Provisioning Smartcard

Candidate Version 1.1 – 24 Mar 2004

Open Mobile Alliance

OMA-WAP-ProvSC-V1_1-20040324-C

Contents

1. Scope

Open Mobile Alliance (OMA) 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 Open Mobile Alliance continues the work of the WAP Forum to define a set of specifications to be used by service applications. For information on the WAP architecture, please refer to "*Wireless Application Protocol Architecture Specification"* [WAPARCH].

This document defines the information format and the access methods of WAP provisioning data present on smart cards used for wireless telecom applications. A particular case is when a WIM application is already present on such smart card type.

Provisioning WAP connectivity data on a smart card with SIM, USIM or WIM application will have advantages (i.e. pre-provisioning of personalised data during manufacturing, portability, controlled access to sensitive data like login/passwords etc.)

2. References

2.1 Normative References

[CREQ] "Specification of WAP Conformance Requirements", WAP forum ™, WAP-221-CREQ, URL:http://www.openmobilealliance.org/

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

[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 v1.1: Cryptographic Token Information Syntax Standard", RSA Laboratories, June 6, 2000. URL: ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-15/pkcs-15v1_1.pdf

[PROVCONT] "Provisioning content specification 1.1", Open Mobile Alliance ™, OMA-WAP-ProvCont-v1.1, URL: http://www.openmobilealliance.org/

[WIM] "Wireless Identity Module Specification", Open Mobile Alliance™, OMA-WAP-WIM-v1_1, URL: http://www.openmobilealliance.org/

[TS51 011] Specification of the Subscriber Identity Module – Mobile equipment (SIM-ME) interface. (ETSI TS 51 011, R5), URL: http://www.3gpp.org

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

2.2 Informative References

[WAPARCH] "WAP Architecture". WAP forum™, WAP-210-WAPArch, URL:http://www.openmobilealliance.org/

[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] "Provisioning Architecture Overview 1.1", Open Mobile Alliance™, OMA-WAP-PROVARCH-v1_1, URL:http://www.openmobilealliance.org/

3. Terminology and Conventions

3.1 Conventions

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].

All sections and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

3.2 Definitions

Access conditions	A set of security attributes associated with a file.
AID - Application Identifier.	A data element that identifies an application in a smart 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 6.8) 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.

Application Information	Some of the information provisioned into the phone can relate to identity and applications rather than to plain connectivity.
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 smart 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 [TS51 011].
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 6.6) 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 smart card that initiates an action and solicits a response from the smart card.
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	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	Files containing directories of data objects (not keys or certificates) ([PKCS#15], section 6.7) known to the PKCS#15 application.
DODF-wim	The Data Object Directory Files contain directories of data objects (not keys or certificates) ([PKCS#15], section 6.7) used in WTLS and TLS, 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 6.7) 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.
FCP	File Control Parameter
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	([PKCS#15], section 6.2) 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 6.3) contain directories of private keys known to the PKCS#15 application.
PuKDF - The Public Key Directory Files	Files ([PKCS#15], section 6.4) 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 SIM, the UICC or a smart card used in a secondary smart 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 - Trusted Provisioning Server	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 - Wireless 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.
2G UICC	UICC activated in a 2G mode that has physical characteristics of UICC [TS102 221] but logical characteristics of SIM [TS51 011]
3G UICC	UICC activated in a 3G mode that has physical and logical characteristics of the UICC [TS102 221]

3.3 Abbreviations

2G	Second generation network i.e. GSM
3G	Third generation network i.e. UMTS
ADF	Application Dedicated File
AID	Application Identifier
ALW	Always
AODF	Authentication Object Directory File
ASN	Abstract Syntax Notation

