OPENWAVE®

WAP Push Technology Overview

About Openwave

Openwave Systems Inc. (Nasdaq: OPWV) is the worldwide leader of open IP-based communication infrastructure software and applications. Openwave is a global company headquartered in Redwood City, California. For more information, please visit www.openwave.com.

WAP Push Technology Overview

Table of Contents

Introduction

This paper describes the Push functionality offered by the Openwave® Push Proxy Gateway (PPG), an integral component of the Openwave® Mobile Access Gateway. Together they provide a powerful, robust platform for building Push-enabled applications.

WAP Push ushers in a new generation of application capabilities, allowing application developers to create new interactive applications and dramatically improve the usability of existing applications. Based on open standards, WAP Push introduces a new set of capabilities that enhance usability, enable secure transmission of data, allow fine control of content delivery and user interaction, control access to subscribers, provide backward compatibility with earlier Push implementations offered by Openwave and much more. These new capabilities will help to increase usage and adoption of the mobile Internet and provide more opportunities for content developers to create innovative applications.

Now, Push content is no longer restricted to a simple plain text SMS message. WAP Push and the functionality of the Openwave PPG provide a wide range of fundamental application capabilities that allow content developers to realize the potential of the mobile Internet.

The intention of the paper is to provide the reader with an overview of the concept of Push (as defined by the WAP Forum) and how that concept is manifest using the Openwave Push Proxy Gateway.

What Is Push?

Push is the delivery of content to the mobile device without previous user interaction. A Push operation is accomplished by allowing a Push Initiator (PI) to transmit Push content and delivery instructions to a Push Proxy Gateway (PPG), which then delivers the Push content to the WAP client (henceforth referred to as "device" or "terminal") according to the delivery instructions.

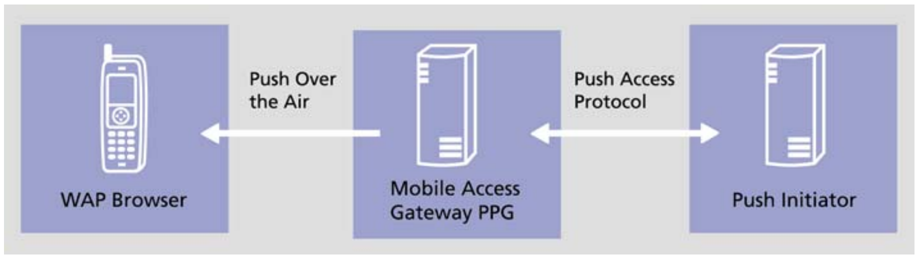


Figure 1. WAP Push Framework

The PI is typically an application that runs on an ordinary Web server. It communicates with the PPG using the Push Access Protocol (PAP) over HTTP. The PPG uses the Push Over-The-Air (OTA) Protocol to deliver the Push content to the client. Figure 1 illustrates the salient points of the framework required for Push.

The over-the-air aspect of the Push framework contains a number of bearer adaptations as specified by WAP and can be categorized into one of the following areas:

- SMS-based: Unreliable/unconfirmed Push delivery to client; limited bandwidth; the SMS bearer can be used as a delivery channel when confirmation is not critical or to initiate a data session with a device that is currently not connected.
- Circuit-switched: Reliable Push delivery; larger available bandwidth; requires triggering of circuit activation.
- Packet-switched: Efficient use of network resources; larger available bandwidth; reliable delivery; triggering of context may be required.

The standardization and implementation of Push services address a number of fundamental issues that developers and carriers face when building and deploying mobile data applications:

- Developer Interface
  The interface to the PPG is PAP transported using HTTP as defined in the WAP Specifications. The Openwave® WAP Push Library provides Java™ APIs that encapsulate the WAP 1.2.1 Push Access Protocol. Using Openwave WAP Push Library, developers can dramatically simplify the process of creating Push-enabled applications.

- User Addressing
  From a developer's perspective, the mobile device may be addressed by MSISDN or Openwave's Subscriber Identifier. Both can be harvested from requests made to a Web site or from the user directly by prompting within the application. The MSISDN is contained in the HTTP header *x-up-calling-line-id* and the Subscriber Identifier is contained in the header *x-up-subno*. Additional HTTP headers that identify the Push Proxy address are *x-up-wappush-secure* and *x-up-wappush-unsecure*.

