# 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://webapp.etsi.org/IPR/home.asp>).

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 Project 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 decides to modify the contents of the present document, it will be re-released by TC SCP 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 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](http://portal.etsi.org/Help/editHelp!/Howtostart/ETSIDraftingRules.aspx) (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 102 241 [9].

It describes the technical characteristics and methods for testing the UICC API for Java CardTM (ETSI TS 102 241 [9]) implemented in a UICC Platform. It specifies the following parts:

* test applicability;
* test environment description;
* tests format;
* test area reference;
* conformance requirements;
* test area files;
* test coverage;
* test procedure;
* a description of the associated testing tools that shall be used.

# 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] Sun Microsystems Java Card™ Specification: "Java Card™ 2.2.2 Application Programming Interface".

[2] Sun Microsystems Java Card™ Specification: "Java Card™ 2.2.2 Runtime Environment (JCRE) Specification".

[3] Sun Microsystems Java Card™ Specification: "Java Card™ 2.2.2 Virtual Machine Specification".

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

[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 102 223: "Smart cards; Card Application Toolkit (CAT)".

[7] ETSI TS 102 222: "Integrated Circuit Cards (ICC); Administrative commands for telecommunications applications".

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

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

[10] ETSI TS 123 040: "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Technical realization of Short Message Service (SMS) (3GPP TS 23.040)".

[11] ETSI TS 101 267: "Digital cellular telecommunications system (Phase 2+); Specification of the SIM Application Toolkit for the Subscriber Identity Module - Mobile Equipment (SIM-ME) interface (3GPP TS 11.14)".

[12] ETSI TS 131 213: "Universal Mobile Telecommunications System (UMTS); LTE; Test specification for (U)SIM; Application Programming Interface (API) for Java Card™ (3GPP TS 31.213)".

## 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 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

**Conformance Requirement Reference (CRR):** description of the expected card behaviour according to ETSI TS 102 241 [9]

**expected state:** state in which the UICC is supposed to be after the execution of the test procedure applied on the relevant initial conditions

**security parameters:** minimum security requirements defined for the applet installation process

**test area:** set of Test Cases applicable to a specific part (class method, CAT RE behaviour, etc.) of the ETSI TS 102 241 [9]

**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:** java 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)

**test toolkit applet:** applet designed to test a specific functionality of the UICC API (ETSI TS 102 241 [9])

## 3.2 Abbreviations

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

ADF Application Dedicated File

AID Application IDentifier

APDU Application Protocol Data Unit

API Application Programming Interface

ARR Access Rule Reference

ATR Answer To Reset

BER Basic Encoding Rules

CAP Converted Applet

CARU Cyclic Always Read and Update

CAT Card Application Toolkit

CAT RE Card Application Toolkit Runtime Environment

CR Comprehension Required

CRE CAT Runtime Environment

CRR Conformance Requirements Reference

CRRC Conformance Requirement Reference Context Error

CRRN Conformance Requirement Reference Normal

CRRP Conformance Requirement Reference Parameter Error

DCS Data Coding Scheme

DF Dedicated File

DIR DIRectory

EF Elementary File

FCP File Control Parameters

FID File IDentifier

GSM Global System for Mobile communications

ICCID Integrated Circuit Card IDentification

JCRE Java Card™ Runtime Environment

LCSI Life Cycle Status Information

ME Mobile Equipment

MF Master File

MSL Minimum Security Level

MT Mobile Terminated

NAA Network Access Application

NOK Not OK

PIN Personal Identification Number

PL Preferred Languages

RAPDU Response Application Protocol Data Unit

RFU Reserved for Future Use

SDK Software Development Kit

SE Security Environment

SFI Short File Identifier

STK SIM ToolKit

SW Status Word

TAR Toolkit Application Reference

TARU Transparent Always Read and Update

TLV Tag Length Value

TP Transfer layer Protocol

# 4 Test environment

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

## 4.1 Applicability

The test defined in the present document are applicable to cards implementing ETSI TS 102 241 [9] unless otherwise stated.

## 4.2 Test environment description

The general architecture for the test environment is:



NOTE: Figure 4.2 shows the test architecture required to test interoperability at both API and bytcode level. The latter is currently not included in the current specification. The diagram is for information.

Figure 4.2

## 4.3 Tests format

### 4.3.1 Test area reference

Each test area is referenced as follows:

API Testing: 'Api\_[package name]\_[class name]\_[method name]' where:

* package name:
* uicc.access package: '1'.
* uicc.toolkit package: '2'.
* uicc.system package: '3'.
* uicc.access.fileadministration: '4'.
* class/interface name:
* yyy: 3 letters for each class.

NOTE 1: See annex A for full classes acronyms list.

* method name:
* zzzz[input parameters].

NOTE 2: See annex A for full methods name acronyms list.

CRE: Cat Runtime Environment testing: 'Cre\_[Clause name]\_[Subclause name]':

* Clause name:
* xxx: 3 letters for each clause

NOTE 3: See annex A for full clause acronyms list.

* Subclause name
* yyyy: : 4 letters for each subclause

NOTE 4: See annex A for full subclause acronyms list.

#### 4.3.1.1 Conformance requirements

The conformance requirements are expressed in the following way:

* Method prototype as listed in ETSI TS 102 241 [9].
* Normal execution:
* Contains normal execution and correct parameters limit values, each referenced as a Conformance Requirement Reference Normal (CRRN).
* Parameter errors:
* Contains parameter errors and incorrect parameter limit values, each referenced as a Conformance Requirement Reference Parameter Error (CRRP).
* Context errors:
* Contains errors due to the context the method is used in, each referenced as a Conformance Requirement Reference Context Error (CRRC).

