This presentation will provide an understanding of the JAX-WS programming model, supported by the Feature Pack for Web services.

Agenda

- Java™ API for XML based Web services (JAX-WS) overview
- Architecture and details

This presentation will begin with an overview of the Java API for XML based Web services or JAX-WS. Next, the presentation will explain the architecture and details of the JAX-WS support.

Section

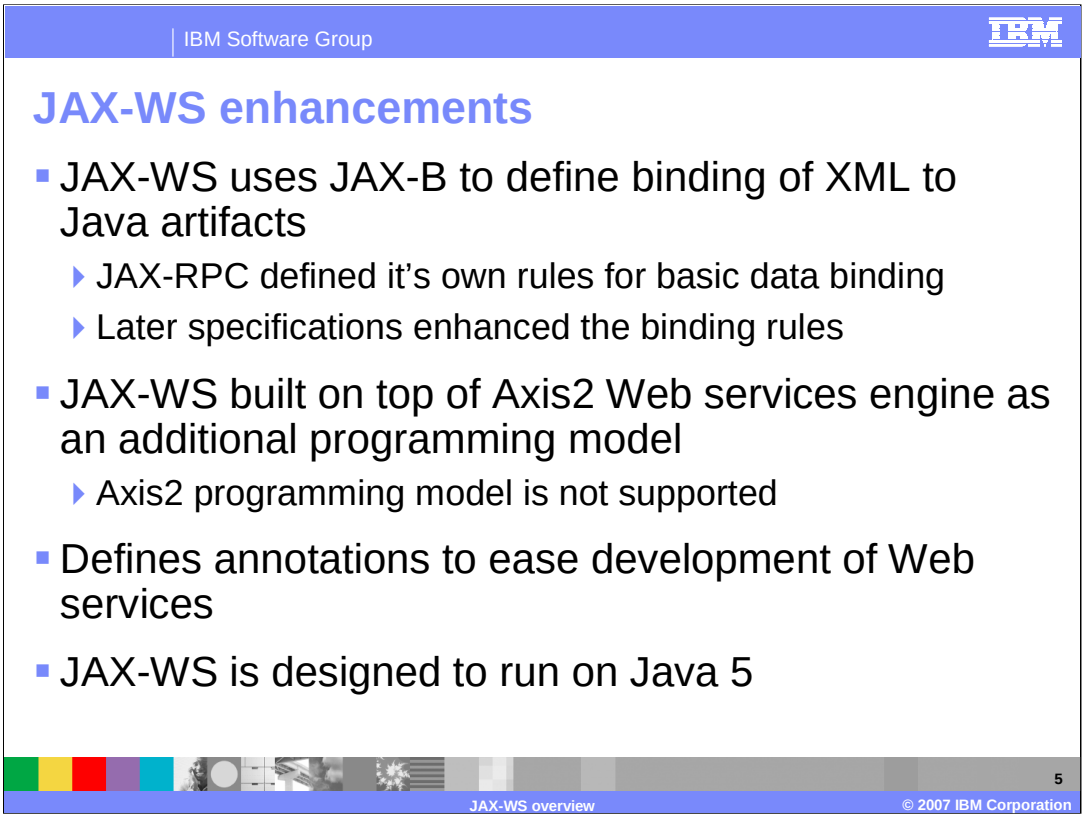
JAX-WS overview

This section will provide an overview of JAX-WS.

JAX-WS design

- JAX-WS is a follow on to the JAX-RPC 1.1 specification
  - The next evolution of JAX-RPC
  - More than JAX-RPC 2.0
- JAX-RPC defined conventions for developing remote procedure call (RPC) based Web services
  - JAX-WS deals with both RPC and message based services
- Core specification for a newly redesigned API stack for Web services
  - This “integrated stack” includes JAX-WS 2.0, JAX-B 2.0, and SAAJ 1.3

The Java API for XML Web services (JAX-WS) is a key component of a newly redesigned API stack for Web services. This "integrated stack" includes JAX-WS 2.0, JAXB 2.0, and SAAJ 1.3. JAX-WS began as an addition to the JAX-RPC 1.1 specification, but evolved further into its own specification as it came to deal with more than just remote procedure call based services. JAX-WS goes further by dealing with message based services and remote procedure call based services.