- Client Not Online
  The client may not be online when the Push Initiator submits a message. The intermediary PPG must either defer the Push for later delivery (depending on the time stamp data supplied by the PI) or initiate connectivity with the client. Initiating connectivity is achieved using one of the transport options discussed on the previous page, such as the SMS bearer. This is achieved using a standard mechanism, the Session Initiation Request (SIR), defined within the WAP Push specification.

- Discovery of Capabilities of the Mobile Device
  The ability of the device to accept Push is either announced by the browser during session establishment or by analysis of previous interactions the device has had with the gateway. The generally accepted manner in which the device does this is by using standard HTTP [RFC2616] headers. The Push initiator may also query the client's capabilities using the WAP PAP defined CCQ query.

- Content Type Actions
  When a mobile device announces support for Push content types in its capabilities, how it behaves when receiving and processing the Push content is defined within the WAP Push specification. As a result, application developers can be assured of some level of uniform, deterministic handset behavior. These actions may be determined in two ways:

  - Standardized content: WAP Forum defines three content types that should behave in a specific way on the handset—Service Indications, Service Loads and Cache Operations.

  - The WAP Forum defines an application dispatching mechanism. In the case where there are multiple independent applications on the device, content can be targeted to a particular application by specifying the application ID in Push request. This mechanism allows WAP Push to be an extensible delivery mechanism that can be used to deliver content to existing and future applications on the mobile device.

  These standardized content types may be used by the content developer to build a Push application that has a more consistent, universal and deterministic level of behavior.

- Intruding on the User
  Since Push content is delivered without previous user interaction, the content developer must consider how the content may interrupt the user. Based on the presentation rules defined by WAP, the Push Initiator can specify different priority settings for the content that is pushed to the device. A lower priority may not interrupt the user, whereas a high priority message might make a noise and display a prompt to the user. Browser vendors have adopted these specifications so the developer is reasonably assured that there is uniform behavior for the application.

What Can Be Pushed?

This section focuses on which content can be pushed and which options require the Push Initiator to control parameters of delivery (e.g., timestamps, choice of bearer, etc.).

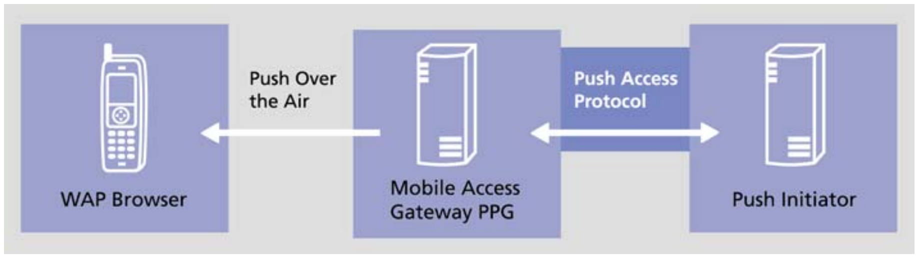
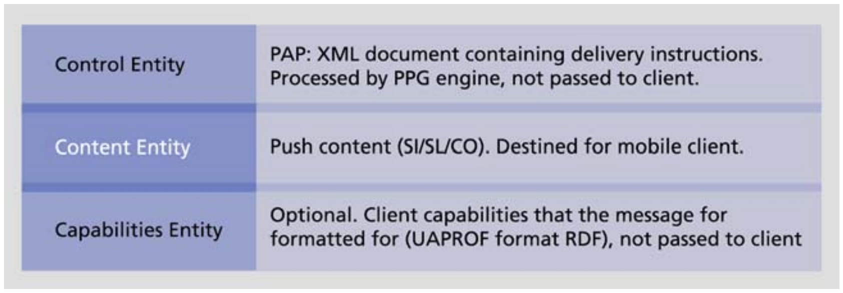


Figure 2. Push Access Protocol

The primary access point to the Push system is the PAP Push submission. The PAP Push submission is delivered as a multi-part document, a MIME content type transported over HTTP. The Push Initiator "posts" the content to the Push Proxy Gateway.



