# WAP Provisioning Bootstrap

WAP-184-PROVBOOT
Proposed Version 23-February-2000

**Wireless Application Protocol**
**WAP Provisioning Bootstrap Specification**

***Disclaimer:***

*This document is subject to change 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].

Provisioning is the process by which a WAP client is configured with a minimum of user interaction. The term covers both OTA provisioning and provisioning by means of, e.g., SIM cards. This specification defines a part of the provisioning process, namely, the bootstrap process, which is an OTA mechanism used to provision unconfigured WAP clients when, e.g., a SIM card containing WAP provisioning information is not available.

# 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

[WAPARCH] “WAP Architecture Specification”, WAP Forum, 30-April-1998, URL: http://www.wapforum.org/

[WAPPUSH] “WAP Push OTA Specification”, WAP Forum, 15-August-1999, URL: http://www.wapforum.org/

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

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

[PROVUAB] “WAP Provisioning User Agent Behavior Specification”, WAP Forum, WAP-185-PROVUAB, URL: http://www.wapforum.org/

[WAPWDP] “Wireless Datagram Protocol Specification”, WAP Forum, 05-November-1999, URL: http://www.wapforum.org/

[GSM0338] “Alphabets and Language Specific Information”, ETSI, URL: http://www.etsi.org/

## 3.2 Informative References

[WBXML] “WAP Binary XML Content Format”, WAP Forum, 15-August-1999, 04-November-1999, URL: http://www.wapforum.org/

[WTLS] “Wireless Transport Layer Security”, WAP Forum, 05-November-1999, URL: http://www.wapforum.org/

[E2ESEC] “Transport Layer End to End Security Specification”, WAP Forum, WAP-187, URL: http://www.wapforum.org/

# 4. Definitions and Abbreviations

## 4.1 Terminology

This specification uses the following words for defining the significance of each particular requirement:

MUST

This word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification.

MUST NOT

This phrase, or the phrase "SHALL NOT", mean that the definition is an absolute prohibition of the specification.

SHOULD

This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.

SHOULD NOT

This phrase, or the phrase "NOT RECOMMENDED" mean that there may exist valid reasons in particular circumstances when the particular behaviour is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behaviour described with this label.

MAY

This word, or the adjective "OPTIONAL", mean that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation which does not include a particular option MUST be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does include a particular option MUST be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides.)

## 4.2 Definitions

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

Bootstrap Document

A provisioning document with information of relevance to the bootstrap process only.

Bootstrap process (bootstrapping)

The process by which the unconfigured ME is taken from the initial state to or through the TPS Access State. This process can be system specific.

Configuration Context

A Configuration Context is a set of connectivity and application configurations associated with a single TPS. A TPS can be associated with several Configuration Contexts. However, 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

This connectivity information 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.

Continuous provisioning

The process by which the ME is provisioned with further infrastructure information at or after the TPS Access state. The information received during the bootstrap MAY be modified. This process is generic and optional. Continuous implies that the process can be repeated multiple times, but not that it is an ongoing activity.

Network Access Point

A physical access point is an interface point between the wireless network and the fixed network. It is often a RAS (Remote Access Server), an SMSC, a USSDC, or something similar. It has an address (often a telephone number) and an access bearer.

Privileged Context

A privileged context is a special context in which it is possible to define the number of additional contexts allowed. Not all WAP service providers are, however, allowed to bootstrap the privileged context.

Provisioned state

The state in which the ME has obtained connectivity information extending its access capabilities for content, applications or continuous provisioning. This state is reached when the bootstrap process has provided access to generic proxies, or the continuous provisioning process has been performed.

Provisioning Document

A particular instance of a XML document encoded according to the provisioning content type specification [PROVCONT].

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.

TPS Access State

The state in which the ME has obtained a minimum set of infrastructure components that enables the ME to establish the first communication channel(s) to WAP infrastructure, i.e. a trusted WAP proxy. This allows continuous provisioning  but may also provide sufficient information to the ME to access any other WAP content or application.

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 enduser from harmful proxy navigation information.

