WAP Push Message

Version 16-August-1999

Wireless Application Protocol
Push Message Specification

Contents

1. Scope

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 providing new services. To enable operators and manufacturers to meet the challenges in advanced services, differentiation, and fast/flexible service creation, WAP defines a set of protocols in transport, session and application layers. For additional information on the WAP architecture, refer to "*Wireless Application Protocol Architecture Specification*" [WAP].

This specification defines the push message, which is used by a WAP push application to deliver the content to a WAP client. In particular, it defines the following:

- General format of the push message
- Headers of the push message
- Body of the push message
- Proxy rules for header handling

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

[PushOTA] "WAP Push OTA Protocol", WAP Forum, 08-Nov-1999.
URL: http://www.wapforum.org

[PushPAP] "WAP Push Access Protocol", WAP Forum, 08-Nov-1999.
URL: http://www.wapforum.org

[RFC822] " Standard for the Format of ARPA Internet Text Messages", D. Crocker, August 1982,
URL: http://www.ietf.org/rfc /rfc822

[HTTP] "Hypertext Transfer Protocol – HTTP/1.1", R. Fielding, et al. June 1999
URL: http://www.ietf.org/rfc/rfc2616.txt

[WINA] "WAP Interim Naming Authority", WAP Forum,
URL:http://www.wapforum.org/wina/

3.2 Informative References

[WAE] "Wireless Application Environment Specification", WAP Forum, 04-Nov-1999.
URL: http://www.wapforum.org

[WAP] "Wireless Application Protocol Architecture Specification", WAP Forum, 30-Apr-1998. URL: http://www.wapforum.org

[WSP] "Wireless Session Protocol", WAP Forum, 05-Nov-1999.
URL: http://www.wapforum.org

4. Definitions and Abbreviations

4.1 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].

Application - A value-added data service provided to a WAP Client. The application may utilise both push and pull data transfer to deliver content

Application-Level Addressing - the ability to address push content between a particular user agent on a WAP client and push initiator on a server.

Bearer Network - a network used to carry the messages of a transport-layer protocol between physical devices. Multiple bearer networks may be used over the life of a single push session.

Client – in the context of push, a client is a device (or service) that expects to receive push content from a server. In the context of pull a client, it is a device initiates a request to a server for content or data. See also “device”.

Contact Point – address information that describes how to reach a push proxy gateway, including transport protocol addres and port of the push proxy gateway.

Content - subject matter (data) stored or generated at an origin server. Content is typically displayed or interpreted by a user agent on a client. Content can both be returned in response to a user request, or being pushed directly to a client.

Content Encoding - when used as a verb, content encoding indicates the act of converting a data object from one format to another. Typically the resulting format requires less physical space than the original, is easier to process or store, and/or is encrypted. When used as a noun, content encoding specifies a particular format or encoding standard or process.

Content Format – actual representation of content.

Context – an execution space where variables, state and content are handled within a well-defined boundary.

Device – is a network entity that is capable of sending and/or receiving packets of information and has a unique device address. A device can act as either a client or a server within a given context or across multiple contexts. For example, a device can service a number of clients (as a server) while being a client to another server.

End-user - see “user”

Extensible Markup Language - is a World Wide Web Consortium (W3C) recommended standard for Internet mark-up languages, of which WML is one such language. XML is a restricted subset of SGML.

Multicast Message - a push message containing a single OTA client address which implicitly specifies more than OTA client address.

Push Access Protocol - a protocol used for conveying content that should be pushed to a client, and push related control information, between a Push Initiator and a Push Proxy/Gateway.

Push Framework- - the entire WAP push system. The push framework encompasses the protocols, service interfaces, and software entities that provide the means to push data to user agents in the WAP client.

Push Initiator - the entity that originates push content and submits it to the push framework for delivery to a user agent on a client.

Push OTA Protocol - a protocol used for conveying content between a Push Proxy/Gateway and a certain user agent on a client.

Push Proxy Gateway - a proxy gateway that provides push proxy services.

Push Session - A WSP session that is capable of conducting push operations.

Server - a device (or service) that passively waits for connection requests from one or more clients. A server may accept or reject a connection request from a client. A server may initiate a connection to a client as part of a service (push).

