



Push Proxy Gateway Service

Version 13-Jul-2001

Wireless Application Protocol
WAP-249-PPGService-20010713-a

A list of errata and updates to this document is available from the WAP Forum™ Web site, <http://www.wapforum.org/>, in the form of SIN documents, which are subject to revision or removal without notice.

© 1999-2001, Wireless Application Protocol Forum, Ltd. All Rights Reserved. Terms and conditions of use are available from the WAP Forum™ Web site (<http://www.wapforum.org/what/copyright.htm>).

© 1999-2001, Wireless Application Protocol Forum, Ltd. All rights reserved.

Terms and conditions of use are available from the WAP Forum™ Web site at <http://www.wapforum.org/what/copyright.htm>.

You may use this document or any part of the document for internal or educational purposes only, provided you do not modify, edit or take out of context the information in this document in any manner. You may not use this document in any other manner without the prior written permission of the WAP Forum™. The WAP Forum authorises you to copy this document, provided that you retain all copyright and other proprietary notices contained in the original materials on any copies of the materials and that you comply strictly with these terms. This copyright permission does not constitute an endorsement of the products or services offered by you.

The WAP Forum™ assumes no responsibility for errors or omissions in this document. In no event shall the WAP Forum be liable for any special, indirect or consequential damages or any damages whatsoever arising out of or in connection with the use of this information.

WAP Forum™ members have agreed to use reasonable endeavors to disclose in a timely manner to the WAP Forum the existence of all intellectual property rights (IPR's) essential to the present document. The members do not have an obligation to conduct IPR searches. This information is publicly available to members and non-members of the WAP Forum and may be found on the "WAP IPR Declarations" list at <http://www.wapforum.org/what/ipr.htm>. Essential IPR is available for license on the basis set out in the schedule to the WAP Forum Application Form.

No representations or warranties (whether express or implied) are made by the WAP Forum™ or any WAP Forum member or its affiliates regarding any of the IPR's represented on this 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.

This document is available online in PDF format at <http://www.wapforum.org/>.

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

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

Document History	
WAP-151-PPGService-19990816-a	Approved
WAP-151_100-PPGService-20000218-a	SIN
WAP-151_101-PPGService-20010422-a	SIN
WAP-249-PPGService-20010713-a	Current

Contents

1. SCOPE.....	4
2. REFERENCES	5
2.1 NORMATIVE REFERENCES	5
2.2 INFORMATIVE REFERENCES	5
3. TERMINOLOGY AND CONVENTIONS	6
3.1 CONVENTIONS	6
3.2 DEFINITIONS.....	6
3.3 ABBREVIATIONS.....	7
4. INTRODUCTION.....	8
5. PPG OPERATIONS	9
5.1 PUSH SUBMISSION PROCESSING.....	9
5.1.1 Push Submission Acceptance or Rejection.....	9
5.1.2 Over-the-Air Message Delivery	10
5.2 RESULT NOTIFICATION.....	14
5.2.1 Time of Result Notification.....	14
5.2.2 Result Notification Contents	14
5.3 PAP STATUS QUERY	14
5.4 DELIVERY CANCELLATION.....	14
6. CLIENT ADDRESSING	16
6.1 CLIENT ADDRESS FORMAT	16
6.2 CLIENT ADDRESS EXAMPLES	17
APPENDIX A. STATIC CONFORMANCE REQUIREMENTS (NORMATIVE)	18
A.1 PUSH PROXY GATEWAY FEATURES	18
A.1.1 Predicates.....	18
A.1.2 Operations	18
A.1.3 Client Addressing.....	19
APPENDIX B. CHANGE HISTORY (INFORMATIVE)	20

1. Scope

This section is informative.

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*" [WAPARCH].

One part of the WAP effort is the specification of an architecture, illustrated in figure 1, which allows content to be pushed from wired networks to WAP compliant mobile devices. The scope of this document is the specification of the Push Proxy Gateway, a gateway intended to provide push connectivity between wired and wireless networks.

