TRUSTED CONNECTIVITY ALLIANCE

eUICC Profile Package: Interoperable Format Technical Specification

Version 2.1

Published by simalliance now Trusted Connectivity Alliance

February 2017

Intellectual Property Rights (IPR) Disclaimer

Attention is drawn to the possibility that some of the elements of any material available for download from the specification pages on Trusted Connectivity Alliance's website may be the subject of Intellectual Property Rights (IPR) of third parties, some, but not all, of which are identified below.

Trusted Connectivity Alliance shall not be held responsible for identifying any or all such IPR, and has made no inquiry into the possible existence of any such IPR. TRUSTED CONNECTIVITY ALLIANCE SPECIFICATIONS ARE OFFERED WITHOUT ANY WARRANTY WHATSOEVER, AND IN PARTICULAR, ANY WARRANTY OF NON-INFRINGEMENT IS EXPRESSLY DISCLAIMED. ANY IMPLEMENTATION OF ANY TRUSTED CONNECTIVITY ALLIANCE SPECIFICATION SHALL BE MADE ENTIRELY AT THE IMPLEMENTER'S OWN RISK, AND NEITHER TRUSTED CONNECTIVITY ALLIANCE, NOR ANY OF ITS MEMBERS OR SUBMITTERS, SHALL HAVE ANY LIABILITY WHATSOEVER TO ANY IMPLEMENTER OR THIRD PARTY FOR ANY DAMAGES OF ANY NATURE WHATSOEVER DIRECTLY OR INDIRECTLY ARISING FROM THE IMPLEMENTATION OF ANY TRUSTED CONNECTIVITY ALLIANCE SPECIFICATION.

Table of Contents

1. Objective

The objective of this document is to define the technical specification of a standard format to be used for the loading and installation of an interoperable Profile Package in any compliant eUICC.
This specification is based on the following SIMalliance document: eUICC Profile Package: Interoperability Functional Requirements.

2. Introduction

The embedded UICC (eUICC), and the subsequent requirement for remote provisioning, has introduced the need for a number of operations, previously carried out in personalisation centres by individual UICC vendors, to be performed remotely in an open ecosystem.

This document specifies the structure and coding required to build, remotely load and install a profile in an eUICC.

The Profile Package, as technically specified in this document, represents the structure of data to be built by the Profile Creator and to be loaded in the eUICC in order for the eUICC to be personalised according to the content of the Profile Package.

This specification is intended primarily for Profile Creator providers, Profile Creator users (i.e. Mobile Network Operators or MNOs) and eUICC vendors in order for them to elaborate and exchange profiles with guaranteed interoperability.

In order to reduce complexity, the definition of the Profile Package does not support 2G SIM applications. This is not a limitation; for a terminal (e.g. a 2G M2M module) to be able to sustain remote provisioning of an eUICC according to this definition of the Profile Package, it shall support features defined in standard releases which also mandate the support of a UICC containing a USIM application to access a 2G network. This is aligned with requirements expressed in the GSMA Remote Provisioning Technical Specification [GS RPT], which require support of Release 9 for a device supporting eUICC.

eUICC ecosystem
The following illustration shows an example eUICC system environment. On the server side, interoperability is achieved on different levels (e.g. by the GSMA Remote Provisioning Technical Specifications [GS RPT]). The Subscription Manager (divided into two parts according to this specification) must interact with different entities like other SM, EUM (eUICC Manufacturer) or MNO.

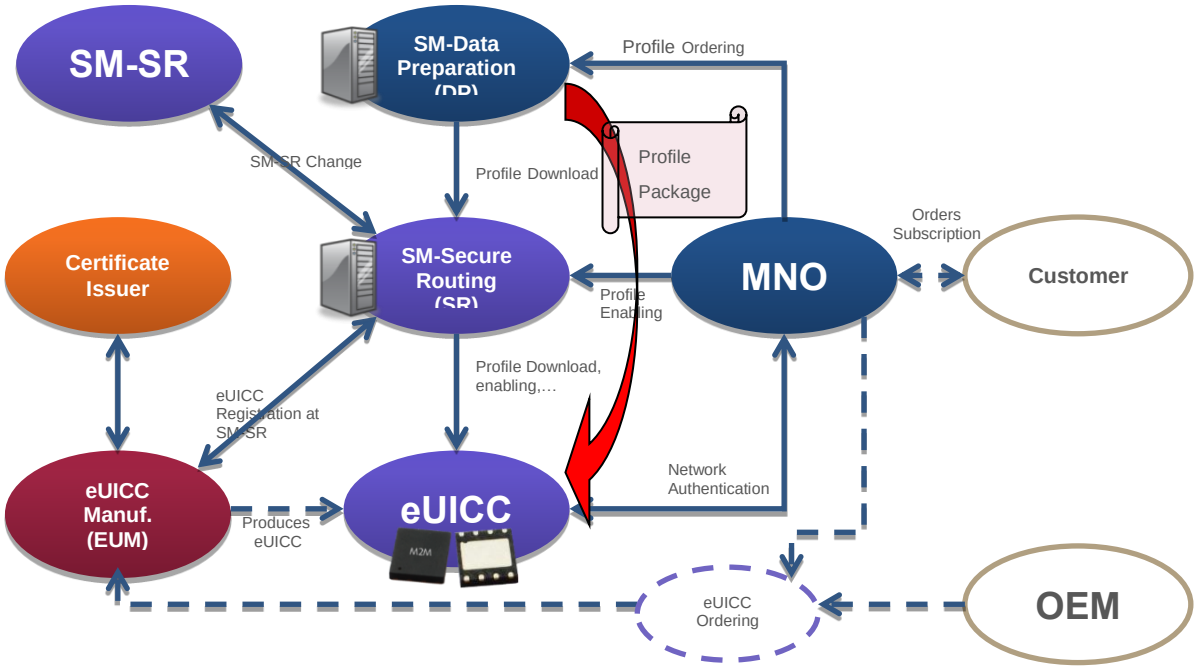


Figure 1: Example of eUICC ecosystem

3. Principles

- This specification is based on the requirements defined in the following SIMalliance specification: eUICC Profile Package: Interoperability Functional Requirements V1.1.
- This specification also takes into account the requirements defined in section 6.5 of ETSI TS 103 383.
- The standards referenced by this specification are only included to provide references on the context and the encoding of parameters used in this specification. They neither mandate the implementation of the version referenced nor mandate the support of related functionality.
- This specification also takes into account these GSMA documents:
  - Embedded SIM Remote Provisioning Architecture SGP.01 V1.1
  - Remote Provisioning Architecture for Embedded UICC Technical Specification SGP.02 V3.1

4. References

4.1 Normative References

- [101 220]: ETSI TS 101 220 V12.0.0: Smart Cards; ETSI numbering system for telecommunication application providers (Release 12)
- [102 221]: ETSI TS 102 221 V12.0.0: Smart Cards; UICC-Terminal interface; Physical and logical characteristics (Release 12)
- [102 222]: ETSI TS 102 222 V7.1.0: Integrated Circuit Cards (ICC); Administrative commands for telecommunications applications (Release 7)

- [102 226]: ETSI TS 102 226 V12.0.0: Smart Cards; Remote APDU structure for UICC based applications (Release 12)
- [USIM]: 3GPP TS 31.102 V12.6.0: Characteristics of the Universal Subscriber Identity Module (USIM) application (Release 12)
- [ISIM]: 3GPP TS 31.103 V12.2.0: Characteristics of the IP Multimedia Services Identity Module (ISIM) application (Release 12)
- [CSIM]: 3GPP2 C.S0065-C v1.0: cdma2000 Application on UICC for Spread Spectrum Systems
- [GP CS]: GlobalPlatform Card Specification V2.3
- [GP UC]: GlobalPlatform Card Specification UICC Configuration V2.0
- [GP CIC]: GlobalPlatform Card Specification Common Implementation Configuration – V2.0
- [GP AA]: Confidential Card Content Management; GlobalPlatform Card Specification Amendment A V1.1
- [GP AB]: GlobalPlatform Card Remote Application Management over HTTP Card Specification v2.2 – Amendment B v1.1.3
- [GP AC]: GlobalPlatform Card Technology Contactless Services Card Specification v2.2 – Amendment C Version 1.2
- [X.680]: ITU-T X.680 (11/2008): Abstract Syntax Notation One (ASN.1): Specification of basic notation including Corrigendum 1 and 2
- [X690]: ITU-T X.690 (11/2008): ASN.1 Encoding Rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER) including Corrigendum 1 and 2
- [GS RPT]: GSMA Remote Provisioning Architecture for Embedded UICC Technical Specification V3.0
- [TUAK]: 3GPP TS 35.231 V13.0.0: Specification of the TUAK algorithm set
- [3GTEST]: 3GPP TS 34.108 V12.3.0: Common test environments for User Equipment (UE); Conformance testing (Release 12)
- [S0016] 3GPP2 C.S0016-D Over-the-Air Service Provisioning of Mobile Stations in Spread Spectrum Standards Release D
- [MILENAGE]: 3GPP TS 35.206 V13.0.0: Specification of the MILENAGE Algorithm Set
- [CAVE]: TIA TR-45.AHAG Common Cryptographic Algorithms, Revision D.2

4.2 Informative References

- [GS RPA]: GSMA Remote Provisioning Architecture for Embedded UICC V1.1
- [102 383]: ETSI TS 103 383 V12.7.0: Smart Cards; Embedded UICC; Requirements Specification (Release 12)

5. Abbreviations

ADF	Application Dedicated File
AID	Application Identifier
APDU	Application Protocol Data Unit

ASN.1	Abstract Syntax Notation One
CASD	Controlling Authority Security Domain
CDMA	Code Division Multiple Access
CSIM	cdma2000 Subscriber Identify Identity Module
DF	Dedicated File
DGI	Data Grouping Identifier
DO	Data Object
EAP	Extensible Authentication Protocol
EF	Elementary File
eUICC	embedded UICC
EUM	eUICC Manufacturer
FCP	File Control Parameters
GBA	Generic Bootstrapping Architecture
HCI	Host Controller Interface
ICCID	Integrated Circuit Card ID
ID	Identifier
IMSI	International Mobile Subscriber Identity
ISIM	IP Multimedia Services Identity Module
LCSI	Life Cycle Status Information
M2M	Machine to Machine
MAC	Message Authentication Code
MAC-A	MAC used for authentication and key agreement
MBMS	Multimedia Broadcast/Multicast Service
MNO	Mobile Network Operator
MNO-SD	Mobile Network Operator Security Domain (Root SD of a Profile)
NAA	Network Access Application
NAC	Network Access Control
OID	Object Identifier
OS	Operating System (of the eUICC)
OTA	Over the Air
PE	Profile Element
PIN	Personal Identification Number
POL	Policy Rules within the Profile
PUK	PIN Unblocking Key
RAM	Remote Application Management
RFM	Remote File Management
SCP	Secure Channel Protocol
SD	Security Domain
SP	Service Provider
SQN	Sequence Number
SSD	Supplementary Security Domain
SWP	Single Wire Protocol
USIM	Universal Subscriber Identity Module

6. Definitions

embedded UICC	A UICC which is not easily accessible or replaceable, is not intended to be removed or replaced in the terminal, and enables the secure changing of Subscriptions.
Policy Rules	Defines the atomic action of a Policy and the conditions under which it is executed.
Profile	Combination of a file structure, data and applications on an eUICC.
Profile Creator	External entity in charge of creating the Profile Package based on MNO requirements, protecting the Profile Package from modification and/or content access.
Profile Element	A Profile Element is a part of the Profile Package representing one or several features of the Profile encoded using TLV structures based on ASN.1 description
Profile Package	A Personalised Profile using an interoperable description format transmitted to an eUICC in order to load and install a Profile
Provisioning	The downloading and installation of a Profile into an eUICC
Remote Provisioning	Provisioning done by the subscription manager on an eUICC outside of his premises, using a secure data link.

7. Profile Package General Structure

7.1 Introduction

The Profile Package is a collection of Profile Elements (PE) which uses a common description language. This description language is independent from the transport protocol. Each PE is described and can be processed by the eUICC independently from the others. A specific sequence is required for many PEs, however, because they will be processed by the eUICC in the context of previous PEs (i.e. some elements of the profile may be created only after higher level elements, such as a directory, is created; NAA parameters are applied to the NAA file structure created by previous PEs etc.). Examples of Profile Elements include: a file; a reference to a file system structure; a set of parameters for a specific NAA; an interoperable application etc.

The description of every PE in this specification is based on ASN.1 specified in [X.680] and encoded in TLV structures using DER (Distinguished Encoding Rule) encoding as specified in [X690]. This provides a flexible description and avoids the limitations of APDU protocol.

An identification number shall be associated to every PE. This identification number is used for error reporting.

A PE starts with a header containing the following information:

- PE identification number
- Optional flag indicating that the support of this PE is mandatory
- PE type
- PE length

7.2 Error management

A PE can be flagged in order to indicate that the support of the feature described by this PE is mandatory. If this feature is not supported by the eUICC, an error is reported to the Profile Creator, the processing of the Profile Package is cancelled and all of the PE already processed shall be discarded.

If a PE is not flagged as mandatory, and if the eUICC does not support the associated feature, the error is reported but the processing of the Profile Package continues. The coding of the error message is defined in section 8.11.

In order to avoid errors and warnings during the processing of a Profile Package, the Profile Creator may audit the targeted eUICC before building a Profile Package. In that case, all the features described in the Profile Package will be entirely supported by the eUICC. This is the best way to ensure predictable behaviour of the Profile when installed on a specific eUICC. If this procedure is not followed, a functional Profile in the eUICC may still be possible, but available features may be restricted.

The features that shall be supported by the Profile are also described in the Profile header. In case the eUICC does not support one of the features listed in this Profile header, the eUICC shall immediately return an error code and abort the processing of the Profile. This second mechanism complements the list of mandatory features encoded in the Profile header and is required for some specific features (e.g. proprietary features) that are not in the standardised list of features.

The behaviour of the eUICC when processing an incorrectly defined Profile Package (e.g. PE not provided in the right order, mandatory field missing or creation of an existing file) is unspecified. It may result in the installation of a Profile with unexpected behaviour or the failure of the installation whether the PE is mandated or not.

7.3 ASN.1 Module

The PE format is defined in a single, self-contained, ASN.1 definition module called `PEDefinitions`, with an ISO Object Identifier in the SIMalliance namespace:

```
PEDefinitions {joint-iso-itu-t(2) international-organizations(23)
simalliance(143) euicc-profile(1) spec-version(1) version-two(2)}
DEFINITIONS
AUTOMATIC TAGS
EXTENSIBILITY IMPLIED ::=
BEGIN
```

Two encoding/decoding attributes are defined:

- AUTOMATIC TAGS means that the tags are defined automatically using the encoding rules unless a tag notation is present in the PE format definition
- EXTENSIBILITY IMPLIED means that data types may contain additional elements that are not defined in this specification. eUICCs shall be ready to receive values with unknown tags following those tags defined in this specification. This is useful when processing PEs from a newer version of this specification and to handle proprietary tag values. When an eUICC encounters one of these unknown values, it shall report either an error or a warning using the code `invalid-parameter` as defined in section 8.11.

8. Profile Package Elements Definition

8.1 Common types

8.1.1 General Purpose types

To avoid ambiguity regarding the maximum allowed size of integers and octets strings, the following types and values, that are referenced in various PE definitions, are defined:

```
-- Basic integer types, for size constraints
maxUInt8 INTEGER ::= 255
UInt8 ::= INTEGER (0..maxUInt8)
maxUInt15 INTEGER ::= 32767
UInt15 ::= INTEGER (0..maxUInt15)
maxUInt16 INTEGER ::= 65535
UInt16 ::= INTEGER (0..maxUInt16)
-- maxUInt31 INTEGER ::= 2147483647
-- UInt31 ::= INTEGER (0..maxUInt31)
```

8.1.2 Profile specific types

The following types are used within several PE definitions:

```
ApplicationIdentifier ::= OCTET STRING (SIZE(5..16))
```

8.1.3 PE Header

The PE header is present at the beginning of all PEs described in this specification

```
PEHeader ::= SEQUENCE {
      mandated NULL OPTIONAL,
 -- if set, indicate that the support of this PE is mandatory
```

```
      identification UInt15 -- Identification number of this PE
}
```

The `mandated` field is used to indicate that the support of this PE is mandatory for the installation of this profile. If the eUICC does not support the following PE, it shall abort the processing of the profile and return an error to the sender of the profile.
The `identification` field is used to uniquely identify a PE within the profile. It will be used for error reporting to the sender of the profile.

The list of supported PEs is defined below:

```
ProfileElement ::= CHOICE {
      header ProfileHeader,

/* PEs */
      genericFileManagement PE-GenericFileManagement,
      pinCodes PE-PINCodes,
      pukCodes PE-PUKCodes,
      akaParameter PE-AKAParameter,
      cdmaParameter PE-CDMAParameter,
      securityDomain PE-SecurityDomain,
      rfm PE-RFM,
      application PE-Application,
      nonStandard PE-NonStandard,
      end PE-End,
      rfu1 PE-Dummy, -- this avoids renumbering of tag values
      rfu2 PE-Dummy, -- in case other non-file-system PEs are
      rfu3 PE-Dummy, -- added here in future versions
      rfu4 PE-Dummy,
      rfu5 PE-Dummy,

/* PEs related to file system creation using templates defined in this
specification */
      mf PE-MF,
      cd PE-CD,
      telecom PE-TELECOM,
      usim PE-USIM,
      opt-usim PE-OPT-USIM,
      isim PE-ISIM,
      opt-isim PE-OPT-ISIM,
      phonebook PE-PHONEBOOK,
      gsm-access PE-GSM-ACCESS,
      csim PE-CSIM,
      opt-csim PE-OPT-CSIM,
...
}

PE-Dummy ::= SEQUENCE {
}
```

It is important that PEs are sent in an order which do not create unresolved dependencies. The following rules shall be considered:

ProfileHeader
Shall be the first element and provided once within a profile download only.

PE-MF
May be provided once as the first element of the file system creation after the `ProfileHeader` PE. If this PE is not used, the MF shall be created as the first element of the file system using the `PE-GenericFileManagement`.

PE-CD
The use of this PE is optional and shall come after the creation of the MF.

PE-TELECOM
The use of this PE is optional and shall come after the creation of the MF.

PE-USIM
The use of this PE is optional and shall come after the creation of the MF.

PE-OPT-USIM
The use of this PE is optional and shall come after PE-USIM.

PE-ISIM
The use of this PE is optional and shall come after the creation of the MF.

PE-OPT-ISIM
The use of this PE is optional and shall come after PE-ISIM.

PE-GSM-ACCESS
The use of this PE is optional and shall come after PE-USIM.

PE-PHONEBOOK
The use of this PE is optional and shall come after PE-USIM.

PE-CSIM
The use of this PE is optional and shall come after the creation of the MF.

PE-OPT-CSIM
The use of this PE is optional and shall come after PE-CSIM.

PE-GenericFileManagement
Dependencies within the file system creation need to be considered. E.g. the DF Telecom may only be created when the MF has been created before.

PE-AKAParameters
If this PE is provided, it shall be present in the context of the creation of a NAA filesystem. It may be provided once or several times per NAA. If several sets of parameters are provided for one NAA, the set of parameters used by this NAA is not defined. This element is not allowed in the context of MF, SDs and applications.

PE-PINCodes

Shall be created in the context according to their scope. Global PINs (Application PINs according to ETSI TS 102 221) shall be provided once in the context of the creation of the MF of the UICC. Local PINs may be provided once in the context of the creation of a DF or ADF. Only a single PE-PINCodes is allowed in the context of the MF or in the context of a DF/ADF.

PE-PUKCodes
May only be provided once within the context of the UICC file system (MF). It needs to include all PUK codes for the complete profile. If this PE is not present in the Profile Package then no PUK codes are defined.

PE-SecurityDomain
Should be created after the creation of the file system, NAA parameters and PIN/PUK configuration.

PE-Application
Shall be provided after the creation of the SD the application will be associated to.

PE-RFM
Shall be provided after the creation of the SDs.

PE-NonStandard
In general this element may be provided in any position after the profile header. Further restrictions depend on the respective application.

PE-End
Shall be provided once at the end of the Profile Package.

8.2 Profile header

The Profile header PE is used once at the beginning of the profile in order to give various indications on the content on the profile:

```
ProfileHeader ::= SEQUENCE {
      major-version UInt8, -- set to 2 for this version of the specification
      minor-version UInt8, -- set to 1 for this version of the specification
      profileType UTF8String OPTIONAL, -- Profile type
      iccid OCTET STRING (SIZE (10)), -- ICCID of the Profile
      pol OCTET STRING  OPTIONAL,
      eUICC-Mandatory-services ServicesList,
      eUICC-Mandatory-GFSTEList SEQUENCE OF OBJECT IDENTIFIER,
      connectivityParameters OCTET STRING OPTIONAL
}
```

When receiving the Profile header, the eUICC shall check the `major-version`. If the version indicated by the Profile is not supported by the eUICC (e.g. if it is an earlier or an older version), the eUICC shall return an error `unsupported-profile-version` and stop the processing of the Profile. The `minor-version` is only informative, however, this may indicate that the Profile contains elements that the eUICC will not be able to process if it supports an older version of the specification. In that case, these elements will be ignored by the eUICC unless they are marked as mandatory in the PE header.

The `profileType` is a free optional text indicating for example, the name of the Profile issuer and the type of Profile.

The `iccid` contains the ICCID of the profile, the consistency of this value with the value provided in EF_{ICCID} is not checked by the eUICC and this value is not used by the eUICC in this version of the specification. It shall

be encoded non-swapped as per ITU E.118 representation and padded with 'F' if less digits are used (Example: 8947010000123456784F).

The `pol` contains the policy rules within a Profile (e.g. POL1 value as defined by GSMA in [GS RPT], Table 66). If this variable is not supplied in the Profile Package, its value shall be set to all 0 in the eUICC.

The `ServicesList` is used to indicate the services that shall be supported by the eUICC for the installation of a Profile. When a service is present in this sequence, and not supported or not known by the eUICC, the installation of the Profile Package shall be aborted.

```
ServicesList ::= SEQUENCE {
/* Contactless */
      contactless NULL OPTIONAL,

/* NAAs */
      usim NULL OPTIONAL,
      isim NULL OPTIONAL,
      csim NULL OPTIONAL,

/* NAA algorithms */
      milenage NULL OPTIONAL,
      tuak128 NULL OPTIONAL,
      cave NULL OPTIONAL,

/* USIM/ISIM services */
      gba-usim NULL OPTIONAL,
      gba-isim NULL OPTIONAL,
      mbms NULL OPTIONAL,
      eap NULL OPTIONAL,
/* Application Runtime environment */
      javacard NULL OPTIONAL,
      multos NULL OPTIONAL,

/* NAAs */
      multiple-usim NULL OPTIONAL,
      multiple-isim NULL OPTIONAL,
      multiple-csim NULL OPTIONAL,

/* Additional algorithms */
      tuak256 NULL OPTIONAL,
      usim-test-algorithm NULL OPTIONAL,

/* File type */
      ber-tlv NULL OPTIONAL,

/* Linked files */
      dfLink NULL OPTIONAL
}
```

The following list gives the features that the eUICC shall support in order to provide the associated service:

- contactless: support the SWP and HCI interfaces as well as the associated APIs
- usim: the USIM application as defined by 3GPP [USIM]

- isim: the ISIM application as defined by 3GPP [ISIM]
- csim: the CSIM application as defined by 3GPP2 [CSIM]
- milenage: the milenage AKA authentication algorithm as defined by 3GPP [MILENAGE]
- tuak128: the TUAK AKA authentication algorithm as defined by 3GPP [TUAK] with 128 bit key length
- tuak256: the TUAK AKA authentication algorithm as defined by 3GPP [TUAK] with 256 bit key length
- cave: the CAVE authentication algorithm as defined by TIA [CAVE]
- gba-usim: support of GBA authentication context in the USIM application
- gba-isim: support of GBA authentication context in the ISIM application
- mbms: support of the MBMS authentication context in the USIM application
- eap: support of the UICC EAP client
- javacard: support of the Java Card™ runtime environment
- multos: support of the Multos™ runtime environment
- multiple-usim: support of multiple USIM instances – requires "usim" to be present in the list
- multiple-isim: support of multiple ISIM instances – requires "isim" to be present in the list
- multiple-csim: support of multiple CSIM instances – requires "csim" to be present in the list
- ber-tlv: support of the BER-TLV Elementary File type
- dfLink: support of DF Link feature
- usim-test-algorithm: support of Test USIM Parameters for authentication test algorithm as defined by 3GPP [3GTEST]

When the Profile Package contains BER-TLV files, or DF links without indication in the `ServicesList` that these features shall be supported and the eUICC receiving this Profile Package does not support one of these features, the eUICC shall send a status code set to "`feature-not-supported`" without any "`additional-information`" and the installation shall continue without creating the BER-TLV file or the DF link. If "`mandated`" is set in the corresponding PE header, the installation of the Profile shall be aborted.

