|  |  |
| --- | --- |
| 3GPP TS 31.130 V18.1.0 (2024-06) | |
| Technical Specification | |
| 3rd Generation Partnership Project;  Technical Specification Group Core Network and Terminals;  (U)SIM Application Programming Interface (API);  (U)SIM API for Java™ Card;  (Release 18) | |
|  | |
|  |  |
|  | |
| The present document has been developed within the 3rd Generation Partnership Project (3GPP TM) and may be further elaborated for the purposes of 3GPP. The present document has not been subject to any approval process by the 3GPPOrganizational Partners and shall not be implemented. This Specification is provided for future development work within 3GPPonly. The Organizational Partners accept no liability for any use of this Specification. Specifications and Reports for implementation of the 3GPP TM system should be obtained via the 3GPP Organizational Partners' Publications Offices. | |

|  |
| --- |
|  |
| ***3GPP***  Postal address  3GPP support office address  650 Route des Lucioles - Sophia Antipolis  Valbonne - FRANCE  Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16  Internet  http://www.3gpp.org |
| ***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.  © 2024, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).  All rights reserved.  UMTS™ is a Trade Mark of ETSI registered for the benefit of its members  3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners LTE™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners  GSM® and the GSM logo are registered and owned by the GSM Association |

Contents

Foreword 4

1 Scope 5

2 References 5

3 Definitions and abbreviations 6

3.1 Definitions 6

3.2 Abbreviations 6

4 Description 6

4.0 Overview 6

4.1 (U)SIM Java Card™ Architecture 7

5 File Access API 7

6 (U)SAT Framework 7

6.0 Overview 7

6.1 Applet triggering 8

6.1.1 Exception Handling 8

6.2 Definition of Events 8

6.3 Registration 12

6.4 Proactive command handling 13

6.5 Envelope response handling 13

6.6 System Handler management 13

6.7 (U)SAT Framework behaviour 14

7 UICC toolkit applet 14

8 Geo Location API 14

9 SUCI API 14

10 GBA\_U API 15

10.0 Overview 15

10.1 Access control 15

Annex A (normative): Java Card™ (U)SIM API 16

Annex B (normative): Java Card™ (U)SIM API identifiers 17

Annex D (normative): USIM API jar files 20

Annex E (informative): Change History 21

# 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.

In the present document, modal verbs have the following meanings:

**shall** indicates a mandatory requirement to do something

**shall not** indicates an interdiction (prohibition) to do something

The constructions "shall" and "shall not" are confined to the context of normative provisions, and do not appear in Technical Reports.

The constructions "must" and "must not" are not used as substitutes for "shall" and "shall not". Their use is avoided insofar as possible, and they are not used in a normative context except in a direct citation from an external, referenced, non-3GPP document, or so as to maintain continuity of style when extending or modifying the provisions of such a referenced document.

**should** indicates a recommendation to do something

**should not** indicates a recommendation not to do something

**may** indicates permission to do something

**need not** indicates permission not to do something

The construction "may not" is ambiguous and is not used in normative elements. The unambiguous constructions "might not" or "shall not" are used instead, depending upon the meaning intended.

**can** indicates that something is possible

**cannot** indicates that something is impossible

The constructions "can" and "cannot" are not substitutes for "may" and "need not".

**will** indicates that something is certain or expected to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

**will not** indicates that something is certain or expected not to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

**might** indicates a likelihood that something will happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

**might not** indicates a likelihood that something will not happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

In addition:

**is** (or any other verb in the indicative mood) indicates a statement of fact

**is not** (or any other negative verb in the indicative mood) indicates a statement of fact

The constructions "is" and "is not" do not indicate requirements.

# 1 Scope

The present document defines the (U)SIM Application Programming Interface extending the "UICC API for Java Card™" [2].

This API allows to develop a (U)SAT application running together with a (U)SIM application and using 3GPP network features.

The present document includes information applicable to 3GPP network operators, service providers, server – (U)SIM – database manufacturers.

# 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] ETSI TS 101 220: "Integrated Circuit Cards (ICC); ETSI numbering system for telecommunication; Application providers (AID)".

[2] ETSI TS 102 241 V18.0.0: "UICC API for Java Card™"

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

[4] 3GPP TS 51.011 Release 4: "Specification of the Subscriber Identity Module- Mobile Equipment (SIM – ME) interface".

[5] 3GPP TS 23.041: "Technical realization of Cell Broadcast Service (CBS)".

[6] 3GPP TS 31.101: "UICC-terminal interface; Physical and logical characteristics".

[7] 3GPP TS 31.111: "USIM Application Toolkit (USAT)".

[8] 3GPP TS 51.014 Release 4: "Specification of the SIM Application Toolkit for the Subscriber Identity Module – Mobile Equipment (SIM – ME) interface".

[9] 3GPP TS 31.115: "Secured packet structure for the (U)SIM Toolkit applications".

[10] 3GPP TS 23.040: "Technical realization of the Short Message Service (SMS)".

[11] Void.

[12] Void.

[13] Void.

[14] 3GPP TS 23.032: "Universal Geographical Area Description (GAD)".

[15] IEC 61162-1: "Maritime navigation and radio communication equipment and systems – Digital interfaces".

[16] 3GPP TS 23.003: "Numbering, Addressing and Identification".

[17] 3GPP TS 31.103: "Characteristics of the IP Multimedia Services Identity Module (ISIM) application".

[18] 3GPP TS 33.220: "Generic Authentication Architecture (GAA);Generic Bootstrapping

# 3 Definitions and abbreviations

## 3.1 Definitions

For the purposes of the present document, the terms and definitions defined in ETSI TS 102 241 [2] apply.

