

WAP WINA Process Document

WAP-212-WINAProcess

Version 04-Feb-2002

Wireless Application Protocol WAP Interim Naming Authority Process Document

Disclaimer:

The document is subject to change without notice.

Contents

| | |
|--|-----------|
| 1. SCOPE..... | 3 |
| DOCUMENTS STATUS..... | 4 |
| 1.1 COPYRIGHT NOTICE | 4 |
| 1.2 ERRATA..... | 4 |
| 1.3 COMMENTS | 4 |
| 2. REFERENCES | 5 |
| 2.1 NORMATIVE REFERENCES..... | 5 |
| 2.2 INFORMATIVE REFERENCES | 5 |
| 3. DEFINITIONS AND ABBREVIATIONS..... | 6 |
| 3.1 DEFINITIONS..... | 6 |
| 3.2 ABBREVIATIONS..... | 6 |
| 4. INTRODUCTION | 7 |
| 4.1 OVERVIEW | 7 |
| 4.2 MOTIVATION | 7 |
| 4.3 ASSIGNMENTS/NAMING RULES | 7 |
| 4.3.1 <i>Well-Known Elements</i> | 7 |
| 4.3.2 <i>Registered Elements</i> | 7 |
| 4.3.3 <i>Private Elements</i> | 8 |
| 4.4 RELATIONSHIP TO IANA | 8 |
| 5. WINA REGISTRATION PROCEDURE..... | 8 |
| 6. NAME AND NUMBER SPACES | 10 |
| 6.1 WSP..... | 10 |
| 6.1.1 <i>WSP Extended Content Type Codes</i> | 10 |
| 6.1.2 <i>WSP Header Field Name Assignments</i> | 10 |
| 6.1.3 <i>WSP Well Known Parameter Assignments</i> | 11 |
| 6.1.4 <i>WSP Extended Methods</i> | 11 |
| 6.2 WTP..... | 11 |
| 6.2.1 <i>WTP TPI (Transport Information Item)</i> | 11 |
| 6.2.2 <i>WTP Option TPI</i> | 11 |
| 6.3 WDP PORT NUMBERS..... | 12 |
| 6.4 INFORMATION ELEMENT IDENTITIES FOR USER DATA HEADER OVER USSD..... | 12 |
| 6.5 WAG | 12 |
| 6.5.1 <i>WBXML Public Document ID</i> | 12 |
| 6.5.2 <i>WMLScript Library ID</i> | 12 |
| 6.5.3 <i>WBMP image types</i> | 13 |
| 6.5.4 <i>PUSH Application ID</i> | 13 |
| 6.5.5 <i>EFI namespaces</i> | 13 |
| 6.6 WSG | 15 |
| 6.6.1 <i>WTLS KeyExchangeSuite</i> | 15 |
| 6.6.2 <i>WTLS BulkCipherAlgorithm</i> | 15 |
| 6.6.3 <i>WTLS MACAlgorithm</i> | 15 |
| 6.6.4 <i>WTLS Compression Method</i> | 15 |
| 6.6.5 <i>WTLS Parameter Index per key exchange algorithm</i> | 16 |
| 6.6.6 <i>ISO Object Identifiers (OIDs)</i> | 16 |

1. Scope

The purpose of this document is to define the administration mechanism for WAP namespaces. The naming authority acts as an interim authority for namespaces that later might be folded into an existing name space administration. The interim naming authority could potentially become a permanent body for WAP specific naming issues if no suitable alternatives are available.

Documents Status

This document is available online in the following formats:

PDF format at URL, <http://www.wapforum.org/>.

1.1 Copyright Notice

© Copyright Wireless Application Forum Ltd, 2000. Terms and conditions of use are available from the Wireless Application Protocol Forum Ltd. web site at <http://www.wapforum.org/docs/copyright.htm>.