Control Entity	PAP: XML document containing delivery instructions. Processed by PPG engine, not passed to client.
Content Entity	Push content (SI/SL/CO). Destined for mobile client.
Capabilities Entity	Optional. Client capabilities that the message for formatted for (UAPROF format RDF), not passed to client

Figure 3. Push Access Protocol Multi-part Document

A PAP submission to the PPG can be broken down into the following three components:

- Control Entity
  This component of Push submission contains the basic data pertaining to the Push—i.e., who the intended recipient is, when delivery is to be attempted, when it will expire and what underlying network bearer is to be used. Additionally, a confirmation can be requested and a URL can be specified indicating where that confirmation is to be delivered after the Push is received.
- Content Entity
  The content component is the only part of the Push submission that is delivered to the handset and is described in more detail later.

- Capabilities Entity
  This component of the Push submission may be used by the Push Initiator to target particular types of mobile devices. Using WAP User Agent Profile Schema, in Resource Definition Language (RDF), the content developer may indicate which hardware vendor and browser manufacturer is intended as a recipient. If the PPG recognizes that a user does not match these capabilities, the Content Entity is not delivered. Additionally, a Push Initiator using PAP can query the capabilities associated with a user's device.

Content Entity

As mentioned, the content entity component of the PAP Push submission is the only part that is delivered to the mobile device. The following WAP standardized types of content are available:

- Service Indication
  More commonly referred to as an "alert" and similar to the implementation of an UP.Notify alert, the Service Indication presents the user with a text label and an associated URL with a choice to load the URL. The label is presented to the user, depending on user preference, or stored in the user's inbox for subsequent presentation. Identifiers in the content Service Indication offer the ability to replace duplicate content—i.e., in the case of repeated submissions, only the latest will be presented to the user.
- Service Load
  Service Loads look the same as the Service Indication; the difference is that the user may not interfere with accessing of the associated URL. The user is not presented with a choice—the URL is automatically activated.
- Cache Operation
  This content type is used to access the dynamic storage that is available on the mobile device. Previously downloaded content may be invalidated or re-validated. The URL that was used to download it identifies the content. The intention is to provide content developers initiators with an ability to have more control of their service. An example may be a frequently downloaded stock ticker or a service, which requires periodic re-crediting of a user's account.

In the case where the user defers processing of a Push, an indicator is activated on the mobile device's display highlighting that there is content available for presentation. The exact nature of this indication is dependent on the particular handset vendor, but an example of this may take the form of an "@" sign on the device.

Using the "application identifier," the Push can be targeted to a particular application running on the mobile device, providing application level addressing. By default it is assumed that the Push will be targeted at the browser.

Openwave® Multimedia Messaging Services Center and Openwave® Download Fun also use Push as a key enabling technology. More information on these products can be found on the Openwave Web site at http://www.openwave.com. Other media types may also be pushed to the device in conjunction with the WAP- defined types discussed above. An example of this is the combination of WML and Service Indication/Load highlighted later in this paper.

In addition, Openwave has made enhancements to the processing of the content to include:

- **Plain text**: May be used if the PI wishes to send a plain text message to a handset that does not support WAP Push.
- **WML**: Used in combination with WAP content types, as in the following illustration:

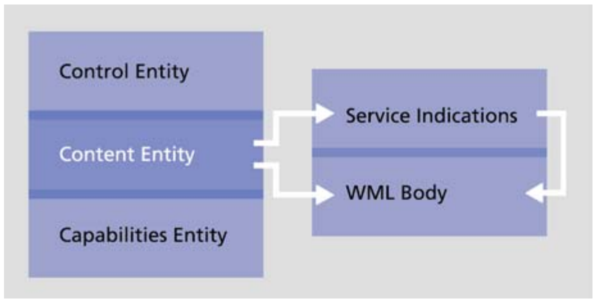


Figure 4. Processing with WML Content

The WML is composed in a multi-part component of the content entity. The WML is associated with a Service Indication or Service Load in order to retain the integrity of how the user may be interrupted. The URL of the Service Indication and Service Load references the WML body. On receipt of this content, the WML is placed in the handset cache, and the Service Indication/Load is presented to the user. The reference of the content is contained in the HTTP Header Content-Location.

On activation, the content is retrieved from the handset cache rather than establishing a session with a gateway.

