# WAP Provisioning Smart Card

WAP-186-PROVSC-20010710-a

Version 10-July-2001

**Wireless Application Protocol**
**WAP Provisioning Smart Card Specification**

A list of errata and updates to this document is available from the WAP Forum TM Web site, http://www.wapforum.org/, in the form of SIN documents, which are subject to revision or removal without notice.

# Contents

# 1. Scope

The Wireless Application Protocol (WAP) is a result of continuous work to define an industry-wide specification for developing applications that operate over wireless communication networks. The scope for the WAP Forum is to define a set of specifications to be used by service applications. The wireless market is growing very quickly, and reaching new customers and services. To enable operators and manufacturers to meet the challenges in advanced services, differentiation and fast/flexible service creation WAP Forum defines a set of protocols in transport, security, transaction, session and application layers. For additional information on the WAP architecture, please refer to "*Wireless Application Protocol Architecture Specification*" [WAPARCH].

This document defines the files on a WIM card or on a SIM card that have to be used to store WAP provisioning data. This allows WIM and SIM cards to be manufactured in large quantities, and enables all WAP phones to interact with the storage provisioning framework. It also defines how trusted certificates are stored on a GSM-SIM.

# 2. Document Status

This document is available online in the following formats:

- PDF format at http://www.wapforum.org/.

## 2.1 Copyright Notice



## 2.2 Errata

Known problems associated with this document are published at http://www.wapforum.org/.

## 2.3 Comments

Comments regarding this document can be submitted to the WAP Forum in the manner published at http://www.wapforum.org/.

# 3. References

## 3.1 Normative References


[CREQ] "Specification of WAP conformance requirements", WAP Forum, WAP-221-CREQ, URL: http://www.wapforum.org/

[GSM02.17] Digital cellular telecommunications system (Phase 2+); Subscriber Identity Modules (SIM); Functional characteristics (GSM 02.17 version 7.1.1 Release 1998)

[GSM11.11] Digital cellular Telecommunications system (Phase 2+); Specification of the Subscriber Identity Module - Mobile Equipment (SIM - ME) interface (GSM 11.11 version 7.2.0 Release 1998)

[ISO7816-4] ISO/IEC 7816-4 (1995): "Information technology - Identification cards - Integrated circuit(s) cards with contacts - Part 4: Inter-industry commands for interchange".

[ISO7816-5] ISO/IEC 7816-5 (1994): "Identification cards - Integrated circuit(s) cards with contacts - Part 5: Numbering system and registration procedure for application identifiers".

[PKCS#15] PKCS #15: Cryptographic Token Information Standard", version 1.0, RSA Laboratories, April 1999. URL: ftp://ftp.rsa.com/pub/pkcs/pkcs-15/pkcs15v1.doc

[PROVCONT] "WAP Provisioning Content Type Specification", WAP Forum, WAP-183-PROVCONT, URL: http://www.wapforum.org/

[RFC2119] "Key words for use in RFCs to Indicate Requirement Levels ", S. Bradner, March 1997. URL: http://www.ietf.org/rfc/rfc2119.txt

[WIM] "WAP Identity Module Specification", WAP Forum, WAP-198-WIM-20000218-a, URL: http://www.wapforum.org/

[TS102.221] Smart Cards; UICC-Terminal interface; Physical and logical characteristics (ETSI TS 102 221, R4), URL: http://www.3gpp.org

## 3.2 Informative References


[ISO7816-9] ISO/IEC 7816-9 (2000): "Identification cards - Integrated circuit(s) cards with contacts - Part 9: Additional inter-industry commands and security attributes".

[ISO8824-1] ISO/IEC 8824-1 (1995): “Information technology – Abstract Syntax Notation One (ASN.1) – Specification of basic notation”.

[ISO8825-2] ISO/IEC 8825-2 (1995): “Information technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)”.

[PROVARCH] “WAP Provisioning Architecture Overview Specification”, WAP Forum, WAP-182-PROVARCH, URL: http://www.wapforum.org/

[WAPARCH] “WAP Architecture Specification”, WAP Forum, WAP-100-WAPARCH, URL: http://www.wapforum.org/

[WTLS] “Wireless Transport Security Layer Specification”, WAP Forum, WAP-199-WTLS, URL: http://www.wapforum.org/

# 4. Definitions and Abbreviations

## 4.1 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

## 4.2 Definitions

This section introduces terminology that will be used throughout this document.

**Access conditions**
A set of security attributes associated with a file.

**AID**
Application Identifier. A data element that identifies an application in a card. An application identifier may contain a registered application provider number in which case it is a unique identification for the application. If it contains no application provider number, then this identification may be ambiguous.

**ALW**
Always. Access condition indicating a given function is always accessible.

**AODF**
The Authentication Object Directory Files ([PKCS#15], section 6.5.7) contain directories of authentication objects (e.g. PINs) known to the PKCS#15 application.

**Application**
The implementation of a well-defined and related set of functions that perform useful work on behalf of the user. It may consist of software and or hardware elements and associated user interfaces.

**ASN.1 object**
Abstract Syntax Notation object as defined in [ISO8824-1]. A formal syntax for describing complex data objects.

**ATR**
Answer-to-Reset. Stream of data sent from the card to the reader in response to a RESET condition.

**BER**
Basic Encoding Rules. Rules for encoding an ASN.1 object into a byte sequence.

**Binary Files**
Binary Files are equivalent to transparent files as described in [GSM11.11].

**Cardholder**
The person or entity presenting a smart card for uses.

**Card Issuer**

The organization or entity that owns and provides a smart card product.

**CDF**

Certificate Directory Files ([PKCS#15], section 6.5.5) contain directories of certificates known to the PKCS#15 application.

**CHV**

CardHolder Verification. Also called the PIN. Typically a 4 to 8 digit number entered by the cardholder to verify that the cardholder is authorized to use the card.

**Command**

A message sent by the ME to the card that initiates an action and solicits a response from the card.

**Configuration Context**

A Configuration Context is a set of connectivity and application configurations typically associated with a single TPS. However, the configuration context can also be independent of any TPS. A TPS can be associated with several Configuration Contexts, but a TPS cannot provision a device outside the scope of the Configuration Contexts associated with that particular TPS. In fact, all transactions related to provisioning are restricted to the Configuration Contexts associated with the TPS.

