JBoss WS User Guide



Version: 1.1.0.GA

1. JBossWS Runtime Overview	1
1.1. Key Features of JBossWS	1
1.2. Other relevant resources on the topic	1
2. Creating a Web Service using JBossWS runtime	2
2.1. Creating a Dynamic Web project	2
2.2. Configure JBoss Web Service facet settings	4
2.3. Creating a Web Service from a WSDL document using JBossWS runtime	7
2.4. Creating a Web service from a Java bean using JBossWS runtime 1	3
3. Creating a Web Service Client from a WSDL Document using JBoss WS 2	21
4. JBoss WS and development environment 2	25
4.1. JBossWS Preferences 2	25
4.2. Default Server and Runtime 2	29

JBossWS Runtime Overview

JBossWS is a web service framework developed as a part of the JBoss Application Server. It implements the JAX-WS specification that defines a programming model and run-time architecture for implementing web services in Java, targeted at the Java Platform, Enterprise Edition 5 (Java EE 5).

JBossWS integrates with most current JBoss Application Server releases as well as earlier ones, that did implement the J2EE 1.4 specifications. Even though JAX-RPC, the web service specification for J2EE 1.4, is still supported JBossWS does put a clear focus on JAX-WS.

1.1. Key Features of JBossWS

For a start, we propose you to look through the table of main features of JBossWS Runtime:

Feature	Benefit
JAX-RPC and JAX-WS support	JBossWS implements both the JAX-WS and JAX-RPC specifications.
EJB 2.1, EJB3 and JSE endpoints	JBossWS supports EJB 2.1, EJB3 and JSE as Web Service Endpoints.
WS-Security 1.0 for XML Encryption/Signature of the SOAP message	WS-Security standardizes authorization, encryption, and digital signature processing of web services.
JBoss AS	JBoss Application Server 5 (JavaEE 5 compliant) web service stack.
Support for MTOM/XOP and SwA-Ref	Message Transmission Optimization Mechanism (MTOM) and XML-binary Optimized Packaging (XOP) more efficiently serialize XML Infosets that have certain types of content.

Table 1.1. Key Functionality for JBossWS

1.2. Other relevant resources on the topic

You can find some extra information on:

• JBossWS Tools Wiki FAQ. [http://www.jboss.org/community/wiki/JBossWS-FAQ#Tools]

Creating a Web Service using JBossWS runtime

In this chapter we provide you with the necessary steps to create a Web Service using JBossWS runtime. First you need to create a Dynamic Web project:

2.1. Creating a Dynamic Web project

Before creating a web service, you should have a Dynamic Web Project created:

🗧 New Project 🗙	
Select a wizard —	
Create a Dynamic Web project	
<u>W</u> izards:	
type filter text	
 ▷ ➢ Java ▷ ➢ Java EE ▷ ➢ JavaScript ▷ ➢ JPA ▷ ➢ Plug-in Development ▽ ➢ Web 	
Dynamic Web Project	
a Static Web Project	
Back Next > Enish Cancel	

Figure 2.1. Dynamic Web Project

Create a Web project by selecting *New > Project... > Dynamic Web project*. Enter the following information:

- Project Name: enter a project name
- Target runtime: any server depending on your installation. If it is not listed, click New button and browse to the location where it is installed to. You may set *Target Runtime* to *None*, in this case, you should add <u>JBoss Web Service facet to the project</u>.

E New Dynamic Web Project 🗙				
Dynamic Web Project 🚬				
Create a standalone Dynamic Web project or add it to a new or existing Enterprise Application.				
Project name: test				
Project contents:				
☑ Use <u>d</u> efault				
Directory: /home/grid/workspace/documents/test Browse				
Target Runtime				
<none> \v New</none>				
Dynamic Web Module version				
2.4 🗸				
Configuration				
Default Configuration V Modify				
The default configuration provides a good starting point. Additional facets can later be installed to add new functionality to the project.				
EAR Membership				
Add project to an EAR				
EAR Project Name: EAR V New				
(?) < Back				

Figure 2.2. Dynamic Web Project Wizard

- Configuration: You may add <u>JBoss Web Service facet to the project</u> by clicking Modify... button. The opened page is like Figure 2.4.
- Configure Web Module values:

🗧 New Dynamic Web Project 🗙 🗙
Web Module
Configure web module settings.
Context Root:
test
Content Directory:
WebContent
Java Source Directory:
src
(?) Cancel

Figure 2.3. Web Module Settings Configuration

If you added the JBoss Web Service facet to the project, now the Finish button is disable. You must click Next button to set more information about the JBoss Web Service facet. The page is like Figure 2.5. Then click on the Finish button.