`eUICC-Mandatory-GFSTEList` contains a list of OIDs identifying file system templates which shall be supported by the eUICC in order for the Profile to be correctly installed on the eUICC. This list may contain the OIDs associated to the file system template defined in "ANNEX A (Normative): File Structure Templates Definition" of this specification. If a template OID present in the list is not supported by the eUICC the installation of the Profile Package shall be aborted.

The `connectivityParameters` contains the connectivity parameters as defined in GSMA in [GS RPT], in table 92, not including '3A07' DGI.

Usage rules: This PE shall be used once and shall be the first PE of the Profile Package.

8.3 File system

8.3.1 File system templates

Templates are defined in Annex A of this document. These templates are used to accelerate the creation of the file system in the Profile. Their use is optional. An alternate mechanism is defined in order to allow the creation of files without using these templates.

These templates define default values for:

- File size, number of records and record size
- Access conditions
- Content

These default values are not defined for all the files. In that case, these values shall be provided in the Profile. There are 2 types of templates:

- Created by default templates: All the files described in these templates will be created, even if they are not listed in the PE provided in the Profile Package (i.e. Flagged as OPTIONAL), except if they are tagged with "`doNotCreate`" in the "`File`" sequence.
- Not created by default templates: Only the file listed in the PE provided in the Profile Package will be created.

The templates also indicate an access rule reference which can be used to build the Access Rules Reference file content.
When using a template containing a hierarchy of files, Profile Creator shall take care to not instantiate files within a DF without instantiating the DF before.

8.3.2 File related types

These types are required for file system and file PE definitions.

The Profile Package uses only expanded format for the coding of the Access Rules.

```
ProprietaryInfo ::= SEQUENCE {
      specialFileInformation [PRIVATE 0] OCTET STRING (SIZE (1)) DEFAULT '00'H,

      /* fillPattern, repeatPattern
      only one of the parameters may be present. Coding and rules defined within
ETSI TS 102 222 [102 222] apply
      */

      fillPattern [PRIVATE 1] OCTET STRING (SIZE(1..200)) OPTIONAL,
      repeatPattern [PRIVATE 2] OCTET STRING (SIZE(1..200)) OPTIONAL
}

Fcp ::= SEQUENCE {
            /* The fileDescriptor shall be encoded as defined in
             ETSI TS 102 222 [102 222] */
         fileDescriptor [2] OCTET STRING (SIZE(2..4)) OPTIONAL,

            /* fileID
                  For ADFs, the fileID is a temporary value (named temporary
file ID in this document) used only during the profile creation. It has to be
unique within a profile and is used for referencing files within this ADF using
the file path.
            */
         fileID [3] OCTET STRING (SIZE(2)) OPTIONAL,

            /* dfName
                  Only applies for ADFs
            */
         dfName [4] ApplicationIdentifier OPTIONAL,

            /* lcsi
                  Coding according to ETSI TS 102 222 [102 222]
            */
         lcsi [10] OCTET STRING (SIZE (1)) DEFAULT '05'H,
```

```
            /* securityAttributesReferenced
            Either containing EF ARR ID[2] + record number[1] or
            record number[1] only and EF ARR ID implicitly known from the
            context, i.e. '2F06' within the MF and '6F06' otherwise
            */
         securityAttributesReferenced [11] OCTET STRING OPTIONAL,

            /* efFileSize
                  Mandatory for EF file types
                  Not allowed for DF files and EF link files
                  Shall be encoded on the minimum number of octets possible
                   (i.e. no leading bytes set to '00' are allowed)*/
         efFileSize [0] OCTET STRING OPTIONAL,

            /* pinStatusTemplateDO
                  Not allowed for EF files
                  Mandatory for DF/ADF files
            */
         pinStatusTemplateDO [PRIVATE 6] OCTET STRING OPTIONAL,

            /* shortEFID
                  Not allowed for DF files
                  Optional for EF file types / equivalent to ETSI TS 102 222
                  shortEFID not available: in case of a template file, SFI is set
according to the respective file specification. For files created by using
GenericFileManagement, SFI is calculated from FID
                  shortEFID available but not value: no SFI is supported
                  for this EF
                  shortEFID available with a length of 1 byte:
                  The Short File Identifier is coded from bits b8 to b4.
                  Bits b3,b2,b1 = 000.
            */
         shortEFID [8] OCTET STRING (SIZE (0..1)) OPTIONAL,

            /* proprietaryEFInfo
                  Optional for EF file types
                  Not allowed for DF files
            */
         proprietaryEFInfo [5] ProprietaryInfo OPTIONAL,

            /* linkPath
                  Specifies the path to the file to which shall be linked,
                  also valid for DFs/ADFs. Files within ADFs are addressed
                  by the temporary file ID of the respective ADF. For the coding
                  see filePath.
            */
         linkPath [PRIVATE 7] OCTET STRING OPTIONAL
}

File ::= SEQUENCE OF CHOICE {
      doNotCreate NULL, /* Indicates that this file shall not be created by the
eUICC even if present in a PE referencing a "Created by Default" template.
```

```
This flag has no effect for the creation of files in the MF and shall not be used
for all the files listed in a "Not Created by Default" template*/
      fileDescriptor Fcp,
      fillFileOffset UInt16,
      fillFileContent OCTET STRING

}
```

The "File" type is used during the creation of the file system when using a template. It contains 2 optional elements that are used to modify the content of the template during file creation or to set the content when it is not defined in the template.

The "Fcp" type contains all the file control parameters required for an ADF, DF or EF creation. All the elements contained in the "Fcp" are marked as optional. The parameters to be provided within the "Fcp" are context specific (See 8.3.3 and 8.3.5).

The "pinStatusTemplateDO" shall contain only a list of PIN Key Reference values coded according to table 9.3 of ETSI TS 102 221 [102 221] and used within the (A)DF. This list shall be returned by the (A)DF within the PIN status template DO according to ETSI TS 102 221 [102 221]. It shall not contain the full data object as defined in ETSI TS 102 221 [102 221] (e.g. '01810A'H as a typical value for an ADF_USIM).

Within "File" type, "Fcp" may be repeated to create a sequence of files (like several EF ICON files).
The "fillFileContent" type, preceded optionally by a "fillFileOffset" type, is used to set the content of a file (See "fillFileContent & fillFileOffset" field description in section 8.3.5). These types may be used repetitively for each file created.

The eUICC shall process the elements contained in the "File" type according to the diagram below to create no, one or several files and optionally fill them with content.

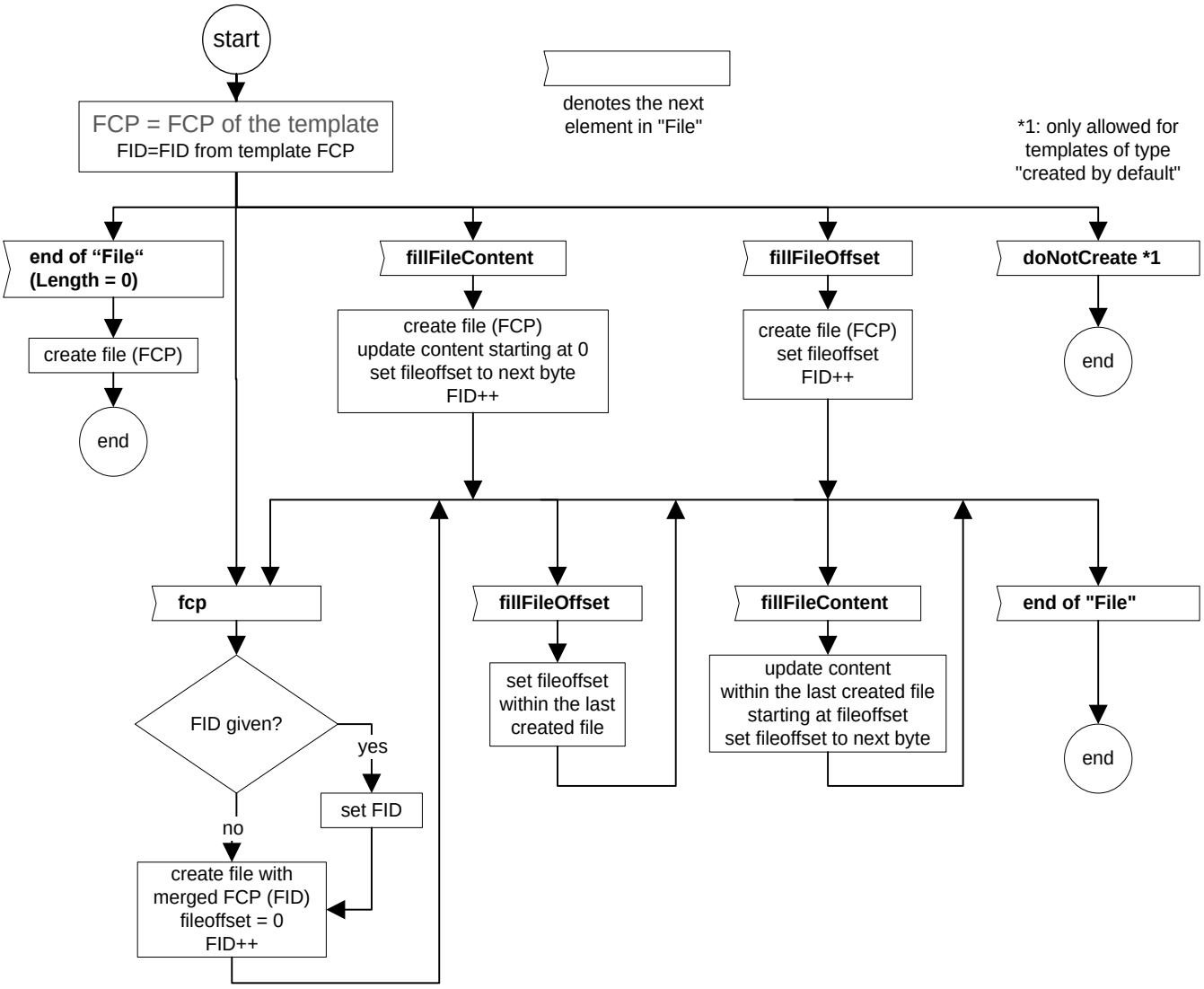


Figure 2: Processing of "File" type

NOTE: Not all sequences allowed by this diagram are useful (e.g. several sequential "`fillFileOffset`"). However, the processing defined above simplifies the rules to be followed and the implementation on the eUICC.

8.3.3 Template Modification Rules

For each template, default settings are defined within ANNEX A (Normative): File Structure Templates Definition. If no value is defined for a specific parameter, it has to be provided as a parameter within the template instance parameters (e.g. content of EF IMSI).
To overwrite parameters of the template, the following parameters may be specified within the FCP parameters defined within a PE. Depending on the file type defined in the template, the following parameters may be provided within the FCP of a PE to change the settings of the template for a respective file.
Changing the file type (byte 1 of fileDescriptor) as defined in the template is not allowed in the profile package.

Parameter	ADF		DF	DF Link	EF	EF Link
fileDescriptor	F		F	F	C	C (See Note 2)
fileID	C		C	C	C	C
dfName	M		F	F	F	F
lcsi	C		C	C	C	C
securityAttributesReferenced	C		C	C	C	C
efFileSize	F		F	F	C	C (See Note 2)
pinStatusTemplateDO	M		M	C (See Note 2)	F	F
shortEFID	F		F	F	C	C
proprietaryEFInfo	F		F	F	C	C (See Note 2)
linkPath	F		F	C (See Note 3)	C (See Note 1)	C (See Note 3)

M: Mandatory
Parameters marked as mandatory have to be provided. Otherwise the file creation will fail.

C: Conditional
Parameters marked with conditional may always be provided if the default value of the template shall be modified (e.g. change of securityAttributesReferenced).
In case no default value is defined within the template the respective conditional parameter is mandatory. Otherwise the creation will fail.

F: Forbidden
These parameters shall not be provided within the FCP since they are invalid within the respective context.

Note 1: Files defined as independent files within the template can be linked to an existing file (for files where content is required it is also possible to turn the file into a link rather than providing content). In this case the settings of the source file for fileDescriptor, efFileSize and proprietaryEFInfo will be applied for creating the file (the respective settings from the template will be ignored).

Note 2: Allowed only when a link is changed into an independent file. fileDescriptor and efFileSize can be used to modify the file size; proprietaryEFInfo can be used to alter the respective settings if needed.

Note 3: In case a link shall be turned in an independent file an empty linkPath needs to be provided. For EFs the FCP may include the parameters to define the file size (efFileSize and file Descriptor for record oriented files). By providing a linkPath value the link will be changed to the referenced file.

All file default contents defined within the template are defined as either repeat or fill patterns. There are two ways to alter the default:

- Overwrite Repeat/Fill Pattern:

A repeat or fill pattern provided within the respective "`Fcp`" will overwrite the default pattern completely. It does not matter whether the default has been defined as repeat or fill pattern. This means that in case the "`Fcp`" in the PE includes a fill pattern, but the template is defined as repeat pattern, the fill pattern from the PE will be applied (and vice versa).
This might be needed for some files where the default template size shall be modified (e.g. EF ICI, EF OCI).

- Using "`fillFileCOntent`" / "`fillFileOffset`":

Providing file content within "`fillFileCOntent`" / "`fillFileOffset`" will have the same effect as creating a file with a fill/repeat pattern and thereafter updating the content via Update.

8.3.4 File system PEs

8.3.4.1. MF PE

This PE is used to create and set the content of the files at the MF level. It is based on the template defined in Annex A, section 9.2. The template referenced by the PE is a "Created by default" type template. The rules associated with this kind of template will be used by the eUICC.

```
PE-MF ::= SEQUENCE {
      mf-header PEHeader,
      templateID OBJECT IDENTIFIER,
```

```
      mf File,
      ef-pl File OPTIONAL,
      ef-iccid File,
      ef-dir File,
      ef-arr File,
      ef-umpc File OPTIONAL
}
```

Usage rules: This PE shall be used only once at the beginning of the profile Package.

8.3.4.2. DF CD PE

This PE is used to create the DF CD and to create and set the content of the files at the DF CD level. It is based on the template defined in Annex A, section 9.3. The use of this PE is optional. If this PE is not received by the eUICC, it will not create DF CD in the profile. The template referenced by the PE is a "Not created by default" type template.

```
PE-CD ::= SEQUENCE {
      cd-header PEHeader,
      templateID OBJECT IDENTIFIER,
      df-cd File,
      ef-launchpad File OPTIONAL,
      ef-icon File OPTIONAL
}
```

Usage rules: This PE may be used only once after the creation of the MF.

8.3.4.3. DF TELECOM PE

This PE is used to create the DF TELECOM, to create the DFs under the DF TELECOM and to create and set the content of the EFs at the DF TELECOM and sub DFs level. It is based on the template defined in Annex A, section 9.4. The use of this PE is optional. If this PE is not received by the eUICC, it will not create DF TELECOM in the profile. The template referenced by the PE is a "Not created by default" type template.

```
PE-TELECOM ::= SEQUENCE {
      telecom-header PEHeader,
      templateID OBJECT IDENTIFIER,
      df-telecom File,
      ef-arr File OPTIONAL,
      ef-rma File OPTIONAL,
      ef-sume File OPTIONAL,
      ef-ice-dn File OPTIONAL,
      ef-ice-ff File OPTIONAL,
      ef-psismsc File OPTIONAL,
      df-graphics File OPTIONAL,
        ef-img File OPTIONAL,
        ef-iidf File OPTIONAL,
        ef-ice-graphics File OPTIONAL,
        ef-launch-scws File OPTIONAL,
        ef-icon File OPTIONAL,
      df-phonebook File OPTIONAL,
        ef-pbr File OPTIONAL,
        ef-ext1 File OPTIONAL,
        ef-aas File OPTIONAL,
```

```
        ef-gas File OPTIONAL,
        ef-psc File OPTIONAL,
        ef-cc File OPTIONAL,
        ef-puid File OPTIONAL,
        ef-iap File OPTIONAL,
        ef-adn File OPTIONAL,
        ef-pbc File OPTIONAL,
        ef-anr File OPTIONAL,
        ef-puri File OPTIONAL,
        ef-email File OPTIONAL,
        ef-sne File OPTIONAL,
        ef-uid File OPTIONAL,
        ef-grp File OPTIONAL,
        ef-ccp1 File OPTIONAL,
      df-multimedia File OPTIONAL,
        ef-mml File OPTIONAL,
        ef-mmdf File OPTIONAL,
      df-mmss File OPTIONAL,
        ef-mlpl File OPTIONAL,
        ef-mspl File OPTIONAL,
        ef-mmssmode File OPTIONAL
}
```

Usage rules: This PE may be used only once after the creation of the MF. Additional files may be required that are not part of this template. These files shall be created using the GenericFileManagement PE.

8.3.4.4. USIM Related Files and Directories

8.3.4.4.1. USIM "Created by default" Files PE

This PE is used to create a USIM ADF and to create and set the content of the files either mandatory or always used at the DF USIM level. The template referenced by the PE is a "Created by default" type template. This PE is based on the template defined in Annex A, section 9.5.1.

```
PE-USIM ::= SEQUENCE {
      usim-header PEHeader,
      templateID OBJECT IDENTIFIER,
      adf-usim File,
      ef-imsi File,
      ef-arr File,
      ef-keys File OPTIONAL,
      ef-keysPS File OPTIONAL,
      ef-hpplmn File OPTIONAL,
      ef-ust File, /* The content of UST file shall be modified by the eUICC
during profile installation according to the functionality supported by the eUICC
platform  i.e. in the case where a service is not supported (and not indicated as
required) the related bit(s) will be set to zero */
      ef-fdn File OPTIONAL,
      ef-sms File OPTIONAL,
      ef-smsp File OPTIONAL,
      ef-smss File OPTIONAL,
      ef-spn File,
      ef-est File,
```

```
      ef-start-hfn File OPTIONAL,
      ef-threshold File OPTIONAL,
      ef-psloci File OPTIONAL,
      ef-acc File,
      ef-fplmn File OPTIONAL,
      ef-loci File OPTIONAL,
      ef-ad File OPTIONAL,
      ef-ecc File,
      ef-netpar File OPTIONAL,
      ef-epsloci File OPTIONAL,
      ef-epsnsc File OPTIONAL
}
```

Usage rules: This PE may be used several times after the creation of the MF. This PE shall be followed immediately by PE-OPT-USIM if required and before any other PE.

8.3.4.4.2. USIM "Not Created by default" Files PE

This PE is used to create the files less often used under the USIM DF previously created. The template referenced by the PE is a "Not created by default" type template. This PE is based on the template defined in Annex A, section 9.5.2.

```
PE-OPT-USIM ::= SEQUENCE {
      optusim-header PEHeader,
      templateID OBJECT IDENTIFIER,
      ef-li File OPTIONAL,
      ef-acmax File OPTIONAL,
      ef-acm File OPTIONAL,
      ef-gid1 File OPTIONAL,
      ef-gid2 File OPTIONAL,
      ef-msisdn File OPTIONAL,
      ef-puct File OPTIONAL,
      ef-cbmi File OPTIONAL,
      ef-cbmid File OPTIONAL,
      ef-sdn File OPTIONAL,
      ef-ext2 File OPTIONAL,
      ef-ext3 File OPTIONAL,
      ef-cbmir File OPTIONAL,
      ef-plmnwact File OPTIONAL,
      ef-oplmnwact File OPTIONAL,
      ef-hplmnwact File OPTIONAL,
      ef-dck File OPTIONAL,
      ef-cnl File OPTIONAL,
      ef-smsr File OPTIONAL,
      ef-bdn File OPTIONAL,
      ef-ext5 File OPTIONAL,
      ef-ccp2 File OPTIONAL,
      ef-ext4 File OPTIONAL,
      ef-acl File OPTIONAL,
      ef-cmi File OPTIONAL,
      ef-ici File OPTIONAL,
      ef-oci File OPTIONAL,
```

```
      ef-ict File OPTIONAL,
      ef-oct File OPTIONAL,
      ef-vgcs File OPTIONAL,
      ef-vgcss File OPTIONAL,
      ef-vbs File OPTIONAL,
      ef-vbss File OPTIONAL,
      ef-emlpp File OPTIONAL,
      ef-aaem File OPTIONAL,
      ef-hiddenkey File OPTIONAL,
      ef-pnn File OPTIONAL,
      ef-opl File OPTIONAL,
      ef-mbdn File OPTIONAL,
      ef-ext6 File OPTIONAL,
      ef-mbi File OPTIONAL,
      ef-mwis File OPTIONAL,
      ef-cfis File OPTIONAL,
      ef-ext7 File OPTIONAL,
      ef-spdi File OPTIONAL,
      ef-mmsn File OPTIONAL,
      ef-ext8 File OPTIONAL,
      ef-mmsicp File OPTIONAL,
      ef-mmsup File OPTIONAL,
      ef-mmsucp File OPTIONAL,
      ef-nia File OPTIONAL,
      ef-vgcsca File OPTIONAL,
      ef-vbsca File OPTIONAL,
      ef-gbabp File OPTIONAL,
      ef-msk File OPTIONAL,
      ef-muk File OPTIONAL,
      ef-ehplmn File OPTIONAL,
      ef-gbanl File OPTIONAL,
      ef-ehplmnpi File OPTIONAL,
      ef-lrplmnsi File OPTIONAL,
      ef-nafkca File OPTIONAL,
      ef-spni File OPTIONAL,
      ef-pnni File OPTIONAL,
      ef-ncp-ip File OPTIONAL,
      ef-ufc File OPTIONAL,
      ef-nasconfig File OPTIONAL,
      ef-uicciari File OPTIONAL,
      ef-pws File OPTIONAL,
      ef-fdnuri File OPTIONAL,
      ef-bdnuri File OPTIONAL,
      ef-sdnuri File OPTIONAL,
      ef-iwl File OPTIONAL,
      ef-ips File OPTIONAL,
      ef-ipd File OPTIONAL
}
```

Usage rules: This PE can be used once for each USIM application immediately after the creation of the USIM Mandatory files. It may be followed by PE-PINCodes for the creation of USIM local PINs.

8.3.4.4.3. *DF PHONEBOOK PE*

This PE is used to create the DF PHONEBOOK inside the ADF USIM and the EFs contained in DF PHONEBOOK. It is based on part of the template defined in Annex A, section 9.4. The use of this PE is optional. If this PE is not received by the eUICC, it will not create DF TELECOM in the profile. The template referenced by the PE is a "Not created by default" type template.

```
PE-PHONEBOOK ::= SEQUENCE {
      phonebook-header PEHeader,
      templateID OBJECT IDENTIFIER,
      df-phonebook File,
      ef-pbr File OPTIONAL,
      ef-ext1 File OPTIONAL,
      ef-aas File OPTIONAL,
      ef-gas File OPTIONAL,
      ef-psc File OPTIONAL,
      ef-cc File OPTIONAL,
      ef-puid File OPTIONAL,
      ef-iap File OPTIONAL,
      ef-adn File OPTIONAL,
      ef-pbc File OPTIONAL,
      ef-anr File OPTIONAL,
      ef-puri File OPTIONAL,
      ef-email File OPTIONAL,
      ef-sne File OPTIONAL,
      ef-uid File OPTIONAL,
      ef-grp File OPTIONAL,
      ef-ccp1 File OPTIONAL
}
```

Usage rules: This PE may be used only once after the creation of the USIM ADF.

8.3.4.4.4. *DF GSM ACCESS PE*

This PE is used to create the DF GSM ACCESS and to create and set the content of the files at the DF GSM ACCESS level. The use of this PE is optional. If this PE is not received by the eUICC, it will not create DF GSM ACCESS in the profile. The template referenced by the PE is a "Not created by default" type template. This PE is based on the template defined in Annex A, section 9.5.4.

```
PE-GSM-ACCESS ::= SEQUENCE {
      gsm-access-header PEHeader,
      templateID OBJECT IDENTIFIER,
      df-gsm-access File,
      ef-kc File OPTIONAL,
      ef-kcgprs File OPTIONAL,
      ef-cpbcch File OPTIONAL,
      ef-invscan File OPTIONAL
}
```

Usage rules: This PE may be used only once after the creation of the USIM ADF.