User - a user is a person who interacts with a user agent to view, hear, or otherwise use a rendered content. Also referred to as end-user.

User agent - a user agent (or content interpreter) is any software or device that interprets resources. This may include textual browsers, voice browsers, search engines, etc.

XML – see *Extensible Markup Language*

4.2 Abbreviations

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

CPI	Capability and Preference Information
DNS	Domain Name Server
DTD	Document Type Definition
HTTP	Hypertext Transfer Protocol
IANA	Internet Assigned Numbers Authority
IP	Internet Protocol
OTA	Over The Air
PAP	Push Access Protocol
PI	Push Initiator
PPG	Push Proxy Gateway
QOS	Quality of Service
RDF	Resource Description Framework
RFC	Request For Comments
SGML	Standard Generalized Markup Language
SI	Service Indication
SIA	Session Initiation Application
SIR	Session Initiation Request
SL	Service Loading
SSL	Secure Socket Layer
TLS	Transport Layer Security
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UTC	Universal Time Co-ordinated
WAP	Wireless Application Protocol
WDP	Wireless Datagram Protocol
WSP	Wireless Session Protocol
WBXML	WAP Binary XML
WINA	WAP Interim Naming Authority
WTLS	Wireless Transport Layer Security
XML	Extensible Mark-up Language

5. Introduction

The architecture consists of a distributed client/server application, with a server residing in the *push proxy gateway (PPG)* or a *push initiator (PI)*, and a client residing in the mobile device. It is the *push initiator* that initially intends to send a push message to the client. The *push initiator* typically first sends the message by using the Push Access Protocol (PAP) [PushPAP] to the PPG through the wired network and the PPG sends the message by using the Push OTA Protocol [PushOTA] over the wireless network.

Every push message contains headers and a body. The push initiator originally creates the push message and sends it to the PPG by using an appropriate mechanism in PAP. The PPG examines the message and performs the required encoding and transformation. In the process, it generally should not remove any headers or the body of the message, although it may perform encoding and/or transforming. The PPG, however, may add additional headers to the message to enable the needed OTA services.

The push message, including the headers and the body, is delivered hop by hop, optionally encoded or transformed, but the information carried in the headers and the body is generally preserved end to end (i.e., from a PI to a WAP client).

The overall push architecture is outlined in Figure 1.

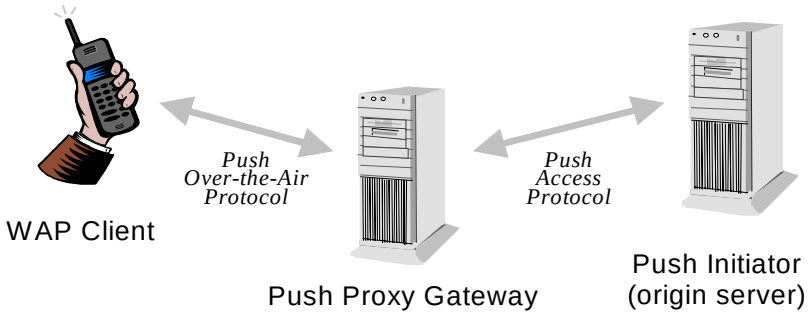

Figure 1: WAP Push Architecture

6. Push Message Definition

6.1 Message Format

A push message contains headers and a body. It uses the generic message format of RFC 822 [RFC822] for transferring textual entities, but allows binary message bodies. The message consist of zero or more headers, an empty line (i.e., a line with nothing preceding the CRLF) indicating the end of the header fields, and an optional message body.

The message headers are defined in 6.2. The message body is defined in 6.3.

6.2 Message Headers

6.2.1 Generic Headers

The message headers in this category are based on the Internet message headers in common use. These headers are defined in [HTTP]. The push message is equivalent to a response message in HTTP 1.1 when the semantics of each HTTP header is examined.

6.2.1.1 Age

As defined in [HTTP].

6.2.1.2 Cache-Control

As defined in [HTTP], but only the cache-response-directives are applicable.

6.2.1.3 Content-Disposition

As defined in [HTTP].