ATR	Answer-to-Reset
CDF	Certificate Directory File
CHV	CardHolder Verification
DER	Distinguished Encoding Rules
DF	Dedicated File
DIR	Directory File
DNS	Domain Name Server
DO	Data Object
DODF	Data Object Directory File
EF	Elementary File
ETSI	European Telecommunication Standardization Institute
IC	Integrated Circuit
ICC	Integrated Circuit(s) Card
ID	Identifier
ISO	International Organization for Standardization
ME	Mobile Equipment
MF	Master File
MMS	Multimedia Messaging Service
ODF	Object Directory File
OID	Object Identifier
PIN	Personal Identification Number
PIN-G	General Personal Identification Number according to [WIM]
SC	Smart Card
SIM	Subscriber Identity Module
TPS	Trusted Provisioning Server
UICC	Universal Integrated Circuit(s) Card
WAP	Wireless Application Protocol
WIM	Wireless Identity Module
WTLS	Wireless Transport Layer Security

4. Introduction

The WAP provisioning framework specifies mechanisms to provision devices with connectivity and application information.

The purpose of this document is to specify the implementation of WAP provisioning data on the smart card present in mobile phones - e.g. SIM smart card – allowing pre-configuration of the devices and/or application access parameters.

Very few end users in a mass-market environment will be able (or interested) to perform proper set up of the various connectivity and/or application access information e.g. access point, proxy, local DNS server, MMS proxy/relay. The mobile network operator is now able to pre-configure the subscriber identity smart card with appropriate information so that the end user when inserting the smart card into the device could get direct web browsing or e-mail access.

WAP provisioning framework defines that each bearer network has unique provisioning mechanisms, i.e. network specific procedures to initiate the phone or in the case of deploying the Multimedia Messaging (MMS) application it would be necessary to configure the MMS access specific parameters.
Usage of the smart card to provision the device (compared to an other the air provisioning mechanism) has several advantages for the end user and the mobile network operator point of view; immediate device bootstrapping i.e. no remote connection required for the bootstraping procedure– protected access and storage of user applicative authentification information i.e. login and passwords – portability of provisioning information from one device to another one etc.

Mobile network operator can leverage on its actual subscriber identity module smart card deployement and distribution network to, in addition, provide WAP pre-provisioned smart cards to its customer.

5. Architecture

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 and application information that cannot be changed by the provisioning agent, i.e. by the ME. The card issuer is the only one that can modify this file.
- Config1 File: used to store connectivity and application 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 and application 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 WAP provisioning on telecom smart cards platforms

We can sort out two main types of smart cards used for wireless telecom networks, characterised by their physical and logical characteristics:

SIM smart cards platforms [TS51 011]
UICC smart cards platforms [TS102 221]

This document aims at specifying WAP provisioning data to be hosted directly by such smart card platform type or by a WIM application [WIM] present on it.

5.2.1 Generic Behaviour

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

- Provisioning files on the WIM application present on the SIM or UICC smart card

- Provisioning files on the SIM or UICC smart card

Other non-default provisioning data MAY be read from any location on the smart card. The reading of this information is implementation dependent.

Trusted Certificates can be read in any order i.e. between the ones stored in the ME and the ones stored in the SC Trusted Certificate EF.

6. WAP Provisioning Smart Card

The information format for WAP Provisioning is based on [PKCS#15] specification. The smart 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.

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 `(CommonObjectAttributes.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 `(CommonObjectAttributes.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 `(CommonObjectAttributes.authId)`
- Object identifier indicating a WAP provisioning object and the type of the provisioning object `(CommonDataObjectAttributes.applicationOID)`
- Pointer to the contents of the provisioning document (Path.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 smart 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 any other EF(DODF).

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 ICC

This chapter specifies a special case of the WAP provisioning in the smart card hosting a WIM application.

This chapter deals with provisioning data only, for handling of trusted certificates refer to [WIM].

7.1 WAP provisioning data on WIM application

7.1.1 Introduction

The [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.

The WIM application file structure (PKCS#15) contains at least an Authentication Object Directory File (AODF), a Certificate Directory File (CDF), and a Data Object Directory File (DODF).