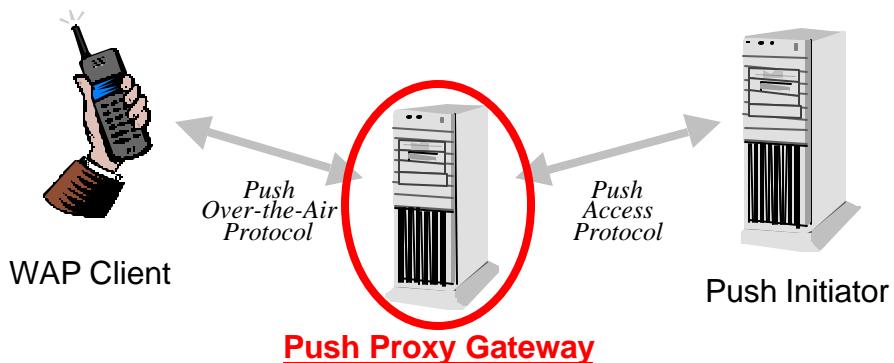


Figure 1. WAP Push Architecture

2. References

2.1 Normative References

- [CREQ] “Specification of WAP Conformance Requirements”. WAP Forum™. WAP-221-CREQ-20010425-a. URL:<http://www.wapforum.org/>
- [Mobitex] “Mobitex Interface Specification (MIS)”. Rev R4A. Document number LZY 232 105. Ericsson. URL:<http://www.ericsson.com/>
- [PushOTA] “Push OTA Protocol Specification”. WAP Forum™. WAP-235-PushOTA-20010425-a. URL: <http://www.wapforum.org/>
- [PushPAP] “Push Access Protocol Specification”. WAP Forum™. WAP-247-PAP-20010429-a. URL: <http://www.wapforum.org/>
- [PushMsg] “Push Message Specification”. WAP Forum™. WAP-251-PushMessage-20010322-a. URL: <http://www.wapforum.org/>
- [RFC791] “Internet Protocol”. J. Postel. September 1981. URL:<http://www.ietf.org/rfc/rfc791.txt>
- [RFC822] “Standard for the Format of ARPA Internet Text Messages ”. David H. Crocker. August 1982. URL: <http://www.ietf.org/rfc/rfc822.txt>
- [RFC1951] “DEFLATE Compressed Data Format Specification version 1.3”. P. Deutsch. May 1996. URL: <http://www.ietf.org/rfc/rfc1951.txt>
- [RFC2119] “Key words for use in RFCs to Indicate Requirement Levels”. S. Bradner. March 1997. URL: <http://www.ietf.org/rfc/rfc2119.txt>
- [RFC2234] “Augmented BNF for Syntax Specifications: ABNF”. D. Crocker, Ed., P. Overell. November 1997. URL: <http://www.ietf.org/rfc/rfc2234.txt>
- [RFC2373] “IP Version 6 Addressing Architecture”. R. Hinden, et al. July 1998. URL: <http://www.ietf.org/rfc/rfc2373.txt>
- [RFC2616] “Hypertext Transfer Protocol -- HTTP/1.1”. R. Fielding, et al. June 1999. URL: <http://www.ietf.org/rfc/rfc2616.txt>
- [WBXML] “Binary XML Content Format Specification”. WAP Forum™. WAP-192-WBXML-20000515-a. URL: <http://www.wapforum.org/>
- [WDP] “Wireless Datagram Protocol”. WAP Forum™. WAP-259-WDP-20010614-a. URL: <http://www.wapforum.org/>
- [WINA] “WAP Interim Naming Authority”. WAP Forum™. URL: <http://www.wapforum.org/wina/>
- [WSP] “Wireless Session Protocol”. WAP Forum™. WAP-230-WSP-20010705-a. URL: <http://www.wapforum.org/>

2.2 Informative References

- [PushArch] “Push Architectural Overview”. WAP Forum™. WAP-250-PushArchOverview-20010703-p. URL: <http://www.wapforum.org/>
- [WAPARCH] “Wireless Application Protocol Architecture Specification”. WAP Forum™. WAP-210-WAPArch-20010712-a. URL: <http://www.wapforum.org/>

3. Terminology and Conventions

This section is informative.