8.3.4.5. ISIM Related Files and Directories

8.3.4.5.1. *ISIM "Created by default" Files PE*

This PE is used to create an ISIM ADF and to create and set the content of the mandatory files at the DF ISIM level. The template referenced by the PE is a "Created by default" type template. This PE is based on the template defined in Annex A, section 9.6.1.

```
PE-ISIM ::= SEQUENCE {
      isim-header PEHeader,
      templateID OBJECT IDENTIFIER,
      adf-isim File,
      ef-impi File,
      ef-impu File,
      ef-domain File,
      ef-ist File, /* The content of IST file shall be modified by the eUICC
during profile installation according to the functionality supported by the eUICC
platform i.e. in the case where a service is not supported (and not indicated as
required) the related bit(s) will be set to zero */
      ef-ad File OPTIONAL,
      ef-arr File
}
```

Usage rules: This PE may be used several times after the creation of the MF. This PE shall be followed immediately by PE-OPT-ISIM if required and before any other PE.

8.3.4.5.2. ISIM "Not Created by default" Files PE

This PE is used to create the optional files under the ISIM ADF previously created. The template referenced by the PE is a "Not created by default" type template. This PE is based on the template defined in Annex A, section 9.6.2.

```
PE-OPT-ISIM ::= SEQUENCE {
      optisim-header PEHeader,
      templateID OBJECT IDENTIFIER,
      ef-pcscf File OPTIONAL,
      ef-sms File OPTIONAL,
      ef-smsp File OPTIONAL,
      ef-smss File OPTIONAL,
      ef-smsr File OPTIONAL,
      ef-gbabp File OPTIONAL,
      ef-gbanl File OPTIONAL,
      ef-nafkca File OPTIONAL,
      ef-uicciari File OPTIONAL
}
```

Usage rules: This PE can be used once for each ISIM application immediately after the creation of the ISIM Mandatory files. It may be followed by PE-PINCodes for the creation of ISIM local PINs.

8.3.4.6. CSIM Related Files and Directories

8.3.4.6.1. CSIM "Created by default" Files PE

This PE is used to create a CSIM ADF and to create and set the content of the mandatory files at the DF CSIM level. The template referenced by the PE is a "Created by default" type template. This PE is based on the template defined in Annex A, section 9.7.1.

```
PE-CSIM ::= SEQUENCE {
      csim-header PEHeader,
      templateID OBJECT IDENTIFIER,
      adf-csim File,
      ef-arr File,
      ef-call-count File,
      ef-imsi-m File,
      ef-imsi-t File,
      ef-tmsi File,
      ef-ah File,
      ef-aop File,
      ef-aloc File,
      ef-cdmahome File,
      ef-znregi File,
      ef-snregi File,
      ef-distregi File,
      ef-accolc File,
      ef-term File,
      ef-acp File,
      ef-prl File,
      ef-ruimid File,
      ef-csim-st File,
      ef-spc File,
      ef-otapaspc File,
      ef-namlock File,
      ef-ota File,
      ef-sp File,
      ef-esn-meid-me File,
      ef-li File,
      ef-usgind File,
      ef-ad File,
      ef-max-prl File,
      ef-spcs File,
      ef-mecrp File,
      ef-home-tag File,
      ef-group-tag File,
      ef-specific-tag File,
      ef-call-prompt File
}
```

Usage rules: This PE may be used several times after the creation of the MF. This PE shall be followed immediately by PE-OPT-CSIM if required and before any other PE.

8.3.4.6.2. CSIM "Not Created by default" Files PE

The PE is used to create the optional files under the CSIM DF previously created. The template referenced by the PE is a "Not created by default" type template. This PE is based on the template defined in Annex A, section 9.7.2.

```
PE-OPT-CSIM ::= SEQUENCE {
      optcsim-header PEHeader,
      templateID OBJECT IDENTIFIER,
```

```
      ef-ssci File OPTIONAL,
      ef-fdn File OPTIONAL,
      ef-sms File OPTIONAL,
      ef-smsp File OPTIONAL,
      ef-smss File OPTIONAL,
      ef-ssfc File OPTIONAL,
      ef-spn File OPTIONAL,
      ef-mdn File OPTIONAL,
      ef-ecc File OPTIONAL,
      ef-me3gpdopc File OPTIONAL,
      ef-3gpdopm File OPTIONAL,
      ef-sipcap File OPTIONAL,
      ef-mipcap File OPTIONAL,
      ef-sipupp File OPTIONAL,
      ef-mipupp File OPTIONAL,
      ef-sipsp File OPTIONAL,
      ef-mipsp File OPTIONAL,
      ef-sippapss File OPTIONAL,
      ef-puzl File OPTIONAL,
      ef-maxpuzl File OPTIONAL,
      ef-hrpdcap File OPTIONAL,
      ef-hrpdupp File OPTIONAL,
      ef-csspr File OPTIONAL,
      ef-atc File OPTIONAL,
      ef-eprl File OPTIONAL,
      ef-bcsmscfg File OPTIONAL,
      ef-bcsmspref File OPTIONAL,
      ef-bcsmstable File OPTIONAL,
      ef-bcsmsp File OPTIONAL,
      ef-bakpara File OPTIONAL,
      ef-upbakpara File OPTIONAL,
      ef-mmsn File OPTIONAL,
      ef-ext8 File OPTIONAL,
      ef-mmsicp File OPTIONAL,
      ef-mmsup File OPTIONAL,
      ef-mmsucp File OPTIONAL,
      ef-auth-capability File OPTIONAL,
      ef-3gcik File OPTIONAL,
      ef-dck File OPTIONAL,
      ef-gid1 File OPTIONAL,
      ef-gid2 File OPTIONAL,
      ef-cdmacnl File OPTIONAL,
      ef-sf-euimid File OPTIONAL,
      ef-est File OPTIONAL,
      ef-hidden-key File OPTIONAL,
      ef-lcsver File OPTIONAL,
      ef-lcscp File OPTIONAL,
      ef-sdn File OPTIONAL,
      ef-ext2 File OPTIONAL,
      ef-ext3 File OPTIONAL,
      ef-ici File OPTIONAL,
      ef-oci File OPTIONAL,
```

```
      ef-ext5 File OPTIONAL,
      ef-ccp2 File OPTIONAL,
      ef-applabels File OPTIONAL,
      ef-model File OPTIONAL,
      ef-rc File OPTIONAL,
      ef-smscap File OPTIONAL,
      ef-mipflags File OPTIONAL,
      ef-3gpduppext File OPTIONAL,
      ef-ipv6cap File OPTIONAL,
      ef-tcpconfig File OPTIONAL,
      ef-dgc File OPTIONAL,
      ef-wapbrowsercp File OPTIONAL,
      ef-wapbrowserbm File OPTIONAL,
      ef-mmsconfig File OPTIONAL,
      ef-jdl File OPTIONAL
}
```

Usage rules: This PE can be used once for each CSIM application immediately after the creation of the USIM Mandatory files. It may be followed by PE-PINCodes for the creation of CSIM local PINs.

8.3.5 Generic File management PE

This PE is used in order to create files in a generic way. This will typically be used in case files are not defined in a file system template (e.g. application specific files, future standard files not covered by the templates) or if specific templates are not supported by the eUICC Profile Package interpreter.

The Generic File Management PE consists of a list of file system operations and follows the same approach as the one described within the existing standards to establish files within a Profile.
Any file system operation is always executed within the current context. Files are always created within the current DF. File updates are always applied to the currently selected EF.

The default selection at the beginning of this PE is as follows:

- Current DF: MF
- Current EF: no selection

The following operations are available:

`filePath`:
Selects an already existing DF or ADF according to the rules in ETSI TS 102 221 [102 221] for "select by path from MF". It is a concatenation of file identifiers and has even length or length zero for selecting the MF. To select an ADF or a DF in an ADF, the temporary File ID of the ADF shall be used.

`createFCP`:
The `createFCP` structure is used to create files (See "`Fcp`" type in section 8.3.2). Coding of the parameters is based on ETSI 102 222 [102 222] and tailored to profile download to minimise the profile download size and to support linked files.
The following file types can be created using this structure; Linear Fixed, Binary, Cyclic, BER-TLV, linked EFs, linked ADFs/DFs.
The file with the exception of ADFs will always be created within the currently selected DF/ADF. In case a DF/ADF is created it will be automatically selected. No EF will be selected in this case. When an EF has been created it will be automatically selected as the current EF.

The creation of a file may require the support of a specific feature. If such a feature is not indicated as "mandatory" in the `eUICC-Mandatory-services`, the related files may or may not be created as required by the Profile Package (e.g. EF_UST, EF_IST, GBA or MBMS related files, BER-TLV files). In these cases `feature-not-supported` shall be returned by the eUICC.

`fillFileOffset` & `fillFileContent`:
These commands are used to provide content for EFs. There is always a current `fillFileOffset` pointer. After EF selection the current `fillFileOffset` pointer is set to the beginning of the file (e.g. after creation or after filePath).
`fillFileOffset` is a binary pointer. Record based files and binary files are handled in the same way.
For record based files the current `fillFileOffset` may reference to any byte within a record.

e.g. LF file with 100 records and 20 bytes per record:
Default `fillFilePointer` = 1
`fillFileOffset`: 20 sets `fillFilePointer` to beginning of record 2 (Byte 21)
`fillFileContent`: 00 updates byte 21 and sets `fillFilePointer` to record 2 byte 2 (Byte 22)
`fillFileOffset`: 24 sets `fillFilePointer` pointer to beginning of record 3 byte 6 (Byte 46)
`fillFileContent`: 00112233445566778899AABBCCDDEEFF updates byte 46 to 61 and sets `fillFilePointer` to record 3 byte 2 (Byte 62)

`fillFilePointer`: > 2000 undefined behaviour but writing a value beyond the file size will generate an error “Invalid parameter”.

For Cyclic files the record pointer of the file is not affected during the creation of the file and the setting of its content. After creation of the Profile in the eUICC, the record pointer will be set to the first record created during the processing of the Profile Package.

Content is personalised using `fillFileContent`. The content will be personalised starting at the current `fillFileOffset` pointer which will be implicitly set to the next unpersonalised content (`fillFileOffset` pointer new= `fillFileOffset` pointer + length of `fillFileContent`).

Since all parameters (except securityAttributesReferenced) for the Fcp type are optional the minimum parameters must be provided for generic File Creation:

Parameter	Create ADF	Create DF	Create DF Link	Create EF	Create EF Link
fileDescriptor	M	M	M	M	M
fileID	M	M	M	M	M
dfName	M	F	F	F	F
lcsi	O	O	O	O	O
securityAttributesReferenced	M	M	M	M	M
efFileSize	F	F	F	M	F
pinStatusTemplateDO	M	M	F	F	F
shortEFID	F	F	F	O	O
proprietaryEFInfo	F	F	F	O	F
linkPath	F	F	M	F	M

M: Mandatory
This parameter has to be set within the FCP when the respective type is created. Otherwise creation will fail.

O: Optional
Parameters which are optional do not need to be provided since they either address optional features (ShortEFID) or a default will be applied (LCSI, proprietaryInfo, pinStatus_TemplateDO: copy of file addressed in linkPath).

F Forbidden
This parameter shall not be provided within the respective context.

```
/* Create GenericFileManagement
*/
PE-GenericFileManagement ::= SEQUENCE {
            gfm-header PEHeader,
            fileManagementCMD SEQUENCE (SIZE (1..MAX)) OF FileManagement
      }

FileManagement ::= SEQUENCE (SIZE (1..MAX)) OF CHOICE {
      filePath [0] OCTET STRING, -- Use Temporary File ID for ADF
      createFCP [APPLICATION 2] Fcp,
      fillFileOffset UInt16,
      fillFileContent [1] OCTET STRING
      }
```

Usage rules: This PE may be used at any time after the creation of the `ProfileHeader`. It shall be the first element of the file system creation in case it is used to create the MF instead of using PE-MF.

8.4 NAA(s)

8.4.1 NAA Parameters

An NAA is implicitly installed in the context of the creation of the NAA file structure and includes the following PE provided subsequent to the PEs describing its file system:

- PE-AKAParameters
- PE-CDMAParameter (if <NAA> = CSIM)

8.4.2 AKA Parameters PE

This PE is used to set the parameters for AKA authentication algorithms like Milenage [MILENAGE], TUAK [TUAK] and usim-test-algorithm [3GTEST].

```
MappingParameter ::= SEQUENCE {
      mappingOptions    OCTET STRING (SIZE(1)),
      mappingSource     ApplicationIdentifier
}

AlgoParameter ::= SEQUENCE {
      algorithmID INTEGER {
         milenage(1),
         tuak(2),
         usim-test-algorithm(3)
      },
      algorithmOptions  OCTET STRING (SIZE(1)),
      key OCTET STRING,
      opc OCTET STRING, /* OPc for Milenage; TOPc for TUAK; ignored in case of
usim-test-algorithm */
```

```
      /* rotationConstants only apply for Milenage; ignored in case of TUAK and
usim-test-algorithm */
      rotationConstants OCTET STRING (SIZE (5)) DEFAULT '4000204060'H,

      /* xoringConstants only apply for Milenage; ignored in case of TUAK and
usim-test-algorithm */
      xoringConstants   OCTET STRING (SIZE (80)) DEFAULT
'00000000000000000000000000000000000000000000000000000000000000010000000000000000
00000000000000020000000000000000000000000000000400000000000000000000000000000000
8 'H,
      authCounterMax    OCTET STRING (SIZE(3)) OPTIONAL, /* ignored in case of
usim-test-algorithm */

      /* Number of iterations of Keccak-f[1600] (noted: Π ) permutation as
recommended by 3GPP TS 35.231 [TUAK] in section 7.2.
      This parameter only applies for TUAK; ignored otherwise.*/
      numberOfKeccak    UInt8 DEFAULT 1
}

PE-AKAParameter ::= SEQUENCE {
      aka-header PEHeader,
      algoConfiguration CHOICE {
            mappingParameter  MappingParameter,
            algoParameter     AlgoParameter
      },

      sqnOptions        OCTET STRING (SIZE(1)) DEFAULT '02'H, /* ignored in case
of usim-test-algorithm */
      -- maximum value for sqnDelta and sqnAgeLimit is '07FFFFFFFFFF'H
      sqnDelta          OCTET STRING (SIZE(6)) DEFAULT '000010000000'H, /*
ignored in case of usim-test-algorithm */
      sqnAgeLimit       OCTET STRING (SIZE(6)) DEFAULT '000010000000'H, /*
ignored in case of usim-test-algorithm */

      -- Sequence numbers do not include the index (IND)
      -- maximum for any values within sqnInit is '07FFFFFFFFFF'H
      sqnInit SEQUENCE (SIZE (32)) OF OCTET STRING (SIZE (6)) DEFAULT {
/* Index 0 */'000000000000'H, '000000000000'H, '000000000000'H,
'000000000000'H,'000000000000'H, '000000000000'H, '000000000000'H,
'000000000000'H, '000000000000'H, '000000000000'H, '000000000000'H,
'000000000000'H,'000000000000'H, '000000000000'H, '000000000000'H,
'000000000000'H, '000000000000'H, '000000000000'H, '000000000000'H,
'000000000000'H,'000000000000'H, '000000000000'H, '000000000000'H,
'000000000000'H, '000000000000'H, '000000000000'H, '000000000000'H,
'000000000000'H,'000000000000'H, '000000000000'H, '000000000000'H,
/* Index 31 */'000000000000'H } /* ignored in case of usim-test-algorithm */
}
```

The `algorithm-Options` is encoded as follows:

Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Meaning
-	-	-	-	-	X	X	X	RES size (0: 32 bits ,1: 64 bits, 2:128 bits, 3: 256 bits)[1]
		X	X	X				MAC-A and MAC-S size (0: 64 bits ,1: 128 bits, 2: 256 bits)[1]
	X	-	-	-	-	-	-	CK and IK size (0: 128 bits ,1: 256 bits)[1]
X	-	-	-	-	-	-	-	RFU
Note 1: Setting only applies for TUAK. Will be ignored in case of Milenage and usim-test-algorithm								

In case of Milenage algorithm and USIM test algorithm, RES, MAC-S, CK and IK size are fixed by their respective specifications (RES size is fixed to 128 bits for USIM test algorithm).

The `sqnOptions` is encoded as follows:

Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Meaning
-	-	-	-	-	-	-	X	Anonymity Key (AK) (1: not used, 0: used)
-	-	-	-	-	-	X		SQN wrap around (1: not allowed, 0: allowed) In case SQN wrap around is allowed it means that SQN verification will be disabled if the respective SEQ value has reached the maximum value 07FFFFFFFFFF
-	-	-	-	-	X	-	-	SQN Delta (1: not used, 0: used)
-	-	-	-	X	-	-	-	SQN Age Limit (1: not used, 0: used)
X	X	X	X	-	-	-	-	RFU

The `mappingOptions` data element, if present, indicates the AKA parameters the current NAA uses from the application referenced by `mappingSource` and is encoded as follows:

Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Meaning
-	-	-	-	-	X	X	-	00: dot not share SQN parameters and/or values 01: share `sqnInit, sqnOptions, sqnDelta, sqnAgeLimit` 10: share `sqnOptions, sqnDelta, sqnAgeLimit` 11: share `sqnOptions`, `sqnDelta`, `sqnAgeLimit` and SQN array
X	X	X	X	X	-	-	X	RFU

The `algorithmID`, `algorithmOptions, key, opc, rotationConstants, xoringConstants` and `numberOfKeccak` are always shared when the mapping is used.
Every NAA which supports PE-AKA maintains either its own SQN array created implicitly by the eUICC (including 32 SQNs initialised to zero or to the value given in the optional `sqnInit` array) or a reference to the SQN array of another NAA depending on the settings provided in the `mappingOptions` parameters.

Key size: The `key` OBJECT STRING shall have a length of 16 bytes in case of the Milenage or usim-test-algorithm and 16 or 32 bytes in case of the TUAK algorithm.

OPc size: The `opc` OBJECT STRING shall have a size of 16 bytes in case of the Milenage and 32 bytes in case of the TUAK algorithm.

In case a value is provided for `authCounterMax` it defines the accumulated number of Authenticate Commands for all the NAA over the complete life time of the profile (independent from resets, profile de-/activation). It shall be provided once in a Profile Package. Once the actual number of Authenticate commands reaches the defined value the command should fail and return '6F00'h as the respective error code.

Usage rules: This PE shall be used once after the creation of a NAA using Milenage or TUAK authentication algorithm (e.g. USIM, ISIM or CSIM using Milenage). Only one Algorithm's parameter set should be provided

in a given NAA. If more than one set of parameters is provided in the Profile, the indication of which set of parameters has to be used is out of scope of this specification.

8.4.3 CSIM Parameters PE

This PE is used to set the parameters for the CSIM authentication algorithms CAVE [CAVE]. It may be provided within the context of an ADF_CSIM.

```
PE-CDMAParameter ::= SEQUENCE {
      cdma-header PEHeader,

/* A-Key for CAVE Authentication */
      authenticationKey OCTET STRING (SIZE(8)),

/*
Optional value for ssd
Bytes 1..8: value if shared secret data A
Bytes 9..16: value if shared secret data B
*/
      ssd OCTET STRING (SIZE (16)) OPTIONAL,

/*
   Shared Secrets for HRPD access authentication
   Includes the shared secret data. This field is coded as defined in section
4.5.7.10 HRPD Access Authentication CHAP SS Parameters of [S0016].
*/
      hrpdAccessAuthenticationData OCTET STRING (SIZE (9..32)) OPTIONAL,

/*
   Parameters for simple IP authentication are coded as defined in section 4.5.7.7
SimpleIP CHAP SS Parameters of [S0016].
*/
      simpleIPAuthenticationData OCTET STRING (SIZE (10..483)) OPTIONAL,

/*
   Parameters for mobile IP authentication are coded as defined in section 4.5.7.8
MobileIP SS Parameters of [S0016].
*/
      mobileIPAuthenticationData OCTET STRING (SIZE (19..957)) OPTIONAL
}
```

Usage rules: This PE shall be used once after the creation of a NAA using Cave authentication algorithm (e.g. CSIM). Only one Algorithm's parameters set should be provided in a given NAA. If more than one set of parameters is provided in the Profile Package, the indication of which set of parameters has to be used if out of scope of this specification.

8.5 PIN and PUK codes

8.5.1 Pin Code PE

This PE is used to set the PIN codes related to the MF for the global ones or related to a DF.

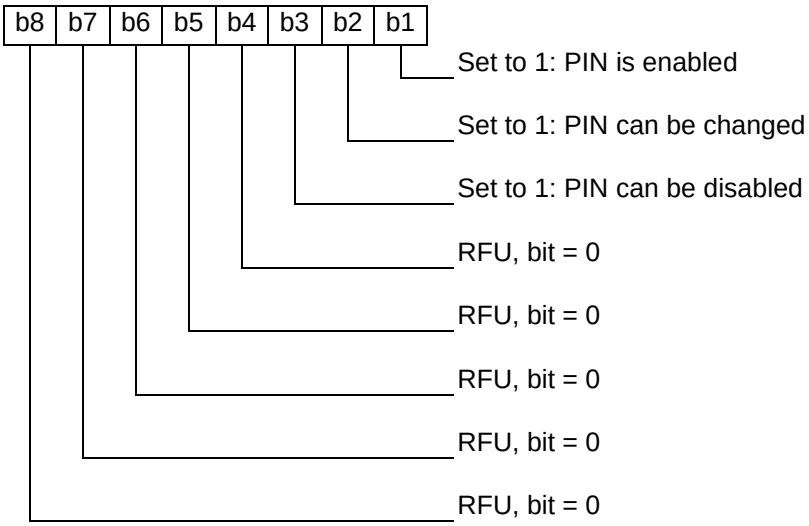
NOTE: Universal PIN is not supported.

```
PINKeyReferenceValue ::= INTEGER {
pinAppl1(1),            -- PIN global of App 1
pinAppl2(2),            -- PIN global of App 2
pinAppl3(3),            -- PIN global of App 3
pinAppl4(4),            -- PIN global of App 4
pinAppl5(5),            -- PIN global of App 5
pinAppl6(6),            -- PIN global of App 6
pinAppl7(7),            -- PIN global of App 7
pinAppl8(8),            -- PIN global of App 8
adm1(10),         -- Administrative Key 1
adm2(11),         -- Administrative Key 2
adm3(12),         -- Administrative Key 3
adm4(13),         -- Administrative Key 4
adm5(14),         -- Administrative Key 5
secondPINAppl1(129),    -- PIN local of App 1
secondPINAppl2(130),    -- PIN local of App 2
secondPINAppl3(131),    -- PIN local of App 3
secondPINAppl4(132),    -- PIN local of App 4
secondPINAppl5(133),    -- PIN local of App 5
secondPINAppl6(134),    -- PIN local of App 6
secondPINAppl7(135),    -- PIN local of App 7
secondPINAppl8(136),    -- PIN local of App 8
adm6(138),        -- Administrative Key 6
adm7(139),        -- Administrative Key 7
adm8(140),        -- Administrative Key 8
adm9(141),        -- Administrative Key 9
adm10(142)        -- Administrative Key 10
}

PINConfiguration ::= SEQUENCE {
      keyReference PINKeyReferenceValue,
      pinValue OCTET STRING (SIZE (8)),
      unblockingPINReference PUKKeyReferenceValue OPTIONAL,
      pinAttributes UInt8 DEFAULT 7,
      maxNumOfAttemps-retryNumLeft UInt8 DEFAULT 51
/* maxNumOfAttemps-retryNumLeft is encoded as follows: max Number of Attempts is
encoded in the high nibble of this value (Bits b8 to b5) and the Number of retry
left is encoded in the low nibble of this value (Bits b4 to b1)*/
}

PE-PINCodes ::= SEQUENCE {
      pin-Header PEHeader,
      pinCodes CHOICE {
            pinconfig SEQUENCE (SIZE (1..26))OF PINConfiguration,
            filePath OCTET STRING /* temporary File ID for ADF, coding according
to section 8.3.5 */
      }
      /* PIN can be either defined in the current context or shared
       with another DF/ADF
       Up to 26 PIN could be defined according to TS 102 221 [102 221]
```

```
        */
}
```

If the RetryNumLeft is greater than MaxNumOfAttemps then the behaviour of the eUICC is undefined.

The coding of the PINAttributes is as follow:



Usage rules: This PE shall be used during the file system creation right after the creation of the MF or right after the creation of the PUK codes if any for global PINs or right after the creation of an ADF or a DF for local PINs. The use of this PE shall be unique in all these contexts.

