



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

PREVIEW  
iTech Standards (standards.itg-ai)  
https://standards.itg-ai.com/standards/sist/e219e2e5-5341-487e-9c61-5430e9734b0e/etsi-ts-103-115-v9.4.0-

## Reference

---

RTS/SCP-00HCL\_API\_TESTv940

## Keywords

---

API, NFC, Smart Card, testing

**ETSI**

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

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

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

**Important notice**

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

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

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

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

**Copyright Notification**

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.

The content of the PDF version shall not be modified without the written authorization of ETSI.  
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2015.  
All rights reserved.

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

# Contents

Intellectual Property Rights .....	7
Foreword.....	7
Modal verbs terminology.....	7
1 Scope .....	8
2 References .....	8
2.1 Normative references .....	8
2.2 Informative references.....	9
3 Definitions, symbols and abbreviations .....	9
3.1 Definitions.....	9
3.2 Symbols.....	9
3.3 Abbreviations .....	10
3.4 Formats.....	10
3.4.1 Format of the table of optional features .....	10
3.4.2 Format of the applicability table .....	11
3.4.3 Status and Notations .....	11
4 Applicability.....	12
4.1 Table of optional features.....	12
4.2 Applicability table .....	12
4.3 Information provided by the device supplier.....	15
4.4 Execution requirements .....	15
5 Test environment.....	16
5.1 Test environment description .....	16
5.2 Tests format.....	16
5.2.1 Test area reference .....	16
5.2.1.1 Conformance requirements .....	17
5.2.1.2 Test suite files .....	17
5.2.1.3 Initial conditions .....	17
5.2.1.4 Test procedure.....	17
5.3 Initial conditions.....	18
5.4 Package name.....	18
5.5 AID coding .....	18
5.6 Test equipment .....	19
5.6.1 Test tool .....	19
5.6.2 Java Software Development Kit .....	20
6 Test cases.....	20
6.1 Package uicc.hci.framework.....	20
6.1.1 Class HCIDevice .....	20
6.1.1.1 Method getHCIService.....	20
6.1.1.1.1 Conformance requirements.....	20
6.1.1.1.2 Test suite files.....	20
6.1.1.1.3 Initial conditions .....	20
6.1.1.1.4 Test procedure .....	21
6.1.1.2 Method getPowerMode.....	22
6.1.1.2.1 Conformance requirements.....	22
6.1.1.2.2 Test suite files.....	22
6.1.1.2.3 Initial conditions .....	22
6.1.1.2.4 Test procedure .....	23
6.1.1.3 Method isHCIServiceAvailable .....	23
6.1.1.3.1 Conformance requirements.....	23
6.1.1.3.2 Test suite files.....	24
6.1.1.3.3 Initial conditions .....	24
6.1.1.3.4 Test procedure .....	24
6.1.2 Interface HCIService .....	26
6.1.2.1 Method register .....	26