**(U)SAT Framework :** (U)SAT extension of the CAT Runtime Environment.

## 3.2 Abbreviations

For the purposes of the present document, the abbreviations defined in ETSI TS 102 241 [2] apply.

# 4 Description

## 4.0 Overview

This API is an extension to the ETSI TS 102 241 [2] "UICC API for Java Card™" and requires the implementation of this specification.

The classes and interfaces described in this specification inherit functionality from the classes and interfaces specified in the "UICC API for Java Card™".

The (U)SAT Framework described in this specification is an extension of the CAT Runtime Environment defined in ETSI TS 102 241 [2].

## 4.1 (U)SIM Java Card™ Architecture

The overall architecture of the (U)SIM API is based on the "UICC API for Java Card™" defined in ETSI TS 102 241 [2].



Figure 1: (U)SIM Java Card™ Architecture

# 5 File Access API

The (U)SIM file access API consists of the package *uicc.usim.access*. This package defines additional constants to those defined in the *uicc.access* package from ETSI TS 102 241 [2]. The access to the file system, defined in TS 51.011 [4] and TS 31.102 [3], is the one specified in ETSI TS 102 241 [2] via the UICC *FileView* Interface. When selecting a cyclic file the current record number is defined, this applies also to files located under DFGSM.

# 6 (U)SAT Framework

## 6.0 Overview

The (U)SIM toolkit API consists of the *uicc.usim.toolkit* package for toolkit features defined in TS 31.111 [7] and TS 51.014 [8], and is based on the *uicc.toolkit* package defined in ETSI TS 102 241 [2]*.*

## 6.1 Applet triggering

See ETSI TS 102 241 [2].

### 6.1.1 Exception Handling

The following clause describes the handling of exceptions by the (U)SAT Framework in addition to the behaviour defined in ETSI TS 102 241 [2] for the CAT Runtime Environment.

If an Applet triggered by EVENT\_FORMATTED\_SMS\_PP\_ENV event throws an ISOException with the reason code (0x6FXX), it shall be sent to the terminal.

Other Exceptions shall not be propagated to the terminal.

## 6.2 Definition of Events

The following events can trigger a Toolkit Applet in addition to the events defined in ETSI TS 102 241 [2], all short values are reserved in ETSI TS 102 241 [2]:

Table 1: (U)SAT event list

|  |  |
| --- | --- |
| Event Name | Reserved short value |
| EVENT\_FORMATTED\_SMS\_PP\_ENV | 2 |
| EVENT\_FORMATTED\_SMS\_PP\_UPD | 3 |
| EVENT\_UNFORMATTED\_SMS\_PP\_ENV | 4 |
| EVENT\_UNFORMATTED\_SMS\_PP\_UPD | 5 |
| EVENT\_UNFORMATTED\_SMS\_CB | 6 |
| EVENT\_MO\_SHORT\_MESSAGE\_CONTROL\_BY\_NAA | 10 |
| EVENT\_FORMATTED\_SMS\_CB | 24 |
| EVENT\_EVENT\_DOWNLOAD\_IWLAN\_ACCESS\_STATUS | 30 |
| EVENT\_EVENT\_DOWNLOAD\_NETWORK\_REJECTION | 31 |
| EVENT\_EVENT\_DOWNLOAD\_CSG\_CELL\_SELECTION | 33 |
| EVENT\_EVENT\_DOWNLOAD\_DATA\_CONNECTION\_STATUS\_CHANGE | 37 |
| Reserved for 3GPP (for future usage) | 113 |
| Reserved for 3GPP (for future usage) | 114 |
| Reserved for 3GPP (for future usage) | 115 |
| Reserved for 3GPP (for future usage) | 116 |
| EVENT\_EVENT\_DOWNLOAD\_SLICES\_STATUS\_CHANGE | 117 |
| EVENT\_EVENT\_DOWNLOAD\_CAG\_CELL\_SELECTION | 118 |
| EVENT\_EVENT\_DOWNLOAD\_IMS\_REGISTRATION | 119 |
| EVENT\_EVENT\_DOWNLOAD\_INCOMING\_IMS\_DATA | 120 |
| EVENT\_FORMATTED\_USSD | 121 |
| EVENT\_UNFORMATTED\_USSD | 122 |

*EVENT\_FORMATTED\_SMS\_PP\_ENV, EVENT\_UNFORMATTED\_SMS\_PP\_ENV,  
EVENT\_FORMATTED\_SMS\_PP\_UPD, EVENT\_UNFORMATTED\_SMS\_PP\_UPD*

There are two ways for a card to receive a Short Message Point to Point: via an ENVELOPE(SMS-PP Download) APDU as defined in TS 31.111 [7] and TS 51.014 [8] or an UPDATE RECORD EFSMS APDU as defined in TS 31.102 [3] and TS 51.011 [4]. The EFSMS can beeither located under the DFTelecom or under any ADF as defined in TS 31.102 [3] and TS 51.011 [4].

The received Short Message may be:

- formatted according to TS 31.115 [9] or an other protocol to identify explicitly the toolkit applet for which the message is sent;

- unformatted (e.g. a toolkit applet specific protocol ) then the (U)SAT Framework will pass this data to all registered toolkit applets.

When the Short Message is received as Concatenated Short Messages as defined in TS 23.040 [10], it is the responsibility of the (U)SAT Framework to link single Short Messages together to re – assemble the original message before any further processing. The original Short Message shall be placed in one SMS TPDU TLV (with TP-UDL field coded on one octet) included in the *USATEnvelopeHandler*. The concatenation control headers used to re-assemble the short messages in the correct order shall not be present in the SMS TPDU. The TP-elements of the SMS TPDU and the Address (TS – Service-Centre-Address) shall correspond to the ones in the last received Short Message (independently of the Sequence number of Information-Element-Data).