1.2 Errata

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

1.3 Comments

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

2. References

2.1 Normative references

- [EFIFRM] External Functionality Interface Framework, WAP-231-EFI,
- [EFICDP] EFI Class Definition Process, WAP-263-EFICDP,
- [PUSHOTA] WAP Push OTA Protocol (WAP-189-PushOTA)
- [WAE] WAP Wireless Application Group, Wireless Application Environment Specification 30-April-1998
- [WAP] WAP Architecture Working Group "Wireless Application Protocol Architecture Specification", version 1.0
- [WDP] WAP Wireless Transport Group, Wireless Datagram Protocol Specification 30-April-1998
- [WAPWIM] WAP Wireless Identity Module Specification 05-Nov-1999
- [WTP] WAP Wireless Transport Group, Wireless Transaction Protocol Specification 30-April-1998
- [WAPGSMUD] "WAP over GSM USSD", WAP Forum, 30 April, 1998
- [GSM0340] "Digital cellular telecommunications system (Phase 2+), Technical realisation of the Short Message Service (SMS), Point-to-Point (PP)", GSM 03.40, ETSI
- [RFC2119] "Key words for use in RFCs to Indicate Requirement Levels", S. Bradner, March 1997. URL: <http://www.ietf.org/rfc/rfc2119.txt>

2.2 Informative references

3. Definitions and Abbreviations

3.1 Definitions

The following are terms and conventions used throughout this specification.

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 by [RFC2119].

3.2 Abbreviations

For the purposes of this specification, the following abbreviations apply.

| | |
|--------------|--|
| EFI | External Functionality Interface |
| ETSI | European Telecommunication Standardisation Institute |
| IANA | Internet Assigned Numbers Authority |
| IP | Internet Protocol |
| MS | Mobile Station |
| UDP | Unreliable Datagram Protocol |
| USSD | Unstructured Supplementary Service Data |
| USSDC | Unstructured Supplementary Service Data Centre |
| WAE | Wireless Application Environment |
| WAG | Wireless Application Group |
| WAP | Wireless Application Protocol |
| WDP | Wireless Datagram Protocol |
| WG | Working Group |
| WINA | WAP Interim Naming Authority |
| WSP | Wireless Session Protocol |
| WTP | Wireless Transaction Protocol |

4. Introduction

4.1 Overview

WAP is defined in a dynamic, yet undeveloped, environment. The WAP protocol stack has been developed in a very short time frame, and not all features needed for commercial deployment have been fully specified. Standardisation of protocols and content is of utmost importance, but it needs to be balanced with a dynamic process for the definition of new parameters and content.

The role of the naming authority in the WAP space is to ensure a swift and harmonious deployment of WAP-based services, and to make it possible to develop extensions to the core WAP protocol suite.

The final Naming Authority has not yet been defined and contracted, but in the mean time WAP namespaces are administered by the WAP Interim Naming Authority (WINA). The Specification Committee assumes this role.

4.2 Motivation

The WAP protocols have been carefully designed to allow for extensions in certain areas. New functionality can be added through content types, parameters and access methods without any changes to the base protocols. The registration process is needed to ensure that the set of such values is developed in an orderly, well specified, and public manner.

Most of the WAP specifications have to be thoroughly defined in working groups, but in many cases companies must be allowed to try new solutions to new problems. In addition, future domain specific solutions will require new solutions that not all WAP implementations need to support (e.g. Telematics-specific content types). WAP needs mechanisms to allow this dynamic development to happen, and still allow the WAP community to regain some control and visibility. When one of these “proprietary or experimental extensions” turns out to be generally useful it can be incorporated into the next version of the core specifications.

A lack of naming authority would be a severe threat to the continuous development of the protocol/application space in WAP, and would very likely lead to interoperability problems in the future.

4.3 Assignments/Naming Rules

To simplify the management of assigned names and numbers, a simple categorisation scheme is implemented. This categorisation scheme is the basis for the proposed name and number management rules.

The naming is based on a three-layered structure, with Well-Known, Registered and Private names or codes. The assignment of names, parameters and codes is done on an as-needed basis. The code spaces are sufficiently large to allow for all the assumed needs within this version of the protocol suite.

This document uses the word “element” to identify various entities like content type, header page, information item, parameter, etc.

4.3.1 Well-Known Elements

Well-Known elements are part of the core WAP specifications, and reserved for allocation by the WAP WG that controls the specification. A new Well-Known element can be defined by submitting a work item or change request to the appropriate WAP working group. The WG defines the exact format of the element, or references a well-defined entry in a specification from another standardisation organisation. The Naming Authority approves Well-Known elements based on proposals from an appropriate working group.

