This presentation will explain the new policy sets feature for WS-reliable messaging in the Feature Pack for Web Services.

Agenda

- WS-Reliable messaging
- Problem determination

This presentation will begin by explaining the policy set support for WS-reliable messaging in the Feature Pack for Web Services. This support is specific to the JAX-WS programming model.

Section

WS-Reliable messaging

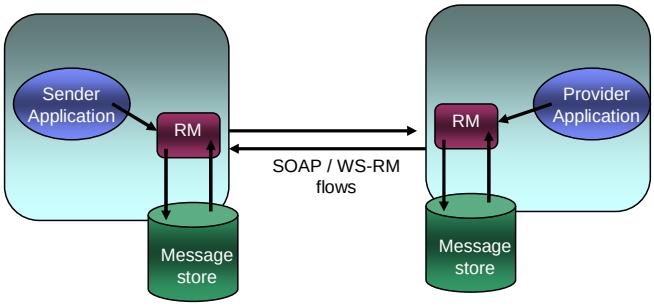
The next section discusses the Web services reliable messaging quality of service.

Functional description

- Ws-Reliable messaging is a protocol for the reliable delivery of messages between a two endpoints
- WS-RM used over SOAP/HTTP adds reliability to the transport of Web services messages
  - Traditional SOAP/HTTP is not reliable
- WS-Reliable messaging is interoperable
  - Open, published WS-Reliable messaging standard
  - Alternative Web service reliability mechanisms (for example, SOAP/JMS) are not yet interoperable

Web services reliable messaging is a specification that details how to ensure reliable delivery of messages between two endpoints. When WS-reliable messaging is used with SOAP messages sent over an HTTP transport, it can be used to ensure reliable message transport. WS-reliable messaging is also focused on interoperability, through an open, published standard, that can be adopted in other technologies.