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.

Within this document, `courier` font is used to identify literal names of elements, attributes, parameters, and values in referenced specifications. For example, the following table indicates that the [PushPAP] message-state attribute contains the "pending" value.

PAP Attribute	Value
message-state	"pending"

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

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 address 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.

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.

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.

Registration Context - a state where the PPG is aware of at least the last capabilities and preferences conveyed from the terminal.

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.

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.

3.3 Abbreviations

ABNF	Augmented Backus-Naur Form
DTD	Document Type Definition
HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
MAN	Mobitex Subscription Number
OTA	Over The Air
OTA-HTTP	(Push) OTA over HTTP
OTA-WSP	(Push) OTA over WSP
PAP	Push Access Protocol
PI	Push Initiator
PPG	Push Proxy Gateway
QoS	Quality of Service
RFC	Request For Comments
SIR	Session Initiation Request
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
WAP	Wireless Application Protocol
WDP	Wireless Datagram Protocol
WSP	Wireless Session Protocol
WBXML	WAP Binary XML
WINA	WAP Interim Naming Authority
XML	Extensible Mark-up Language

4. Introduction

This section is informative.

This document is part of the WAP push specification suite. These specifications address the needs of a content provider seeking to "push" (i.e., send without a synchronous request) content to a client (i.e., a WAP-compliant mobile device). This is in contrast to "pull" technology, which requires a synchronous request from the client.

Push to a client is facilitated by a gateway between the wired and wireless networks. This gateway is called the Push Proxy Gateway (PPG). The purpose of this document is to specify the function of PPG.

In addition to the PPG, the WAP push architecture provides protocols to push content to the gateway and on to the client, additional functionality within WAP clients, new addressing schemes, and several standard message and content types. These are outside the scope of this document. For a complete overview, see [PushArch].

5. PPG Operations

This section defines the operations performed by a PPG. These operations include push submission processing, result notification, delivery cancellation, and Push Access Protocol (PAP) status query.

PPG operations are defined as handling each push submission (and subsequent operations related to its push message) independently from other push submissions. However, there may be limited interaction between push submissions. For example, a PPG implementation MAY support multiple delivery priorities. This could cause one message to affect the time at which another (e.g., lower priority) message is processed, and consequently the ultimate success or failure of its delivery. Note that a PPG is not required to deliver push messages in any specific order.

5.1 Push Submission Processing

A Push Initiator (PI) triggers push message processing by sending the PPG a push message. Push submission processing includes four operations. The following three operations must be performed in order:

- push submission acceptance or rejection,
- over-the-air message delivery, if the message is accepted and can be delivered in accordance with PPG policies and PI requirements; and
- message delivery result notification, if the message is accepted and the push initiator has requested message delivery notification.

The fourth operation may, as determined by the PPG implementation, be performed at any time after push message acceptance:

- PAP push message response.

These four functions are described in this section.

5.1.1 Push Submission Acceptance or Rejection

Each PAP push submission received by the PPG is either accepted or rejected.

The PPG SHOULD accept a PAP push submission if it might ultimately be delivered to the OTA client. The PPG MUST reject any push submission containing a PAP `push-message` element that is not valid with respect to its document type definition (DTD). Additional criteria used to determine whether to accept or reject a `push-message` are implementation dependent.

An accepted, undelivered PAP push submission for which message handling (described in the next section) for over-the-air delivery have not been completed MUST have the following message status reportable:

PAP Attribute	Value
<code>message-state</code>	"pending"

5.1.1.1 Replacement of a Previously Submitted Push Message

This OPTIONAL function allows replacement of a previously submitted, still pending push message.

If the PPG supports replacement, and the message is in a state from which message replacement may be assured, the PPG MUST replace the message as requested by a PAP `push-message` message [PushPAP].