The minimum requirement for the (U)SAT Framework is to process a concatenated short message with the following properties:

- the Information Element Identifier is equal to the 8-bit reference number.

- it contains uncompressed 8 bit data or uncompressed UCS2 data.

EVENT\_FORMATTED\_SMS\_PP\_ENV

Upon reception of a TS 31.115 [9] formatted Short Message Point to Point (Single or Concatenated) via an ENVELOPE, the (U)SAT Framework shall:

- verify the security of the Short Message as per TS 31.115 [9];

- trigger the toolkit applet registered with the corresponding TAR;

- take the optional Application Data posted by the triggered toolkit applet if present;

- secure and send the response packet using SMS-DELIVER-REPORT or SMS-SUBMIT.

When the toolkit applet is triggered, data shall be provided deciphered.

EVENT\_UNFORMATTED\_SMS\_PP\_ENV

Upon reception of an unformatted Short Message Point to Point (Single or Concatenated) via an ENVELOPE, the (U)SAT Framework shall trigger all the Toolkit Applets registered to this event.

NOTE 1: As a consequence of the *EnvelopeResponseHandler* availability rules specified in clause 6.6, only the first triggered toolkit applet is guaranteed to be able to send back a response.

EVENT\_FORMATTED\_SMS\_PP\_UPD

Upon reception of a TS 31.115 [9] formatted Short Message Point to Point (Single or Concatenated) via an UPDATE RECORD EFSMS, the (U)SAT Framework shall:

- update the EFSMS file with the data received, it is then up to the receiving toolkit applet to change the SMS stored in the file (i.e. the toolkit applet need to have access to the EFSMS file)

- verify the security of the Short Message as per TS 31.115 [9];

- convert the UPDATE RECORD EFSMS APDU into a COMPREHENSION TLV List;

- trigger the toolkit applet registered with the corresponding TAR;

When the toolkit applet is triggered, data shall be provided deciphered.

The *USATEnvelopeHandler* provided to the applet shall:

- return *BTAG\_SMS\_PP\_DOWNLOAD* to the *getTag()* method call**;**

- return the Comprehension TLV list length to the *getLength()* method call;

The *USATEnvelopeHandler* provided to the applet shall contain the following COMPREHENSION TLVs:

- Device Identities TLV

The Device Identities Comprehension TLV is used to store the information about the absolute record number in the EFSMS file and the value of the EFSMS record status byte, and is formatted as defined below:

|  |
| --- |
| Device identities Comprehension TLV |
| Device Identities tag |
| length = 02 |
| Absolute Record Number |
| Record Status |

With the absolute record number the toolkit applet can update EFSMS in absolute mode to change the received SMS (e.g. in a readable text).

For Concatenated Short Message the Absolute Record Number and the Record Status will correspond to the last UPDATE RECORD EFSMS APDU received.

- Address TLV

The value is the TS-Service-Centre-Address (RP-OA) of the last UPDATE RECORD EFSMS APDU.

- SMS TPDU TLV

The value is the SMS TPDU provided deciphered and reassembled, if needed

- AID TLV

The AID comprehension TLV is present only if the EFSMS file updated is under an ADF. The value is the AID of the ADF as defined TS 31.111 [7].

The order of the TLVs given in the *USATEnvelopeHandler* is not specified,

NOTE 2: To get each COMPREHENSION TLV, it is recommended that the applet uses the *ViewHandler.findTLV()* methods

*EVENT\_UNFORMATTED\_SMS\_PP\_UPD*

Upon reception of an unformatted Short Message Point to Point (Single or Concatenated) via UPDATE RECORD EFSMS APDU, the (U)SAT Framework shall :

- update the EFSMS file with the data received;

- convert the UPDATE RECORD EFSMS APDU data into a COMPREHENSION TLV List (as described for *EVENT\_FORMATTED\_SMS\_PP\_UPD*);

- trigger all the Toolkit Applets registered to this event.

The content of EFSMS may have been modified by a previously triggered Toolkit Applet..

*EVENT\_FORMATTED\_SMS\_CB, EVENT\_UNFORMATTED\_SMS\_CB*

The received Cell Broadcast Message, via an ENVELOPE (CELL BROADCAST DOWNLOAD) APDU as defined in TS 31.111 [7] and TS 51.014 [8] and, can be either:

- formatted according to TS 31.115 [9] or an other protocol to identify explicitly the toolkit applet for which the message is sent;

- unformatted ( e.g. using a toolkit applet specific protocol ), then the (U)SAT Framework will pass this data to all registered toolkit applets.

When the Cell Broadcast Message is received as multiple pages as defined in TS 23.041 [5], it is the responsibility of the (U)SAT Framework to link single pages together to re‑assemble the original message before any further processing. The original Cell Broadcast message shall be placed in one Cell Broadcast page TLV included in the *USATEnvelopeHandler*. The message parameters shall correspond to the ones in the last received Cell Broadcast page (independently of the Page Parameter).

EVENT\_FORMATTED\_SMS\_CB

Upon reception of a TS 31.115 [9] formatted Cell Broadcast message, the (U)SAT Framework shall:

- verify the security of the Cell Broadcast message as per TS 31.115 [9];

- trigger the toolkit applet registered with the corresponding TAR;

When the toolkit applet is triggered, data shall be provided deciphered.

EVENT\_UNFORMATTED\_SMS\_CB

Upon reception of an unformatted Cell Broadcast message, the (U)SAT Framework shall trigger all the Toolkit Applets registered to this event.