For historical reasons, in the previous Web services design there was considerable overlap of data binding functionality between JAX-RPC and JAXB APIs. This is because JAX-RPC originally included basic data binding functionality as part of its specification. When JAXB emerged after JAX-RPC, the need to separate the Web services definition and data binding components became clearer. JAX-WS is built upon the AXIS2 Web services engine, though it's important to understand that the Feature Pack for Web services does not support the AXIS 2 programming model. The Feature Pack for Web services instead provides support for its own implementation of a JAX-WS programming model. JAX-WS helps define annotations to make development of Web services easier. JAX-WS is designed to be run on Java 5 or later.

JAX-WS enhancements

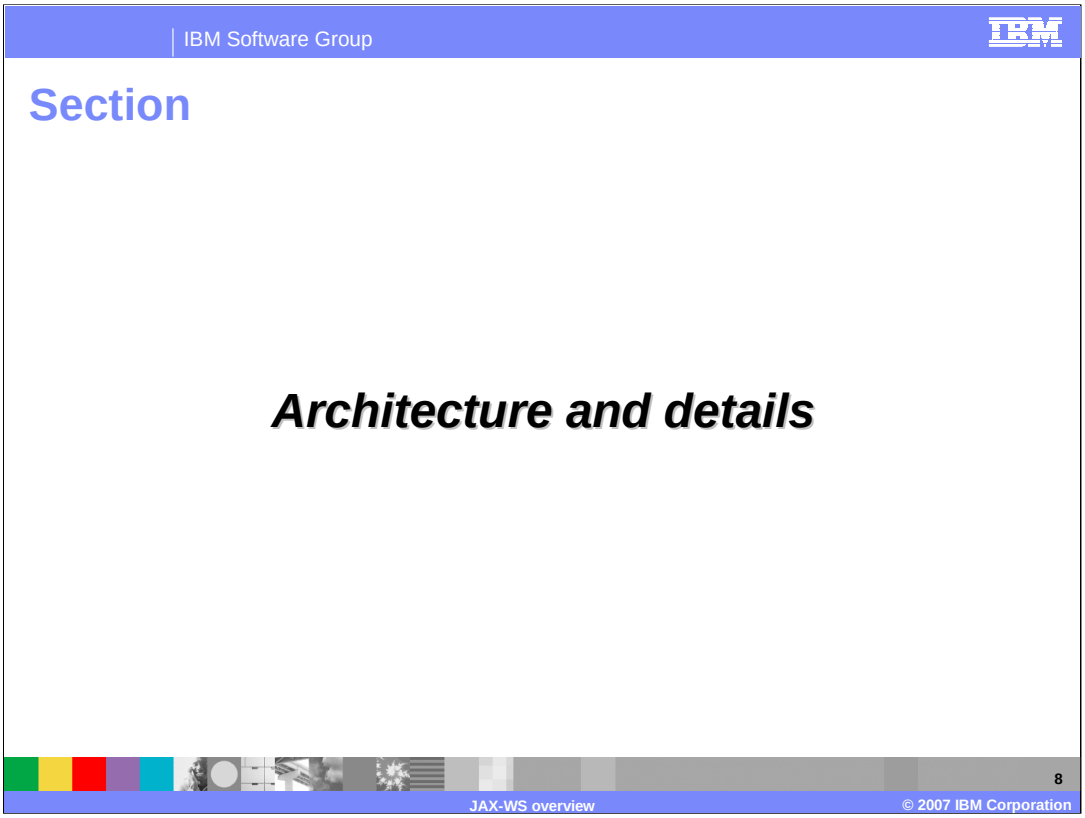
- Asynchronous support for clients
  - Using either a polling or callback mechanism
- Transport neutral
  - Support for HTTP/HTTPS in this release
- As in JAX-RPC, JAX-WS provides a mechanism for application handlers
  - Insert or retrieve data from a message as it moves through the Web services engine

JAX-WS provides other important enhancements, such as support for asynchronous polling and callback. The JAX-WS specification is transport neutral, but the Feature Pack for Web services only provides support for HTTP and HTTPS. Much like JAX-RPC, JAX-WS defines a model for creating application handlers that can deal with message data as it works through the Web services runtime.