A PPG that does not support the replace operation MUST reject the push submission if the PI requests replacement.

5.1.2 Over-the-Air Message Delivery

Over-the-Air message delivery consists of two functions:

- Message handling
- Over-the-air message transmission.

These functions are described in this section.

5.1.2.1 Message Handling

The PPG may transform the push message (as defined in [PushMsg]) entity contained within the push submission in preparation for over-the-air transmission. Typical reasons for transformation include compilations/optimisations for over-the-air efficiency, and translation of entities to a content type acceptable to the client. This section describes the transformations.

5.1.2.1.1 Entity and Header Transformation

A PPG MUST NOT transform the body of any entity, which falls under the scope of a No-Transform cache control directive as defined in [RFC2616]; otherwise, a PPG MAY translate entities in an implementation-dependent manner. The headers of all transformed entities MUST be revised as needed to correctly represent the transformed entity. All transformations MUST be in conformance with the requirements of [PushMsg].

5.1.2.1.1.1 WSP specific transformations

A PPG MUST support binary header encoding as specified in [WSP]. It MUST also encode content entities into their compact binary format [WBXML] (if such is specified) for transmission over OTA-WSP [PushOTA], unless it is positively known that the addressed terminal supports the non-encoded format. For example, Service Indication must usually be encoded into WBXML [WBXML] when delivered in connectionless mode.

5.1.2.1.1.2 HTTP specific transformations

A PPG SHOULD support content encoding for OTA transmission over OTA-HTTP [PushOTA] in order to minimize the volume of data sent over the air. When supported, the PPG MUST support deflate coding as specified in [RFC1951].

5.1.2.1.2 x-Wap-Application-Id (Application-ID) header processing

A PPG MUST process a [PushMsg] X-Wap-Application-Id (Application-ID) header as follows:

If the header contains a [PushMsg] absoluteURI format Application-ID for which an app-assigned-code has been registered with [WINA], the PPG MUST remove any [PushMsg] app-assigned-code format Application-ID (if present) from the header and then substitute the registered app-assigned-code format Application-ID for the absoluteURI format Application-ID.

If the header contains a [PushMsg] absoluteURI format Application-ID for which no app-assigned-code has been registered with [WINA], the PPG MUST use this value unless a [PushMsg] app-assigned-code format Application-ID is present. In this case (if the app-assigned-code format Application-ID is present), the absoluteURI format Application-ID must be removed.

A header containing only a [PushMsg] app-assigned-code format Application-ID requires no substitutions or deletions.

If the resulting header identifies a default application known to the client, the PPG MAY delete this header.

If no [PushMsg] X-Wap-Application-Id header is present in the push message, the PPG MUST, unless the client's default Application-ID is the WML user agent, add this header. If added, the Application-ID MUST be that of the WML user agent.

A PPG MAY remove any header, which specifies a default value known to the client. This default may be specified in the over-the-air protocol, provisioned, or established using an implementation-dependent mechanism. For example, an X-Wap-Application-Id header might be removed if a client has only one push application, optimising over-the-air communications. X-Wap-Application-Id headers containing a registered value MUST NOT be sent over the air without being encoded in numeric format.

5.1.2.1.3 Message State

For each push submission for which errors are encountered in the steps above, or for which it is apparent that successful message delivery is not possible, message delivery MUST NOT be attempted. Note that this may cause a PAP resultnotification-message to be sent. Messages that fail the entity and header transformation process MUST have the following status reportable:

PAP Attribute	Value
message-state	"undeliverable"
Code	"transformation-failure"

If message handling is successfully completed, an undelivered message MUST have the following status reportable:

PAP Attribute	Value
message-state	"pending"

5.1.2.2 Over-the-Air Transmission

The purpose of this function is to deliver messages to the OTA client. Key elements of this function are selection of Push OTA [PushOTA] protocol, selection of confirmed or unconfirmed push, and message delivery. A PPG implementation may include tests for message expiration and cancellation, message retransmission and delivery timeout, bearer management and WSP session (if OTA-WSP is used) or registration context (if OTA-HTTP is used) management.