4.3.2 Registered Elements

Individual companies that are members of WAP can apply for Registered elements. The description of an element does not have to be approved by the appropriate Working Group, but at least a general description of the functionality and format must be submitted to the WAP Naming Authority. The Naming Authority Registers elements and assigns any

necessary numbers. WINA should refuse registration of elements if they are in conflict with the general WAP architecture or with general naming conventions.

4.3.3 Private Elements

The private element space is unregulated and any company can announce or use a Private (or experimental) Element. No description of the functionality needs to be submitted to WINA. Private elements are experimental in nature, and could be used for prototyping of functionality.

4.4 Relationship to IANA

The Internet Assigned Numbers Authority already manages a number of name and code spaces. These include for example:

- Content Types
- Port Numbers
- Character Sets

WINA is a complement to IANA, and deals with spaces not presently administered by IANA. These are either unique code spaces, or special encoding of already known name spaces.

5. WINA Registration Procedure

This procedure creates a distinction between “Well-Known WAP elements” and “Proprietary Extensions”. A Well-Known WAP element is defined by the appropriate Working Group, and can be made mandatory in a version of the protocol. The definition of proprietary (registered) extensions is outlined below.

In order to allocate a Registered Element, an openly available description of the element is needed. A precise specification is recommended, but not required. The proposed element should be sent for registration to WAP, using the WINA Request Web Interface found at <http://www.wapforum.org/wina/>.

The table below lists the type of information that should be submitted in the proposal. It should, among other things, include a short technical description of how the element interacts with rest of the WAP protocol suite.

| Information to provide | Explanation |
|---|--|
| The name of the element | For example “x-wap.specialContent” |
| The proposed encoding of the element | For example “0x5A01” |
| The specification it references | For example “WSP, April 30, 1998” |
| The name space in which the registrations is done | For example “WSP Content Type Codes” |
| A description of the use of the element in the WAP space | For example “The content type is used to encode car-related data and events”. Some of the encoding mechanisms might also be explained. |
| Possible previous registration of the element with other naming authorities | For example “The content type name “x-wap.specialContent” is already registered with IANA, as required with WINA” |
| Possible related documents | For Example “the format and encoding of the content type is described in the specification “Car Telematics, Version 1.4, Car Standards Organisation, 1998” “ |

Private elements are not formally registered and can be used for experimental purposes, prototypes and closed environments.

The WINA manages a set of Web pages on the WAP Forum web site (<http://www.wapforum.org/wina>). These pages publish the Well-Known and Registered elements, as well as announced use for Private elements.

The pages also include the information submitted with the application, as well as a mechanism to submit requests for element assignment.

6. Name and Number spaces

6.1 WSP

6.1.1 WSP Extended Content Type Codes

In order to register a content type as a WSP content type, and assign a code to it, it must first be registered with IANA. Content types are divided into:

| Classes of Content Types | Code Values | Encoding |
|--------------------------|-----------------------|---|
| Well-known-Content-Types | 0x00 – 0xFF | One byte: value Two byte: length + value |
| Registered-Content-Types | 0x0100 – 0xFFFF | Three byte: length + 2 byte value |
| Private-Content-Types | 0x010000 – 0xFFFFFFFF | Four byte: length + 3 byte value |

The Naming Authority will assign Registered content type codes on an as-requested basis.

6.1.2 WSP Header Field Name Assignments

Header fields are divided into code pages. The code pages are negotiated at session setup. The Well-Known Header Field Names are owned by the WPG/WSP working group.

6.1.2.1 Default Header Code Page

The default header code page includes generic header codes, applicable in most environments.

| Classes of Header-Fields | Code Values |
|--------------------------|-------------|
| Well-known-Header-Fields | 0x00 – 0x7F |

All the header names and codes on the default code page are owned by the WPG/WSP working group.

6.1.2.2 Code Page Name Assignments

Well-Known header code page names are proposed to WINA by the WSP working group. The name of the page can have any format, except starting with “x-“ (reserved for experimental) or “v-“ (reserved for registered).