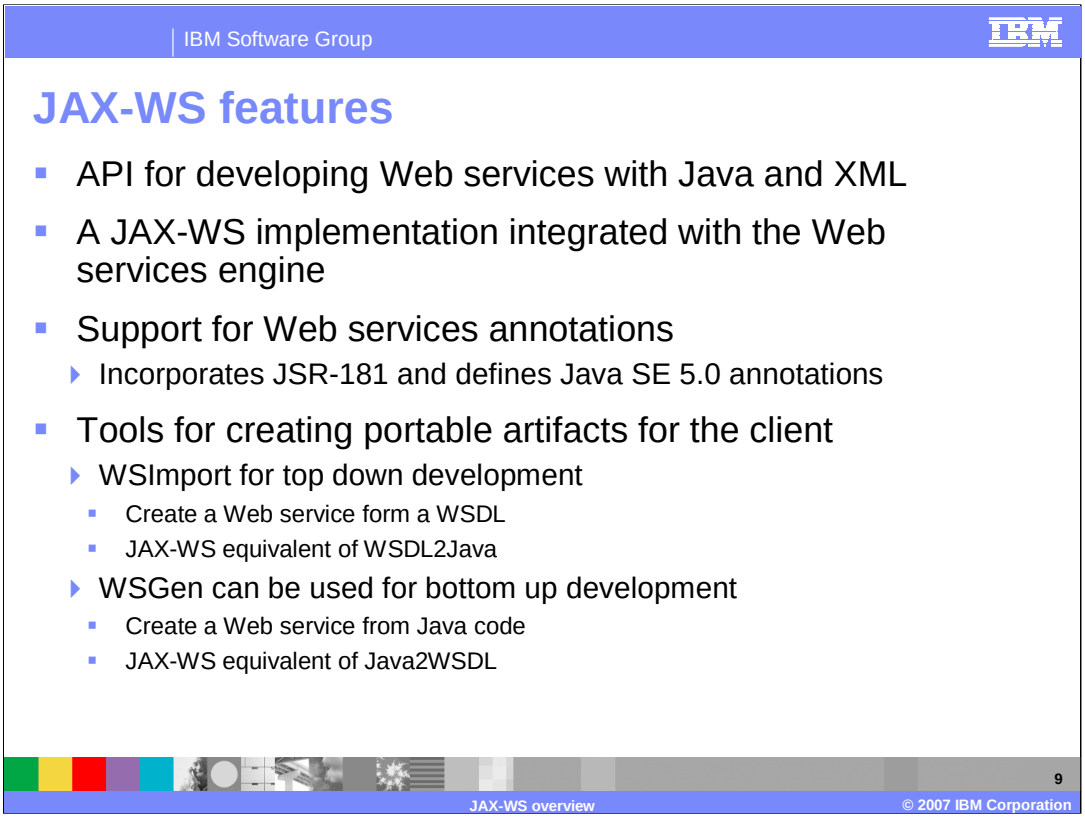
JAX-WS enhancements

- JAX-WS provides support for SOAP 1.2
  - Does not add SOAP 1.2 support with JAX-RPC
- Also allows for generic XML/HTTP as a protocol binding
  - Create clients and providers that do not use SOAP for their wire level message format
- Supports Message Transmission Optimization Mechanism (MTOM)
  - Improved method for sending binary attachments

The JAX-WS specification provides support for the SOAP 1.2 specification. It also allows for a generic XML over HTTP protocol binding, for applications that want to deal in pure XML and avoid SOAP. JAX-WS is designed to work with the Message Transmission Optimization Mechanism or MTOM. This is an improved method for sending large binary attachments with Web services.



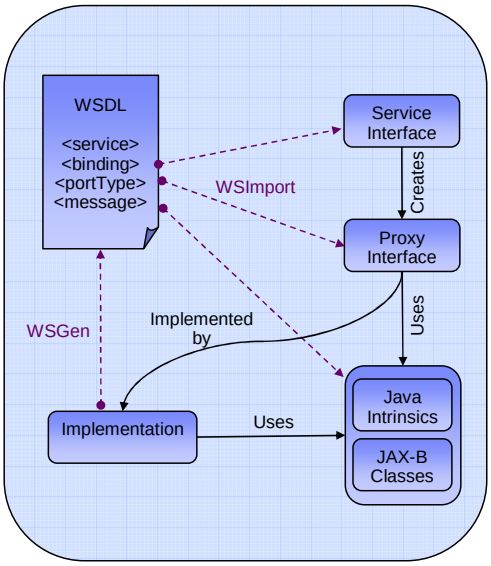
The next section discusses the architecture of the JAX-WS programming model.