6.1.2.1.1	Conformance requirements.....	26
6.1.2.1.2	Test suite files.....	26
6.1.2.1.3	Initial conditions.....	26
6.1.2.1.4	Test procedure.....	27
6.1.2.2	Method deregister.....	28
6.1.2.2.1	Conformance requirements.....	28
6.1.2.2.2	Test Suite Files.....	28
6.1.2.2.3	Initial conditions.....	28
6.1.2.2.4	Test procedure.....	28
6.1.2.3	Method activateEvent.....	29
6.1.2.3.1	Conformance requirements.....	29
6.1.2.3.2	Test Suite Files.....	30
6.1.2.3.3	Initial conditions.....	30
6.1.2.3.4	Test procedure.....	31
6.1.2.4	Method deactivateEvent.....	33
6.1.2.4.1	Conformance requirements.....	33
6.1.2.4.2	Test suite files.....	34
6.1.2.4.3	Initial conditions.....	34
6.1.2.4.4	Test procedure.....	35
6.1.2.5	Method requestCallbackNotification.....	38
6.1.2.5.1	Conformance requirements.....	38
6.1.2.5.2	Test Suite Files.....	38
6.1.2.5.3	Initial conditions.....	38
6.1.2.5.4	Test procedure.....	38
6.1.2.6	Method getEventNotificationStatus.....	39
6.1.2.6.1	Conformance requirements.....	39
6.1.2.6.2	Test Suite Files.....	39
6.1.2.6.3	Initial conditions.....	39
6.1.2.6.4	Test procedure.....	40
6.1.3	Interface HCIMessage.....	43
6.1.3.1	Method isHeading.....	43
6.1.3.1.1	Conformance requirements.....	43
6.1.3.1.2	Test Suite Files.....	43
6.1.3.1.3	Initial conditions.....	43
6.1.3.1.4	Test procedure.....	44
6.1.3.2	Method isComplete.....	44
6.1.3.2.1	Conformance requirements.....	44
6.1.3.2.2	Test suite files.....	44
6.1.3.2.3	Initial condition.....	44
6.1.3.2.4	Test procedure.....	44
6.1.3.3	Method getType.....	45
6.1.3.3.1	Conformance requirements.....	45
6.1.3.3.2	Test Suite Files.....	45
6.1.3.3.3	Initial condition.....	45
6.1.3.3.4	Test procedure.....	46
6.1.3.4	Method getInstruction.....	46
6.1.3.4.1	Conformance requirements.....	46
6.1.3.4.2	Test Suite Files.....	46
6.1.3.4.3	initial condition.....	46
6.1.3.4.4	Test procedure.....	47
6.1.3.5	Method getReceiveOffset.....	47
6.1.3.5.1	Conformance requirements.....	47
6.1.3.6	Method getReceiveLength.....	47
6.1.3.6.1	Conformance requirements.....	47
6.1.3.6.2	Test Suite Files.....	48
6.1.3.6.3	Initial condition.....	48
6.1.3.6.4	Test procedure.....	48
6.1.3.7	Method getReceiveBuffer.....	48
6.1.3.7.1	Conformance requirements.....	48
6.1.3.7.2	Test suite files.....	49
6.1.3.7.3	Initial condition.....	49
6.1.3.7.4	Test procedure.....	49

6.1.4	Class HCIException.....	49
6.1.4.1	Method throwIt .....	49
6.1.4.1.1	Conformance requirements.....	49
6.1.4.1.2	Test suite files.....	49
6.1.4.1.3	Initial conditions.....	49
6.1.4.1.4	Test procedure .....	50
6.1.5	Interface HCIListener .....	50
6.1.5.1	Method onCallback .....	50
6.1.5.1.1	Conformance requirements.....	50
6.1.5.1.2	Test Suite Files .....	51
6.1.5.1.3	Initial conditions.....	51
6.1.5.1.4	Test procedure .....	52
6.2	Package uicc.hci.services .....	53
6.2.1	Package CardEmulation Service.....	53
6.2.1.1	Interface CardEmulationMessage .....	53
6.2.1.1.1	Method prepareAndSendGetParameterCommand .....	53
6.2.1.1.2	Method prepareAndSendSendDataEvent .....	56
6.2.1.1.2.3	Initial conditions.....	57
6.2.1.1.3	Method selectingMessage.....	58
6.2.1.1.3.3	Initial conditions.....	58
6.2.1.2	Interface CardEmulationService .....	59
6.2.1.2.1	Method getCardRFTYPE .....	59
6.2.1.2.2	Test Suite Files .....	59
6.2.1.2.3	Initial conditions.....	59
6.2.1.3	Interface CardEmulationListener .....	60
6.2.1.3.1	Method onCallback .....	60
6.2.2	Package Connectivity Service.....	63
6.2.2.1	Interface ConnectivityService.....	63
6.2.2.1.1	Method prepareAndSendConnectivityEvent .....	63
6.2.2.1.2	Method prepareAndSendTransactionEvent(byte[] aid, short aidOffset, short aidLen, byte[] parameters, short parametersOffset, short parametersLen).....	67
6.2.2.1.3	Method prepareAndSendTransactionEvent (byte[] parameters, short parametersOffset, short parametersLen).....	71
6.2.3	Package Reader Service.....	74
6.2.3.1	Interface ReaderMessage .....	74
6.2.3.1.1	Method restartReaderModeProcedure .....	74
6.2.3.1.2	Method prepareAndSendWriteXchgDataCommand .....	77
6.2.3.1.3	Method prepareAndSendGetParameterCommand .....	82
6.2.3.2	Interface ReaderListener .....	86
6.2.3.2.1	Method onCallback .....	86
<b>Annex A (normative):</b>	<b>Class, methods and tests acronyms .....</b>	<b>89</b>
A.1	HCI framework .....	89
A.1.1	Class HCIDevice .....	89
A.1.2	Interface HCIService.....	89
A.1.3	Interface HCIMessage.....	89
A.1.4	Interface HCIListener.....	89
A.1.5	Class HCIException .....	89
A.2	HCI Services .....	90
A.2.1	Package cardemulation.....	90
A.2.1.1	Interface CardEmulationListener.....	90
A.2.1.2	Interface CardEmulationMessage.....	90
A.2.1.3	Interface CardEmulationService.....	90
A.2.2	Package connectivity.....	90
A.2.2.1	Interface ConnectivityListener.....	90
A.2.2.2	Interface ConnectivityMessage.....	90
A.2.2.3	Interface ConnectivityService.....	90
A.2.3	Readermode.....	90
A.2.3.1	Interface RaederListener.....	90
A.2.3.2	Interface ReaderMessage .....	91
A.2.3.3	Interface ReaderService .....	91

