



Smart Cards; UICC Application Programming Interface for Java Card™ for Contactless Applications (Release 11)

PREVIEW
iTech Standards (mailto:info@iteh.ai)
https://standards.iteh.ai/catalog/standards/sist/fe14d610-d73c-4e8d-9449-53896cd65c92/etsi-ts-102-705-v11.0.0-2013-09

ReferenceRTS/SCP-THCIAPIVB00

Keywords

API, smart card

ETSI

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

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

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

Important notice

Individual copies of the present document can be downloaded from:

<http://www.etsi.org>

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

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at

<http://portal.etsi.org/tb/status/status.asp>

If you find errors in the present document, please send your comment to one of the following services:

http://portal.etsi.org/chaicor/ETSI_support.asp

Copyright Notification

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

© European Telecommunications Standards Institute 2013.
All rights reserved.

DECTTM, PLUGTESTSTM, UMTSTM and the ETSI logo are Trade Marks of ETSI registered for the benefit of its Members.
3GPPTM and LTETM 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	4
Foreword.....	4
1 Scope	5
2 References	5
2.1 Normative references	5
2.2 Informative references.....	6
3 Definitions and abbreviations.....	6
3.1 Definitions.....	6
3.2 Abbreviations	6
4 Description	7
4.1 Architecture	7
4.2 Card Emulation Mode	9
4.3 Reader Mode	9
4.4 Connectivity Service	10
5 Interaction with Proactive Functionality	10
6 Java Card Resource Handling	11
Annex A (normative): Java Card™ Platform HCI API for the UICC	12
Annex B (normative): Java Card™ Platform HCI API for the UICC identifiers	13
Annex C (normative): HCI API package version management.....	14
Annex D (informative): Change history	15
History	16

Intellectual Property Rights

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

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

Foreword

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

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

Version x.y.z

where:

- x the first digit:
 - 0 early working draft;
 - 1 presented to TC SCP for information;
 - 2 presented to TC SCP for approval;
 - 3 or greater indicates TC SCP approved document under change control.
- y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.
- z the third digit is incremented when editorial only changes have been incorporated in the document.

1 Scope

The present document describes the UICC Application Programming Interface for Java Card™ for contactless Applications. Its purpose is to provide access for a contactless Applet to the services provided by the HCI protocol defined in [4] for the communication via the CLF. In the scope of the present document contactless means support for the RF Technologies referenced by the HCI specification [4]. Low level functionality to manage gates and pipes as defined in the HCI specification [4] is not in the scope of the present document. Registration of contactless parameters and management of contactless Applets in card emulation mode is defined in "GlobalPlatform Card Specification Amendment C" [8]. Related APIs are provided in "Java Card API and Export File for Card Specification v2.2.1 (org.globalplatform) v1.5" [12] and "Java Card Contactless API and Export File for Card Specification v2.2.1 (org.globalplatform.contactless)" [13].

2 References

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

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

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

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

2.1 Normative references

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

- [1] ISO/IEC 7816-3 (2006): "Identification cards - Integrated circuit cards - Part 3: Cards with contacts - Electrical interface and transmission protocols".
- [2] ETSI TS 102 221: "Smart Cards; UICC-Terminal interface; Physical and logical characteristics".
- [3] ETSI TS 101 220: "Smart Cards; ETSI numbering system for telecommunication application providers".
- [4] ETSI TS 102 622: "Smart Cards; UICC - Contactless Front-end (CLF) Interface; Host Controller Interface (HCI)".
- [5] ETSI TS 102 241: "Smart Cards; UICC Application Programming Interface (UICC API) for Java Card (TM)".
- [6] ETSI TS 102 223: "Smart Cards; Card Application Toolkit (CAT)".
- [7] ETSI TS 102 226: "Smart Cards; Remote APDU structure for UICC based applications".
- [8] GlobalPlatform: "GlobalPlatform Card Specification Version 2.2, Amendment C: Contactless Services" Version 1.0.

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

- [9] Sun Microsystems "Application Programming Interface, Java Card™ Platform, 3.0.1 Classic Edition".
- [10] Sun Microsystems "Runtime Environment Specification, Java Card™ Platform, 3.0.1 Classic Edition".

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

NOTE: SUN Java Card Specifications can be downloaded at <http://java.sun.com/products/javacard/specs.html>.

[12] GlobalPlatform: "Java Card API and Export File for Card Specification v2.2.1 (org.globalplatform)" v1.5.

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

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

2.2 Informative references

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

Not applicable.

3 Definitions and abbreviations

3.1 Definitions

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

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

contactless state: corresponds to the logical state of the contactless framework

HCP message: message as specified in TS 102 622 [4]

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

RF Technology: radio frequency technology supported by the HCI (TS 102 622 [4]) protocol specification

3.2 Abbreviations

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

APDU Application Protocol Data Unit

NOTE: According to ISO/IEC 7816-3 [1].