5.1.2.2.1 Selection of Push OTA Protocol

A mobile terminal may support both OTA-WSP and OTA-HTTP [PushOTA]. The PPG selects the OTA protocol variant for connection-oriented push in an implementation dependent manner. A PPG could hand over the decision to the terminal by sending a *Session Initiation Request* (SIR) that contains lists of contact points for both OTA-WSP and OTA-HTTP. This approach and the SIR are defined in [PushOTA].

5.1.2.2.2 Bearer Network Selection

If the QoS section of the PAP push-message element requires a specific bearer and/or network to be used, the PPG MUST use the specified bearer and/or network, or fail to deliver the message with the following messages status reportable:

PAP Attribute	Value
message-state	"undeliverable"
Desc	An appropriate, implementation-dependent value
event-time	Time or estimated time of failure

5.1.2.2.3 Session or Registration Context Selection/Creation

The PPG may use an existing WSP session (if OTA-WSP is used) or registration context (if OTA-HTTP is used), or take implementation-dependent action(s) to create a suitable WSP session or registration context (e.g. send an OTA Session Initiation Request). If the PPG elects to attempt no further delivery action(s) due to the lack of and/or failure to create a suitable WSP session or registration context, the following messages status MUST be reportable:

PAP Attribute	Value
message-state	"undeliverable"
Desc	An appropriate, implementation-dependent value
event-time	Time or estimated time of failure

5.1.2.2.4 Delivery Time Constraints

If the PPG supports delivery time constraints, the PPG MUST NOT deliver the push message prior to the PAP *deliver-after-timestamp* time and MUST, if unable to deliver by the PAP *deliver-before-timestamp* time, fail with the following message status reportable:

PAP Attribute	Value
message-state	"expired"
Desc	An appropriate, implementation-dependent value
event-time	Time or estimated time of failure

5.1.2.2.5 Delivery

Assuming no errors, if OTA-WSP is used for OTA delivery, the PPG MUST deliver either a confirmed (Po-ConfirmedPush) or unconfirmed (Po-Push or Po-Unit-Push) [PushOTA] push primitive; if OTA-HTTP is used for OTA delivery, the PPG MUST deliver messages by using the HTTP POST method. The use of confirmed or unconfirmed push depends on the PAP *delivery-method* attribute and implementation-dependent PPG policies.

5.1.2.2.5.1 Unconfirmed Push

A PPG MUST deliver "unconfirmed" messages using OTA-WSP (Po-Push.req or Po-Unit-Push.req primitive) or OTA-HTTP. If OTA-HTTP is used, the PPG MUST report the same PAP result-notification message as if the message were pushed in an unconfirmed manner using OTA-WSP.

If the PPG sends a Po-Push.req or Po-Unit-Push.req primitive, or the PPG sends messages by using OTA-HTTP instead of these primitives, the following message status MUST be reportable:

PAP Attribute	Value
message-state	"delivered"
Delivery-method	"unconfirmed"
event-time	Time or estimated time of delivery

5.1.2.2.5.2 Confirmed Push

A PPG MUST deliver "confirmed" messages using OTA-WSP (Po-ConfirmedPush.req primitive) or OTA-HTTP. The remaining process depends on the type of push as follows:

If the PPG sends a Po-ConfirmedPush.req primitive or uses OTA-HTTP, the outcome depends as follows on whether or not the push message is acknowledged:

Success: If the PPG receives a Po-ConfirmedPush.res primitive indicating successful delivery to the OTA client, or a HTTP response including a X-Wap-Push-Status header indicating successful delivery, possibly after a PPG's implementation-dependent retries, the following message status MUST be reportable:

PAP Attribute	Value
message-state	"delivered"
Delivery-method	"confirmed"
event-time	Time or estimated time of delivery

