contactless interface terminology with ETSI TS 102 705	9.3.0	9.4.0

Date	Meeting	Plenary Doc	CR	Rev	Cat	Subject/Comment	Old	New
2014-04	SCP#63	SCP(14)000119r1	040	2	F	Correction and clarification of initial conditions and test cases.	9.3.0	9.4.0
		SCP(14)000114r1	044	1	F	Corrections on CRRN3 of onCallback method (Card Emulation)	9.3.0	9.4.0
		SCP(14)000115r1	045	1	F	Corrections on test case Id1 of onCallback method (Card Emulation)	9.3.0	9.4.0
		SCP(14)000116	046	-	F	Addition of AID for Api_1_Hsr_Ace_3	9.3.0	9.4.0
		SCP(14)000117r1	047	1	D	Corrections on the wording in test cases 6.2.3.1.2.4 and 6.2.3.2.1.4	9.3.0	9.4.0
		SCP(14)000118	048	-	F	Reader mode test cases: corrections related to ISO APDUs	9.3.0	9.4.0
2014-06	SCP#64	SCP(14)000158	049	-	F	prepareAndSendWriteXchgDataCommand test case: improvement of test case coverage	9.3.0	9.4.0
		SCP(14)000159	050	-	F	Clarification on test cases ID3 and ID6 of onCallback method (Card Emulation)	9.3.0	9.4.0
		SCP(14)000161	051	-		Unknown_Power_Mode	9.3.0	9.4.0
		SCP(14)000160	052	-	F	RF_Transmission_Errror	9.3.0	9.4.0
2014-12	SCP#66	SCP(14)000316	053	-	F	Test case 6.1.2.3: correction of execution requirements and applicability	9.3.0	9.4.0
2015-02	SCP#67	SCP(15)000027	055	-	F	Removal of Annexes E and F	9.3.0	9.4.0
		SCP(15)000028r1	056	1	F	Correction of requirements CRRN4 and CRRN3 in 6.2.2.1.1/2	9.3.0	9.4.0
		SCP(15)000029	057	-	B	Connectivity test cases: addition of reader mode scenarios	9.3.0	9.4.0
		SCP(15)000030	058	-		Correction of Api_2_CEI_Ocb_1 (used in test case 6.2.1.3.1)	9.3.0	9.4.0

History

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