6.2.1.4 Content-Encoding

As defined in [HTTP].

6.2.1.5 Content-Language

As defined in [HTTP].

6.2.1.6 Content-Length

As defined in [HTTP].

6.2.1.7 Content-Location

As defined in [HTTP].

6.2.1.8 Content-MD5

As defined in [HTTP].

6.2.1.9 Content-Range

As defined in [HTTP].

6.2.1.10 Content-Type

As defined in [HTTP]. This header is REQUIRED.

6.2.1.11 Date

As defined in [HTTP].

6.2.1.12 Etag

As defined in [HTTP].

6.2.1.13 Expires

As defined in [HTTP].

6.2.1.14 From

As defined in [HTTP].

6.2.1.15 Last-Modified

As defined in [HTTP].

6.2.1.16 Transfer-Encoding

As defined in [HTTP].

6.2.1.17 Vary

As defined in [HTTP].

6.2.2 WAP Headers

The headers in this category are WAP headers. Those headers start with "X-Wap-" prefix. The header definition rules in this sub-section follow the rules in [HTTP].

6.2.2.1 X-Wap-Application-Id

This header is used for application id, usage of which is defined in [PushOTA].

```
X-Wap-Application-Id = “X-Wap-Application-Id” “:” app-id
app-id = ( absoluteURI [“;” “app-encoding=” app-assigned-code] |
                app-assigned-code )
app-assigned-code = 1*8HEX
```

If X-WAP-Application-Id is not present, the application id for the WML User Agent is assumed. WINA [WINA] handles registration of `app-assigned-code`.

6.2.2.2 X-Wap-Content-URI

This header is semantically equivalent to the HTTP Request-URI, and is used in caching [WAPCA] in the same way as the HTTP Request-URI is used.

```
X-Wap-Content-URI  = “X-Wap-Content-URI” “:” absoluteURI
```

6.2.2.3 X-Wap-Initiator-URI

This header identifies the WAP push initiator. If `X-Wap-Content-URI` is present, its value is considered as the default value for `X-Wap-Initiator-URI`. If `X-Wap-Content-URI` is not present, the default value of `X-Wap-Initiator-URI` is considered to be the same as the value of `Content-Location,` if present.

```
X-Wap-Initiator-URI = “X-Wap-Initiator-URI” “:” URI
```

6.2.3 Header Extensions

6.2.3.1 WAP Header Extensions

All WAP header extensions MUST have X-Wap- prefix and the new headers MUST be registered with WINA [WINA].

6.2.3.2 User Header Extensions

If the implementation does not want the headers to be registered, the new headers MUST be prefixed by X- and MUST NOT use X-Wap- prefix.

6.2.3.3 Non-Normative Internet Message Headers

Although some implementations MAY use other Internet message headers not specified in this document, those headers MAY be ignored by some other implementations.

6.3 Message Body

The message body can be any MIME content type, including multipart MIME content type, and optionally encoded or transfer encoded. Any handling process for the message body MUST support a non-nested multipart content type, and MAY support nested multipart content types.

7. Proxy Rules

Any proxy, including a WAP Push Proxy Gateway, MUST pass on any push message headers defined in this specification, unless it is known that those headers can be removed without changing the meaning of the message.

It MAY change the field values of the `Content-` headers (see section 6.2.1) and MAY delete or replace those headers as the result of message encoding, transforming, or optimisation.

8. Static Conformance Requirements

This static conformance clause defines a minimum set of features that should be implemented to support WAP push message. A feature can be optional (O), mandatory (M) or conditional (C (<condition>)). If optional/conditional features have labels (O.<n> or C.<n>), support of at least one in the group of options labelled by the same number is required.

8.1 Features

Item	Functionality	Reference	Status
MSG-010	Generic Headers	6.2.1	O
MSG-011	Content-Type header	6.2.1.10	M
MSG-020	WAP Headers	6.2.2	O
MSG-030	Header Extensions	6.2.3	O
MSG-040	Message Body	6.3	O
MSG-041	Non-nested multipart content type support	6.3	C (MSG-040)
MSG-042	Nested multipart content type support	6.3	O
MSG-050	Proxy Rules	7	M