#### 4.3.1.2 Test area files

The files included in the Test Area use the following naming convention:

* Test Source: Test\_[Test Area Reference].java.
* Test Applet: [Test Area Reference]\_[Test applet number].java.
* Cap File: [Test Area Reference].cap.

The applet numbers start from '1'.

The test source shall use common interfaces defined in annex D.

The Cap File format is described in Java Card™ Virtual Machine Specification [3].

Test files can be run in any order.

All files from the same test area are located in the same subfolder.

#### 4.3.1.3 Test coverage

The table above each test procedure indicates the correspondence between the Conformance Requirements Reference (CRR) and the different test cases.

#### 4.3.1.4 Test procedure

Each test procedure contains a table to indicate the expected responses from the API and/or the APDU level as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | | | |
| Id | Description | API/CAT RE Expectation | APDU Expectation |
|  | *Test Case detailed description* | *API and/or CAT RE expected behaviour.* | *Expected response at APDU level.* |

## 4.4 Initial conditions

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

* unless otherwise stated, the file system and the files' content shall fulfil the requirements described in annex B;
* unless otherwise stated, before installing the applet(s) relevant to the current test procedure, all packages specific to other test procedures shall not be present.

When both statements apply, a test procedure is said to be in the "Default Initial Conditions" state.

## 4.5 Package name

Java packages integrating this Test Suite shall follow this naming convention:

**uicc.test.access.[Test Area Reference]:** Java Card packages containing Test Area References for the ETSI TS 102 241 [9] uicc.access package.

**uicc.test.system.[Test Area Reference]:** Java Card packages containing Test Area References for the ETSI TS 102 241 [9] uicc.system package.

**uicc.test.toolkit.[Test Area Reference]:** Java Card packages containing Test Area References for the ETSI TS 102 241 [9] uicc.toolkit package.

**uicc.test.access.fileadministration.[Test Area Reference]:** Java Card packages containing Test Area References for the ETSI TS 102 241 [9] uicc.access.fileadministration package.

**uicc.test.catre.[Test Area Reference]:** Java Card packages containing Test Area References for the ETSI TS 102 241 [9] CAT Runtime Environment.

**uicc.test.util:** for the Test util package defined in this Test Suite.

where the Test Area Reference is written in lower case.

EXAMPLE: The package *../uicc.test.access.[Test Area Reference]* creates the following directory structure *../uicc/test/access/[Test Area Reference]/Api\_1\_...\_[1..n].\*,*  where '*Api\_1\_...\_[1..n].\*'*  are the different test applets Java source files used in *[Test Area Reference]*.

## 4.6 AID Coding

The AID coding for the Test Packages, Applet classes and Applets shall be as specified in ETSI TS 101 220 [4]. In addition, the following TAR and Application Provider specific data values are defined for use within the present document:

AID coding

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte 1 | |  | | Byte 12 | | Byte 13 | | Byte 14 | | Byte 15 | | Byte 16 | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | Application Provider specific data | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | TAR | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | Specified in TS 101 220 [] | | | | | |

TAR Coding (3 bytes/ 24 bits):

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| b1 | |  | | b4 | | b5 | |  | | b8 | | b9 | |  | | b12 | | b13 | |  | | b16 | | b17 | |  | | b24 | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Applet instance number |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Applet class number |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Package number |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | RFU |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Test Part Identifier |

Applet instance number, Applet Class number, Package number:

* For package AID, package number shall start from 0 and class and instance numbers shall be 0.
* For class AID, package number is the number of the class package, class number shall start from 1 and instance shall be 0.
* For instance AID, package and class number are the number of class and package of which instance belongs, and instance number shall start from 1.

Test part Identifier (bits b1-b4):

* 0000 reserved (as TAR= '00.00.00' is reserved for Issuer Security Domain).
* 0001 API uicc.access.
* 0010 API uicc.toolkit.
* 0011 API uicc.system.
* 0100 API uicc.access.fileadministration.
* 0101 CAT RE.
* 1101 ADF2.
* 1110 ADF1.
* 1111 uicc.util.
* other values are RFU.

Application Provider specific data (1 byte):

* '00' for Package.
* '01' for Applet class.
* '02' for Applet Instance.

EXAMPLE: The AID of Package uicc.util is 'A0 00 00 00 09 00 05 FF FF FF FF 89 F0 00 00 00'.

## 4.7 Test equipment

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

### 4.7.1 Test tool

This test tool shall meet the following requirements:

* be able to send and receive APDU command to the UICC;
* the result of the I/O commands must be presented at the application layer;
* be able to provide results of the tests;
* shall send and/or compare all data specified in test file.

### 4.7.2 Interfaces and classes use

The test tool shall use some interfaces and classes, defined in annex D. They define the only allowed methods to write the test sources.

Interfaces and classes are defined as follow:

* UiccAdministrativeCommandsService defines administrative methods from ETSI TS 102 222 [7];
* UiccApplicationManagementService defines methods to load, install, select and delete applications;
* UiccCardManagementService defines methods to manage the card and its files;
* UiccToolkitService defines methods to manage toolkit commands;
* APDUResponse defines method to retrieve and check status words and data received from the card;
* UiccAPITestCardService defines the static method to get a reference of the class implementing all interfaces;
* UiccTestModel is an abstract class which shall be extended by every test source class; it defines the entry point run() method of the test script.

### 4.7.3 Util package

Annex D includes java source code of TestToolkitApplet abstract class of the uicc.util package. Each test applet shall extend this abstract class in order to retrieve test results when selecting it.

### 4.7.4 Java Software Development kit version

Java software development kit (SDK) version supported by JavaCard 2.2.2 specifications ([1], [2], [3]) is 1.4.1.