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

7.1.2 Access to the WAP Provisioning file structure

To access the WAP Provisioning file structure, the ME MUST select the WIM application as defined in [WIM].

In that case, the WAP provisioning file structure has the same location as the WIM application file structure.

7.1.3 File Overview

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

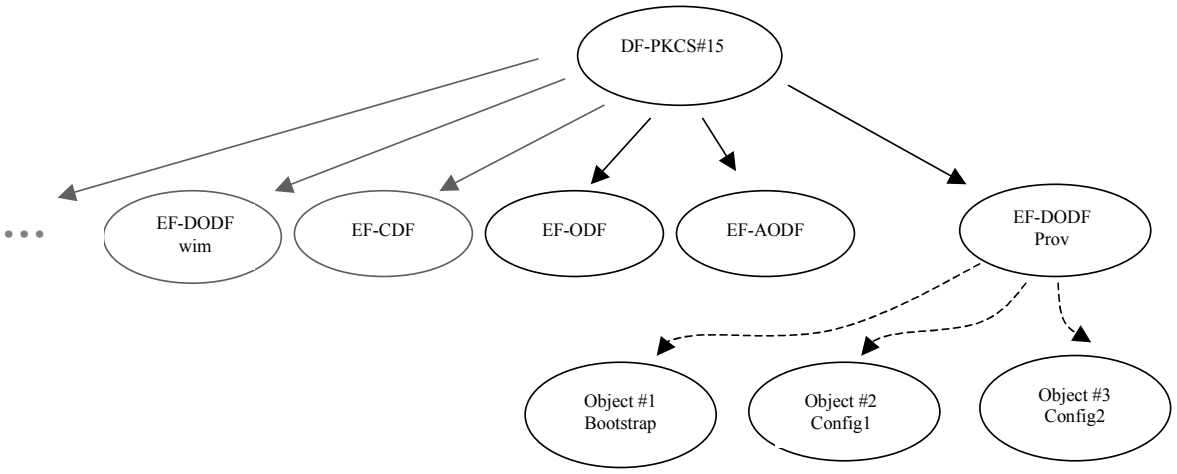

Figure 1: File structure for WAP provisioning data on WIM application

7.1.4 Access method

The WAP provisioning parameters are located under the same PKCS#15 information structure as the WIM application.

The access of WAP provisioning files will be possible using WIM related data storage commands. This requires the WIM application to be active (i.e. selected using direct selection method).

WIM commands used to access provisioning data will be formatted according to the WIM activation mode as defined in [WIM].

7.1.5 Access Conditions

The ME is informed about the access conditions of provisioning files by evaluating the “private” and “modifiable” flags present in the corresponding DODF-prov file structure c.f. 6.2

In the case where one of the above mentioned flag is set, cardholder verification is required. The access rights for provisioning files stored within a WIM smart card application are granted in verifying the PIN-G as defined in the WIM specification [WIM] i.e. the DODF-prov “authId” references to the PIN-G entry in AODF

The ME will retrieve characteristics and location of the PIN-G from the AODF.

Access conditions for files are described in the chapter **Error! Reference source not found.** and the PIN reference format is described in section PIN Reference Format.

8. WAP Provisioning data on WIM-less telecom smart card

8.1 Introduction

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

The support of WAP smart card provisioning data will be indicated to the ME's user agent, by the presence in the EF DIR of a WAP provisioning application template as defined here after.

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

The recommended format of EF(DIR) is a linear fixed record in order to be in line with [TS102 221].

EF (DIR) MUST contain the application template used for a PKCS#15 application as defined in [PKCS15]. Application template MUST consist of Application identifier (tag 0x4F) and Path (tag 0x51) information.

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

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

UICC smart card platforms can support two modes of activation: 2G and 3G.

