# Visual Web Tools Reference Guide

Version: 2.0.0.GA

Copyright © 2007 Red Hat

### **Table of Contents**

1. Visual Web Tools	1
2. Key Features of Visual Web Tools	2
3. Java Server Faces Support	5
3.1. Support for JSF Environments: JSF-RI, MyFaces, Facelets or any Custom	5
3.2. Facelets Support	6
3.3. Working with Projects	14
3.3.1. Creating a New JSF Project	14
3.3.2. Importing Existing JSF Projects with Any Structure	20
3.3.3. Adding JSF Capability to Any Existing Eclipse Project	20
3.3.4. Adding Your Own Project Templates	24
3.4. Graphical Editor and Viewing for JSF Configuration Files	25
3.4.1. Diagram	25
3.4.2. Creating New View (Page)	26
3.4.3. Tree View	
3.4.4. Source View	
3.4.5. Content Assist	
3.4.6. Error Reporting	31
3.5. Managed Beans	
3.5.1. Code Generation for Managed Beans	
3.5.2. Add Existing Java Beans to a JSF Configuration File	
3.6. Create and Register a Custom Converter	41
3.7. Create and Register a Custom Validator	46
3.8. Create and Register Referenced Beans	51
4. Struts	57
4.1. Support for Struts 1.1, 1.2.x	57
4.2. Working with Projects	58
4.2.1. Creating a New Struts Project	58
4.2.2. Importing an Existing Struts Project with Any Structure	63
4.2.3. Adding Struts Capability to an Existing Web Application	63
4.3. Graphical Editor for Struts Configuration Files	71
4.3.1. Diagram Mode	71
4.3.2. Tree Mode	74
4.3.3. Source Mode	75
4.4. Graphical Editor for Tiles Files	78
4.4.1. Create New Tiles File	78
4.4.2. Tree View	79
4.4.3. Diagram View	81
4.4.4. Source	
4.5. Graphical Editor for Struts Validation Files	86
4.6. Support for Multiple Struts Modules	93
4.6.1. Struts Modules	93
4.6.2. When Importing a Struts Project	93
4.6.3. Editing Modules in an Existing Project	94
4.6.4. Adding New Modules	95

4.7. Code Generation for Action, FormBean, Forward and Exception Classes	97		
4.8. Struts Configuration File Debugger			
4.9. Customizable Page Links Recognizer	101		
5. JBoss Tools Palette	103		
5.1. Using the Palette	104		
5.1.1. JBoss Tools Palette	104		
5.1.2. Inserting Tags into a JSP File	105		
5.1.3. Adding Custom JSF Tags to the JBoss Tools Palette	107		
5.1.4. Drag-and-Drop	108		
5.1.5. Import Button	110		
5.2. Palette Options	111		
5.2.1. Palette Editor	112		
5.2.2. Show/Hide	118		
5.2.3. Import	120		
5.3. Rich Faces Support	120		
6. Web Projects View	122		
6.1. Web Projects View	122		
6.2. Project Organization	122		
6.3. Drag and Drop	123		
6.3.1. For a Property			
6.3.2. For Managed Bean Attributes			
6.3.3. Navigation Rules	126		
6.3.4. For a Tag Library File Declaration			
6.3.5. For JSP Pages			
6.4. Developing the Application	129		
6.5. Expanding Tag Library Files			
6.6. Drag and Drop Tag Libraries on to JBoss Tools Palette	131		
6.7. Create and Import ISF and Struts Projects			
7 Verification and Validation	133		
7.1. JBoss Developer Studio Verification			
7.1.1 ISF Project Verification	133		
7.1.2. Struts Project Verification	137		
8 Editors	141		
8.1 Editors Features	141		
8.1.1 OpenOn	141		
8 1 1 1 XMI Files	141		
8 1 1 2 ISP Pages	143		
8.1.2. Code Assist and Dynamic Code Assist (based on project data)	144		
8 1 2 1 Content Assist Features	145		
8 1 2 1 1 Content Assist	145		
8 1 2 1 2 ISE Project Files	145		
8.1.2.1.2. Jor Project Thes	145		
812122 Content Assist Rased on Project Data	148		
8 1 2 1 2 3 Content Assist within Tree ISE Editor	150		
8 1 2 1 3 Struts Project Files	155		
8 1 2 1 3 1 Content Assist for Struts Configuration File	155		
8.1.2.1.3.2. Content Assist for Struts ISP File			
8.1.2.1.4. JSP Pages			
8.1.2.1.4.1. Content Assist for ISF Tags			

8.1.2.1.4.2. Content Assist for JSTL Tags	158
8.1.2.1.4.3. Content Assist for HTML Tags	158
8.1.2.1.4.4. Content Assist for JavaScript Tags	159
8.1.2.1.4.5. Content Assist within JSF Configuration Editor	159
8.1.2.1.5. Content Assist for Rich Faces components	160
8.1.2.2. Adding dynamic code assist to custom components that were added to JBoss T	ſools
Palette	163
8.1.3. Full Control over Source Files - Synchronized Source and Visual Editing	164
8.2. Visual Page Editor	172
8.2.1.	172
8.2.1.1. Advanced Settings	179
8.2.2. Setup notes for Linux	184
8.2.2.1. How to Start the Visual Page Editor under Linux	184
8.2.3. JSP syntax validation	184
8.2.4. JSP Page Preview	184
8.3. More Editors	185
8.3.1. Graphical Properties Editor	185
8.3.2. Graphical TLD Editor	187
8.3.2.1. Tree view	187
8.3.2.2. Source view	187
8.3.3. Graphical Web Application File (web.xml) Editor	192
8.3.3.1. Tree View	192
8.3.3.2. Source View	193
8.3.3.3. Content Assist	194
8.3.3.4. Errors Checking and Validation	195
8.3.4. Graphical Tiles Files Editor	196
8.3.4.1. Graphical Editor For Tiles Files	196
8.3.4.2. Create New Tiles File	196
8.3.4.3. Tree View	197
8.3.4.4. Diagram View	199
8.3.4.5. Source	201
8.3.5. Graphical Editor for Struts Validation Files	204
8.3.6. Spring IDE	211
8.3.7. CSS Editor	216
8.3.8. JavaScript Editor	218
8.3.9. XSD Editor	220
8.3.10. Support for XML Schema	226
9. JBoss Tools Preferences	228
9.1. CodeAssist	230
9.2. Editors	231
9.3. JBoss Servers	233
9.4. JSF	234
9.5. JSF Flow Diagram	235
9.6. JSF Page	236
9.7. JSF Propject	238
9.8. Packaging Archives	240
9.9. Plug-in Insets	241
9.10. Resource Insets	243
9.11 Seam	

9.13. Struts       246         9.14. Struts Automatic       247         9.15. Struts Customization       248         9.16. Struts Flow Diagram       249
9.14. Struts Automatic
9.15. Struts Customization
9.16. Struts Flow Diagram
9.17. Struts Pages
9.18. Struts Project
9.19. Struts Support
9.20. Title Diagram
9.21. Verification
9.22. View
9.23. Visual Page Editor
9.24. XDoclet
9.25. XDoclet Templates
9.26. XDoclets Variables
9.27. Changing Default Environment During Project Creation
0.20 Changing Default Deviced Translate Device Deviced Constinue

# 1

# Visual Web Tools

In JBoss Tools there is an extensive collection of specialized wizards, editors and views that can be used in various scenerios. The following chapters walksthrough these features.

2

# **Key Features of Visual Web Tools**

Here is the table of main features of Visual Web Tools:

<b>Table 2.1.</b>	Key Functionali	ty for Visual We	b Tools

Feature	Benefit	Chapter
Visual Page Editor	Powerful and customizable visual page editor. Possibility to develop an application using any web tech- nology: jsf, seam, struts, jsp, html and others. Developing using four tabs: visual/source, visual, source and preview.Fast and easy switch- ing between these tabs. Split screen design of visual and source views. Full and instant synchroniz- ation between source and visual views. Integration with properties and outline views. Graphical tool- bar to add inline styling to any tag.	visual page editor
JBoss Tools Palette	Organizing various tags by groups, inserting tags into a jsp or xhtml page with one click, adding custom or 3rd party tag libraries into the palette, easy controling the number of tag groups shown on the palette.	jboss tools palette
Web Projects View	Visualizing and displaying projects by function. Easy selecting of dif- ferent kinds of items and dropping them into jsp pages. Using context menus to develop the application. Using icon shortcuts to create and import JSF and Struts projects. Ex- panding and inspecting tag library files. Selecting custom and third- party tag libraries to drag and drop onto the JBoss Tools Palette.	web projects view
JSF and Facelets support	Step-by-step wizards for creating	jsf support

Feature	Benefit	Chapter
	new JSF and Facelets projects with a number of predefined templates, importing existing ones and adding JSF capabilities to non-jsf web projects.	
Flexible and customizable project template management	Jump-start development with out- of-the-box templates or easily cus- tomized templates for re-use.	working with projects
Support for Managed Beans	Adding new managed beans, gen- erating code for attributes, proper- ties and getter/setter methods.	managed beans
OpenOn	Easy navigation between views and other parts of your projects.	facelets support
Content Assist	Code completion proposals while working in java, xml, jsp, xhtml, xhtml, seam project and jsf config- uration files. Content assist based on project data (dynamic code as- sist); with graphical editor. Code completion for values from prop- erty files, beans attributes and methods, navigation rule outcomes and jsf variables.	content assist
Support for Custom Converters and Validators	Fast creating of custom converters and validators with tree view of faces-config.xml file.	converters and validators
Verification and Validation	All occuring errors will be imme- diately reported by verification feature, no matter in what view you are working. Constant valida- tion and errors checking allows to catch many of the errors during de- velopment process that signific- antly reduces development time.	verification and validation
Drag-and-Drop	Possibility of inserting any tag onto the page you are editing by just drag-and-droping it from the palette to this page. Adding any properties, managed bean attrib- utes, navigation rules, tag library file declarations, jsp files from web projects view by clicking them and draging to source code.	visual page editor drag-and-drop

Feature	Benefit	Chapter
Struts Support	Step-by-step wizards for creating a new struts project with a number of predefined templates, importing existing ones and adding struts capabilities to non-struts web projects.	struts support
Support for JSF and JSF Configur- ation Files	Working on files using three modes: diagram, tree and source. Synchronization between the modes and full control over the code. Easy moving around the dia- gram using the Diagram Navigator. Working with struts projects that have multiple modules. Possibility to use Struts configuration file de- bugger allowing to set break points on struts diagram and then launch the server in debug mode.	graphical editor for jsf graphical editor for struts
Rich Faces Support	Tight integration between JBDS and RichFaces frameworks. Easy managing RichFaces components in any web application.	rich faces support

3

## **Java Server Faces Support**

JBoss Developer Studio is especially designed for supporting JSF and JSF-related technologies. JBDS provides extensible and exemplary tools for building JSF-based applications as well as adding JSF capabilities to existing web projects, importing JSF projects (created outside JBDS) and choosing any JSF implementation while developing JSF application.

JBoss Developer Studio allows you to develop JSF applications much faster and with far fewer errors so sparing your time.

# 3.1. Support for JSF Environments: JSF-RI, MyFaces, Facelets or any Custom

With JBoss Developer Studio, we don't lock you into any one JavaServer Faces implementation. Select the one you want to use for your project.

When you:

- Create a new JSF project
- Add JSF capability to any existing Eclipse project
- Add JSF capability to any existing project (created outside JBDS)

You can always select which JSF implementation to use.

You can also create your own custom JSF environments.

9	New JSF Project 🗙		
Create JSF Project			
The Create New Project Wizard is used for creating a brand new project. If you already have a pre-existing project, just use the Import Project Wizard to start working with it in Red Hat Developer Studio.			
Project Name*	JSFProject		
	✓ Use default path*		
Location*	/home/user/workspace/JSFProject Change		
JSF Environment*	JSF 1.1.01 - Reference Implementation		
Template*	JSFBlank		
0	< Back Next > Einish Cancel		

Figure 3.1. Choosing JSF Environment

JBoss Developer Studio will add all the required libraries for the selected version to your project.

#### 3.2. Facelets Support

Facelets extends JavaServer Faces by providing a lightweight framework that radically simplifies the design of presentation pages for JSF. JBoss Developer Studio provides support for Facelets in a variety of ways.

The Create New JSF Project wizard contains templates for creating Faceletsprojects based on version 1.2 of the JSF Reference Implementation:

۲	New JSF Project X		
Create JSF Project			
The Create New Project Wizard is used for creating a brand new project. If you already have a pre-existing project, just use the Import Project Wizard to start working with it in Red Hat Developer Studio.			
Project Name*	JSFProjectwithFacelets		
	Use default path*		
Location*	he/user/workspace/JSFProjectwithFacelets		
JSF Environment*	JSF 1.2 with Facelets		
	JSF 1.1.02 - Reference Implementation		
Template*	JSF 1.2		
	JSF 1.2 with Facelets		
	MyFaces 1.1.4		
0	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish Cancel		

Figure 3.2. Choosing Facelets Environment

Once you select the environment, you can select one of the six available templates:

😔 New JSF Project 🗙			
Create JSF Project			
The Create New Project Wizard is used for creating a brand new project. If you already have a pre-existing project, just use the Import Project Wizard to start working with it in JBoss Tools.			
Project Name*	JSFProjectWithFacelets		
	☑ Use default path*		
Location*	/root/workspace/JSFProjectWithFacelets		
JSF Environment*	JSF 1.2 with Facelets		
Template*	FaceletsBlankWithoutLibs		
	FaceletsKickStartWithRILibs		
	FaceletsKickStartWithoutLibs		
0	< <u>Back</u> <u>N</u> ext > <u>F</u> inish Cancel		

Figure 3.3. Choosing Facelets Template

The JBoss Tools Palette comes with the Facelets components ready to use. A useful tip appears when you hover the mouse cursor over the tag:



#### **Figure 3.4. Facelets Components**

Code assist for Facelets tags is available when editing xhtml files:

📾 inputname.xhtml 🕴	
<f:loadbundle basename="resources" var="msg"></f:loadbundle>	<b>_</b>
<pre><ui:composition template="/templates/common.xhtml"></ui:composition></pre>	
<pre>dui:define &gt;Input User Name</pre>	
<pre></pre>	The literal name for this definition. This name will match up with a tag in a target template.
<ul><li>define</li></ul>	
<h :="" mess<="" th=""><td></td></h>	
<form j<="" th=""><th></th></form>	
#{m	
<10	
<10	
<th></th>	
	u đầ
%	<b>_</b> _
Facelets Hello Application	
	_
Error Message	
#(msg.prompt) # {person.name} Say Hello	
	▼
Visual/Source Visual Source Preview	

Figure 3.5. XHTML File Code Assist

• In any HTML tag you also get the code assist for *"jsfc"* attribute



Figure 3.6. Code Assist for Jsfc Attribute

• Then you get the code assist for JSF components are available on the page

	<b>a</b>	*inputname.xhtml 🛙		- 6	1
		<pre>ns:f="http://java.sun.co ns:c="http://java.sun.co</pre>	m/jsf/core" m/jstl/core">	Ĥ	
		:f:loadBundle basename="r	esources" var="msg" />	Н	
		ui:composition template=	"/templates/common.xhtml">		
		<ui:define name="page&lt;/th&gt;&lt;td&gt;Title">JSF 1.2 and Facelets under Tomcat. KickStart Application<td>&lt;</td><td></td></ui:define>	<		
		<pre><ui:define body<="" name="page&lt;/pre&gt;&lt;/th&gt;&lt;td&gt;&lt;pre&gt;Header*&gt;JSF 1.2 Hello Application&lt;/ui:define&gt;&lt;/pre&gt;&lt;/td&gt;&lt;td&gt;c&lt;/td&gt;&lt;td&gt;-&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;th&gt;&lt;/th&gt;&lt;th&gt;đ&lt;/th&gt;&lt;th&gt;&lt;pre&gt;&lt;ui:define name=" th=""><th>"&gt; mmary="true" showDetail="false" style="color: red; font-weight: m" id="helloForm"&gt;</th><th>=</th><th></th></ui:define></pre>	"> mmary="true" showDetail="false" style="color: red; font-weight: m" id="helloForm">	=	
		<input jsfc="*&lt;/th"/> <th>" /&gt;</th> <th></th> <th></th>	" />		
Renders an HTML "input" elem Decode Behavior Obtain the Map from the "requ ExternalContext. If the Map co of the component, pass the v setSubmittedValue() method e an instance of EditableValue Encode Behavior Render the clientId of the com "name" attribute. Render the of the value of the "value" attribut specified, render its value as	ent iest onta alu of t lold poi cun ute	t of "type" "text". ParameterMap" property of the ains an entry for the "clientid" ie of the entry to the the component, which must be fer. nent as the value of the rent value of the component as . If the "styleClass" attribute is e value of the "class" attribute.	In triview In trivi	×	
			© h:inputText		
			(inputTextarea 🖉		

Figure 3.7. Code Assist for JSF Components

• Then you will see all available attributes for the component.



#### Figure 3.8.

Using JBoss Developer Studio OpenOn feature, you can easily navigate between the Facelets templates and other parts of your projects. Just by holding down the Control key while hovering the mouse cursor over a reference to a template, the reference becomes a hyperlink to open that template.



Figure 3.9. Template Hyperlink

#### 3.3. Working with Projects

#### 3.3.1. Creating a New JSF Project

JBoss Developer Studio provides the following when working with JSF:

- Create new JSF projects
- Import (open) existing JSF projects
- Add JSF capability to any existing Eclipse project
- Import and add JSF capability to any existing project created outside Eclipse

JBoss Developer Studio allows you to create brand new JSF projects . A new JSF project will have all JSF libraries, tag libraries and a JSF configuration file. JBoss Developer Studio comes with a number of predefined project templates. These templates are flexible and easily customizable.

To create a brand new JSF project, select *File* > *New* > *Project* > *JBoos Tools Web* > *JSF* > *JSF Project* and click *Next* :

<b>9</b>	New Project	X
Select a wizard		
<u>W</u> izards:		
type filter text		
🕨 🗁 J2EE		-
👂 🗁 Java		
👂 🗁 JBoss jBPM		
▽ 🗁 JBoss Tools Web		
🗢 🗁 JSF		
🛣 JSF Project		
👂 🗁 Struts		
👂 🗁 JPA		
👂 🗁 Plug-in Development		
< <u>B</u> ack	Next > Enish	Cancel

Figure 3.10. Choosing a JSF Project

On this form enter Project Name. You can also select where to create the project.