WINA registers code page names of the form “v-CompanyToken-“. “CompanyToken” is a short abbreviation that uniquely identifies the company, e.g. a stock ticker symbol or some other well-known abbreviation (like “NOK” or “IBM”). The “CompanyToken” must be at least 2 characters in length. In order to register a code page name at least one header code on the page must be defined.

Experimental users do not need to register, but the code page name must start with “x-CompanyToken-“.

6.1.2.3 Code Page header field name assignments

The appropriate working group administers Well-Known code pages (currently, the WAP WPG working group).

Registered and Private Code pages are administered by the company/organisation that has proposed them. Header codes of registered code pages should be registered with WAP for public visibility. The registration should include at least a short description of the purpose of the header.

Experimental Code Pages, as well as the header codes in an experimental code page, are administered by the entity that created the page.

6.1.3 WSP Well Known Parameter Assignments

Parameters have various types: integer, text, version, etc. The parameters 0x00 – 0x09 are defined in [WSP].

| Classes of Parameters | Code Values | Encoding |
|-----------------------|----------------------|--|
| Well-known-Parameters | 0x00 – 0xFF | One byte: value Two byte: length + value |
| Registered-Parameters | 0x0100 – 0xFFFF | Three byte: length + 2 byte value Three byte: length + 2 byte value |
| Private-Parameters | 0x010000 – 0xFFFFFFF | Four byte: length + 3 byte value |

Parameters are related to Content-Types, and can have different interpretation with different Content Types. A code for a Well-Known-Parameter can only be assigned if it is used as part of a Well-Known-Content type.

In order to register a parameter, a short description on its use should be submitted to the Naming Authority.

The Well-Known Parameters are owned by the WPG/WSP working group.

6.1.4 WSP Extended Methods

Well-Known Extended Methods names are proposed to WINA by the WSP working group. The name of the page can have any format, except starting with "x-" (reserved for experimental) or "v-" (reserved for registered).

WINA registers names of extended methods of the form "v-CompanyToken-". "CompanyToken" is a short abbreviation that uniquely identifies the company, e.g. a stock ticker symbol or some other well-known abbreviation (like "MOT" or "UP"). The "CompanyToken" must be at least 2 characters in length.

Experimental users do not need to register, but the extended method name must start with "x-CompanyToken-".

6.2 WTP

6.2.1 WTP TPI (Transport Information Item)

The number space for the Transport Information Item is very limited as the protocol allows for only 16 values. All the values are in the Well-Known numbering space and must be proposed by the WPG/WTP working group.

| Classes of TPI | Code Values |
|----------------|-------------|
| Well-known-TPI | 0x00 – 0x0F |

6.2.2 WTP Option TPI

The number space of the Option TPI allows for 256 values, and these are divided between Well-Known, Registered and Private/Experimental.

If the number space is exhausted it is still possible to assign a new code page for options by allocating another TPI code for the purpose.

In order to register an Option TPI code a short description of the functionality must be submitted.

The Well-Known Option TPIs are owned by the WPG/WTP working group.

| Classes of Abort Reasons | Code Values |
|--------------------------|-------------|
| Well-known-Option-TPI | 0x00 – 0x7F |
| Registered-Option-TPI | 0x80 – 0xDF |
| Private-Option-TPI | 0xE0 – 0xFF |

6.3 WDP Port Numbers

The WDP port space is equal to the UDP port space. IANA acts as the assignment authority for this name space.

6.4 Information Element Identities for User Data Header over USSD

The WAP specification [WAPGSMUD] defines how the User Data Header shall be mapped onto the GSM Unstructured Supplementary Service Data (USSD). The User Data Header is specified in [GSM 03.40] for the GSM Short Message Service (SMS). It's an extensible mechanism for adding control information related to the user data in the short message. The Information Element is a container mechanism that holds the control information. Each Information Element has a unique identifier. For GSM SMS the identifiers are defined in [GSM 03.40]. When the User Data Header is used over USSD the Information Element identifiers are defined here.

| Information Element | Identity |
|---------------------------------------|-------------|
| Same as GSM 03.40 sections 9.2.3.24 | 0x00 – 0xDF |
| USSD Dialogue Control Protocol (UDCP) | 0xE0 |
| Reserved for future use | 0xE1 – 0xFF |

The [WAPGSMUD] specified the USSD Dialogue Control Protocol (UDCP). It's used to hide the alternating half-duplex nature of a USSD dialogue.

