ETSI TS 131 131 V15.0.0 (2019-04)

Universal Mobile Telecommunications System (UMTS); LTE; C-language binding to (U)SIM API (3GPP TS 31.131 version 15.0.0 Release 15)

Reference
RTS/TSGC-0631131vf00

Keywords
LTE,UMTS

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 prevailing version of an ETSI deliverable is the one made publicly available in PDF format at www.etsi.org/deliver.

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
https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

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

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables 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 (https://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.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Foreword

This Technical Specification (TS) has been produced by ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities, UMTS identities or GSM identities. These should be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between GSM, UMTS, 3GPP and ETSI identities can be found under http://webapp.etsi.org/key/queryform.asp.

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.

Contents

Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG 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

A Subscriber Identity Module Application Programming Interface (SIM API) has been defined in TS 42.019 [4] as a technology-independent API by which toolkit applications and (U)SIMs co-operate. That specification is independent of the programming language technology used to create the application, the platform used to host the application and the runtime environment used to execute the application.

The present document includes information applicable to (U)SIM toolkit application developers creating applications using the C programming language ISO/IEC 9899 [7]. The present document describes an interface between toolkit applications written in the C programming language and the (U)SIM in order to realize the co-operation set forth in TS 42.019 [4]. In particular, the API described herein provides the service of assembling proactive commands and disassembling the responses to these commands for the application programmer.

Software tools, integrated software development environments and software management systems that may be used to create application programs are explicitly out of scope of the present document.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] 3GPP TS 31.111: "USIM Application Toolkit (USAT)".

[3] 3GPP TS 23.048: "Security Mechanisms for the (U)SIM application toolkit; Stage 2".

[4] 3GPP TS 42.019: "Subscriber Identity Module Application Programming Interface (SIM API); Stage 1".

[5] ISO 639 (1988): "Code for the representation of names of languages".

[6] 3GPP TS 23.038: "Alphabets and language-specific information".

[7] ISO/IEC 9899: "Programming Languages - C".

[8] 3GPP TS 11.14: "Specification of the SIM Application Toolkit for the Subscriber Identity Module - Mobile Equipment (SIM – ME) interface".

[9] Tool Interface Standard (TIS) Executable and Linking Format Specification Version 1.2.

[10] SYSTEM V Application Binary Interface, Edition 4.1.

[11] 3GPP TS 51.011: "Specification of the Subscriber Identity Module - Mobile Equipment (SIM-ME) interface".

[12] Void.

[13] 3GPP TS 31.115: "Secured packet structure for (U)SIM Toolkit applications".

[14] 3GPP TS 31.116: "Remote APDU Structure for (U)SIM Toolkit applications".

[15] 3GPP TS 31.102: "Characteristics of the USIM Application".

[16] 3GPP TS 31.101: "UICC-Terminal Interface, Physical and Logical Characteristics".

3 Definitions and abbreviations

3.1 Definitions

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

application: computer program that defines and implements a useful domain-specific functionality
The term may apply to the functionality itself, to the representation of the functionality in a programming language, or to the realization of the functionality as executable code.

application executable: representation of an application as collection of executable codes

application program: representation of an application in a programming language such as assembly language, C, Java, WML or XHTML

Application Programming Interface (API): collection of entry points and data structures that an application program can access when translated into an application executable

byte code: processor-independent representation of a basic computer operation such as "increment by one" that is executed by computer program called a byte code interpreter

data structure: memory address that can be accessed by an application executable in order to read or write data

entry point: memory address that can be branched to by an application executable in order to access functionality defined by an application-programming interface
Depending on the software technology, an entry point is also called a subroutine, a function or a method.

executable code: generic term for either byte code or native code

framework: defines a set of Application Programming Interface (API) functions for developing applications and for providing system services to those applications

loadfile: representation of an application executable that is transmitted from the terminal to the smart card operating system
A loadfile typically includes information about the application executable in addition to the application executable itself.

native code: processor-dependent representation of a basic computer operation such as "increment by one" that is executed by the hardware circuitry of a computer's central processing unit

toolkit application: uses the commands described in TS 31.111 [2] and TS 11.14 [8]