JSF Version allows you to select which JSF implementation to use:

New JSF Project					
Create JSF Project					
The Create New Project Wizard is used for creating a brand new project. If you already have a pre-existing project, just use the Import Project Wizard to start working with it in JBoss Tools.					
Project Name*	JSFProject				
	Use default path*				
Location*	/root/workspace/JSFProject Change				
JSF Environment*	JSF 1.1.02 - Reference Implementation				
Template*	JSFBlankWithLibs 🗧				
0	< Back Next > Einish Cancel				

Figure 3.11. Creating a New JSF Project

You can also pick a different template on which to base the project. Almost all templates come in two variations: with jsf libraries and without ones. Some servers already provide jsf libs and you take risk of getting conflicting libraries while deploying your project. So the idea here is to avoid such conflicts. Select a template without libs if you use a server with its own jsf libraries.

8	New JSF Project				
Create JSF Project					
The Create New Project Wizard is used for creating a brand new project. If you already have a pre-existing project, just use the Import Project Wizard to start working with it in JBoss Tools.					
Project Name*	MyProjectFacelets				
	Use default path*				
Location*	/root/workspace/MyProjectFacelets				
JSF Environment*	JSF 1.2 with Facelets				
Template*	FaceletsBlankWithoutLibs				
	FaceletsKickStartWithRILibs				
	FaceletsKickStartWithoutLibs				
0	< <u>Back</u> <u>N</u> ext > <u>Finish</u> Cancel				

Figure 3.12. JSF Templates

The JSFBlank template will create a standard Web project structure with all JSF capabilities.

The JSFKickStart template will create the same standard structure but will also include a sample application that is ready to run.

9	New JSF Project ×				
Create JSF Project					
The Create New Project Wizard is used for creating a brand new project. If you already have a pre-existing project, just use the Import Project Wizard to start working with it in JBoss Tools.					
Project Name*	JSFProject				
	☑ Use default path*				
Location*	/root/workspace/JSFProject Change				
JSF Environment*	JSF 1.2 with Facelets				
Template*	FaceletsBlankWithoutLibs				
	FaceletsKickStartWithRILibs				
	FaceletsKickStartWithoutLibs				
0	< Back Next > Finish Cancel				

Figure 3.13. Choosing JSF Template

On the next screen select what *Servlet version* to use and whether to register this application with JBoss AS (or other server) for running and testing of your application.

*Context Path* is the name under which the application will be deployed.

The *Runtime* value tells Eclipse where to find Web libraries in order to build (compile) the project. It is not possible to finish project creation without selecting Runtime. If you don't have any values, select *New*... to add new Runtime.

*Target Server* allows you specify whether to deploy the application. The Target Server corresponds to the Runtime value selected above.

If you don't want to deploy the application, uncheck this value:

<b>9</b>	New JSF Project	×
Web		
Servlet Version	2.5	•
Context Path*	JSFProject	
Runtime:*	JBoss 4.2 Runtime	New
Target Server:	JBoss Application Server 4.2	New Select All
(?)	< Back Next > Finish	Cancel
U.	Euron Deves	

Figure 3.14. Registering the Project on Server

When you are all done, you should have the project that has been created in the Package Explorer view:



Figure 3.15. A New Project in the Package Explorer

At this point you can open faces-config.xml and start working on your application.

#### 3.3.2. Importing Existing JSF Projects with Any Structure

For detailed information on migration projects to JBoss Developer Studio see Migration Guide [../../../../documentation/guides/build/Exadel-migration/en/html\_single/index.html].

#### 3.3.3. Adding JSF Capability to Any Existing Eclipse Project

With JBoss Developer Studio you can add JSF capability (JSF libraries, tag libraries) to any existing Eclipse project in your workspace.

By adding JSF capability to your project, you can now edit any file using JBoss Developer Studio editors, such as JSF configuration editor, JBoss Tools JSP editor and any others.

Right click the project and select *JBoss Tools* > *Add JSF Capabilities*. This will start the process of adding all necessary libraries, files to make this a Web JSF project:

Project Explorer	8 😑 😫 🗸 🗖 🖬		
👂 😂 jsfHello			
StratsHello			
🔻 👑 WebProject			
👌 🎥 Java Beso	New	•	
▷ ▷ build	Sho <u>w</u> In	Shift+Alt+W	
VebConte	Copy	Ctrl+C	
	Copy Qualified Name		
	Paste	Ctrl+V	
	× Delete	Delete	
	Build Path	,	
	Refactor	Shift+Alt+T 🕨	
	lana art		
	Import		
	Export		
	🔘 Run XDoclet	Shift+Ctrl+F1	
	Refresh	F5	
	Close Project		
	Close Unrelated Projects		
	Validate		
	Add Spring Project Nature		
	<u>R</u> un As	,	
	Debug As	,	
	Profile As	•	
	T <u>e</u> am	•	
	Comp <u>a</u> re With	•	
	Restore from Local History	<i></i>	
	Source	•	
	JBoss Tools	•	S Add Struts Capabilities
	PDE Tools	•	Add JSF Capabilities
	Java EE	•	Remove Struts Canabilities
	Properties	Alt+Enter	Add Custom Capabilities
	. Tob - 1948	- at terrest	Remove ISF Capabilities

Figure 3.16. Adding JSF Capabilities

The wizard will first show you the web.xml file location and the project name:

<b>9</b>	Import JSF Project	×
Project Location Please select web.xm	I location	
web.xml Location*	/home/user/workspace/WebProject/WebContent/WEB-INF/web.xml	<u>C</u> hange
Project Name*	WebProject	
	Help < Back Next > Finish	Cancel
Ø	Heip < Back Next > Buish	Cancel

#### Figure 3.17. Project Location

On the last form you can set the different folders for your project as well as register this application with a servlet container.

Make sure to select Add JSF Libraries for JBoss Developer Studio to add all required JSF related libraries to this project.

*Context Path* is the name under which the application will be deployed.

The *Runtime* value tells Eclipse where to find Web libraries in order to build (compile) the project. It is not possible to finish project import without selecting Runtime. If you don't have any values, select *New*... to add new Runtime.

*Target Server* allows you specify whether to deploy the application. The Target Server corresponds to the Runtime value selected above.

If you don't want to deploy the application, uncheck this value:

Import JSF Project X					
Project Folders					
Select Project Fold	ers				
Web Root:*	/home/user/workspace/WebProject/WebContent	<u>C</u> hange			
Source Folder	/home/user/workspace/WebProject/src	<u>C</u> hange			
Classes Folder	/home/user/workspace/WebProject/build/classes	Change			
Lib Folder	/home/user/workspace/WebProject/WebContent/WEB-INF/lib	<u>C</u> hange			
	Add Libraries				
Environment	JSF 1.1.02 - Reference Implementation	-			
Servlet Version:	2.4	•			
Context Path*	WebProject				
Runtime:*	JBoss 4.2 Server Adapter Runtime	New			
Target Server:	JBoss Application Server 4.2	New Select All Deselect All			
0	Help < Back Next > Finish	Cancel			

Figure 3.18. Project Folders

Once your project is imported you can see that JSF related libraries have been added to your project jsf-api.jar and jsf-impl.jar.

You are now ready to work with JSF by creating a new JSF configuration file:



Figure 3.19. Creating a New JSF Configuration File

#### 3.3.4. Adding Your Own Project Templates

JBoss Developer Studio has a powerful templating capability for creating new and importing existing Struts and JSF projects. This templating facility has a variety of aspects to consider. But, let's start with the most straightforward case.

Let's say you have a project that you want to use as the basis for a new template. Follow these steps to make a template out of it:

• In the Web Projects view, right-click the project and select JBoss Tools JSF > Save As Template as template

🔝 Web Projects δ	3 🔰 Package Exp	er 🗖 🗖		
1 I. S			ې 🖻	
ISFProject	JBoss Tools JSF	•	Save A	s Template
	🗙 Delete		Remov	e JSF Capabilities
Properties				
			Add Cu	istom Capabilities
			Add Of	RM Capability

Figure 3.20. Saving Your Project as Template

- In the first dialog box, you can choose a name for the template (defaults to the project name) and confirm what run-time implementation of the project's technology will be used
- Select *Next* and you will be sent to a dialog box with your project structure displayed with check boxes. Here you can check only those parts and files in your project directory that should be part of the template
- At this point, unless you want to designate some extra files as having Velocity template coding inside them, you should select Finish

That's it. Now, you can use this template with any new or imported project that uses the same run-time implementation as the project you turned into a template.

#### 3.4. Graphical Editor and Viewing for JSF Configuration Files

The JSF configuration file editor has three main viewers (modes):

- Diagram
- Tree
- Source

The modes can be selected via the tabs at the bottom of the editor.

The JSF configuration editor also comes with a very useful OpenOn selection feature.

#### 3.4.1. Diagram

The Diagram view displays the navigation rules in the JSF configuration file:

🔺 *faces	-config.xml ස	- 8
	hooseLocale.jsp	/storeFront.jsp carDetail
/c	arDetail.jsp	/confirmChoices.jsp carDetail customerInfo
		/customerInfo.jsp /finish.jsp finish
Diagram	Tree Source	

Figure 3.21. Diagram View

#### 3.4.2. Creating New View (Page)

To create a new page (view), you can click the page icon on this toolbar and then click anywhere on the diagram. A New Page Wizard will appear.

To create a transition (rule) connecting pages:

- Select the transition icon from the toolbar (2nd from the bottom).
- Click the source page.
- Click the target page.

A transition will appear between the two pages:

<u>7</u> *1	'aces-config.xml ස	
	/storeFront.jsp	/carDetail.jsp

Figure 3.22. Transition Between JSP Pages

It is also possible to create a new page by right-clicking anywhere on the diagram and selecting New View .

🔊 *faces-config.xml 🛛		- 8
storeFront /carDetail.jsp	/storeFront.jsp carDetail /confirmChoices.jsp Choices	
	Auto Layout Select Element	P
	🍄 Verify	
	Paste Ctrl + V	
	Input Methods	
Diagram Tree Source		

Figure 3.23. Creating a New View

To edit an existing transition, first select the transition line. Then, place the mouse cursor over the last black dot (on the target page). The mouse cursor will change to a big +. At this point, drag the line to a new target page:

	/carDatail ico
/storeFront.isp	, car be carrisp
	X 🔤 🗖
carDetail	
	/chooseLocale.jsp

Figure 3.24. Editing Transition Between Views

#### 3.4.3. Tree View

The Tree mode for the editor displays all JSF application artifacts referenced in the configuration file in a tree format. Select any node and its properties will appear in the right-hand area:

🖄 *faces-conf	ig.xml 🛙						- 8
Faces Con	fig Editor						
✓ faces-cor	nfig		✓ Managed Bean				<b>^</b>
🗢 🖹 faces-	config.xml*	Â	Managed-Bean-Name:	NA			
App	lication nponents		Managed-Bean-Class:	components.model.lmageAre			j
Þ 🗟 Cor	verters		Managed-Bean-Scope:	application		•	
🗢 🎯 Mar	aged Beans		Description:	Causes the def	ault VariableRes	solver imple 🦲	=
Þ 🥔 M	A			managed b	ean, NA of the o	class, comp 👻	
Þ 🥔 s	5A			4		•	
Þ 🥔 g	JerA		<ul> <li>Properties</li> </ul>				
Þ 🥏 f	raA		name	class	value	<u>A</u> dd	
00	ustomer		shape		poly	Remove	
00	arstore		alt		NAmerica	<u>L</u> emove	
🗢 🍓 Nav	igation Rules		coords		53,109,1,11	<u>E</u> dit	
Þ 🇞 /	chooseLocale					Up	
Þ 🇞 /	storeFront.jsp	•					
4						Down	-
Diagram Tree	Source						

Figure 3.25. Tree View

To edit, right-click any node and select one of the available actions in the context menu. You can also edit in the properties window to the right:

🖄 *faces-config.xml 🛙				- 0	
Faces Config Editor					
✓ faces-config	- Managed	Beans			
▽ 🏂 faces-config.xml* 🦳	name	class	scope	<u>A</u> dd	
Application	NA	components.m	od: applicatio	<u>R</u> emove	
Components	SA	components.m	ode application	Edit	
V 🤤 Converters	gerA	components.m	ode application	<u>E</u> uit	
New	•	Managed Bean onents.m	ode application	<u>U</u> р	
h a t cut	CHLIX	carstore.Custo	me session	Down	
		carstore.CarSto	ore session		
N Copy	Ctrl + V				
A Delete	Delete				
V 🖓 Na Properties					
Þ 🐞 💛 Verify					
👂 🍓 /storeFront.jsp 👻					
Diagram Tree Source					

Figure 3.26. Editing in Tree View

#### 3.4.4. Source View

The Source mode for the editor displays a text view of the JSF configuration file. All three viewers are always synchronized, so any changes made in one of the viewers will immediately appear in the others:

🔊 *faces-config.xml 🕴	• 0
<pre><managed-bean>     <description>     Causes the default VariableResolver implementation to instantiate     the managed bean, CustomerBean of the class, carstore.CustomerBean     in session scope if the bean does not already exist in any scope.     </description>     <managed-bean-name> customer </managed-bean-name>     <managed-bean-class> carstore.CustomerBean </managed-bean-class>     <managed-bean-scope> session </managed-bean-scope> </managed-bean>     <managed-bean>     </managed-bean>  &lt;</pre>	
<pre>The main backing file mean  <managed-bean-name> carstore </managed-bean-name> <managed-bean-class> carstore.CarStore </managed-bean-class> <managed-bean-scope> session </managed-bean-scope> </pre>	
<pre><navigation-rule>   <from-view-id>/chooseLocale.jsp</from-view-id>   <navigation-case>         <description>         Any action on chooseLocale should cause navigation to storeFront.jsp         </description>         <from-outcome>storeFront</from-outcome>         cfrom-outcome&gt;         cfrom</navigation-case></navigation-rule></pre>	
<to-view-id>/storeFront.jsp</to-view-id> 	•

Figure 3.27. Source View

#### 3.4.5. Content Assist

Content assist is always available in the Source viewer:



Figure 3.28. Content Assist in Source View

#### 3.4.6. Error Reporting

Errors will be reported by JBoss Developer Studio's verification facility:


#### Figure 3.29. Error Reporting in Source View

Other errors are also reported.

	<man -="" aged="" ame="" bean="" n=""></man>						
0	8 sometag						
0	<pre>2 </pre>						
	<navigation-rule></navigation-rule>						
	<pre><from-view-id>/chooseLocale.</from-view-id></pre>	<pre>jsp</pre>	ew-id>				
	<navigation-case></navigation-case>			•			
				•			
Di	agram Tree Source						
Rroblems 🕱 Tasks Servers							
	2 errors, 9 warnings, 0 infos						
2 e	rrors, 9 warnings, 0 infos						
2 e De	errors, 9 warnings, 0 infos escription	Resource	Path	Location			
2 e De	errors, 9 warnings, 0 infos escription <b>E Errors (2 items)</b>	Resource	Path	Location			
2 e D€ ⊽	errors, 9 warnings, 0 infos escription Errors (2 items) 0 181:12 Element type "sometag	Resource	Path cardemo/WebContent/WB	Location line 181			

Figure 3.30. Others Errors Reporting

You can also work in the Source viewer with the help of the *Outline view*. The Outline views show a tree structure of the JSF configuration file. Simply select any element in the Outline view, and it will jump to the same place in the Source viewer.



#### Figure 3.31. Outline View

If your diagram is large, within Outline view you can switch to a *Diagram Navigator* mode by selecting the middle icon at the top of the view window. It allows you to easily move around the diagram. Just move the blue area in any direction, and the diagram on the left will also move:

🔊 *faces-config.xml 업	🗝 🗖 🚼 Outline 🛛 🗧 🛃 🔻 🖻
/storeFront.jsp storeFront /carDetail /carDetail /carDetail confirmChoices.jsp confirmChoices.jsp confirmChoices.jsp /finish.jsp //customerinfo.jsp finish	/carDetail.jsp /carDetail.jsp /carDetail.jsp /confirmChoices.jsp confirmChoices /finish.jsp /finish.jsp finish
Diagram nee Source	

Figure 3.32. Outline View for Diagram

You can also edit in the *Tree* viewer with the help of the Properties view as shown below:

🔊 faces-config.xml 🛿			- 0	Properties 🛙	- 0
Faces Config Editor					🗉 🏶 🗔 🎽
	▼ Managed Bean		<u>^</u>	Property	Value
🔊 faces-config.xml	Managed-Bean-Name:	carstore		comment	The main backing file mean
Application	Managed-Bean-Class:	carstore.CarStore	Browse	display-name	l
Converters	Managed-Bean-Scope:	session	•	id	
🗢 🗟 Managed Beans	Managed Beans Description:	The main backing file mean		large-icon	
Þ 🥔 NA		_	~	managed-be	a carstore.CarStore
Þ 🥔 SA			▶	managed-be	a carstore
👂 🥏 gerA	<ul> <li>Properties</li> </ul>			managed-be	i session
👂 🏉 fraA	name clas value	Add	±	small-icon	
🥔 customer		Romo			
🥏 carstore		Kerno	ove		
👂 🍓 Navigation Rules		Edit	t		
🗟 Referenced Beans 🔤			p		
		Day			
Diagram Tree Source					

Figure 3.33. Properties View

# 3.5. Managed Beans

## 3.5.1. Code Generation for Managed Beans

JBoss Developer Studio gives you lots of power to work with managed beans.

- Add and generate code for new managed beans
  - Generate code for attributes and getter/setter methods
- Add existing managed beans to JSF configuration file

To start, create a new managed bean in JSF configuration file editor, in the Tree view.

🔊 faces-config.xml 🛛					- 8
Faces Config Editor					
🕶 faces-config	▼ Managed	i Beans			
🗢 📓 faces-config.xml 🦳	name		class	scope	<u>A</u> dd
Application	NA		components.mo	del application	Remove
Components	SA		components.mo	del application	
Converters	gerA		components.mo	del application	Edit
New	•	🥥 Managed Bear	nponents.mo	del application	Up
	611 · V		carstore.Custom	erE session	Down
	Ctrl + X		carstore.CarStor	e session	
Þ 🖉 fi 🕼 Paste	Ctrl + V				
Ø c	Delete				
Properties					
lefe 💛 Verify					
		1			
Diagram Tree Source					

Figure 3.34. Creation of New Managed Bean

When you define a new managed bean, make sure that Generate Source Code is checked as shown in the figure below:

۲	New Managed Bean 🗙
Manage	d Bean
Scope	request 💌
Class*	example.carBean <u>B</u> rowse
Name*	carBean
	Generate Source Code
	Next >> Finish Cancel

Figure 3.35. New Managed Bean

After the "Java" class has been generated you can open it for additional editing. There are two ways to open the "Java" class:

- Click on Managed-Bean-Class link in the editor -or-
- Right click the managed bean and select Open Source

🔊 *faces-config.xml 🛙					- 8	
Faces Config Editor						
✓ faces-config	✓ Managed Bean				<b>^</b>	
▽ 🔊 faces-config.xml* 📄	Managed-Bean-Name:	carBean			1	
Application Components	<u>Managed-Bean-Class:</u>	example.carBean		Browse	i	
D 🖓 Converters	Managed Open Scope:	request		•		
🗢 🎯 Managed Beans	Description:			A	i 🗉	
Þ 🥔 NA 📃		<u>el</u>				
Þ 🥔 SA		<u> </u>			1	
Þ 🥏 gerA	<ul> <li>Properties</li> </ul>					
🕨 🥔 fraA	name	class	value	<u>A</u> dd		
🥏 customer				Remove		
🥔 carstore 🔛						
🥏 carBean				Edit		
Navigation Rules				<u>Ц</u> р		
				Down	•	
Diagram Tree Source						

Figure 3.36. Opening of Created Managed Bean

The generated Java source:

🔊 *faces-config.xml	🕽 carBean.java 🛿	- 8
⊕/**		<u> </u>
<pre>package example;</pre>		
⊖/** * @author root * */ public class carB ⊖ public carBea } }	ean { in() {	
4	III	

Figure 3.37. Java Source Code

You can also generate source code for properties - this also includes "getter" and "setter" methods:



Figure 3.38. Generation of Source Code for Properties

Make sure that all the check boxes are selected:

- Add Java property
- Generate Getter
- Generate Setter

9	Add Property	×
Property		
Property-Name*	carName	•
Property-Class	java.Lang.String	<u>B</u> rowse
Value Kind	value	•
Value		<u>C</u> hange
	🗹 Add Java property	
	🗹 Generate Getter	
	🗹 Generate Setter	
		Finish Cancel

Figure 3.39. "Add Property" Form

Once the generation is complete, you can open the file and see the added property with "getter" and "setter" methods:



Figure 3.40. Generated Java Source Code for Property

## 3.5.2. Add Existing Java Beans to a JSF Configuration File

If you already have a Java bean you can easily add it to a JSF configuration file.