**Connectivity Information**

The information in connectivity provisioning relates to the parameters and means needed to access WAP infrastructure. This includes network bearers, protocols, access point addresses, as well as proxy addresses and Trusted Provisioning Server URL.

**DER**

Distinguished Encoding Rules for encoding ASN.1 objects in byte-sequences. A special case of BER.

**DF**

Dedicated File. A file containing access conditions and, optionally, Elementary Files (EFs) or other Dedicated Files (DFs).

**DODF**

The Data Object Directory Files contain directories of data objects (not keys or certificates) ([PKCS#15], section 6.5.6) known to the PKCS#15 application.

**DODF-wtls**

The Data Object Directory Files contain directories of data objects (not keys or certificates) ([PKCS#15], section 6.5.6) used in WTLS and known to the PKCS#15 application.

**DODF-prov**

The Data Object Directory Files contain directories of data objects (not keys or certificates) ([PKCS#15], section 6.5.6) used in WAP provisioning and known to the PKCS#15 application.

**EF**

Elementary File. A set of data units or records that share the same identifier. It cannot be a parent of another file.

**File identifier**

A 2-byte binary value used to address a file on a smart card.

**Function**

A function contains a command and a response pair.

**ICC**

Integrated Circuit Card. Another name for a smart card.

**MF**

Master File. Mandatory unique dedicated file representing the root of the structure. The MF typically has the file identifier 0x3F00.

**NEV**

An access condition indicating a given function is never accessible.

**ODF**

The mandatory Object Directory File (ODF) ([PKCS#15], section 6.5.1) consists of pointers to other EFs (PrKDFs, PuKDFs, CDFs, DODFs and AODFs), each one containing a directory over PKCS#15 objects of a particular class (here and below, a “directory” means a list of objects).

**Path**

Concatenation of file identifiers without delimitation. The Path type is defined in [ISO7816-4] sub-clause 5.1.2. If the path starts with the MF identifier (0x3F00), it is an absolute path; otherwise it is a relative path. A relative path must start with the identifier of the current DF (or with the identifier '0x3FFF').

**PIN**

Personal Identification Number. See CHV.

**PrKDF**

The Private Key Directory Files ([PKCS#15], section 6.5.2) contain directories of private keys known to the PKCS#15 application.

**PuKDF**

The Public Key Directory Files ([PKCS#15], section 6.5.4) contain directories of public keys known to the PKCS#15 application.

**Record**

A string of bytes within an EF handled as a single entity.

**Record number**

The number, which identifies a record within an EF.

**Smart card**

A device with an embedded microprocessor chip. A smart card is used for storing data and performing typically security related (cryptographic) operations. In WAP context, a smart card may be the GSM Subscriber Identity Module (SIM) or a card used in a secondary card reader of a WAP phone.

**Trusted Proxy**

The trusted (provisioning) proxy has a special position as it acts as a front end to a trusted provisioning server. The trusted proxy is responsible to protect the end-user from malicious configuration information.

**TPS**

A TPS, Trusted Provisioning Server, is a source of provisioning information that can be trusted by a Configuration Context. They are the only entities that are allowed to provision the device with static configurations. In some cases, however, a single TPS is the only server allowed to configure the phone. Provisioning related to a specific TPS is restricted to Configuration Contexts that are associated with this TPS.

**UICC**

Universal ICC. UICC is the ICC defined for the 3G standard [TS102.221].

**WIM**

WAP Identity Module. A tamper-resistant device that is used in performing WTLS and application level security functions, and especially, to store and process information needed for user identification and authentication.

**WTLS**

Wireless Transport Layer Security is the Security layer protocol in the WAP architecture. The WTLS layer operates above the transport protocol layer. The WTLS layer is modular and it depends on the required security level of the given application whether it is used or not. WTLS provides the upper-level layer of WAP with a secure transport service interface that preserves the transport service interface below it. In addition, WTLS provides an interface for managing (e.g. creating and terminating) secure connections.

## 4.3 Abbreviations

For the purposes of this specification the following abbreviations apply.

| | |
|---|---|
| **AID** | Application Identifier |
| **ALW** | Always |
| **AODF** | Authentication Object Directory File |
| **ASN** | Abstract Syntax Notation |
| **ATR** | Answer-to-Reset |
| **BER** | Basic Encoding Rules |
| **CDF** | Certificate Directory File |
| **CHV** | CardHolder Verification |
| **DER** | Distinguished Encoding Rules |
| **DF** | Dedicated File |
| **DIR** | Directory File |
| **DO** | Data Object |
| **DODF** | Data Object Directory File |
| **EF** | Elementary File |
| **ETSI** | European Telecommunication Standardization Institute |
| **GSM** | Global System for Mobile Communication |
| **IC** | Integrated Circuit |
| **ICC** | Integrated Circuit(s) Card |
| **ID** | Identifier |
| **ISO** | International Organization for Standardization |
| **ME** | Mobile Equipment |
| **MF** | Master File |
| **ODF** | Object Directory File |
| **OID** | Object Identifier |

| | |
|---|---|
| **PIN** | Personal Identification Number |
| **PIN-G** | General Personal Identification Number according to [WIM] |
| **PrKDF** | Private Key Directory File |
| **PuKDF** | Public Key Directory File |
| **SIM** | Subscriber Identity Module |
| **TPS** | Trusted Provisioning Server |
| **URI** | Uniform Resource Identifier |
| **UCS2** | Universal two byte coded Character Set |
| **UICC** | Universal Integrated Circuit(s) Card |
| **WAP** | Wireless Application Protocol |
| **WIM** | WAP Identity Module |
| **WTLS** | Wireless Transport Layer Security |

# 5. Architecture

The compatibility between different browsers with respect to the infrastructure (including the Smart Card) is created by defining the file structures in a WIM or SIM card, and by defining the storage framework inside the files.

A generic "WAP file system" solution is defined. It provides a very flexible framework that can be used to tailor the set-up to the needs of the carrier and the user. It can be used both for basic configurations and for generic storage of persistent information.

The information stored in the files Bootstrap, Config1 and Config2 is of type application/vnd.wap.connectivity-wbxml.

## 5.1 Configuration Concept

The ME is able to access a number of separate files. The files can have different content as well as different read/write access rights.