3.2 Abbreviations

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

APDU	Application Protocol Data Unit
API	Application Programming Interface
CAT	Card Application Toolkit
CS	Circuit Switched
DCS	Digital Cellular System
DF	Dedicated File
DTMF	Dual Tone Multiple Frequency
EF	Elementary File
ELF	Executable and Linkable Format
FID	File Identifier
GSM	Global System for Mobile communications
ME	Mobile Equipment
NAA	Network Access Application (SIM or USIM)
OTA	Over The Air
SIM	Subscriber Identity Module
SMS	Short Message Service
STK	SIM ToolKit

SW	Status Word
TAR	Toolkit Application Reference
TLV	Tag, Length, Value
TPDU	Transport Protocol Data Unit
UICC	(not an acronym)
URL	Uniform Resource Locator
USIM	Universal Subscriber Identity Module
USSD	Unstructured Supplementary Services Data

4 Description

The (U)SIM Application consists of the following:

- APDU handlers for communicating with the ME;
- File system and file access control;
- Toolkit Framework that provides services to Toolkit applications.

The present document describes the C programming language binding for the interface between the (U)SIM application and toolkit applications described in TS 42.019 [4]. This API allows application programmers using the C programming language to access functions and data described in TS 31.111 [2] and TS 11.14 [8], such that the (U)SIM-based applications and the services they implement can be developed and loaded onto ICCs. If required and supported by the underlying smart card technology, toolkit applications can be loaded or deleted remotely, after the card has been issued.

4.1 Overview

The *'C'-binding for (U)SIM API* shall provide function calls for pro-active functions and transport functions. The figure below shows the interactions between a typical toolkit application (shown in blue) and the various functional blocks of the (U)SIM (shown in orange). The C-bindings for these APIs are presented in subclause 4.2.

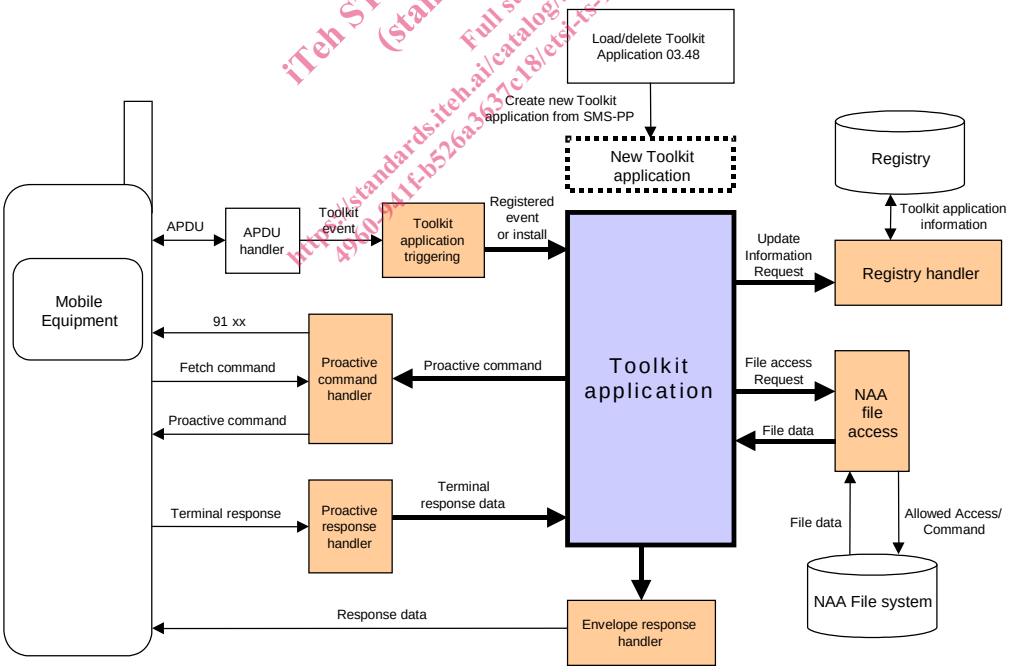

Figure 1

4.2 Design Rationale and Upward Compatibility

Some functions on the C SIM API take parameters that correspond to optional TLVs in TS 31.111 [2] and TS 11.14 [8]. If the actual parameter value passed to the function is NULL, the corresponding TLV is not passed to the ME; an example of an optional parameter is CatIconIdentifier that corresponds to the ICON IDENTIFIER TLV.