You should start the same way you create a new managed bean. Use Browse... to add your existing Java class.

۲	😸 New Managed Bean 🛛 🗙				
Manage					
Scope	request	•			
Class*	example.carBean	Browse			
Name*	carBean				
	Generate Source Code				
	Next >> Finish	Cancel			

Figure 3.41. New Managed Bean Form

Once the class is set, it's *Name* will be set as well. But you can easily substitute it for the other one. Notice that *Generate Source Code* option is not available as the "Java" class already exists.

After adding your class *Next* button will be activated. Pressing it you'll get *Managed Properties* dialog where all corresponding properties are displayed. Check the necessary ones to add them into your JSF Configuration File.

If you don't want to add any, just click Finish .

😸 New Managed Bean 🔀				
Managed Properties Select properties you want to add to the managed-bean				
name	value			
🗖 carName				
	<< Back Finish Cancel			

Figure 3.42. Selection of Bean's Properties.

# 3.6. Create and Register a Custom Converter

1. In the Project Explorer view open *faces-config.xml* and select *Tree* tab.

≰ *faces-config.xml 🛿			- 8
Faces Config Editor			
✓ faces-config	- Converter	rs	
▽ 🗟 faces-config.xml*	id	class	<u>A</u> dd
Application Components			<u>R</u> emove
🕨 🏟 Converters			<u>E</u> dit
👂 🍓 Managed Beans			Цр
Avigation Rules			Down
Referenced Beans			
Validators			
III	4	III	
Diagram Tree Source			

Figure 3.43. Converters

- 2. Select *Converters* and click on *Add* button.
- 3. Type the name of your converter in the Converter-id field and name of the class for converters. After clicking *Finish* button your custom converter is registered under entered name.

9	Add Converter	×
Converter		
Converter-id*	MyConverter	
Converter-Class*	etest.CustomConverter	<u>B</u> rowse
	Finish	Cancel

Figure 3.44. Add Converter Form

4. Let's create *"converter"* class. In the Converter section you should see your Converter-id and Converter-class. Click on *Converter-class* to generate the source code.

🔊 *faces-config.xml 🛿			- C
Faces Config Editor ▼ faces-config	• Converter		
Application     Components	Converter-id: Converter-for-Class	MyConverter	Browse
Converters	Description:		Browse
<ul> <li>Anaged Beans</li> <li>Avigation Rules</li> <li>Referenced Beans</li> </ul>	→ Attributes     ■		
Render Kits	name c	lass	Add <u>R</u> emove <u>E</u> dit <u>Up</u>
Diagram Tree Source			Down

Figure 3.45. Generation of Source Code for Converter Class

5. Java class will be created automatically. Leave everything without changes and click Finish .

9	New Java Class	×
<b>Java Class</b> Create a new Java c	lass.	C
Source fol <u>d</u> er:	JSFKickStart/JavaSource	Browse
Pac <u>k</u> age:	test	Bro <u>w</u> se
Enclosing type:		Browse
Na <u>m</u> e:	CustomConverter	
Modifiers:	public O default O private O protected	
	abstract final static	
<u>S</u> uperclass:	java.lang.Object	Brows <u>e</u>
Interfaces:		<u>A</u> dd
		<u>R</u> emove
Which method stub	s would you like to	
	public static void main(String[] args)	
	<u>C</u> onstructors from superclass	
Do you want to add	Inherited abstract methods	project?
Do you want to add	Generate comments	project?
0	Enish	Cancel

Figure 3.46. New Java Class Form

6. To open converter class click again on Converter-class link in the Converter section. Now you are able to write business logic of converter.



Figure 3.47. Converter Class

# 3.7. Create and Register a Custom Validator

1. In the Project Explorer view open *faces-config.xml* and select *Tree* tab.

🔊 faces-config.xml 🛿			- 8
Faces Config Editor			
▼ faces-config	- Validato	ors	
▼ 🔊 faces-config.xml	id	class	<u>A</u> dd
Components			<u>R</u> emove
👂 🖓 Converters			<u>E</u> dit
👂 🍓 Managed Beans			Up
Navigation Rules			
🗞 Referenced Beans			Down
🔄 Render Kits			
🕨 💕 Validators			
Diagram Tree Source			

Figure 3.48. Validator in Faces Config Editor

- 2. Select Validators, and click on Add button.
- 3. Type the name of your validator in the Valifator-id field and name of the class for validators. After clicking *Finish* button your custom validator is registered under entered name.

9	Add Validator	×
Validator		
Validator-id:*	MyValidator	
Validator-Class:*	e test.CustomValidator	<u>B</u> rowse
	Finish	Cancel

#### Figure 3.49. Adding Validator

4. Let's create "*validator*" class. In the Validator section you should see your Validator-id and Validator-class. Click on *Validator-class* to generate the source code.

🔊 *faces-config.xml 원				- 0
Faces Config Editor				
	- Validator			*
▽ 🔊 faces-config.xml*	Validator-id:	MyValidator		
Application Components	Validator-Class:	test.CustomValidator	<u>B</u> rowse	
D 🧠 Converters	Description:			=
👂 🍓 Managed Beans		( III		
Avigation Rules	- Attributes			
Referenced Beans	name	class	<u>A</u> dd	H
マ ¥ Validators			<u>R</u> emove	
MyValidator			<u>E</u> dit	
			Цр	
			Down	
	+ Properties			•
Diagram Tree Source				

Figure 3.50. Creating Validator Class

5. Java class will be created automatically. Leave everything without changes and click Finish .

9	New Java Class	×
<b>Java Class</b> Create a new Java c	lass.	C
Source fol <u>d</u> er:	JSFKickStart/JavaSource	Br <u>o</u> wse
Pac <u>k</u> age:	test	Bro <u>w</u> se
Enclosing type:		Bro <u>w</u> se
Na <u>m</u> e:	CustomValidator	
Modifiers:	public O default O private O protected	
	abstract final static	
<u>S</u> uperclass:	java.lang.Object	Brows <u>e</u>
Interfaces:		<u>A</u> dd
		Remove
Which method stub	s would you like to	
	public static void main(String[] args)	
	<u>Constructors</u> from superclass	
	Inherited abstract methods	
Do you want to add	comments as configured in the properties of the current	t project?
	Generate comments	
0	Einish	Cancel

Figure 3.51. New Java Class Form

6. To open validator class click again on Validator-class in the Validator section. Now you are able to write business logic of validator.



Figure 3.52. Converter Class Editing

## 3.8. Create and Register Referenced Beans

1. In the Project Explorer view open *faces-config.xml* and select *Tree* tab.

🖄 *faces-config.xml 🛙			- 8
Faces Config Editor			
	▼ References	i Beans	
✓ ▲ faces-config.xml* ♦ Application	name	class	Add
💫 Components D 🧠 Converters			Edit
Managed Beans Mavigation Rules			
Referenced Beans			Down
Validators			
Diagram Tree Source			

Figure 3.53. Validator in Faces Config Editor

- 2. Select *Referenced Beans* and click on *Add* button.
- 3. Type in the name of your Referenced Bean and type in or select Referenced-Bean-Class by using Browse button.

😔 😔	d Referenced Bean	×
Referenced Bean		
Referenced-Bean-Name:*	MyReferencedBean	
Referenced-Bean-Class:*	etest.ReferencedBean	<u>B</u> rowse
	Finish	Cancel

Figure 3.54. Add Validator Form

4. In the Referenced Bean section you should see your Referenced-Bean-Name and Referenced-Bean-Class. Click on the link to open the Java creation wizard.

ୀ sfaces-config.xml ଛ	- B
<ul> <li>★faces-config.xml X</li> <li>Faces Config Editor</li> <li>★ faces-config</li> <li>★ faces-config.xml*</li> <li>♦ Application</li> <li>♦ Components</li> <li>♦ ۞ Converters</li> <li>♦ ۞ Managed Beans</li> <li>♦ ۞ Navigation Rules</li> </ul>	■ ■ <ul> <li>Referenced Bean</li> <li>Referenced-Bean-Name:</li> <li>MyReferencedBean</li> <li>Browse</li> <li>Description:</li> <li> <li></li></li></ul>
<ul> <li>✓ Seferenced Beans</li> <li>✓ MyReferencedBean</li> <li>Sender Kits</li> <li>▷ ✓ Validators</li> </ul>	Id:       Display-Name:       Small-Icon:       Large-Icon:
Diagram Tree Source	

Figure 3.55. Create Validator Class

5. Java class will be created automatically. Leave everything without changes and click Finish .

9	New Java Class	×
<b>Java Class</b> Create a new Java c	lass.	C
Source fol <u>d</u> er:	JSFKickStart/JavaSource	Browse
Pac <u>k</u> age:	test	Bro <u>w</u> se
Enclosing type:		Bro <u>w</u> se
Na <u>m</u> e:	ReferencedBean	
Modifiers:	public O default O private O protected	
	abstract final static	
<u>Superclass</u> :	java.lang.Object	Brows <u>e</u>
Interfaces:		<u>A</u> dd
		<u>R</u> emove
Which method stub	s would you like to	
	public static void main(String[] args)	
	<u>Constructors from superclass</u>	
	Inherited abstract methods	
Do you want to add	comments as configured in the properties of the current	project?
	Ge <u>n</u> erate comments	
0	Einish	Cancel

Figure 3.56. New Java Class Form

6. To open Referenced Bean class click again on *Referenced-Bean-Class* in the Referenced Bean section. Now you are able to write business logic of Referenced Bean.



Figure 3.57. Referenced Bean Class Editing

# 4

# **Struts**

If you prefer develop web applications using Struts technology JBoss Developer Studio also meets your needs.

JBDS supports the Struts 1.1, 1.2.x versions.

# 4.1. Support for Struts 1.1, 1.2.x

When you create a brand new or import an existing project you can set which Struts version to use:

<b>@</b>	New JSF Project
Create JSF Projec	t 🍙 🔶
Specify Project Nan	ne 🥸
The Create New Pro a brand new project project, just use the with it in Red Hat De	ject Wizard is used for creating t. If you already have a pre-existing Import Project Wizard to start working eveloper Studio.
Project Name*	
	Use default path*
Location*	/home/user/workspace
JSF Environment*	JSF 1.2
Template*	JSF 1.1.02 - Reference Implementation
lemplate.	JSF 1.2
	JSF 1.2 with Facelets
	MyFaces 1.1.4
0	< Back Next > Bnish Cancel

Figure 4.1. Choosing Struts Environment

# 4.2. Working with Projects

## 4.2.1. Creating a New Struts Project

JBoss Developer Studio provides the following when working with Struts.

- Create new Struts projects
- Import (open) existing Struts projects

You can import any project structure

- Add Struts capabilities to any existing Eclipse project
- Import and add Struts capabilities to any existing project created outside Eclipse

JBoss Developer Studio includes a New Struts Project Wizard that radically simplifies the process for getting started with a new Struts project. You just need to follow these 4 steps:

1. Select *File* > *New* > *Project*... from the menu bar. Then, select *JBoss Tools Web* > *Struts* > *Struts Project* in this dialog box. Click *Next*:

New Project	X
Select a wizard	
<u>W</u> izards:	
type filter text	
✓ ➢ JBoss Tools Web ✓ ➢ JSF	
ISF Project	_
<ul> <li>▷ ▷ JPA</li> <li>▷ ▷ Plug-in Development</li> <li>▷ ▷ Seam</li> </ul>	
⑦ < Back Next > Binish Cancel	

Figure 4.2. Selecting Struts Wizard

2. On this screen, provide the project name. You can leave all other values as they are:

<b>9</b>	New Struts Project 🗙
Create Struts Proje	ct 🚳
The Create New Project a brand new project. If project, just use the Im with it in Red Hat Deve	ct Wizard is used for creating f you already have a pre-existing port Project Wizard to start working loper Studio.
Project Name*	strutsApplication
	✓ Use default path*
Location*	/home/user/workspace/strutsApplication
Struts Environment*	Struts 1.2
Template*	Blank
0	< Back Next > Finish Cancel

#### **Figure 4.3. Creating Struts Project**

### Tip:

Don't put spaces in project names.

#### Note:

If you select the KickStart template, then the project created will include a simple Hello World type application that is ready-to-run.

3. Next, you can register this application with the current servlet container defined for your workspace (JBoss AS, by default) in order to allow you to test your application more easily while still developing it. A new entry will be added in the servlet container configuration file to enable running the application in-place (called null deployment or link deployment). Uncheck the *"Target Server"* check box if for some reason you don't want to register your application at this point.

۲	New Struts Project	×
Web		
Servlet Version	2.4	
Context Path*	strutsApplication	
Runtime:*	JBoss 4.2 Runtime	▼ New
Target Server:	JBoss Application Server 4.2	New Select All Deselect All
0	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish	Cancel

Figure 4.4. Registering The Project at Server

4. On the next screen, you can select the TLD files to include in this project:

<b>9</b>	New Struts Project	×
Tag Libraries Select tag library files y	you want to use in your project	
TLDs		
<ul> <li>struts-nested.tld</li> <li>fmt.tld</li> <li>sql.tld</li> <li>c.tld</li> <li>x.tld</li> <li>struts-html.tld</li> <li>struts-bean.tld</li> <li>struts-logic.tld</li> <li>struts-tiles.tld</li> </ul>		
0	< <u>Back</u> <u>Next</u> <u>Finish</u>	Cancel

Figure 4.5. Selecting Tag Libraries

After the project is created, you should have the following project structure (if you used the KickStart template):



Figure 4.6. Project Structure

## Tip:

If you want to hide the jar files from Web App Libraries in view, select the down-pointing arrow in the upper right corner, select *Filters*..., check the box next to Name filter patterns (matching names will be hidden), and type \*.jar into the field. Then, click OK.

## 4.2.2. Importing an Existing Struts Project with Any Structure

For detailed information on migration projects to JBoss Developer Studio see Migration Guide [../../../../documentation/guides/build/Exadel-migration/en/html\_single/index.html].

## 4.2.3. Adding Struts Capability to an Existing Web Application

With JBoss Developer Studio you can add Struts capabilities (including Struts libraries, tag libraries and a Struts configuration file) to any existing Web application project in your Eclipse workspace. By adding a Struts Nature to your project, you can now edit files using JBoss Developer Studio editors, such as the Struts configuration editor and the JBoss Tools JSP editor.

Right-click the project and select *JBoss Tools* > *Add Struts Capabilities* from the context menu. This will start the process of adding all necessary libraries and files to make this a Web JSF project.

Package Ex	plorer 12 Neb Projects	- 6	
гаскауе сх		▼	
🐸 JSFHello			1
	Ne <u>w</u>	,	
	Go into		
	Open in <u>N</u> ew Window		
	Ope <u>n</u> Type Hierarchy	F4	
	Sho <u>w</u> In	Shift+Alt+W	
	Сору	Ctrl+C	
	Copy Qualified Name		
	👔 <u>P</u> aste	Ctrl+V	
	X Delete	Delete	
	Build Path	,	
	Source	Shift+Alt+S	
	Refactor	Shift+Alt+T	
	import		
	Export		
	🔘 Run XDoclet	Shift+Ctrl+F1	
	🔶 Refresh	F5	
	Clo <u>s</u> e Project		
	Assign Working Sets		
	<u>R</u> un As	•	
	<u>D</u> ebug As	•	
	Profile As	•	
	Validate		
	Add Spring Project Nature		
	T <u>e</u> am	,	
	Comp <u>a</u> re With	,	
	Restore from Local History		S Add Struts Capabilities
	JBoss Tools	+	Add JSF Capabilities
	PDE Tools	•	Remove Struts Capabilities
	Java EE	,	Add Custom Capabilities
	Properties	Alt+Enter	Remove JSF Capabilities

18

Þ

#### Figure 4.7. Adding Struts Capabilities

The wizard will first show you the web.xml file location and the project name.

9	Import Struts Project	×
Project Location Please select web.xm	llocation	
web.xml Location*	/home/user/workspace/jsfHello/WebContent/WEB-INF/web.xml	<u>C</u> hange
Project Name*	jsfHello	
0	Help < Back Next > Einish	Cancel

**Figure 4.8. Choosing Project Location** 

After hitting *Next*, you will see the following screen. This screen simply means that you need to add at least one Struts module to your project to make this project a Struts project. Adding a Struts module means that a new struts-config.xml will be added to your project and registered in the web.xml file. In addition, all required Struts libraries will be added. To add a Struts module, select the *Add Struts Support* button.

<b>9</b>	Import Struts Project	×		
Project Module	Project Modules			
SThe project sho	ould contain at least one Struts module to be imported			
1				
Name	URI			
Name				
URI				
Path on Dick		Change		
rationoisk		<u>Cnange</u>		
Sources Path		<u>C</u> hange		
Module Root		Change		
	Add Struts Support			
0	Help < Back Next > Enish	Cancel		

Figure 4.9. Project Modules

Here you can select what Struts version, Servlet class, URL pattern and TLDs to add to this project.
( <b>9</b> )	Add Struts Support 🛛 🗙
Struts Project Please select stru	ats support options
Version*	1.2
Servlet Class:*	org.apache.struts.action.ActionServlet
URL Pattern:*	*.do
TLD Files:*	<ul> <li>struts-nested.tld</li> <li>fmt.tld</li> <li>sql.tld</li> <li>c.tld</li> <li>x.tld</li> <li>struts-html.tld</li> <li>struts-bean.tld</li> <li>struts-logic.tld</li> <li>struts-tiles.tld</li> </ul>
0	<u>F</u> inish Cancel

Figure 4.10. Selecting Struts Support Options

When done, you will see the default Struts module configuration information. See how to Edit Struts modules.