8.5.2 PUK Code PE

This PE is used to set the PUK codes at the MF level. This PE shall be used during the file system creation right after the creation of the MF. The use of this PE shall be unique.

```
PUKKeyReferenceValue ::= INTEGER {
pukAppl1(1),            -- PUK global of App 1
pukAppl2(2),            -- PUK global of App 2
pukAppl3(3),            -- PUK global of App 3
pukAppl4(4),            -- PUK global of App 4
pukAppl5(5),            -- PUK global of App 5
pukAppl6(6),            -- PUK global of App 6
pukAppl7(7),            -- PUK global of App 7
pukAppl8(8),            -- PUK global of App 8
secondPUKAppl1(129),    -- PUK local of App 1
secondPUKAppl2(130),    -- PUK local of App 2
secondPUKAppl3(131),    -- PUK local of App 3
secondPUKAppl4(132),    -- PUK local of App 4
secondPUKAppl5(133),    -- PUK local of App 5
secondPUKAppl6(134),    -- PUK local of App 6
secondPUKAppl7(135),    -- PUK local of App 7
secondPUKAppl8(136)     -- PUK local of App 8
}
```

```
PUKConfiguration ::= SEQUENCE {
      keyReference PUKKeyReferenceValue,
      pukValue OCTET STRING (SIZE (8)),
      maxNumOfAttemps-retryNumLeft UInt8 DEFAULT 170
/* maxNumOfAttemps-retryNumLeft is encoded as follows: max Number of Attempts is
encoded in the high nibble of this value (Bits b8 to b5) and the Number of retry
left is encoded in the low nibble of this value (Bits b4 to b1)*/
      }

PE-PUKCodes ::= SEQUENCE {
      puk-Header PEHeader,
      pukCodes SEQUENCE (SIZE (1..16))OF PUKConfiguration
      }
```

If the RetryNumLeft is greater than MaxNumOfAttemps then the behavior of the eUICC is undefined.

Usage rules: The PE shall be used only once in the profile Package, right after the creation of the MF.

8.6 Security domains

8.6.1 Security Domain PE

SDs are installed using the `ApplicationInstance` type (As defined in section 8.7.3) which is also used for application installation. The values standardised for Supplementary SDs shall be used.

For the installation of SDs the following PE is defined:

```
PE-SecurityDomain ::= SEQUENCE {
      sd-Header PEHeader,
      instance ApplicationInstance, -- see section 8.7.3
      keyList SEQUENCE (SIZE (1..MAX)) OF KeyObject OPTIONAL, -- see section 8.6.3
      sdPersoData SEQUENCE (SIZE (1..MAX)) OF OCTET STRING OPTIONAL /* see section
8.6.4 */
}
```

Usage rules: The PE shall be used for every SD creation, starting from MNO-SD.

8.6.2 SD and MNO SD Creation

The first SD to be created is the equivalent of the ISD (Issuer Security Domain) of a UICC and is the root of all the other SDs in the hierarchy under this SD, it is called the MNO-SD. It needs to be installed explicitly using "`PE-SecurityDomain`" within the Profile Package. The MNO-SD shall be installed before any other SD, before any RFM Parameters are set or before any applications are created. In addition there may be SSDs which belong to independent SD hierarchies with a self-extradited SSD as root SD.

Since no package AID nor classAID is standardised for the MNO-SD, it may use the values defined for supplementary SD creation in section 3.3.1.1 of [GP CIC]. The eUICC may ignore the values for package AID and class AID provided in the profile for the MNO-SD and may use vendor specific values instead. The first SD within the sequence of the Profile Package will thus be categorized as the MNO-SD by definition and will be installed with the special MNO-SD privileges defined by the GSMA [GS RPT]. The section 3.2 of [GP UC] (secure channel protocol supported by the ISD) applies to the MNO-SD for profiles compliant to GlobalPlatform

Card Specification UICC Configuration. Following instances of SDs will be installed like regular supplementary SDs as known from GlobalPlatform Card Specification [GP CS].

8.6.3 Key Personalisation

After creation of an SD, the keys which shall be installed can be described with the respective SD PE. The parameters are based on the DGIs for personalisation of SDs as specified within the GlobalPlatform Card Specification Amd A [GP AA], section 4.10. The structure has been optimised to avoid redundancy within the data structure.

```
KeyObject::= SEQUENCE {
      keyUsageQualifier [21] OCTET STRING (SIZE (1..2)), /* see [GPCS] section
11.1.9 */
      keyAccess [22] OCTET STRING (SIZE (1)) DEFAULT '00'H,
      keyIdentifier [2] OCTET STRING (SIZE (1)),
      keyVersionNumber [3] OCTET STRING (SIZE (1)),
      keyCounterValue [5] OCTET STRING OPTIONAL,
      keyCompontents SEQUENCE (SIZE (1..MAX)) OF SEQUENCE {
            keyType [0] OCTET STRING,
            keyData [6] OCTET STRING,
            macLength[7] UInt8 DEFAULT 8
      }
}
```

The coding of the following parameters shall follow the GlobalPlatform Card Specification [GP CS] for SCP02 and SCP03 or GlobalPlatform Card Specification UICC Configuration [GP UC] for SCP80 and SCP81:
`keyUsageQualifier` see below.
`keyAccess` see below.
`keyIdentifier`
`keyVersionNumber`

The ETSI specifications do not define access or usage rules for SCP80 and SCP81 keys. Therefore, the `keyAccess` and `keyUsageQualifier` fields shall be ignored when the `KeyObject` transports such keys. For other GP keys `keyUsageQualifier` field may be ignored where the key usage is implicitly defined by key version and index.
Each key to be personalised must be listed only once. This means there shall be no keys with same `keyIdentifier` and `keyVersionNumber` listed twice.

If the `keyCounterValue` is present, it indicates the initial counter associated for that keyset. If it is absent, the initial counter value shall be set according to the default value of the related protocol (e.g. for SCP02 keyset the default value is ‘0000’h, for SCP03 it is ‘000000’h, for SCP80 it is ‘0000000000’h).

NOTE: This field may be ignored for the keys used by the protocols SCP02, SCP03 or SCP80 that mandate an initial counter to be set to the default value right after their initialization.

To simplify the installation of PKI keys, which consist of multiple key components of different types, the `keyCompontents` structure has been defined. This is so that redundant information can be avoided.
Only `keyTypes` defined in GlobalPlatform Card Specification [GP CS], Table 11-16, may be part of the list. Each `keyComponents` shall be specified only once per key (e.g. including two times the same `keyType` within one `KeyObject` will lead to an error).

`macLength`: For AES KID keys, indicates the length of the MAC in bytes as defined in TS 102 226 [102 226]. This value shall be ignored for other key types.

If `keyType` or any other `KeyObject` parameters are not supported by the eUICC, the error code `feature-not-supported` shall be returned and the installation of the Profile Package shall be aborted.

8.6.4 SD Personalisation

Optionally a list of commands may be provided to personalise the SD (e.g. set IIN, change AID, …). Any commands which can be sent via STORE DATA commands addressing the SD personalisation defined by GlobalPlatform Card Specification [GP CS] may be sent to an SD via this means. Only the content of the STORE DATA commands will be provided (excluding CLA, INS, P1, P2, Lc).
The content shall not be encrypted and shall use DGI format. Parameters using TLV format may be included in DGI '0070' as defined by GlobalPlatform Card Specification [GP CS]. Since there is no limitation in terms of content length for within the `sdPersoData` parameter, the complete DGI structure for the SD personalisation shall be sent in one complete byte array. Each DGI shall be provided in its own `sdPersoData` record. Only standardised DGIs, according to GlobalPlatform Card Specification [GP CS], shall be sent when addressing a SD.

Installation of the CASD, if required inside a Profile, uses the same procedure.

8.6.5 RAM / OTA HTTPs Configuration

Within each SD, the settings for RAM and OTA HTTPs can be configured according to GlobalPlatform Card Specification [GP CS] and ETSI specifications. The TAR values for RAM can be configured as follows:

- Bytes 13-15 of the SD instance AID
- TAR List within SD install parameters

The eUICC shall support settings for OTA HTTPs provided within the `sdPersoData` included in DGI '0070' using tag '85' according to GlobalPlatform Amd B [GP AB] (Section 3.7.1 TLV: Security Domain Administration Session Parameters) in the `PE-SecurityDomain` structure of the respective security domain.

The security level for RAM is defined by the MSL parameter of the SD installation parameters. It is highly recommended to assign TAR values to the Security Domains as specified in TS 101 220 [101 220].

The configuration of the PoR (Proof of Receipt) handling is not part of the Profile definition. The eUICC shall follow the latest ETSI and 3GPP release to provide the necessary level of security.

8.7 Application loading and installation

8.7.1 Application PE

For loading and installing applications, the following PE is defined.

```
PE-Application ::= SEQUENCE {
      app-Header PEHeader,
      loadBlock ApplicationLoadPackage OPTIONAL,
      instanceList SEQUENCE (SIZE (1..MAX)) OF ApplicationInstance OPTIONAL
}
```

Within the Application PE, application code can be loaded and instances can be installed and personalised. An example of application is a Java Card™ Applet. The elements are described in more detail in the following. All parameters are optional to cover the following use cases:

- A library shall be loaded: In this case only the library can be provided by specifying the `ApplicationLoadPackage` structure only (no install, no perso)
- A preloaded application shall be installed which only requires an `ApplicationInstance`: Multiple instances of the same ApplicationLoadPackage can be installed within one Application PE.
- An application shall be loaded providing an `ApplicationLoadPackage` object and installed via an `ApplicationInstance` (optionally multiple `ApplicationInstance` objects)

In case the mandatory parameter of the PEHeader object is set to mandatory, profile installation will fail if one of the subsequent elements cannot be executed (e.g. load fails because of API incompatibility, install fails because of duplicate TAR values …). If mandatory is not set, profile installation will continue with the next PE.

The loading procedure may fail for various reasons, including:

- The eUICC does not support the required runtime environment (e.g. Java Card™)
- The required version of the runtime environment is not available
- A library required by the application is not available
- An algorithm required by the application is not available

Within the subsequent sections, the elements for the PE are described in more detail.

Usage rules: This PE shall be used after the security domain to which the application instance is associated to is created by using PE-SecurityDomain.

8.7.2 ApplicationLoadPackage

The ApplicationLoadPackage parameter includes the application code. It is based on the LOAD command according to GlobalPlatform Card Specification [GP CS]. The only difference to the GP Load Command is that the complete load block is provided within the `loadBlockObject` parameter.

```
ApplicationLoadPackage ::= SEQUENCE {
      loadPackageAID [APPLICATION 15] ApplicationIdentifier,
      securityDomainAID [APPLICATION 15] ApplicationIdentifier OPTIONAL,
      nonVolatileCodeLimitC6 [PRIVATE 6] OCTET STRING OPTIONAL,
      volatileDataLimitC7 [PRIVATE 7] OCTET STRING OPTIONAL,
      nonVolatileDataLimitC8 [PRIVATE 8] OCTET STRING OPTIONAL,
      hashValue [PRIVATE 1] OCTET STRING OPTIONAL,
      loadBlockObject [PRIVATE 4] OCTET STRING
}
```

The following parameters based on the INSTALL command according to GlobalPlatform Card Specification [GP CS] may be ignored by the eUICC in case they are not supported.

`nonVolatileCodeLimitC6`
`volatileDataLimitC7`
`nonVolatileDataLimitC8`
`hashValue`

All the other parameters except `securityDomainAID` are mandatory and need to follow the same rules as defined for the LOAD command as defined in GlobalPlatform Card Specification [GP CS].
In case no value for the optional parameter `securityDomainAID` is provided, the package will be associated to the MNO-SD by default.

8.7.3 ApplicationInstance

The ApplicationInstance is used to instantiate and personalise applications. It is based on the GlobalPlatform Card Specification [GP CS] INSTALL command. To simplify and optimise the process of personalisation, additional parameters have been added which will be described in this section.

```
ApplicationInstance ::= SEQUENCE {
      applicationLoadPackageAID [APPLICATION 15] ApplicationIdentifier,
      classAID [APPLICATION 15] ApplicationIdentifier,
      instanceAID [APPLICATION 15] ApplicationIdentifier,
      extraditeSecurityDomainAID [APPLICATION 15] ApplicationIdentifier OPTIONAL,
      applicationPrivileges [2] OCTET STRING,
      lifeCycleState [3] OCTET STRING (SIZE(1)) DEFAULT '07'H,
      /* Coding according to GP Life Cycle State.  */

      applicationSpecificParametersC9 [PRIVATE 9] OCTET STRING,
      systemSpecificParameters [PRIVATE 15] ApplicationSystemParameters OPTIONAL,
      applicationParameters [PRIVATE 10] UICCApplicationParameters OPTIONAL,
      processData SEQUENCE (SIZE (1..MAX)) OF OCTET STRING OPTIONAL
}

ApplicationSystemParameters ::= SEQUENCE{
      volatileMemoryQuotaC7 [PRIVATE 7] OCTET STRING OPTIONAL,
      nonVolatileMemoryQuotaC8 [PRIVATE 8] OCTET STRING OPTIONAL,
      globalServiceParameters [PRIVATE 11] OCTET STRING OPTIONAL,
      implicitSelectionParameter [PRIVATE 15] OCTET STRING OPTIONAL,
      volatileReservedMemory [PRIVATE 23] OCTET STRING OPTIONAL,
      nonVolatileReservedMemory [PRIVATE 24] OCTET STRING OPTIONAL,
      ts102226SIMFileAccessToolkitParameter [PRIVATE 10] OCTET STRING OPTIONAL,
      ts102226AdditionalContactlessParameters [0]
TS102226AdditionalContactlessParameters OPTIONAL,
      contactlessProtocolParameters [PRIVATE 25] OCTET STRING OPTIONAL, /* Coded
according to Contactless Protocol Parameters Structure as defined in GP Amd. C
*/
      userInteractionContactlessParameters [PRIVATE 26] OCTET STRING OPTIONAL
/* Coded according to User Interaction Parameters Structure as defined in GP
Amd. C */
}

UICCApplicationParameters ::= SEQUENCE {
      uiccToolkitApplicationSpecificParametersField [0] OCTET STRING OPTIONAL,
      uiccAccessApplicationSpecificParametersField [1] OCTET STRING OPTIONAL,
      uiccAdministrativeAccessApplicationSpecificParametersField [2] OCTET STRING
OPTIONAL
}

TS102226AdditionalContactlessParameters ::= SEQUENCE{
      protocolParameterData   OCTET   STRING   /*   Parameters   for   contactless
applications encoded according to TS 102 226 */
}
```

The coding of the following parameters for the `ApplicationInstance` shall follow the coding defined for Install for Install defined by GlobalPlatform Card Specification [GP CS]:

```
applicationLoadPackageAID
classAID
instanceAID
applicationPrivileges
applicationSpecificParametersC9
systemSpecificParameters
```

Providing a SD AID within `extraditeSecurityDomainAID` has the same effect as the Install for Extradition command (GlobalPlatform Card Specification [GP CS]). In case no value for the optional parameter `extraditeSecurityDomainAID` is provided, the instance will be associated to the MNO-SD by default. An application (or SD) shall only be associated to an SD in Life Cycle State PERSONALIZED. In case of an association to an SD in a Life Cycle State different from PERSONALIZED, the error code `invalid-parameter` shall be returned and the installation of the Profile Package shall be aborted. For the MNO-SD instance, no value shall be provided for the `extraditeSecurityDomainAID` parameter: the MNO-SD is associated with itself and it is not subject to extradition, as indicated in the GlobalPlatform Card Specification [GP CS].

The `lifeCycleState` parameter has the same encoding as the Life Cycle State defined within GlobalPlatform Card Specification [GP CS] (section 11.1.1 Life Cycle Coding). For application instances, coding is according to "Table 11-4 Application Life Cycle Coding"; for SDs according to "Table 11-5 Security Domain Life Cycle Coding". If no value is provided the default is INSTALLED & SELECTABLE. For an SD the content of the `lifeCycleState` parameter shall comply with the state of its personalised keys, i.e. it shall be set to PERSONALIZED, when at least one Secure Channel Key Set for one of the supported Secure Channel Protocols and all the keys required by its privileges are provided. Otherwise it shall be set to INSTALLED & SELECTABLE. The behaviour of the eUICC is undefined in case the Life Cycle State contradicts its state of personalised keys.

Initial Contactless Activation State, if any, is provided inside the `contactlessProtocolParameters`.

With `applicationParameters` the ETSI TS 102 226 [102 226] install parameters can be provided to define the access domain for an application. Coding follows the same rules as specified within the referenced documents.
Interpretation of MSL (Minimum Security Level) shall follow the rules defined within ETSI TS 102 226 [102 226] for all applications.

Each Applet Instance can be personalised separately. The same means as for STORE DATA are used to personalise an application instance. All byte strings provided within `processData` will be directly sent to the respective application instance for processing through the "processData" method of the "Application" or "Personalization" interface of the application. The content of the `processData` is specific to the implementation of the application. It should contain all the bytes contained in a STORE DATA command (Including CLA, INS, P1, P2, L) if required by the application but encryption shall not be used. Any data may be sent. Processing is solely up to the application itself. Data will be sent as is to the application for processing. No decryption will be performed by the respective SD. If the application does not implement the "processData" method, the whole PE should be discarded.

8.8 RFM Parameters

This PE is used to set the parameters related to RFM.

```
PE-RFM ::= SEQUENCE {
      rfm-header [0] PEHeader,
```

```
      /* instanceAID
      AID of the RFM instance
      */
      instanceAID [APPLICATION 15] ApplicationIdentifier,

      /* securityDomainAID to which the RFM instance is associated
      */
      securityDomainAID [APPLICATION 15] ApplicationIdentifier OPTIONAL,

      tarList [0] SEQUENCE (SIZE (1..MAX)) OF OCTET STRING (SIZE(3)) OPTIONAL,

      minimumSecurityLevel [1] OCTET STRING (SIZE (1)),

      uiccAccessDomain OCTET STRING,
      uiccAdminAccessDomain OCTET STRING,

      /*
       If the following parameter is available the respective ADF will be the
directory selected by default within an RFM script. In case it is not available
the MF will be the default selection.
        */
      adfRFMAccess ADFRFMAccess OPTIONAL
}

ADFRFMAccess ::= SEQUENCE {
      adfAID ApplicationIdentifier,
      adfAccessDomain OCTET STRING,
      adfAdminAccessDomain OCTET STRING
}
```

Usage rules: This PE can be used several times in the Profile Package after the PE containing the SD and the PE containing the ADF.

The following parameters for RFM can be configured:

`instanceAID`
Indicates the AID of the RFM instance

`securityDomainAID`
References the SD to which the RFM application is associated. If not provided, it will automatically be associated to the MNO-SD.

`tarList` / Definition of TAR values for RFM instance
In case one or multiple TAR values for use with SCP80 shall be assigned to the RFM instance, it is possible to define TAR addresses for each RFM instance in the following way:
In case `tarList` is provided the TAR values are taken from this list. `tarList` shall include at least one TAR if available. In case `tarList` is not available the TAR value defined within bytes 13-15 of the `instanceAID` is used.
The specification of a TAR value is optional but, if absent, the RFM instance cannot be addressed via protocols that require TAR address (e.g. SCP80).

`minimumSecurityLevel`
Define the Minimum Security Level (MSL) for the RFM instance. Interpretation of MSL shall follow the rules defined within ETSI TS 102 226 [102 226].

`uiccAccessDomain`
Access domain of the RFM instance within the MF. Coded according to ETSI TS 102 226 [102 226]. Allows the definition of access rights granted to the RFM application allowing it to perform non administrative operations on MF file system.

`uiccAdminAccessDomain`
Administrative access domain of the RFM instance within the MF. Coded according to ETSI TS 102 226 [102 226]. Allows the definition of access rights granted to the RFM application allowing it to perform administrative operations on MF file system.

`adfRFMAccess`
To address ADFs via RFM, each RFM instance can be associated with one ADF. This optional parameter links the RFM instance to the given ADF. When processing an RFM script, the defined ADF will be selected by default and can be addressed by the file path '7FFF' as it is defined within ETSI standards.
In case this optional parameter is not provided, the RFM instance will be linked only to the MF which will be the default selection in the context of an RFM script.

`adfAID`
AID of the ADF to link to the RFM instance.

`adfAccessDomain`
Access domain of the RFM instance within the referenced ADF. Coded according to ETSI TS 102 226 [102 226]. Allows the definition of access rights granted to the RFM application allowing it to perform non administrative operations on ADF files.

`adfAdminAccessDomain`
Administrative access domain of the RFM instance within the referenced ADF. Coded according to ETSI TS 102 226 [102 226]. Allows the definition of access rights granted to the RFM application allowing it to perform administrative operations on ADF files.

8.9 Non standardised content

This PE is used to send content that can only be processed by specific eUICCs. This content can be either a proprietary element or content standardised in a specification after eUICC creation. The Profile Package can use as many PEs of this type as required.

```
PE-NonStandard ::= SEQUENCE {
      nonStandard-header PEHeader,
      issuerID OBJECT IDENTIFIER,
      content OCTET STRING
}
```

Usage rules: The PE can be used at any place in the Profile Package.

8.10 Profile Package end

This PE is used to indicate the end of the Profile Package to the eUICC.

```
PE-End ::= SEQUENCE {
```

```
      end-header PEHeader
}
```

Usage rules: The PE shall be used as the last element of the Profile Package.

8.11 eUICC Response type

The eUICC response type is defined in the following ASN.1 type definition:

```
PEStatus ::= SEQUENCE {
status INTEGER {
      ok(0), pe-not-supported(1), memory-failure(2),bad-values(3),
      not-enough-memory(4),invalid-request-format(5), invalid-parameter(6),
      runtime-not-supported (7), lib-not-supported (8),
       template-not-supported (9), feature-not-supported (10),
       unsupported-profile-version(31)
      /* ISO 7816 standard status values apply in the range of [24576...28671]
       and [36864...40959] for reporting status values '6xxx'H and '9xxx'H
      proprietary values apply in the range [40960...65535]
      */
      },
identification UInt15 OPTIONAL,
-- Identification number of the PE triggering the error
additional-information UInt8 OPTIONAL
-- Additional information related to the status code
}

EUICCResponse ::= SEQUENCE {
      peStatus SEQUENCE OF PEStatus,
      profileInstallationAborted NULL OPTIONAL,
      statusMessage UTF8String OPTIONAL
}

END
```

The eUICC response may contain several `PEStatus` data objects corresponding to PEs generating different status messages except when all data objects in the PE have been processed successfully. The eUICC may group the status messages into one `EUICCResponse` sent after receiving the `PE-End`, or right after the processing of a PE leading to the abortion of the Profile Package installation. The eUICC may also send several `EUICCResponse` when there is something to report on a specific PE even if the installation process is not aborted. In case there is nothing to report and only in this case, one `EUICCResponse` containing the `ok` status code shall be sent by the eUICC at the end of the Profile Package installation.

The `status` can take the following values:

- `ok`: used at the end of the Profile download and installation in order to indicate that the Profile has been successfully processed by the eUICC. This status shall not be sent for all the PEs but only at the end of the Profile installation. When using this status code, the eUICC shall not indicate any identification of a PE.
- `PE-not-supported`: indicates that a specific PE identified by its identification number is not supported by the eUICC. If this PE is indicated as "mandated" in the PE header, this status is an error status and the processing of the Profile is aborted. Otherwise this is just a warning and the installation of the Profile shall continue.