Failure due to abort: If the PPG receives a Po-PushAbort.ind primitive indicating an aborted push attempt (OTA-WSP) or a X-Wap-Push-Status header indicating that the push message was rejected (OTA-HTTP) the following message status MUST be reportable:

PAP Attribute	Value
message-state	"aborted"
Code	PAP-specified representation of the abort parameter specified in [PushOTA]
Desc	An appropriate, implementation-dependent value
event-time	Time or estimated time of aborted delivery attempt

Failure due to timeout: If OTA-WSP is used, a timeout occurs when the PPG does not receive an OTA Po-ConfirmedPush.cnf primitive within an implementation-dependent period of time. If OTA-HTTP is used, a timeout occurs when the PPG does not receive a response to a HTTP POST request within an

implementation-dependent period of time. If the PPG elects to attempt no further delivery action(s) when a timeout occurs, the following messages status MUST be reportable:

PAP Attribute	Value
message-state	"timeout"
Desc	An appropriate, implementation-dependent value
event-time	Time or estimated time of last delivery attempt

5.2 Result Notification

The PPG MUST, if requested by the push initiator during push message submission, send a PAP resultnotification-message to the push initiator or its designee.

5.2.1 Time of Result Notification

A result notification, if requested, should be sent as soon as practical after the completion (successful or unsuccessful) of the Over-the-Air message delivery process.

5.2.2 Result Notification Contents

The PAP resultnotification-message indicates the reportable message status, which includes the message state and other information as specified earlier in this document. The status should reflect the message just before, within the limits of practicality, sending the result notification.

5.3 PAP Status Query

This OPTIONAL function provides message status on receipt of a PAP statusquery-message.

The status query reply indicates the reportable message status, which includes the message state, and other information as specified earlier in this document. The status should reflect the message just before, within the limits of practicality, sending the result notification.

5.4 Delivery Cancellation

This OPTIONAL function allows delivery cancellation of a pending push message.

If the PPG supports cancellation of a push message, and the message is in a state from which delivery cancellation may be assured, the PPG MUST cancel delivery of the message as requested by a PAP cancel-message, and the following message status MUST be reportable:

PAP Attribute	Value
message-state	"cancelled"
Desc	An appropriate, implementation-dependent value
event-time	Time or estimated time of cancellation

If the PPG cannot assure cancellation of the message delivery, it MUST reject the delivery cancellation.

Successful cancellation of a push message will trigger a delivery result notification, if requested during the push message submission.

6. Client Addressing

Push Initiators are able to identify clients to the PPG using a special textual address format. The PPG MUST transform these addresses into a form that can be used to deliver over the wireless network. Conversely, the PPG MUST transform network-specific addresses into the textual address format for communication to a Push Initiator. If a Push Initiator has used a particular address value to identify a client in a request sent to the PPG, this address value MUST be used when referring to this client in the corresponding response and any subsequent result notification.

A client address is composed of a client specifier and a PPG specifier. Inclusion of the PPG specifier provides a mechanism to ensure that the address is unambiguous, permitting requests to be routed through proxies. The PPG specifier does not necessarily identify a physical PPG, and is not required to be the hostname of the PPG receiving the address from a PI.

There are multiple types of client specifiers. A PPG MUST support at least one of these client specifier types:

- a) User-defined identifiers
- b) Device addresses

User-defined identifiers are arbitrary values that are mapped to wireless network addresses in an unspecified manner. The PPG has complete control over which bearer-level address will be used in delivering the push message to the client. The user-defined identifier MAY be expanded to several bearer-level addresses for one or more clients. In this case the PPG MUST interact with the Push Initiator in the same way as when the user-defined identifier maps to a single bearer-level address. The interpretation of user-defined identifiers is based on a mutual understanding between the Push Initiator and the PPG. This permits them to be assigned values that are useful for the application using push services. For instance, they could be e-mail addresses.

Device addresses use static values from well-known network address spaces. One example is telephone numbers in the public land mobile network (PLMN). The PPG MAY use any of the client's bearer-level addresses in delivering the push message to the client. How the PPG determines this is not specified, but may be based, for instance, on the characteristics of the bearers used by the client.