9	Import Struts Project 🗙
Project Modu Configure Proje	iles ct Modules
Name	URI
<default></default>	/WEB-INF/struts-config.xml
Name	<default></default>
URI	/WEB-INF/struts-config.xml
Path on Disk	/home/user/workspace/jsfHello/WebContent/WEB-INF/struts-config.xml
Sources Path	/home/user/workspace/jsfHello/JavaSource
Web Root	/home/user/workspace/jsfHello/WebContent
0	Help < Back Next > Finish Cancel

Figure 4.11. Project Configuration Information

On the last screen you can set the different folders for your project as well as register this application with a servlet container.

<b>9</b>	Import Struts Project	×
Project Folders	•	
Select Project Fol	ders	S C
Classes Folder	/home/user/workspace/jsfHello/WebContent/WEB-INF/classes	<u>C</u> hange
Lib Folder	/home/user/workspace/jsfHello/WebContent/WEB-INF/lib	Change
Ant Build File		<u>C</u> hange
	Add Libraries	
Environment	Struts 1.2	•
Servlet Version:	2.4	-
Context Path*	jsfHello	
Runtime:*	JBoss 4.2 Server Adapter Runtime	New
Target Server:	JBoss Application Server 4.2	New
		Select All
		Deselect All
0	<u>H</u> elp < <u>B</u> ack <u>N</u> ext > <u>F</u> inish	Cancel

Figure 4.12. Registering the Project at Server

When done, you can open the struts-config.xml file. (The Struts configuration is shown below in the Tree viewer).

😫 Package Explorer 🕄 📃 🗖	🐟 struts-config.xn	ni 🕱		- t	8
수 수 👰 🖻 🔩 🍸	Struts Confi	g Editor			
▽ 🚰 jsfHello		g	• Properties E	ditor	
JavaSource	🔻 🛸 struts-con	ifig.xml	struts-config	1.2	
🕨 🛋 Web App Libraries	🖂 data-so	ources	Name:	struts-config	
IRE System Library [jdk1.5.0_12]	🗟 form-b	eans			
👂 🗁 ant	alobal-	exceptions	Extension:	xml	
▽ 🗁 WebContent	alobal-	forwards	Display-Name:		
D 🗁 META-INF	action-	mappings	Description:		
VEB-INF	Secontrol	ler			
Dib 😂 lib	les resource	-es	ld:		
🖹 faces-config.xml	🗟 plug-in	s	Comment:	Change	
🗇 struts-bean.tld	- p	-	Encoding	UTE-8	
X struts-config.xml			cheoding.		
🗇 struts-html.tid					
🗇 struts-logic.tld					
🗟 web.xml 👻					
🗖 Properties 🔝 Web Projects 🛿 🗧 🗖					
🛣 🛋 这 😒 😑 🔄					
isfHello					
▽ 🔊 strutsApplication					
WEB-ROOT (WebContent)					
ా స్త్రీ default					
🕨 📫 struts-config.xml					
▷ 👜 web.xml	Diagram Tree So	urce			

Figure 4.13. Struts-config.xml File

## 4.3. Graphical Editor for Struts Configuration Files

The Struts configuration file editor has three main viewers (modes): Diagram (shown), Tree and Source. The modes can be selected via the tabs at the bottom of the editor. Any changes made in one mode are immediately visible when you switch to any other mode.

When working in Source view, you always have all the following features available:

- Content Assist
- Open On Selection
- File Folding

## 4.3.1. Diagram Mode

🗟 *struts-config.xml 🕱		- 8
Image: space of the system       Image: space of the system	/pages/inputname.jsp	
	/greeting	
		/pages/greeting.jsp ≫
Diagram Tree Source		

The Diagram mode graphically displays the Web flow of the application defined in the Struts configuration file.

Figure 4.14. Diagram Mode

Just by right-clicking anywhere on the diagram, you can use a context menu to create the building blocks of a Struts application:

- Actions
- Global forwards
- Global exceptions
- JSP Pages

🥐 getName	T	
	/pages/inputname.jsp	
	Add	Action
	Create Comment	💍 Global Forward
	💼 Paste	Ctrl + V
	Select Element	
	Auto-Layout	
	💛 Verify	pages/greeting.jsp
	Properties	
	🅈 Generate Java Code	

Figure 4.15. Diagram Context Menu

Along the upper-left side of the editor is a stack of seven icons for changing the behavior of the cursor in the diagram. The first icon switches to the default regular selection cursor, the second to the marquee selection cursor and the third to the new connection cursor. The last four icons switch the cursor to an insert cursor for each type of Struts build block listed above (and in the order listed).

For instance, clicking on the first of these four icons (the one with the gears) will switch the cursor to insert actions. Clicking anywhere in the diagram with this cursor has the same effect as right-click and selecting Add > Action... from the context menu with the regular cursor active. It's just more efficient to use this cursor if you're adding more than one action at once.



Figure 4.16. Insert Actions Cursor

## 4.3.2. Tree Mode

In the Tree mode, the different elements of the Struts application are organized into functional categories on the left-hand side and a form for editing the properties of currently selected items on the right-hand side.

🕏 *struts-config.xml 🛛			- 8
Struts Config Editor			
	• Properties Ed	itor	<b>^</b>
▽ 🚔 struts-config.xml*	forward		
🛵 data-sources	ld:		
👂 🗟 form-beans	ClassName:	Browse	i
🔯 global-exceptions			
👂 🧼 global-forwards	ContextRelative:	·	
action-mappings 🤯	Module:		
🗢 🍓 /greeting	Name	savhello	ill
👉 sayhello			
controller	Path:	/pages/greeting.jsp <u>C</u> hange	
kan and the sources and the sources and the sources are sources and the sources are sources and the sources are so	Redirect:	▼	
🍋 plug-ins	Small-Icon:		j 🗋
	Large-Icon:		
	Display-Name:		] .
Diagram Tree Source			

Figure 4.17. Tree Mode

You can also right-click on any node in the category tree and perform appropriate operations through a context menu. For instance, by right-clicking on the action-mappings category node, you can add new actions to the application.

🕏 *struts-config.xml 🛿		- 0
Struts Config Editor		
✓ struts-config	- Properties Editor	
▽ 🚔 struts-config.xml*	action-mappings	
🖂 data-sources	ld:	
👂 🎯 form-beans	Type:	
lobal-exceptions ▷ 🔯 global-forwards	Comment:	<u>C</u> hange
🗢 🍓 action-ma 🚳 Create	Action	
✓ ·@ /greetir Sort		
controller     Paste	Ctrl + V	
a resources Proper	ies	
🍣 plug-ins 😵 Verify		
Diagram Tree Source		

Figure 4.18. Tree Context Menu

## 4.3.3. Source Mode

In the Source mode, you have complete editing control of the underlying XML coding:

Struts





#### Figure 4.19. Source Mode

You can take advantage of code assist:

- 8

Struts

★ *stm x</p D</p <st< p=""> <d< p=""> O</d<></st<>	uts-config.xml 83 ml version="1.0" encoding="UTF-8"?> OCTYPE struts-config PUBLIC "-//Apache Software "http://struts.apach ruts-config> ata-sources/> • description • display-name • action-mappings • controller • data-sources • form-beans • global-exceptions • global-forwards • message-resources • plug-in • comment - xml comment	Foundation//DTD Struts Configuration 1.2//EN" te.org/dtds/struts-config_1_2.dtd"> Element : display-name The "display-name" element contains a short (one line) description of the surrounding element, suitable for use in GU tools. Data Type : #PCDATA	
Diagra	m Tree Source		

Figure 4.20. Code Assist

The editor will also immediately flag any errors:



Figure 4.21. Errors in Source Mode

Finally, you can use the Outline view with the editor to easily navigate through the file:



Figure 4.22. Outline View

# 4.4. Graphical Editor for Tiles Files

The Tiles configuration file editor has three main viewers (modes): Tree (shown), Diagram and Source. The modes can be selected via the tabs at the bottom of the editor. Any changes made in one mode are immediately visible when you switch to any other mode.

When working in Source view, you always have all following features available:

- Content Assist
- Open On Selection

## 4.4.1. Create New Tiles File

To create a new Tiles files, right click any folder and select New > Tiles File.



Figure 4.23. Creating a New Tiles File

#### 4.4.2. Tree View

In the Tree mode, the different elements of the Tiles file are organized into functional categories on the left-hand side and a form for editing the properties of currently selected items on the right-hand side.

🗄 tiles-defs.xml 🕴				- 6
Tiles Editor				
✓ tiles-defs	▼ Tiles Cor	nfig Descript	ion	<u> </u>
🗢 📓 tiles-defs.xml	Name:	tiles-defs		
tiles.layout	Encoding			
tiles.layout.weivgroup	cricourig.	L		
tiles.layout.weivitems	- Definitio	ns		
▽ 🖺 tiles.layout.error	name	extends	path	<u>A</u> dd
🔳 title	tiles.layout	:	/jsp/main.jsp	Remove
action	tiles.layout		out	Demove
chooser	tiles.layout		out	<u>E</u> dit
🔚 tiles.head	tiles.layout	.error tiles.lay	out	<u>Up</u>
🔚 tiles.chooser	tiles.head		/tiles/header.jsp	
🔚 tiles.action	tiles.choos	er	/tiles/chooser.jsp	Down
🔚 tiles.foot	tiles.action	1	/tiles/action.jsp	
🔚 tiles.weivgroup	tiles.foot		/tiles/foot.jsp	
🔚 tiles.weivitems	tiles.weivg	roup	/tiles/WeivGroup.jsp	,
🔚 tiles.groups	tiles.weivit	ems	/tiles/WeivItems.jsp	
🔚 tiles.items	tiles.group	s	/tiles/Groups.jsp	
🔚 tiles.error	tiles.items		/tiles/items.jsp	
( III )	tiles.error		/tiles/Error.jsp	*
Tree Diagram Source				

Figure 4.24. Tree View

To edit the file, simply right click any node and select among the available actions:

<ul> <li>tiles.layout</li> <li>tiles.layout.weivgroup</li> <li>tiles.layout.weivitems</li> </ul>	Exte Path	nds	tiles layout
V ilectorer		rollerClass	
Add PutList		rollerUrl	
Rename		/anced	
🔚 tile 👔 Copy 🛛 🕻	Ctrl + C		
🔚 tile 🏦 Paste 🛛 🖓	Ctrl + V	av-Name:	
🔚 tile		ay name.	
🔚 tile 🗡 Delete	Delete	ription:	
🔚 tile 🛛 Properties			
🔚 tiles.weivitems	6		
🔚 tiles.groups	Sma	II-ICON:	
🔚 tiles.items	Larg	e-lcon:	
🔚 tiles.error	▼ Dej	precated	
4	Page	/jsp/	main.jsp
Tree Diagram Source			

Figure 4.25. Editing in Tiles Editor

## 4.4.3. Diagram View

The Diagram mode is shown below:

Struts



Figure 4.26. Diagram Mode

To create new definitions, simply right click anywhere in the diagram:

tiles.head	
tiles.chooser New Def	finition
of Cut	Ctrl + X
s.action Copy	Ctrl + C
🖹 Paste	Ctrl + V
Delete	Delete
Preferen	ces
weivgroup Input Me	thods •
tiles.weivitems	

Figure 4.27. Creating New Definition

You can also use the Diagram toolbar to make editing easier:



Figure 4.28. Diagram Toolbar

#### 4.4.4. Source

The Tiles editor also comes with a Source view that gives you full control over the source. Any changes here will immediately appear in other viewers when you switch to them.

Struts



#### Figure 4.29. Source View

Content assist is available in the Source mode:

	<tiles-definitions> <definition name="tiles.layout" path="/jsp/main.jsp"> <put name="tile" value="Main page"></put> <put name="header" value="tiles.head"></put> <put name="chooser" value="tiles.chooser"></put> <put name="action" value="tiles.action"></put> <put name="footer" value="tiles.foot"></put> </definition></tiles-definitions>		
	definition     Element : definition	yout">	
	# comment - xml comment that can be inserted in a isp page. This de	tion	
	# XSL processing instruction - XSL processing in is identified by its logical name. A definition	n	
	allows to define all the attributes that can	be set	
	in tag from a jsp page, controllerClass The gualified Java class name of the controlle	r your >	
	subclass to call immediately before the til	les is	
	inserted. Only one of controllerClass or	and led	
	The context-relative path to the resource	used as >	
	controller called immediately before the t	iles is	H
	inserted. Only one of controllerClass or		
	<definition name="tiles.head" path="/tiles/header.jsp"></definition>		
	<pre><definition name="tiles.chooser" path="/tiles/chooser.jsp"></definition></pre>		
	<pre><definition name="tites.action" path="/tites/action.jsp"></definition></pre>		
-	Tea Diagram Sauraa	•	
Ir	line Diagram Source		

Figure 4.30. Content Assist in Source Mode

Any errors are immediately reported as shown below:



**Figure 4.31. Errors Reporting** 

You can also use the Outline view together with the editor's Source mode. Selecting any node in the Outline view will jump to that place in the source.



Figure 4.32. Outline View

# 4.5. Graphical Editor for Struts Validation Files

JBoss Developer Studio comes with a visual validation editor. To create a new validation file, right click any folder and select *File > Validation File* from the context menu.

Ne <u>w</u>	÷.	🎦 Project
Go Into		# Package
Open in <u>N</u> ew Window		© Class
Ope <u>n</u> Type Hierarchy	F4	😯 Interface
Sho <u>w</u> In	Shift+Alt+W	💕 Source Folder
Conv.	CHUC	😂 Folder
Copy Qualified Name	Culte	😭 File
Paste	Ctrl+V	<table-of-contents> Struts Config</table-of-contents>
Y Delete	Delete	🏂 Faces Config
	Delete	🖹 Tiles File
<u>B</u> uild Path	•	😽 Validation File
Source	Shift+Alt+S	炎 TLD File
Refactor	Shift+Alt+T 🕨	🚵 JSP File
No Import		🚵 XHTML File
import		📸 HTML File
Export		🚳 CSS File
🔘 Run XDoclet	Shift+Ctrl+F1	🖄 JS File
🔗 Re <u>f</u> resh	F5	📸 Properties File
Clo <u>s</u> e Project		Example
Close Unrelated Projects		Ca cZampie
Assign Working Sets		📑 <u>O</u> ther

Figure 4.33. Creating New Validation File

The validation editor works through a number of viewers.

The Formsets viewer shows forms and their elements for which to define validation rules.

🗹 *MyValidation.xml 🛙				- 8		
Formsets	Page 1			Edit		
▽ 🧐 formset (default)	Msg - Corresponded Message Template					
<ul> <li>Constants</li> <li>AlertWizard</li> </ul>	name	key	resource	Add		
dashboardManagerForm Ø DashboardWizardNew				Edit		
V ashboardViewForm	Arg - Replacement	Value for Message Tem	nplate	Delete		
	name	arg key	resource	Add		
🧐 mask	required	argo Name	Taise	Edit		
				Delete		
	var - Validator Parar var-name	var-value		Add		
	mask	^[()A-Za-z0-9()_]+	-[A-Za-z0-9']	Edit		
				Delete		

Figure 4.34. Formsets Viewer

The Constants viewer lets you set constant values for your validation rules.

MyValidation.xml	B						- 8
Current Global Secti	on globa 🖨	Ň 🕯					
constant-name	co	onstant-va	alue				Add
standartName	^	[()A-Za-z	:0-9_]+[A-Z	Za-z0-9'_]	*[()A-Za-z	:0-9_]*\$	Edit
							Delete
Formsets Validators	Constants Tree	Source					

Figure 4.35. Constansts Viewer

The validation file also can be viewed in a Tree viewer.

🗹 MyValid	ation.xml	×			- B
🗢 🗹 My V	alidation.xr	nl			
💖 gi	obal				
🗢 💖 fo	ormset (def	ault)			
۵ 🖉	AlertWiza	rd			
۵ 🖉	dashboar	dManagerFo	orm		
۵ 🖉	Dashboar	dWizardNe	w		
~ ⊘	dashboar	dViewForm			
~	name				
	💖 mas	sk			
	0 arg	) for require	ed.		
			-	-	
Formsets	Validators	Constants	free	Source	

#### Figure 4.36. Tree Viewer

At any point you have full control over the source by switching to the Source viewer. Any editing in this viewer will immediately be available in other viewers of this editor.

Struts



Figure 4.37. Source Viewer

You can also open your own custom or Struts-standard validation-rules.xml file.

The Validators viewer shows the validation rules for a selected validator. You can of course add your own rules.

MyValidation.xml 🛙	- E
Current Global Section globa 🕏	🖺 📋
-Validators	Depends Change
name	Message Key errors.timeornumber Change
long	Java Class Name com.echopass.provisioning.rra.validator.TimeOrNumbe Change
float	Method validateTimeOrNumber Change
date	Method Param java.lang.Object, org.apache.commons.\ Change
intRange	JavaScript
creditCard	Function Name Change
email	Function Body
un time number	var timeMask = "^([0-9]){1,3}(:[0-5][0-9]){0,2}\$"; var numberMask = "^[0-9]+.?[0-9]?\$";
time_numberforthresholds	var booleanMask = "^active inactive\$"; var anvMask = "^ *\$":
	var percentMask = "^[0-9]+.?[0-9]?%?\$";;
	var statistic = new Array();
Formsets Validators Constants Tre	ee Source

Figure 4.38. Validators Viewer

Here are the validation rules shown in the Source viewer.

Struts



Figure 4.39. Validation Rules

# 4.6. Support for Multiple Struts Modules

## 4.6.1. Struts Modules

JBoss Developer Studio supports working with Struts projects that have multiple modules. You can easily do the following:

- Add new modules
- Edit modules for an existing project or during Struts project import

## 4.6.2. When Importing a Struts Project

During Struts project import, if the project has multiple modules, you will see a screen with all existing modules. You can select each module and edit its details.

<b>9</b>	Import Struts Project	X
Project Modu Configure Proje	<b>iles</b> ct Modules	<b>(</b>
Name	URI	<u>^</u>
<default></default>	/WEB-INF/struts-config.xml	
	/WEB-INF/struts-config-common.xml	
	/WEB-INF/struts-config-common_links.xml	•
Name URI Path on Disk	<default> /WEB-INF/struts-config.xml /home/user/Desktop/Provisioning/WebContent/WEB-INF/struts-config.xml</default>	<u>C</u> hange
Sources Path		<u>C</u> hange
Web Root	/home/user/Desktop/Provisioning/WebContent	<u>C</u> hange
0	< <u>B</u> ack <u>N</u> ext > <u>Finish</u>	Cancel

**Figure 4.40. Configuring Project Modules** 

## 4.6.3. Editing Modules in an Existing Project

To edit modules in an existing project, right click the project and select JBoss Tools > Modules Configuration.