## 4.3 Abbreviations

For the purposes of this specification the following abbreviations apply.

| | |
|---|---|
| **MAC** | Message Authentication Code |
| **MIME** | Multipurpose Internet Mail Extensions |
| **ME** | Mobile Equipment |
| **NAP** | Network Access Point |
| **OTA** | Over The Air |
| **TPS** | Trusted Provisioning Server |
| **WAP** | Wireless Application Protocol |

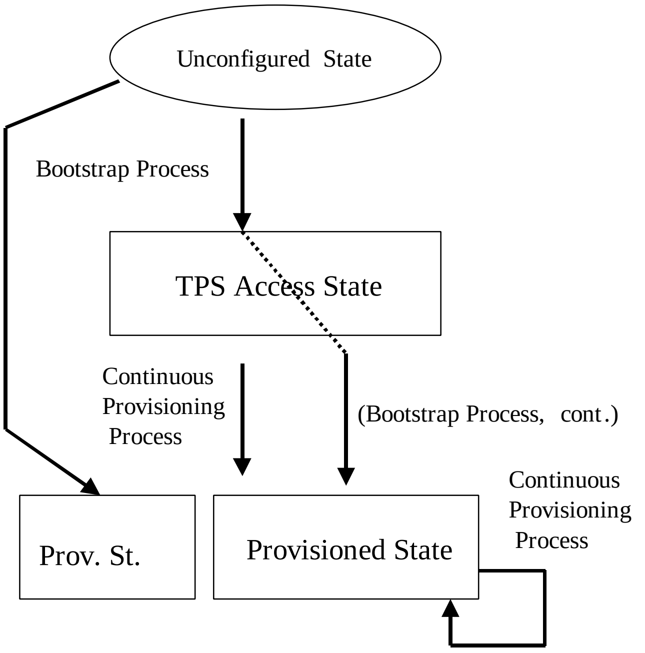
# 5. Introduction

The bootstrap process is performed when an unconfigured configuration context within the ME must be provisioned with WAP connectivity information by means of an OTA (Over-The-Air) mechanism. Since an unconfigured configuration context does not have sufficient information to establish a WAP connection to the infrastructure the bootstrap process will, to some extent, rely on the mechanisms available in the underlying network technology.

The bootstrap process establishes a basic relationship between the device and the network, i.e. an initial set of configuration information. This information, at a minimum a network access point and a proxy, and usually a content location (the Trusted Provisioning Server), specifies an access method to WAP resources. The bootstrap process is in particular able to specify, using a Trusted Proxy, an access path to a Trusted Provisioning Server (TPS).

It is possible to define access to generic WAP proxies in the bootstrap process. It is even possible to omit the definition of a TPS if there is no intention to perform continuous provisioning over the air.

The intended result of this bootstrap process is that the device has a trusted point of configuration, i.e. a TPS. This initial configuration can be leveraged by the continuous provisioning mechanism. Thanks to the separation of the bootstrap and the continuous provisioning the former is allowed to be network and bearer specific while the latter is generic.


Unconfigured State
Bootstrap Process
TPS Access State
Continuous Provisioning Process
(Bootstrap Process, cont.)
Continuous Provisioning Process
Prov. St.
Provisioned State

# 6. The Bootstrap Process

## 6.1 OTA Mechanism

The bootstrap process is initiated when the dedicated server sends a bootstrap document via WAP unconfirmed push [WAPPUSH] with the default push port, the default application ID and the provisioning MIME-type (`application/vnd.wap.connectivity-wbxml`) to the ME. This ensures that the bootstrap mechanism is able to work on most bearers that support network initiated communication using WDP.

## 6.2 Security Mechanisms

Since the bootstrap process enables an infrastructure entity to configure a configuration context of the ME, it is important that the following security requirements are met:

1. The server that is initiating the bootstrap process MUST be authenticated.
2. The client for which the bootstrap process is targeted MUST be authenticated.

It is assumed that the underlying network technology provides a means for identifying and (maybe only implicitly) authenticating the client. For GSM, for example, the routing of SMS based on the MSISDN of the client provides an implicit authentication of the client.

For the server authentication, two methods are considered as described below. In some situations these methods may not apply, due to the implicit security of the underlying network mechanism.

The definition of a Privileged Configuration Context allows to assign a higher level of security to that context than to the optional other contexts.

### 6.2.1 The Generic Security Mechanism

The generic security mechanisms are intended to be available for all bearers. In some cases the generic mechanisms might not apply due to the implicit security of the bearer network. The mechanisms are either based on a secret that is shared between the ME and the correct sender of the bootstrap document or an out-of-band mechanism for delivering some authentication information. What constitutes the shared secret depends on the underlying network technology.

