Push Message

Approved Version 2.2 – 09 Aug 2011

Open Mobile Alliance

OMA-TS-Push_Message-V2_2-20110809-A

Contents

Figures

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. References

2.1 Normative References

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

[PushOTA] "Push OTA Protocol Specification". Open Mobile Alliance™. OMA-TS-PushOTA-V2_2.
URL:http://www.openmobilealliance.org/

[PushPAP] "Push Access Protocol Specification". Open Mobile Alliance™. OMA-TS-PAP-V2_2
URL:http://www.openmobilealliance.org/

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

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

[RFC2396] "Uniform Resource Identifiers (URI): Generic Syntax", T. Berners-Lee et al., August 1998,
URL: http://www.ietf.org/rfc/rfc2396.txt/

[RFC4234] "Augmented BNF for Syntax Specifications: ABNF". D. Crocker, Ed., P. Overell. October 2005,
URL:http://www.ietf.org/rfc/rfc4234.txt

[SCRRULES] "SCR Rules and Procedures", Open Mobile Alliance™, OMA-ORG-SCR_Rules_and_Procedures,
URL:http://www.openmobilealliance.org/

[WAPCache] "WAP Caching Model", WAP-120-WAPCachingMod-19990211-a, WAP Forum™. URL:
http://www.wapforum.org/

2.2 Informative References

[OMADICT] "Dictionary for OMA Specifications", Open Mobile Alliance™,
OMA-ORG-Dictionary, URL:http://www.openmobilealliance.org/

[OMNA] "OMA Naming Authority". Open Mobile Alliance™.
URL: http://www.openmobilealliance.org/tech/OMNA.aspx

[PushArch] "Push Architectural Overview". Open Mobile Alliance™. OMA-AD-Push-V2_2
URL:http://www.openmobilealliance.org/

[WAE] "Wireless Application Environment Specification", WAP Forum™.
WAP-236-WAESpec. URL: http://www.wapforum.org/

[WAP] "Wireless Application Protocol Architecture Specification". WAP Forum™. WAP-210-WAPArch.
URL: http://www.wapforum.org/

[WSP] "Wireless Session Protocol". WAP Forum™. WAP-230-WSP. URL: http://www.wapforum.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

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

3.3 Abbreviations

ABNF	Augmented Backus-Naur Form
HTTP	Hypertext Transfer Protocol
OTA	Over The Air
OTA-HTTP	(Push) OTA over HTTP
OTA-WSP	(Push) OTA over WSP
OTA-SIP	(Push) OTA over SIP
PAP	Push Access Protocol
PI	Push Initiator
PPG	Push Proxy Gateway
RFC	Request For Comments
SIP	Session Initiation Protocol
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
WAP	Wireless Application Protocol
WINA	WAP Interim Naming Authority
WML	Wireless Markup Language

4. 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.

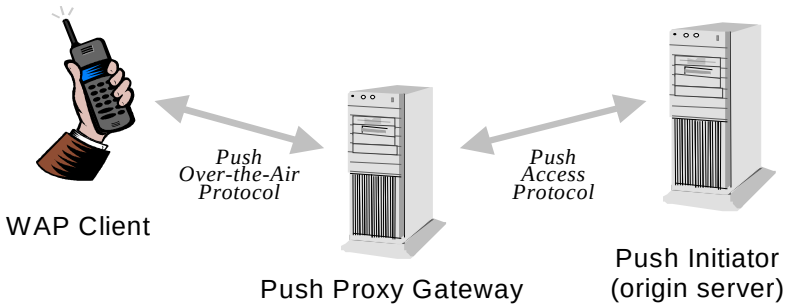

Figure 1: WAP Push Architecture

4.1 Version 2.2

This enabler release defines push message formatting for delivery via the OTA-SIP Push-OTA protocol variant, as necessary to support encapsulation of push messages in SIP MESSAGE or MSRP.

5. Push Message Definition

This section is normative.

5.1 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 consists of one 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 5.2. The message body is defined in 5.3.

5.2 Message Headers

5.2.1 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. Each header is OPTIONAL unless stated otherwise.

5.2.1.1 Age

As defined in [HTTP].

5.2.1.2 Cache-Control

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

5.2.1.3 Content-Disposition

As defined in [HTTP].

5.2.1.4 Content-Encoding

As defined in [HTTP].

5.2.1.5 Content-Language

As defined in [HTTP].

5.2.1.6 Content-Length

As defined in [HTTP].

5.2.1.7 Content-Location

As defined in [HTTP].

5.2.1.8 Content-MD5

As defined in [HTTP].

5.2.1.9 Content-Range

As defined in [HTTP].

5.2.1.10 Content-Type

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

5.2.1.11 Date

As defined in [HTTP].

5.2.1.12 Etag

As defined in [HTTP].

5.2.1.13 Expires

As defined in [HTTP].

5.2.1.14 Last-Modified

As defined in [HTTP].

5.2.1.15 Transfer-Encoding

As defined in [HTTP].

5.2.2 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].

5.2.2.1 X-Wap-Application-Id

This header is used for application id, usage of which is defined in [PushOTA]. The ABNF [RFC2234] format for this header is as follows:

```
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*8HEXDIG
; absolute URI is as defined in [RFC2396]
```

OMNA [OMNA] handles registration of `absoluteURI` and `app-assigned-code`.