6.5 WAG

6.5.1 WBXML Public Document ID

Registered Document ID tokens must specify a DTD and a public document type identifier.

All WBXML tokens assigned for a given document type are specified as part of the specification of that document type. In other words, the assignment of a new token in a document type is owned by the entity who reserved/defined the document type.

| Classes of Document Id's | Code Values | Encoding |
|--------------------------|--------------------|------------|
| Well-known-Document-Id | 0x00 – 0x7F | One byte |
| Registered-Document-Id | 0x80 – 0x3FFF | Two byte |
| Private-Document-Id | 0x4000 – 0x1FFFFFF | Three byte |

The Naming Authority will allocate registered document ID tokens on an as-requested basis.

6.5.2 WMLScript Library ID

Well-Known library names and Id values are proposed to WINA by the WAG working group.

The name of the library can have any format, except starting with "x-" (reserved for private and experimental) or "v-" (reserved for registered).

WINA registers library names of the form "v-CompanyToken-". "CompanyToken" is a short abbreviation that uniquely identifies the company, e.g. a stock ticker symbol or some other well-known abbreviation (like "MOT", "IBM" or "UP").

Experimental users do not need to register, but the library name must start with "x-CompanyToken-".

In order to register a WML Script Library, a short description on its use should be submitted to the Naming Authority. It must include at least a list of functions and their function codes as well as a general description of the functionality.

| Classes of Library Id's | Id Values | Encoding |
|-------------------------|-----------------|----------|
| Well-known-Library-Id | 0x00 – 0xFF | One byte |
| Registered-Library-Id | 0x0100 – 0x7FFF | Two byte |
| Private-Library-Id | 0x8000 – 0xFFFF | Two byte |

The Naming Authority will allocate registered library ID tokens on an as-requested basis.

6.5.3 WBMP image types

All WBMP image types are Well-Known and specified by the WAG/WBMP working group.

6.5.4 PUSH Application ID

The registrant submits an URI to WINA for registration, together with the name and the description of the application, which the URI represents.

- WINA verifies that the URI is not registered
- WINA assigns the next available code for the URI

General Rules for URI:

- MUST be rfc2396 compliant
- URN SHOULD be used for location independent URI [rfc2141][rfc2611]

Well-Known:

The application as represented by the URI must be defined in any WAP specification. The registration request must be submitted by the WAP working group, which is responsible for the specification. URNs are REQUIRED for URI and NID in URN [rfc2141] MUST be 'x-wap-application'. Code in the code range must be assigned by WINA.

Code range: 0x0000 - 0x7FFF.

Example: Application ID for WML User Agent is a URI of urn:x-wap-application:wml.ua and a code of 0x02.

Registered:

URI can be registered with WINA by any entity. The registrant may choose any desirable URI except that the URN with NID 'x-wap-application' cannot be used. Code in the code range must be assigned by WINA.

Code range: 0x8000 - 0xFF000000.

Private/Experimental:

Any URI can be used at will, but any registered URIs in Well-Known and Registered category SHOULD NOT be used. Registration is not required. Any code in the code range can be selected as desirable.

Code range: 0xFF000001 - 0xFFFFFFFF.

6.5.5 EFI namespaces

The registrant must develop an EFI Class in accordance with the EFI Class Definition Process [EFICDP]. Upon completion, the registrant submits the EFI Class name to WINA for registration, together with a description of the class (the class specification) represented by the name.

- WINA verifies that the Class name is not registered and follows the General Rules below
- WINA assigns the name for the Class specification

General Rules for EFI Class names:

- Class names MUST be unique within the scope of EFI

- Class names MUST NOT be one of the reserved words defined in the EFI Framework [EFIFRM]
- Well-known class names MUST NOT contain company names or registered trademarks.
- In the event that there is a clash between class names, WAP Forum member companies SHALL have precedence in selecting the class name.