- `memory-failure`: indicates a failure during the installation of the Profile due to internal memory issue. This status is an error status and the processing of the Profile is aborted.
- `bad-values`: indicates that a least one value in the PE identified by its identification number is out of acceptable value range. If the PE generating this status indicates "mandated" in the PE header and the eUICC cannot apply a default value, this status is an error status and the processing of the Profile shall be aborted. Otherwise this is just a warning and the installation of the Profile shall continue.
- `not-enough-memory`: indicates that the eUICC does not have enough free memory to install the Profile. This status is an error status and the processing of the Profile is aborted.
- `invalid-request-format`: indicates that the order of the PEs is invalid or a structure in a PE is unknown or badly formatted. It is not required that the eUICC is able to detect and reject all the incorrect order of the PEs or all invalid formats. If the eUICC cannot recover the error by ignoring some non-mandatory parts of the Profile or for any other reason, the installation of the Profile may be aborted.
- `invalid-parameter`: indicates that a parameter in a PE description is not supported. This status code shall be used when the eUICC encounters an unknown tag inside a PE. If the PE generating this status indicates "mandated" in the PE header and the eUICC cannot ignore this parameter, this status is an error status and the processing of the Profile is aborted. Otherwise this is just a warning, the installation of the Profile shall continue and the parameter shall be ignored.
- `runtime-not-supported`: indicates that the runtime environment required by the application present in a `PE-Application` is not supported by the eUICC. If the PE generating this status indicates "mandated" in the PE header, this status is an error status and the processing of the Profile is aborted. Otherwise this is just a warning, the installation of the Profile shall continue and the application shall be ignored.
- `lib-not-supported`: indicates that a library required by the application present in a `PE-Application` is not available in the eUICC. If the PE generating this status indicates "mandated" in the PE header, this status is an error status and the processing of the Profile is aborted. Otherwise this is just a warning, the installation of the Profile shall continue and the application shall be ignored.
- `template-not-supported`: indicates that the template indicated by the OBJECT IDENTIFIER in the `templateID` or in the `eUICC-Mandatory-GFSTEList` is not available in the eUICC (i.e. non-standard template or template version not supported). If the `templateID` is inside a PE indicated as "mandated" or if the OID is in the `eUICC-Mandatory-GFSTEList` of the Profile Header, this status is an error status and the processing of the Profile is aborted. Otherwise this is just a warning, the installation of the Profile shall continue and the file system described by this PE shall not be created.
- `feature-not-supported`: indicates that a feature included in the PE or in the `ServicesList` of the Profile Header is not supported by the eUICC. If the PE generating this status indicates "mandated" in the PE header or if this feature is included into the `ServiceList` of the Profile Header, this status is an error status and the processing of the Profile is aborted. Otherwise this is just a warning, the installation of the Profile shall continue and the feature shall be ignored.
- `unsupported-profile-version`: indicates that the major version indicated in the Profile header is not supported by this eUICC. This status is an error status and the processing of the Profile shall be aborted.

The optional tag `profileInstallationAborted` indicates that the installation of the Profile is aborted due to an error specified in the `status` field. When this tag is used, it shall be present in the last EUICCResponse sent by the eUICC.
The optional `statusMessage` can be used in order to give additional information.

For the PEs defined in this specification, the following table identifies the possible status that can be used.

	ok	PE-not-supported	memory-failure	bad-values	not-enough-memory	invalid-request-format	invalid-parameter	runtime-not-supported	lib-not-supported	template-not-supported	feature-not-supported	unsupported-profile-version
Profile Header	*		A	A	A	A	A	A		A	A	A
MF	*		A	O	A	O	O			A		
DF CD	*		A	O	A	O	O			A	C	
DF TELECOM	*		A	O	A	O	O			A	C	
USIM	*	C(1)	A	O	A	O	O			C	C	
OPT USIM	*	C(1)	A	O	A	O	O			C	C	
ISIM	*	C(1)	A	O	A	O	O			C	C	
OPT ISIM	*	C(1)	A	O	A	O	O			C	C	
CSIM	*	C(1)	A	O	A	O	O			C	C	
OPT CSIM	*	C(1)	A	O	A	O	O			C	C	
Generic File Management	*		A	O	A	O	O				C	
AKA Parameters	*	C(1)	A	O		A	A				A	
CSIM Parameters	*	C(1)	A	O		A	A				A	
PIN Code	*		A	O		A	A					
PUK Code	*		A	O		A	A					
Security Domain	*		A	O	A	A	A				C	
Application	*		A	O	A	O	O	C	C		C	
RFM Parameters	*		A	O		A	A				C	
Non Standardized	*	C	A	O	A	O	O	C	C		C	
End	*		A		A							

*: This status is allowed for this PE and this shall not abort the Profile Package installation.
A: This status is allowed for this PE and this shall abort the Profile Package installation.
C: This status is allowed for this PE and this shall abort the Profile Package installation if "mandated" is set in the PE header. Otherwise, this is just a warning, the PE is skipped, and the installation of the profile shall continue.
O: This status is allowed for this PE and the eUICC may abort the Profile Package installation (See status description for conditions).

NOTE 1: The use of `PE-not-supported` is allowed only for eUICC not supporting the feature related to the PE (e.g. if CSIM is not supported by the eUICC, and only in that case, this error code is allowed for PE-CSIM).

9. ANNEX A (Normative): File Structure Templates Definition

9.1 Templates rules and usage

The goal of the templates defined below is to reduce the size of the Profile Package by providing a data compression mechanism. Only the differences between the content and parameters of the files required for a specific Profile, and the content and parameters provided by these templates, need to be included in the Profile Package. Additional templates, along with their management instructions, may be defined later.

Table column information:

- FID: File Identifier.
- File Type: TR= Transparent, LF= Linear Fixed, CY= Cyclic, BER-TLV= BER-TLV coded files, MF= Master File, DF= Dedicated File.
- NB Rec: Number of records in the files for LF or CY files.
- File/Rec Size: File size for TR and BER-TLV files, Record size for LF and CY files.
- Access Rules: reference to the access rules combination defined in the corresponding EF-ARR.
- Default Value: Fill pattern describing the default file or record content unless it is specified as repeat pattern. For record based files, if the number of record is changed, the pattern will be used for all the records. For transparent files using repeat pattern, if the file size is changed, the repeat pattern will be used for the complete file.
- Content Required: If Yes, indicates that there is no default content for the file defined in the template and that it shall be provided when referencing the template in the Profile Package.
- Parameters: Indicates the parameters that shall be provided when referencing the template in the Profile Package in addition to those listed in section 8.3.3.

NOTE: In order to fully benefit from the definition of the templates, only the field listed in the column "parameters" shall be provided in the FILE type, nevertheless section 8.3.3 details the template modification rules when the default value does not fit with the Profile needed by the MNO.

9.2 Files at MF level

This Template is a "Created by default" type template. This template shall be supported by the eUICCs.
Template OID: {joint-iso-itu-t(2) international-organizations(23) simalliance(143) euicc-profile(1) template(2) mf(1)}

FID	EF Name	File Type	NB Rec.	File / Rec Size	Access Rules	Default Value	Content Required	Parameters
3F00	MF	MF			14			pinStatusTemplateDO
2F05	EF PL	TR	NA	2	1	FF...FF	N	
2FE2	EF ICCID	TR	NA	10	11		Y	
2F00	EF DIR	LF	X	X	10		Y	Record size, File size
2F06	EF ARR	LF	X	X	10		Y	Record size, File size
2F08	EF UMPC	TR	NA	5	10	eUICC Platform dependant	N See Note	
NOTE: Only the second byte of this file may be changed. Modification of any other bytes may be ignored by the eUICC.								

9.3 DF CD

This Template is a "Not created by default" type template. This template shall be supported by the eUICCs.
Template OID: {joint-iso-itu-t(2) international-organizations(23) simalliance(143) euicc-profile(1) template(2) cd(2)}

FID	EF Name	File Type	NB Rec.	File / Rec Size	Access Rules	Default Value	Content Required	Parameters
7F11	DF CD	DF			14			pinStatusTemplateDO
6F01	EF LAUNCHPAD	TR	NA	X	2	FF...FF	Y	Size
6F40 to 6F7F	EF ICON	TR	NA	X	2	FF...FF	Y	Size

9.4 DF TELECOM

This Template is a "Not created by default" type template. This template shall be supported by the eUICCs.

Template OID: `{joint-iso-itu-t(2) international-organizations(23) simalliance(143) euicc-profile(1) template(2) telecom(3)}`

Ass. Serv.	FID	EF Name	File Type	NB Rec.	File / Rec Size	Access Rules	Default Value	Content Required	Parameters
	7F10	DF TELECOM	DF			14			pinStatusTemplateDO
	6F06	EF ARR	LF	X	X	10		Y	Record size, File size
	6F53	EF RMA	LF	X	X	3		Y	Record size, File size
	6F54	EF SUME	TR	NA	22	3		Y	Size
	6FE0	EF ICE DN	LF	50	24	9	FF…FF	N	
	6FE1	EF ICE FF	LF	X	X	9	FF…FF	N	Record size, File size
12 and 91	6FE5	EF PSISMSC	LF	X	X	5	-	Y	Record size, File size
	5F50	DF GRAPHICS	DF			14			pinStatusTemplateDO
	4F20	EF IMG	LF	X	X	2	00 FF...FF	N	Record size, File size
	4F40 to 4F7F	EF IIDF	TR	NA	X	2	FF…FF	N	Size
	4F21	EF ICE GRAPHICS	BER-TLV	NA	X	9	FF…FF	N	Size
	4F01	EF LAUNCH SCWS	TR	NA	X	10		Y	Size
	4F80 to 4FBF	EF ICON	TR	NA	X	10		Y	Size
	5F3A	DF PHONEBOOK	DF			14			pinStatusTemplateDO
	4F30	EF PBR	LF	X	X	2		Y	Record size, File size
	4F38 to 4F3F	EF EXT1	LF	X	13	5	00 FF … FF	N	File size
	4F40 to 4F47	EF AAS	LF	X	X	5	FF…FF	N	Record size, File size
	4F48 to 4F4F	EF GAS	LF	X	X	5	FF…FF	N	Record size, File size
	4F22	EF PSC	TR	NA	4	5	00 00 00 00	N	
	4F23	EF CC	TR	NA	2	5	00 00	N	
	4F24	EF PUID	TR	NA	2	5	00 00	N	
	4F50 to 4F57	EF IAP	LF	X	X	5	FF…FF	N	Record size, File size

	4F58 to 4F5F	EF ADN	LF	X	X	5	FF…FF	N	Record size, File size
	4F60 to 4F67	EF PBC	LF	X	2	5	00…00	N	File size
	4F68 to 4F6F	EF ANR	LF	X	X	5	FF…FF	N	Record size, File size
	4F70 to 4F77	EF PURI	LF	X	X	5		Y	Record size, File size
	4F78 to 4F7F	EF EMAIL	LF	X	X	5	FF…FF	N	Record size, File size
	4F80 to 4F87	EF SNE	LF	X	X	5	FF…FF	N	Record size, File size
	4F88 to 4F8F	EF UID	LF	X	2	5	00 00	N	File size
	4F90 to 4F97	EF GRP	LF	X	X	5	00…00	N	Record size, File size
	4F98 to 4F9F	EF CCP1	LF	X	X	5	FF…FF	N	Record size, File size
67	5F3B	DF MULTIMEDIA	DF			14			pinStatusTemplateDO
67	4F47	EF MML	BER-TLV	NA	X	5	FF…FF	N	Size
67	4F48	EF MMDF	BER-TLV	NA	X	5	FF…FF	N	Size
	5F3C	DF MMSS	DF			14			pinStatusTemplateDO
	4F20	EF MLPL	TR	NA	X	2		Y	Size
	4F21	EF MSPL	TR	NA	X	2		Y	Size
	4F22	EF MMSSMODE	TR	NA	1	2		Y	

Files with high update activity in this template are: EF CC and EF PUID.

9.5 USIM

9.5.1 Mandatory USIM EFs

This Template is a "Created by default" type template. This template shall be supported by the eUICCs supporting the USIM application.

Template OID: `{joint-iso-itu-t(2) international-organizations(23) simalliance(143) euicc-profile(1) template(2) usim(4)}`

Ass. Serv.	FID	EF Name	File Type	NB Rec.	File / Rec Size	Access Rules	Default Value	Content Required	Parameters
	XXXX	ADF USIM	ADF			14			AID, Temporary FID, pinStatusTemplateDO
	6F07	EF IMSI	TR	NA	9	2		Y	
	6F06	EF ARR	LF	X	X	10		Y	Record size, File size
	6F08	EF Keys	TR	NA	33	5	07FF…FF	N	
	6F09	EF KeysPS	TR	NA	33	5	07FF…FF	N	
	6F31	EF HPPLMN	TR	NA	1	2	0A	N	
	6F38	EF UST	TR	NA	14	2		Y	
2 or 89	6F3B	EF FDN	LF	20	26	8	FF…FF	N	
10	6F3C	EF SMS	LF	10	176	5	00 FF…FF	N	
12	6F42	EF SMSP	LF	1	38	5	FF...FF	N	
10	6F43	EF SMSS	TR	NA	2	5	FF FF	N	
19	6F46	EF SPN	TR	NA	17	10		Y	
2,6,34 or 35	6F56	EF EST	TR	NA	1	8		Y	
	6F5B	EF START-HFN	TR	NA	6	5	F00000 F00000	N	
	6F5C	EF THRESHOLD	TR	NA	3	2	FF FF FF	N	
	6F73	EF PSLOCI	TR	NA	14	5	FF FF FF FF FF FF FF FF FF FF 00 00 FF 01	N	
	6F78	EF ACC	TR	NA	2	2		Y	
	6F7B	EF FPLMN	TR	NA	12	5	FF…FF	N	
	6F7E	EF LOCI	TR	NA	11	5	FFFFFFFFFFFFFF0000 FF 01	N	
	6FAD	EF AD	TR	NA	4	10	00 00 00 02	N	

	6FB7	EF ECC	LF	1	4	10		Y	
	6FC4	EF NETPAR	TR	NA	128	5	FF…FF	N	
85	6FE3	EF EPSLOCI	TR	NA	18	5	FFFFFFFFFFFFFFFFFFFFFFFF FFFFFF0000 01	N	
85	6FE4	EF EPSNSC	LF	1	80	5	FF…FF	N	

Files with high update activity in this template are: EF Keys, EF KeysPS, EF START-HFN, EF PSLOCI, EF LOCI, EF NETPAR, EF EPSLOCI and EF EPSNSC.

9.5.2 Optional USIM EFs

This Template is a "Not created by default" type template. This template shall be supported by the eUICCs supporting the USIM application.
Template OID: `{joint-iso-itu-t(2) international-organizations(23) simalliance(143) euicc-profile(1) template(2) opt-usim(5)}`

Ass. Serv.	FID	EF Name	File Type	NB Rec.	File / Rec Size	Access Rules	Default Value	Content Required	Parameters
	6F05	EF LI	TR	NA	6	1	FF...FF	N	
13	6F37	EF ACMmax	TR	NA	3	5	000000	N	
13	6F39	EF ACM	CY	1	3	7	000000	N	
17	6F3E	EF GID1	TR	NA	8	2		Y	
18	6F3F	EF GID2	TR	NA	8	2		Y	
21	6F40	EF MSISDN	LF	1	24	2	FF...FF	N	
13	6F41	EF PUCT	TR	NA	5	5	FFFFFF0000	N	
15	6F45	EF CBMI	TR	NA	10	5	FF...FF	N	
19	6F48	EF CBMID	TR	NA	10	2	FF...FF	N	
4 or 89	6F49	EF SDN	LF	10	24	2	FF...FF	N	
3	6F4B	EF EXT2	LF	10	13	8	00 FF...FF	N	
5	6F4C	EF EXT3	LF	10	13	2	00 FF...FF	N	
16	6F50	EF CBMIR	TR	NA	20	5	FF...FF	N	
20	6F60	EF PLMNwAct	TR	NA	40	5	Repeat FFFFFF0000	N	
42	6F61	EF OPLMNwAcT	TR	NA	40	2	Repeat FFFFFF0000	N	

43	6F62	EF HPLMNwAcT	TR	NA	5	2	Repeat FFFFFF0000	N	
36	6F2C	EF DCK	TR	NA	16	5	FF...FF	N	
37	6F32	EF CNL	TR	NA	30	2	FF...FF	N	
11	6F47	EF SMSR	LF	10	30	5	00 FF...FF	N	
6	6F4D	EF BDN	LF	10	25	8	FF...FF	N	
44	6F4E	EF EXT5	LF	10	13	5	00 FF...FF	N	
14	6F4F	EF CCP2	LF	5	15	5	FF...FF	N	
7	6F55	EF EXT4	LF	10	13	8	00 FF...FF	N	
35	6F57	EF ACL	TR	NA	101	8	00 FF…FF	N	
6	6F58	EF CMI	LF	10	11	2	FF...FF	N	
9	6F80	EF ICI	CY	20	38	5	FF…FF 000000 00 01FFFF	N	
8	6F81	EF OCI	CY	20	37	5	FF…FF 000000 01FFFF	N	
9	6F82	EF ICT	CY	1	3	7	000000	N	
8	6F83	EF OCT	CY	1	3	7	000000	N	
57	6FB1	EF VGCS	TR	NA	20	2		Y	
57	6FB2	EF VGCSS	TR	NA	7	5		Y	
58	6FB3	EF VBS	TR	NA	20	2		Y	
58	6FB4	EF VBSS	TR	NA	7	2		Y	
24	6FB5	EF eMLPP	TR	NA	2	2		Y	
25	6FB6	EF AaeM	TR	NA	1	5	00	N	
	6FC3	EF HiddenKey	TR	NA	4	5	FF…FF	N	
45	6FC5	EF PNN	LF	10	16	10		Y	
46	6FC6	EF OPL	LF	5	18	10		Y	
47	6FC7	EF MBDN	LF	3	24	5		Y	
47	6FC8	EF EXT6	LF	10	13	5	00 FF...FF	N	
47	6FC9	EF MBI	LF	10	5	5		Y	
48	6FCA	EF MWIS	LF	10	6	5	00 00 00 00 00	N	
49	6FCB	EF CFIS	LF	10	16	5	01 00 FF...FF	N	
49	6FCC	EF EXT7	LF	10	13	5	00 FF...FF	N	
51	6FCD	EF SPDI	TR	NA	17	2		N	
52	6FCE	EF MMSN	LF	10	6	5	00 00 00 FF…FF	N	

53	6FCF	EF EXT8	LF	10	13	5	00 FF...FF	N	
52	6FD0	EF MMSICP	TR	NA	100	2	FF…FF	N	
52	6FD1	EF MMSUP	LF	X	X	5	FF…FF	N	Record size, File size
52 or 55	6FD2	EF MMSUCP	TR	NA	100	5	FF…FF	N	
56	6FD3	EF NIA	LF	5	11	2	FF...FF	N	
64	6FD4	EF VGCSCA	TR	NA	X	2	00…00	N	Size
65	6FD5	EF VBSCA	TR	NA	X	2	00…00	N	Size
68	6FD6	EF GBABP	TR	NA	X	5	FF…FF	N	Size
69	6FD7	EF MSK	LF	X	X	2	FF...FF	N	Record size, File size
69	6FD8	EF MUK	LF	X	X	2	FF...FF	N	Record size, File size
71	6FD9	EF EHPLMN	TR	NA	15	2	FF…FF	N	
68	6FDA	EF GBANL	LF	X	X	2	FF...FF	N	Record size, File size
71 or 73	6FDB	EF EHPLMNPI	TR	NA	1	2	00	N	
74	6FDC	EF LRPLMNSI	TR	NA	1	2	00	N	
68 or 76	6FDD	EF NAFKCA	LF	X	X	2	FF…FF	N	Record size, File size
78	6FDE	EF SPNI	TR	NA	X	10	00 FF…FF	N	Size
79	6FDF	EF PNNI	LF	X	X	10	00 FF…FF	N	Record size, File size
80	6FE2	EF NCP-IP	LF	X	X	2		Y	Record size, File size
	6FE6	EF UFC	TR	NA	30	10	80 1E 60 C0 1E 90 00 80 04 00 00 00 00 00 00 00 00 F0 00 00 00 00 40 00 00 00 00 00 00 80	N	
96	6FE8	EF NASCONFIG	TR	NA	18	2		Y	
95	6FE7	EF UICCIARI	LF	X	X	2		Y	Record size, File size
97	6FEC	EF PWS	TR	NA	X	10		Y	Size
2 or 99	6FED	EF FDNURI	LF	X	X	8	FF…FF	N	Record size, File size
6 or 99	6FEE	EF BDNURI	LF	X	X	8	FF…FF	N	Record size, File size
4 or 99	6FEF	EF SDNURI	LF	X	X	2	FF…FF	N	Record size, File size
108	6FF0	EF IWL	LF	X	X	3		Y	Record size, File size
108	6FF1	EF IPS	CY	X	4	10	FF…FF	N	File size

108	6FF2	EF IPD	LF	X	X	3	FF…FF	N	Record size, File size

Files with high update activity in this template are: EF ACM, EF ICI, EF OCI, EF ICT, EF OCT, EF MWIS, EF MSK, EF IPS and EF IPD.

Files EF GBABP, EF MSK, EF MUK, EF GBANL and EF NAFKCA are associated with services which require support at the eUICC operating system level. So, even if indicated in the profile, the creation of these files shall be skipped by the eUICC if these functionalities are not supported by the eUICC framework. In that case, the eUICC shall answer to the Profile Creator with a status code set to "`feature-not-supported`" with "`additional-information`" set to '1' if GBA is not supported, to '2' if MBMS is not supported and '3' if both are not supported. The bits related to these services in the EF UST shall also be cleared if not supported by the eUICC. If "mandated" is set in the PE header, the installation of the Profile shall be aborted.

9.5.3 DF Phonebook

The template for DF Phonebook at the USIM level is the same as the template for DF Phonebook at the DF Telecom level. This Template is a "Not created by default" type template. This template shall be supported by the eUICCs.

Template OID: `{joint-iso-itu-t(2) international-organizations(23) simalliance(143) euicc-profile(1) template(2) phonebook(6)}`

9.5.4 DF GSM-ACCESS

This Template is a "Not created by default" type template. This template shall be supported by the eUICCs supporting the USIM application.
Template OID: `{joint-iso-itu-t(2) international-organizations(23) simalliance(143) euicc-profile(1) template(2) gsm-access(7)}`

Ass. Serv.	FID	EF Name	File Type	NB Rec.	File / Rec Size	Access Rules	Default Value	Content Required	Parameters
27	5F3B	DF GSM-ACCESS	DF			14			pinStatusTemplateDO
27	4F20	EF Kc	TR	NA	9	5	FF FF FF FF FF FF FF FF 07	N	
27	4F52	EF KcGPRS	TR	NA	9	5	FF FF FF FF FF FF FF FF 07	N	
39	4F63	EF CPBCCH	TR	NA	10	5	FF..FF	N	
40	4F64	EF InvScan	TR	NA	1	2	00	N	

Files with high update activity in this template are: EF Kc, EF KcGPRS and EF CPBCCH.

9.5.5 DF MexE

There is no template currently defined for this DF.

9.5.6 DF WLAN

There is no template currently defined for this DF.

9.5.7 DF HNB

There is no template currently defined for this DF.

9.5.8 DF SoLSA

There is no template currently defined for this DF.

9.5.9 DF BeCast

There is no template currently defined for this DF.

9.5.10 DF ProSe

There is no template currently defined for this DF.

9.6 ISIM

9.6.1 Mandatory ISIM EFs

This Template is a "Created by default" type template. This template shall be supported by the eUICCs supporting the ISIM application.
Template OID: `{joint-iso-itu-t(2) international-organizations(23) simalliance(143) euicc-profile(1) template(2) isim(8)}`

Ass. Serv.	FID	EF Name	File Type	NB Rec.	File / Rec Size	Access Rules	Default Value	Content Required	Parameters
	XXXX	ADF ISIM	ADF			14			AID, Temporary FID, pinStatusTemplateDO
	6F02	EF IMPI	TR	NA	X	2		Y	Size
	6F04	EF IMPU	LF	1	X	2		Y	Record size
	6F03	EF Domain	TR	NA	X	2		Y	Size
	6F07	EF IST	TR	NA	14	2		Y	
	6FAD	EF AD	TR	NA	3	10	000000	N	
	6F06	EF ARR	LF	X	X	10		Y	Record size, File size

9.6.2 Optional ISIM EFs

This Template is a "Not created by default" type template. This template shall be supported by the eUICCs supporting the ISIM application.
Template OID: `{joint-iso-itu-t(2) international-organizations(23) simalliance(143) euicc-profile(1) template(2) opt-isim(9)}`

Ass. Serv.	FID	EF Name	File Type	NB Rec.	File / Rec Size	Access Rules	Default Value	Content Required	Parameters
1 or 5	6F09	EF P-CSCF	LF	1	X	2		Y	Record size
6 or 8	6F3C	EF SMS	LF	10	176	5	00 FF…FF	N	
8	6F42	EF SMSP	LF	1	38	5	FF...FF	N	
6 or 8	6F43	EF SMSS	TR	NA	2	5	FFFF	N	
7or 8	6F47	EF SMSR	LF	10	30	5	FF...FF	N	

2	6FD5	EF GBABP	TR	NA	X	5	FF…FF	N	Size
2	6FD7	EF GBANL	LF	X	X	2	FF...FF	N	Record size, File size
2 or 4	6FDD	EF NAFKCA	LF	X	X	2	FF…FF	N	Record size, File size
10	6FE7	EF UICCIARI	LF	X	X	2		Y	Record size, File size

Files EF GBABP, EF GBANL and EF NAFKCA are associated with services which require support at the eUICC operating system level. So, even if indicated in the profile, the creation of these files shall be skipped by the eUICC if these functionalities are not supported by the eUICC framework. In that case, the eUICC shall answer to the Profile Creator with a status code set to "`feature-not-supported`" with "`additional-information`" set to '1' if GBA is not supported. The bits related to GBA in the EF IST shall also be cleared if not supported by the eUICC. If "mandated" is set in the PE header, the installation of the Profile shall be aborted.