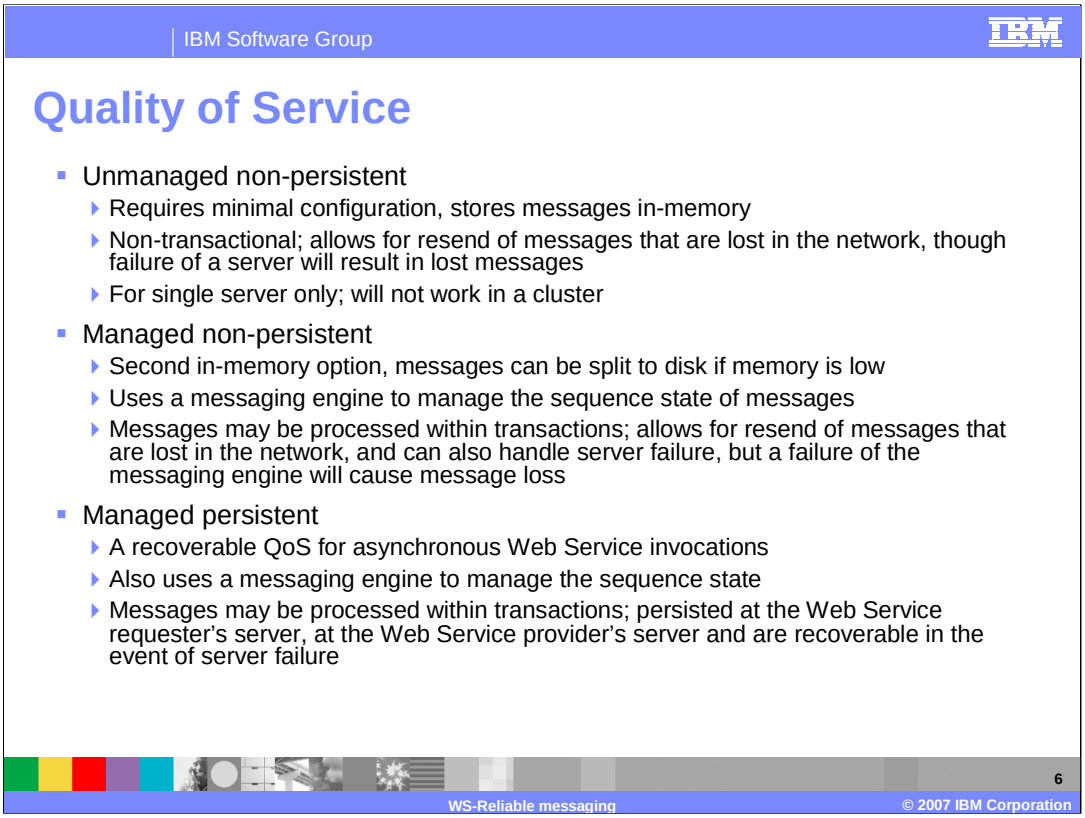
WS-RM architecture



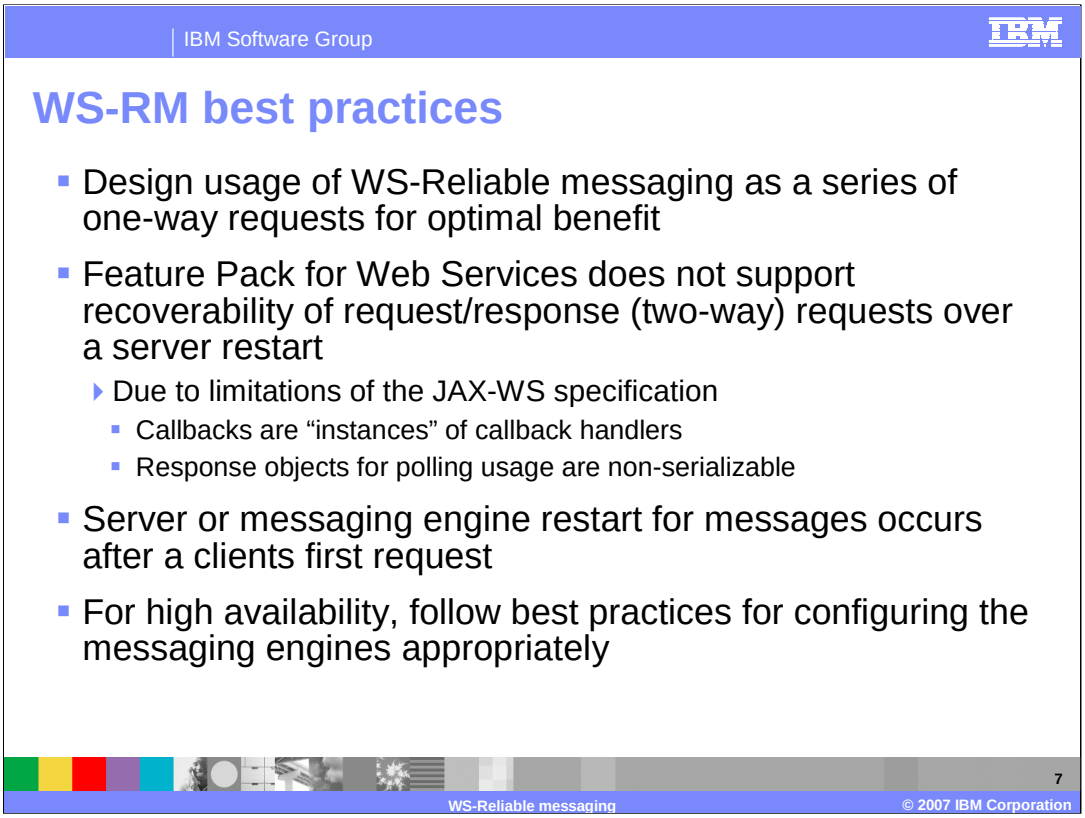
Support for
- Feb 2005 WS-RM 1.0
- OASIS WS-RX WM-RM 1.1

	Thin client	Client container	Web container	EJB container	Additional comments
Unmanaged non-persistent non-transactional and provides resend for network failure	☑	☑	☑	☑	- provider not supported in cluster or zOS - lost messages if process fails
Managed non-persistent transactional, state managed by messaging engine and protects against network loss			☑	☑	- lost messages if messaging engine fails (or stopped/restarted)
Managed persistent recoverable, transactional, state managed by messaging engine and protects against network loss, server, and messaging engine failure			☑	☑	

The architecture for WS-reliable messaging allows for SOAP messages to be placed in a message store on both the provider and sender applications. There are three levels of persistence available for WS-reliable messaging in the Feature Pack for Web Services; unmanaged non-persistent, managed non-persistent, and managed persistent. Each of these persistence levels has certain capabilities shown in the chart above, and explained further on the next slide.



The three levels of reliable messaging persistence are listed here from weakest to strongest reliability. The simplest for of reliable messaging is the unmanaged non-persistent option. This option stores messages in memory, and allows for a non-transactional resend of messages lost in transit. If the server fails, the messages will be lost. This option is only configurable for a single application server environment and will not work in a cluster. The next option is managed, non-persistent. In this option, messages are also stored in memory, though they can also be sent to disk if memory is low. This option uses a messaging engine with the enterprise service bus, to manage the sequence state of the messages. This option can handle some server failure, but the messages will be lost if the messaging engine fails. The last option, and highest level of reliability is managed persistent. This option allows for a recoverable quality of service for asynchronous Web Service calls. This also uses a messaging engine to manage the state of the messages, but in this option messages are processed within a transaction and persisted at the servers. Due to the messages being persisted, this option is recoverable in the case of as server failure.