The bearer-level address may invoke a point-to-multipoint delivery in the wireless network, for example, using cell broadcast. In this case there still MUST be a single result notification, if one has been requested.

6.1 Client Address Format

The external representation of addresses processed by the PPG is defined using ABNF [RFC2234]. The format is compatible with Internet e-mail addresses [RFC822]. The PPG MUST be able to parse this address format, and it MUST be able to determine whether it supports the specified address type or not.

```
wappush-address = [ "/" ] wappush-client-address [ "/" ] "@" ppg-specifier

wappush-client-address = "WAPPUSH" "=" client-specifier
ppg-specifier = dom-fragment *( "." dom-fragment )
dom-fragment = ( ALPHA / DIGIT ) *( ALPHA / DIGIT / "-" )

client-specifier = ( user-defined-identifier / device-address )
user-defined-identifier = ( escaped-value ext-qualifiers "/TYPE=USER" )
device-address = ( global-phone-number ext-qualifiers "/TYPE=PLMN" )
/ ( ipv4 ext-qualifiers "/TYPE=IPv4" )
/ ( ipv6 ext-qualifiers "/TYPE=IPv6" )
/ ( man ext-qualifiers "/TYPE=MAN" )
/ ( escaped-value ext-qualifiers "/TYPE=" address-type )
address-type = 1*address-char
; A network bearer address type [WDP]
```

```

address-char = ( ALPHA / DIGIT / "_" )

ext-qualifiers = *( "/" keyword "=" value )
; for future extensions, e.g. special well-known user-defined identifier types
keyword = 1*( DIGIT / ALPHA / "-" )
value = 1*( %x20-2E / %x30-3C / %x3E-7E )
escaped-value = 1*( safe-char )
; the actual value escaped to use only safe characters by replacing
; any unsafe-octet with its hex-escape
safe-char = ALPHA / DIGIT / "+" / "-" / "." / "%" / "_"
unsafe-octet = %x00-2A / %x2C / %x2F / %x3A-40 / %x5B-60 / %x7B-FF
hex-escape = "%" 2HEXDIG ; value of octet as hexadecimal value

global-phone-number = "+" 1*( DIGIT / written-sep )
written-sep = ( "-" / "." )
ipv4 = 1*3DIGIT 3( "." 1*3DIGIT ) ; IPv4 address value [RFC791]
ipv6 = 4HEXDIG 7( ":" 4HEXDIG ) ; IPv6 address value [RFC2373]
man = 8DIGIT ; Mobitex MAN address format [Mobitex]

```

Each value of a `user-defined-identifier` is a sequence of arbitrary octets. They can be safely embedded in this address syntax only by escaping potentially offending values. The conversion to `escaped-value` is done by replacing each instance of `unsafe-octet` by a `hex-escape` which encodes the numeric value of the octet.

6.2 Client Address Examples

Addresses using user-defined identifiers:

```

WAPPUSH=john.doe%40wapforum.org/TYPE=USER@ppg.carrier.com
; user-defined identifier for john.doe@wapforum.org

wappush=47397547589/type=user@carrier.com
; user-defined identifier for 47397547589

WAPPUSH=47397547589/TYPE=USER@Carrier.com
; equivalent to previous one

WAPPUSH=+155519990730/TYPE=USER@ppg.carrier.com
; user-defined identifier that looks like a phone number

```

Addresses using device addresses:

```

WAPPUSH=+155519990730/TYPE=PLMN@ppg.carrier.com
; device address for a phone number of some wireless network

WAPPUSH=FEDC:BA98:7654:3210:FEDC:BA98:7654:3210/TYPE=IPv6@carrier.com
; device address for an IP v6 address

WAPPUSH=195.153.199.30/TYPE=IPv4@ppg.carrier.com
; device address for an IP v4 address

WAPPUSH=12345678/TYPE=MAN@ppg.carrier.com
; device address for a MAN address

```

Appendix A. Static Conformance Requirements (Normative)

These conformance requirements have been assembled in compliance with the WAP Specification of Conformance Requirements [CREQ].