9.7 CSIM

9.7.1 Mandatory CSIM EFs

This Template is a "Created by default" type template. This template shall be supported by the eUICCs supporting the CSIM application.

Template OID: `{joint-iso-itu-t(2) international-organizations(23) simalliance(143) euicc-profile(1) template(2) csim(10)}`

Ass. Serv.	FID	EF Name	File Type	NB Rec.	File / Rec Size	Access Rules	Default Value	Content Required	Parameters
	XXXX	ADF CSIM	ADF			14			AID, Temporary FID, pinStatusTemplateDO
	6F06	EF ARR	LF	X	X	10		Y	Record size, File size
	6F21	EF CALL_COUNT	CY	10	2	7	00 00	N	
	6F22	EF IMSI_M	TR	NA	10	6	00…00	N	
	6F23	EF IMSI_T	TR	NA	10	6	00…00	N	
	6F24	EF TMSI	TR	NA	16	13	00 00 00 00 00 00 00 00 00 FF FF FF FF 00 00 00	N	
	6F25	EF AH	TR	NA	2	5	00 00	N	
	6F26	EF AOP	TR	NA	1	5		Y	
	6F27	EF ALOC	TR	NA	7	5		Y	
	6F28	EF CDMAHOME	LF	X	5	5	00...00	N	File size
	6F29	EF ZNREGI	LF	X	8	5	00…00	N	File size
	6F2A	EF SNREGI	TR	NA	7	5		Y	
	6F2B	EF DISTREGI	TR	NA	8	5	00…00	N	
	6F2C	EF ACCOLC	TR	NA	1	2		Y	
	6F2D	EF TERM	TR	NA	1	5		Y	
	6F2F	EF ACP	TR	NA	7	5		Y	
	6F30	EF PRL	TR	NA	X	2		Y	
	6F31	EF RUIMID	TR	NA	5	4		Y	
	6F32	EF CSIM_ST	TR	NA	6	2		Y	
	6F33	EF SPC	TR	NA	3	3	00…00	N	

	6F34	EF OTAPASPC	TR	NA	1	5	00	N	
	6F35	EF NAMLOCK	TR	NA	1	5		Y	
	6F36	EF OTA	TR	NA	17	2		Y	
	6F37	EF SP	TR	NA	1	2		Y	
	6F38	EF ESN_MEID_ME	TR	NA	8	10	00…00	N	
	6F3A	EF LI	TR	NA	6	1	FF…FF	N	
	6F42	EF USGIND	TR	NA	1	2		Y	
	6F43	EF AD	TR	NA	3	10	00…00	N	
	6F45	EF MAX_PRL	TR	NA	4	2		Y	
	6F46	EF SPCS	TR	NA	1	12		Y	
	6F55	EF MECRP	TR	NA	3	5	00..00	N	
	6F70	EF HOME_TAG	TR	NA	X	2		Y	
	6F71	EF GROUP_TAG	TR	NA	X	2		Y	
	6F72	EF SPECIFIC_TAG	TR	NA	X	2		Y	
	6F73	EF CALL_PROMPT	TR	NA	X	2		Y	

Files with high update activity: EF CALL_COUNT, EF TMSI, EF ALOC, EF ZNREGI, EF SNREGI and EF DISTREGI.

9.7.2 Optional CSIM EFs

This Template is a "Not created by default" type template. This template shall be supported by the eUICCs supporting the CSIM application.

Template OID: `{joint-iso-itu-t(2) international-organizations(23) simalliance(143) euicc-profile(1) template(2) opt-csim(11)}`

Ass. Serv.	FID	EF Name	File Type	NB Rec.	File / Rec Size	Access Rules	Default Value	Content Required	Parameters
	6F2E	EF SSCI	TR	NA	1	5		Y	
2	6F3B	EF FDN	LF	20	26	8	FF…FF	N	
6	6F3C	EF SMS	LF	X	X	5	00 FF…FF	N	Record size, File size
7	6F3D	EF SMSP	LF	X	X	5	FF…FF	N	Record size, File size
6	6F3E	EF SMSS	TR	NA	X	5	FF…FF	N	Size
	6F3F	EF SSFC	TR	NA	X	5		Y	Size
10	6F41	EF SPN	TR	NA	35	10		Y	
	6F44	EF MDN	LF	X	11	5		Y	File size
	6F47	EF ECC	TR	NA	X	10	FF…FF	N	Size
14,15	6F48	EF ME3GPDOPC	TR	NA	1	5	00	N	
14,15	6F49	EF 3GPDOPM	TR	NA	1	5		Y	
14	6F4A	EF SIPCAP	TR	NA	4	2		Y	
15	6F4B	EF MIPCAP	TR	NA	5	2		Y	
14	6F4C	EF SIPUPP	TR	NA	X	2		Y	Size
15	6F4D	EF MIPUPP	TR	NA	X	2		Y	Size
14	6F4E	EF SIPSP	TR	NA	1	5		Y	
15	6F4F	EF MIPSP	TR	NA	1	5		Y	
14	6F50	EF SIPPAPSS	TR	NA	X	5		Y	Size
	6F53	EF PUZL	TR	NA	X	2		Y	Size
	6F54	EF MAXPUZL	TR	NA	5	5		Y	
8	6F56	EF HRPDCAP	TR	NA	3	2		Y	

8	6F57	EF HRPDUPP	TR	NA	X	2		Y	Size
	6F58	EF CSSPR	TR	NA	1	2	FF	N	
8	6F59	EF ATC	TR	NA	1	2		Y	
	6F5A	EF EPRL	TR	NA	X	2		Y	Size
9	6F5B	EF BCSMScfg	TR	NA	1	2		Y	e
9	6F5C	EF BCSMSpref	TR	NA	1	5	FF	N	
9	6F5D	EF BCSMStable	LF	X	X	2	00 FF…FF	N	Record size, File size
9	6F5E	EF BCSMSp	LF	X	2	5	FF FF	N	File size
18	6F63	EF BAKPARA	LF	X	X	2		Y	Record size, File size
18	6F64	EF UpBAKPARA	CY	X	X	2		Y	Record size, File size
19	6F65	EF MMSN	LF	X	X	5	00 00 00 FF…FF	N	Record size, File size
20	6F66	EF EXT8	LF	X	X	5	FF…FF	N	Record size, File size
19	6F67	EF MMSICP	TR	NA	X	2	FF…FF	N	Size
19	6F68	EF MMSUP	LF	X	X	5	FF…FF	N	Record size, File size
19,21	6F69	EF MMSUCP	TR	NA	X	5	FF…FF	N	Size
22	6F6A	EF AUTH_CAPABILITY	LF	X	5	2	00…00	N	File size
16	6F6B	EF 3GCIK	TR	NA	32	2		Y	
25	6F6C	EF DCK	TR	NA	20	5		Y	
23	6F6D	EF GID1	TR	NA	X	2		Y	Size
24	6F6E	EF GID2	TR	NA	X	2		Y	Size
26	6F6F	EF CDMACNL	TR	NA	X	2		Y	Size
34	6F74	EF SF_EUIMID	TR	NA	7	4		Y	
2	6F75	EF EST	TR	NA	X	8		Y	Size
	6F76	EF HIDDEN_KEY	TR	NA	4	5		Y	

17	6F77	EF LCSVER	TR	NA	X	2		Y	Size
17	6F78	EF LCSCP	TR	NA	X	2		Y	Size
4	6F79	EF SDN	LF	X	X	2		Y	Record size, File size
3	6F7A	EF EXT2	LF	X	13	8		Y	File size
5	6F7B	EF EXT3	LF	X	13	2		Y	File size
28	6F7C	EF ICI	CY	X	X	5		Y	Record size, File size
27	6F7D	EF OCI	CY	X	X	5		Y	Record size, File size
29	6F7E	EF EXT5	LF	X	13	5		Y	File size
33	6F7F	EF CCP2	LF	X	X	5		Y	Record size, File size
	6F80	EF AppLabels	TR	NA	X	2		Y	Size
	6F81	EF MODEL	TR	NA	126	5	FF..FF	N	
36	6F82	EF RC	TR	NA	X	10		Y	Size
6	6F83	EF SMSCAP	TR	NA	4	2		Y	
35	6F84	EF MIPFlags	TR	NA	1	2		Y	
14	6F85	EF 3GPDUPPExt	TR	NA	X	2		Y	Size
35	6F87	EF IPV6CAP	TR	NA	21	2		Y	
14	6F88	EF TCPConfig	TR	NA	2	2		Y	
14	6F89	EF DGC	TR	NA	3	2		Y	
37	6F8A	EF WAPBrowserCP	TR	NA	X	2		Y	Size
37	6F8B	EF WAPBrowserBM	TR	NA	X	5		Y	Size
19	6F8C	EF MMSConfig	TR	NA	8	2		Y	
38	6F8D	EF JDL	TR	NA	X	2		Y	Size

Files with high update activity: EF SMS, EF SMSP, EF BCSMSpref, EF BCSMStable, EF BCSMSp, EF BAKPARA, EF UpBAKPARA, EF ICI, EF OCI and EF WAPBrowserBM

9.8 EAP

There is no template currently defined for this DF.

9.9 Access Rules Definition

The following table lists the access rules used by the different templates defined above.

File Access Conditions						Access Rules	Values
Read	**Update**	**Incr.**	**Act.**	**Deact.**	**Delete**		
ALWAYS	PIN 1	NEVER	ADM 1	ADM 1	ADM 1	1	8001019000 800102A406830101950108 800158A40683010A950108
PIN 1	ADM 1	NEVER	ADM 1	ADM 1	ADM 1	2	800101A406830101950108 80015AA40683010A950108
ADM 1	ADM 1	NEVER	ADM 1	ADM 1	ADM 1	3	80015BA40683010A950108
ALWAYS	NEVER	NEVER	NEVER	NEVER	ADM 1	4	8001019000 80015A9700
PIN 1	PIN 1	NEVER	ADM 1	ADM 1	ADM 1	5	800103A406830101950108 800158A40683010A950108
PIN 1	ADM 1	NEVER	PIN 1	ADM 1	ADM 1	6	800111A406830101950108 80014AA40683010A950108
PIN 1	PIN 1	PIN 1	ADM 1	ADM 1	ADM 1	7	800103A406830101950108 800158A40683010A950108 840132A406830101950108
PIN 1	PIN 2 (See NOTE 1)	NEVER	ADM 1	ADM 1	ADM 1	8	800101A406830101950108 800102A406830181950108 800158A40683010A950108
ALWAYS	PIN1	NEVER	PIN 1	PIN 1	ADM 1	9	8001019000 80011AA406830101950108 800140A40683010A950108
ALWAYS	ADM 1	NEVER	ADM 1	ADM 1	ADM 1	10	8001019000 80015AA40683010A950108
ALWAYS	NEVER	NEVER	ADM 1	ADM 1	NEVER	11	8001019000 800118A40683010A950108 8001429700
PIN 1	NEVER	NEVER	NEVER	NEVER	NEVER	12	800101A406830101950108 80015A9700
PIN 1	PIN 1	NEVER	PIN 1	ADM 1	ADM 1	13	800113A406830101950108 800148A40683010A950108
MF/ADF/DF Access Conditions							
Delete self	Terminate	Activate	Deactivate	Create DF	Create EF		
ADM 1	NEVER	ADM 1	ADM 1	ADM 1	ADM 1	14	80015EA40683010A950108

NOTE 1: PIN2 refers to local PIN1
NOTE 2: No access conditions are defined for Resize. Therefore, access condition for Resize is set by default to "NEVER". If ADM1 access condition is required for Resize, the following access rule can be used: 84 01 D4 A4 06 83 01 0A 95 01 08

These access rules may be used by the Profile Creator in order to set the content of EF ARR files.

NOTE: If different access conditions are encoded in the EF_{ARR} files provided in the Profile Package, Profile creator should ensure to provide the modified references for all the files defined in the templates and affected by these modifications.

10. ANNEX B (Normative): List of OIDs

All the OIDs used in this specification are located under the SIMalliance branch dedicated for this specification: `{joint-iso-itu-t(2) international-organizations(23) simalliance(143) euicc-profile(1)}`

The table below lists the OIDs currently assigned:

ASN.1 Notation	Dot Notation	Comments
`{joint-iso-itu-t(2) international-organizations(23) simalliance(143) euicc-profile(1) spec-version(1) version-two(2)}`	2.23.143.1.1.2	Specification version
`{joint-iso-itu-t(2) international-organizations(23) simalliance(143) euicc-profile(1) template(2) mf(1)}`	2.23.143.1.2.1	MF system template
`{joint-iso-itu-t(2) international-organizations(23) simalliance(143) euicc-profile(1) template(2) cd(2)}`	2.23.143.1.2.2	DF CD file system template
`{joint-iso-itu-t(2) international-organizations(23) simalliance(143) euicc-profile(1) template(2) telecom(3)}`	2.23.143.1.2.3	DF TELECOM file system template
`{joint-iso-itu-t(2) international-organizations(23) simalliance(143) euicc-profile(1) template(2) usim(4)}`	2.23.143.1.2.4	ADF USIM "created by default" file system template
`{joint-iso-itu-t(2) international-organizations(23) simalliance(143) euicc-profile(1) template(2) opt-usim(5)}`	2.23.143.1.2.5	ADF USIM "not created by default" file system template
`{joint-iso-itu-t(2) international-organizations(23) simalliance(143) euicc-profile(1) template(2) phonebook(6)}`	2.23.143.1.2.6	DF PHONEBOOK under ADF USIM file system template
`{joint-iso-itu-t(2) international-organizations(23) simalliance(143) euicc-profile(1) template(2) gsm-access(7)}`	2.23.143.1.2.7	DF GSM-ACCESS under ADF USIM file system template
`{joint-iso-itu-t(2) international-organizations(23) simalliance(143) euicc-profile(1) template(2) isim(8)}`	2.23.143.1.2.8	ADF ISIM "created by default" file system template
`{joint-iso-itu-t(2) international-organizations(23) simalliance(143) euicc-profile(1) template(2) opt-isim(9)}`	2.23.143.1.2.9	ADF ISIM "not created by default" file system template
`{joint-iso-itu-t(2) international-organizations(23) simalliance(143) euicc-profile(1) template(2) csim(10)}`	2.23.143.1.2.10	ADF CSIM "created by default" file system template
`{joint-iso-itu-t(2) international-organizations(23) simalliance(143) euicc-profile(1) template(2) opt-csim(11)}`	2.23.143.1.2.11	ADF CSIM "not created by default" file system template

11. ANNEX C (Informative): Example of Profile Package

11.1 Example of Profile Package structure

The following example shows a typical structure of a Profile Package containing a USIM application and a supplementary SD containing an application.

Profile Element	Comments
ProfileHeader	
PE-MF	
PE-PUKCodes	Only one set of PUK codes exist in a Profile Package
PE-PINCodes	Creates the Global PIN codes
PE-TELECOM	
PE-GenericFileManagement	To be repeated in order to create the files required in the DF Phonebook under DF Telecom
PE-USIM	Creates a USIM ADF and the associated files
PE-OPT-USIM	
PE-PHONEBOOK	Creates DF PHONEBOOK under USIM ADF
PE-AKAParameter	Sets the AKA parameters related to the previously created USIM
PE-PINCodes	Creates the local PIN code structure at the USIM ADF level
PE-GenericFileManagement	To be repeated in order to create additional files required in the ADF USIM
PE-GSM-ACCESS	
PE-SecurityDomain	Creates the MNO-SD
PE-SecurityDomain	Creates a SSD
PE-Application	Loads a USAT application
PE-Application	Loads an application in the SSD
PE-RFM	Sets the RFM parameters for the Profile
PE-End	End of the Profile Package

11.2 Example of Profile Package content

11.2.1 Overview

Here is a sample of Profile Package content that can be used during the testing of the Profile download process. Testing the Profile Package interpreter implementation in the eUICC is conducted by using the proper test specification.

This Profile, defined in the following chapters, contains the following Components:

- MF and USIM ADF
- PIN and PUK codes
- NAA using Milenage algorithm
- MNO-SD supporting SCP80 in 3DES
- SSD supporting SCP80 in 3DES
- An application instantiated in the MNO SD
- An application instantiated in the SSD
- RFM application

The parameters below have been chosen to personalize the Profile:

- Profile type: "SIMalliance Profile Package"
- ICCID: '89019990001234567893'
- IMSI: 234101943787656

- MNO-SD AID / TAR: 'A000000151000000' / 'B20100'
- UICC RFM application AID / TAR: 'A00000055910100001' / 'B00000'
- USIM RFM application AID / TAR: ' A00000055910100002' / 'B00020'
- Executable Load File AID for SD: 'A0000001515350'
- Executable Module AID for SD: 'A000000151535041'
- SSD AID / TAR: 'A00000055910100102736456616C7565' / '6C7565'
- All access rules are defined in chapter 9.9.

11.2.2 Profile HEADER

ASN.1 Format	DER TLV encoding

headerValue ProfileElement ::= header : {
  major-version 2,
  minor-version 1,
  profileType "SIMalliance Sample Profile",
  iccid '89019990001234567893'H,
  eUICC-Mandatory-services {
  usim NULL,
  milenage NULL,
  javacard NULL
  },
  eUICC-Mandatory-GFSTEList {
  { 2 23 143 1 2 1 }, --id-MF
  { 2 23 143 1 2 4 } --id-USIM
  }
}
	A0 48
	  80 01 02
	  81 01 01
	  82 1A 53494D616C6C69616E63652053616D706C652050726F66696C65
	  83 0A 89019990001234567893
	  A5 06
	  81 00
	  84 00
	  8B 00
	
	  A6 10
	  06 06 67810F010201
	  06 06 67810F010204

11.2.3 PE MF (Using Template)

ASN.1 Format	DER TLV encoding

mfVal ProfileElement ::= mf : {
  mf-header {
  mandated NULL,
  identification 1
  },
  templateID { 2 23 143 1 2 1 },
  mf {
  fileDescriptor : {
  pinStatusTemplateDO '01020A'H
  }
  },
  ef-pl {
  fileDescriptor : {
-- EF PL modified to use Access Rule 15 within EF ARR
  securityAttributesReferenced '0F'H
  }
  },
  ef-iccid {
-- swapped ICCID: 98109909002143658739
  fillFileContent : '98109909002143658739'H
  },
  ef-dir {
  fileDescriptor : {
	B0 8201F8
	  A0 05
	  80 00
	  81 01 01
	
	  81 06 67810F010201
	  A2 07
	  A1 05
	  C6 03 01020A
	
	  A3 05
	  A1 03
	
	  8B 01 0F
	
	  A4 0C
	
	  83 0A 98109909002143658739
	
	  A5 27
	  A1 09

-- Shareable Linear Fixed File	
-- 4 records, record length: 38 bytes	
fileDescriptor '42210026'H,	82 04 42210026
efFileSize '98'H	80 01 98
},	
-- USIM AID: A0000000871002FF33FF018900000100	
fillFileContent : '61184F10A0000000871002FF33FF01890000010050045553494D'H	83 1A 61184F10A0000000871002FF33FF01890000010050045553494D
},	
ef-arr {	A6 82019E
fileDescriptor : {	A1 0A
-- Shareable Linear Fixed File	
-- 15 records, record length: 37 bytes	
-- ARR created with content recommended in Annex A (Section 9.9) plus one	
additional record for use with EF_PL	
fileDescriptor '42210025'H,	82 04 42210025
efFileSize '022B'H	80 02 022B
},	
fillFileContent :	
'8001019000800102A40683010195010880015 8A40683010A950108'H,	83 1B 80010190008001 02A40683010195010880015 8A40683010A950108
fillFileOffset : 10,	82 01 0A
fillFileContent : '800101A406830101950108800 15AA40683010A950108'H,	83 16 800101A40683010195010880015AA40683010A950108
fillFileOffset : 15,	82 01 0F
fillFileContent : '80015BA40683010A950108'H,	83 0B 80015BA40683010A950108
fillFileOffset : 26,	82 01 1A
fillFileContent : '800101900080015A9700'H,	83 0A 800101900080015A9700
fillFileOffset : 27,	82 01 1B
fillFileContent : '800103A4068301019501088001 58A40683010A950108'H,	83 16 800103A40683010195010880015 8A40683010A950108
fillFileOffset : 15,	82 01 0F
fillFileContent : '800111A40683010195010880014AA40683010A950108'H,	83 16 800111A40683010195010880014AA40683010A950108
fillFileOffset : 15,	82 01 0F
fillFileContent :	83 21
'800103A40683010195010880015 8A40683010A95010884013 2A40683010195010 8'H,	800103A40683010195010880015 8A40683010A950108840132A406830101950108
fillFileOffset : 4,	82 01 04
fillFileContent :	83 21
'800101A40683010195010880010 2A40683018195010880015 8A40683010A950108'H,	800101A406830101950108800102A40683018195010880015 8A40683010A950108
fillFileOffset : 4,	82 01 04
fillFileContent :	
'80010190008001 1AA40683010195010880014 0A40683010A950108'H,	83 1B 8001019000800 11AA406830101950108800140A40683010A950108
fillFileOffset : 10,	82 01 0A
fillFileContent : '800101900080015AA40683010A950108'H,	83 10 800101900080015AA40683010A950108
fillFileOffset : 21,	82 01 15
fillFileContent : '8001019000800118A40683010A9501088001429700'H,	83 15 8001019000800118A40683010A9501088001429700
fillFileOffset : 16,	82 01 10
fillFileContent : '800101A40683010195010880015A9700'H,	83 10 800101A40683010195010880015A9700
fillFileOffset : 21,	82 01 15
fillFileContent : '800113A40683010195010880014 8A40683010A950108'H,	83 16 800113A406830101950108800148A40683010A950108
fillFileOffset : 15,	82 01 0F
fillFileContent : '80015EA40683010A950108'H,	83 0B 80015EA40683010A950108
fillFileOffset : 26,	82 01 1A

-- Rule 15: [Read: Always][Update/CreateEF: PIN Appl 1|PIN Appl 2][Deactivate, Activate, DeleteSelf: ADM1] fillFileContent : '80010190008001 02A010A406830101950108A406830102950108800158A40683010A950108'H } }	83 25 80010190008001 02A010A406830101950108A406830102950108800158A40683010A950108

11.2.4 PE MF (Using Generic File Management)

This is an alternative method used for creating the MF file system. Only one method shall be present in a real Profile Package.