*Bootstrap security by means of a shared secret*

In order to provide security by means of a shared secret, the generator of the bootstrap document MUST attach a mac and a secmethod attribute in the wap-provisioningdoc tag. If the secmethod attribute takes the value USERPIN, the shared secret is based on a user pin. If the secmethod attribute takes the value NETWPIN, the shared secret is based on a network specific shared secret. Finally, if the secmethod is USERNETWPIN, the shared secret is a network specific shared secret appended with a user PIN.

When presented to the user, the format of a user PIN MUST be a string of decimal digits. Internally, however, for computational purposes, the PIN MUST be converted to a binary number consisting of the number of octets required to hold the biggest decimal value with the same number of digits as the PIN. Likewise, any network PIN must be converted to a binary format.

The mac is calculated in the following way:

First, the bootstrap document is constructed with a MAC attribute consisting of an empty string and encoded in the WBXML format [WBXML]. The so encoded document and the shared secret in binary format are then input as the data and key, respectively, for the HMAC calculation based on the SHA-1 algorithm as defined in the WTLS specification [WTLS]. The output of the HMAC (M = HMAC-SHA(K, A)) calculation is encoded as a string of hexadecimal digits where each pair of consecutive digits represent a byte. The bootstrap document is then reconstructed with a MAC consisting of the hexadecimal encoded output from the HMAC calculation.

This calculation is repeated in the ME when checking the validity of the MAC.

*Bootstrap security by means of an out-of-band delivery of authentication information*

In order to provide security by means of an out-of-band delivery of authentication information, the generator of the bootstrap document MUST omit the mac attribute of the wap-provisioningdoc tag but include the secmethod attribute with the value USERPINMAC. Instead, the user has to receive a PIN from the generator of the boostrap document by some out-of-bands mechanism. The PIN is then used to check the validity of the boostrap document. The PIN must consist of a 2*L decimal digits (where L is a number bigger than 4), and must be generated as follows:

1) Let A be the WBXML encoded bootstrap document.
2) Generate a random string K of decimal digits (i.e. octets with hexadecimal values 30 to 39) with length L.
3) Calculate the array of octets M = HMAC-SHA(K, A)
4) Generate a string m of length L from M according to m(i) = M(i) mod 10 + 48, where i refers to the individual elements of the string m and array M, respectively.
5) Generate the PIN code C as a concatenation C = K || m

When the ME receives a bootstrap document with a secmethod attribute in the root tag set to USERPINMAC, the process is repeated:

1) Let A be the WBXML encoded bootstrap document.
2) Retrieve the string K from the first half of the PIN code C = K || m, which has the length 2*L
3) Calculate the array of octets M' as above
4) Generate a string m' as above
5) If m' and m are identical, the bootstrap document can be accepted.

### 6.2.2 Additional Bearer Specific Security Mechanisms

Some bearers may require support for special security mechanisms in addition to the generic security mechanism. This could, for example, be the case if the shared secret available for the generic security mechanism is not considered sufficiently safe. Specification of such additional mechanisms is relegated to the next chapter.

# 7. Network Specific Adaptions

## 7.1 Adaption to GSM

In GSM, the IMSI must be used as the shared bootstrap secret.

### 7.1.1 SIM

Bootstrap data stored on the SIM card has the highest priority. If data is found on the SIM card no over the air bootstrap procedure is valid for that Configuration Context. Update of bootstrap data can be done only using out of band methods, i.e. SIM card specific methods (for example a non-WAP over the air configuration method).

The SIM may store data of a Privileged Configuration Context.

### 7.1.2 Cell Broadcast

The mechanism for Cell Broadcast bootstrap is network initiated. No user pin or shared secret needs to be used as the same bootstrap message is delivered to all mobiles within a particular area.

The security mechanism consists of the assigned broadcast channel 421. The channel shall be reserved to be used for transport of WAP bootstrap parameters, and can thus not be used by anyone else.

A bootstrap message is addressed to a particular group of mobiles, i.e. the mobiles of a carrier, using the network code. The SIM card, an thus device has a network code as part of the IMSI parameter. This network code is compared to the network code provided by the network and a parameter in the provisioning content type.

The mechanism for approving a bootstrap message based on a Network Code match is defined as follows. There are three network codes:

- A provisioning Network Code as available from the bootstrap document
- A SIM Network Code (the Network Code = Mobile Country Code & Mobile Network Code) available from the IMSI on the SIM
- A System Network Code (Mobile Country Code and Mobile Network Code as specified in the system information messages transmitted on the broadcast control channel)

The System Network Code, Provisioning Network Code and SIM Network Code MUST be equal in order for the message to be accepted as valid bootstrap information.