*EVENT\_MO\_SHORT\_MESSAGE\_CONTROL\_BY\_NAA*

Upon reception of an ENVELOPE (MO SHORT MESSAGE CONTROL defined in TS 51.014 [8] and TS 31.111 [7]) APDU as defined in TS 31.101 [6] and TS 51.011 [4] the (U)SAT Framework shall trigger the Toolkit Applet registered to this event. The (U)SAT Framework shall not allow more than one Toolkit Applet to be registered to this event at a time(e.g. if a Toolkit Applet is registered to this event but not in selectable state the (U)SAT Framework shall not allow another Toolkit Applet to register to this event).

*EVENT\_FORMATTED\_USSD, EVENT\_UNFORMATTED\_USSD*

The received USSD String, via an ENVELOPE (USSD Data Download) APDU as defined in TS 31.111 [7], may be:

- formatted according to TS 31.115 [9] or an other protocol to identify explicitly the toolkit applet for which the message is sent;

- unformatted (e.g. a toolkit applet specific protocol) then the (U)SAT Framework will pass this data to all registered toolkit applets.

When the USSD Message is received as concatenated as defined in TS 31.115 [9], it is the responsibility of the (U)SAT Framework to link single USSD Messages together to re‑assemble the original message before any further processing. The original USSD message shall be placed in one USSD String TLV included in the *USATEnvelopeHandler*. The USSD String parameters (DCS, PFI, CCF) shall correspond to the ones in the last received USSD String (independently of the CCF Sequence number).

EVENT\_FORMATTED\_USSD

Upon reception of a TS 31.115 [9] formatted USSD Message via an ENVELOPE, the (U)SAT Framework shall:

- verify the security of the USSD Message as per TS 31.115 [9];

- trigger the toolkit applet registered with the corresponding TAR;

- take the optional Application Data posted by the triggered toolkit applet if present;

- secure and send the response packet.

When the toolkit applet is triggered, data shall be provided deciphered.

EVENT\_UNFORMATTED\_USSD

Upon reception of an unformatted USSD String via an ENVELOPE, the (U)SAT Framework shall trigger all the Toolkit Applets registered to this event.

NOTE 3: As a consequence of the *EnvelopeResponseHandler* availability rules specified in clause 6.6, only the first triggered toolkit applet is guaranteed to be able to send back a response.

*EVENT\_EVENT\_DOWNLOAD\_IWLAN\_ACCESS\_STATUS*

*EVENT\_EVENT\_DOWNLOAD\_NETWORK\_REJECTION*

*EVENT\_EVENT\_DOWNLOAD\_CSG\_CELL\_SELECTION*

*EVENT\_EVENT\_DOWNLOAD\_IMS\_REGISTRATION*

*EVENT\_EVENT\_DOWNLOAD\_INCOMING\_IMS\_DATA*

*EVENT\_EVENT\_DOWNLOAD\_DATA\_CONNECTION\_STATUS\_CHANGE*

*EVENT\_EVENT\_DOWNLOAD\_CAG\_CELL\_SELECTION*

*EVENT\_EVENT\_DOWNLOAD\_SLICES\_STATUS\_CHANGE*

Upon reception of an ENVELOPE (Event Download) APDU command as defined in TS 31.111 [7] the (U)SAT Framework shall trigger all the Toolkit applets registered to the corresponding event.

The following events defined in TS 31.111 [7] shall be raised upon reception of the corresponding APDU defined in either TS 51.011 [4] or TS 31.101 [6].

EVENT\_PROFILE\_DOWNLOAD

EVENT\_MENU\_SELECTION, EVENT\_MENU\_SELECTION\_HELP\_REQUEST

EVENT\_CALL\_CONTROL\_BY\_NAA

EVENT\_TIMER\_EXPIRATION

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

EVENT\_STATUS\_COMMAND

EVENT\_EVENT\_DOWNLOAD\_LANGUAGE\_SELECTION

EVENT\_EVENT\_DOWNLOAD\_BROWSER\_TERMINATION

EVENT\_EVENT\_DOWNLOAD\_DATA\_AVAILABLE

EVENT\_EVENT\_DOWNLOAD\_CHANNEL\_STATUS

EVENT\_EVENT\_DOWNLOAD\_ACCESS\_TECHNOLOGY\_CHANGE

EVENT\_EVENT\_DOWNLOAD\_DISPLAY\_PARAMETER\_CHANGED

EVENT\_EVENT\_DOWNLOAD\_LOCAL\_CONNECTION

EVENT\_EVENT\_DOWNLOAD\_NETWORK\_SEARCH\_MODE\_CHANGE

EVENT\_EVENT\_DOWNLOAD\_BROWSING\_STATUS

EVENT\_PROACTIVE\_HANDLER\_AVAILABLE

EVENT\_EXTERNAL\_FILE\_UPDATE

EVENT\_FIRST\_COMMAND\_AFTER\_ATR

EVENT\_UNRECOGNIZED\_ENVELOPE

## 6.3 Registration

A Toolkit Applet shall register to events described in 6.2 as defined in ETSI TS 102 241 [2].

Constants for these events are available in *uicc.usim.toolkit.ToolkitConstants* interface in Annex A.

The *uicc.toolkit.ToolkitException* TAR\_NOT\_DEFINED shall be thrown if a Toolkit Applet has no TAR defined and registers to events: EVENT\_FORMATTED\_SMS\_PP\_ENV, EVENT\_FORMATTED\_SMS\_PP\_UPD, EVENT\_FORMATTED\_SMS\_CB, EVENT\_FORMATTED\_USSD.

