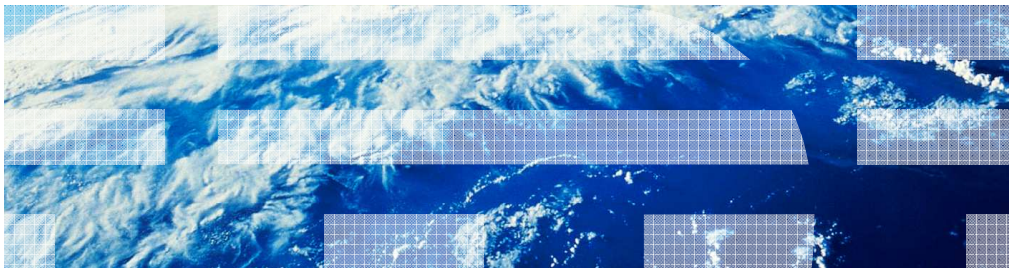


IBM WebSphere Application Server V8

JAX-WS and web services for Java EE updates



This presentation describes support for JAX-WS and web services for Java EE updates included in IBM WebSphere Application Server V8.

Table of contents

- Overview
- JAX-WS v2.2 updates
- web services for Java EE (JSR109) V1.3 updates
- Other updates – Java message service
- Custom properties policy type
- References

This presentation provides an overview of some of the web services specification updates that are supported in WebSphere Application Server V8, and information about the new custom properties policy type.

Overview

What will the updates for JAX-WS and web services for Java EE do for you?

What are the updates for JAX-WS and web services for Java EE?

- IBM WebSphere Application Server V8 contains new features for JAX-WS web services:
 - Support for updates to the Java API for XML-Based web Services (JAX-WS) (JSR 224) version 2.2 specification
 - Support for updates to the web services for Java EE (JSR 109) version 1.3 specification
 - Updates to JAX-WS JMS support based on SOAP Over Java Message Service 1.0
 - New custom properties policy type and binding

WebSphere Application Server V8 supports specification updates for JAX-WS V2.2, web Services for Java EE Version 1.3, JAX-WS JMS support based on SOAP over JMS 1.0, and a new custom properties policy type.

JAX-WS V2.2 updates

This section provides an overview of the updates to WebSphere Application Server V8 to support JAX-WS V2.2.

JAX-WS

- Java API for XML-Based web Services
 - Version 2.0 in web services Feature Pack for WebSphere Application Server V6.1
 - Version 2.1 in WebSphere Application Server V7.0
 - Version 2.2 in WebSphere Application Server V8.0
- JAX-WS is the successor to Java API for XML-Based RPC (JAX-RPC)
 - JAX-RPC is *proposed optional* in Java EE 6
 - *Proposed optional* is the specification equivalent of *deprecated*
 - JAX-RPC might not be required in future versions of the Java EE specification
- Compared with JAX-RPC, JAX-WS features include:
 - Continues to use SOAP Messages and WSDL documents
 - Annotation based programming model
 - Support for asynchronous invocations on the service requester
 - Portability of generated code
 - Java EE 6 *required technology*

JAX-WS, The Java API for XML-Based web Services, or JSR-224, has been supported in several versions of WebSphere Application Server. Version 2.0 of the JAX-WS specification was supported in the web services Feature Pack on WebSphere Application Server V6.1. Version 2.1 of the JAX-WS specification was supported in WebSphere Application Server V7. Version 2.2 of the specification is supported in WebSphere Application Server V8.

JAX-WS is the successor to the Java API for XML-Based RPC (JAX-RPC). It is important to note that JAX-RPC is marked “proposed optional” in the Java Enterprise Edition (Java EE) version 6 specification. “Proposed Optional” is the Java EE specification equivalent of deprecated, and future versions of the Java EE specification might not require JAX-RPC support.

Comparing JAX-RPC to JAX-WS, JAX-WS continues to use SOAP messages and WSDL Documents. JAX-WS also provides new features such as an annotation-based programming model, support for asynchronous invocations on service requesters, and portability of generated code. JAX-WS is also a required technology in Java EE6, and as such should be part of future Java EE specifications.

JAX-WS 2.2 updates (1 of 2)

- JAX-WS 2.2 updates from 2.1 include:
 - Client Service factory method support for web services features for MTOM, Respect Binding, and Addressing
 - Not returning a response to the requester when a WSDL-less WebServiceProvider implementation returns null
 - Allowing a service to be packaged without the JAX-B wrapper bean classes

In WebSphere Application Server V8, the JAX-WS support was updated to be compliant with V2.2 of the JAX-WS specification. These are some of the updates in JAX-WS included in V8 as part of JAX-WS V2.2. Note that some of these items are discussed in the topic “Java EE 6.0 web Services Technologies”. See that presentation for more information on those changes.