If you didn't add the JBoss Web Service facet to the project, click on the Finish button. Next you will need to add JBoss Web Service facet to the project.

2.2. Configure JBoss Web Service facet settings

If you have already created a new Dynamic Web project and not set the JBoss Web Service facet to the project, the next step is to add JBoss Web Service facet to the project. Right-click on the

project, select its *Properties* and then find *Project Facets* in the tree-view on the left-side of the project properties dialog. Tick on the check box for JBoss Web Services. You will see what like this:

E		Properties for zz		4 X
type filter text	Project Facets			v z)∨ v
Resource	Configuration: <custom></custom>		Save As	Delete
Builders Drools FreeMarker Context Hibernte Settings Java Build Path Java Code Style Java Editor Java EE Module Dependend Java Dependendes (Java Script JEE Module Dependencies (JSP Fragment Project Archives Project Archives Project Archives Project Archives Project Archives Project Archives Project Archives Project References Run/Debug Settings Service Policies Targeted Runtimes Task Tags Templates	Project Facet Ve		Save As_	
Web Content Settings Web Page Editor Web Project Settings				
▶ XDoclet	((
	Further configuration required		Revert	Арр(у Тэ
0			Cancel	ОК

Figure 2.4. Choose JBoss Web Service Facet

At the bottom-left of the right-side of the project properties dialog, there is a error link: *Further configuration required...*. You must click the link to set more information about JBoss Web Service facet.

Click on the Further configuration required... link. In the opened window

Chapter 2. Creating a Web Service using JBossWS runtime

Modify F	aceted Project	X
JBossWS Facet		
A JBossWS runtime has not been chosen		
 Server Supplied JBossWS Runtime 		
۲		0 New
Package all JBossWS runtime jars into the deploy	yment archive	
0	< <u>B</u> ack Next >	ок

Figure 2.5. Configure JBoss Web Service Facet

Server Supplied JBossWS Runtime: If you have already set a JBoss runtime to the project's target runtime, you may choose *Server Supplied JBossWS Runtime* and then click *Ok* to finish the configuration of JBoss Web Service facet.

If the project has no *Target Runtime* settings, you should check the second radio button and specify a JBossWS runtime from the list. You also can create a new JBossWS runtime, click on the *New...* button will bring you to another dialog to configure new JBossWS runtime.

Chapter 2. Creating a Web Service using JBossWS runtime

New JBossWS Ru	ntime 🗙
JBossWS Runtime	
Create a JBossWS Runtime	
Name:	
Version	
Home Folder:	Browse
Customize JBoss Web Service runtime jars	
O	Einish Cancel

Figure 2.6. Configure JBossWS Runtime

See how to configure a new JBossWS runtime *here*.

After setting the information about JBoss Web Service facet, for saving the result, you should click the Apply or OK button at the bottom-right of the right-side of the project properties dialog.

2.3. Creating a Web Service from a WSDL document using JBossWS runtime

In this chapter we provide you with the necessary steps to create a Web Service from a WSDL document using JBossWS runtime.

At first, please make sure that you have already created a dynamic Web project with JBoss Web Service facet installed.

See how to make it <u>here</u> and <u>here</u>.

To create a Web Service using JBossWS runtime select *File > New > Other > Web Services > Web Service* to run Web Service creation wizard.