Some proactive commands have a very large number of optional TLVs, such as SETUP CALL. Therefore, this API offers two variants that address this aspect, CatSetupCall and CatSetupCallEx. The first function, CatSetupCall, takes as parameters everything that is necessary to issue a successful SETUP CALL proactive command (i.e. everything required to construct the mandatory TLVs as required by TS 31.111 [2] and TS 11.14 [8]) and also includes optional user interface TLVs (title and icon) for ease of use.

The second function, CatSetupCallEx, takes a parameter block that can be extended in future versions of the present specification. The parameter block contains members that correspond to all mandatory and optional TLVs for the SETUP CALL proactive command.

The reason for introducing the "…Ex" variants is threefold:

- Rather than extend the parameter list of a function to take a large number of optional parameters for each call, it is preferable to set up the parameters using named structure members before issuing the call to the function.
- If a future version of TS 31.111 [2] or TS 11.14 [8] extends the optional parameters for a proactive command, the corresponding parameter block can be extended to encompass these parameters without changing the function prototype.
- Any source code written for an older version of this C SIM API can be recompiled with a later version without change and will remain upwardly compatible at the source as long as the suggested coding standards are adhered to.

4.3 Application Triggering

The application-triggering portion of the SIM Toolkit Framework is responsible for the activation of toolkit applications, based on the APDU received by the card.

The ME shall not be adversely affected by the presence of applications on the (U)ICC card. For instance a syntactically correct Envelope shall not result in an error status word in case of a failure of an application. The only application as seen by the ME is the (U)SIM application. As a result, a toolkit application may return an error, but this error will not be sent to the ME.

The difference between an application and a toolkit application is that a toolkit application does not typically handle APDUs directly. It will handle higher-level messages. Furthermore the execution of a function could span over multiple APDUs, in particular, the proactive protocol commands.

All the applications that have registered interest in the event are triggered in order of their priority.

- The current context is switched to the toolkit application.
- A pending transaction is aborted.
- The current file context of the toolkit application is the MF.
- The current file context of the current selected application is unchanged.

On termination of a toolkit application execution of CatExit():

- The context switches back to the context of the current selected application, the NAA application.
- A pending toolkit application transaction is aborted.

Here after are the events that can trigger a toolkit application:

EVENT_PROFILE_DOWNLOAD

Upon reception of the Terminal Profile command by the (U)SIM, the Toolkit Framework stores the ME profile and then triggers the registered toolkit application that may want to change their registry. A toolkit application may not be able to issue a proactive command.

EVENT_MENU_SELECTION, EVENT_MENU_SELECTION_HELP_REQUEST

A toolkit application might be activated upon selection in the ME's menu by the user, or request help on this specific menu.

In order to allow the user to choose in a menu, the Toolkit Framework shall have previously issued a SET UP MENU proactive command. When a toolkit application changes a menu entry of its registry object, the Toolkit Framework shall dynamically update the menu stored in the ME during the current card session. The SIM Toolkit Framework shall use the data of the EFsume file (TS 51.011 [11] and TS 31.102 [15]) when issuing the SET UP MENU proactive command.

The positions of the toolkit application menu entries in the item list, the requested item identifiers and the associated limits (e.g. maximum length of item text string) are defined at the loading of the toolkit application.

If at least one toolkit application registers to EVENT_MENU_SELECTION_HELP_REQUEST, the SET UP MENU proactive command sent by the Toolkit Framework shall indicate to the ME that help information is available. A toolkit application registered for one or more menu entries may be triggered by the event EVENT_MENU_SELECTION_HELP_REQUEST, even if it is not registered to this event. A toolkit application registered for one or more menu entries should provide help information.

EVENT_FORMATTED_SMS_PP_ENV, EVENT_UNFORMATTED_SMS_PP_ENV,

EVENT_FORMATTED_SMS_PP_UPD, EVENT_UNFORMATTED_SMS_PP_UPD

A toolkit application can be activated upon the reception of a short message. There are two ways for a card to receive an SMS: via the Envelope SMS-PP Data Download or the UpdateRecord EFsms instruction.

The reception of the SMS by the toolkit application cannot be guaranteed for the Update Record EFsms instruction.

The received SMS may be:

- formatted according to TS 23.048 [3] or an other protocol to identify explicitly the toolkit application for which the message is sent;
- unformatted or using a toolkit application specific protocol the Toolkit Framework will pass this data to all registered toolkit applications.

