Push Message

Approved Version 2.3 – 22 Nov 2011

Open Mobile Alliance

OMA-TS-Push_Message-V2_3-20111122-A

Each Open Mobile Alliance member has agreed to use reasonable endeavors to inform the Open Mobile Alliance in a timely manner of Essential IPR as it becomes aware that the Essential IPR is related to the prepared or published specification. However, the members do not have an obligation to conduct IPR searches. The declared Essential IPR is publicly available to members and non-members of the Open Mobile Alliance and may be found on the "OMA IPR Declarations" list at http://www.openmobilealliance.org/ipr.html. The Open Mobile Alliance has not conducted an independent IPR review of this document and the information contained herein, and makes no representations or warranties regarding third party IPR, including without limitation patents, copyrights or trade secret rights. This document may contain inventions for which you must obtain licenses from third parties before making, using or selling the inventions. Defined terms above are set forth in the schedule to the Open Mobile Alliance Application Form.

NO REPRESENTATIONS OR WARRANTIES (WHETHER EXPRESS OR IMPLIED) ARE MADE BY THE OPEN MOBILE ALLIANCE OR ANY OPEN MOBILE ALLIANCE MEMBER OR ITS AFFILIATES REGARDING ANY OF THE IPR'S REPRESENTED ON THE "OMA IPR DECLARATIONS" LIST, INCLUDING, BUT NOT LIMITED TO THE ACCURACY, COMPLETENESS, VALIDITY OR RELEVANCE OF THE INFORMATION OR WHETHER OR NOT SUCH RIGHTS ARE ESSENTIAL OR NON-ESSENTIAL.

THE OPEN MOBILE ALLIANCE IS NOT LIABLE FOR AND HEREBY DISCLAIMS ANY DIRECT, INDIRECT, PUNITIVE, SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR EXEMPLARY DAMAGES ARISING OUT OF OR IN CONNECTION WITH THE USE OF DOCUMENTS AND THE INFORMATION CONTAINED IN THE DOCUMENTS.

Contents

1. Scope

Following on from the development of the Wireless Application Protocol (WAP) by the WAP Forum, the Open Mobile Alliance (OMA) establishes industry wide service enabler specifications for developing applications that operate over wireless and wired communication networks. The scope for the OMA is to define a set of specifications to be used by service applications. The wireless and converged services market continues to grow very quickly, reaching new customers and providing new services. To enable network operators, manufacturers, and service providers to meet the challenges in advanced services, differentiation, and fast/flexible service creation, OMA defines a set of protocols in transport, session and application layers.

This specification defines the Push Message format, which is used by a OMA Push-enabled application to deliver the content to a Push 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

[CacheOp] “Cache Operation”, WAP-175_102-CacheOp-20010816-a, WAP Forum™. URL: http://www.wapforum.org/

[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/

[RFC2046] “Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types”, N. Freed et al, November 1996, URL: http://tools.ietf.org/rfc/rfc2046.txt

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

[RFC2387] “The MIME Multipart/Related Content-type”, E. Levinson, August 1998, The Internet Society. URL: http://www.ietf.org/rfc/rfc2387.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/

[ServiceIndication] “Service Loading”, OMA-WAP-167_ServiceInd-20010731-a, WAP Forum™. URL: http://www.wapforum.org/

[ServiceLoad] “Service Loading”, WAP-168-ServiceLoad-20010731-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 Architecture ". Open Mobile Alliance™. OMA-AD-Push-V2_3. 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	An implementation of a related set of functions that perform useful work, often enabling one or more services **[OMADICT]**
Client	A device, user agent, or other entity that acts as the receiver of a service [OMADICT]
Client Application	An Application that interacts with an OMA Push Client for reception of Push Content, including OMA service enabler user-agents (e.g. Browsers, Multimedia Messaging Clients, Instant Messaging Clients, etc), and other applications designed to utilize the OMA Push enabler.
Content	Digitized work that is processed, stored, or transmitted. It includes such things as text, presentation, audio, images, video, executable files, etc. Content may have properties such as media type, mime type, etc [OMADICT]
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 Architecture	The architecture of the entire OMA Push enabler. The Push Architecture encompasses the protocols, service interfaces, and software entities that provide the means to push data to user agents in the WAP client.
Push Client	A terminal software entity that receives Push Content from a Push Proxy Gateway, and provides the user-agent side interfaces of the OMA Push enabler, including the Push Client-Application Interface (Push-CAI).
Push Content	Content that is delivered using OMA Push,
Push Initiator	An entity or service that initiates Push content delivery to Push clients [OMADICT]
Push Message	A Push Content package that includes one or more Push Content items and related metadata (e.g. headers).
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 gateway acting as a Push proxy for Push Initiators, providing over-the-air Push message delivery services to Push clients [OMADICT]

3.3 Abbreviations

ABNF	Augmented Backus-Naur Form
HTTP	Hypertext Transfer Protocol
OMA	Open Mobile Alliance
OMNA	Open Mobile Naming Authority
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 OMA Push Architecture is described in [PushArch]. The OMA Push Architecture and specifications define how the entities of OMA Push, the Push Proxy Gateway (PPG) and Push Client, deliver Push Content as originated by Push Initiators (PI) to the targeted Push Clients, and ultimately to Client Applications.

OMA Push defines a variety of standardized content (MIME) types usable for various purposes, e.g. Service Indication [ServiceIndication], Service Loading [ServiceLoad], and Cache Operation [CacheOp]. In addition, the Push Message is defined as a generic Push Content package, which enables the delivery of any arbitrary content type that can be encapsulated for delivery as defined in this specification.

Every Push Message contains Push Content, which includes content headers and a content body. The Push Content is typically created by a Push Initiator, which packages the content into the Push Message format and sends it to the PPG via PAP. The PPG examines the message and performs any required encoding and transformation. In the process, it generally does not remove any headers or modify 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 Push-OTA [PushOTA] services.

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.

4.2 Version 2.3

This enabler release defines Push Message formatting for delivery via the OTA-PTM (Point-to-Multipoint) Push-OTA protocol variant, including packaging of multiple Push Content entities in a single Push Message.

5. Push Message Definition

This section is normative.

5.1 Message Format

A Push Message contains headers and a content body. It uses the generic Internet Text Message format [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.

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 OMA header extensions MUST have "X-Oma" prefixes and MUST be registered with OMNA [OMNA].
Note: WAP header extensions defined in earlier versions have "X-Wap-" prefixes and were 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-" or “X-Oma” 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 (such as multipart/mixed or multipart/alternative [RFC2046], or multipart/related [RFC2387]), and optionally encoded or transfer encoded.

In OTA-PTM, delivery of multiple Push Content objects in a single Push operation is possible via packaging the objects as entities in a multipart package. Note that such a collection of Push Content objects is still intended to be delivered to a single Client Application, identified by the application id header (see Section 5.2.2.1) of the Push Message.

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
OMA-TS-Push_Message-V2_3	22 Nov 2011	Status changed to Approved by TP: OMA-TP-2011-0406-INP_Push_V2_3_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	