The files required to enable WAP provisioning storage on the Smart Card are the following:

- Bootstrap File: used to store connectivity information that cannot be changed by the provisioning agent, i.e. by the ME. This file can only be modified by the card issuer.

- Config1 File: used to store connectivity information that can be changed by the provisioning agent, i.e. by the ME. Then, the user can modify connectivity parameters stored in this file in entering the correct enabled PIN (see section 9.5).

- Config2 File: used to store connectivity information that can be changed by the provisioning agent, i.e. by the ME. Then, the user can modify connectivity parameters stored in this file.

The use of multiple files enables the use of the Smart Card file access features to protect part of the configuration data from change by the ME (browser).

The smart card MUST support at least one of provisioning files (Bootstrap, Config1, Config2).
The ME MUST support all provisioning files.

Any provisioning file may contain information on how to connect to the TPS (Trusted Provisioning Server) as defined in [PROVCONT].

## 5.2 Support of Different Smart Card Capabilities

The specification supports a number of different capabilities from the Smart Card point of view:

- Smart cards with WIM functionality
- Smart cards with WIM functionality in addition to GSM SIM functionality
- Smart cards with only GSM SIM functionality

### 5.2.1 Generic Behaviour

The browser MUST use the default provisioning parameters from the first available provisioning files in the following order:

- Provisioning parameters on the WIM physically present on the active SIM,
- Provisioning parameters on the active SIM,

The active SIM is the SIM card selected as defined in [GSM02.17].

Other non-default provisioning data MAY be read from any available WIM, SIM. The reading of this information is implementation dependent.

Trusted Certificates can be read in any order.

# 6. WAP Provisioning Smart Card (ICC)

The information format for WAP Provisioning is based on [PKCS#15] specification. The card operations that are relevant for provisioning include:

- Application selection
- Cardholder verification
- File access (select file, read, write)

The [PKCS#15] specification defines a set of files. Within the PKCS#15 application, the starting point to access these files is the Object Directory File (ODF). The EF(ODF) contains pointers to other directory files. These directory files contain information on different types of objects (keys, certificates, authentication objects (PIN), data objects, etc).

EF(ODF) contains pointers to one or more Data Object Directory Files (DODF). Each DODF is regarded as the directory of data objects known to the PKCS#15 application. For the purposes of WAP provisioning, EF(DODF-prov) contains pointers the provisioning data objects, namely Bootstrap File, Config1 File and Config2 File.

The WAP provisioning data (provisioning files) are stored as PKCS#15 opaque data objects. The WAP provisioning files are located under the PKCS#15 DF and it is up to the card issuer to decide their identifier and their location.

## 6.1 Object Directory File, EF(ODF)

The EF(ODF) MUST contain the record describing the DODF-prov. The EF(ODF) can be read but it MUST NOT be modifiable by the user.

The EF(ODF) is described in section 9.1 and [PKCS#15].

Informative note 1: If a path starts with 3F00, it is an absolute path (starting from root).

## 6.2 Provisioning Data Object Directory File, EF(DODF-prov)

The EF(DODF-prov) MUST contain information on provisioning objects:

- Readable label describing the provisioning document (`PKCS15CommonObjectAttributes.label`). The ME could display this label to the user.
- Flags indicating whether the provisioning document is private (i.e., is protected with a PIN) and/or modifiable (`PKCS15CommonObjectAttributes.flags`). The card issuer decides whether or not a file is private (it does not need to be if it does not contain any sensitive information)
- Reference to a PIN used to protect this object (`PKCS15CommonObjectAttributes.authId`)
- Object identifier indicating a WAP provisioning object and the type of the provisioning object (`PKCS15CommonDataObjectAttributes.applicationOID`)
- Pointer to the contents of the provisioning document (`PKCS15Path.path`)

The EF(DODF-prov) MUST contain the types of provisioning documents (indicated using object identifiers) to be used by the ME. The following types are described in this specification:

- Bootstrap
- Config1
- Config2

If a type exists on the card but it is not in the EF(DODF-prov) then this type MUST NOT be used.

The contents of the provisioning document are defined in [PROVCONT].

A dedicated OID is required and defined for each provisioning file:

- `Bootstrap OID = { joint-isu-itu-t(2) identified-organizations(23) wap(43) provisioning(5) bootstrap(1)}`
- `Config1 OID = { joint-isu-itu-t(2) identified-organizations(23) wap(43) provisioning(5) configuration_1(2)}`
- `Config2 OID = { joint-isu-itu-t(2) identified-organizations(23) wap(43) provisioning(5) configuration_2(3)}`

The ME MUST use the OID to distinguish the EF(DODF-prov) from the EF(DODF-wtls).

The WAP provisioning data are located under the PKCS#15 directory allowing the card issuer to decide the identifiers and the file locations. General data object attributes and associated pointers are located in the EF(DODF-prov). The EF(DODF-prov) can be read but it MUST NOT be modifiable by the user.

The EF(DODF-prov) is described in section 9.3 and [PKCS#15].

# 7. WAP provisioning data on WIM

This chapter specifies a special case of the WAP provisioning in the smart card supporting a WIM.
This chapter deals with provisioning data only. For handling of trusted certificates see [WIM].

## 7.1 WAP provisioning data on WIM card only

### 7.1.1 Introduction

The WAP Identity Module (WIM) specification [WIM] defines service primitives for the WIM and information format based on [PKCS#15] specification. The WIM specification also specifies a mapping of the service primitives to smart card commands, so that a WIM can be implemented as a smart card application.

In the WIM application DF(PKCS#15) contains at least an Authentication Object Directory File (AODF), a Certificate Directory File (CDF), and a Data Object Directory File concerned with the persistent storage of WTLS session data (DODF-wtls).

For WAP provisioning an additional DODF MUST be supported, namely DODF-prov as described in Section 6.2

### 7.1.2 File Overview

The file structure for the WAP provisioning data within the WIM application is described below.