EFI Classes may be submitted to WINA as “Registered” or “Well-known” classes. The implications for these two types of classes are as follows.

Registered:

- WINA provides protection of the class name only after submission
- MUST have a named owner that maintains the class specification
- The owner can make changes to the class specification without control of the WAP Forum
- It is also under the owner's control to provide any means for ensuring interoperability

The class specification written according to the above is chosen to be under control of the issuer only or to be proposed as a well-known WAP-Class-Specification. If a class fails to get support, it can be registered as a registered class and left in that state.

When the class specification is in the state registered, it needs to have an assigned owner. The owner could be a person, a company or an organisation.

Registered class specifications are outside of the WAP-Forum's purview and conformance testing needs to be driven through the owner if wanted.

Well-known:

- Uses the standard name space publicly available through WAP-Forum web sites
- Class specifications are developed through standard WAP process
- Reliability and interoperability is provided by WAP-Forum means like test assertions and conformance statements

Class specifications are the responsibility of WAP Forum, specifically WAG. If there is an appropriate drafting committee WAP Forum can assign the responsibility to it. WAG is understood as a placeholder for the charter-documents of any group appointed by WAG.

In order to register a vendor specific name, the following information should be submitted to the Naming Authority:
vendor-class-name: Desired name for the vendor-specific class

Company name: Name of the company

Contact person: Name of the contact person within the company with eMail address of the contact person

Description: Optionally, a short description of how this class will be used.

Registered class will be accessed/placed within the EFI namespace in the following form: "efi:/vnd.<vendor-name>", where <vendor-name> is the name of the registered vendor.

The names of the classes are case-insensitive

| Classes | Allowed characters | Encoding |
|--------------------|--------------------------|--------------------|
| Well-Known classes | [A..Z, a..z, 0..9, "_"] | 1 to 16 characters |
| Registered classes | [A..Z, a..z, 0..9, "_"] | 1 to 16 characters |

6.6 WSG

6.6.1 WTLS KeyExchangeSuite

Well-Known Key Exchange Suites and assigned number values are proposed to WINA by the WSG working group. The name of the exchange suite can have any format, except starting with "x-" (reserved for private and experimental) or "v-" (reserved for registered).

Experimental users do not need to register, but the exchange suite name must start with "x-CompanyToken-".

In order to register a WTLS Key Exchange Suite, a short description on its use should be submitted to the Naming Authority. It must include at least a general description of the functionality, and the key size limit.

| Classes of Key Exchange Suites | Code Values | Encoding |
|---------------------------------|-------------|----------|
| Well-known-Key-Exchange-Suite | 0x00 – 0xDF | One byte |
| Private-Bulk-Key-Exchange-Suite | 0xE0 – 0xFF | one byte |

6.6.2 WTLS BulkCipherAlgorithm

Well-Known Bulk Cipher Algorithms and assigned number values are proposed to WINA by the WSG working group.

The name of the algorithm can have any format, except starting with "x-" (reserved for experimental) or "v-" (reserved for private and registered).

Experimental users do not need to register, but the algorithm name must start with "x-CompanyToken-".

In order to register a WTLS Bulk Cipher Algorithm, a short description on its use should be submitted to the Naming Authority. It must include at least a general description of the functionality and

- the type
- key material
- expanded key material
- effective key bits
- IV size
- block size

| Classes of Bulk Cipher Algorithms | Code Values | Encoding |
|-----------------------------------|-------------|----------|
| Well-known-Bulk-Cipher-Algorithm | 0x00 – 0xDF | One byte |
| Private-Bulk-Cipher-Algorithm | 0xE0 – 0xFF | one byte |

6.6.3 WTLS MACAlgorithm

Well-Known MAC Algorithms and assigned number values are proposed to WINA by the WSG working group.

The name of the algorithm can have any format, except starting with "x-" (reserved for private and experimental) or "v-" (reserved for registered).

Experimental users do not need to register, but the algorithm name must start with "x-CompanyToken-".