Validate		
Add Spring Project Nature		
T <u>e</u> am	•	S Add Struts Capabilities
Comp <u>a</u> re With	,	Add JSF Capabilities
Restore from Local History		
JBoss Tools	•	Remove Struts Capabilities
PDE Tools	•	Add Custom Capabilities
Java EE	•	Remove JSF Capabilities
Properties	Alt+Enter	Verify
		Modules Configuration

#### Figure 4.41. Choosing Modules Configuration

You will see the same screen as above where you will be able to select a module and edit its details:

<b>9</b>	Modules Configuration	×
Struts Projec	t	
Name	URI Delete	d Add
<default></default>	/WEB-INF/struts-config.xml	Delete
	/WEB-INF/struts-config-common.xml	Delete
	/WEB-INF/struts-config-common_links.xm	
	/WEB-INF/struts-config-eca.xml	•
Name URI	<default> /WEB-INF/struts-config.xml</default>	
Path on Disk	esktop/Provisioning/WebContent/WEB-INF/struts-config	g.xml <u>C</u> hange
Sources Path	/home/user/workspace/Provisioning/src	<u>C</u> hange
Web Root	/home/user/Desktop/Provisioning/WebContent	<u>C</u> hange
	Finis	h Cancel

**Figure 4.42. Modules Configuration** 

## 4.6.4. Adding New Modules

Adding a new module is very simple. First switch to Web Project view. Expand your project to the Configuration folder. Under that folder you should see the current modules. Right click on Configuration and select New > Struts Config.



Figure 4.43. Adding New Modules

You will see the screen below. You can specify a new module name and also add the new Struts configuration file to web.xml file.

۲	Struts Config 🛛 🗙
Struts C	onfig
Folder*	/StrutsHello/WebContent/WEB-INF Browse
Name*	struts-config-mod1
Version	1.1
Module	/mod1
	Register in web.xml
	Finish Cancel

Figure 4.44. Adding New Modules

# 4.7. Code Generation for Action, FormBean, Forward and Excep-

## tion Classes

JBoss Developer Studio comes with a code generation feature. You can generate stub code for Struts Actions, FormBeans, Forwards and Exceptions.

The code generation in JBoss Developer Studio is based on Velocity templates which can be modified for your use. The templates are located at *{RedHatDeveloperStudioHome} > templates > codegeneration*.

There are a number of ways to invoke code generation. One is simply right-clicking the Struts diagram and selecting *Generate Java Code*....

2				
1	🥐 getName			
۵		/pages/inputnam	e.jsp	
Č 4		> <u>@</u>	Add	•
å			Create Comment	
			Paste Ctrl + V	/
			Select Element	ng.jsp
			Auto-Layout	
		•	Verify	
			Properties	
		3	Generate Java Code	
			Input Methods	•
Dia	gram Tree Source			

Figure 4.45. Selecting Generate Java Code

On this screen you can select for which elements to generate code. If you select Next you will be able to specify more options for each of the categories you selected.

	Generate Java Code - Step 1 🗙
	Generate classes for Actions
	Generate classes for FormBeans
	Generate classes for Forwards
	Generate classes for Exceptions
	Overwrite existing files
Base Package:	
	Next >> Generate Cancel

#### Figure 4.46. Generate - Step 1

## Tip:

please be careful not to override your existing files.

When generation is complete, a result window will appear letting you know how many classes were generated:

<b>9</b>	Generate - Finish	×
Generation finished.		<b>(</b>
Message		
Generated classes: 1 Form beans: 1		
(	111	
		Finish

**Figure 4.47. Generation Finished** 

You don't always have to generate code for all elements at once. You can invoke generation for just an individual Struts artifact as well. Right-click an element on the diagram of the Struts configuration file and select *Generate Java Code*... from the context menu.



#### Figure 4.48. Generation For Individual Struts Artifact

The same can be done from within the Tree viewer for the editor of the Struts configuration file.



Figure 4.49. Generation in Struts Config Editor

# 4.8. Struts Configuration File Debugger

JBoss Developer Studio comes with Struts configuration file debugger. It allows you to set break points on Struts diagram and then simply launch the server in debug mode.

Simply right click an Action or a page and select Add Breakpoint.

🐟 *s	struts-config.xml 🛿			- 0	
	struts-config.xml 🛛	/pages/inputname.jsp	t		
		Run on Server	tem	'greeting.jsp	
		Copy Cut	Ctrl + C Ctrl + X Ctrl + V		
		🗙 Delete	Delete		
		Add Breakpoint			
		Input Methods	•		
Diag	ram Tree Source				

Figure 4.50. Adding Breakpoint

## 4.9. Customizable Page Links Recognizer

Custom page links allow you to define custom Struts page links that will be recognizable in the Struts application diagram. You can define these links by selecting Window > Preferences from the menu bar and then selecting *JBoss Tools* > *Web* > *Struts* > *Customization* from the Preferences dialog box.

#### Struts

😔 Preferences 🗙							
type filter text	Customization		<b>⇔</b> • ⇔+				
👂 Java 💽							
JBoss jBPM	Link Recognizer						
	Tag Attribute I	Refer to Link Type	Add				
JBoss Servers	html:link action	action Struts	Edit				
Packaging Archive	html:link page	page Struts					
⊽ Web	html:link forward	forward Struts	Delete				
Editors	html:frame action	action Struts					
▷ JSF	html:frame_page	page Struts					
▷ Seam	html:frame forward	forward Struts					
	html:form action	action Struts					
Automation	logic:forwa name	forward Struts					
Customizati	logic:redire forward	forward Struts					
Project							
Struts Pages							
Verification							
XDoclet							
IPA							
Plug-in Development							
Run/Debug							
Server							
Spring							
SOL Development							
D Team							
		Restore Defaults	Apply				
⑦ OK Cancel							

Figure 4.51. Customization Panel

# 5

# **JBoss Tools Palette**

The JBoss Tools Palette allows you to:

- Insert tags into a JSP page with one click
- Add custom and 3rd party tags.

The JBoss Tools Palette provides possibility to add any tag libraries to it. Or you can choose a necessary one from the list of already existent tag libraries:

- HTML
- JBoss
- JSF
- JSTL
- MyFaces
- Oracle ADF Faces
- Struts
- XHTML
| Palette Editor |   |              |       |           |
|----------------|---|--------------|-------|-----------|
| ▽ 🗁 XStudio    | Â | name         | value |           |
| 👂 🆏 Icons      |   | element type | group |           |
| ▽ 🚿 Palette    |   | name         | JSF   |           |
| 🕨 🗀 HTML       |   | description  |       |           |
| Þ 🔒 JBoss      | = | hidden       | no    |           |
| 🔻 🚔 JSF        |   |              |       |           |
| 👂 🐸 Core       |   |              |       |           |
| 👂 🐸 Facelets   |   |              |       |           |
| D 🐸 HTML       |   |              |       |           |
| Þ 🔒 JSTL       |   |              |       |           |
| 🕨 🔒 MyFaces    |   |              |       |           |
|                |   |              |       |           |
|                |   |              |       |           |
|                |   |              |       | OK Cancel |

Figure 5.1. Pallete Editor

# 5.1. Using the Palette

#### 5.1.1. JBoss Tools Palette

In the Palette Editor window(Figure 5.1. "Palette Editor") we can see that every group contains its own subgroups. For example, JSF includes Core, Facelets, HTML. So, in the Palette every group has the next view: <Group name> <Subgroup name>.

🧭 JBoss Tools Palette 😫 🦳 🗖
🛠 🗟 🔗
🗁 JBoss Ajax4jsf
➢ JBoss RichFaces
궏 JBoss Seam
SF HTML

Figure 5.2. JBoss Tools Palette

By default the Palette is represented in Web Development Perspective with four groups (Figure 5.2. "JBoss Tools Palette"). If you can't see it, select *Window* > *Show View Other...* > *JBoos Tools Web* > *JBoss Tools Palette* from the menu bar.

By using Show/Hide button you can add any predefined group of tag libraries. It's also possible to create your own group.

## 5.1.2. Inserting Tags into a JSP File

A new tag can be added into any text file including jsp, xhtml # htm(l).

It's very simple to do this. Place the cursor in the JSP page where you want to add a tag and then click the tag in the palette. In the example below, the commandButton tag has been inserted. Notice also that if you place the cursor over any tag, a balloon tip is shown with all the "*tag*" attributes.



Figure 5.3. Inserting Tag

The cursor position after adding a tag into a file is specified by "|" symbol in the tag template on the right in the Palette Editor window.

<b>9</b>	Palette Editor					
マ 🐸 Form	•	name	value			
🛄 button		element type	macro			
checkbox		name	input			
choosing	=	icon				
<ul> <li>fieldset</li> </ul>		large icon				
<ul> <li>file</li> </ul>		description	<html></html>			
<ul> <li>form</li> </ul>		start text	<input t<="" th=""/> <th>ype=" " name=""&gt;</th> <th></th> <th></th>	ype=" " name="">		
hidden		end text				
input		automatically reforma	no			
input button						
input image	_					
	C					
				ОК	Can	cel

#### Figure 5.4. Palette Editor

Above you can see where the cursor position for HTML/Form/input is set. So, after adding this tag into your file the cursor will be in the attribute "type". Then, you can straight use the combination of buttons Ctrl + Space to inquire about a prompting.



Figure 5.5. Cursor position

## 5.1.3. Adding Custom JSF Tags to the JBoss Tools Palette

There are two ways to add any custom or 3rd party tag library to the JBoss Tools Palette:

- Drag-and-drop from the Web Projects view
- The Import button on the JBoss Tools Palette

Before you can add your custom component library, you need to make sure it is included in your project. Either place the ".*tld*" file or the ".*jar*" that includes your tag library under the lib folder in your project.

## 5.1.4. Drag-and-Drop

Switch to the Web Projects view and expand the Tag Libraries folder. If the view not active, select *Window > Show View > Web Projects* from the menu bar.



Figure 5.6. Web Projects View

Also make sure that the JBoss Tools Palette is open. Select the tag library that you want to add and simply dragand-drop it on to the JBoss Tools Palette.

You will see the following dialog window. As you can see JBoss Developer Studio takes care of all the details. You just need to set the Group name to which to add this tag library. You can either add this tag library to an existing Group or just create a new one.

	Import Tags from TLD File 🛛 🗙
TLD File*	/JsfApplication//META-INF/html_basic.tld
Name*	Html
Default Prefix	h
Library URI	http://java.sun.com/jsf/html
Add to Exist	ting Group JSF
<ul> <li>Create New</li> </ul>	Group
	OK Cancel

Figure 5.7. Import Tags From TLD File Form

Once you are finished, you will see the new tag library added to the JBoss Tools Palette.

🧭 JBoss Tools Palet 😫 🦳 🗖
🛠 🗟 🛇
🗁 JSF Core
▷ JSF Facelets
🕞 JSF HTML 🔹 🖈
🐼 JSF HTML taglib
🐼 column
commandButton
e commandLink
🖽 dataTable
🖏 form
📑 graphicImage
🦳 inputHidden
** inputSecret
ab inputText
EB inputTextares

#### Figure 5.8. JBoss Tools Palette with New Tag Library

## 5.1.5. Import Button

The same you can do with *Import* button.

🧭 JBoss Tools Palet 😫 🧧	• 🗆
<b>%</b> 🐻	2
🗁 JSF Core	Import
JSF Facelets	Import
JSF HTML	- *
🐼 JSF HTML taglib	
column	
commandButton	
ee commandLink	
🔠 dataTable	

#### **Figure 5.9. Import Button**

On this screen you can select *Browse* to locate the tag library that you want to add.

<b>9</b>	Import Tags from TLD File 🛛 🗙
Attribute TLD File	e must be set.
TLD File*	° <u>B</u> rowse
Name*	
Default Prefix	
Library URI	
Add to Exist	ting Group
O Create New	/ Group
	OK Cancel

#### Figure 5.10. Import Tags From TLD File Form

Now select the TLD file you want to be added:

۲	Import Tags from TLD File	×
🙆 Attribute Nar	me must be set.	4
	Sedit TLD	×
TLD File*	TLD File*	
Name:*	🗢 🗁 JsfApplication	A
Default Prefix	< sql-1_0-rt.tid	
Library URI	permittedTaglibs.tld	H
	📀 c.tid	
Add to Exi	📀 sql.tld	
O Add to EA	📀 c-1_0-rt.tld	
O Create Net	♦ scriptfree.tld	
	<pre> fmt-1_0-rt.tld</pre>	
	<pre> fmt.tid </pre>	
		-
	OK Cance	ł

Figure 5.11. Select TLD File

# 5.2. Palette Options

There is possibility to configure the JBoss Tools Palette:

- to edit the palette content by adding, removing or changing the palette elements
- to show/hide groups, subgroups
- to import groups, subgroups



Figure 5.12. Palette Buttons

### 5.2.1. Palette Editor

To edit the Pallete view use Edit button. The Palette Editor provides following possibilities:

• to work with set of icons.



#### Figure 5.13. Creating a set of icons

• to edit icons in the chosen set

<b>9</b>	Palette Ec	ditor	×
V Studio	name	value	
マ 🖏 Icons	element type	set of icons	
▶ ‰ C Create	•	Treate Set	
👌 鞼 F 🔝 Copy Set	Ctrl + C	Import Icon	
⊳ 🖏 j: 🛷 Cut Set	Ctrl + X		
▶ ₩L Paste	Ctrl + V		
P 40 S X Delete Set M Struts Common	Delete		
Struts HTML     Struts Logic			
		0	K Cancel

Figure 5.14. Editing icons

• to edit a group

( <b>9</b> )	Palette	Editor		×
マ 🗁 XStudio	name	value		
Icons	element type	partition		
✓ Ø Pate     Create	•	湭 Create Gro	oup	
🕨 🧰 🗈 Paste	Ctrl + V			
Þ 🗎 JSF				_
Þ 🔒 jstl				_
MyFaces				_
👂 🤷 Oracle ADF Faces				
Struts				
A HTML				
			ОК	Cancel

Figure 5.15. Editing a group

• to edit a subgroup

<b>9</b>	Palette Ed	litor	×
▽ 🗁 XStudio	name	value	
Icons	element type	group	
	name	HTML	
Edit Namesp	ace		
▷ 😑 JE		no	
کا 🔁 د	P		
Þ 🚔 J⁵ Edit			
D 🖻 M	CHLLC		
▷ (a) Copy	Ctrl + X		
A General S Contraction Co	Ctrl + V		
▶ 🗎 x			
I Delete	Delete		
Properties			
		ОК	Cancel

Figure 5.16. Editing a subgroup

• to edit a subgroup content

Palette Editor				
<ul> <li>✓</li></ul>	name     element type     name	value sub-group Block		
	icon	http://www.w3.org/TR/REC-html40		
D ≧ Coreate D ≧ Fo D ≧ Fr	e Ctrl + C	Create Macro		
Ø Cut Ø Cut Ø Paste	Ctrl + ) Ctrl + V	no		
Proper	e Delete	2		
		OK Cancel		

Figure 5.17. Editing a subgroup content

• to edit the Palette element or macro

Palette Editor						×	
▼ 🗁 XStudio	Â	name		value			1
👂 🆏 Icons		element type		macro			
	=	name		dd			
ar 🔁 HTML		icon					
⊽ 🐸 Block		large icon					
↔ dd 💦		description		tml>			
div	Ed	dit 🕅	}	d>			
<>> dl	C 🗊	ору	Ctrl +	с			
<>> dt	of C	ut	Ctrl +	X			
⊗li	X D	elete	Delet	e			
<o>&gt; ol</o>				-			
	Pr	operties					
					ок	Cancel	ו

Figure 5.18. Editing the Palette element

Parameters of the Palette element are put into the table on the right. Table cells are provided with editors and modal dialogue windows for choosing necessary icons. For 'start text' and 'end text'(or macro) there is possibility to control the cursor position by using "|" symbol.

**Figure 5.19. Parameters of the Palette element** 

## 5.2.2. Show/Hide

Show/Hide is a very useful feature that allows you to control the number of tag groups that are shown on the palette.

• Click Show/Hide button

🧭 JBoss Tools Palette 🛛 🗖 🗖	
<ul> <li>JBoss Ajax4jsf</li> <li>JBoss RichFaces</li> <li>JBoss Seam</li> <li>JSE HTMI</li> </ul>	Hide
JSF HTML	

Figure 5.20. Show/Hide Button

• In the dialog Show/Hide Drawers check the groups you want to be shown on the palette:

۲	Show/Hide Drawers	×
1	> 🗹 HTML	Show All
1	> 🗹 JBoss	
1	> 🗹 JSF	Hide All
1	> 🗆 JSTL	
1	> 🗹 MyFaces	
1	Oracle ADF Faces	
1	>      Struts	
	XHTML	
_		
	Ok	Cancel

Figure 5.21. Show/Hide Drawers

• Click OK. The new groups will now be displayed on the palette:

🚿 JBoss Tools Palette 🛛		6	- 8
	×	۵,	0
C HTML Block			
HTML Core			
HTML Form			
🗁 HTML Frames			
HTML Scripts			
🗁 HTML Table			
🕞 HTML Text			
🗁 JBoss Ajax4jsf			
🗁 JBoss RichFaces			
🗁 JBoss Seam			
JSF HTML			
MyFaces Extensions			
🗁 MyFaces Sandbox			
🗁 MyFaces Tomahawk			
XHTML List			
XHTML Object			
XHTML Structural			
XHTML Table			
🗁 XHTML Text			
XHTML XForms Basic			

Figure 5.22. New Added Groups

## 5.2.3. Import

The Import button lets you add a custom or 3rd party tag library to JBoss Tools Palette. See here how to add.

# 5.3. Rich Faces Support

JBoss Developer Studio comes with a tight integration with *Rich Faces* component framework. After installing JBDS Rich Faces components as well as *Ajax4jsf* ones are already on the JBoss Tools Palette:



Figure 5.23. Rich Faces Components

# 6

# Web Projects View

Web Projects is a special view that comes with JBoss Developer Studio.

If the Web Projects view's tab is not visible next to the Package Explorer tab, select Window > Show View > Web*Projects* from the menu bar.

# 6.1. Web Projects View

With the Web Projects view, you can:

- Visualize the project better because the project artifacts for JSF and Struts projects are organized and displayed by function.
- Select these kinds of items to drag and drop into JSP pages:
  - JSF managed bean attributes
  - JSF navigation rules outcomes
  - Property file values
  - Tag library files
  - Tags from tag libraries
  - JSP page links
- Use context menus to develop the application (all create and edit functions are available)
- Use icon shortcuts to create and import JSF and Struts projects
- Expand and inspect tag library files
- · Select custom and third-party tag libraries to drag and drop onto the JBoss Tools Palette

# 6.2. Project Organization

The Web Projects view organizes your project in a different way. The physical structure of course stays the same. The new organization combines common project artifacts together which makes it simpler to locate what you are looking for and develop. The screen shot below shows a JSF project and a Struts project in Web Projects view.