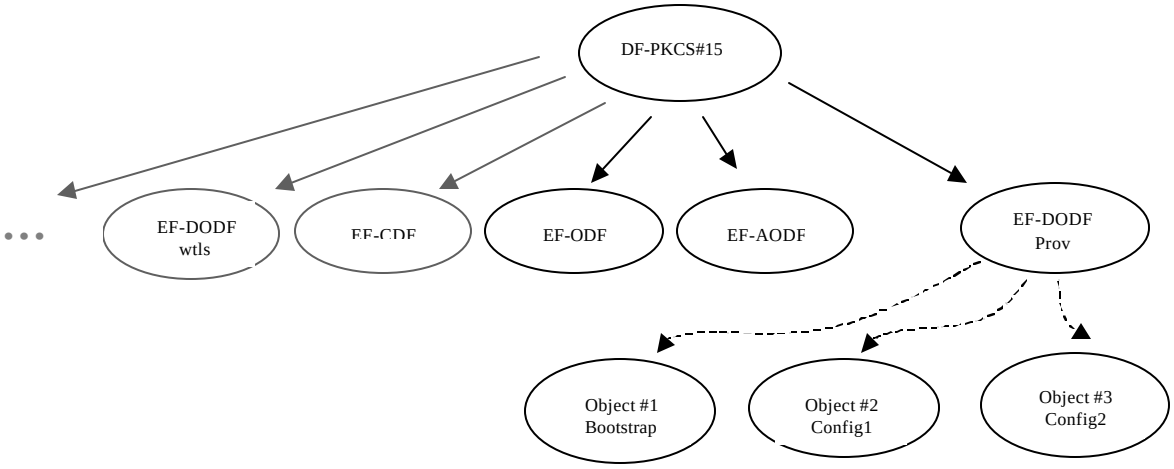


**Figure 1:** File structure for WAP provisioning data on WIM card

### 7.1.3 Access method

WIM commands Read Binary and Update Binary, as defined in [WIM], are used to access WAP provisioning data.

### 7.1.4 Access Conditions

The ME is informed of the access conditions by reading the DODF-prov file in order to know whether objects are private or public. If the object is private then the ME reads the AODF that contains generic authentication object attributes such as PIN length, PIN padding character, etc. The AODF contains pointers to the DF in which the PIN file resides.

Access conditions for files are described in the chapter 9 and the PIN reference format is described in section A.6.

In the case where access conditions require PIN verification, the access rights for provisioning files stored within a WIM card are granted in verifying the PIN-G as defined in the WIM specification [WIM].

## 7.2 WAP provisioning data on SIM-WIM card

### 7.2.1 Introduction

The SIM-WIM provisioning is a special case of the WIM provisioning described in the previous section. The WAP data provisioning MUST be stored in the PKCS#15 structure of the WIM application.
The WAP provisioning data are located under the PKCS#15 directory allowing the card issuer to decide the identifiers and the file locations.

## 7.2.2 Files Overview

The files used for WAP provisioning on SIM-WIM card are the same as in the WIM provisioning.

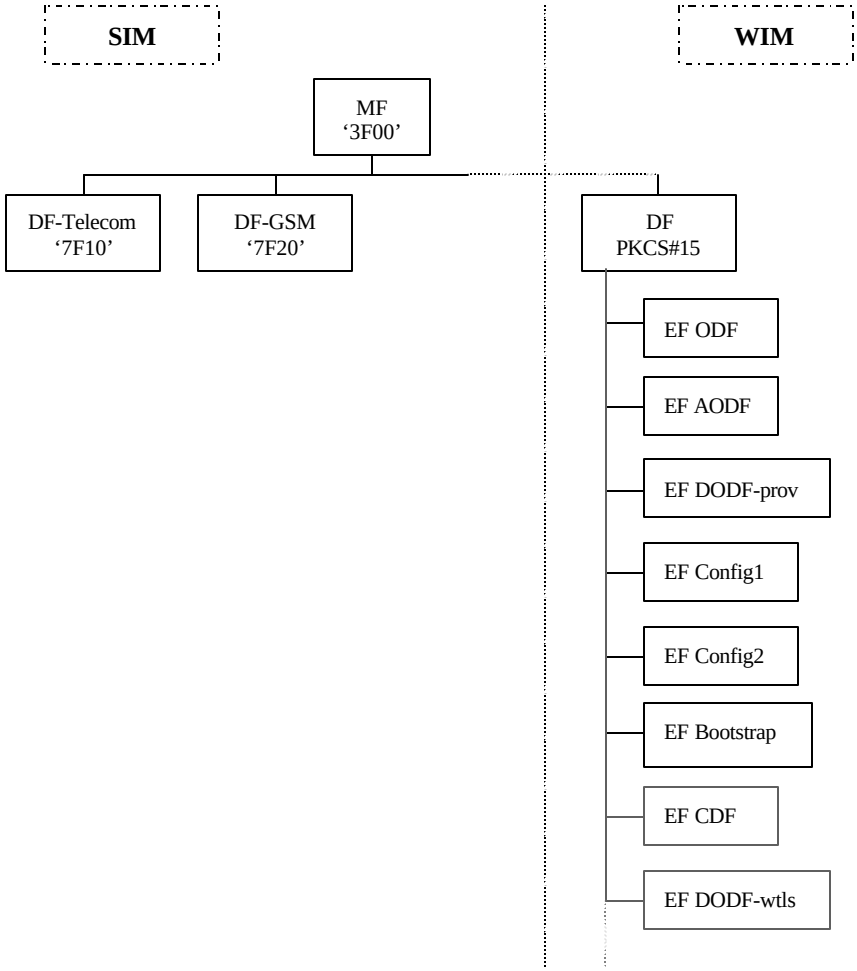


**Figure 2:** File structure for WAP provisioning data on SIM-WIM card

## 7.2.3 Access Method

WIM commands Read Binary and Update Binary, as defined in [WIM], are used to access WAP provisioning data.

### 7.2.4 Access Conditions

The ME is informed of the access conditions by reading the DODF-prov file in order to know whether objects are private or public. If the object is private then the ME reads the AODF that contains generic authentication object attributes such as PIN length, PIN padding character, etc. The AODF contains pointers to the DF in which the PIN file resides.

Access conditions for files are described in the chapter 9.

For the SIM/WIM card either a Global PIN or an application local PIN can be used to limit access to the provisioning files. The usage of a card Global PIN allows the card issuer to avoid the introduction of an additional PIN that the user should remember.

For the SIM/WIM card, the Global PIN is the GSM CHV1 and its pin reference is '01' (indicated in `PKCS#15PinAttributes.pinReference`).

As described in section A.6, bit8 of Reference P2 is set to 0 for a card Global PIN and is set to 1 for an application local PIN.