ASN.1 Format	DER TLV encoding

altMFVal ProfileElement ::= genericFileManagement : {
gfm-header {
mandated NULL,
identification 1
},
fileManagementCMD {
{
-- create MF
createFCP : {
fileDescriptor '7821'H,
fileID '3F00'H,
securityAttributesReferenced '0E'H,
pinStatusTemplateDO '01020A'H
},
-- create PL
createFCP : {
fileDescriptor '4121'H,
fileID '2F05'H,
securityAttributesReferenced '0F'H,
efFileSize '03'H,
shortEFID '28'H
},
-- create ICCID
createFCP : {
fileDescriptor '4121'H,
fileID '2FE2'H,
securityAttributesReferenced '0B'H,
efFileSize '0A'H
},
-- swapped ICCID: 98109909002143658739
fillFileContent : '98109909002143658739'H,
-- create DIR
-- Shareable Linear Fixed File
-- 4 records, record length: 38 bytes
createFCP : {
fileDescriptor '42210026'H,
	A1 820236
	A0 05
	80 00
	81 01 01
	
	A1 82022B
	30 820227
	
	62 10
	82 02 7821
	83 02 3F00
	8B 01 0E
	C6 03 01020A
	
	62 11
	82 02 4121
	83 02 2F05
	8B 01 0F
	80 01 03
	88 01 28
	
	62 0E
	82 02 4121
	83 02 2FE2
	8B 01 0B
	80 01 0A
	
	81 0A 98109909002143658739
	
	62 13
	82 04 42210026

fileID '2F00'H,	83 02 2F00
securityAttributesReferenced '0A'H,	8B 01 0A
efFileSize '98'H,	80 01 98
shortEFID 'F0'H	88 01 F0
},	
-- USIM AID: A0000000871002FF33FF018900000100	
fillFileContent :	81 1A 61184F10A0000000871002FF33FF01890000010050045553494D
'61184F10A0000000871002FF33FF01890000010050045553494D'H,	
-- create ARR	
createFCP : {	62 11
-- Shareable Linear Fixed File	
-- 15 records, record length: 37 bytes	
fileDescriptor '42210025'H,	82 04 42210025
fileID '2F06'H,	83 02 2F06
securityAttributesReferenced '0A'H,	8B 01 0A
efFileSize '022B'H	80 02 022B
},	
fillFileContent :	
'8001019000800102A406830101950108800158A40683010A950108'H,	81 1B 8001019000800102A406830101950108800158A40683010A950108
fillFileOffset : 10,	02 01 0A
fillFileContent : '800101A40683010195010880015AA40683010A950108'H,	81 16 800101A40683010195010880015AA40683010A950108
fillFileOffset : 15,	02 01 0F
fillFileContent : '80015BA40683010A950108'H,	81 0B 80015BA40683010A950108
fillFileOffset : 26,	02 01 1A
fillFileContent : '800101900080015A9700'H,	81 0A 800101900080015A9700
fillFileOffset : 27,	02 01 1B
fillFileContent : '800103A406830101950108800158A40683010A950108'H,	81 16 800103A406830101950108800158A40683010A950108
fillFileOffset : 15,	02 01 0F
fillFileContent : '800111A40683010195010880014AA40683010A950108'H,	81 16 800111A40683010195010880014AA40683010A950108
fillFileOffset : 15,	02 01 0F
fillFileContent :	81 21
'800103A406830101950108800158A40683010A950108840132A406830101950108'H,	800103A406830101950108800158A40683010A950108840132A406830101950108
fillFileOffset : 4,	02 01 04
fillFileContent :	81 21
'800101A406830101950108800102A406830181950108800158A40683010A950108'H,	800101A406830101950108800102A406830181950108800158A40683010A950108
fillFileOffset : 4,	02 01 04
fillFileContent :	
'80010190008001 1AA406830101950108800140A40683010A950108'H,	81 1B 80010190008001 1AA406830101950108800140A40683010A950108
fillFileOffset : 10,	02 01 0A
fillFileContent : '800101900080015AA40683010A950108'H,	81 10 800101900080015AA40683010A950108
fillFileOffset : 21,	02 01 15
fillFileContent : '8001019000800118A40683010A9501088001429700'H,	81 15 8001019000800118A40683010A9501088001429700
fillFileOffset : 16,	02 01 10
fillFileContent : '800101A40683010195010880015A9700'H,	81 10 800101A40683010195010880015A9700
fillFileOffset : 21,	02 01 15
fillFileContent : '800113A406830101950108800148A40683010A950108'H,	81 16 800113A406830101950108800148A40683010A950108
fillFileOffset : 15,	02 01 0F
fillFileContent : '80015EA40683010A950108'H,	81 0B 80015EA40683010A950108
fillFileOffset : 26,	02 01 1A

```
-- Rule 15: [Read: Always][Update/CreateEF: PIN Appl 1|PIN Appl 2][Deactivate,
Activate, DeleteSelf: ADM1]
    fillFileContent :
'80010190008001O2A010A406830101950108A406830102950108800158A40683010A950108'H,
-- create UMPC
      createFCP : {
        fileDescriptor '4121'H,
        fileID '2F08'H,
        securityAttributesReferenced '0A'H,
        efFileSize '05'H
      }
    }
  }
}
```

```
      81 25
80010190008001O2A010A406830101950108A406830102950108800158A40683010A950108

      62 0E
        82 02 4121
        83 02 2F08
        8B 01 0A
        80 01 05
```

11.2.5 PE PUK

ASN.1 Format	DER TLV encoding

`pukVal ProfileElement ::= pukCodes : {`
`  puk-Header {`
`    mandated NULL,`
`    identification 2`
`  },`
`  pukCodes {`
`    {`
`      keyReference pukAppl1,`
`-- PUK = 00000000`
`      pukValue '3030303030303030'H,`
`      -- maxNumOfAttemps:9, retryNumLeft:9`
`      maxNumOfAttemps-retryNumLeft 153`
`    },`
`    {`
`      keyReference pukAppl2,`
`-- PUK = 12345678`
`      pukValue '3132333435363738'H`
`    },`
`    {`
`      keyReference secondPUKAppl1,`
`-- PUK = 12345678`
`      pukValue '3132333435363738'H,`
`-- maxNumOfAttemps:8, retryNumLeft:8`
`      maxNumOfAttemps-retryNumLeft 136`
`    }`
`  }`
`}`
	`A3 3F`
	`  A0 05`
	`    80 00`
	`    81 01 02`
	
	`  A1 36`
	`    30 11`
	`      80 01 01`
	
	`      81 08 3030303030303030`
	
	`      82 02 0099`
	
	`    30 0D`
	`      80 01 02`
	
	`      81 08 3132333435363738`
	
	`    30 12`
	`      80 02 0081`
	
	`      81 08 3132333435363738`
	
	`      82 02 0088`

11.2.6 PE PIN

ASN.1 Format	DER TLV encoding

pinVal ProfileElement ::= pinCodes : {
  pin-Header {
  mandated NULL,
  identification 3
  },
  pinCodes pinconfig : {
  {
  keyReference pinAppl1,
-- PIN = 1234
  pinValue '31323334FFFFFFFF'H,
  unblockingPINReference pukAppl1
  },
  {
  keyReference pinAppl2,
-- PIN = 0000
  pinValue '30303030FFFFFFFF'H,
  unblockingPINReference pukAppl2
  },
  {
  keyReference adm1,
-- PIN = 5678
  pinValue '35363738FFFFFFFF'H,
  pinAttributes 1
  }
  }
}
	A2 41
	  A0 05
	  80 00
	  81 01 03
	
	  A1 38
	  A0 36
	  30 10
	  80 01 01
	  81 08 31323334FFFFFFFF
	  82 01 01
	
	  30 10
	  80 01 02
	
	  81 08 30303030FFFFFFFF
	  82 01 02
	
	  30 10
	  80 01 0A
	
	  81 08 35363738FFFFFFFF
	  83 01 01

11.2.7 PE USIM (Using Template)

ASN.1 Format	DER TLV encoding

usimValue ProfileElement ::= usim : {
  usim-header {
  mandated NULL,
  identification 4
  },
  templateID { 2 23 143 1 2 4 },
  adf-usim {
  fileDescriptor : {
  fileID '7FF1'H,
  dfName 'A0000000871002FF33FF018900000100'H,
  pinStatusTemplateDO '01810A'H
  }
  },
	B3 77
	  A0 05
	  80 00
	  81 01 04
	
	  81 06 67810F010204
	  A2 1D
	  A1 1B
	  83 02 7FF1
	  84 10 A0000000871002FF33FF018900000100
	  C6 03 01810A

ef-imsi {
-- numerical format: 234101943787656
fillFileContent : '082943019134876765'H
},
ef-arr {
fileDescriptor : {
linkPath '2F06'H
}
},
ef-ust {
-- Service Dialling Numbers, Short Message Storage
fillFileContent : '0A2E178CE73204000000000000'H
},
ef-spn {
-- ASCII format: "SIMalliance"
fillFileContent : '0253494D616C6C69616E6365'H
},
ef-est {
-- Services deactivated
fillFileContent : '00'H
},
ef-acc {
-- Access class 2
fillFileContent : '0040'H
},
ef-ecc {
-- Emergency Call Code 911
fillFileContent : '19F1FF01'H
}
}
	A3 0B
	
	83 09 082943019134876765
	
	A4 06
	A1 04
	C7 02 2F06
	
	A8 0F
	
	83 0D 0A2E178CE73204000000000000
	
	AD 0E
	83 0C 0253494D616C6C69616E6365
	
	AE 03
	83 01 00
	
	B2 04
	
	83 02 0040
	
	B6 06
	
	83 04 19F1FF01

11.2.8 PE USIM (Using Generic File Management)

This is an alternative method used for creating the USIM file system. Only one method shall be present in a real Profile Package.

ASN.1 Format	DER TLV encoding

altUsimValue ProfileElement ::= genericFileManagement : {
gfm-header {
mandated NULL,
identification 4
},
fileManagementCMD {
{
-- ADF_USIM
createFCP : {
fileDescriptor '7821'H,
fileID '7FF1'H,
dfName 'A0000000871002FF33FF018900000100'H,
	A1 82029E
	A0 05
	80 00
	81 01 04
	
	A1 820293
	30 82028F
	
	62 22
	82 02 7821
	83 02 7FF1
	84 10 A0000000871002FF33FF018900000100

```
        securityAttributesReferenced '0A'H,
        pinStatusTemplateDO '01810A'H
      },

-- EF IMSI
      createFCP : {
        fileDescriptor '4121'H,
        fileID '6F07'H,
        securityAttributesReferenced '02'H,
        efFileSize '09'H,
        shortEFID '38'H
      },
    -- provide content for EF_IMSI
    -- numerical format: 234101943787656
      fillFileContent : '082943019134876765'H,

-- EF_ARR Link
      createFCP : {
        fileDescriptor '42210025'H,
        fileID '6F06'H,
        securityAttributesReferenced '0A'H,
        shortEFID 'B8'H,
        linkPath '2F06'H
      },

-- EF Keys
      createFCP : {
        fileDescriptor '4121'H,
        fileID '6F08'H,
        securityAttributesReferenced '05'H,
        efFileSize '21'H,
        shortEFID '40'H,
        proprietaryEFInfo {
          specialFileInformation '80'H,
          fillPattern '07FF'H
        }
      },

-- EF_KeysPS
      createFCP : {
        fileDescriptor '4121'H,
        fileID '6F09'H,
        securityAttributesReferenced '05'H,
        efFileSize '21'H,
        shortEFID '48'H,
        proprietaryEFInfo {
          specialFileInformation '80'H,
          fillPattern '07FF'H
        }
      },
```

```
        8B 01 0A
        C6 03 01810A


      62 11
        82 02 4121
        83 02 6F07
        8B 01 02
        80 01 09
        88 01 38


      81 09 082943019134876765

      62 14
        82 04 42210025
        83 02 6F06
        8B 01 0A
        88 01 B8
        C7 02 2F06

      62 1A
        82 02 4121
        83 02 6F08
        8B 01 05
        80 01 21
        88 01 40
        A5 07
          C0 01 80
          C1 02 07FF

      62 1A
        82 02 4121
        83 02 6F09
        8B 01 05
        80 01 21
        88 01 48
        A5 07
          C0 01 80
          C1 02 07FF
```

```
-- EF HPPLMN
      createFCP : {
        fileDescriptor '4121'H,
        fileID '6F31'H,
        securityAttributesReferenced '02'H,
        efFileSize '01'H,
        shortEFID '90'H,
        proprietaryEFInfo {
-- specialFileInformation with Default value
          specialFileInformation '00'H,
          fillPattern '0A'H
        }
      },

-- EF UST
      createFCP : {
        fileDescriptor '4121'H,
        fileID '6F38'H,
        securityAttributesReferenced '02'H,
        efFileSize '0E'H,
        shortEFID '20'H
      },
      -- provide UST settings
      -- Service Dialling Numbers, Short Message Storage
      fillFileContent : '0A2E178CE73204000000000000'H,

-- EF_FDN
      createFCP : {
        fileDescriptor '4221001A'H,
        fileID '6F3B'H,
        securityAttributesReferenced '08'H,
        efFileSize '0208'H,
        shortEFID ''H,
        proprietaryEFInfo {
          fillPattern '00FF'H
        }
      },

-- EF_SMS
      createFCP : {
        fileDescriptor '422100B0'H,
        fileID '6F3C'H,
        securityAttributesReferenced '05'H,
        efFileSize '06E0'H,
        shortEFID ''H,
        proprietaryEFInfo {
          fillPattern '00FF'H
        }
      },
```

```
62 16
  82 02 4121
  83 02 6F31
  8B 01 02
  80 01 01
  88 01 90
  A5 03
    C1 01 0A

62 11
  82 02 4121
  83 02 6F38
  8B 01 02
  80 01 0E
  88 01 20

81 0D 0A2E178CE73204000000000000

62 19
  82 04 4221001A
  83 02 6F3B
  8B 01 08
  80 02 0208
  88 00
  A5 04
    C1 02 00FF

62 19
  82 04 422100B0
  83 02 6F3C
  8B 01 05
  80 02 06E0
  88 00
  A5 04
    C1 02 00FF
```

-- EF SMSP createFCP : { fileDescriptor '42210026'H, fileID '6F42'H, securityAttributesReferenced '05'H, efFileSize '26'H, shortEFID ''H },	62 12 82 04 42210026 83 02 6F42 8B 01 05 80 01 26 88 00

-- EF_SMSS
createFCP : {
fileDescriptor '4121'H,
fileID '6F43'H,
securityAttributesReferenced '05'H,
efFileSize '02'H,
shortEFID ''H,
proprietaryEFInfo {
specialFileInformation '80'H
}
},
	62 15
	82 02 4121
	83 02 6F43
	8B 01 05
	80 01 02
	88 00
	A5 03
	C0 01 80

-- EF_SPN createFCP : { fileDescriptor '4121'H, fileID '6F46'H, -- provide the full access rule including EF ARR File ID securityAttributesReferenced '6F060A'H, efFileSize '11'H, shortEFID ''H }, **-- ASCII format: "SIMalliance"** **fillFileContent : '0253494D616C6C69616E6365'H,**	62 12 82 02 4121 83 02 6F46 8B 03 6F060A 80 01 11 88 00 81 0C 0253494D616C6C69616E6365
-- EF_EST createFCP : { fileDescriptor '4121'H, fileID '6F56'H, securityAttributesReferenced '08'H, efFileSize '01'H, shortEFID '28'H }, -- EST Services deactivated fillFileContent : '00'H,	62 11 82 02 4121 83 02 6F56 8B 01 08 80 01 01 88 01 28 81 01 00
-- EF_START-HFN createFCP : { fileDescriptor '4121'H, fileID '6F5B'H, securityAttributesReferenced '05'H,	62 1B 82 02 4121 83 02 6F5B 8B 01 05

```
        efFileSize '06'H,                                    80 01 06
        shortEFID '78'H,                                     88 01 78
        proprietaryEFInfo {                                  A5 08
          specialFileInformation '80'H,                        C0 01 80
-- use of repeat pattern to initialize the content
          repeatPattern 'F00000'H                              C2 03 F00000
        }
      },

-- EF THRESHOLD
      createFCP : {                                        62 16
        fileDescriptor '4121'H,                              82 02 4121
        fileID '6F5C'H,                                      83 02 6F5C
        securityAttributesReferenced '02'H,                  8B 01 02
        efFileSize '03'H,                                    80 01 03
        shortEFID '80'H,                                     88 01 80
        proprietaryEFInfo {                                  A5 03
          specialFileInformation '80'H                         C0 01 80
        }
      },

-- EF PSLOCI
      createFCP : {                                        62 16
        fileDescriptor '4121'H,                              82 02 4121
        fileID '6F73'H,                                      83 02 6F73
        securityAttributesReferenced '05'H,                  8B 01 05
        efFileSize '0E'H,                                    80 01 0E
        shortEFID '60'H,                                     88 01 60
        proprietaryEFInfo {                                  A5 03
          specialFileInformation '80'H                         C0 01 80
        }
      },
      -- Initialize PSLOCI
      fillFileOffset : 7,                                  02 01 07
      fillFileContent : '00F1100000FF01'H,                 81 07 00F1100000FF01

-- EF ACC
      createFCP : {                                        62 11
        fileDescriptor '4121'H,                              82 02 4121
        fileID '6F78'H,                                      83 02 6F78
        securityAttributesReferenced '02'H,                  8B 01 02
        efFileSize '02'H,                                    80 01 02
        shortEFID '30'H                                      88 01 30
      },
      -- Provide Content for ACC
      -- Access class 2
      fillFileContent : '0040'H,                           81 02 0040

-- EF FPLMN
      createFCP : {                                        62 11
```

```
        fileDescriptor '4121'H,                                  82 02 4121
        fileID '6F7B'H,                                          83 02 6F7B
        securityAttributesReferenced '05'H,                      8B 01 05
        efFileSize '0C'H,                                        80 01 0C
        shortEFID '68'H                                          88 01 68
      },

-- EF LOCI
      createFCP : {                                            62 16
        fileDescriptor '4121'H,                                  82 02 4121
        fileID '6F7E'H,                                          83 02 6F7E
        securityAttributesReferenced '05'H,                      8B 01 05
        efFileSize '0B'H,                                        80 01 0B
        shortEFID '58'H,                                         88 01 58
        proprietaryEFInfo {                                      A5 03
          specialFileInformation '80'H                             C0 01 80
        }
      },
      -- Initialize LOCI
      fillFileOffset : 7,                                      02 01 07
      fillFileContent : '0000FF01'H,                           81 04 0000FF01

-- EF_AD
      createFCP : {                                            62 16
        fileDescriptor '4121'H,                                  82 02 4121
        fileID '6FAD'H,                                          83 02 6FAD
        securityAttributesReferenced '0A'H,                      8B 01 0A
        efFileSize '04'H,                                        80 01 04
        shortEFID '18'H,                                         88 01 18
        proprietaryEFInfo {                                      A5 03
-- use of fillPattern in Combination with fillFileContent (not
efficient in this example)
          fillPattern '00'H                                        C1 01 00
        }
      },
      -- Initialize AD
      fillFileOffset : 3,                                      02 01 03
      fillFileContent : '02'H,                                 81 01 02

-- EF_ECC
      createFCP : {                                            62 13
        fileDescriptor '42210004'H,                              82 04 42210004
        fileID '6FB7'H,                                          83 02 6FB7
        securityAttributesReferenced '0A'H,                      8B 01 0A
        efFileSize '04'H,                                        80 01 04
        shortEFID '08'H                                          88 01 08
      },
      -- Initialize ECC
      -- Emergency Call Code 911
      fillFileContent : '19F1FF01'H,                           81 04 19F1FF01
```

`-- EF NETPAR`
`createFCP : {`
`fileDescriptor '4121'H,`
`fileID '6FC4'H,`
`securityAttributesReferenced '05'H,`
`efFileSize '80'H,`
`shortEFID ''H,`
`proprietaryEFInfo {`
`specialFileInformation '80'H`
`}`
`},`
	`62 15`
	`82 02 4121`
	`83 02 6FC4`
	`8B 01 05`
	`80 01 80`
	`88 00`
	`A5 03`
	`C0 01 80`

`-- EF EPSLOCI`
`createFCP : {`
`fileDescriptor '4121'H,`
`fileID '6FE3'H,`
`securityAttributesReferenced '05'H,`
`efFileSize '12'H,`
`shortEFID 'F0'H,`
`proprietaryEFInfo {`
`specialFileInformation '80'H`
`}`
`},`
 	
`-- Initialize EF EPSLOCI`
`fillFileOffset : 15,`
`fillFileContent : '000001'H,`
	`62 16`
	`82 02 4121`
	`83 02 6FE3`
	`8B 01 05`
	`80 01 12`
	`88 01 F0`
	`A5 03`
	`C0 01 80`
	
	`02 01 0F`
	`81 03 000001`

`-- EF_EPSNSC`
`createFCP : {`
`fileDescriptor '4121'H,`
`fileID '6FE4'H,`
`securityAttributesReferenced '05'H,`
`efFileSize '50'H,`
`shortEFID 'C0'H,`
`proprietaryEFInfo {`
`specialFileInformation '80'H`
`}`
`}`
`}`
`}`
`}`
	`62 16`
	`82 02 4121`
	`83 02 6FE4`
	`8B 01 05`
	`80 01 50`
	`88 01 C0`
	`A5 03`
	`C0 01 80`

11.2.9 PE USIM PIN

ASN.1 Format	DER TLV encoding

usimPin ProfileElement ::= pinCodes : { -- Local USIM PIN
  pin-Header {
  mandated NULL,
  identification 5
  },
  pinCodes pinconfig : {
  {
  keyReference secondPINAppl1,
-- PIN = 1234
  pinValue '31323334FFFFFFFF'H,
  unblockingPINReference secondPUKAppl1,
  -- PIN is Enabled
  pinAttributes 1,
  -- maxNumOfAttemps:2, retryNumLeft:2
  maxNumOfAttemps-retryNumLeft 34
  }
  }
}
	A2 25
	  A0 05
	  80 00
	  81 01 05
	
	  A1 1C
	  A0 1A
	  30 18
	  80 02 0081
	  81 08 31323334FFFFFFFF
	  82 02 0081
	
	  83 01 01
	
	  84 01 22

11.2.10 PE NAA

11.2.10.1. Milenage AKA

ASN.1 Format	DER TLV encoding

akaMilenage ProfileElement ::= akaParameter : {
  aka-header {
  mandated NULL,
  identification 6
  },
  algoConfiguration algoParameter : {
  algorithmID milenage,
  algorithmOptions '01'H, -- RES and MAC 64 bits, CK and IK 128 bits
 	
  key '000102030405060708090A0B0C0D0E0F'H,
  opc '0102030405060708090A0B0C0D0E0F00'H,
-- rotationConstants uses default: '4000204060'H,
-- xoringConstants uses default: '00000000000000000000000000000001000000000000000020000000000000004000000000000008'H,
  authCounterMax '010203'H
  }
-- sqnOptions uses default: '02'H, i.e. Anonymity key used, SQN wrap around not allowed
-- sqnDelta uses default: '000010000000'H
-- sqnAgeLimit uses default: '000010000000'H
-- sqnInit: uses default: all bytes zero
	A4 3A
	  A0 05
	  80 00
	  81 01 06
	
	  A1 31
	  A1 2F
	  80 01 01
	  81 01 01
	
	  82 10 000102030405060708090A0B0C0D0E0F
	  83 10 0102030405060708090A0B0C0D0E0F00
	
	  86 03 010203

}	

11.2.10.2 CDMA

This example is provided for information but is not is not intended to be included in a test Profile without the full definition of CDMA application and files.

