



| IBM Software Group

IBM WebSphere Application Server V6.1

Web Services – Notification



@business on demand.

© 2006 IBM Corporation
Converted to video May 15, 2015

This presentation will explain the support for the Web Services Notification specification in WebSphere® Application Server V6.1.

Agenda

- Overview
- Architecture / Big-Picture
- Summary and Reference



This presentation will begin with an overview of the Web Services Notification specification. It will then discuss the architecture and general concepts behind the specification and then finish with a summary.

Section

Overview



The next section will provide an overview of the Web Services Notification Specification.

What is publish/subscribe messaging?

- A one-to-many pattern of message distribution based on registration of interest
- Message oriented middleware (MOM) information is passed between components in a system in discrete segments called 'messages'
- In publish/subscribe, the following steps are followed;
 1. An application that wishes to receive data of a particular type 'subscribes' itself with the messaging provider specifying the 'topic' they are interested in
 2. An application that wishes to emit data 'publishes' to the messaging provider on the 'topic' for which the data is associated
 3. The messaging provider receives the message from the publishing application, and matches the topic against existing 'subscriptions', then makes the message available to the subscribing application



Publish/Subscribe messaging otherwise known as pub/sub, is a way of implementing a one to many message distribution pattern. Information is passed between the participants of the pattern in discrete units called messages. Applications that wish to receive messages subscribe themselves to applications that provide messages. Subscriptions are based on topic information, that allows a message provider to decide who to send messages to based on what topics other applications have subscribed to. When the messaging provider receives information from a publishing application, it can then send messages to the various subscribing applications.

Implementation of WS-Notification

- WS-Notification describes a Publish and Subscribe (Pub/Sub) messaging model for Web Services
 - ▶ Used to implement a one-to-many message distribution pattern
 - ▶ Based on the WS-Resource Framework specification
- The WS-Notification implementation within WebSphere Application Server is based upon existing technology
 - ▶ Supports the WS-Notification 1.3 specification
 - ▶ Service integration technologies artifacts to provide messaging operations
 - ▶ Service Integration Bus Web Services enablement (SIBWS) to configure Web service handling and endpoints.



The Web Services Notification specification deals with how to implement a publish and subscription messaging model with Web Services. A publish and subscription model is most often used to implement a one to many message distribution solution. The Web Services Notification support within WebSphere Application Server V6.1 is based on a number of existing technologies. The first, is the Service Integration technologies that allow for messaging operations. The second, is the implementation for Service Integration Bus Web Services that allow for Web Services to integrate with the Java messaging technologies that were introduced with WebSphere Application Server V6. By using these existing technologies, WebSphere Application Server now supports the Web Services Notification 1.3 specification.

WS-Notification (cont.)

- WS-Notification defines port types and API interfaces for applications to act as a NotificationProducer or NotificationConsumer
 - ▶ NotificationProducers are applications that want to insert messages into a system
 - ▶ NotificationConsumers are applications that want to receive messages (usually from a NotificationProducer)
- Supports 2 types of notification
 - ▶ Base notification applications must be written to manage subscriptions and receipt of publications
 - ▶ Brokered notification applications interact with a notification broker service



The Web Services Notification specification defines port types and APIs to interact with two specific types of Web Service applications, a NotificationProducer and a NotificationConsumer. A NotificationProducer application is a Web Service application that has the ability to create and insert messages into a messaging system. A NotificationConsumer application can then receive messages from a messaging system. Together these types of Web Services applications allow for a robust publish and subscription solution. The specification supports two types of notifications. Base notification applications are more complex, and must handle all aspects of working with a messaging environment. Brokered notification applications interact with a notification service provided by an application server, in order to more easily manage and interact with a messaging infrastructure.

Section

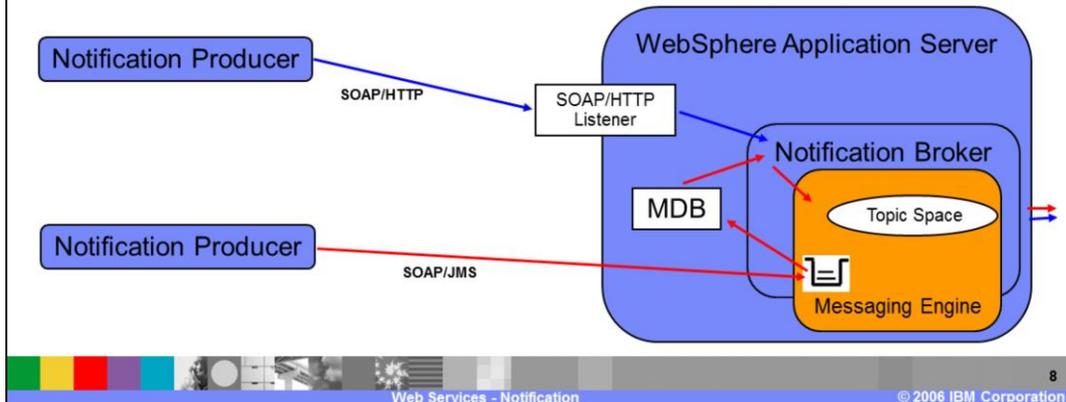
Architecture / Big-Picture



The next section will explain the architecture and general ideas behind the Web Services Notification Specification.