As a result of these enhancements, the content being pushed can now be enriched and structured using markup, allowing great flexibility in what is presented to the user, such as a selectable menu with multiple options that are highly descriptive. This demonstrates that Push is much more than a notification system; it may be used as a robust, interactive, general content delivery vehicle that enhances existing applications and enables creation of innovative, interactive applications.

The Value of WAP Push

From a developer's perspective, there are many advantages of developing with WAP Push, such as:

- Push may be used as a channel to increase the user base for existing applications, by informing the existing user of new features/enhancements, and as a means to attract new users.
- The use of the industry standard Push Access Protocol provides the developer with the ability to use a single interface to communicate with the Openwave PPG and other Push proxies that conform to the WAP Push specification.
- The deterministic nature of the content delivery and presentation to the user allows the content developer to build a service that has a more uniform and consistent look-and-feel.

- The enhancements provided by Openwave provide the content developer with much more flexibility in the content being sent and enable them to associate capabilities with a Push submission. This flexibity allows for the developer to build interactive Push applications that provide a richer user experience.
- The translation facilities provided by the Openwave PPG enable the Push content types to be automatically translated to legacy equivalents, thus enabling a much broader, addressable user community.

Push Backward Compatibility

The Openwave PPG offers backward compatibility to the UP.Notify system of alerts, which is the pre-existing, non-WAP-based mechanism for Push that was specifically tailored for Openwave Mobile Browsers. The Openwave PPG supports pre-existing applications that use the UP.Notify protocol to send Push messages. For those applications that intend to use the Openwave WAP Push Library to create Push submissions, the PPG provides a content translation function, enabling a single Push to be targeted at both WAP Push and UP.Notify devices without extra coding by developers.

The PPG uses the user's current active data or an analysis of the user's previous interaction with the gateway to determine whether to use content translation and to determine the protocol for delivery of the Push content to the user.

At a macroscopic level:

- Service Indication maps to an alert.
- Service Load maps to a signal/pre-fetch.
- Cache Operation maps to a cache operation.

Detailed matrices of the translation of individual content parameters are available.

UP.Notify Push

This section briefly outlines the pre-existing UP.Notify Push system.

For UP.Notify, the Mobile Access Gateway platform refers to content originated in the application and sent to the user agent as a "notification." Some notifications are sent directly to the user agent ("pushed"), while others are queued in the Mobile Access Gateway to be downloaded by the user agent ("pulled") in the background during idle periods of browsing activity (e.g., reading a WML page). Hence, the Push mechanism takes advantage of both the Pull and Push channels available to provide more options to enhance the end-user experience.

An example of the Pull channel being used in the background is pre-loading the next WML page while the user is reading a current page. When the user clicks to see the next page, the new page can be display instantaneously. Although the mechanism is based on Pull technology, it is a Push in concept because the user did not request the preloaded page.

It may seem somewhat counter-intuitive for a Push, in this case, to be achieved by a Pull transaction. However, from the application's point of view, this is transparent. The application pushes a notification to the Mobile Access Gateway. It just so happens that the application-level protocol between the Mobile

Access Gateway and user-agent is based on the synchronous Pull transaction in particular connectivity scenarios. This underlying functionality is completely independent of the content developer.

SMS and Push

In terms of service, Push is sometimes compared to SMS. In fact, there are a number of significant differences. The Push Proxy Gateway may use SMS as an underlying bearer, if required by the Push Initiator. However, the Push system can add a lot value to SMS and should be considered as complementary. The following are a list of additional benefits offered by the Push system:

- Active content
  Content is active, can be used to stimulate user access and is compatible across different device vendors. Additional enhanced features, such as the ability to include WML and content replacement using HREF or unique identifiers, offer the ability to create an enhanced user experience. In this case, if the Push is unread by the user, it may be "silently replaced" by subsequent Push messages under the control of the Push Initiator.
- Push Access Protocol
  PAP is an open standard access protocol allowing the service provider to build a scalable development community with available sophisticated developer toolkits. Developers can use one interface to build Push applications that can work across different WAP-compliant Push Proxy Gateways.
- Rich Feature Set
  The WAP Push system offers a rich feature set including application level control of intrusion, application level confirmation of receipt of Push message, delivery notification to the Push Initiator, multi-recipient addressing as well as device capability discovery and utilization.
- Cache Control
  This offers the ability to affect the dynamic storage of data on the mobile device.
- Over The Air
  In terms of the air interface, the PPG offers access to a number of bearers and the ability for the initiator to specify, in terms of Quality of Service, which is the preferred mechanism. Additionally, authentication and secure Push can be used to provide highly secure Push message delivery.
- PPG Control
  The PPG offers the service provider a number of sophisticated mechanisms to tailor service for particular Push Initiators and to offer differing degrees of access to a wide user community of WAP-capable and pre-existing Openwave Mobile Browsers.