API Application Programming Interface

CLF Contactless Front-end

NOTE: According to TS 102 622 [4].

CRS Contactless Registry Service

HCI Host Controller Interface

NOTE: According to TS 102 622 [4].

HCP Host Controller Protocol

NOTE: According to TS 102 622 [4].

RF Radio Frequency

SWP Single Wire Protocol

NOTE: According to TS 102 613 [14].

4 Description

4.1 Architecture

The present document describes an API and a Contactless Framework that enables Java Card™ Platform based Applets, defined in [9], [10] and [11], to send and receive messages using the HCI protocol as specified in TS 102 622 [4] and to act as contactless Applets. The Contactless Framework shall support card emulation mode and reader mode as specified in the HCI protocol (TS 102 622 [4]) specification.

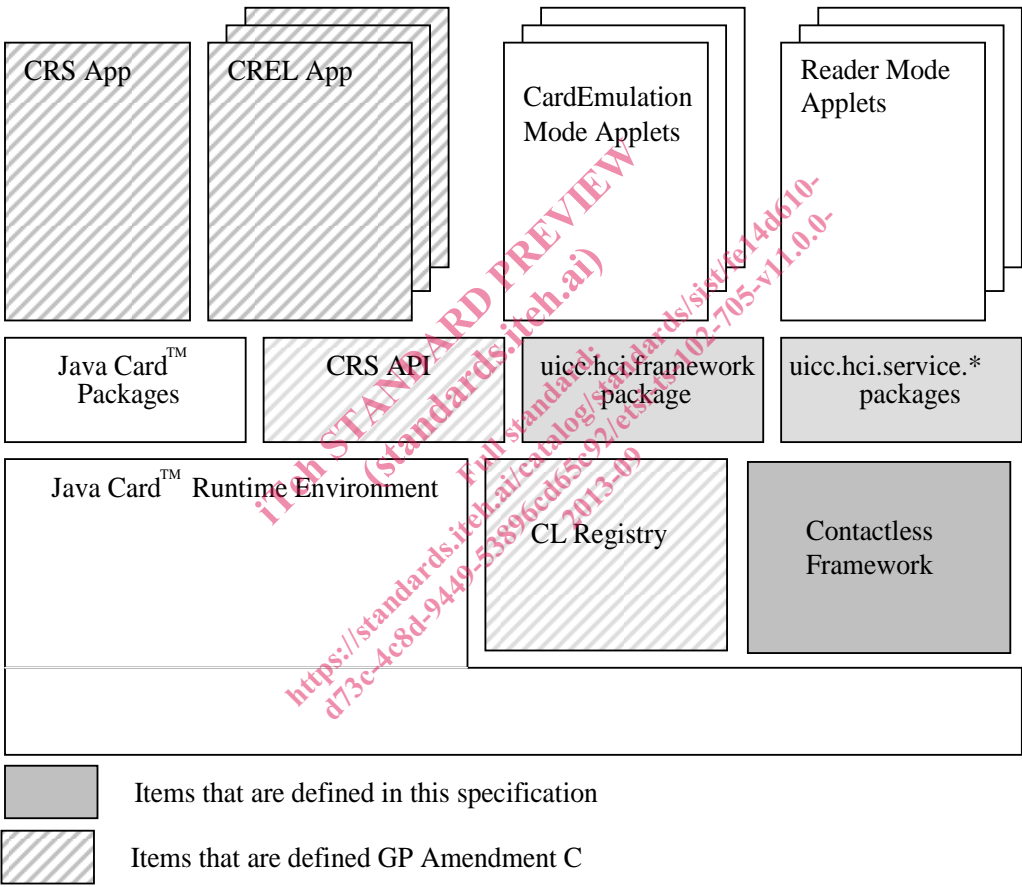


Figure 1

The functionality of the Contactless Framework and the configuration of contactless parameters and the management of contactless Applets in card emulation mode are based on the functionality provided by the Contactless Registry Service (CRS), the related APIs, the CRS Application and other features and concepts which are defined in the "GlobalPlatform Amendment C" [8] and the related APIs "Java Card API and Export File for Card Specification v2.2.1 (org.globalplatform) v1.5" [12] and "Java Card Contactless API and Export File for Card Specification v2.2.1 (org.globalplatform.contactless)" [13].

The API is event driven and based on the Observer/Listener pattern. Every HCI service is encapsulated by a dedicated Service interface. These Service interfaces shall allow the registration of Listener Interfaces and the activation of events. The Listener Interfaces shall be implemented by Java objects to receive HCI messages and events in the *onCallback* method. The Registration of Listener Interfaces and activation of events shall be persistent.

An *HCIMessage* object shall encapsulates one HCP message according to the HCI protocol as specified in TS 102 622 [4]. HCI message for the different contactless modes shall be identified by different types of interfaces. It is not guaranteed that any Applet originated HCI messages are sent before the completion of the execution of the current Applet. The Contactless Framework sends the Applet originated HCI messages in the same order as they are submitted by the Applet.