For best results it is recommended that an application that requires reliable messaging be designed to operate as a series of one-way requests. The Feature Pack for Web Services does not support recoverability of two-way requests when a server restart occurs. This is due to limitations in the current JAX-WS specification. For messages to be resent, a further request must be issued by the client, this is required so the server or messaging engine is informed to resend messages. For an environment that will be using WS-RM and requires high availability, it is recommended to follow the best practices for configuring highly available messaging engines.

WS-RM problem determination

- Administrative console can be used to view and manage the RM layer
  - See Information center for panel documentation
- TCPMon or a network trace will give a lot of information about the messages that are being exchanged
- If that fails, try these trace strings
  - org.apache.sandesha2*=all [1]
  - com.ibm.ws.websvcs.rm*=all
  - com.ibm.ws.sib.wsrm*=all
  - Things to look for in the trace
    - 'SandeshaGlobalInHandler' traces out the headers from each inbound message
    - 'SenderWorker' traces out the headers for each outbound message
  - Notes:
    - Be sure not to use sandesha2.*=all, the '.' will make it capture less trace!

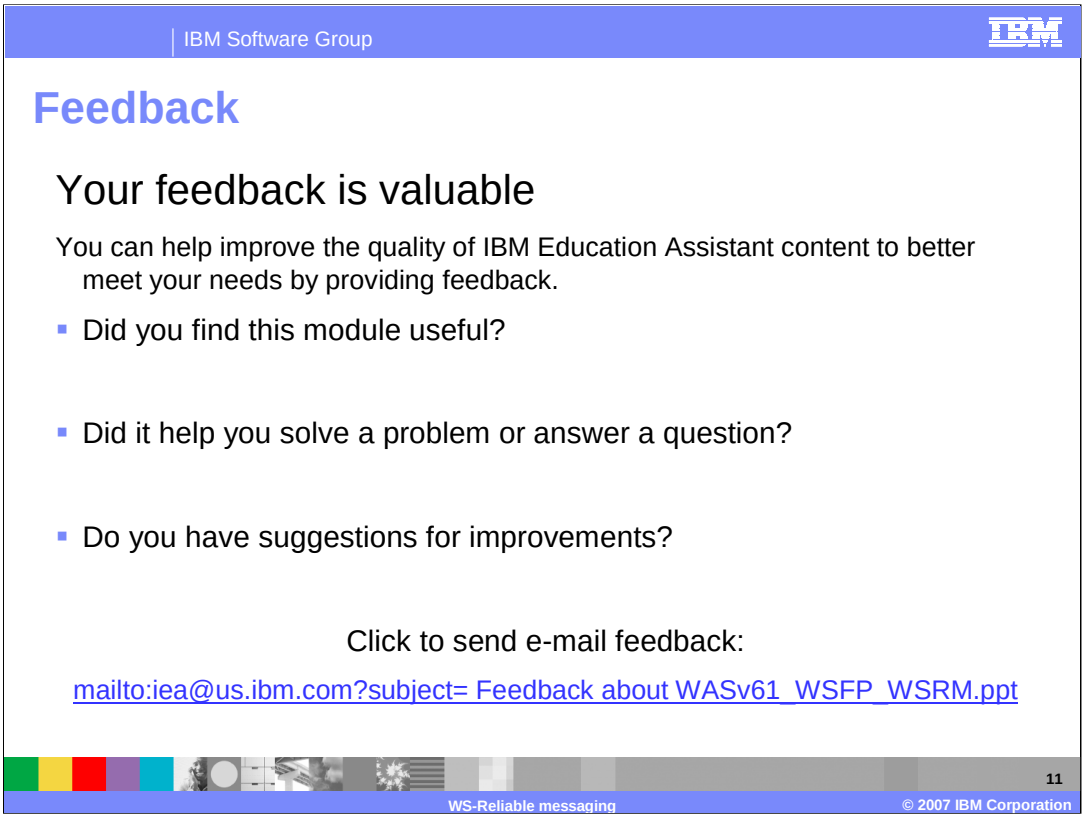
The administrative console can be used to manage the reliable messaging layer for Web services applications. Messages can be viewed and managed through various panels. Documentation can be found in the Information center for the Feature Pack for Web Services. A TCP monitor can be used to view the messages that are being exchanged over the network. Otherwise the above trace strings can be used to provide additional information specific to WS-reliable messaging problems.

The next section provides a summary of this presentation.

Summary

- The Feature Pack for Web Services provides Policy Sets for WS-Reliable Messaging
  - Specific to JAX-WS

This presentation explained the policy set support for WS-reliable messaging in the Feature Pack for Web Services. This support is specific to the JAX-WS programming model.



You can help improve the quality of IBM Education Assistant content by providing feedback.

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 WebSphere

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

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 or changes in the products or programs 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 (for example, 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.