Let's get through the wizard step-by-step:

C We	b Service ×
Web Services Select a service implementation or defin level of service and client generation.	ition and move the sliders to set the
	a bean Web Service v ct/HelloWorld.wsdl
Start service	Configuration: Server: JBoss AS 4.2 Web service runtime: JBossWS Service project: JBossWSProject Service EAR project: aEAR
Client type: Java Proxy	~
	Configuration: No client generation.
 Publish the Web service Monitor the Web service 	
⑦ < <u>B</u> ac	k Next > Einish Cancel

Figure 2.7. New Web Service Wizard

First, please select Top down Java bean Web Service from the Web Service type list, and select a WSDL document from workspace, click on the Server name link on the page will bring you to another dialog. Here you can specify the server to a JBoss Server and Web Service runtime to JBossWS runtime:

Service Deployment Configuration	×	
Choose from the list of runtimes and deployment servers, or use the default settings.		
Server-Side Deployment Selection:		
 Choose server first 		
 Choose Web service runtime first 		
 Explore options 		
Web service runtime:		
Apache Axis		
Apache Axis2		
JBossWS		
Server:		
Existing Servers		
😺 org.jboss.ide.eclipse.as.42		
Server Types		
OK Cancel	1)

Figure 2.8. Select Server and Web Service runtime

Click on the *Finish* button to see the next wizard view opened:

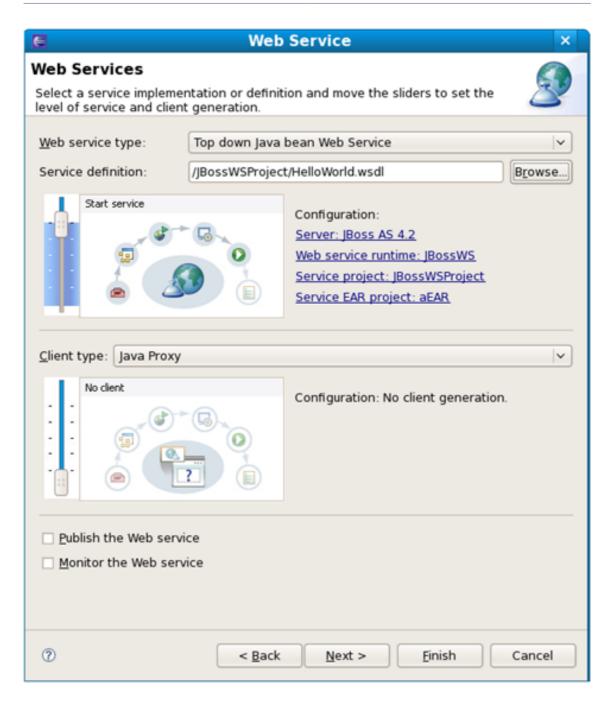


Figure 2.9. New Web Service Wizard

Click on the *Next* button to proceed:

C	Web Service	×
	e Code Generation Configuration riate option for the code generation	2
Custom package name JAX-WS specification Catalog file Binding files	org.example.www.helloworld 2.0	↓ Add Add
		Remove
☑ Generate default We	b Service Implementation classes Web.xml	
0	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish	Cancel

Figure 2.10. New Web Service Wizard

On this page, the default package name comes from the namespace of the WSDL document, you also can change it to any valid package name you want. JAX-WS specification should be set to 2.0 if your JBossWS runtime in JBoss Server is JBossWS native runtime. You can specify a catalog file and binding files if you have them. If you want the wizard to generate empty implementation classes for the Web Service, check the *Generate default Web Service implementation classes* check box. If you want to update the default Web.xml file with the Web Service servlets configured, check the *Update the default Web.xml* check box. Click on the *Next* or on the *Finish* button to generate code.

Once the Web Service code is generated, you can view the implementation class and add business logic to each method.

```
🕖 Greeterimpi.java 🕱
  package org.apache.hello_world_soap_http;
  import javax.jws.WebService;
  @WebService(name = "GreeterImpl", serviceName = "Greeter", endpointInterface =
  public class GreeterImpl implements Greeter {
       public String sayHi() {
 8
           return "";
       }
       public String greetMe(String requestType) {
    return "";
 θ
       }
       public void greetMeOneWay(String requestType) {
 8
           return;
       }
  I
       public void pingMe() {
           return;
       }
  }
                                                                                      >
```

Figure 2.11. The generated implementation Java code

View the Web.xml file:

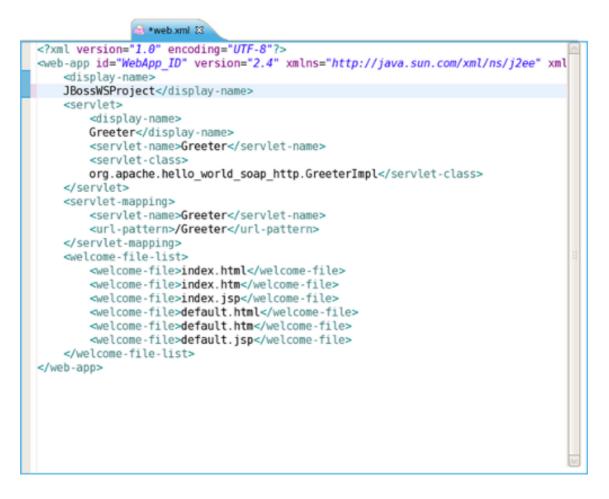


Figure 2.12. Web.xml

In the next chapter you will find out how to create a Web service from a Java bean.

2.4. Creating a Web service from a Java bean using JBossWS runtime

The Web Service wizard assists you in creating a new Web service, configuring it for deployment, and then deploying it to the server.

To create a Web service from a bean using JBoss WS:

Setup JBoss WS and development environment.

Create a Dynamic Web project.

Add JBossWS Facet to Web project.

Create a Web Service from a java bean:

• Switch to the Java EE perspective *Window > Open Perspective > Java EE*.

• In the Project Explorer view, select the bean that you created or imported into the source folder of your Web project.

		🖬 😒 Java EE
	HelloWorld.java Control	Soutline 🛪 👘 🗆
<pre>> Urest > % Deeployment Descriptor: test > % Deeployment Descriptor: test > % Java Resources: src > % org.example.www.helloworld > % HelloWorld.java > % Libraries > % build > % WebContent > % JavaScript Support > % JSR-109 Web Services</pre>	<pre>package org.example.www.helloworld; *import javax.jws.WebMethod;[] @WebService(name = "HelloWorld", targetNamespace = "http://www.example.org/HelloWorld") public class HelloWorld{ = @WebMethod(action = "http://www.example.org/HelloWorld/sayHello") @WebResult(name = "sayHelloResponse", partName = "sayHelloReguest", partName = "sayHelloRequest", partName = "sayHelloRequest", partName = "sayHelloResponse") public String sayHello(@WebParam(name = "sayHelloRequest", partName = "sayHelloResponse") public String sayHelloResponse2", partName = "sayHelloResponse2") public String sayHello2(@WebParam(name = "sayHelloRequest2", partName = "sayHelloRe return "Hello q"; } } } } </pre>	■ org.example.www ■ org.example.www 1 import declaration ■ 0 HelloWorld ■ sayHello(String ■ sayHello2(String) ■ sayHello2(String)
	🗄 Problems 🐵 Tasks 📼 Properties 🛎 Servers 🛎 🕷 Data Source Explorer 🗟 Snippets	\$ 0 Ø = 5 ⁰ ° 0
	Server State Status	
	Boss AS 4.2 at Ic-Stopped	
0*	Writable Smart Insert 26 : 1	

Figure 2.13. Select the Bean Created

• Click *File > New > Other*. Select Web Services in order to display various Web service wizards. Select the Web Service wizard. Click on the Next button.

Chapter 2. Creating a Web Service using JBossWS runtime

		New		
elect a wizard				
Create a new XML	web service.			
<u>V</u> izards:				
<u>v</u> izarus.				
👂 🗁 SQL Develo	pment			
👂 🗁 User Assist				
🕨 🗁 Web				
ᠵ 🗁 Web Service	es			
影 Ant Files				
🗊 Unit Test	UDDI			
🏄 Web Sen	vice			
🔊 Web Sen	vice Client			_
A WSDL				
👂 🗁 XML				
👂 🗁 Examples				
0	< <u>B</u> ack	Next >	Finish	Cancel

Figure 2.14. New Web Service

• On the first Web Service wizard page: select Bottom up Java bean Web service as your Web service type, and select the Java bean from which the service will be created:

Chapter 2. Creating a Web Service using JBossWS runtime

Web Service ×	
Web Services	
Select a service implementation.	
E Browse Classes	×
Web service type: Bottom up Java bean Web Service Class name: (? = any character, * = any string) 	•
Service implementation: Browse HelloWorld	
Start service Configuration: Matching items:	_
Server: JBoss AS 4.2 HelloWorld - org.example.www.helloworld	
Web service runtime: JBossWS	
Service project: test	
Service EAR project: testEAR	
Client type: Java Proxy	
No dext. Configuration: No client generation.	
	-
OK Can	ei
<u>P</u> ublish the Web service	
□ <u>M</u> onitor the Web service	