NOTE 1: The Contactless Framework may not have enough resources to send several HCI messages submitted during the same *onCallback* method execution. The Applet should be aware of this limitation (e.g. use suitable error handling strategy, or send only one HCI message in the *onCallback* method at a time).

Any *onCallback()* method of a Listener interface shall not be invoked again while another *onCallback()* method is still being executed. 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.

The HCI event *EVT_FIELD_OFF* shall be buffered and sent by the Contactless Framework as soon as the Contactless Framework becomes the current context.

All other HCI messages shall be delivered to the Applet instance in the same order as they were received by the Contactless Framework.

Contactless State is the logical state of the Contactless Framework it can take the value enabled and disabled. It refers to the "contactless functionality in the UICC" as used in TS 102 223 [6].

This state can be changed with the mechanisms defined in TS 102 223 [6], and by the method *setCommunicationInterface()* API method of "GlobalPlatform Amendment C" [8].

The Contactless State applies only to the Card Emulation Mode and the Reader Mode, and it does not apply to the Connectivity service.

When the Contactless State is disabled, the Contactless Framework shall throw an *HCIException* with reason code *HCI_CURRENTLY_DISABLED* when an Applet invokes a method which requires that the Contactless State is enabled.

When the Contactless State is enabled and the state of the SWP [14] interface is *DEACTIVATED* and when the Contactless Framework needs to send data over the SWP [14] interface then it shall send the proactive command *ACTIVATE* defined in TS 102 223 [6] if supported by the terminal.

NOTE 2: An Applet may use the method *HCIDevice.isHCIServiceAvailable()* to check if the Contactless Framework supports sending the *ACTIVATE* command on pre Rel-11 implementations.

The API is split into two parts. One is a generic framework that provides a factory class to retrieve the different Service instances that are provided by the HCI implementation, and that allows discovery of whether the UICC is inserted into a HCI network. The second part of the API implements the Services that are defined for the HCI protocol, card emulation mode, reader mode and connectivity service.

As stated in TS 102 622 [4], 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. In the case of a data loss an error shall be reported back to the Application which has initiated the data transmission. In case of a communication in CardEmulation Mode where a command from an external reader cannot be forwarded to an Application due to a data link layer problem the Contactless Framework shall not take any action but wait for the reader to resolve the problem.

4.2 Card Emulation Mode

In card emulation mode there exist two exclusive ways to exchange messages over the HCP [4]. The first is based on APDUs provided to the Applet through its *process()* method as specified in "Application Programming Interface, Java Card™ Platform, 3.0 Classic Edition" [9]. The second is made available by the package *uicc.hci.services.cardemulation* defined in this specification.

The *uicc.hci.services.cardemulation* package shall provide the communication technologies for the card emulation mode defined by the HCP as specified in [4]. 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 in the HCI architecture as specified in [4]. The parameters to be used by the HCI layer may be provided to the framework as defined in "GlobalPlatform Amendment C" [8].

For the API defined in this specification the card emulation capability shall be provided to Applets through a service interface implemented by the Contactless Framework. 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. 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.

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

If the current application was selected through a *SELECT* by DF name, the Contactless Framework shall handle an application session termination according to TS 102 221 [2] independent of the interface used for APDU exchange.

Applet selection and deselection shall be performed by the Contactless Framework according to the rules defined in the "Java Card™ Runtime Environment Specification, 3.0.1 Classic Edition" [10] and in "GlobalPlatform Amendment C" [8].

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, 3.0.1 Classic Edition" [10].

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. The HCP message that resulted in the selection of this Applet according to the rules defined in "GlobalPlatform Amendment C" [8] shall be provided by the *CardEmulationMessage*.

If the HCI event *EVT_FIELD_OFF* or *EVT_CARD_DEACTIVATED* defined by the HCP specified in TS 102 622 [4] is received by the Contactless Framework and the UICC is still powered, the Applet instance shall be deselected according to "GlobalPlatform Amendment C" [8] by invocation of the *deselect()* method.

When the HCI event *EVT_FIELD_OFF* is received and if the Applet instance has activated this event the Contactless Framework shall raise an *EVENT_FIELD_OFF* before the invocation of the *deselect()* method of the Applet instance.

After the deselection of the Applet instance, it shall not be invoked by any other event defined in the interface *CardEmulationListener* until the Applet instance is selected again.

4.3 Reader Mode

The functionality to support the reader mode is provided in the package *uicc.hci.services.reader*. In reader mode the communication technologies defined by the contactless platform for reader mode [4] are supported. The Contactless Framework shall bind the services defined in *uicc.hci.services.reader* to the corresponding resources (e.g. gates and pipes) defined by the contactless platform for reader mode [4].

An Applet has to be in the selectable state (according to the Java Card™ specification [9], [10] and [11]) to act as a contactless Applet in reader mode.