Figure 6.1. Web Projects View

# 6.3. Drag and Drop

## 6.3.1. For a Property

Expand the Resources Bundles folder that holds all the Property files in your project. Select the file from which you want to add the property and then select the property.

We will be dragging and dropping a property file value inside the outputText tag for the "value" attribute.

#### Figure 6.2. OutputText Tag

Select the property:



**Figure 6.3. Selecting Property** 

Drag the property and drop it between the quotes for the value attribute in the JSP file. Notice that JBoss Developer Studio added the correctly formatted expression for referring to the property value: #{Message.header} automatically.

```
<html>
<head>
<title>Input User Name Page</title>
</head>
<body>
<f:view>
<hl><h:outputText value="#{Message.header}"/></hl>
<h:messages style="color: red"/>
```

**Figure 6.4. Inserted Property** 

You can actually place the tag anywhere in the page, not just inside an existing tag. In this case, JBoss Developer Studio will place the complete tag **<h:outputText value=''#{Message.header}''/>** in the page.

#### 6.3.2. For Managed Bean Attributes

Select a *"managed bean"* attribute and then drag and drop it onto the JSP page. We are going to place it inside the *"value"* attribute of the inputText tag.



Figure 6.5. Selecting Managed Bean Attrubute

Once again, JBoss Developer Studio adds the correct expression, #{user.name}.

#### Figure 6.6. Added Expression

#### 6.3.3. Navigation Rules

Select the navigation rule under *Configuration* > *faces-config.xml* > *Navigation Rules*:



#### Figure 6.7. Selecting Navigation Rule

Drag and drop it inside the commandButton tag:

```
<f:validateLength maximum="30" minimum="3"/>
</h:inputText>
<h:commandButton action="hello" value="Say Hello!" />
</h:form>
</f:view>
```

#### Figure 6.8. Naviagation Rule in CommandButton Tag

You could do the same if the navigation rule was defined inside an action method:



Figure 6.9. Navigation Rule in Action Method

Here is how it would look after drag and drop:



Figure 6.10. Inserted Navigation Rule

#### 6.3.4. For a Tag Library File Declaration

Select a TLD file:



Figure 6.11. Selecting TLD File

Then drag and drop it onto the JSP page to add a declaration at the top of the page:



Figure 6.12. Inserted TLD File

## 6.3.5. For JSP Pages

You can also drag and drop a JSP page path to a JSP page to create a forward as shown:



Figure 6.13. Creating JSP Forward

# 6.4. Developing the Application

It is also possible to develop your application right from the Web Projects view. Simply right-click any node in the tree and select an appropriate action from the context menu. For instance, this screen capture shows creating a new navigation rule.

🔝 *Web Projects 🛿 🕴	Package Explorer	- 0 )	▲ *faces-config.xml 없
	<u>k</u> (k (š (š	🖻 😫	8
<ul> <li>✓ ➢ Configuration</li> <li>✓ ▲ faces-config</li> <li>▷ ④ Application</li> <li>✓ factory</li> <li>✓ factory</li> <li>✓ lifecycle</li> <li>۞ Component</li> <li>♀ Converters</li> <li>▷ 嗪 Managed B</li> </ul>	xml* hts leans		/pages/inputUserName.jsp hello /pages/hello.jsp
Navigation Reference	New	1=11	P 🔞 Rule
🖳 Render Kits	of Cut	Ctrl + 2	x
🧭 Validators	🗎 Сору	Ctrl + (	
👂 🗁 Beans	💼 Paste	Ctrl + \	/
🕨 🗁 Tiles	💢 Delete	Delet	e
▶ 🗟 web.xml	Properties		
	Verify		

Figure 6.14. Creating New Navigation Rule

# 6.5. Expanding Tag Library Files

You can easily expand any TLD file in the project. Browse to the Tag Libraries folder. Right-click a TLD file and select *Expand*. The TLD file will now be expanded.

You can then select any tag and drag it onto a JSP page.

🔝 Web Projects 😂 📃	٥		
🖹 🖻 😒 🖃 🍕	₽		
▽ 🟦 JSFKickstart	A		
🕨 🧽 WebContent			
🗢 Tag Libraries			
	=		
🗇 c.tld			
<pre> fmt-1_0-rt.tid </pre>			
<pre> fmt-1_0.tid </pre>	Π		
<pre> fmt.tld</pre>			
🗇 fn.tld			
🗢 🗇 html_basic.tld			
Listeners			
Validator			
CommandButton			
commandLink			
dataTable			
I form			
graphicImage			

Figure 6.15. Expanding Tag Library File

# 6.6. Drag and Drop Tag Libraries on to JBoss Tools Palette

Read Adding Tag Libraries to learn about this.

# 6.7. Create and Import JSF and Struts Projects

You can also create and import JSF and Struts project from Web Projects view by selecting the buttons below.

From left to right:

1. Create New JSF Project

- 2. Import JSF Project
- 3. Create New Struts Project
- 4. Import Struts Project



Figure 6.16. Web Projects View Buttons

7

# **Verification and Validation**

As you are developing your project, JBoss Developer Studio Verification constantly provides dynamic validation, consistency checking and error checking. This greatly reduces your development time as it allows you to catch many of the errors during development. JBoss Developer Studio provides dynamic verification for both JSF and Struts projects.

# 7.1. JBoss Developer Studio Verification

## 7.1.1. JSF Project Verification

JBoss Developer Studio checks for many different rules for a JSF project that can be configured by selecting Win-dow > Preferences from the menu bar, selecting JBoss Tools > Web > Verification from the Preferences dialog box and then expanding the JSF Rules node.

<b>9</b>	Preferences	×
type filter text	Verification	<b>⇔</b> ∙ ⇔∗
<ul> <li>✓ JBoss Tools</li> <li>▷ JBoss Servers</li> <li>Packaging Archives</li> <li>✓ Web</li> <li>▷ Editors</li> <li>▷ JSF</li> <li>▷ Seam</li> </ul>	<ul> <li>Rules Configuration Options</li> <li>Verification Level: any</li> <li>JSF Rules</li> <li>Check Faces Config Application Attributes</li> <li>Check Faces Config Component Attributes</li> </ul>	
Seam     Seam     Struts     Verification     XDoclet	<ul> <li>Check Faces Config Converter Attributes</li> <li>Check Faces Config Factory Attributes</li> <li>Check List Entries Attributes</li> <li>Check Managed Bean Attributes</li> </ul>	
<ul> <li>Plug-in Development</li> <li>Run/Debug</li> <li>Server</li> <li>Spring</li> <li>SQL Development</li> </ul>	<ul> <li>Check Managed Property Attributes</li> <li>Check Map Entries Attributes</li> <li>Check Navigation Rules</li> <li>Check Phase Listener Attributes</li> <li>Check Referenced Bean Attributes</li> <li>Check Render Kit Attributes</li> </ul>	
Team TestNG	Check Renderer Attributes  Check Renderer Attri	Tancel

Figure 7.1. JSF Rules

Suppose you are working in the Source viewer for a JSF configuration file as shown below:



Figure 7.2. Faces-config.xml File

While typing a class name, you might make a minor typo (like *"jsfHello.PersonBean9"* instead of *"jsf-Hello.PersonBean"*). After saving the file, verification checks to make sure everything is correct and finds the error below:



**Figure 7.3. Error in Source View** 

Notice that the Package Explorer View shows a marked folder and a marked file where the error is.

You can place the cursor over the line with an error message and get a detailed error message:





Verification also checks navigation rules:



Figure 7.5. Checking Navigation Rules

If you provide a page name that does not exist, verification will let you know about that:



Figure 7.6. Page Name Verification

You can always call up verification explicitly by right-clicking any element in the tree and selecting Verify from the context menu. This works from both the Tree and Diagram viewers for the JSF configuration file editor. You can also invoke verification from the Web Projects view. Below we are checking all of the elements in the configuration file.

🔊 faces-config.xml 🛛		
Faces Config Ed	itor	
- faces-config		- Application
▽ 🙍 faces-config.x	ml	Action-Listener:
Application		i di
🗟 Componer	New	• Hu:
🗟 Converter:	Open Dec	aration <u>"</u>
🗢 층 Managed E	Edit	
🕨 🥔 user	Сору	Ctrl + C
🗢 🍖 Navigation	of Cut	Ctrl + X
🕨 🏇 /pages/i		
🗟 Reference	💢 Delete	Delete
🔄 Render Kits	Properties	
🧭 Validators		
	veni y	
Diagram Tree Source		

Figure 7.7. Verify Command

# 7.1.2. Struts Project Verification

JBoss Developer Studio provides the same functionality for Struts projects. To configure Struts project verification select Window > Preferences from the menu bar, select JBoss Tools > Web > Verification from the Preferences dialog box and then expand the Struts Rules node.



Figure 7.8. Struts Rules

Suppose you are working in the Source viewer for a Struts configuration file as shown below:

```
tName" path="/pages/inputname.jsp"/>
NameForm" path="/greeting" scope="request" type="sample.GreetingAction">
ayHello" path="/pages/greeting.jsp"/>
="org.apache.struts.tiles.TilesPlugin">
perty="moduleAware" value="true"/>
perty="moduleAware" value="true"/>
perty="definitions-config" value="/WEB-INF/tiles1.xml,/WEB-INF/tiles-examples-defs.xml"/>
v
```

Figure 7.9. Struts Configuration File

While typing a class name or entering it from the graphical editor, you might make a minor typo (like *"sample.GreetingAction1"* instead of *"sample.GreetingAction"*). After saving the file, verification checks to make sure everything is correct and finds the error below:



Figure 7.10. Error Reporting

Notice that the Package Explorer View shows a marked folder and a marked file where the error is.

You can place the cursor over the line with the error to view a detailed error message:



Figure 7.11. Error Message

The verification also checks to make sure you have specified the correct JSP page for the forward:


Figure 7.12. JSP Page Verification

Once you place the cursor over the line, you can see the error message:

	<pre>&lt;action name="GetNameForm" path="/greeting" scope="request" type="sample.GreetingAction&lt;/pre&gt;</pre>		
0	Forward sayHello of Action /greeting path attribute reference to non-existent page	Ľ	-

Figure 7.13. Error Message

You can always invoke the verification by switching to the Diagram viewer, right-clicking and selecting *Verify* from the context menu:

🕏 *struts-config.xml 🛿			- 0
↓      ↓     ↓     ↓     ↓     ↓     ↓     ↓	/page s/inputname.jsp		
	Add  Create Comment		
	Paste Ctrl + V	ages/greeting.jsp	
	Select Element Auto-Layout		
	Verify		
	Properties		
	🏅 Generate Java Code		
Diagram Tree Source	Input Methods		

Figure 7.14. Verify Command

# 8

# Editors

JBDS is supplied with a huge range of various editors for different file types.

# 8.1. Editors Features

JBoss Developer Studio has powerful editor features that help you easily navigate within your application and make use of content and code assist no matter what project file (jsp, xhtml, xml, css, etc.) you are working on.

# 8.1.1. OpenOn

OpenOn let's you easily navigate through your project without using the Package Explorer view (project tree). With OpenOn, you can simply click on a reference to another file and that file will be opened.

OpenOn is available for the following files:

- XML files
- JSP/XHTML Pages
- Java files

# 8.1.1.1. XML Files

Press and hold down the Ctrl key. As you move the mouse cursor over different file references in the file, they display an underline. When you have the mouse cursor over the name of the file you want to open, click and the file will open in its own editor. In this example the managed bean NameBean will open.

Sa faces-config.xml 🛛 □
<pre><?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE faces-config PUBLIC "-//Sun Microsystems, Inc.//DTD JavaServer Faces Config</th>     </pre>

Figure 8.1. NameBean Managed Bean

This is the result of using OpenOn



Figure 8.2. NameBean Java Class

You can also try OpenOn with defined attributes.

[	🔬 faces-config.xml 🖾 📃	٥
ſ	<pre><?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE faces-config PUBLIC "-//Sun Microsystems, Inc.//DTD JavaServer Faces Config     "http://iava.sun.com/dtd/web_facesconfig.l.l.dtd">     </pre>	
	<faces-config> <managed-bean></managed-bean></faces-config>	
	<pre><description>User Name Bean</description> <managed-bean-name>user</managed-bean-name></pre>	
	<pre><managed-bean-class>demo.User</managed-bean-class> <managed-bean-scope> <managed-property></managed-property></managed-bean-scope></pre>	
	<property-name>name</property-name> <property-class>12va.lang.String</property-class>	
	<value></value>	

Figure 8.3. OpenOn With Defined Attributes

You can also open any JSP pages.

```
</managed-bean>
<navigation-rule>
<from-view-id>/pages/inputUserName.jsp</from-view-id>
<navigation-case>
<from-outcome>hello</from-outcome>
<to-view-id>/pages/hello.jsp</to-view-id>
</navigation-case>
</navigation-rule>
</faces-config>
```

#### Figure 8.4. JSP Page OpenOn

#### 8.1.1.2. JSP Pages

OpenOn is also very useful in JSP pages. It will allow you to quickly jump to the reference instead of having to hunt around in the project structure.

You can easily open the imported property files.

```
faces-config.xml inputUserName.jsp X
```

#### Figure 8.5. OpenOn With Imported Property Files

Use OpenOn to open a CSS file used with a JSP page:

# Figure 8.6. OpenOn With CSS File

Open managed beans:

```
<h:form id="greetingForm">
<h:outputText value="#{Message.prompt_message}"/>
<h:inputText value="#{user.name}" required="true">
<f:validateLength maximund="30" minimum="3"/>
</h:inputText>
```

Figure 8.7. OpenOn With Maneaged Beans

For JSP files in a JSF project, you can also easily open the navigation rules by applying OpenOn to the JSF tag for the navigation outcome:

Figure 8.8. OpenOn With JSF Tag

# 8.1.2. Code Assist and Dynamic Code Assist (based on project data)

- 8

# 8.1.2.1. Content Assist Features

# 8.1.2.1.1. Content Assist

Content assist is available when working with

- Seam project files
- JSF project files
- Struts project files
- JSP files

# 8.1.2.1.2. JSF Project Files

When working with JSF project in JBoss Developer Studio, you can use various Content Assist features while developing:

- Content Assist for XML, JSP and JSF configuration files
- Content Assist based on project data
- Content Assist with graphical JSF editor

#### 8.1.2.1.2.1. Content Assist for XML, JSP and JSF configuration files

At any point when working with any XML, JSP and JSF configuration files Content Assist is available to help you. Simply type *Ctrl-Space* to see what is available.

Content Assist for JSF configuration file:

😹 *faces-config.xml 🛿	- 8			
<pre><?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE faces-config PUBLIC "-//Sun Microsystems, Inc.//DTD JavaServer Faces Config</td>     </pre>				
application     component     converter     factory     factory     lifecycle     managed-bean     navigation-rule     referenced-bean     render-kit     validator	Element : application Elements The "application" element provides a mechanism to define the various per-application-singleton implementation artifacts for a particular web application that is utilizing JavaServer Faces. For nested elements that are not specified, the JSF implementation must provide a suitable default. sed (such as "/index.jsp" if you are using the default ViewHandler) A proper prefix of a view identifier, plus a trailing "*" character. This pattern			

Figure 8.9. Content Assist in JSF Configuration File

Content Assist for JSF JSP file:

	*faces-config.xml	🗟 inputUserName.jsp 🛛		- 0
	≪@ taglib uri="ht ≪@ taglib uri="ht	:tp://java.sun.com/jsf/co :tp://java.sun.com/jsf/ht	re" prefix="f" %> ml" prefix="h" %>	<u>^</u>
	<f:loadbundle var="&lt;/td"><td>"Message" basename="demo</td><td>.Messages*/&gt;</td><td></td></f:loadbundle>	"Message" basename="demo	.Messages*/>	
	<html> <head> <title>Inp </title></head> <body></body></html>	out User Name Page <td>&gt;</td> <td></td>	>	
	<f:view></f:view>			
,	<ul> <li>f:actionListen</li> <li>f:attribute</li> <li>f:convertDate</li> <li>f:converter</li> <li>f:converter</li> <li>f:convertNum</li> <li>f:facet</li> <li>f:loadBundle</li> <li>f:param</li> <li>f:selectItems</li> </ul>	er :Time :ber	<ul> <li>Register an ActionListener instance on the</li> <li>UIComponent associated with the closest parent UIComponent custom action.</li> </ul>	=
	(	urco		)
VIS	ual/source visual so	urce		

Figure 8.10. Content Assist in JSP File

Content Assist for other JSF XML project files (web.xml shown):



Figure 8.11. Content Assist in web.xml File

#### 8.1.2.1.2.2. Content Assist Based on Project Data

JBoss Developer Studio takes Content Assist to the next level. Studio will constantly scan your project and you will be able to insert code into the JSP page from your project that includes:

- Values from Property files
- "Managed beans" attributes and methods
- Navigation Rule Outcomes
- JSF variables (context, request etc...)

The first screenshot shows how to insert message from a Properties files. You simply put the cursor inside the *"value"* attribute and press *Ctrl-Space*. JBoss Developer Studio will scan your project and show a list of possible values to insert.

Editors



Figure 8.12. Inserting Message

In the following screen shot we are inserting a *"Managed bean"* attribute value. Again, by simply clicking *Ctrl-Space*, JBoss Developer Studio will show a list of all possible values that you can insert:

Once you select a Managed bean, it will show you a list of all available attributes for the selected Managed bean (userBean).

<pre><h:form id="greetingForm">     <h:outputlext #="" pre="" value="#[Mess     &lt;h:inputText value=" wser"<=""></h:outputlext></h:form></pre>	age.prompt_message}"/>	
<f:validatelength maxim<br=""></f:validatelength>	© user.gof © user.name	
<h:commandButton action="ho</th> <th>l user nextPage</th> <th></th>	l user nextPage	
    [4] Visual/Source Visual Source	↓ JSP expression - JSP expression <%−%>	

#### Figure 8.13. Attributes List

Code Assist based on project data will also prompt you for navigation rules that exist in your JSF configuration file.



#### Figure 8.14. Code Assist

#### 8.1.2.1.2.3. Content Assist within Tree JSF Editor

JBoss Developer Studio also provides Content Assist when working within the Tree JSF configuration editor. Just click *Ctrl-Space*.