Cell Broadcast may transmit data for a Privileged Configuration Context.

*Guidelines for Network Management*

For the use of CB, the following configurations of CB in the network have to be provided

- The CB parameter Geographical Scope MUST be coded to "PLMN wide validity" implying automatically the coding of the CB parameter Display mode as “Normal Display”.
- The CB Parameter Update Number has a value of 0 when the bootstrap parameters are broadcast the first time. If the bootstrap parameters will change in future, the bootstrap message is appropriately adapted by the operator, i.e. a normal change procedure is invoked at the CBC (Cell Broadcast Centre), leading to an update of the content of the bootstrap message and to an automatic increment of the Update Number. A new Update Number can be taken by the MS as an indication of a change of the bootstrap parameters and trigger a verification process of the parameter set. The Update Number is incremented cyclically between 0 and 15.
- The CB parameter Message Identifier (MI) indicates the logical CB channel on which a CB message is broadcast. There will be one single WAP-CB-Channel carrying the bootstrap information: MI = 421.
- CB scheduling messages MUST be used as follows: If the duration of the schedule period is assumed to be one minute, i.e. 32 CB messages can be broadcast within one schedule period, there are 32 CB message slots. The first CB message slot carries the *scheduled* schedule message (CB message slot 0). CB messages that are to be broadcast (e.g. the bootstrap message) are spread over the CB message slots according to their repetition rate. If there is a free CB message slot without any CB message to be broadcast, the network SHALL send *unscheduled* schedule messages in this slots, i.e. schedule messages that are not broadcast in CB message slot 0, but in any other CB message slot.

### 7.1.3 SMS

The bootstrap mechanism when using SMS is network initiated. Bootstrapping over SMS MUST use one of the generic security mechanisms. To this end, the network specific shared secret is binary format of the IMSI.

This mechanism MUST NOT be used to transfer bootstrap data to a Privileged Configuration Context, unless network shared secret is used.

### 7.1.4 USSD

The bootstrap adaptation using GSM USSD is network initiated. Bootstrapping over USSD MUST use one of the generic security mechanisms. To this end, the network specific shared secret is the binary format of the IMSI.

All WAP messages, including the pushed provisioning message, are distinguished by a reserved DCS (Data Coding Scheme, GSM 03.38).

USSD must not be used to transfer bootstrap data to a Privileged Configuration Context, unless the network shared secret is used.

### 7.1.5 User Agent Behaviour

All configuration data, including the bootstrap data, is tied to a specific identity of the SIM, i.e. the IMSI. If a new (different) SIM is inserted the device should keep the original configuration private (not visible). The phone might store multiple configurations, each tied to a particular IMSI.

When receiving a bootstrap document, the ME must validate the document (where applicable) using the prescribed methods. Only bootstrap documents that are properly authenticated SHALL be accepted.

The bootstrap over point to point bearers such as GSM SMS and GSM USSD is a one-time event per configuration context. Within a configuration context, the bootstrap process cannot be re-performed unless the context is reset (using an out of band method). The bootstrap data set established over broadcast bearers (e.g. Cell Broadcast) can be updated using the same bearer.

The ME MUST set a configuration context to the unconfigured state if the user requests a (possibly WAP-specific) reset of the ME, thereby allowing the bootstrap process to be reinitiated for that context. In all cases an update of bootstrap information causes reset of the configuration context.

## 7.2 Adaption to TDMA

TBD

## 7.3 Adaption to CDMA

### 7.3.1 SMS

The bootstrap protocol MAY be delivered to the ME using various transport mechanisms.  These mechanisms SHALL include the use of mobile-terminated SMS as per TIA/EIA-637-A on the paging channel. The bootstrap message includes a MAC, which is calculated based on shared secret data (SSD)and is concatenated with the provisioning content element as described in the Generic Security Mechanism section.

For the mobile station to authenticate to the TPS the following methods SHALL be used. The SSD SHALL be a combination of known ESN and SPC values. More specifically, the SSD is the 32-bit ESN appended with the 24-bit SPC (service programming code), or SSD entered by the user. The TPS SHALL employ a hash algorithm to transform the ESN and SPC values into an HMAC calculation as per section 6.2.1. which will be included in the bootstrap data and validated by the client.

Only the methods NETWPIN and USERNETWPIN are allowed to be used to bootstrap the privileged configuration context.

This mechanism may transmit data for a Privileged Configuration Context if network shared secret is used.