<b>Annex B (normative):</b>	<b>AIDs - to be reserved .....</b>	<b>92</b>
B.1	Package HCI framework .....	92
B.1.1	Class HCIDevice .....	92
B.1.2	Interface HCIService .....	92
B.1.3	Interface HCIMessage .....	93
B.1.4	Interface HCIListener .....	93
B.1.5	Class HCIException .....	93
B.2	HCI Services .....	93
B.2.1	Package cardemulation .....	93
B.2.1.1	Interface CardEmulationListener .....	93
B.2.1.2	Interface CardEmulationMessage .....	94
B.2.1.3	Interface CardEmulationService .....	94
B.2.2	Package connectivity .....	94
B.2.2.1	Interface ConnectivityListener .....	94
B.2.2.2	Interface ConnectivityMessage .....	94
B.2.2.3	Interface ConnectivityService .....	95
B.2.3	Package readermode .....	95
B.2.3.1	Interface ReaderMessage .....	95
B.2.3.2	Interface ReaderListener .....	95
B.2.3.3	Interface ReaderService .....	96
<b>Annex C (normative):</b>	<b>Requirements .....</b>	<b>97</b>
C.1	Non-occurrence and out-of-scope requirements .....	97
C.1.1	Package uicc.hci .....	97
C.1.2	ETSI TS 102 705 prose part .....	97
C.2	FFS requirements .....	97
C.2.1	Package uicc.hci .....	97
C.2.2	ETSI TS 102 705 prose part .....	98
<b>Annex D (normative):</b>	<b>Test Specification for Java Card™ Platform HCI API for the UICC .....</b>	<b>99</b>
<b>Annex E (normative):</b>	<b>Void .....</b>	<b>100</b>
<b>Annex F (informative):</b>	<b>Void .....</b>	<b>101</b>
<b>Annex G (informative):</b>	<b>Core specification version information .....</b>	<b>102</b>
<b>Annex H (informative):</b>	<b>Change history .....</b>	<b>103</b>
History .....		105

---

## Intellectual Property Rights

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

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 a unique conditional status expression which is defined immediately following the table. For nested conditional expressions, the syntax "IF ... THEN (IF ... THEN ... ELSE...) ELSE ..." is to be used to avoid ambiguities.

#### References to items

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

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

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

## 4 Applicability

### 4.1 Table of optional features

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

**Table 4.1: Options**

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

### 4.2 Applicability table

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

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

**Table 4.2 a): Applicability of tests**

Clause	Test case number and description	Release	Execution requirements	Rel-9 UICC	Support
6.1.1.1	Method getHCIService (ID2, ID4)	Rel-9	SAA1	M	
6.1.1.1	Method getHCIService (ID1-1)	Rel-9	SAA1	M	
6.1.1.1	Method getHCIService (ID1-2)	Rel-9	SAA2, SAA1	M	
6.1.1.1	Method getHCIService (ID1-3)	Rel-9	SAA3	M	
6.1.1.1	Method getHCIService (ID3-1)	Rel-9	SAA6	M	
6.1.1.1	Method getHCIService (ID3-2)	Rel-9	SAA1, SAA7	M	
6.1.1.1	Method getHCIService (ID3-3)	Rel-9	SAA8	M	
6.1.1.1	Method getHCIService (ID5-1)	Rel-9	SAA4, 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	