The ME is informed of the access conditions by reading the DODF file. The ME finds the PIN reference in the PKCS#15 AODF for provisioning (`PKCS#15PinAttributes.pinReference`). If the PIN reference is the card Global PIN and CHV1 was verified, access conditions are granted.

In the case where access conditions require PIN verification, the access rights for provisioning files stored within the WIM part of a SIM-WIM card are granted by verifying the Global PIN which is in this specific case the CHV1, as defined in [GSM11.11] or by verifying the application local PIN-G, as defined in [WIM].

# 8. WAP data on GSM-SIM card

## 8.1 Introduction

This section is to describe the structure for storing provisioning and bootstrapping data and trusted certificates on the WIM-less SIM card.

Trusted Certificates on the SIM are 'read only' and cannot be changed by the ME.

The ME MUST read the EF(DIR) file indicating the presence of the WAP data application. The EF(ODF) and EF(DODF-prov) MUST be used by the ME to determine which WAP provisioning files are available on the SIM. The EF(ODF) and EF(CDF) MUST be used by the ME to determine which trusted certificates are available on the SIM.

WAP provisioning files and trusted certificates will be located under the DF(PKCS#15) and it is up to the card issuer to decide their location.

The EF(DIR) (ID '2F00') MUST be located under the master file as defined in [ISO7816-5] specification.

To get the DF(PKCS#15) identifier, the ME MUST read the EF(DIR). The ME MUST use the indirect selection method as defined in [GSM11.11] to select the DF(PKCS#15).

Informative note 2: The recommended format of EF(DIR) is linear fixed record in order to be in line with 3G TS 31.101 specification [TS102.221].

Informative note 3: In the case of WAP provisioning in the UICC, the direct application selection method with the PKCS#15 AID will be used as described in [TS102.221]. If the EF(DIR) only contains the PKCS#15 AID and not the path, then the ME will use direct application selection method.

## 8.2 Files Overview

The file structure for the WAP provisioning data within the GSM-SIM card is described below.

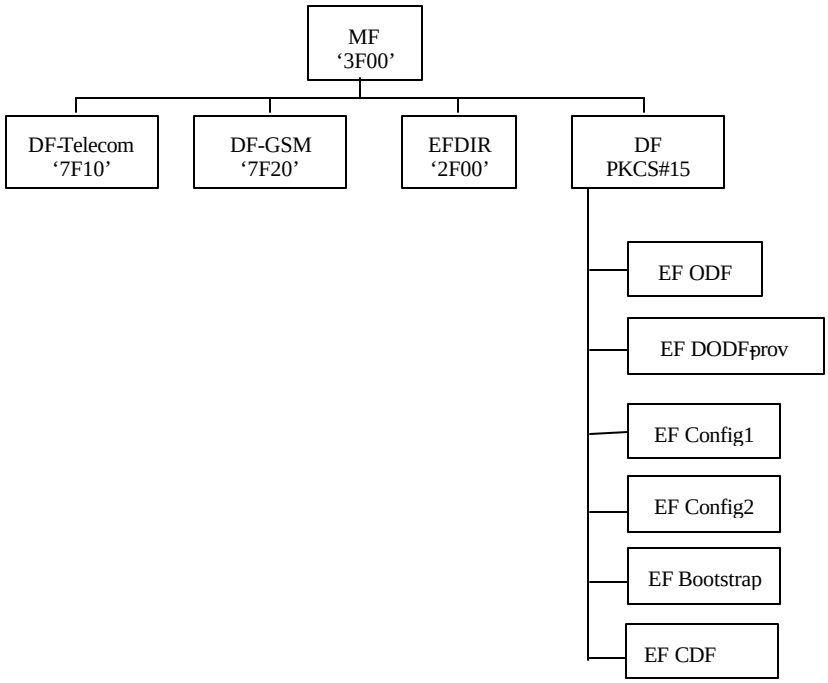


**Figure 3:** File structure for WAP data on GSM-SIM card

## 8.3 Access Method

SIM commands Read Binary and Update Binary, as defined in [GSM11.11], are used to access WAP provisioning data.

## 8.4 Access Conditions

The ME is informed of the access conditions by reading CDF and DODF-prov files in order to know whether objects are private or public. If the object is private then the ME implicitly knows that the CHV1 must be verified.

Access conditions for files are proposed in the chapter 9.

In the case where access conditions require PIN verification, the access rights for provisioning files stored within a GSM-SIM card are granted in verifying the CHV1 as defined in the GSM 11.11 specification [GSM11.11].

# 9. Files Description

All files defined are binary files as defined in ISO7816-4 specification [ISO7816-4]. These files are read and updated using commands related to the application they belong to either the GSM application or the WIM application. See respective access methods in sections 7.1.3, 7.2.3 and 8.3.

In this section, only files used for the provisioning are described. All others files of the WIM application used for WTLS are described in the specification [WIM].

The file size proposed hereafter is a recommended minimum size. Larger files can be created (or extended later) in order to cope with possible extension of the provisioning file content.

The content of the files is defined separately in [PROVCONT].

## 9.1 EF ODF

The mandatory Object Directory File (ODF) ([PKCS#15], section 6.5.1) contains pointers to other EFs (e.g. DODF-prov), each one containing a directory of PKCS#15 objects of a particular class.

The File ID is specified in [PKCS#15]. The file size is decided by the card issuer.

In the case of WIM and SIM-WIM cards, the EF(ODF) contains, in addition to WIM parameters, pointers to the DODF-prov. The EF(ODF) must be formatted as defined in the [WIM] specification.

In the case of GSM-SIM, the EF(ODF) is described below:

| Identifier: default 0x5031, see [PKCS#15] | Structure: Binary | Mandatory |
|---|---|---|
| File size: decided by the card issuer | Update activity: low | |
| Access Conditions:<br>READ ALW or CHV1 (SIM, See section 8.4)<br>UPDATE ADM<br>INVALIDATE ADM<br>REHABILITATE ADM | | |
| Description | | |
| See sections 6.1, A.2 | | |

## 9.2 EF CDF

An optional Certificate Directory File (CDF) ([PKCS#15], section 6.5.1) contains directories of certificates. A CDF pointed to by a Trusted Certificates field in the ODF, contains references to trusted certificates.

The EF(CDF) must be formatted as defined in the [WIM] specification.