WS-Notification architecture

- Supports the following Web Service bindings for connection to a WS-Notification service point
 - ▶ SOAP/HTTP document/literal
 - ▶ SOAP/HTTPS document/literal
 - ▶ SOAP/JMS document/literal



The Web Services Notification specification supports a number of Web Services bindings. The choice of binding to use is a business decision based on existing Web Service infrastructure and requirements. In situations where there is no business imperative it is recommended to use the HTTP or HTTPS binding in preference to SOAP/JMS since this is more efficient solution. The potential inefficiency of the SOAP/JMS approach stems from the fact that invocation using a SOAP/JMS binding requires an extra trip through the messaging provider in order to insert the event notification into the service integration bus topic space as shown in the diagram on the slide. While this is not an unacceptable overhead it should not be encouraged in situations where there is a free choice of bindings.

Base notification applications

- Subscriber applications locate and subscribe to NotificationProducers on behalf of NotificationConsumers
 - ▶ The Subscribe operation has several parameters that allow the consuming application to indicate which type of notifications it is interested in
- The NotificationProducer application is responsible for the following tasks:
 - ▶ Accept a Subscribe operation
 - ▶ Maintain a list of active subscriptions that have been accepted
 - ▶ Generate event notification messages
 - ▶ Match generated event notifications against active subscriptions
 - ▶ Distribute event notifications to consumer applications with subscriptions that match the notification
 - ▶ Work with a SubscriptionManager service to handle the lifecycle of a Subscription



Base notification applications are required to manage all the intricacies of a complex messaging application. A base notification producer must be written to locate and issue a subscription request to available notification producer applications. A base notification producer application must handle much more. Including the ability to accept subscription requests from notification consumers, and the capability to manage a list of subscriptions. The notification producer must also generate notification messages, and match those against active subscriptions, and then be able to distribute the notification messages to the consumer applications that have subscribed to that notification. Finally the notification producer must work with a subscription manager server to manage the life cycle of all active subscriptions. These capabilities require the writing of a complex application.

Brokered notification

- WebSphere Application Server implements a new NotificationBroker service
 - ▶ A point of separation between producing applications that wish to insert event notifications into the system and consuming applications that wish to receive them
 - ▶ Provides APIs that support the functionality that must be implemented in base notification applications
- Allows developers to focus on business logic rather than implementation details
 - ▶ Such as maintaining lists of active subscribers, parsing and matching of topics, distribution of event notifications to subscribers and handling of subscription lifecycle



In a brokered notification application notification producers and consumers interact with a notification broker service provided by WebSphere Application Server V6.1. The notification broker acts as a point of separation between notification producing and consuming applications, and provides APIs to functionality that implements most of the features that must be implemented manually in a base notification application. This allows developers to focus on the more important tasks of implementing the business logic of the application, and allow the notification broker service to handle the messaging complexities.

Section

Summary and Reference

The next section is a summary of the presentation.

Summary and Reference

- Introduction to the WS-N concepts
 - ▶ Pub-sub messaging
- WS-N architecture
 - ▶ Explanation of base and brokered notification



This presentation explained the basic concepts behind the Web Services Notification specification, including the concept of a publish and subscribe messaging system. It also described the architecture and availability of a notification broker service within WebSphere Application Server V6.1. It also explained the key differences between the concept of base and brokered notification applications.

What is WS-Notification?



- WS-Notification is a group of three specifications
 - ▶ Also an introductory white paper which is subsumed into the specs in the latest revisions
- The (deprecated) whitepaper and committee homepage can be found at the following addresses
 - ▶ <http://www-106.ibm.com/developerworks/library/ws-pubsub/WS-PubSub.pdf>
 - ▶ http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsn
- The specifications are BaseNotification (which defines basic producer/consumer roles), BrokeredNotification (which extends BaseNotification to define a broker) and WS-Topics (which defines topic syntaxes and modeling)
 - ▶ http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-pr-02.pdf
 - ▶ http://docs.oasis-open.org/wsn/wsn-ws_brokered_notification-1.3-spec-pr-02.pdf
 - ▶ http://docs.oasis-open.org/wsn/wsn-ws_topics-1.3-spec-pr-01.pdf
- Also worth reading is the IBM Systems Journal article on WS-Notification
 - ▶ <http://www.research.ibm.com/journal/sj/444/niblett.html>



This slide details the specifications behind WS-N.

Trademarks, Copyrights, and Disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM	CICS	IMS	MQSeries	Tivoli
IBM (logo)	Cloudscape	Informix	OS/390	WebSphere
eflago/business	DB2	iSeries	OS/400	xSeries
ALX	DB2 Universal Database	Lotus	pSeries	zSeries

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds.

Other company, product and service names may be trademarks or service marks of others.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or program(s) described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2005,2006. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.