EVENT_FORMATTED_SMS_PP_ENV

This event is triggered by an envelope APDU containing an SMS_DATADOWNLOAD BER TLV with an SMS_TPDU simple TLV according to TS 23.048 [3].

The Toolkit Framework shall:

- verify the TS 23.048 [3] security of the SMS TPDU;
- trigger the toolkit application registered with the corresponding TAR defined at application loading;
- take the optional Application Data posted by the triggered toolkit application if present;
- secure and send the response packet.

The toolkit application will only be triggered if the TAR is known and the security verified. Application data will also be deciphered.

EVENT_UNFORMATTED_SMS_PP_ENV

The registered toolkit applications will be triggered by this event and get the data transmitted in the APDU envelope SMS_DATADOWNLOAD.

EVENT_FORMATTED_SMS_PP_UPD

This event is triggered by Update Record EFsms with an SMS TP-UD field formatted according to TS 23.048 [3].

The Toolkit Framework shall:

- update the EFsms file with the data received, it is then up to the receiving toolkit application to change the SMS stored in the file (i.e. the toolkit application need to have access to the EFsms file);
- verify the TS 23.048 [3] security of the SMS TPDU;
- convert the Update Record EFsms in a TLV List, an EnvelopeHandler;

- trigger the toolkit application registered with the corresponding TAR defined at application loading.

EVENT_UNFORMATTED_SMS_PP_UPD

The SIM Toolkit Framework will first update the EFsms file, convert the received APDU as described above, and then trigger all the registered toolkit applications. All of them may modify the content of EFsms (i.e. the toolkit applications need to have access to the EFsms file).

EVENT_UNFORMATTED_SMS_CB

When the ME receives a new cell broadcast message, the cell broadcast page may be passed to the card using the envelope command. e.g. the application may then read the message and extract a meaningful piece of information that could be displayed to the user, for instance.

EVENT_CALL_CONTROL_BY_SIM

When the NAA is in call control mode and when the user dials a number, this number is passed to the Toolkit Framework. Only one toolkit application can handle the answer to this command: call barred, modified or accepted.

EVENT_EVENT_DOWNLOAD_MT_CALL, EVENT_EVENT_DOWNLOAD_CALL_CONNECTED,

EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED, EVENT_EVENT_DOWNLOAD_LOCATION_STATUS,

EVENT_EVENT_DOWNLOAD_USER_ACTIVITY, EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE,

EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS

The toolkit application will be triggered by the registered event download trigger, upon reception of the corresponding Envelope command. In order to allow the toolkit application to be triggered by these events, the Toolkit Framework shall have previously issued a SET UP EVENT LIST proactive command. When a toolkit application changes one or more of these requested events of its registry, the Toolkit Framework shall dynamically update the event list stored in the ME during the current card session.

EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM

Before sending an SMS MO entered by the user, the SMS is submitted to the Toolkit framework. Only one toolkit application can register to this event.

EVENT_TIMER_EXPIRATION

This event is registered when the application executes a successful Toolkit CatGetTimer(). The toolkit application can then manage this (these) timer(s), and it will be triggered at the reception of the APDU Envelope TIMER EXPIRATION. The Toolkit Framework shall reply busy to this Envelope APDU if it cannot guaranty to trigger the corresponding toolkit application.

EVENT_UNRECOGNIZED_ENVELOPE

The application registered to this event shall be triggered by the framework if the BER-TLV tag contained in the ENVELOPE APDU is not defined in the associated release of TS 31.111 [2] and TS 11.14 [8] and if no corresponding constant is defined in the list of the ToolkitConstants interface. By providing the means to transfer an arbitrary block of data, the Unrecognized Envelope Event will allow a toolkit application to handle the evolution of the specifications TS 31.111 [2] and TS 11.14 [8].

EVENT_STATUS_COMMAND

At reception of a STATUS APDU command, the SIM Toolkit Framework shall trigger the registered toolkit application.

A range of events is reserved for experimental and proprietary usage (from -128 to -1). As the definition of these events is not standardized, the use of these events may make the toolkit application behave differently on different platforms.

The toolkit application shall be triggered for the registered events upon reception, and shall be able to access to the data associated to the event using OpenEnvelope() or the low-level functions.

The order of triggering the toolkit application shall follow the priority level of each toolkit application defined at its loading. If several toolkit applications have the same priority level, the last loaded toolkit application takes precedence.