In the case of GSM-SIM, the EF(CDF) is described below:

| Identifier: see ODF | Structure: Binary | Optional |
|---|---|---|
| File size: decided by the card issuer | Update activity: low | |
| Access Conditions:<br>READ ALW<br>UPDATE ADM or NEV<br>INVALIDATE ADM or NEV<br>REHABILITATE ADM or NEV | | |
| Description | | |
| See [WIM] | | |

## 9.3 EF DODF-prov

This Data Object Directory File provisioning contains directories of provisioning data objects ([PKCS#15], section 6.5.6) known to the PKCS#15 application.

The File ID is described in the EF(ODF). The file size depends on the number of provisioning objects stored in the card. Thus, the file size is decided by the card issuer.

| Identifier: See ODF | Structure: Binary | Mandatory |
|---|---|---|
| File size: decided by the card issuer | Update activity: low | |
| Access Conditions:<br>READ ALW / PIN-G (WIM, See section 7.1.4)<br>ALW / CHV1 or PIN-G (SIM-WIM, See section 7.2.4)<br>ALW / CHV1 (SIM, See section 8.4)<br>UPDATE ADM<br>INVALIDATE ADM<br>REHABILITATE ADM | | |
| Description | | |
| See sections 6.2, A.4 | | |

## 9.4 EF Bootstrap

$EF_{Bootstrap}$ can be modified only by the card issuer

$EF_{Bootstrap}$ is initialised by setting all bytes to 'FF'.

| Identifier: See DODF | Structure: Binary | Optional |
|---|---|---|
| Recommended minimum file size: 150 bytes | Update activity: low | |
| Access Conditions:<br>READ ALW / PIN-G (WIM, See section 7.1.4)<br>ALW / CHV1 or PIN-G (SIM-WIM, See section 7.2.4)<br>ALW / CHV1 (SIM, See section 8.4)<br>UPDATE ADM<br>INVALIDATE ADM<br>REHABILITATE ADM | | |
| Description | | |
| See [PROVCONT] | | |

## 9.5 EF Config1

$EF_{Config1}$ can be modified by the user

$EF_{Config1}$ is initialised by setting all bytes to 'FF'.

| Identifier: See DODF | Structure: Binary | Optional |
|---|---|---|
| Recommended minimum file size: 150 bytes | Update activity: low | |
| Access Conditions:<br>READ ALW / PIN-G (WIM, See section 7.1.4)<br>ALW / CHV1 or PIN-G (SIM-WIM, See section 7.2.4)<br>ALW / CHV1 (SIM, See sections 8.4)<br>UPDATE CHV1 or PIN-G (See sections 7.1.4, 7.2.4, 8.4)<br>INVALIDATE ADM<br>REHABILITATE ADM | | |
| Description | | |
| See [PROVCONT] | | |

## 9.6 EF Config2

$EF_{Config2}$ can be modified by the user.

$EF_{Config2}$ can be initialised by setting all bytes to 'FF'.

| Identifier: See DODF | Structure: Binary | Optional |
|---|---|---|
| Recommended minimum file size: 150 bytes | Update activity: low | |
| Access Conditions:<br>READ ALW / PIN-G (WIM, See section 7.1.4)<br>ALW / CHV1 or PIN-G (SIM-WIM, See section 7.2.4)<br>ALW / CHV1 (SIM, See section 8.4)<br>UPDATE ALW / CHV1 or PIN-G (See sections 7.1.4, 7.2.4, 8.4)<br>INVALIDATE ADM<br>REHABILITATE ADM | | |
| Description | | |
| See [PROVCONT] | | |

## 9.7 EF Trusted Certificates

Data syntax is in accordance with [WIM] and access rights are described below:

| Identifier: see CDF | Structure: Binary | Optional |
|---|---|---|
| File size: decided by the card issuer | Update activity: low | |
| Access Conditions:<br>READ ALW<br>UPDATE ADM or NEV<br>INVALIDATE ADM or NEV<br>REHABILITATE ADM or NEV | | |
| Description | | |
| See [WIM] | | |

# 10. Requirements for the ME

The first part of this section concerns the provisioning and reading of trusted certificates on the SIM-WIM, or on the WIM card, and the second one addresses the WAP provisioning and trusted certificates on the GSM-SIM card.

The ME MUST support the WIM provisioning if the ME is a phone supporting the WIM.
The ME MUST support the GSM-SIM provisioning if the ME is a GSM phone.

Informative note 4:
The ME can determine whether the card supports logical channels in checking historical bytes of the ATR, as indicated in [WIM] and as specified in [ISO7816-4].
An example of content for each logical record EF(DIR), EF(ODF) and EF(DODF-prov) is described in the table of appendix A and implementation details are provided in appendix B (Static Conformance Requirement).

## 10.1 Requirements on the WIM or SIM-WIM

To support the WAP provisioning on the WIM and SIM-WIM, the ME MUST perform the following steps:

- Select WIM application (direct application selection), as defined in [WIM],
- Read ODF to locate the DODF-prov,
- Read DODF-prov to locate the provisioning files,
- Read the provisioning files,

The ME MUST support the update binary command in order to allow the update of Config1 or/and Config2 files.

Prior to accessing protected files the ME MUST read the AODF to know PIN references required.

For reading of trusted certificates see [WIM].

## 10.2 Requirements on the GSM-SIM

To support the WAP provisioning and reading of trusted certificates on the SIM, the ME MUST perform the following steps:

- Read EF(DIR) to find the file identifier (and path of the PKCS#15 DF),
- Select PKCS#15 DF (indirect selection), as defined in [GSM11.11],
- Read ODF,
- Read DODF-prov to locate the provisioning files,
- Read the provisioning files,
- Read CDF if available
- Read trusted certificates

The ME MUST support the update binary command in order to allow the update of Config1 or/and Config2 files.

# Appendix A. Informative Notes

## A.1 Example of EF(DIR)

Example contents for a PKCS #15 application template on an IC card using indirect application selection.

Value notation:

```
{
    aid   'A000000063504B43532D3135'H,
    label "PROVISIONING",
    path  '3F007F80'H,
}
```

The recommended value of the optional label field is “PROVISIONING” but this value and its coding (either UTF8 or UCS2) can be changed in order to ensure interoperability with the EF(DIR) described in [TS102.221].