Figure 2.15. Set Web Service Common values

- Select the stages of Web service development that you want to complete using the slider:
 - Develop: this will develop the WSDL definition and implementation of the Web service. This includes such tasks as creating modules that will contain generated code, WSDL files, deployment descriptors, and Java files when appropriate.
 - Assemble: this ensures the project that will host the Web service or client gets associated to an EAR when required by the target application server.
 - Deploy: this will create the deployment code for the service.
 - Install: this will install and configure the Web module and EARs on the target server.
 - Start: this will start the server once the service has been installed on it. The serverconfig.wsdd file will be generated.
 - Test: this will provide various options for testing the service, such as using the Web Service Explorer or sample JSPs.
- Select your server: the default server is displayed. If you want to deploy your service to a different server click the link to specify a different server.
- Select your runtime: ensure the JBoss WS runtime is selected.

- Select the service project: the project selected in your workspace is displayed. To select a
 different project click on the project link. If you are deploying to JBoss Application Server
 you will also be asked to select the EAR associated with the project. Ensure that the project
 selected as the Client Web Project is different from the Service Web Project, or the service
 will be overwritten by the client's generated artifacts.
- If you want to create a client, select the type of proxy to be generated and repeat the above steps for the client. The better way is to create a web service client project separately.

Click on the Next button.

• On the JBoss Web Service Code Generation Configuration page, set the following values:

Web Service	×
JBoss Web Service Code Generation Configuration Please input the appropriate option for the code generation	2
 ✓ Generate WSDL file ✓ Update the default Web.xml 	
⊘ < <u>B</u> ack <u>N</u> ext > <u>F</u> inish Cance	2

Figure 2.16. Set Web Service values for Code Generation

- Generate WSDL file: select it, you will get a generated WSDL file in your project. But this wsdl's services' address location values are not a real address.
- After the Web service has been created, the following option can become available depending on the options you selected: Update the default web.xm file. If selected, you may test the web service by Explorer.

Click on the Next button.

• On this page, the project is deployed to the server. You can start the server and test the web service. If you want to publish the web service to a UDDI registry, you may click the Next button to publish it. If not, you may click the Finish button.

Web Service	×
Server startup	
Start the server from this page.	<u>a</u>
In order to proceed the server "JBoss AS 4.2 at localhost" must	t be st
Once the server is started the "next" button will be enabled.	
The "back" button can be used while the server is starting to change any previous settings in this wizard.	
Currently the server is stopped. Start server	

Figure 2.17. Start a Server

After the Web Service has been created, the following options may become available depending on the options selected:

- the generated web services code
- If you selected to generate a WSDL file, you will get the file in your project's wsdl folder.

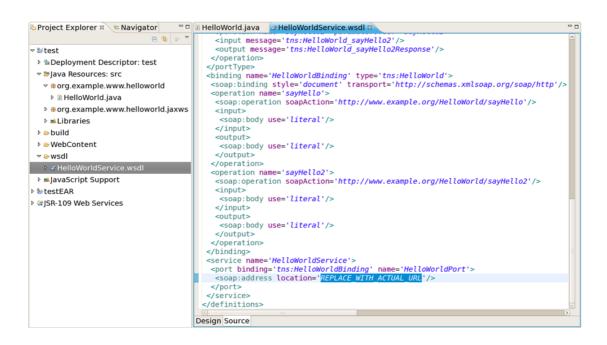


Figure 2.18. The Generated HelloWorldService.wsdl File in the wsdl Folder

• If you selected to update the default web.xml, you will test the web service in the browser. Open the Explorer, input the url for the web service according to web.xml plus ?wsdl., you will get the WSDL file from Explorer.

Chapter 2. Creating a Web Service using JBossWS runtime

Figure 2.19. The Updated web.xml file

In the next chapter you will be able to create a Web Service Client from a WSDL document using JBoss WS.

Creating a Web Service Client from a WSDL Document using JBoss WS

To create a Web Service Client from a WSDL Document using JBoss WS you need to fulfil the following steps:

Setup JBoss WS and development environment.