The *uicc.toolkit.ToolkitException*.EVENT\_ALREADY\_REGISTERED shall be thrown if there is another Toolkit Applet already registered to *EVENT\_MO\_SHORT\_MESSAGE\_CONTROL\_BY\_NAA.*

## 6.4 Proactive command handling

There is no extension of the CAT Runtime Environment by the (U)SAT Framework for proactive command handling.

## 6.5 Envelope response handling

For the events defined in the present document, the following rules apply:

A Toolkit Applet can post a response by using the *post()* methodor the *postAsBERTLV()* method defined in ETSI TS 102 241 [2]. The (U)SAT Framework shall return the Status Word as defined in TS 31.111 [7] and in TS 51.014 [8] depending on the current NAA.

Case of EVENT\_MO\_SHORT\_MESSAGE\_CONTROL\_BY\_NAA:

- The rules defined for *EVENT\_CALL\_CONTROL\_BY\_NAA* in ETSI TS 102 241 [2] apply.

Case of EVENT\_UNFORMATTED\_SMS\_PP\_ENV:

- See ETSI TS 102 241 [2].

Case of EVENT\_FORMATTED\_SMS\_PP\_ENV:

- When the *post()* or the *postAsBERTLV()* method is invoked, the (U)SAT Framework shall, according to bit 6 of the second octet of the SPI defined in TS 31.115 [9], build a SMS-DELIVER-REPORT or a SMS-SUBMIT.

In case of a SMS-DELIVER-REPORT and if the post response is too large to be contained in a SMS-DELIVER-REPORT, the (U)SAT Framework shall issue Response Packets as defined in TS 31.115 [9].

In case of a SMS-DELIVER-REPORT, the (U)SAT Framework shall return the Status Word for RP-ACK or RP-ERROR as defined in TS 31.111 [7] and in TS 51.014 [8] depending on the current NAA.

In case of SMS-SUBMIT the boolean value method parameter shall be ignored by the (U)SAT Framework. If the SMS-SUBMIT is to be used, the (U)SAT Framework shall build and issue a Send Short Message proactive command as defined in TS 31.111 [7] and in TS 51.014 [8] depending on the current NAA .

Case of EVENT\_FORMATTED\_USSD:

- When the *post()* or the *postAsBERTLV()* method is invoked, the (U)SAT Framework shall build a USSD String to be sent back in the Return Result Component contained in the subsequent Facility message. In that case the (U)SAT Framework shall return the Status Word as defined in TS 31.111 [7].

Case of EVENT\_UNFORMATTED\_USSD:

- See ETSI TS 102 241 [2].

## 6.6 System Handler management

For the handler management of the *ProactiveHandler*, the *ProactiveResponseHandler*, the *EnvelopeHandler* and the *EnvelopeResponseHandler*, the rules defined in ETSI TS 102 241 [2] apply.

*USATEnvelopeHandler:*

The single system instance of the *USATEnvelopeHandler* and the single system instance of the *EnvelopeHandler* are two distinct objects instances.

- When available the *USATEnvelopeHandler* shall remain available and its content shall remain unchanged from the invocation to the termination of the *processToolkit()* method.

- The TLV List provided in the *USATEnvelopeHandler* are the same as in the *EnvelopeHandler*.

- The handler availability of the *USATEnvelopeHandler* is the same handler availability as the *EnvelopeHandler* including all the events defined in ETSI TS 102 241 [2].

The following table describes the minimum availability of the handlers for all the events at the invocation of the *processToolkit()* method of the Toolkit Applet. The rules described in this table apply in addition to the rules described in "UICC API for Java Card™"

Table 2: Handler availability for each event

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| EVENT\_ | | Reply busy allowed | EnvelopeHandler /  USATEnvelopeHandler | EnvelopeResponseHandler | Nb of triggered / registrered Applet |
| \_FORMATTED\_SMS\_PP\_ENV | | Y  (see Note 1) | Y | Y | 1 / n (per TAR) |
| \_FORMATTED\_SMS\_PP\_UPD | | N | Y | N | 1 / n (per TAR) |
| \_UNFORMATTED\_SMS\_PP\_ENV | | Y | Y | Y | n / n |
| \_UNFORMATTED\_SMS\_PP\_UPD | | N | Y | N | n / n |
| \_FORMATTED*\_*SMS\_CB | | Y | Y | N | 1/n (per TAR) |
| \_UNFORMATTED\_SMS\_CB | | Y | Y | N | n / n |
| \_MO\_SHORT\_MESSAGE\_CONTROL\_BY\_NAA | | N | Y | Y | 1 / 1 |
| \_FORMATTED*\_*USSD | | Y | Y | Y | 1 / n (per TAR) |
| UNFORMATTED*\_*USSD | | Y | Y | Y | n / n |
| EVENT\_DOWNLOAD | |  |  |  |  |
| \_IWLAN\_ACCESS\_STATUS | | Y | Y | N | n/n |
| \_NETWORK\_REJECTION | | Y | Y | N | n/n |
| \_IMS\_REGISTRATION | | Y | Y | N | n/n |
| \_INCOMING\_IMS\_DATA | | Y | Y | N | n/n |
| \_DATA\_CONNECTION\_STATUS\_CHANGE | | Y | Y | N | n/n |
| \_CAG\_CELL\_SELECTION | | Y | Y | N | n/n |
| \_SLICES\_STATUS\_CHANGE | Y | Y | N | n/n |
| NOTE 1: The framework may reply busy and not trigger the toolkit applet if e.g. a PoR using SMS SUBMIT is required in the incoming message and a proactive session is ongoing. | | | | | |

## 6.7 (U)SAT Framework behaviour

The (U)SAT Framework is a (U)SAT extension of the CAT Runtime Environment as defined in ETSI TS 102 241 [2]. In addition, the (U)SAT Framework shall consider the EVENT\_EVENT\_DOWNLOAD\_\* defined in this specification when issuing the SET UP EVENT LIST system proactive command.