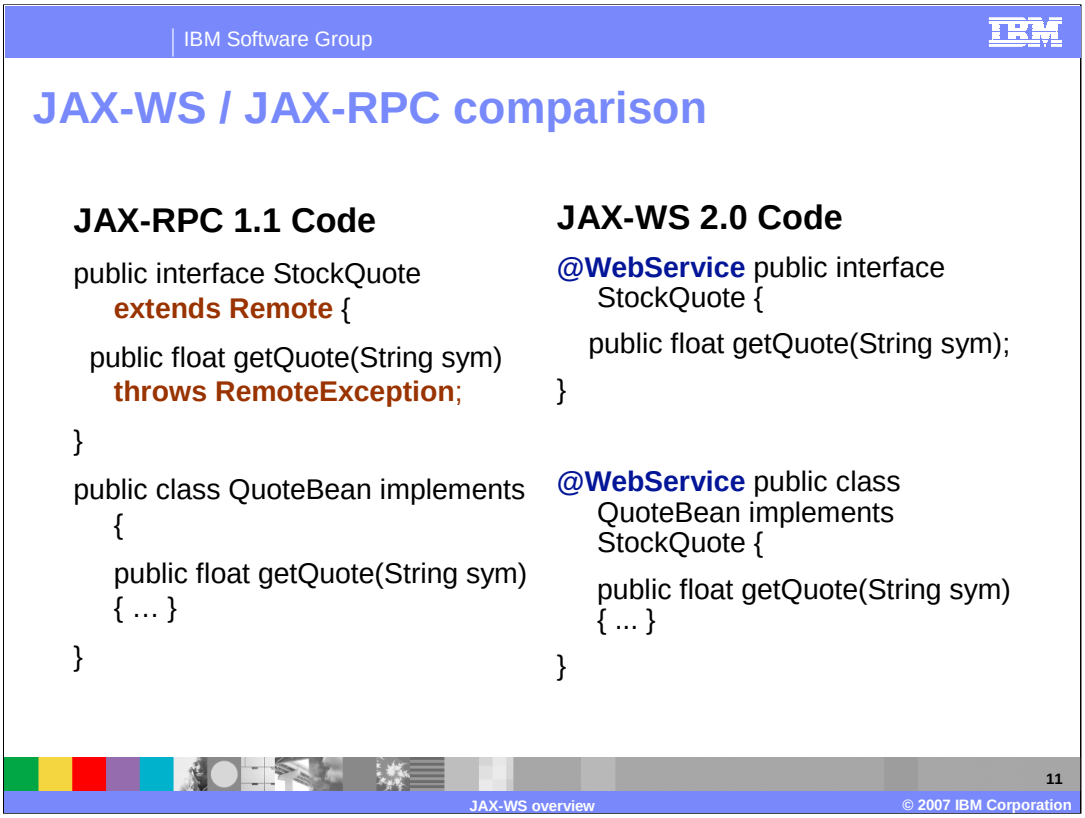
JAX-WS includes the standard implementation of the specification along with tools for building Web services, such as annotations. JAX-WS features an API for creating Web services with Java and XML. The JAX-WS implementation is integrated with the Web services engine, and support annotations based on the JAX-WS specification and JSR 181. There are also two command line tools for generating the necessary Java artifacts, WSImport for top down development from a WSDL file, and WSGen for bottom up development from Java code.

JAX-WS 2.0

- Java API for XML Web services
  - Successor to JAX-RPC 1.1
- Maps WSDL to Java
- Maps Java to WSDL
- Supports multiple data bindings
  - JAX-B 2.0 - Preferred
  - SAAJ 1.3 (SOAPMessage)
  - XML Source
  - Activation data source
- Defines WSDL customizations
- To be included in Java SE 6



The Java API for XML based Web services was designed to be a major improvement over JAX-RPC. The specification describes how to map WSDL elements to and from Java artifacts. The specification supported multiple data bindings, JAX-B 2.0 is the preferred binding provider, but SAAJ 1.3, XML source and activation data source are also options. The JAX-WS specification is planned on being included in Java SE 6.



This shows an example of the JAX-WS programming model shown side by side with JAX-RPC. Notice the removal of the remote interface, this is consistent with what is coming in Java EE 5 in the EJB support. Take note of the Web service annotation that tells you that this will be exposed as a Web service. Similar to the rest of Java EE 5, the service endpoint interface isn't required to be specified.

Following is the summary of the presentation.

Summary

- The JAX-WS specification provides enhancements and new features to Web services development
- This specification combines with JAX-B 2.0, and SAAJ 1.3 to form the new Web services “integrated stack”
  - Better integration between WS specifications
  - Easier development

The Java API for XML Web services (JAX-WS) is a key component in Web services development by providing new enhancements such as an annotation based programming model, and other new features. Combined with the JAX-B and SAAJ 1.3 specifications, it helps create an easier development experience for Web services.

The listed Web sites can be useful to understanding the development of JAX-WS.

Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

mailto:iea@us.ibm.com?subject= Feedback about WASv61_WSFP_JAX-WS_Overview.ppt

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

Java, 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.