Create a Dynamic Web project.

Add JBossWS Facet to Web project.

Then you can create a Web Service Client from a WSDL document:

- Switch to the Java EE perspective Window > Open Perspective > Java EE.
- Click *File* > *New* > *Other*. Select Web Services in order to display the various Web service wizards. Select the Web Service Client wizard. Click on the Next button.

New X	
Select a wizard	
Access an existing XML web service	
<u>W</u> izards:	
type filter text	
h 🔁 Men 🗸	
∀	
影 Ant Files	
🗐 Unit Test UDDI	
🏄 Web Service	
🤌 Web Service Client	
AP WSDL	
Þ 🗁 XML	
Examples	
Mext > Finish Cancel	

Figure 3.1. New Web Service Client

• The first and the second Web Service Client wizard pages are the same as for <u>Web Service</u> from a WSDL document.

C Web Service	Client ×
Web Services Select a service definition.	<u>_</u>
Service definition:	Browse
<u>C</u> lient type: Java Proxy	v]
Deploy client	Configuration: Server: JBoss v4.2 Web service runtime: JBossWS Client project: client Client EAR project: testEAR
(?) < <u>B</u> ack <u>Next</u>	t > Finish Cancel

Figure 3.2. Set Web Service Common values

8	Web Service Client	×
	de Generation Configuration iate option for the code generation	5
Custom package name	org.example.www.helloworld	
JAX-WS specification	2.0	×
Catalog file		Add
Binding files		Add
		Remove
⑦ < <u>B</u>	ack Next > Finish	Cancel

Figure 3.3. Set Web Service values related to WSDL file

The only difference is:

• Client Type: Support of Java Proxy only.

Click on the Finish button.

After the Web Service Client has been created, the following may occur depending on the options you selected:

- the generated web service and client codes
- a client sample class.

	9 3 9 3 9 3 9 3 9 9 9 9 9 9 9 9 9 9 9 9	📑 😒 Java EE
🖕 Project Explorer 🗱 📄 😤 👘 🖱 🗖		Bi Outline 11 **
 ¹ Gent ¹ Substynet: Discription: clent ¹ Substynet: Clent ¹ Substynet: Discription: clent ¹ Substynet: Clent ¹ Subs	<pre>package org.example.wow.hell.world.clientsample: lapet org.example.wow.hell.world.tientsample: public clientSample { public clientSample { public clientSample { public clientSample {</pre>	

Figure 3.4. Client Sample Class

JBoss WS use a Java class to test Web Service. A client sample class will be generated, you may run this client as a java application to call a web service.

JBoss WS and development environment

In this chapter you will learn how to change JBossWS preferences and how to set default server and runtime.

4.1. JBossWS Preferences

In this section you will know how JBossWS preferences can be modified during the development process.

JBossWS preferences can be set on the JBossWS preference page. Click on *Window* > *Preferences* > *JBoss Tools* > *Web* > *JBossWS Preferences*.

On this page you can manage the JBossWS Runtime. Use the appropriate buttons to Add more runtimes or to Remove those that are not needed.

Chapter 4. JBoss WS and development environment

				Prefe	rences		
ty	pe filter text		JBo	ssWS Preferen	ces		⇔ • ⇔~ •
⊳ ⊳	JavaScript JBoss jBPM JBoss Tools JPA Plug-in Development Project Archives Report Design	<	Z	Name 1 jboss-4.2.2.GA	Versio 4.2.2	Path /home/user/Eclipse/jboss-4.2.2.G	A <u>A</u> dd <u>E</u> dit <u>B</u> emove
	Run/Debug Server Service Policies Team Validation	=					
<	Axis Emitter Axis2 Preferences JBossWS Preferences Popup Dialog Selection Project Topology Resource Management Scenario Defaults Server and Runtime	*					
0	D					ок	Cancel

Figure 4.1. JBossWS Preferences Page

Clicking on *Add* or *Edit* button will open the form where you can configure a new JBossWS runtime and change the path to JBossWS runtime home folder, modify the name and version of the existing JBossWS runtime settings. Press Finish to apply the changes.