# 7 UICC toolkit applet

See ETSI TS 102 241 [2].

# 8 Geo Location API

The Geo Location API consists of the package *uicc.usim.geolocation*. This package defines services to allow an Applet to perform a geographical location operation, depending of the ME capabilities. When a geographical location operation is requested, the API will follow a defined way to choose either "Geographical Location Request" toolkit command or "Provide Local Information" toolkit command as defined in TS 31.111 [7] to determine the location information. The result is formatted using GAD shapes as defined in TS 23.032 [14] or in the format of NMEA sentences defined in IEC 61162-1 [15].

# 9 SUCI API

The SUCI API consists of the package *uicc.usim.suci*. This package defines services to allow an Applet to perform a SUCI computation upon reception of terminal request.

If an applet has registered an object implementing the interface *SUCICalculator* to the USIM application, then when the ME sends a GET IDENTITY APDU Command in SUCI context to this USIM application, the (U)SAT framework shall invoke the *getSUCI* method in order to retrieve the SUCI to be returned to the ME. Only one object can be registered per USIM application. The reference to the object is needed to dereference the object. If no object is registered, the USIM shall return the SUCI computed by its own means, according to the USIM configuration.

When SUPI type is IMSI, and Identity Context is SUCI 5G NSWO, the (U)SAT framework shall convert the SUCI returned by *getSUCI* in SUCI NAI format (see TS 23.003 [16]).

If an exception is raised, the (U)SAT framework behaviour is implementation specific.

# 10 GBA\_U API

## 10.0 Overview

The GBA\_U API consists of the package *uicc.usim.gba\_u*. This package defines services to allow an applet to perform cryptographic operations (i.e. encryption or signature) using Ks\_int\_NAF, as defined in TS 33.220 [18], which is remained on UICC and is derived during GBA\_U procedures as defined in TS 31.102 [3] for USIM or in TS 31.103 [17] for ISIM.

During the initialization of cryptographics objects (i.e. *GBAUCipher*, *GBAUSignature* classes) using *init()* methods:

- the application (i.e. USIM, ISIM) hosting the Ks\_int\_NAF is specified through *adfAID*, *adfAIDOff* and *adfAIDLen* parameters

- the NAF\_ID is specified through *nafID*, *nafIDOff* and *nafIDLen* parameters.

- the framework shall check whether the applet is defined with corresponding NAF\_ID for requested application ADF AID (i.e. USIM, ISIM) as defined in clause 10.1,

If the applet is authorized to use the Ks\_int\_NAF, then the framework shall check additionnally:

- the GBA service is available in requested application (for USIM as defined in TS 31.102 [3], i.e. Service n°68 is "available" in UST or for ISIM as defined in TS 31.103 [17], i.e. Service n°2 is "available" in IST)

- the GBA\_U bootstrap procedure was executed as defined in TS 31.102 [3] or TS 31.103 [17]

- the GBA\_U NAF derivation was executed as defined in TS 31.102 [3] or TS 31.103 [17]

- the NAF\_ID associated to the calling applet is included in EFGBANL

If all conditions are satisfied the cryptographics objects are then initialized and returned, otherwise the framework shall raise an exception with associated reason defined in *GBAUException* or *javacard.security.CryptoException* classes.

## 10.1 Access control

On the initialization of cryptographics objects (i.e. *init()* method from *GBAUCipher*, *GBAUSignature* classes) the framework shall check:

- the *adfAID*, *adfAIDOff* and *adfAIDLen* parameters refer to an existing USIM or an existing ISIM

- the EFAC\_GBAUAPI as defined in TS 31.102 [3] or TS 31.103 [17] is present in the corresponding ADF

- at least one record of the EFAC\_GBAUAPI matches with:

- the caller applet's AID of the AID value and

- the *nafID*, *nafIDOff* and *nafIDLen* parameters of the NAF\_ID value

If all conditions are satisfied the cryptographics objects are then initialized and returned, otherwise the framework shall raise a *GBAUException* exception with reason code GBA\_U\_UNALLOWED\_ACCESS.

Annex A (normative):  
Java Card™ (U)SIM API

The attached files "31130\_Annex\_A\_Java.zip", and "31130\_Annex\_A\_HTML.zip" contains source files and html documentation for the Java Card™ (U)SIM API.

Annex B (normative):  
Java Card™ (U)SIM API identifiers

The attached file "31130\_Annex\_B\_Export\_files.zip" contains the export files for the uicc.usim.\* package.

NOTE: Since version 18.1.0 the export files are delivered in export format version 2.3.Annex C (normative):  
(U)SIM API package version management

The following table describes the relationship between each TS 31.130 specification version and its packages AID and Major, Minor versions defined in the export files.