ASN.1 Format	DER TLV encoding

cdmaParam ProfileElement ::= cdmaParameter : {
cdma-header {
mandated NULL,
identification 15
},
authenticationKey '0102030405060708'H,
ssd '0123456789ABCDEF0123456789ABCDEF'H,
--HRDP Access Authentication Value:
0x43484150434841504348415043484150
hrpdAccessAuthenticationData
'821A420A821A420A821A420A821A420A80'H,
	A5 81 E9
	A0 05
	80 00
	81 01 0F
	
	81 08 0102030405060708
	82 10 0123456789ABCDEF0123456789ABCDEF
	
	83 11
	821A420A821A420A821A420A821A420A80

/* Simple IP CHAP SS Parameters: - Value: entry 00: 0x43484150434841504348415043484150 entry 01: 0x44554D4D5944554D4D5944554D4D5944554D4D5944554D4D5944554D4D5944 entry 02: 0x4E4144414E414441 */ simpleIPAuthenticationData '30821A420A821A420A821A420A821A420A80FD1155353565115535356511553535651155353565115535356510909C8288829C828882'H,	 84 3B 30821A420A821A420A821A420A821A420A80FD1155353565115535356511553535651155353565115535356510909C8288829C828882
/* Mobile IP SS Parameters: - Value: entry 00: - MN-AAA-SS: 0x31323334353637383930313233343536 - MN-HA-SS: 0x30303131323233333434353536363737 entry 01: - MN-AAA-SS: 0x44554D4D5944554D4D5944554D4D5944554D4D5944554D4D5944554D4D5944 - MN-HA-SS: 0x4E4144414E414441 entry 02: - MN-AAA-SS: 0x4E4144414E414441 - MN-HA-SS: 0x44554D4D5944554D4D5944554D4D5944554D4D5944554D4D5944554D4D5944 */ mobileIPAuthenticationData '3081899199A1A9B1B9C1C981899199A1A9B40C0C0C4C4C8C8CCCCD0D0D4D4D8D8DCDC7E88AA9A9AB288AA9A9AB288AA9A9AB288AA9A9AB288AA9A9AB288AA9A9AB28884E4144414E414441242720A220A720A220FD1155353565115535356511553535651155353565115535356510'H	 85 74 3081899199A1A9B1B9C1C981899199A1A9B40C0C0C4C4C8C8CCCCD0D0D4D4D8D8DCDC7E88AA9A9AB288AA9A9AB288AA9A9AB288AA9A9AB288AA9A9AB288AA9A9AB28884E4144414E414441242720A220A720A220FD1155353565115535356511553535651155353565115535356510

}	

11.2.11 PE MNO SD

11.2.11.1. Example 1

ASN.1 Format	DER TLV encoding

mnoSdValue ProfileElement ::= securityDomain : {
  sd-Header {
  mandated NULL,
  identification 7
  },
  instance {
  applicationLoadPackageAID 'A0000001515350'H,
  classAID 'A000000151535041'H,
  instanceAID 'A000000151000000'H,
  applicationPrivileges '82DC00'H,
  -- Secured
  lifeCycleState '0F'H,
  -- SCP80 supported, extradition supported
  applicationSpecificParametersC9 '810280008201F08701F0'H,
  -- other parameters may be necessary
  applicationParameters {
  -- TAR: B20100, MSL: 12
  uiccToolkitApplicationSpecificParametersField
  '0100000100000002011203B2010000'H
  }
  },
  keyList {
  {
  -- C-ENC + R-ENC
  keyUsageQualifier '38'H,
  -- may be used by SD and application
  keyAccess '00'H,
  -- ENC key
  keyIdentifier '01'H,
  keyVersionNumber '01'H,
  keyCompontents {
  {
  -- DES mode implicitly known (as an example)
  keyType '80'H,
  -- This value may be freely changed
  keyData '112233445566778899AABBCCDDEEFF10'H
  }
  }
  },
  {
	A6 81BB
	  A0 05
	  80 00
	  81 01 07
	
	  A1 44
	  4F 07 A0000001515350
	  4F 08 A000000151535041
	  4F 08 A000000151000000
	  82 03 82DC00
	
	  83 01 0F
	
	  C9 0A 810280008201F08701F0
	
	  EA 11
	
	  80 0F 0100000100000002011203B2010000
	
	  A2 6C
	  30 22
	
	  95 01 38
	
	  82 01 01
	  83 01 01
	  30 17
	  30 15
	
	  80 01 80
	
	  86 10 112233445566778899AABBCCDDEEFF10
	
	  30 22

```
      -- C-MAC + R-MAC
      keyUsageQualifier '34'H,                                  95 01 34
      -- may be used by SD and application
      keyAccess '00'H,
      -- MAC key
      keyIdentifier '02'H,                                      82 01 02
      keyVersionNumber '01'H,                                   83 01 01
      keyCompontents {                                          30 17
        {                                                         30 15
          -- DES mode implicitly known(as an example)
          keyType '80'H,                                            80 01 80
          -- This value may be freely changed
          keyData '112233445566778899AABBCCDDEEFF10'H               86 10 112233445566778899AABBCCDDEEFF10
        }
      }
    },
    {                                                         30 22
      -- C-DEK + R-DEK
      keyUsageQualifier 'C8'H,                                  95 01 C8
      -- may be used by SD and application
      keyAccess '00'H,
      -- data ENC key
      keyIdentifier '03'H,                                      82 01 03
      keyVersionNumber '01'H,                                   83 01 01
      keyCompontents {                                          30 17
        {                                                         30 15
          -- DES mode implicitly known (as an example)
          keyType '80'H,                                            80 01 80
          -- This value may be freely changed
          keyData '112233445566778899AABBCCDDEEFF10'H               86 10 112233445566778899AABBCCDDEEFF10
        }
      }
    }
  }
}
```

11.2.11.2. PE MNO SD compliant UICC Configuration (Example 2)

ASN.1 Format	DER TLV encoding
mnoSdCompValue ProfileElement ::= securityDomain : {	A6 82 01 99
sd-Header {	A0 05
mandated NULL,	80 00
identification 7	81 01 07
},	
instance {	A1 48
applicationLoadPackageAID 'A0000001515350'H,	4F 07 A0 00 00 01 51 53 50
classAID 'A000000151535041'H,	4F 08 A0 00 00 01 51 53 50 41
instanceAID 'A000000151000000'H,	4F 08 A0 00 00 01 51 00 00 00
applicationPrivileges '82FC80'H,	82 03 82 FC

```
    -- Secured
    lifeCycleState '0F'H,                                   80 83 01 0F
    -- SCP80 supported and SCP03 mode 70
    applicationSpecificParametersC9
'810280008102037082 01F08701F0'H,
    -- other parameters may be necessary                    C9 0E 81 02 80 00 81 02 03 70 82 01 F0 87 01 F0
    applicationParameters {
      -- TAR: B20100, MSL: 12                               EA 11 80 0F
      uiccToolkitApplicationSpecificParametersField
         '01000001000000020112 03B2010000'H                 01 00 00 01 00 00 00 02 01 12 03 B2 01 00 00
    }
  },
  keyList {
    {                                                       A2 82 01 26
         -- KeySet SCP80 KVN 01 Kid 01                      30 22
      -- C-ENC + R-ENC
      keyUsageQualifier '38'H,
      -- may be used by SD and application                   95 01 38
      keyAccess '00'H,
      -- ENC key
      keyIdentifier '01'H,
      keyVersionNumber '01'H,                               82 01 01
      keyCompontents {                                      83 01 01
        {                                                   30 17
          -- DES mode implicitly known (as an example)        30 15
          keyType '80'H,
          -- This value may be freely changed               80 01 80
          keyData '11223344 5566778899AABBCCDDEEFF10'H
        }                                                   86 10
      }                                                     11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF 10
    },
    {
         -- KeySet SCP80 KVN 01 Kid 02
      -- C-MAC + R-MAC
      keyUsageQualifier '34'H,
      -- may be used by SD and application                 30 22
      keyAccess '00'H,                                      95 01 34
      -- MAC key
      keyIdentifier '02'H,
      keyVersionNumber '01'H,                               82 01 02
      keyCompontents {                                      83 01 01
        {                                                   30 17
          -- DES mode implicitly known(as an example)        30 15
          keyType '80'H,
          -- This value may be freely changed               80 01 80
          keyData '11223344 5566778899AABBCCDDEEFF10'H
        }                                                   86 10
      }                                                     11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF 10
    },
    {
```

```
          -- KeySet SCP80 KVN 01 Kid 03                          30 22
      -- C-DEK + R-DEK
      keyUsageQualifier 'C8'H,
      -- may be used by SD and application                       95 01 C8
      keyAccess '00'H,
      -- data ENC key
      keyIdentifier '03'H,
      keyVersionNumber '01'H,                                    82 01 03
      keyCompontents {                                           83 01 01
        {                                                        30 17
          -- DES mode implicitly known (as an example)             30 15
          keyType '80'H,
          -- This value may be freely changed                    80 01 80
          keyData '11223344556677889AABBCCDDEEFF10'H
        }                                                        86 10
      }                                                          11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF 10
    },
    {
         -- KeySet SCP03 KVN 30 Kid 01                          30 22
      -- C-ENC + R-ENC
      keyUsageQualifier '38'H,
      -- may be used by SD and application                       95 01 38
      keyAccess '00'H,
      -- ENC key
      keyIdentifier '01'H,
      keyVersionNumber '30'H,                                    82 01 01
      keyCompontents {                                           83 01 30
        {                                                        30 17
          -- AES (16, 24, or 32 long keys)                       30 15
          keyType '88'H,
          -- This value may be freely changed                    80 01 88
          keyData '11111111030303031111111103030303'H
        }                                                        86 10
      }                                                          11 11 11 11 03 03 03 03 11 11 11 11 03 03 03 03
    },
    {
         -- KeySet SCP03 KVN 30 Kid 02                          30 22
      -- C-MAC + R-MAC
      keyUsageQualifier '34'H,
      -- may be used by SD and application                       95 01 34
      keyAccess '00'H,
      -- MAC key
      keyIdentifier '02'H,
      keyVersionNumber '30'H,                                    82 01 02
      keyCompontents {                                           83 01 30
        {                                                        30 17
          -- AES (16, 24, or 32 long keys)                       30 15
          keyType '88'H,
          -- This value may be freely changed                    80 01 88
          keyData '22222222030303032222222203030303'H
```

```
        }                                                                   86 10
      }                                                                     22 22 22 22 03 03 03 03 22 22 22 22 03 03 03 03
    },
    {
         -- KeySet SCP03 KVN 30 Kid 03                                     30 22
      -- C-DEK + R-DEK
      keyUsageQualifier 'C8'H,
      -- may be used by SD and application                                 95 01 C8
      keyAccess '00'H,
      -- data ENC key
      keyIdentifier '03'H,
      keyVersionNumber '30'H,                                             82 01 03
      keyCompontents {                                                    83 01 30
        {                                                                 30 17
          -- AES (16, 24, or 32 long keys)                                30 15
          keyType '88'H,
          -- This value may be freely changed                             80 01 88
          keyData '33333333030303033333333303030303'H
        }                                                                 86 10
      }                                                                   33 33 33 33 03 03 03 03 33 33 33 33 03 03 03 03
    },
      {-- Token AES scheme as example
      keyUsageQualifier  '81'H,                                           30 25
      -- may be used by SD                                                95 01 81
      keyAccess  '01'H,
      -- Key Id 01                                                        96 01 01
      keyIdentifier  '01'H,
      keyVersionNumber '70'H,                                             82 01 01
      keyCompontents  {                                                   83 01 70
      {                                                                   30 17
      -- AES (16, 24, or 32 long keys)                                    30 15
      keyType  '88'H,
      -- This value may be freely changed                                 80 01 88
      keyData  'CDFE56B7B72FAE6A047341F003D7A48D'H
            }                                                             86 10
        }                                                                 CD FE 56 B7 B7 2F AE 6A 04 73 41 F0 03 D7 A4 8D
  },
      {-- Receipt the AES scheme shall be supported
      keyUsageQualifier  '44'H,                                           30 25
      -- may be used by SD                                                95 01 44
      keyAccess  '01'H,
      -- Key Id 01                                                        96 01 01
      keyIdentifier  '01'H,
      keyVersionNumber '71'H,                                             82 01 01
      keyCompontents  {                                                   83 01 71
      {                                                                   30 17
      -- AES (16, 24, or 32 long keys)                                    30 15
      keyType  '88'H,
      -- This value may be freely changed                                 80 01 88
      keyData  '11121314212223243132333441424344'H
```

} **}** **}** **},** **sdPersoData {** **'0070084206606162636465'H,** **'00700A450814341280 14341280'H** **}** **}**	86 10 11 12 13 14 21 22 23 24 31 32 33 34 41 42 43 44 A3 1C 04 0B 00 70 08 42 06 60 61 62 63 64 65 04 0D 00 70 0A 45 08 14 34 12 80 14 34 12 80

11.2.12 PE SSD

ASN.1 Format	DER TLV encoding

ssdValue ProfileElement ::= securityDomain : {
sd-Header {
mandated NULL,
identification 8
},
instance {
applicationLoadPackageAID 'A0000001515350'H,
classAID 'A000000151535041'H,
instanceAID 'A00000055910100102736456616C7565'H,
-- by default extradited under MNO extraditeSecurityDomainAID
'A000000151000000'H
-- Security Domain + Trusted Path
applicationPrivileges '808000'H,
-- Personalized
lifeCycleState '0F'H,
-- SCP80 supported, extradition supported
applicationSpecificParametersC9 '810280008201F0'H,
applicationParameters {
-- TAR: 6C7565, MSL: 12
uiccToolkitApplicationSpecificParametersField
'01000001000000020112036C756500'H
}
},
keyList {
{
-- C-ENC + R-ENC
keyUsageQualifier '38'H,
-- may be used by SD and application
keyAccess '00'H,
-- ENC key
keyIdentifier '01'H,
keyVersionNumber '01'H,
keyCompontents {
{
	A6 81C0
	A0 05
	80 00
	81 01 08
	
	A1 49
	4F 07 A0000001515350
	4F 08 A000000151535041
	4F 10 A00000055910100102736456616C7565
	
	82 03 808000
	
	83 01 0F
	
	C9 07 810280008201F0
	EA 11
	
	80 0F 01000001000000020112036C756500
	
	A2 6C
	30 22
	
	95 01 38
	
	82 01 01
	83 01 01
	
	30 17

```
          -- DES mode implicitly known (as an example)
          keyType '80'H,
          -- This value may be freely changed
          keyData '8811223344556677881122334455667788'H
        }
      }
    },
    {
      -- C-MAC + R-MAC
      keyUsageQualifier '34'H,
      -- keyAccess '00'H, may be used by SD and application
      -- MAC key
      keyIdentifier '02'H,
      keyVersionNumber '01'H,
      keyCompontents {
        {
          -- DES mode implicitly known (as an example)
          keyType '80'H,
          -- This value may be freely changed
          keyData '8811223344556677881122334455667788'H
        }
      }
    },
    {
      -- C-DEK + R-DEK
      keyUsageQualifier 'C8'H,
      -- keyAccess '00'H, may be used by SD and application
      -- data ENC key
      keyIdentifier '03'H,
      keyVersionNumber '01'H,
      keyCompontents {
        {
          -- DES mode implicitly known (as an example)
          keyType '80'H,
          -- This value may be freely changed
          keyData '8811223344556677881122334455667788'H
        }
      }
    }
  }
}
```

```
        30 15
          80 01 80
          86 10 8811223344556677881122334455667788
    30 22
      95 01 34
      82 01 02
      83 01 01
      30 17
        30 15
          80 01 80
          86 10 8811223344556677881122334455667788
    30 22
      95 01 C8
      82 01 03
      83 01 01
      30 17
        30 15
          80 01 80
          86 10 8811223344556677881122334455667788
```

11.2.13 PE APPLICATION 1

ASN.1 Format	DER TLV encoding
`applet1 ProfileElement ::= application : {`	`A8 820263`
`  app-Header {`	`  A0 05`

```
    mandated NULL,
    identification 9
  },
  loadBlock {
    loadPackageAID 'A000000559101001'H,
    loadBlockObject '
01002EDECAFFED020204000108A0000005591010011B636F6D2F67736D612F
65756963632F746573742F6170706C657431020021002E0021000F003B002A
00210066000A000E0000008A040F000000000000004010004003B04030107A0
000000620101000110A0000000090005FFFFFFFF8912000000010110A00000
00871005FFFFFFFF8913200000000107A0000000620001030000F010BA00000
055910100111223300080600210000448003000FF00050400000033FFFF0030
004081070082000080020081080108070066000110188C00007A04328F0001
3D8C00022E181D252904160461081B8B0003700C1B181D044116048B00041B
8C00057A00207A02301E046B071967041877017702211D7500160001000200
098D00062D1A048E0200071770027A02108D0008058E020009007A08000A00
00000000000000000005002A000A0680030001000200060000010380030103
80030206000005A06810F000181040006811000018109000900E0000000A05
06040E0C04200709050B008A010001000204000000068107820080028108 00
8100010016000500000000010900080018002600000000070100300023000 1
000000000501003300270008000000008010040002E001800000000FF0200
5A0016000A000000000000A0016FFFF0016001600180016001BFFFF001FFFFF
011004B43105681040056810900668 00A10B6800636800200241'H
  },
  instanceList {
    {
      applicationLoadPackageAID 'A000000559101001'H,
      classAID 'A0000005591010011 12233'H,
      instanceAID 'A000000559101001 11223301'H,
      applicationPrivileges '000000'H,
      applicationSpecificParametersC9 '00'H,
      applicationParameters {
        uiccToolkitApplicationSpecificParametersField
          -- TAR: 112233
          '0100000000000311223300'H
      }
    }
  }
}
```

```
    80 00
    81 01 09

  A1 820218
    4F 08 A000000559101001
    C4 82020A
01002EDECAFFED020204000108A0000005591010011B636F6D2F67736D612F
65756963632F746573742F6170706C657431020021002E0021000F003B002A
00210066000A000E0000008A040F000000000000004010004003B04030107A0
000000620101000110A0000000090005FFFFFFFF8912000000010110A00000
00871005FFFFFFFF8913200000000107A0000000620001030000F010BA00000
055910100111223300080600210000448003000FF00050400000033FFFF0030
004081070082000080020081080108070066000110188C00007A04328F0001
3D8C00022E181D252904160461081B8B0003700C1B181D044116048B00041B
8C00057A00207A02301E046B071967041877017702211D7500160001000200
098D00062D1A048E0200071770027A02108D0008058E020009007A08000A00
00000000000000000005002A000A0680030001000200060000010380030103
80030206000005A06810F000181040006811000018109000900E0000000A05
06040E0C04200709050B008A010001000204000000068107820080028108 00
8100010016000500000000010900080018002600000000070100300023000 1
000000000501003300270008000000008010040002E001800000000FF0200
5A0016000A000000000000A0016FFFF0016001600180016001BFFFF001FFFFF
011004B43105681040056810900668 00A10B6800636800200241

  A2 3E
    30 3C
      4F 08 A000000559101001
      4F 0B A0000005591010011 12233
      4F 0C A000000559101001 11223301
      82 03 000000
      C9 01 00
      EA 0D


        80 0B 0100000000000311223300
```

11.2.14 PE APPLICATION 2

ASN.1 Format	DER TLV encoding

applet2 ProfileElement ::= application : {
  app-Header {
  identification 10
  },
	A8 820194
	  A0 03
	  81 01 0A

```
  loadBlock {
    loadPackageAID 'A000000559101003'H,
    loadBlockObject '
01002EDECAFFED020204000108A0000005591010031B636F6D2F67736D612F
65756963632F746573742F6170706C657433020021002E0021000F00150016
000E002F000A00090000004301F4000000000000020100040015020301 07A0
0000006201010001 07A0000006200010300 0F010BA0000005591010034455
66000806000E000000800300FF0007010000002C07002F000110188C00007A
04328F00013D8C00022E181D252904160461081B8B0003700C1B181D044116
048B00047A00207A08000A0000000000000000000000050016000506800300 01
0002000600000103800301038003020900090000000505060 40E0C0B004301
0001000200000000030081000 1000C0005000000000109000 8000E00220000
00000701002C0011000100000000000 5000CFFFF000C000C000E011004B431
066800A1'H
  },
  instanceList {
    {
      applicationLoadPackageAID 'A000000559101003'H,
      classAID 'A00000055910100 3445566'H,
      instanceAID 'A00000055910100344556601'H,
  extraditeSecurityDomainAID 'A00000055910100102736456616C7565'H,
      applicationPrivileges '000000'H,
      applicationSpecificParametersC9 '00'H
    }
  }
}
```

```
  A1 820148
    4F 08 A000000559101003
    C4 82013A
01002EDECAFFED020204000108A0000005591010031B636F6D2F67736D612F
65756963632F746573742F6170706C657433020021002E0021000F00150016
000E002F000A00090000004301F4000000000000020100040015020301 07A0
0000006201010001 07A0000006200010300 0F010BA0000005591010034455
66000806000E000000800300FF0007010000002C07002F000110188C00007A
04328F00013D8C00022E181D252904160461081B8B0003700C1B181D044116
048B00047A00207A08000A0000000000000000000000050016000506800300 01
0002000600000103800301038003020900090000000505060 40E0C0B004301
0001000200000000030081000 1000C0005000000000109000 8000E00220000
00000701002C0011000100000000000 5000CFFFF000C000C000E011004B431
066800A1

  A2 41
    30 3F
      4F 08 A000000559101003
      4F 0B A00000055910100 3445566
      4F 0C A00000055910100344556601
      4F 10 A00000055910100102736456616C7565
      82 03 000000
      C9 01 00
```

11.2.15 PE RFM UICC

ASN.1 Format	DER TLV encoding
`rfmUicc ProfileElement ::= rfm : {`	`A7 20`
`  rfm-header {`	`  A0 03`
`    identification 11`	`    81 01 0B`
`  },`	
`  -- Instance AID`	
`  instanceAID 'A00000055910100001'H,`	`  4F 09 A00000055910100001`
`  tarList {`	`  A0 05`
`    'B00000'H`	`    04 03 B00000`
`  },`	
`  -- cryptographic checksum + counter higher`	
`  minimumSecurityLevel '12'H,`	`  81 01 12`
`  -- full access`	
`  uiccAccessDomain '00'H,`	`  04 01 00`
`  -- full access`	
`  uiccAdminAccessDomain '00'H`	`  04 01 00`
`}`	

11.2.16 PE RFM USIM

ASN.1 Format	DER TLV encoding

rfmUsim ProfileElement ::= rfm : {
  rfm-header {
  identification 12
  },
  -- Instance AID
  instanceAID 'A00000055910100002'H,
  tarList {
  'B00020'H
  },
  -- cryptographic checksum + counter higher
  minimumSecurityLevel '12'H,
  -- full access
  uiccAccessDomain '00'H,
  -- full access
  uiccAdminAccessDomain '00'H,
  adfRFMAccess {
  adfAID 'A0000000871002FF33FF018900000100'H,
  -- UICC access condition: ADM1
  adfAccessDomain '02000100'H,
  -- UICC access condition: ADM1
  adfAdminAccessDomain '02000100'H
  }
}
	A7 40
	  A0 03
	  81 01 0C
	
	  4F 09 A00000055910100002
	  A0 05
	  04 03 B00020
	
	  81 01 12
	
	  04 01 00
	
	  04 01 00
	  30 1E
	  80 10 A0000000871002FF33FF018900000100
	
	  81 04 02000100
	
	  82 04 02000100

11.2.17 PE END

ASN.1 Format	DER TLV encoding

endValue ProfileElement ::= end : {
** end-header {**
** mandated NULL,**
** identification 99**
** }**
}
	AA 07
	  A0 05
	  80 00
	  81 01 63

11.2.18 EUICC RESPONSE

The following eUICC Response shows how a warning can be reported.

ASN.1 Format	DER TLV encoding

`respValue EUICCResponse ::= {`
`  peStatus {`
`    {`
`-- Library not supported in Application 2 loaded in the SSD`
`      status lib-not-supported,`
`      identification 10`
`    }`
`  }`
`}`
	`30 0A`
	`  A0 08`
	`    30 06`
	
	`      80 01 08`
	`      81 01 0A`

12. ANNEX D (Informative): Document history

The table below indicates changes that have been incorporated into the present document since it was created by SIMalliance.

Version	Date	Brief Description of Changes
V1.0	26/06/2015	1st Release of Document
V1.01	06/07/2015	Addition of SIMalliance OID value
V2.0	18/04/2016	- Clarifications and editorial corrections - Update of references - Support of MF creation using generic file management mechanism - Addition of indication for support of 128 bit and 256 bits TUAK key length - Addition of multiple USIM/ISIM/CSIM support indication - Addition of optional connectivity parameters in the Profile header - Clarification on usage of short file ID - Calrification on Figure 2 about "File" object processing - Modification and clarification in template modification rules: pinStatusTemplateDO becomes mandatory for ADF and DF, efFileSize and proprietaryEFInfo become conditional for EF links - Addition of EF_{UMPC} in the MF - Clarification on PE usage rules and addition of rule for PE-Application - Correction in PE-OPT-USIM (Addition of missing ef-vbsca) - Corrections and clarifications in AKA Parameters PE - Correction for PIN and PUK coding for maxNumOfAttemps-retryNumLeft - Clarifications for MNO-SD creation - Addition of independent SD beside the MNO-SD - Corrections and clarifications in key personalisation - Clarification on SD personalisation - Corrections in RAM / OTA HTTPs Configuration - Correction of default value of life cycle state - Additions/corrections for the support of contactless applications - Addition of missing instance AID in RFM parameters. Clarification on the usage of Tar list - Clarification and addition of status usage in the eUICC response - Clean up of Access conditions, addition of values and addition of access conditions for ADF and DF - Some corrections in templates - Revision of the example section
V2.1	24/02/2017	- SQN Clarification about authCounterMax - Clarification of Error Management Rules - Correction of File Size for LF files in templates - Fill and Repeat Pattern Size Limitation - PE-Application Dependency Rule - Security Domain RAM/OTA HTTPs Configuration clarification - SQN Clarification - Addition of BER-TLV file type to the ServiceList - Addition of TUAK parameter for the number of iterations of Keccak - Removal of ADF Link Discrepancy between section 8.3.3 and 8.3.5 - Limitation in the encoding of efFileSize

		- Corrections in the example section - Update of references to GP specifications - Addition of SQN array sharing between NAA - Addition of DF Link support to the ServicesList - Addition USIM Test Algorithm for PE AKA parameters - Additional correction of File Size for LF files in templates - Additional example of PE MNO SD compliant with UICC Configuration - Clarification for CSIM parameters - Common files types and fields - Clarification to eUICC-Mandatory-services list and eUICC-Mandatory-GFSTEList - Removal of reference to requirement document - Template Clarification - Addition on Principles - Avoid empty OPTIONAL SEQUENCE OF Elements - GBA and MBMS features clarifications - Clarification of Security Domain Life Cycle State - EXTENSIBILITY IMPLIED clarification - Mandatory templates support - PinStatusTemplateDO clarification - extraditeSecurityDomainAID for MNO-SD - PE-AKA corrections - Keyaccess & KeyUsageQualifier for OTA keys clarifications - CSIM Parameters additional corrections - Use of authentication algorithms references - BER TLV and DF link processing when not supported - Clarification on usage of error codes - Clarification for some RFM parameters - Correction of MNO-SD example1 privileges - FileDescriptor clarification - Minor version Update - Addition of CDMA Parameters example