				Preferences		×
ty	pe filter	text]	JBossWS Preferences		•
Þ	Guvnor Help		<u> </u>	Name Versio Path jboss-4.2.2.GA 2.0 /home/user/Eclipse/jboss-4.2.2.GA	Add Edit	
	HQL eq			Edit JBossWS Runtime	mov	e
Þ	Install). Java	Edit JBossWS				
Þ	JavaSo JBoss j	Input new valu	ies			
Þ	JBoss	Name:		.2.2.GA		
Þ	JPA Plug-ir Droioc	Version Home Folder:	4.2.2 /home/	\$ User/Eclipse/jboss-4.2.2.GA Browse	ə	
Þ	Projec Report	Customize	Boss \	Web Service runtime jars		
Þ	Run/De Server					
Þ	Servic Team					
	Valida Web					
~	Web S Axis	0		Einish		
		sWS Preterence				
¢	D			ок	Cancel	

Figure 4.2. Edit JBossWS Runtime

WS container allows Source and JavaDoc locations to be set via the Properties dialog on each contained .jar: right-click on any .jar file in the Project Explorer view, select *Properties*. Choose *Java Source Attachment* and select location (folder, JAR or zip) containing new source for the chosen .jar using one of the suggested options (workspace, external folder or file) or enter the path manually:

C Properties	for /home/user/jbdevstudio/jboss-eap/jboss-as/client/javassist.jar	×
type filter text	Java Source Attachment	⇔~⇔~ ▼
Java Source Attachment Javadoc Location Native Library	Select the location (folder, JAR or zip) containing the source for 'javassist.jar': Logation path:	Workspace External <u>Fi</u> le External F <u>o</u> lder
Ø	Restore Defaul	ts <u>Apply</u> Cancel

Figure 4.3. Classpath Container: Java Source Attachment

Click on Apply and then on Ok.

To change JavaDoc Location choose *Javadoc Location* and specify URL to the documentation generated by Javadoc. The Javadoc location will contain a file called *package-list*:

Chapter 4. JBoss WS and development environment

pe filter text	Javadoc Location		⇔ ≺ ⇔ ~	
Java Source Attachment		umentation generated by Javadoc. The Javad	doc location will	
Javadoc Location	contain a file called 'package-list'.			
Native Library	Iavadoc URL (e.g. 'http://www.sample-url.org/doc/' or 'file:/c:/myworkspace/myproject/doc')			
	Javadoc location path:		<u>B</u> rowse	
	 Javadoc in archive 			
	 Externa 	file O Wor <u>k</u> space file		
	Path within archive:		Browse	
			ha baaba	
		Restore <u>D</u> efau	ults <u>A</u> pply	

Figure 4.4. Classpath Container: Javadoc Location

Click on Apply and then on Ok.

4.2. Default Server and Runtime

Open *Window > Preferences > Web Services > Server and Runtime*. On this page, you can specify a default server and runtime.

For ease of use, the better way is to set runtime to JBoss WS.

After server and runtime are specified, click on the Apply button to save the values.

Chapter 4. JBoss WS and development environment

E	Prefer	ences	
type filter text	Server and Runtime		⇔ <
General	Server:	JBoss v4.2	
Þ Ant	Web service runtime:		
Data Management	web service runtime:	JEOSSWS	
▷ Help			
Install/Update			
▷ Java			
JavaScript			
JPA			
> Mylyn			
Plug-in Development			
Remote Systems			
Run/Debug			
Server			
Service Policies			
▷ Team			
Usage Data Collector Validation			
Validation Veb			
✓ Web Services			
Axis Emitter			
Axis Efficier Axis2 Preferences			
JBossWS Preferences			
Popup Dialog Selection			
Project Topology			
Resource Management			
Scenario Defaults			
Server and Runtime			
Test Facility Defaults			
Wizard Validation			
WSDL Files			
XDoclet			
▷ XML			
			Restore Defaults Apply
-			
0			OK Cancel

Figure 4.5. Specifing a default server and runtime

On the whole, this guide covers the fundamental concepts of work with tooling for JBossWS. It describes how to easily create a Web Service and a Web Service Client using JBossWS Runtime and adjust JBossWS and development environment as well.

If the information on JBossWS tools in this guide isn't enough for you, ask questions on our *forum* [http://www.jboss.com/index.html?module=bb&op=viewforum&f=201]. Your comments and suggestions are also welcome.