Push API

To help developers use powerful Push features, Openwave has implemented an abstraction layer that takes away most of the complexity of Push from the developer, while still exposing all the rich features of the Push standard. The Openwave WAP Push Library, as the abstraction layer is called, is illustrated below:

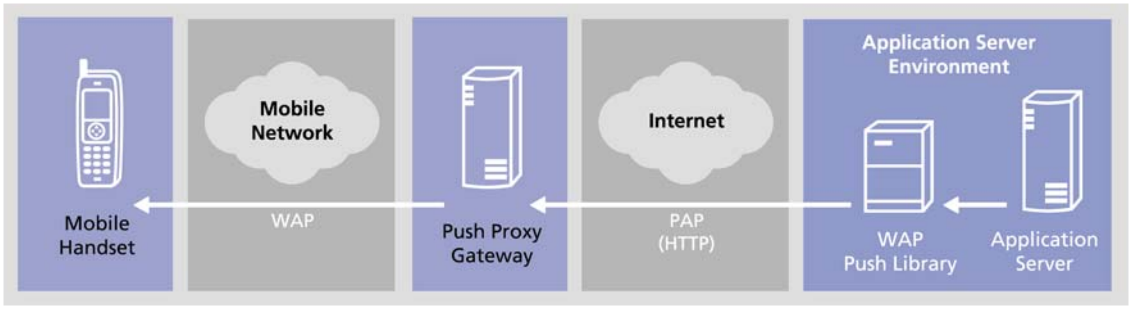


Figure 5. Openwave WAP Push Library

The Openwave WAP Push Library provides Java APIs that encapsulate the WAP 1.2.1 Push Access Protocol. With the library and tools provided, developers can more quickly build Push applications and services. The application server can use the WAP Push Library to initiate Push operations. The Push Proxy Gateway handles those operations and sends the appropriate information to the mobile device.

The Openwave WAP Push Library offers the following advantages:

- Open standard support for WAP Push
- Subscriber addressing by MSISDN as well as Subscriber Id
- Result Notification and Client Capability query support
- Secure Push support
- Multi-part Push content entity support
- Java-based API
- Convenient GUI-based tool to assist with testing and debugging
- Robust sample code and developer documentation
- Developer technical support via the Openwave Developer Program

Further information on the Openwave WAP Push Library is available from the Openwave Developer Web site at http://developer.openwave.com.

Note: It is also possible to initiate UP.Notify Push requests using the UP.Notify library provided with Openwave® SDK 4.1.

Push Service Example

Following are service examples of Push, examining how each of the content types discussed earlier may be used to provide an illustration.

Below is a simple example of a Service Indication of new mail from the mail server to the user:

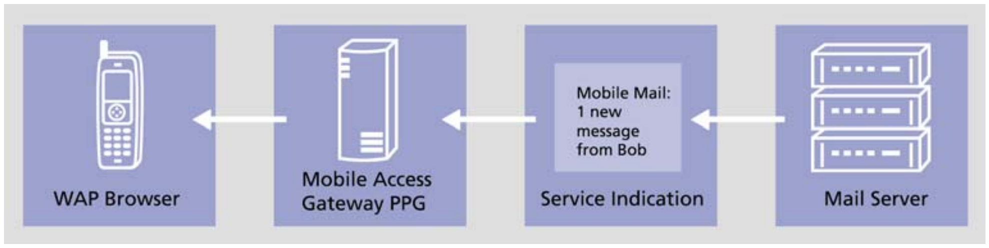

WAP Browser
Mobile Access Gateway PPG
Mobile Mail: 1 new message from Bob
Service Indication
Mail Server

Figure 6. Service Indication of New Mail

The Openwave WAP Push Developer Library may be used to easily construct and send the PAP submission. The examples to do this are contained in the Openwave WAP Push Library Developer's Guide. Note that this example reuses the Service Indication identifier, and is therefore constructed from the lower level Push library classes. Simpler examples, without reusing the Service Indication identifier are also documented in the developer guide.