Client service factory method support for web service features is the ability to use `WebServiceFeature` instances on the `Service.create` method to configure the service.

Not returning a response to the requester when a WSDL-less `WebServiceProvider` implementation returns a null will turn such a request in to a one-way message. If the `WebServiceProvider` implementation returns a null, then the JAX-WS runtime will send back an HTTP 202 acknowledgement to the requester. It will not return a SOAP message as a response. Note that if WSDL is specified, then that WSDL determines the contract, and this behavior is not engaged.

Allowing a service to be packaged without the JAX-B wrapper bean classes means that for Document Literal Wrapped services, the JAX-B wrapper beans do not need to be packaged with the service. If they are not packaged, they are generated automatically. Any JAX-B wrapper beans that are packaged with the service will be used; any that are not packaged will be generated.

JAX-WS 2.2 updates (2 of 2)

- JAX-WS 2.2 updates from 2.1 include:
 - Only exposing public, non-static, and non-final methods in services with an implicit SEI, based on the `WebMethod` or `WebService` annotations or both
 - Supporting web service feature annotations for MTOM, RespectBinding, and Addressing on injected `WebServiceRef` port references
 - Configuring WS-Addressing through WS-Policy assertions in WSDL

Only exposing public, non-static, non-final methods in services with implicit SEIs, based on annotations, means that for a web service that does not specify a service endpoint interface, the methods that are published are based on the method declarations and annotations. Only non-static, non-final, public methods in the implementation class is considered for publishing as part of the implicit SEI. Those methods will be published according to the `@WebMethod` or `@WebService` annotation declarations.

Configuring WS-Addressing through WS-Policy assertions in WSDL allows configuring WS-Addressing based on WSDL. This is supported on both the service requester and provider. WS-Policy assertions will be generated based on the addressing annotations on a provider if WSDL is not packaged with that service. That means a service requester can use `?WSDL` to dynamically retrieve the WSDL from the provider and the WS-Policy assertions in the returned WSDL will be used to configure addressing on that requester. If the addressing requirements on the provider change, those changes will be reflected in the WSDL returned to the requester, changing the addressing configuration on the requester.

Web Services for Java EE (JSR109) V1.3 updates

This section provides an overview of the web Services for Java EE (JSR109) V1.3 Updates.

Some of the updates related to JSR 109 1.3 include

- Support for singleton session beans as JAX-WS endpoints
- Support for web beans as JAX-WS endpoints and JAX-WS application handlers
- Web services support for java:global, app, and module naming contexts
- Support for web services in EJBs within a WAR

http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/topic/com.ibm.websphere.nd.doc/info/ae/ae/cejib_ejbinwar.html

A singleton EJB is essentially a session bean with the `@Singleton` annotation or the `session-type` in the bean's deployment descriptor entry is set to "Singleton". The runtime code is updated to support this feature.

The runtime is updated to support web Beans ("JSR 299 beans") as JAX-WS service endpoint implementation classes so that the new features specified by JSR 299 (such as life cycle, event notification, dependency injection, and so on) can be used.

Version 8 supports non-local JNDI names with service-refs (like `java:global`, `java:app`, and `java:module`) so that service-refs can be shared across modules, applications and servers within a cell.

Version 8 supports EJB content in WAR modules in runtime. Basically, JAX-WS EJB-based web services can now be deployed in a war module as per the EJB 3.1 specification (JSR318 Section 20.5.2). The EJB function that is supported for beans packaged inside EJB Java archive (JAR) modules is also supported for beans packaged inside WAR modules. A bean that is packaged inside a WAR module can have the same behavior as a bean that is packaged inside an EJB JAR module. All types of EJB 3 beans are supported in WAR modules.

The rules for packaging EJB content in a WAR module are different from the rules for packaging EJB content in a JAR module. Refer to the information center link for more details

Other updates – Java Message Service

- Updates to JAX-WS stack for JMS support based on SOAP Over Java Message Service 1.0 as per

<http://dev.w3.org/cvsweb/~checkout-/2008/ws/soapjms/soapjms.html?content-type=text/html;%20charset=utf-8>

- The most recent SOAP over JMS Specification is 1.0. The JAX-WS JMS runtime is updated to make use of the new features like using TextMessage in request message.
- The main purpose is to ensure interoperability between the JMS implementations of different web services vendors

The JMS runtime is updated to support the Java message Service 1.0 as per the link. For example:

Support for TextMessage in request/response apart from BytesMessage.