UICC smart card platform activated in a 2G mode has the logical characteristics of the SIM smart card platform [TS51 011]. In that case, smart card operation for accessing WAP provisioning data conform to the ones defined for the SIM as specified in chapter 8.2

UICC smart card platform activated in a 3G mode has the physical en logical characteristics according to [TS102 221]. In that case, smart card operation for accessing WAP provisioning data are specified in chapter 8.3.

8.2 WAP provisioning data on SIM or UICC activated in 2G mode

8.2.1 Access to the WAP Provisioning file structure

To select the PKCS15 application, the ME MUST evaluate the PKCS#15 application template present in the EF (DIR), then the ME MUST use the indirect selection method as defined in [TS51 011] to select the application.

WAP provisioning files and trusted certificates will be located under the DF(PKCS#15).

8.2.2 Files Overview

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

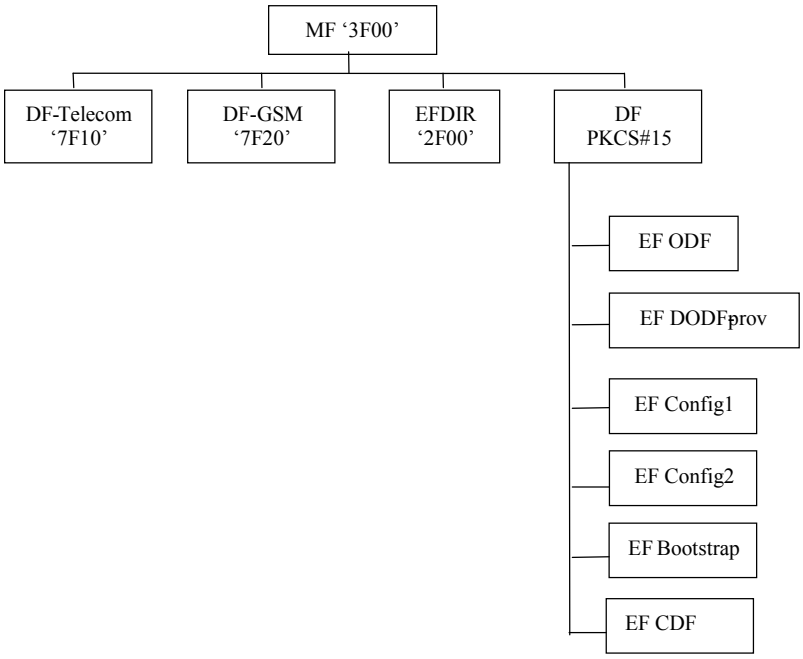

Figure 2: File structure for WAP data on SIM smart card or 2G UICC

8.2.3 Access Method

SIM commands Read Binary and Update Binary, as defined in [TS51 011], are used to access WAP provisioning data.

8.2.4 Access Conditions

The ME is informed of the access conditions of provisioning files by evaluating the "private" and "modifiable" flags in the corresponding CDF and DODF-prov files structure c.f. 6.2

In the case where one of the above mentioned flag is set, cardholder verification is required. The ME implicitly knows that the CHV1 must be verified as defined in [TS51 011].
Remark: in that case the DODF-prov "authId" is not significant since no AODF is present.

Access conditions for files are proposed in the chapter **Error! Reference source not found.**.

8.3 WAP provisioning data on UICC activated in 3G mode

8.3.1 Access to the WAP Provisioning file structure

To select the PKCS#15 application, the ME:

- MUST evaluate the PKCS#15 application template – i.e. PKCS#15 AID - present in the EF (DIR),
- MUST open a logical channel using MANAGE CHANNEL command as specified in [TS102 221],
- MUST select the PKCS#15 ADF using the PKCS#15 AID as parameter of the SELECT command, using direct application selection as defined in [TS102 221].

WAP provisioning files and trusted certificates will be located under the PKCS#15 ADF.