📾 inputUserName.jsp 🛛 🔞 inde	k.jsp 🔝 *faces-config.xml 🛛 🖳 web.xml	- 0
Faces Config Editor		
▼ faces-config	- Factories	
✓ ▲ faces-config.xml*     ▲ Application	Application-Factory:	Browse
Components	Faces-Context-Factory:	ssibility
Converters	Lifecycle-Factory:	ssibility.internal
Managed Beans Managed Beans	Render-Kit-Factory: # com.sun.acces	ssibility.internal.resources
Seferenced Beans	Lifecycle     Description of the com.sun.corba     Description of the com.sun.corba     Description of the com.sun.corba	.se
🧭 Validators	tom.sun.corba	.se.impl
	com.sun.corba de com.sun.corba	.se.impl.activation .se.impl.copyobject
	# com sun corba	se impl corba
Diagram Tree Source		

Figure 8.15. Content Assist in Tree JSF Configuration Editor

If a field contains right class name and you click a link near the field you will come to the file with this class otherwise a new Java Class dialog will be shown:

	New Java Class	×
<b>Java Class</b> Create a new Java	class.	C
Source folder:	JSF1.2KickStartWithoutLibs/JavaSource	Br <u>o</u> wse
Pac <u>k</u> age:	java.beans	Bro <u>w</u> se
Enclosing type:		Bro <u>w</u> se
Na <u>m</u> e:	Bean	
Modifiers:	public     O default     O private     O protected	
	abstract final static	
<u>S</u> uperclass:	java.lang.Object	Brows <u>e</u>
Interfaces:		<u>A</u> dd
		<u>R</u> emove
Which method stub	s would you like to	
	public static void main(String[] args)	
	<u>Constructors from superclass</u>	
	Inherited abstract methods	
Do you want to add	comments as configured in the properties of the curren	t project?
	Generate comments	
0	Einish	Cancel

# Figure 8.16. New Java Class

If you entered an incorrect name in the field error markers will be shown for field labels and tree items:

📾 hello.jsp 🛛 🗟 web.xml	aces-config.xml ස		- 8				
Faces Config Editor							
	▼ Factories						
🗢 🌇 faces-config.xml*	Application-Factory:	java.beans.BeanDescript	Browse				
Application	Faces-Context-Factory:	com.newjava	Browse				
Converters	Lifecycle-Factory:		Browse				
🗢 ≽ Managed Beans	Render-Kit-Factory:		Browse				
∀ 🥔 user2	✓ Lifecycle						
name     Navigation Bulos	phase-listener						
Referenced Beans			<u>A</u> dd				
Render Kits			<u>R</u> emove				
✓ Validators			<u>E</u> dit				
✓ Extensions			Цр				
			<u>D</u> own				
4							
Diagram Tree Source							

Figure 8.17. Error Markers

- To add a new property to a managed bean expand *Managed Beans* and select <name\_of\_bean>
- Click Add button in the Properties panel
- In the dialog Add Property define a new property. From here also you can generate setters and getters methods:

📾 hello.jsp 🛛 📾 web.xn	nl 🚺 faces-config	g.xmt 🕱	- 8
Faces Config Edito	or		
▼ faces-config	▼ Mana	aged Bean	
🗢 🛃 faces-config.xml	Manage	ed-Bean-Name: user	
Application	<b>@</b>	Add Property	Browse
襑 Components	Property		Diowse
🖗 Converters			<b></b>
🗢 🎯 Managed Bean	·	<u> </u>	
🔻 🥏 user	Property-Name:*	price 💌	
name	Property Class.		
Ravigation Rule	Property-Class:	<u>B</u> rowse	
Referenced Be	Value Kind	value 💌	<u>A</u> dd
Render Kits	Value:	Change	<u>B</u> emove
S Systemsions			Edit
extensions		🗹 Add Java property	
		✓ Generate Getter	Шр
		☑ Generate Setter	Down
		Hnish Cancel	
	Display	-Name:	
	Small-lo	con:	
	Large-le	con:	
4		L	
Diagram Tree Source			

Figure 8.18. Create New Property

Here you can also add an additional navigation rule to the faces-config.xml file.

- Select Navigation Rules
- Click Add button
- In the dialog *Add Rule* define a view and give a name to the rule. Click Finish:

📾 hello.jsp 🛛 🗟 web.xm	🏂 *faces-config.xn	ni 🕱		- 0
Faces Config Edito	r			
▼ faces-config	▼ Navigation	on Rules		
▽ 🔊 faces-config.xml*	from-view-i	d		
Application	3	Add Rule	×	Add
😋 Components	Navigation Rule			<u>R</u> emove
Converters			899	Edit
🗢 🦂 Managed Bean				
ar 🥔 user	From-View-id: /page	es/hello.jsp	Browse	<u>U</u> p
• name	Display-Name			Down
V 🧠 Navigation Rule	bispidy Hame.			
v v /pages/input				
Referenced Be				
Render Kits				
Validators				
faces-config		Finish	Cancel	
		- Filish		
4				
Diagram Tree Source				

Figure 8.19. Add New Navigation Rule

# 8.1.2.1.3. Struts Project Files

# 8.1.2.1.3.1. Content Assist for Struts Configuration File

Editors



Figure 8.20. Struts Content Assist

#### 8.1.2.1.3.2. Content Assist for Struts JSP File



Figure 8.21. Struts JSP Content Assist

# 8.1.2.1.4. JSP Pages

# 8.1.2.1.4.1. Content Assist for JSF Tags

JBDS provides full code completion for JSF tags:

<pre></pre>	Renders an HTML "input" element of "type" "text". Decode Behavior Obtain the Map from the "requestParameterMap" p ExternalContext. If the Map contains an entry for i of the component, pass the value of the entry to i setSubmittedValue() method of the component, w an instance of EditableValueHolder. Encode Behavior Render the clientId of the component as the value "name" attribute. Render the current value of the the value of the "value" attribute. If the "styleClass specified, render its value as the value of the "cla	inputSecret inputText inputTextarea property of the the "clientid" the which must be a of the component as is" attribute is ass" attribute. var=Message			
✓ @ body         ✓ @ f:view         ✓ (rich:calendar population)         ✓ (rich:calendar population)         ✓ (rich:calendar population)         ✓ (rich:calendar population)         ✓ (rich:calendar population)					

Figure 8.22. JSF Tags Content Assist

When the tag is selected the required attributes, if there any, are already inserted and the cursor is located to the first attribute. As this point you can ask for attribute proposals.

	<body< th=""><th>&gt; f:view</th><th></th><th></th><th></th><th></th><th>_</th></body<>	> f:view					_
	<	h:inpu	tText	value="	>		
	~	rich:c	alenda	r popup	© #{Message}		•
	<	/rich:	calenda	ar>	@ #{user}	Ξ	
	<	br>			⑧ #{applicationScope}		
		<n 3<="" th=""><th>&gt; <h:out< th=""><th>tputTex</th><th>@ #{cookie}</th><th></th><th></th></h:out<></th></n>	> <h:out< th=""><th>tputTex</th><th>@ #{cookie}</th><th></th><th></th></h:out<>	tputTex	@ #{cookie}		
۵			<h: out<="" th=""><th>tputTex</th><th><pre>@ #{facesContext}</pre></th><th></th><th></th></h:>	tputTex	<pre>@ #{facesContext}</pre>		
			<h :="" co<="" th=""><th>nmandLi</th><th>@ #{header}</th><th></th><th>-</th></h>	nmandLi	@ #{header}		-
		<th>3&gt;</th> <th></th> <th>⑧ # {headerValues }</th> <th></th> <th></th>	3>		⑧ # {headerValues }		
	<th>/f:vie v&gt;</th> <th>W&gt;</th> <th></th> <th>⑧ # {initParam}</th> <th></th> <th></th>	/f:vie v>	W>		⑧ # {initParam}		
		·			⑧ # {param}		
					⑧ # {paramValues}		
				L	#{requestScope}		
						Ŧ	
	4						
						3	¢
Vi	sual/Source	Visual	Source	Preview			

#### Figure 8.23. Attributes Content Assist

# 8.1.2.1.4.2. Content Assist for JSTL Tags



Figure 8.24. JSTL Tags Content Assist

# 8.1.2.1.4.3. Content Assist for HTML Tags

Content assist for HTML tags has the same mechanism as for JSF tags:



Figure 8.25. HTML Tags Content Assist

You can use as well attributes proposals for HTML tags:



Figure 8.26. HTML Tags Content Assist

# 8.1.2.1.4.4. Content Assist for JavaScript Tags



Figure 8.27. JavaScript Tags Content Assist

# 8.1.2.1.4.5. Content Assist within JSF Configuration Editor

📾 inputUserName.jsp 🛛 📾 inde	x.jsp 💽 *faces-config.xml 🕴 📟 web.xml	- 8
Faces Config Editor		
▼ faces-config	▼ Factories	
▼ 🛃 faces-config.xml*	Application-Factory:	Browse
Application Components	Faces-Context-Factory:	essibility
Converters	Lifecycle-Factory:	essibility.internal
Managed Beans	Render-Kit-Factory:	essibility.internal.resources
Referenced Beans	Lifecycle     Description     Descripti     Descripti     Description     Description     Description	hs so
Validators	ter com.sun.com	ba.se.impl
	⊕ com.sun.com	ba.se.impl.activation ba.se.impl.copyobject
	A com sun cort	Do se impl corba
Diagram Tree Source		

Figure 8.28. Content Assist in JSF Configuration Editor

# 8.1.2.1.5. Content Assist for Rich Faces components

JBDS indeed provides code completion for Rich Faces framework components. All you have to do is to install RichFaceslibrariesintoyourproject.Seehere[ht-

] how to install.



Figure 8.29. Content Assist for Rich Faces Components

- To insert a Rich Faces component on a page expand JBoss Rich Faces group on the palette
- Click on some component
- Put the needed attributes in the Insert Tag dialog and click Finish button

•	Inser	t Tag	(
<rich:calendar> a</rich:calendar>	ittributes		
Attribute name	Value		
jointPoint			
localValueSet			
locale			
monthLabels			-
monthLabelsShort			
optionalFooter			
optionalHeader			
popup	false		
preloadDateRangel	3eg		
preloadDateRangel	End		
rendered			
_		[] [	
0		Einish	Cancel

Figure 8.30. Insert Tag

The Rich Faces tag will be inserted on your page displayed in source and visual modes:



Figure 8.31. Rich Faces Component

# 8.1.2.2. Adding dynamic code assist to custom components that were added to JBoss Tools Palette

Here is what you need to do to add project based code assist to a custom component added in JBoss Developer Studio:

1. Create a new xml file in <*JBDS\_home>studio/eclipse/plugins/org.jboss.tools.common.kb\_\*\*\*/schemas/tld/*. For example call it JeniaFaces.xml. The file should be written according to <*JBDS\_home>/studio/eclipse/plugins/org.jboss.tools.common.kb/kb.jar/org/jboss/tools/common/kb/kb-schema\_1.0.dtd* 

Here is how you set what is available for code assist:

• Adds code assist for JSF pre-defined objects, such as value= "#{param}":

```
<AttributeType ...>
<proposal type="jsfVariables"/>
</AttributeType>
```

• Adds bundle resource (property file) code assist:

• Adds managed bean property code assist:

• Adds managed bean property but with a specified type:

• Adds managed bean method with a signature:

2. Add information on your xml file in <JBDS\_home>/studio/eclipse/plugins/org.jboss.common.kb\_\*\*\*/plugin.xml

```
<tld
jsf="true"
name="Jenia Faces"
schema-location="schemas/tld/myJSF.xml"
uri="http://www.jenia.org/jsf/dataTools"/>
```

3. Restart Eclipse. You should now have code assist for the component.

# 8.1.3. Full Control over Source Files - Synchronized Source and Visual Editing

JBoss Developer Studio offers the flexibility to edit any files in either source or extra visual modes at the same time.

The project is yours and so is the source. JBoss Developer Studio provides you many different graphical editors to speed your application development. At the same time, you always have full control over all project source files. Any changes you make in the source view, will immediately appear in the graphical view.

The JSF configuration file editor has three views: Diagram, Tree and Source. All views are synchronized, you can edit the file in any view.

្នុ*faces-config.xml នេ 🖳 🗖 🗖
/pages/inputUserName.jsp
/pages/hello.jsp
Diagram Tree Source

Figure 8.32. Diagram View

staces-config.xml ⊠				
Faces Config Editor				
	▼ Factories			
<ul> <li>faces-config.xml*</li> <li>Application</li> <li>Components</li> <li>Converters</li> <li>Managed Beans</li> <li>Navigation Rules</li> </ul>	Application-Factory:         Faces-Context-Factory:         Lifecycle-Factory:         Render-Kit-Factory:         V         Lifecycle	Browse         Browse         Browse         Browse         Browse		
ම Referenced Beans බි Render Kits I ✔ Validators	phase-listener	Add Remove Edit Up Down		
Diagram Tree Source				

Figure 8.33. Tree View

🔊 *faces-config.xml 🕴	- 8
xml version="1.0" encoding="UTF-8"?	<u>ڪ</u>
<pre><!DOCTYPE faces-config PUBLIC "-//Sun Microsystems, Inc.//DTD JavaServer Faces Conf      "http://iava_sun_com/dtd/web_facesconfig_l_l_dtd"&gt;</pre>	
<pre>sfaces.config&gt;</pre>	
<managed-bean></managed-bean>	
<pre><description>User Name Bean</description></pre>	
<man -="" aged="" ame="" bean="" n="">u ser</man>	
<man aged-bean-class="">demo.User</man>	
<man aged-bean-scope="">session</man>	
<man -="" aged="" property=""></man>	
<pre><pre>roperty-name</pre></pre>	
<property-class>java.lang.String</property-class>	
<value></value>	
<navigation-rule></navigation-rule>	
<pre><from-view-id>/pages/inputUserName.jsp</from-view-id></pre>	
<navigation-case></navigation-case>	
<from-outcome>hello</from-outcome>	
<to-view-id>/pages/hello.jsp</to-view-id>	
	v
	9
Diagram Tree Source	

Figure 8.34. Source View

The same applies to all other JBoss Developer Studio editors.

Web XML editor is shown. Web XML editor has a graphical view (Tree) and source (Source).

🔊 *faces-config.xml 🔤 web.xml	×				- 8
Web XML Editor					
▼ web	▼ Web Descrip	tor 2.4			*
🗢 💀 web.xml	Name:	web			
Context Params Context Params	ntext Params Display-Name: JSFKicks	JSFKickStart	JSFKickStart		
🕨 🔄 Listeners	Description:			4	
👂 🤤 Servlets		4	III	× •	=
🗑 session-config	✓ Context Params				
a) welcome-file-list	param-name		param-value	<u>A</u> dd	
😂 Error Pages	javax.faces.ST	ATE_SAVING_METHOD	server	<u>R</u> emove	
/ JSP Config				Edit	
Security Constraints				Up	
Security Roles	4			Down	
Carl Environ	- Advanced				•
Tree Source					



🔊 *faces-config.xml 🔛 web.xml 🕴 📃					
xml version="1.0"?	1				
<pre></pre> web-app version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee"					
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://java.sun.com/>					
<display-name>JSFKickStart</display-name>					
<context-param></context-param>					
<pre><param-name>javax.faces.STATE_SAVING_METHOD</param-name></pre>					
<pre><param-value>server</param-value></pre>					
<li>listener&gt;</li>					
<pre><listener-class>com.sun.faces.config.ConfigureListener</listener-class></pre>					
Faces Servlet					
<servlet></servlet>					
<pre><servlet-name>Faces Servlet</servlet-name></pre>					
<pre><servlet-class>javax.faces.webapp.FacesServlet</servlet-class></pre>					
<load-on-startup>1</load-on-startup>					
Faces Servlet Mapping					
<servlet-mapping></servlet-mapping>					
<pre><servlet-name>Faces Servlet</servlet-name></pre>					
<url>pattern&gt;*.jsf</url>					
<login-config></login-config>	<login-config></login-config>				
<auth-method>BASIC</auth-method>					
Tree Source					

# Figure 8.36. Source View

JBoss Developer Studio TLD file editor shown in Tree view. At any point you can edit the source by going switching to Source view.

⊘ html_basic.tid ⊠			- 0		
Red Hat Tag Library Edito	Red Hat Tag Library Editor				
tml_basic	▼ Tag Library		Ĥ		
🗢 🧇 html_basic.tld 📃	Tlibversion:	1.0			
Listeners	Jspversion:	1.2	=		
commandButton	Shortname:	h	н		
🖹 commandLink —	URI:	http://java.sun.com/jsf/html			
🗷 dataTable	Display-Name:				
l form	Creat Lance				
graphicImage	Small-Icon:				
🕗 inputHidden	Large-Icon:				
inputSecret	Description:	This tag library contains JavaServer Faces compone			
🛃 inputText		UlComponent + HTML RenderKit Renderer combina			
			•		
Tree Source					

Figure 8.37. Tree Editor



Figure 8.38. Source Editor

The Struts configuration file editor has three views: Diagram, Tree and Source. All views are synchronized, you can edit the file in any view.

*struts	-config.xml 🕱			
	🔗 getName			
		/pages/inputnam	e.jsp	
			/greeting	
			sayhello	
				/pages/greeting.jsp

Figure 8.39. Struts Diagram View

Source view. Any changes here will immediately appear in all other views.



Figure 8.40. Struts Source View

# 8.2. Visual Page Editor

JBoss Developer Studio comes with a powerful and customizable Visual Page Editor (VPE). You can use the Visual Page Editor to develop an application using any technology: JSF, Struts, JSP, HTML and others.

Current VPE version has four tabs: Visual/Source, Visual, Source and Preview.

Split screen design with instant synchronization between source and visual views:



Figure 8.41. Visual/Source View

You can also switch to pure Visual design:

🗟 CI	🗟 customerinfo.jsp 🕱 🗖 🗖					
•		+	🗘 B Z 🗓 🔏			
×	c:out					
\$ 2	CAR DEMO					
	Please fill i	n your name and address	-			
	Title	#{customer.titleOptions}				
	First Name	#{customer.firstName}	Error Message			
	Error Messages	Middle Initial	#{c			
	Error Message	Last Name	#{customer.lastName}			
		Mailing Address	#{customer.mailingAddres			
		City	#{customer.city}			
		State	AL 💌			
		Zip Code	#{customer.zl;			
	Error Message	Credit Card Number				
	Error Message	Expiry Date	01     2002       f:selectItem       f:selectItem			
		2002 💌				
	Finish Thanks for stopping by!					
html	body f:view	h:form h:panelGrid h:panelG	rid h:outputText			
Visua	Visual/Source Visual Source Preview					

Figure 8.42. Visual View

Or work just in Source view. Note that selection bar is now available not only in Visual mode but also in Source one:



Figure 8.43. Source View

No matter what view you select, you get full integration with Properties and Outline views:


Figure 8.44. Properties And Outline Views

Preview mode is read-only, it only shows how the page will look like in a browser.

### Editors

customerinfo.js	p 🛙	- 8
c:out		
CARE	ОЕМО	50200000000
Please fill in	your name and address.	
Title	#{customer.titleOptions}	
First Name	#{customer.firstName}	Error Message
Error Messages	Middle Initial	#{c
Error Message	Last Name	#{customer.lastName}
	Mailing Address	#{customer.mailingAddres
	City	#{customer.city}
	State	AL 🔽
	Zip Code	#{customer.zip
Error Message	Credit Card Number	
Error Message	Expiry Date	01     2002       f:selectItem       f:selectItem
	2002 💌	
Finish th Insert Conter	anks for stopping by! nt for h:panelGrid	
Visual/Source Vis	ual Source Preview	

## Figure 8.45. Preview Mode

Use the graphical toolbar to add inline styling to any tag.