## A.2 Example of EF(ODF)

The ODF contains the following record describing the DODF for provisioning data. Other object directory files are omitted.

```
myODF PKCS15ODF ::= {
    dataObjects  : path : {
        path '4405'H
    }
    trustedCertificates : path : {
        path '4406'H
    }
}
```

## A.3 Example of EF(DODF-prov)

The DODF for provisioning data (file ID 4405) contains the following objects description:

```
myDODF PKCS15DODF ::= {
    opaqueDO : {
        commonObjectAttributes {
            label "Bootstrap",
            flags  {private},
            authId '01'H
        },
        classAttributes {
            applicationOID { joint-isu-itu-t(2) identified-organizations(23)
            wap(43)  provisioning(5) bootstrap(1)}
        },
        typeAttributes indirect : path : {
            path '4431'H,
        }
    },
    opaqueDO : {
        commonObjectAttributes {
            label "Config 1 ",
            flags  {private, modifiable},
            authId '01'H
        },
        classAttributes {
            applicationOID { joint-isu-itu-t(2) identified-organizations(23)
wap(43)  provisioning(5) configuration_1(2)}
        },
        typeAttributes indirect : path : {
            path '4432'H,
        }
    },
opaqueDO : {
        commonObjectAttributes {
            label "Config 2 ",
            flags  {modifiable},
            authId '01'H

        },
        classAttributes {
            applicationOID { joint-isu-itu-t(2) identified-organizations(23)
wap(43)  provisioning(5) configuration_2(3)}
        },
        typeAttributes indirect : path : {
            path '4433'H,
        }
    }

}
```

Informative note 5: file IDs are examples, they are defined by card issuer.

## A.4 Generic DER encoding for the provisioning Files

The table below describes the contents of each logical record.

L is the length of 'label' field. It is required that the length is the same in each record. This way records have fixed length (L + 24hex).

| Bytes | Content (all numbers are hexadecimal) |
|---|---|
| 1 | 30 |
| 1 | L + 1B |
| 1 | 30 |
| 1 | L + 09 |
| 1 | 0C |
| 1 | L |
| L | Label |
| 2 | 03 02 |
| 2 | 07 80 – private |
| 2 | 04 01 |
| 1 | 01 – authId 1 |
| 7 | 30 06 06 04 67 2B 05 |
| 1 | 01 – bootstrap<br>02 – config1<br>03 – config2 |
| 6 | A1 06 30 04 04 02 |
| 2 | file ID |

Note that the ME can determine the label length by reading the 6th byte of the file. Then, it is easy to find offsets for

- label
- type of file (bootstrap, config1, config2)
- file ID

The provisioning documents are contained in files with file IDs 4431, 4432 and 4433.

## A.5 Example of DER encoding for the Bootstrap File.

```
30 24
      30 12
            0C 09 42 6F 6F 74 73 74 72 61 70
                  -- "Bootstrap"
            03 02 07 80 -- private
            04 01 01 –- authId 1

      30 06
            06 04 67 2B 05 01
-- joint-isu-itu-t(2) identified-organizations(23) wap(43)  provisioning(5)
bootstrap(1)}
      A1 06
            30 04
                  04 02 44 31 -- path '4431'
```

The second and third records are encoded in a similar way. Note that the outermost SEQUENCE is omitted.

## A.6 PIN Reference Format

A card PIN format is defined in [ISO7816-4] page 26 table 62 and is presented in the following table:

| b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | Meaning |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | --No information is given |
| 0 | - | - | - | - | - | - | - | --Global reference data (e.g., card password) |
| 1 | - | - | - | - | - | - | - | --Specific reference data (e.g., DF specific password) |
| - | x | x | - | - | - | - | - | 00 (other values are RFU) |
| - | - | - | x | x | x | x | x | --Reference data number |

**Table 1:** Coding of Reference P2

# Appendix B. Static Conformance Requirement

This section is normative. The notation used in this appendix is specified in [CREQ].

## B.1 Provisioning Smart Card Support on ICC

| Item | Function | Reference | Status | Requirement |
|---|---|---|---|---|
| PROVSC-ICC-001 | Provisioning Smart Card implemented on ICC | 5.2, 6 | M | PROVSC-WIM-ICC-001<br>OR<br>PROVSC-GSM-ICC-001 |

### B.1.1 WIM and SIM-WIM Device Implementation

| Item | Function | Reference | Status | Requirement |
|---|---|---|---|---|
| PROVSC-WIM-ICC-001 | Provisioning Smart Card implemented on WIM or SIM-WIM variant of ICC | 5.2, 7 | O | PROVSC-WIM-ICC-101<br>AND<br>PROVSC-WIM-ICC-102<br>AND<br>PROVSC-WIM-ICC-103 |

#### B.1.1.1 General WIM and SIM-WIM Device Options

| Item | Function | Reference | Status | Requirement |
|---|---|---|---|---|
| PROVSC-WIM-ICC-101 | ODF contains pointer to DODF-prov | 6.1, 9.1 | O | |
| PROVSC-WIM-ICC-102 | Storage of PKCS#15 DODF-prov | 6.2, 9.3, 7.1.1 | O | |
| PROVSC-WIM-ICC-103 | Storage of provisioning data | 5.1 | O | PROVSC-WIM-ICC-104<br>OR<br>PROVSC-WIM-ICC-105<br>OR<br>PROVSC-WIM-ICC-106 |
| PROVSC-WIM-ICC-104 | Storage of Bootstrap for read by the ME | 9.4 | O | |
| PROVSC-WIM-ICC-105 | Storage of Config1 for read/update by the ME | 9.5 | O | |
| PROVSC-WIM-ICC-106 | Storage of Config2 for read/update by the ME | 9.6 | O | |

## B.1.2 GSM-SIM Device Implementation

| Item | Function | Reference | Status | Requirement |
|---|---|---|---|---|
| PROVSC-GSM-ICC-001 | Provisioning Smart Card implemented on GSM SIM variant of ICC | 5.2, 8 | O | PROVSC-GSM-ICC-101<br>AND<br>PROVSC-GSM-ICC-102<br>AND<br>PROVSC-GSM-ICC-103<br>AND<br>PROVSC-GSM-ICC-104<br>AND<br>PROVSC-GSM-ICC-105 |