8.3.1.1.1 Files Overview

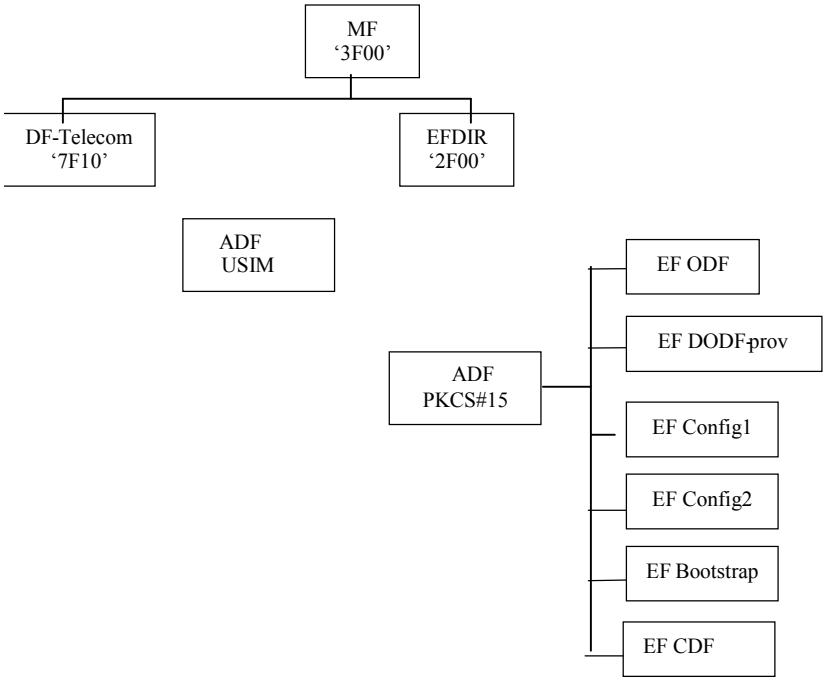

Figure 3: File structure for WAP data on 3G UICC

8.3.2 Access Method

UICC commands Read Binary and Update Binary, as defined in [TS102 221], are used to access WAP provisioning data.

8.3.3 Access Conditions

The ME is informed of the access conditions of provisioning files by evaluating the "private" and "modifiable" flags in the corresponding CDF and DODF-prov files structure c.f. 6.2

In the case where one of the above mentioned flag is set, cardholder verification is required. The ME must evaluate the PIN references that must be verified as defined in [TS102 221] i.e. evaluate the FCP
Remark: in that case the DODF-prov "authId" reference is not significant since no AODF is present.

Access conditions for files are proposed in the chapter **Error! Reference source not found.**.

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 smart card platform or the WIM application. See respective access methods in sections 7.1.4, 8.2.3 and 8.3.2.

In this section, only files used for the provisioning are described. All other files of the WIM application are described in the WIM 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 5.5.1) contains pointers to other EFs, each one containing a directory of PKCS#15 objects of a particular class (e.g. DODF-prov).

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

In the case of WIM ICC, 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 SIM or UICC, 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: READ ALW UPDATE ADM INVALIDATE ADM REHABILITATE ADM		
Description		
See sections A.2		

9.2 EF CDF

An optional Certificate Directory File (CDF) ([PKCS#15], section 6.6) 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 SIM or UICC, the EF(CDF) is described below:

Identifier: see ODF	Structure: Binary	Optional
File size: decided by the card issuer	Update activity: low	

Access Conditions:	
READ	ALW
UPDATE	ADM or NEV
INVALIDATE	ADM or NEV
REHABILITATE	ADM or NEV
Description	
See [WIM], Example of EF (ODF)	

9.3 EF DODF-prov

This Data Object Directory File provisioning contains directories of provisioning data objects ([PKCS#15], section 6.7) 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 smart card. Thus, the card issuer decides the file size.