|  |  |  |
| --- | --- | --- |
| uicc.usim.access package | | |
| TS 31.130 | Major, Minor | **AID** |
|  | 1.0 | A000000087 1005 FFFF FFFF 89 13 100000 |
| 7.7.1 | 1.1 |
| 8.3.0 | 1.2 |
| 9.1.0 | 1.3 |
| 12.0.0 | 1.4 |
| 17.0.0 | 1.5 |
| 17.1.0 | 1.6 |
| 17.3.0 | 1.7 |
| 17.4.0 | 1.7 |
| 18.1.0 | 1.8 |

|  |  |  |
| --- | --- | --- |
| uicc.usim.toolkit package | | |
| TS 31.130 | Major, Minor | **AID** |
|  | 1.0 | A000000087 1005 FFFF FFFF 89 13 200000 |
| 7.1.0 | 1.1 |
| 7.2.1 | 1.2 |
| 7.7.1 | 1.3 |
| 7.9.0 | 1.4 |
| 8.3.0 | 1.5 |
| 9.1.0 | 1.6 |
| 9.4.0 | 1.7 |
| 10.4.0 | 1.8 |
| 14.2.0 | 1.9 |
| 17.0.0 | 1.10 |
| 17.1.0 | 1.11 |
| 17.4.0 | 1.12 |
| 18.1.0 | 1.13 |

|  |  |  |
| --- | --- | --- |
| uicc.usim. geolocation package | | |
| TS 31.130 | Major, Minor | **AID** |
|  | 1.0 | A000000087 1005 FFFF FFFF 89 13 300000 |
| 13.1.0 | 2.0 |

|  |  |  |
| --- | --- | --- |
| uicc.usim.suci package | | |
| TS 31.130 | Major, Minor | **AID** |
| 15.1.0 | 1.0 | A000000087 1005 FFFF FFFF 89 13 400000 |

|  |  |  |
| --- | --- | --- |
| uicc.usim.gba\_u package | | |
| TS 31.130 | Major, Minor | **AID** |
| 18.1.0 | 1.0 | A000000087 1005 FFFF FFFF 89 13 500000 |

The package AID coding is defined in ETSI TS 101 220 [1]. The (U)SIM API packages' AID are not modified by changes to Major or Minor Version.

The Major Version shall be incremented if a change to the specification introduces byte code incompatibility with the previous version.

The Minor Version shall be incremented if a change to the specification does not introduce byte code incompatibility with the previous version.

For a table describing the versioning of a package, a line is introduced only upon changes of Major or Minor version of its package.

The package *uicc.usim.access* contains only constants, therefore it may not be loaded on the UICC.

Annex D (normative):  
USIM API jar files

The attached files "31130\_Annex\_D.jar", contains class files for the Java Card™ (U)SIM API.