### 7.3.2 User Agent Behaviour

All configuration contexts, including the bootstrap data, are tied to a specific NAM (Number Assignment Module) of the ME. The phone MAY store multiple configuration context, privileged or otherwise, per NAM. Each configuration context is specific to a certain NAM.

The ME SHALL check the validity of the bootstrap document using the authentication procedures specified in section 6.2.1. Only NETWPIN or USERNETWPIN are allowed to bootstrap the privileged context.. If *secmethod* consists of USERNETWPIN, the user PIN is also validated. In the event of an authentication failure, the mobile SHALL not update its memory with TPS configuration data.

The ME MUST set a configuration context to its unconfigured state if the user requests a (possibly WAP-specific) reset of the ME, thereby allowing the bootstrap process to be reinitiated for that context.

# A Appendix. Static Conformance Requirement

This static conformance requirement lists a minimum set of functions that can be implemented to help ensure that WAP provisioning implementations and ME implementations will be able to inter-operate. This section describes only the set of functions needed to enable WAP data bootstrap provisioning.

The “Status” column indicates whether the function is mandatory (M) or optional (O).

## A.1 General Bootstrap Feature

The Bootstrap functionality must support the functions described below in addition to the functions defined in [PROVCONT]. Only the functions specific to WAP bootstrap of mobile devices are described below.

| Item | Functionality | Reference | Status |
|---|---|---|---|
| CP-B-001 | Support for the WAP-PROVISIONINGDOC | 6.1 | M |
| CP-B-002 | Support for WAP-PROVISIONINGDOC read from WIM/SIM | 6 | O |
| CP-B-003 | Support for WAP-PROVISIONINGDOC received Over The Air | 6 | O |
| CP-B-004 | Support for WAP-PROVISIONINGDOC read from the device (pre-configured bootstrap) | 6 | O |
| CP-B-005 | Support for WAP-PROVISIONINGDOC generic security mechanism | 6.2.1 | M |

### A.1.1 Bearer Support

| Item | Functionality | Reference | Status |
|---|---|---|---|
| CP-BCT-001 | Support for the GSM | 7.1 | O |
| CP-BCT-002 | Support for US-CDMA | 7.3 | O |
| CP-BCT-003 | Support for US-TDMA | 7.2 | O |
| CP-BCT-004 | Support for Generic Mechanism Over The Air Mechanism | 6.2.1 | O |

## A.2 GSM Features

| Item | Functionality | Reference | Status |
|---|---|---|---|
| CP-BGSM-001 | Support for WAP-PROVISIONINGDOC read from WIM | 7.1.1 | O |
| CP-BGSM-002 | Support for WAP-PROVISIONINGDOC read from SIM | 7.1.1 | O |
| CP-BGSM-003 | Support for WAP-PROVISIONINGDOC received by Cell Broadcast | 7.1.2 | O |
| CP-BGSM-004 | Support for WAP-PROVISIONINGDOC received over SMS bearer | 7.1.3 | O |
| CP-BGSM-005 | Support for WAP-PROVISIONINGDOC received over USSD bearer | 7.1.4 | O |

## A.3 US-CDMA Features

| Item | Functionality | Reference | Status |
|---|---|---|---|
| CP-BCDMA-001 | Support for WAP-PROVISIONINGDOC received over SMS bearer | 7.3 | O |

## A.4 US-TDMA Features

| Item | Functionality | Reference | Status |
|---|---|---|---|
| CP-BTDMA-001 | Support for WAP-PROVISIONINGDOC received over SMS bearer | 7.2 | O |

## A.5 Generic Features

| Item | Functionality | Reference | Status |
|---|---|---|---|
| CP-BGF-001 | Support for multiple configuration context | 4.2 | O |
| CP-BGF-002 | Support for privileged configuration context | 4.2 | M |

## A.6 Generic Security Features

This section is valid only if the over the air mechanism has been implemented.

| Item | Functionality | Reference | Status |
|---|---|---|---|
| CP-BSF-001 | Support for NETWPIN | 6.2.1 | O |
| CP-BSF-002 | Support for USERPIN | 6.2.1 | O |
| CP-BSF-003 | Support for USERNETWPIN | 6.2.1 | M |
| CP-BSF-004 | Support for USERPINMAC | 6.2.1 | M |

# B Appendix. History and Contact Information

| Document history | | |
|---|---|---|
| **Date** | **Status** | **Comment** |
| 23-Feb-2000 | | WAP-184-PROVBOOT |

**Contact Information**

http://www.wapforum.org.

technical.comments@wapforum.org