Identifier: See ODF	Structure: Binary	Mandatory
File size: decided by the card issuer	Update activity: low	
Access Conditions:		
READ	ALW	
	or PIN-G (WIM, See section 7.1.5)	
	or CHV1 (SIM, See section 8.2.4)	
	or Universal / application / Local PIN (UICC, 8.3.3)	
UPDATE	ADM	
INVALIDATE	ADM	
REHABILITATE	ADM	
Description		
See sections 6.2, Example of EF (DODF-prov)		

9.4 EF Bootstrap

Only the card issuer can modify EF Bootstrap

Setting all bytes to ‘FF’ initialises EF Bootstrap.

Identifier: See DODF	Structure: Binary	Optional

Recommended minimum file size: 150 bytes	Update activity: low
Access Conditions: READ ALW or PIN-G (WIM, See section 7.1.5) or CHV1 (SIM, See section 8.2.4) or Universal / application / Local PIN (UICC, 8.3.3) UPDATE ADM INVALIDATE ADM REHABILITATE ADM	
Description	
See [PROVCONT]	

9.5 EF Config1

The user can modify EFConfig1

Setting all bytes to 'FF' initialises EFConfig1.

Identifier: See DODF	Structure: Binary	Optional
Recommended minimum file size: 150 bytes	Update activity: low	
Access Conditions: READ ALW or PIN-G (WIM, See section 7.1.5) or CHV1 (SIM, See section 8.2.4) or Universal / application / Local PIN (UICC, 8.3.3) UPDATE PIN-G (WIM, See section 7.1.5) or CHV1 (SIM, See section 8.2.4) or Universal / application / Local PIN (UICC, 8.3.3) INVALIDATE ADM REHABILITATE ADM		
Description		

See [PROVCONT]

9.6 EF Config2

The user can modify EFConfig2.

Setting all bytes to 'FF' can initialise EFConfig2.

Identifier: See DODF	Structure: Binary	Optional
Recommended minimum file size: 150 bytes	Update activity: low	
Access Conditions: READ ALW or PIN-G (WIM, See section 7.1.5) or CHV1 (SIM, See section 8.2.4) or Universal / application / Local PIN (UICC, 8.3.3) UPDATE ALW or PIN-G (WIM, See section 7.1.5) or CHV1 (SIM, See section 8.2.4) or Universal / application / Local PIN (UICC, 8.3.3) INVALIDATE ADM 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:	
READ	ALW
UPDATE	ADM or NEV
INVALIDATE	ADM or NEV
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 WIM smart card, the second one addresses the WAP provisioning and trusted certificates on the SIM and the third one WAP provisioning and trusted certificates on UICC smart card.

The ME MUST support the WAP provisioning data on WIM if the ME is a mobile phone supporting the WIM.

The ME MUST support the WAP provisioning data on WIM-less telecom smart card if the ME is mobile phone.

Informative note 4:

The ME can determine whether the smart 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 **Error! Reference source not found.**) and implementation details are provided in **Error! Reference source not found.**.

10.1 Requirements on the WIM on smart card

To support the WAP provisioning on the WIM on smart card, 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 SIM or 2G UICC

To support the WAP provisioning and reading of trusted certificates on the SIM or 2G UICC, the ME MUST perform the following steps:

- Read EF (DIR) to evaluate the WAP provisioning application template and find the file identifier (and path of the PKCS#15 DF),
- Select PKCS#15 DF (indirect selection), as defined in [TS51 011],
- 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.

10.3 Requirements on the 3G UICC

To support the WAP provisioning on the 3G UICC, the ME MUST perform the following steps:

- Select WAP provisioning file structure as specified in 8.3.1
- Read ODF to locate the DODF-prov,
- 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.

Change History (Informative)

A.1 Approved Version History

Reference	Date	Description

A.2 Draft/Candidate Version 1.2 History

Document Identifier	Date	Sections	Description
	2002-09-02		The initial version of this document.
OMA-WAP-ProvSC-V1_1-20040324-C	2002-11-12		Candidate.
OMA-WAP-ProvSC-V1_1-20040324-C	2004-03-24	All 9.6	Template update. Candidate. Bug fix to EF Config2 (CR DM -2004-0059)

Appendix B. Static Conformance Requirements (Normative)

The notation used in this appendix is specified in [IOPPROC].