In order to register a WTLS MAC Algorithm, a short description on its use should be submitted to the Naming Authority. It must include at least a general description of the functionality, and the key and hash size.

| Classes of MAC Algorithms | Index Values | Encoding |
|---------------------------|--------------|----------|
| Well-known-MAC-Algorithm | 0x00 – 0xDF | One byte |
| Private-MAC-Algorithm | 0xE0 – 0xFF | One byte |

6.6.4 WTLS Compression Method

Well-Known Compression Methods and assigned number values are proposed to WINA by the WSG working group.

The name of the method can have any format, except starting with "x-" (reserved for experimental) or "v-" (reserved for registered).

WINA registers method names of the form "v-CompanyToken-". "CompanyToken" is a short abbreviation that uniquely identifies the company, e.g. a stock ticker symbol or some other well-known abbreviation (like "MOT", "IBM" or "NOK").

Experimental users do not need to register, but the method name must start with "x-CompanyToken-".

In order to register a WTLS Compression Algorithm, a short description on its use should be submitted to the Naming Authority. It must include at least a general description of the functionality.

| Classes of Compression Methods | Index Values | Encoding |
|--------------------------------|--------------|----------|
| Well-known-Compression-Method | 0x00 – 0x7F | One byte |
| Registered-Compression-Method | 0x80 – 0xDF | One byte |
| Private-Compression-Method | 0xE0 – 0xFF | One byte |

6.6.5 WTLS Parameter Index per key exchange algorithm

The parameter space is private for each key exchange or cipher algorithm.

| Classes of Parameter Index | Index Values | Encoding |
|----------------------------|--------------|----------|
| Well-known-Parameter-Index | 0x00 – 0x7F | One byte |
| Registered-Parameter-Index | 0x80 – 0xDF | One byte |
| Private-Parameter-Index | 0xE0 – 0xFE | One byte |
| Well-Known-Parameter-Index | 0xFF | One byte |

6.6.6 ISO Object Identifiers (OIDs)

WAP's security group WSG has applied for, and received, a root OBJECT IDENTIFIER (OID) from ISO/ITU-T. This OID and its descendants can be used in any specification, which needs to define protocol data units distinguishable with OIDs. WINA will be in charge of the registration procedure for allocating object identifiers below its root, and probably also some organization of these identifiers at the top-level. Refer to the WINA Web page for the up-to-date ISO Object Identifiers definitions. The following is just an example.

WAP's root object identifier:

```
wap OBJECT IDENTIFIER ::= {joint-isu-itu-t(2) identified-organizations(23) 43}
```

WSG has defined the following identifiers:

```
wap-modules OBJECT IDENTIFIER ::= {wap 0}
```

-- for use when defining ASN.1 modules. Can be used by any WAP group.

```
wap-wsg OBJECT IDENTIFIER ::= {wap 1}
```

-- Used in WIM specification

```
wap-at OBJECT IDENTIFIER ::= {wap 2}
```

-- Attributes branch, for use by any WAP group

```
wap-ce OBJECT IDENTIFIER ::= {wap 3}
```

-- Certificate extensions branch, for use by any WAP group

```
wap-oc OBJECT IDENTIFIER ::= {wap 4}
```

-- Object class branch, for use by any WAP group

In "public" branches (i.e. not the {wap-wsg} branch), the following object identifiers has been defined:

wap-ce-domainInformation OBJECT IDENTIFIER ::= {wap-ce 1}

-- Certificate extension

wap-oc-wapEntity OBJECT IDENTIFIER ::= {wap-oc 1}

-- Auxillary Object Class

Process to register OIDs

The WINA Web Page will be used to register the OIDs. When a request is received by WINA, the request will be forwarded to the WSG group. When the group approves the OID, the requester will be informed of the approval and the registered OIDs will be updated on the WINA web pages. An OID is not considered to be registered until WSG approves it and is published on the WINA web page. The following information is required by WINA for OID registration:

OID-name: Desired name for the OID

Company name: Name of the company

Contact person: Name of the contact person within the company with eMail address of the contact person

Description: Optionally, a short description of how this OID will be used.

Appendix A Specification-track Document History

Document: WAP WINA Process Document

Document Identifier: WAP-212

Base Specification Approval Date: Feb 2002