A.1 Push Proxy Gateway Features

A.1.1 Predicates

These items are only used as predicates and do not state any requirements on the implementation.

Item	Function	Reference	Status	Requirement
PPG-CO-S-001	Confirmed push is supported		O	(OTA-CO-S-002 OR OTA-CO-S-003) AND PPG-GEN-S-013

A.1.2 Operations

Item	Function	Reference	Status	Requirement
PPG-GEN-S-001	Push Submission Rejection	5.1.1	M	
PPG-GEN-S-002	Incomplete message handling reportable	5.1.1	M	
PPG-GEN-S-003	Entity transformation under the scope of a No-Transform cache control directive	5.1.2.1.1	M	
PPG-GEN-S-004	Revising headers of transformed entities	5.1.2.1.1	M	
PPG-GEN-S-005	X-Wap-Application-Id header processing	5.1.2.1.2	M	
PPG-GEN-S-006	Registered X-Wap-Application-Id value sent over-the-air in numeric encoded format	5.1.2.1.2	M	
PPG-GEN-S-007	Reportable message states	5.1.2.1.3	M	
PPG-GEN-S-008	Bearer Network Selection (QoS)	5.1.2.2.2	M	
PPG-GEN-S-009	Reporting of failed Session or Registration Context Selection/Creation	5.1.2.2.3	M	
PPG-GEN-S-010	Delivery Time Constraints	5.1.2.2.4	M	
PPG-GEN-S-011	Delivery	5.1.2.2.5	M	
PPG-GEN-S-012	Reportable status associated with unconfirmed push	5.1.2.2.5.1	M	
PPG-GEN-S-013	Reportable statuses associated with confirmed push	5.1.2.2.5.2	O	
PPG-GEN-S-014	Sending of resultnotification-message	5.2	M	

PPG-GEN-S-015	PAP Status Query	5.3	O	
PPG-GEN-S-016	Delivery Cancellation	5.4	O	
PPG-GEN-S-017	Handling message cancellation request	5.4	M	
PPG-GEN-S-018	Support for WSP specific transformations	5.1.2.1.1.1	M	
PPG-GEN-S-019	Support for HTTP specific transformations	5.1.2.1.1.2	O	OTA-CO-S-003
PPG-GEN-S-020	Support for push message replacement	5.1.1.1	O	
PPG-GEN-S-021	Support for binary header encoding	5.1.2.1.1.1	M	
PPG-GEN-S-022	Support for content encoding using WBXML	5.1.2.1.1.1	M	
PPG-GEN-S-023	Support for content encoding using 'deflate'	5.1.2.1.1.2	O	

A.1.3 Client Addressing

Item	Function	Reference	Status	Requirement
PPG-ADD-S-001	Client Addressing	6	M	PPG-ADD-S-002 OR PPG-ADD-S-003
PPG-ADD-S-002	Support for user-defined identifiers	6	O	
PPG-ADD-S-003	Support for device addresses	6	O	
PPG-ADD-S-004	Support for client address format	6.1	M	

Appendix B. Change History

(Informative)

Type of Change	Date	Section	Description
WAP-151-PPGService-19990816-a	16-Aug-1999		The initial version of this document
WAP-151_100-PPGService-20000218-a	18-Feb-2000	7.1	The ABNF definition for global-phone-number is erroneous
Class0	6-Mar-2001	All	Support for OTA-HTTP (WAP-249-PPGService)
WAP-151_101-PPGService-20010422-a	22-Apr-2001	Appendix A	New SCR table format
Class3	22-May-2001	All	New specification template (WAP-249-PPGService)
WAP-249_100-PPGService-20010703-p	13-Jul-2001	4,5,6 Appendix A, B	Bug fixes
WAP-249_101-PPGService-20010703-p	13-Jul-2001	6.1, 6.2	Support for Mobitex addresses format