B.1 Provisioning Smart Card Support on ICC

Item	Function	Reference	Status	Requirement
PROVSC-ICC-001	Provisioning Smart Card implemented on WIM-ICC, SIM or UICC	5.2, **Error! Reference source not found.**	**M**	PROVSC-WIM-ICC-001 OR PROVSC-SIM-001 OR PROVSC-UICC-001

B.1.1 WIM Device Implementation

Item	Function	Reference	Status	Requirement
PROVSC-WIM-ICC-001	Provisioning Smart Card implemented on WIM-ICC	5.2, **Error! Reference source not found.**	**O**	PROVSC-WIM-ICC-101 AND PROVSC-WIM-ICC-102 AND PROVSC-WIM-ICC-103

B.1.1.1 General 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 OR PROVSC-WIM-ICC-105 OR 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 SIM Device Implementation

Item	Function	Reference	Status	Requirement

PROVSC-SIM-001	Provisioning Smart Card implemented on SIM	5.2, 8.2	O	PROVSC-SIM-101 AND PROVSC-SIM-102 AND PROVSC-SIM-103 AND PROVSC-SIM-104 AND PROVSC-SIM-105

B.1.2.1 General SIM Device Options

Item	Function	Reference	Status	Requirement
PROVSC-SIM-101	Indirect application selection support	8.2	O	
PROVSC-SIM-102	Storage of EF(DIR)	8.1	O	
PROVSC-SIM-103	Storage of PKCS#15 ODF	**Error! Reference source not found.**, 9.1	O	
PROVSC-SIM-104	Storage of PKCS#15 DODF-prov	6.2, 9.3	O	
PROVSC-SIM-105	Storage of provisioning data	5.1	O	PROVSC-SIM-106 OR PROVSC-SIM-107 OR PROVSC-SIM-108
PROVSC-SIM-106	Storage of Bootstrap for read by the ME	9.4	O	
PROVSC-SIM-107	Storage of Config1 for read/update by the ME	9.5	O	
PROVSC-SIM-108	Storage of Config2 for read/update by the ME	9.6	O	
PROVSC-SIM-109	Storage of PKCS#15 CDF	9.2	O	
PROVSC-SIM-110	Storage of Trusted certificates for read by ME	9.2, 9.7	O	

B.1.3 UICC Device Implementation

Item	Function	Reference	Status	Requirement
PROVSC-UICC-001	Provisioning Smart Card implemented on UICC	5.2, 8.3	O	PROVSC-UICC-101 AND PROVSC-UICC-102 AND PROVSC-UICC-103 AND PROVSC-UICC-104 AND PROVSC-UICC-105 AND PROVSC-UICC-106

B.1.3.1 General UICC Device Options

Item	Function	Reference	Status	Requirement
PROVSC-UICC-101	Direct application selection support	8.3.1	O	

PROVSC-UICC-102	Logical channel	8.3.1	O	
PROVSC-UICC-103	Storage of EF(DIR)	8.1	O	
PROVSC-UICC-104	Storage of PKCS#15 ODF	6.1, 9.1	O	
PROVSC-UICC-105	Storage of PKCS#15 DODF-prov	6.2, 9.3	O	
PROVSC-UICC-106	Storage of provisioning data	5.1	O	PROVSC-UICC-107 OR PROVSC-UICC-108 OR PROVSC-UICC-109
PROVSC-UICC-107	Storage of Bootstrap for read by the ME	9.4	O	
PROVSC-UICC-108	Storage of Config1 for read/update by the ME	9.5	O	
PROVSC-UICC-109	Storage of Config2 for read/update by the ME	9.6	O	
PROVSC-UICC-110	Storage of PKCS#15 CDF	9.2	O	
PROVSC-UICC-111	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, **Error! Reference source not found.**	**M**	PROVSC-WIM-C-001 OR PROVSC-SIM-C-001 OR PROVSC-UICC-C-001