The PAP Push submission, in this example, looks like this:

```
--asdlfkjiurwghasf
Content-Type: application/xml; charset=UTF-8
<?xml version="1.0"?>
      <!DOCTYPE pap PUBLIC "-//WAPFORUM//DTD PAP 1.0//EN"
            "http://www.wapforum.org/DTD/pap_1.0.dtd">

<pap product-name="PI Push Submission Service Indication)">
<push-message push-id="PushSI"      source-reference="Push Team">
<address address-
value="WAPPUSH=+18885551111/TYPE=PLMN@ppg.openwave.com"/>
</push-message>
</pap>
--asdlfkjiurwghasf
Content-type: text/vnd.wap.si; charset=UTF-8

<?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.mailservice.com/inbox?user=jane.doe"
si-id=1234>
Mobile Mail: 1 new message from Bob
</indication>
</si>
--asdlfkjiurwghasf
```

By passing the same `href` parameter (URL) with each Service Indication, a replacement function can be used on the mobile device. This is useful for presenting only the latest notification to the user when there are multiple mail notifications. If multiple service indications need to specify different URLs, a common `si-id` parameter may be passed with each Service Indication to perform the same replacement function.

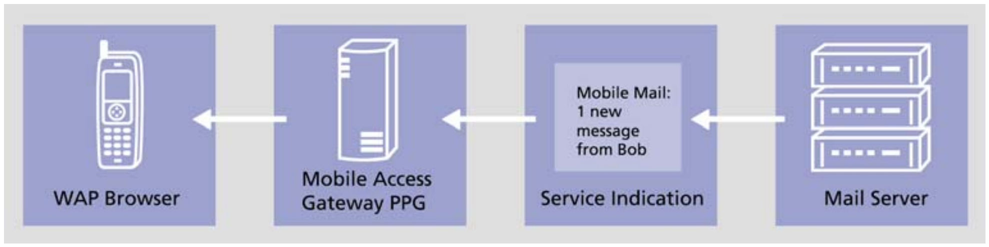

WAP Browser
Mobile Access Gateway PPG
Mobile Mail: 1 new message from Bob
Service Indication
Mail Server

Figure 7. Initial Message Notification

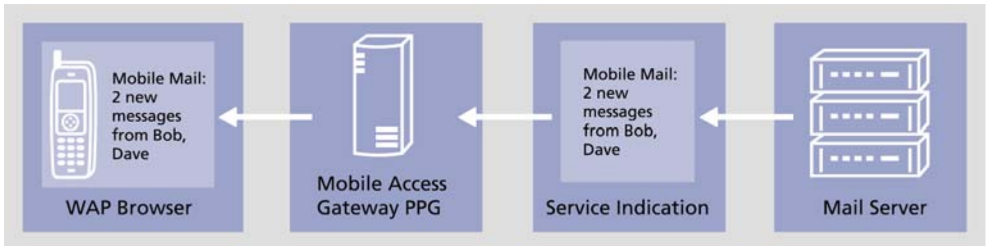

Mobile Mail: 2 new messages from Bob, Dave
WAP Browser
Mobile Access Gateway PPG
Mobile Mail: 2 new messages from Bob, Dave
Service Indication
Mail Server

Figure 8. Subsequent Message Notification

In Figure 8, using the same `href` or `si-id` parameter, the Push Initiator can ensure that only the latest unviewed pushed notification for that service is presented to the user.

Conclusion

WAP Push ushers in a new generation of application capabilities, allowing application developers to create new interactive applications and dramatically improve the usability of existing applications. Using open standards, WAP Push introduces a new set of capabilities that enhance usability, enable secure transmission of data, allow fine control of content delivery and user interaction, control access to subscribers and provide backward compatibility with earlier Push implementations offered by Openwave. These new capabilities will help to increase usage and adoption of the mobile Internet and provides more opportunities for content developers to create new, innovative applications.

Author
Fergus Wills
Product Technologist
Openwave Systems Inc.

Feedback
whitepaper.feedback@openwave.com