### B.1.2.1 General GSM-SIM Device Options

| Item | Function | Reference | Status | Requirement |
|---|---|---|---|---|
| PROVSC-GSM-ICC-101 | Indirect application selection support | 8.1 | O | |
| PROVSC-GSM-ICC-102 | Storage of EF(DIR) | 8.1 | O | |
| PROVSC-GSM-ICC-103 | Storage of PKCS#15 ODF | 6.1, 9.1 | O | |
| PROVSC-GSM-ICC-104 | Storage of PKCS#15 DODF-prov | 6.2, 9.3 | O | |
| PROVSC-GSM-ICC-105 | Storage of provisioning data | 5.1 | O | PROVSC-GSM-ICC-106<br>OR<br>PROVSC-GSM-ICC-107<br>OR<br>PROVSC-GSM-ICC-108 |
| PROVSC-GSM-ICC-106 | Storage of Bootstrap for read by the ME | 9.4 | O | |
| PROVSC-GSM-ICC-107 | Storage of Config1 for read/update by the ME | 9.5 | O | |
| PROVSC-GSM-ICC-108 | Storage of Config2 for read/update by the ME | 9.6 | O | |
| PROVSC-GSM-ICC-109 | Storage of PKCS#15 CDF | 9.2 | O | |
| PROVSC-GSM-ICC-110 | Storage of Trusted certificates for read by ME | 9.2, 9.7 | O | |

## B.2 Provisioning Smart Card Support on ME

| Item | Function | Reference | Status | Requirement |
|---|---|---|---|---|
| PROVSC-C-001 | Provisioning Smart Card implemented on ME (Client) | 5.2, 6 | M | PROVSC-WIM-C-001<br>OR<br>PROVSC-GSM-C-001 |

## B.2.1 ME Support for WIM and SIM-WIM Implementation

| Item | Function | Reference | Status | Requirement |
|---|---|---|---|---|
| PROVSC-WIM-C-001 | Provisioning Smart Card implemented on WIM or SIM-WIM variant of ICC | 5.2, 7 | O | PROVSC-WIM-C-101 AND PROVSC-WIM-C-102 AND PROVSC-WIM-C-103 AND PROVSC-WIM-C-104 AND PROVSC-WIM-C-105 AND PROVSC-WIM-C-106 AND PROVSC-WIM-C-107 AND PROVSC-WIM-C-108 |

### B.2.1.1 General ME Support for WIM and SIM-WIM Options

| Item | Function | Reference | Status | Requirement |
|---|---|---|---|---|
| PROVSC-WIM-C-101 | Use of pointer to DODF-prov in PKCS#15 ODF | 6.1, 9.1, 10.1 | O | |
| PROVSC-WIM-C-102 | Use of PKCS#15 AODF | 10.1, 7.1.4, 10.1 | O | |
| PROVSC-WIM-C-103 | Use of PKCS#15 DODF-prov | 6.2, 9.3, 7.1.1, 10.1 | O | |
| PROVSC-WIM-C-104 | Read Bootstrap data | 9.4, 10.1 | O | |
| PROVSC-WIM-C-105 | Read/Update Config1 data | 9.5, 10.1 | O | |
| PROVSC-WIM-C-106 | Read/Update Config2 data | 9.6, 10.1 | O | |
| PROVSC-WIM-C-107 | Use of PKCS#15 CDF | 9.2, 10.1 | O | |
| PROVSC-WIM-C-108 | Read Trusted certificates | 9.7, 10.1 | O | |

## B.2.2 ME Support for GSM SIM Implementation

| Item | Function | Reference | Status | Requirement |
|---|---|---|---|---|
| PROVSC-GSM-C-001 | Provisioning Smart Card implemented on GSM SIM variant of ICC | 5.2, 8 | O | PROVSC-GSM-C-101 AND PROVSC-GSM-C-102 AND PROVSC-GSM-C-103 AND PROVSC-GSM-C-104 AND PROVSC-GSM-C-105 AND PROVSC-GSM-C-106 AND PROVSC-GSM-C-107 AND PROVSC-GSM-C-108 AND PROVSC-GSM-C-109 |

### B.2.2.1 General ME Support for GSM SIM Options

| Item | Function | Reference | Status | Requirement |
|---|---|---|---|---|
| PROVSC-GSM-C-101 | Indirect application selection supported | 8.1, 10.2 | O | |
| PROVSC-GSM-C-102 | Use of EF(DIR) | 8.1, 10.2 | O | |
| PROVSC-GSM-C-103 | Use of PKCS#15 ODF | 6.1, 9.1, 10.2 | O | |
| PROVSC-GSM-C-104 | Use of PKCS#15 DODF-prov | 6.2, 9.3, 10.2 | O | |
| PROVSC-GSM-C-105 | Read Bootstrap data | 9.4, 10.2 | O | |
| PROVSC-GSM-C-106 | Read/Update Config1 data | 9.5, 10.2 | O | |
| PROVSC-GSM-C-107 | Read/Update Config2 data | 9.6, 10.2 | O | |
| PROVSC-GSM-C-108 | Use of PKCS#15 CDF | 9.2, 10.2 | O | |
| PROVSC-GSM-C-109 | Read Trusted certificates | 9.7, 10.2 | O | |

# Appendix C. History and Contact Information

| Document history | | |
|---|---|---|
| **Date** | **Status** | **Comment** |
| 12-September-2000 | Draft Version | WAP-186-PROVSC-20000912-d.doc |
| 31-January-2001 | Draft Version | WAP-186-PROVSC-20010131-d.doc |
| 28-February-2001 | Draft Version | WAP-186-PROVSC-20010228-d.doc |
| 15-March-2001 | Draft Version | WAP-186-PROVSC-20010315-d.doc |
| 25-April-2001 | Draft Version | WAP-186-PROVSC-20010425-d.doc |
| 26-April-2001 | Draft Version | WAP-186-PROVSC-20010426-d.doc |
| 18-May-2001 | Proposed Version | Status changed to Proposed (no change in version date) |
| 10-July-2001 | Proposed Version | WAP-186_001-PROVSC and WAP-186_002-PROVSC SCDs included |
| 10-July-2001 | Approved Version | WAP-186-PROVSC-20010710-a |
| **Contact Information**<br>http://www.wapforum.org.<br>technical.comments@wapforum.org | | |