B.2.1 ME Support for WIM Implementation

Item	Function	Reference	Status	Requirement
PROVSC-WIM-C-001	Provisioning Smart Card implemented on WIM-ICC	5.2.1, **Error! Reference source not found.**	**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 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	7.1.5, 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 SIM Implementation

Item	Function	Reference	Status	Requirement
PROVSC-SIM-C-001	Provisioning Smart Card implemented on SIM	5.2, 8.2	**O**	PROVSC-SIM-C-101 AND PROVSC-SIM-C-102 AND PROVSC-SIM-C-103 AND PROVSC-SIM-C-104 AND PROVSC-SIM-C-105 AND PROVSC-SIM-C-106 AND PROVSC-SIM-C-107 AND PROVSC-SIM-C-108 AND PROVSC-SIM-C-109

B.2.2.1 General ME Support for SIM Options

Item	Function	Reference	Status	Requirement
PROVSC-SIM-C-101	Indirect application selection supported	8.2.1, 10.2	O	
PROVSC-SIM-C-102	Use of EF (DIR)	8.2.1, 10.2	O	
PROVSC-SIM-C-103	Use of PKCS#15 ODF	6.1, 9.1, 10.2	O	
PROVSC-SIM-C-104	Use of PKCS#15 DODF-prov	6.2, 9.3, 10.2	O	
PROVSC-SIM-C-105	Read Bootstrap data	9.4, 10.2	O	
PROVSC-SIM-C-106	Read/Update Config1 data	9.5, 10.2	O	
PROVSC-SIM-C-107	Read/Update Config2 data	9.6, 10.2	O	
PROVSC-SIM-C-108	Use of PKCS#15 CDF	9.2, 10.2	O	
PROVSC-SIM-C-109	Read Trusted certificates	9.7, 10.2	O	

B.2.3 ME Support for UICC Implementation

Item	Function	Reference	Status	Requirement

PROVSC-UICC-C-001	Provisioning Smart Card implemented on UICC	5.2, 8.2	**O**	PROVSC-UICC-C-101 AND PROVSC- UICC -C-102 AND PROVSC- UICC -C-103 AND PROVSC- UICC -C-104 AND PROVSC- UICC -C-105 AND PROVSC- UICC -C-106 AND PROVSC- UICC -C-107 AND PROVSC- UICC -C-108 AND PROVSC- UICC -C-109 AND PROVSC-UICC-C-110

B.2.3.1 General ME Support for UICC Options

Item	Function	Reference	Status	Requirement
PROVSC-UICC-C-101	Direct application selection supported	8.3.1	O	
PROVSC-UICC-C-102	Use of EF (DIR), for the case of application selection by use of the EF DIR file	8.3.1	O	
PROVSC-UICC-C-103	Logical channel	8.3.1	O	
PROVSC-UICC -C-104	Use of PKCS#15 ODF	6.1, 9.1, 10.3	O	
PROVSC-UICC -C-105	Use of PKCS#15 DODF-prov	6.2, 9.3, 10.3	O	
PROVSC-UICC -C-106	Read Bootstrap data	9.4, 10.3	O	
PROVSC-UICC -C-107	Read/Update Config1 data	9.5, 10.3	O	
PROVSC-UICC -C-108	Read/Update Config2 data	9.6, 10.3	O	
PROVSC-UICC -C-109	Use of PKCS#15 CDF	9.2, 10.3	O	
PROVSC-UICC -C-110	Read Trusted certificates	9.7, 10.3	O	

Appendix C. Informative Notes

C.1 Example of EF (DIR)

Exemple of coding of a WAP provisioning application template.

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].

C.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
    }
}
```

C.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, card issuer defines them.

C.4 Generic DER encoding for the provisioning Files (DODF-prov)

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 02 – config1 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.

C.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.

C.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