Support for SOAP over JMS binding namespace value in annotation or in WSDL document.

Support for arbitrary JNDI context properties. User can specify arbitrary JNDI-related properties within the JMS endpoint URL. These properties would start with “the prefix of jndi-”.

Custom properties policy type

This section provides an overview of the new Custom properties policy type. Examples are shown for creating a Custom properties binding in the WebSphere Application Server administrative console.

Custom properties policy type

- Configure generic properties that are not supported in other policy types
- Alternative way to set some binding properties instead of setting them programmatically on the BindingProvider object
- Only supported for service clients
- Configure properties in binding
- Could use to assign an endpoint address to a service reference or service client to target a request to a particular server
 - Multiple service references for a service client can be configured with different endpoint addresses

The IBM WebSphere Application Server V8 supports a new CustomProperties policy for use in JAX-WS web services. You can use the CustomProperties policy and binding to set some generic properties that are not supported in other policy types. The additional properties are set in the binding, but the policy set that is attached to the resource must contain the Custom properties policy in order for the Custom properties binding to be used. The CustomProperties binding is only supported for service clients and is not allowed for service providers. The CustomProperties policy type provides an alternate way to set some binding properties that would otherwise be set on the `javax.xml.ws.BindingProvider` object using the JAX-WS programming model.

For example, you can configure the `BindingProvider.ENDPOINT_ADDRESS_PROPERTY` in a CustomProperties binding. By assigning a different binding with a different endpoint address to each service reference of a service client, you could target a request to a particular server based on the specified service reference.

Creating custom properties binding (1 of 2)

- Example of creating general client binding containing custom properties

[General client policy set bindings](#) > New

Use this page to create a client binding which is reusable across policy sets and applications. Use the Add button to select policy bindings and then be sure to provide configuration. Empty bindings are deleted.

* Bindings configuration name
clientBinding1

Description
Test binding

Add ▾ Delete

Custom properties
HTTP transport
JMS transport
SSL transport
WS-ReliableMessaging
WS-Security

In the WebSphere Application Server administrative console, click Add and select Custom properties in the dropdown box when you are creating a new binding or adding Custom properties to an existing binding. After you have configured the Custom properties binding, you can view or edit the properties by clicking on the Custom properties policy name in the table. You can also create and configure a Custom properties binding using wsadmin scripting commands.

Creating custom properties binding (2 of 2)

- Specify required property and value
- At least one property must be specified
- Click New to create additional property and OK when complete

[General client policy set bindings](#) > [New](#) > **Custom properties**

Use this page to define custom properties to be used with the Custom properties policy type.

Custom properties

Select	Name	Value
<input type="checkbox"/>	<input type="text" value="javax.xml.ws.service.endpoint.address"/>	<input type="text" value="http://www.ibm.com/testURL"/>

You must specify at least one property when you are creating a Custom properties binding. Enter the property name and value, and click New to specify each additional property.

Summary

- IBM WebSphere Application Server V8 contains new features for JAX-WS web services:
 - Support for updates to JAX-WS V2.2 specification
 - Support for updates to JSR 109 V1.3 specification
 - Updates to JAX-WS JMS support based on SOAP over JMS 1.0
 - New custom properties policy type and binding

WebSphere Application Server V8 supports several specification updates and a new policy type for JAX-WS web services.

References

- JSR 224: Java API for XML-Based web Services (JAX-WS)
<http://jcp.org/en/jsr/detail?id=224>
- JSR 109: Implementing Enterprise web services
<http://jcp.org/en/jsr/detail?id=109>
- SOAP Over Java Message Service 1.0
<http://dev.w3.org/cvsweb/~checkout~/2008/ws/soapjms/soapjms.html?content-type=text/html;%20charset=utf-8>
- Information Center articles for IBM WebSphere Application Server V8
 - Configuring the custom properties policy and binding using the administrative console
http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp?topic=/com.ibm.websphere.nd.iseries.doc/info/seriesnd/ae/twbs_wsspspcustomprops.html
 - Configuring the custom properties policy and binding using scripting
http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/rxml_customproperties_policybind.html

This slide contains links to useful information.

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 email feedback:

[mailto:iea@us.ibm.com?subject=Feedback about WASV8 JAXWS.ppt](mailto:iea@us.ibm.com?subject=Feedback%20about%20WASV8%20JAXWS.ppt)

This module is also available in PDF format at: [../WASV8_JAXWS.pdf](#)

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



Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, and WebSphere are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. Other company, product, or service names may be trademarks or service marks of others.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2011. All rights reserved.