Annex E (informative):  
Change History

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Change history** | | | | | | | |
| **Date** | **Meeting** | **TDoc** | **CR** | **Rev** | **Cat** | **Subject/Comment** | **New version** |
|  | TP-27 |  |  |  |  | Generation of Version 7.0.0 based on version 6.2.0 | 7.0.0 |
|  | TP-27 | TP-050023 | 009 |  |  | Allow passing of specified status words through the (U)SAT Framework | 7.0.0 |
|  | CT-28 | CP-050139 | 011 |  |  | Allign paragraph numbering between TS 31.130 and ETSI TS 102 241 | 7.1.0 |
|  | CT-28 | CP-050139 | 013 |  |  | Delete version and author info from the Java source code | 7.1.0 |
|  | CT-28 | CP-050141 | 014 |  |  | Addition of new events EVENT\_FORMATTED\_USSD and EVENT\_UNFORMATTED\_USSD | 7.1.0 |
|  | CT-29 | CP-050340 | 016 |  |  | Adding missing constant values | 7.2.0 |
|  |  |  |  |  |  | 2005-11: Adds missing attachment files and adds line to table in annex C. | 7.2.1 |
|  | CT-33 | CP-060391 | 019 | 1 |  | Correction of misnamed constant | 7.3.0 |
|  |  |  | 020 | 1 |  | Addition of missing event download I-WLAN access status |  |
|  | CT-34 | CP-060546 | 0022 | 2 |  | Clarification on getShortMessageLength() method when applied on a SMS Cell Broadcast. | 7.4.0 |
|  |  | CP-050548 | 0024 | 1 |  | Correction of the USATEnvelopeHandlerSystem method prototype |  |
|  | CT-35 | CP-070068 | 0027 | 1 |  | Correction of Annex A JAVA.zip, package uicc.usim.toolkit | 7.5.0 |
|  |  |  | 0028 | 2 |  | Update the reference to Java Card 2.2.2 |  |
|  | CT-36 | CP-070302 | 0029 | - |  | Correction of the reference to ETSI TS 102 241 | 7.6.0 |
|  |  | CP-070298 | 0029 | - |  | Correction of references to ETSI TS 102 223 and ETSI TS 102 221 |  |
|  | CT-38 | CP-070844 | 0032 | 1 |  | Introduction of new constant values for files in the USIM application | 7.7.0 |
|  |  |  |  |  |  | Annex A and B attachments provided (2008-08) | 7.7.1 |
|  | CT-42 | CP-080908 | 0034 | 2 |  | Introduction of a geographical location discovery Java Card™ API | 8.0.0 |
|  | CT-43 | CP-090196 | 0035 | 1 |  | Introduction of missing constant values for USIM files | 8.1.0 |
|  | CT-45 | CP-090719 | 0039 | 2 |  | Alignment of constants with 31.111 | 8.2.0 |
|  | CT-46 | CP-090788 | 0040 | 1 |  | References update | 8.3.0 |
|  | CT-46 | CP-091013 | 0042 | 1 |  | Support of missing event EVENT\_EVENT\_DOWNLOAD\_NETWORK\_REJECTION | 8.3.0 |
|  | CT-46 | CP-091013 | 0045 | 1 |  | Support of missing constants in USAT Terminal Profile | 8.3.0 |
|  | CT-46 | - | - | - |  | Upgrade of the specification to Rel-9 | 9.0.0 |
|  | CT-47 | CP-100185 | 0047 | 1 |  | Addition of missing constant values | 9.1.0 |
|  | CT-47 | CP-100198 | 0048 | 2 |  | Supporting operator controlled CSG list for H(e)NB | 9.1.0 |
|  | CT-47 | CP-100198 | 0049 | 2 |  | Support of CSG cell discovery and CSG selection event | 9.1.0 |
|  | -------- |  |  |  |  | Spec reissued as v9.1.1 due to a bad version number on the cover sheet | 9.1.1 |
|  | CT-50 | CP-100836 | 0046 | 1 |  | Update reference to "Java Card 3.0.1 Classic" reference | 9.2.0 |
|  | SP-51 | - | - | - |  | Upgrade of the specification to Rel-10 | 10.0.0 |
|  | CT-52 | CP-110507 | 0050 | 1 |  | Addition of events and reservation of constant values for Java API | 10.1.0 |
|  | CT-54 | CP-110905 | 0053 | - |  | Correction to TAG\_CSG\_SELECTION\_STATUS | 10.2.0 |
|  | CT-54 | CP-110905 | 0054 | - |  | Correction to constant value in TerminalProfile.java | 10.2.0 |
|  | CT-55 | CP-120154 | 0059 |  |  | Correction to TAG\_CSG\_SELECTION\_STATUS | 10.3.0 |
|  | CT-55 | CP-120154 | 0060 |  |  | Correction to constant value in TerminalProfile.java | 10.3.0 |
|  | CT-55 | CP-120154 | 0058 | 1 |  | Update the reference to ETSI TS 102 241 | 10.3.0 |
|  | -------- |  |  |  |  | Editorial version correcting the three lines above | 10.3.1 |
|  | CT-56 | CP-120393 | 0061 | 1 |  | Correct implementation of CR 0059 for TAG\_CSG\_SELECTION\_STATUS\_N | 10.4.0 |
|  | CT-56 | CP-120392 | 0062 | 1 |  | Adding a constant value in USATTerminalProfile.java for the indication of IMS support | 10.4.0 |
|  | CT-56 | CP-120393 | 0063 | 1 |  | Adding constant values in USIMConstants.java for missing file identifiers | 10.4.0 |
|  | SP-57 |  |  |  |  | Automatic upgrade to Rel-11 | 11.0.0 |
|  | SP-65 |  |  |  |  | Automatic upgrade to Rel-12 | 12.0.0 |
|  | CT-70 | CP-150827 | 0071 |  |  | Missing rule for SMS\_PP envelope response handling | 13.0.0 |
|  | CT-73 | CP-160550 | 0072 | 5 |  | Geo Location API corrections  Note 1: known problem within the change request, to be fixed at CT-74  Note 2: in the CR, the body of the CR and the attached annexes are not identical. The body of the CR contains the correct text and is implemented. | 13.1.0 |
|  | CT-74 | CP-160788 | 0073 | 1 |  | Geo Location API format alignment | 13.2.0 |
|  | CT-75 | CP-170166 | 0075 |  |  | Geolocalization API document aligment | 13.3.0 |
|  | SA-75 |  |  |  |  | Automatic upgrade to Rel-14 | 14.0.0 |
|  | CT-78 | CP-173150 | 0077 | - |  | Update of reference to ETSI TS 102 241 | 14.1.0 |
|  | CT-78 | CP-173150 | 0078 | - |  | Editorial change of Java Card reference | 14.1.0 |
|  | CT-78 | CP-173143 | 0076 | 3 |  | Corrections in Annex C | 15.0.0 |
|  |  |  |  |  |  | Added missing attachments | 15.0.1 |
| 2019-03 | CT#83 | TP-050023 | 0081 | 1 | F | SUCI Package | 15.1.0 |
| 2019-09 | CT#85 | CP-192013 | 0085 | 1 | F | Add support for ENVELOPE (EVENT DOWNLOAD - Data Connection Status Change) | 15.2.0 |
| 2019-09 | CT#85 | CP-192014 | 0083 | 1 | F | Update of reference to ETSI TS 102 241 | 15.2.0 |
| 2019-09 | CT#85 | CP-192014 | 0084 | - | F | Clarification for SUCI API | 15.2.0 |
| 2020-01 |  |  |  |  |  | 5G logo updated in a cover page as agreed in CT#86 | 15.2.1 |
| 2020-02 |  |  |  |  |  | Attachments updated | 15.2.2 |
| 2020-06 | CT#88e | CP-201148 | 0086 | 3 | F | Update the scope of 31.130 to cover 4/5G aspects | 15.3.0 |
| 2020-07 | - | - | - | - | - | Update to Rel-16 version (MCC) | 16.0.0 |
| 2022-03 | CT#95e | CP-220135 | 0089 | 1 | B | Aligment with TS 31.111 and TS 31.102 (Rel-17) | 17.0.0 |
| 2022-12 | CT#98e | CP-223084 | 0091 | 2 | F | Toolkit Event CAG Selection support (EVENT\_EVENT\_DOWNLOAD\_CAG\_CELL\_SELECTION) | 17.1.0 |
| 2023-03 | CT#99 | CP-230107 | 0093 | - | F | 3GPP reserved events allocation | 17.2.0 |
| 2023-06 | CT#100 | CP-231102 | 0094 | 1 | F | Aligment with TS 31.102 - EFNID addition | 17.3.0 |
| 2024-03 | CT#103 | CP-240139 | 0100 | - | F | TS 31.130 17.3.0 pack delivery and tag terminal profile index correction | 17.4.0 |
| 2024-03 | CT#103 |  |  |  |  | MCC update to Rel-18 | 18.0.0 |
| 2024-06 | CT#104 | Cp-241215 | 0102 | 2 | B | Aligments with TS 31.111 and TS 31.102 and new GBA\_U API introduction | 18.1.0 |