5.2.2.2 X-Wap-Content-URI

This header is used as a substitute for the Request-URI [HTTP] when push content is placed in the cache [WAPCache]. The ABNF [RFC2234] format for this header is as follows:

```
X-Wap-Content-URI  = "X-Wap-Content-URI" ":" absoluteURI
       ; absolute URI is as defined in [RFC2396]
```

5.2.2.3 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. The ABNF [RFC2234] format for this header is as follows:

```
X-Wap-Initiator-URI = "X-Wap-Initiator-URI" ":" URI
       ; URI is as defined in [RFC2396]
```

5.2.2.4 X-Wap-Push-Info

The `X-Wap-Push-Info` header is used in a push message sent by the PPG to provide the terminal with the following indications regarding each particular push transaction. It can carry the following attribute tokens:

- `authenticated`: used as the *Authenticated Flag* described in [PushOTA]. The *Initiator URI* mentioned in that section is represented by the `X-Wap-Initiator-URI`.
- `trusted`: used as the *Trusted Flag* as described in [PushOTA].
- `last`: used as the *Last Flag* as described in [PushOTA].
- `response`: indicates that a message body MAY be included in the response. The terminal MUST NOT include any message body in the response if this token is not present.

The ABNF [RFC4234] format is:

```
X-Wap-Push-Info = "X-Wap-Push-Info" ":" token *("," token)
token = ("authenticated" | "trusted" | "last" | "response")
```

Unrecognised token values MUST be ignored by the terminal.

In OTA-WSP the Push-Flag header MUST be used instead of the X-Wap-Push-Info header. Since Push messages sent via the SIP MESSAGE method do not support a message body in the response, the response token MUST NOT be sent if the SIP MESSAGE is used, and MUST be ignored by a OTA-SIP Push Client if received.

5.2.3 Header Extensions

5.2.3.1 WAP Header Extensions

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

5.2.3.2 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 the "X-Wap-" prefix.

5.2.3.3 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.

5.3 Message Body

The message body can be any MIME content type, including multipart MIME content types, and optionally encoded or transfer encoded.

5.4 Media Type

OTA-WSP and OTA-HTTP are allowing to send push message headers within the WSP/HTTP headers. In OTA-SIP this is only partially possible. In environments where necessary push message headers can't be embedded in the transport protocol the message SHOULD be encapsulated into a message/vnd.oma.push media type.

A message/vnd.oma.push object is a two-part entity, where the first part contains the message metadata and the second part is the message content. The two parts are separated from each other by a blank line.

A complete message looks something like this:

(message-metadata-headers)

(encapsulated MIME message-body)

The end of the message body is defined by the framing mechanism of the protocol used or by a Content-Length header. A Content-Type header MUST be present and MUST carry the Internet Media Type of the encapsulated MIME message-body in the field value.

The header syntax follows HTTP as defined in section 4.2 of [HTTP]. The sequence CR LF is used as the end-of-line marker.

```
CR             = <US-ASCII CR, carriage return (13)>
LF             = <US-ASCII LF, linefeed (10)>
message-header = field-name ":" [ field-value ]
field-name     = token
field-value    = *( field-content | LWS )
field-content  = <the OCTETs making up the field-value
                 and consisting of either *TEXT or combinations
                 of token, separators, and quoted-string>
```

Characters are encoded in UTF-8 and field-names SHOULD be in ASCII.

Example:

```
Date: Tue, 04 Nov 2008 11:10:24 GMT[CR][LF]
Last-Modified: Mon, 03 Nov 2008 04:58:08 GMT[CR][LF]
Content-Length: 325[CR][LF]
Content-Type: text/vnd.wap.si[CR][LF]
X-Wap-Initiator-URI: http://www.yourfriendlyserviceprovider.com/yourmail[CR][LF]
[CR][LF]
<?xml version="1.0"?>
<!DOCTYPE si PUBLIC "-//WAPFORUM//DTD SI 1.0//EN"
             "http://www.wapforum.org/DTD/si.dtd">
<si>
<indication href="http://www.xyz.com/email/123/abc.wml"
            created="2008-11-03T04:58:08Z"
            si-expires="2008-11-05T00:00:00Z">You have 4 new emails
</indication>
</si>
```

6. Proxy Rules

This section is normative.

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 5.2) and MAY delete or replace those headers as the result of message encoding, transforming, or optimisation.

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
Approved Version: OMA-TS-Push_Message-V2_2-20110809-A	09 Aug 2011	Status changed to Candidate by TP: OMA-TP-2011-0282-INP_Push_V2_2_ERP_for_Final_Approval

Appendix B. Static Conformance Requirements (Normative)

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

B.1 Terminal Features

Item	Function	Reference	Requirement
MSG-GEN-C-001-O	Generic Headers	5.2.1	
MSG-GEN-C-002-M	Content-Type header	5.2.1.10	
MSG-GEN-C-003-O	WAP Headers	5.2.2	
MSG-GEN-C-004-O	Header Extensions	5.2.3	
MSG-GEN-C-005-O	Message Body	5.3	MSG-GEN-C-006
MSG-GEN-C-006-O	Non-nested multipart content type support	5.3	
MSG-GEN-C-007-O	Nested multipart content type support	5.3	

B.2 Push Proxy Gateway Features

Item	Function	Reference	Requirement
MSG-GEN-S-001-O	Generic Headers	5.2.1	
MSG-GEN-S-002-M	Content-Type header	5.2.1.10	
MSG-GEN-S-003-O	WAP Headers	5.2.2	
MSG-GEN-S-004-O	Header Extensions	5.2.3	
MSG-GEN-S-005-O	Message Body	5.3	MSG-GEN-S-006
MSG-GEN-S-006-O	Non-nested multipart content type support	5.3	
MSG-GEN-S-007-O	Nested multipart content type support	5.3	
MSG-GEN-S-008-M	Proxy Rules	6	