<pre> <h:commandbutton action="fini &lt;h:graphicImage id=" duke"="" url="/images/duke.gif" value="#{bundle.finishButton}"></h:commandbutton> ()) </pre>	sh* />
	Þ
- ÷ ÷ B I U /	<u>A</u>
× [	
cout and a second secon	
CAR DEMO	
Please fill in your name and address.	
Title #{customer.titleOptions}	
First #{customer.firstName} Error Message	
Error Messages Middle Initial #{c	-
	▶ I
html body f:view h:form h:panelGrid h:panelGrid h:inputText	×
Visual/Source Visual Source Preview	

Figure 8.46. Graphical Toolbar

With just a click or drag-and-drop insert any tags from the palette on to the page you are editing.

🖻 c	customerInfo.jsp 🛙 🗖 🗖	🍯 jBoss To 🐹 🖳 🗖
	<f:selectitem itemlabel="2008" itemvalue="2008"></f:selectitem>	<b>%</b> 🗐 🔍
		C IDece Ainvelief
	An output level values //	Boss RichFaces
		JBoss Seam
		🗁 JSF HTML 🔹 🖈
-	<pre>ch:commandButton value="#{bundle.finishButton}" action="finish" /&gt;</pre>	🐼 JSF HTML taglib
		column
<	<pre>ch:graphicImage id="duke" url="/images/duke.gif" /&gt;</pre>	commandButton
	droutoutToyt, values"#fhundie buuleball" (>	commandLink
	an:ourputiext value= #{oundle.buyLabel} />	🔢 dataTable
		5! form
	H	📓 graphicImage
	2 (h = 4 fa = ran-	inputHidden
		It inputSecret
-		ab inputText
aj		inputTextarea
$\cap$	Error Credit Cand Number	1 message
-	Message	messages
		OutputFormat
		/ output abel
	Error Expiry Date fselectItem	T output ink
	Message	
	fiselectItem	
	2002 -	
		SelectBeeleenChe
	Thanks for stanning by	SelectBooleanche
	Prinsi de Tritarias foi stopping by:	BesteetManyCrieck
	Insert Content for n:panelGrid	
		■ selectManyMenu
htm	al body f:view h:form h:commandButton 🔀	selectOneListbox
Visi	ual/Source Visual Source Preview	selectOneMenu
		<ul> <li>selectOneRadio</li> </ul>

Figure 8.47. Inserting Tags From Palette

## 8.2.1.1. Advanced Settings

There are three buttons on the Visual Page Editor left side:

	<h <h <h <!--</th--><th><pre>:commandButton value="#{bundle.finishButton}" action="finish" /&gt; :graphicImage id="duke" url="/images/duke.gif" /&gt; :outputText value="#{bundle.buyLabel}" /&gt; h:panelGrid&gt; </pre></th><th></th></h </h </h 	<pre>:commandButton value="#{bundle.finishButton}" action="finish" /&gt; :graphicImage id="duke" url="/images/duke.gif" /&gt; :outputText value="#{bundle.buyLabel}" /&gt; h:panelGrid&gt; </pre>	
	4		
•	ſ	🗘 🗘 🗘 B I U 📈 🗛	
a.	f		ŋ
		c:out	1
\$		CAR DEMO	
L	-	Please fill in your name and address.	
		Title #(customer titleOntions)	
		First #{customer.firstName} Error Message	
		Name	
		Error	4
		Middle Initial #{c	1
htr	nl	hody fiview hiform highthert	ļ
na			
Vis	ua	I/Source Visual Source Preview	

Figure 8.48. Visual Page Editor Buttons

• Preferences

Provides quick access to Visual Page Editor preferences.

<b>9</b>	Preferences	(Filtered)
type filter text	Visual Page Editor	() = () =
✓ JBoss Tools ✓ Web ✓ Editors Visual Page Editor	General Templates	Show Border for Unknown Tags   Show Resource Bundles Usage as EL Expressions   Always Prompt for Tag Attributes During Tag Insert   Show Selection Tag Bar   Always Hide Selection Bar Without Prompt     Visual/Source
0		OK Cancel

### Figure 8.49. Visual Page Editor Preferences Window

• Refresh

Refresh displaying information with this button.

• Page Design Options

This button leads to page design options.

9	References t	o Resources
Page De	sign Options	
Actual Run	-Time Absolute Folder	
Path		<u>C</u> hange
Scope:	Page	•
Actual Run	-Time Relative Folder	
Path		
Scope:	Page	
Scope	CSS File Path	Add
		Edit
		Remove
Scope	URI	Prefix Add
		Edit
		Remove
		Ok Cancel

#### Figure 8.50. Page Design Options

This dialog let's you set resources which are usually only resolved in runtime. To set a stylesheet, click *Add* (for CSS File Path section) and add your stylesheet. It works when CSS is defined on your page in the following way:

Code:

<link rel="stylesheet" type="text/css" href="#{facesContext.externalContext.requestContextPath}/style.cs</pre>

This will work fine in runtime, but the Visual Page Editor doesn't know what requestContextPath in design time is.

The next section (URI), let's you add URI taglibs if you are using includes so that the editor knows where to find the tag libraries.

The first two sections let you define actual runtime folders. Here is an example.

Let's say you have the following project structure:

```
WebContent/
pages/
img/
a.gif
header.jsp
main.jsp
```

header.jsp content:

```
My Header
<img src="img/a.gif"/>
```

main.jsp:

```
<jsp:include page="pages/header.jsp" />
```

When you open *main.jsp* in Visual Page Editor, it will not be able to resolve the image from the header, however, it will work fine in runtime. To fix this in design time, click the *Page Design Options* button and set *Actual Run-Time Relative* Folder to *<project>WebContent > pages* and you will see the image appear.

#### • Hide Selection bar

By clicking on the component in Visual view or selecting a code snippet in Source mode you can see the tags tree. If you want to hide the selection bar, use the "Hide Selection Bar" button on the lower right side.

•	🗦 🗘 🗘 🕹 🕹 🕹	
×	c:out	<b>^</b>
₩	CAR DEMO	=
	Please fill in your name and address.	
	Title #{customer.titleOptions} <	
	First #{customer.firstName} Error Message	
	Error Middle Initial #{c	
	Messages	Hide Selection Bar 💌
html	body f:view h:form h:panelGrid h:panelGrid h:inputText	
Visu	al/Source Visual Source Preview	

Figure 8.51. Selection Bar

# 8.2.2. Setup notes for Linux

### 8.2.2.1. How to Start the Visual Page Editor under Linux

Linux users may need to do the following to get the Visual Editor to work correctly on their machines.

The Visual Page Editor requires the library libstdc++.so.5. This library is contained in the compat-libstdc++-33.i386 package.

• To install this package on Fedora Core or Red Hat Enterprise Linux run the following command:

yum install compat-libstdc++-33.i386

• On any other rpm based distributions download libstdc++.so.5 and run the following command:

rpm -Uvh compat-libstdc++-33.i386

• On Debian based distributives run the following command:

apt-get install compat-libstdc++-33.i386

In case you have the library installed and you still have issue with starting the visual page editor then close all browser views/editors and leave one visual page editor open and restart eclipse. This should force a load of the right XULRunner viewer.

## 8.2.3. JSP syntax validation

When working in JBoss Tools JSP editor you are constantly provided with feedback and contextual error checking as you type.

## 8.2.4. JSP Page Preview

JBoss Developer Studio comes with JSP design-time preview features. When designing JSP pages you can easily preview how they will look during runtime. You can even attach your stylesheet to the preview.

JSP preview is available for:

- Struts Pages
- JSF Pages

The preview features are available with Visual Page Editor.

# 8.3. More Editors

Besides Visual Page Editor JBDS provides editors for editing project files of any types: properties, TLD, web.xml, tiles, and so on.

# 8.3.1. Graphical Properties Editor

The Properties editor allows you to work in two different modes and also supports unicode characters.

To create a new properties file, in the Package Explorer view, select New > Properties File from the right-click context menu on the folder where you want to create the file.



Figure 8.52. Selecting Properties File

You can edit the file using a table-oriented "Properties" viewer:

🖻 Messages.properties 🛿		- 0
name	value	Add
header	Hello Demo Application	
prompt_message	Name:	Edit
hello_message	Hello	Delete
		Up
		Do <u>w</u> n
Properties Source		

## Figure 8.53. "Properties" Viewer

You can also use a Source viewer for editing the file:

■ Messages.properties 🕴	- 8
header=Hello Demo Application	A
hello message=Hello	
	=
	v
Properties Source	

#### Figure 8.54. Source Viewer

# 8.3.2. Graphical TLD Editor

The TLD editor comes with same features you will find in all other JBoss Developer Studio editors:

- Graphical and source edit modes
- Validation and error checking

### 8.3.2.1. Tree view

🛇 html_basic.tld ଅ			- 8
Red Hat Tag Library Edi	itor		
▼ html_basic	▼ Tag Library		Â
🗢 🧇 html_basic.tld 🖉	Tlibversion:	1.0	
Listeners	Jspversion:	1.2	
CommandButto	Shortname:	h	=
CommandLink	URI:	http://iava.sun.com/isf/html	
📕 dataTable	Display-Name		
🖹 form	Display Name.		
🖹 graphicImage	Small-Icon:		
🖹 inputHidden	Large-Icon:		
inputSecret	Description:	This tag library contains JavaServer Faces cor	
linputText		UIComponent + HTML RenderKit Renderer cc 🔽	
🧾 inputTextarea			
🖹 message	<ul> <li>Defined</li> </ul>		
🖹 messages	name	tagclass Add	
🖲 outputFormat	commandButto	on com sun faces taglib htr	
🔊 outputLabel	commandLink	com.sun.faces.taglib.htr	
In outputt ink	dataTable	com.sun.faces.taglib.htr	•
Tree Source			



### 8.3.2.2. Source view

You can easily switch from Tree to Source by selecting the Source tab at the bottom of the editor.

```
- 8
📀 html_basic.tld 🔀
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE taglib
PUBLIC "-//Sun Microsystems, Inc.//DTD JSP Tag Library 1.2//EN"
"http://java.sun.com/dtd/web-jsptaglibrary 1 2.dtd">
<taglib>
<tlib-version>1.0</tlib-version>
<jsp-version>1.2</jsp-version>
<short-name>h</short-name>
<uri>http://java.sun.com/jsf/html</uri>
<description>
  This tag library contains JavaServer Faces component tags for all
  UIComponent + HTML RenderKit Renderer combinations defined in the
  JavaServer Faces Specification.
</description>
<!-- =========== Tag Library Validator =========== -->
<validator>
  <validator-class>
    com.sun.faces.taglib.html_basic.HtmlBasicValidator
  </validator-class>
</validator>
٠
4
Tree Source
```



You can easily add a new tag:

oss loois lag Library	Editor	
MyTLD	▼ Tag Library	
MyTLD Hd Rename	Tlibversion: 1.1	
Va New	Validator myTags	
Ch Ch	Listener http://java.sun.com/jsp/jstl/xml1	
Properties	Tag e: My Tag Lib	$\neg$
🕨 🖹 if	Tag File	⊣
👂 퇸 forEach		$\dashv$
🔊 otherwise	Large-Icon:	
👂 🧾 param	Description: My Tag Library	<b>^</b>
👂 🖹 parse		-
👂 📕 set		
👂 🔝 transform	✓ Defined	
👂 🔝 when	name tagclass Add	
Functions	choose org.apache.taglibs.standard.tag.common.core.C	
	out org.apache.taglibs.standard.tag.rt.xml.ExprTag	
	if org.apache.taglibs.standard.tag.common.xml.lf	
	forEach_org.apache.taglibs.standard.tag.common.xml.Fc	

Figure 8.57. Adding a New Tag

You can also easily add a new attribute to an existing tag:

eee loois lag cibrai	,	
MyTLD	▼ Tag	
7 🔷 MyTLD.tld	Name:	if
📕 Listeners 🗹 Validator	Tagclass:	org.apache.taglibs.standard.tag.commor
膨 choose	Teiclass:	<u>B</u> rowse
👂 🖹 out	Bodycontent:	JSP
Add Variable     Add Attribute     Add Attribute As	Fragment	
<ul> <li>p</li> <li>p</li> <li>p</li> <li>P</li> <li>Paste</li> </ul>	Ctrl + Ctrl +	C L conditional tag, which evalutes its body if the supplied XPath expression evalutes to 'true' as a
🕨 🥂 ti 💢 Delete	Dele	ete
Properties		
💛 Verify		
	👻 Tag	

Figure 8.58. Adding a New Attribute

Content assist is available when editing the file using the Source viewer:

Ø*	MyTLD.tid 🛿 🔗 html_basic.tid				- 0	
<	<pre>?xml version="1.0" encoding="U</pre>	TF-8" ?>			<u> </u>	
<	<pre>taglib xmlns="http://java.sun. xmlns:xsi="http://www.w3.or xsi:schemaLocation="http:// version="2.0"&gt; <description>My Tag Library<!--<br--><display-name>My Tag Lib<tlib-version>1.1<short-name>myTags</short-name></tlib-version></display-name></description></pre>	<pre>com/xml/ns/j2ee" rg/2001/XMLSchema-ins java.sun.com/xml/ns/ description&gt; play-name&gt; on&gt; we&gt;</pre>	ta j2	nce" ee http://java.sun.com/xml/ns/j2e		
	<> description		^	Element : short-name		d h a 100
	<> display-name			authoring tool to create names with a mi	be used	value; for
	<> function			example, it may be used as the preferre	d prefix	value in taglib
	<> icon			directives. Do not use white space, and or underscore.	lo not st	tart with digits
	<> listener		=			
	short-name			Data Type : string		
	⇔ tag					
	<> tag-file					
	taglib-extension					
	<> tlib-version					
	⇔uri , , , , , , , , , , , , , , , , , , ,		•	L		
		III			D	
Tree	Source					

Figure 8.59. Content Assist

In the Source viewer, if at any point a tag is incorrect or incomplete, an error will be indicated next to the line and also in the Problems view below.



**Figure 8.60. Error Reporting** 

# 8.3.3. Graphical Web Application File (web.xml) Editor

The Web Application File editor comes with the same features you will find in all other JBoss Developer Studio editors:

- Graphical and source edit modes
- Validation and error checking

### 8.3.3.1. Tree View

🔤 web.xml 🕴					
Web XML Editor					
▼ web	- Servlet			<b>^</b>	
マ 🗟 web.xml	Servlet-Name:	Faces Servlet			
Context Params	Servlet-Class:	javax.faces.webapp.Faces	sSer\ <u>B</u> rowse		
va Filters ▼ A Listeners	Load-on-Startup:	1			
do com.sun.faces.config.Configur	✓ Init Params				
▽ 😂 Servlets	param-name	param-value	<u>A</u> dd		
Faces Servlet:javax.faces.web			Remove	=	
session-config			Edit		
A Mime Mappings					
la welcome-file-list					
a Error Pages			Down		
/ JSP Config	▼ Security Roles	5			
login-config	role-name	role-link	<u>A</u> dd		
Security Roles			<u>R</u> emove		
😂 Env Entries			Edit		
Carl EJB					
Services			Down		
Resources	- Advanced		<u>Powi</u>		
Tree Source	Auvanceu			U	
Jource Jource					



You can add any new elements right in the Tree viewer:

web	✓ Web Descriptor 2.4	
We   New   Rename   Change Time Stamp   Copy   Ctrl   Copy   Ctrl   Copy   Ctrl   Copy   Ctrl   Copy   Ctrl   Paste   Properties   JSP Config   Security Constraints   Security Roles   Security Roles   Env Entries   EJB   Services   Resources   Message Destinations   Iccale-encoding-mapping-list	<ul> <li>Context Param</li> <li>Filter</li> <li>Filter Mapping</li> <li>Listener</li> <li>Servlet Mapping</li> <li>Mime Mapping</li> <li>Mime Mapping</li> <li>Frror Page</li> <li>Security Constraint</li> <li>Security Role</li> <li>Security Role</li> <li>Env Entry</li> <li>Ejb Ref</li> <li>Ejb Local Ref</li> <li>Service Ref</li> <li>Resource Env Ref</li> <li>Resource Ref</li> <li>Message Destination Ref</li> <li>Message Destination</li> </ul>	Add Add Edit Up Down Change Change
	Encoding:	-

Figure 8.62. Adding New Elements

### 8.3.3.2. Source View

Switch to the Source viewer to edit the web.xml file by hand at any time:





Figure 8.63. Source View

### 8.3.3.3. Content Assist

Content assist is available in the Source viewer. Simply click CTRL-Space anywhere in the file.

📟 web.xml 🕱		- 8
<pre><?xml version="1.0"?> <web-app <br="" version="2.4" xmlns="http://ja" xmlns:xsi="http://www.w3.org/2001/XMLSG &lt;display-name&gt;JSFProject&lt;/display-name&gt; &lt;context-param&gt; &lt;param-name&gt;javax.faces.STATE_SAVING_I &lt;param-value&gt;server&lt;/param-value&gt; &lt;/context-param&gt;&lt;/pre&gt;&lt;/td&gt;&lt;td&gt;va.sun.com/xml/ns/j2ee">chema-instance" xsi:schemaLocation="http://java.sun.c &gt; METHOD</web-app></pre>	×	
<ul> <li>context-param</li> <li>description</li> <li>display-name</li> <li>distributable</li> <li>ejb-local-ref</li> <li>ejb-ref</li> <li>env-entry</li> <li>error-page</li> <li>filter</li> <li>filter-mapping</li> </ul>	<ul> <li>Element : context-param</li> <li>The context-param element contains the declaration of a web application's servlet context initialization parameters.</li> <li>Content Model : (description*, param-name, param-value)</li> </ul>	
 		J
Tree Source		J

Figure 8.64. Content Assist

## 8.3.3.4. Errors Checking and Validation

If errors occur anywhere in the file, small red dots will appear next to the lines where the errors occurred. Also, note that the file is marked by a small x in the Package Explorer view.



Figure 8.65. Errors Reporting

## 8.3.4. Graphical Tiles Files Editor

### 8.3.4.1. Graphical Editor For Tiles Files

The Tiles configuration file editor has three main viewers (modes): Tree (shown), Diagram and Source. The modes can be selected via the tabs at the bottom of the editor. Any changes made in one mode are immediately visible when you switch to any other mode.

When working in Source view, you always have all following features available:

- Content Assist
- Open On Selection

### 8.3.4.2. Create New Tiles File

To create a new Tiles files, right click any folder and select New > Tiles Files :



Figure 8.66. Creating New Tiles File

### 8.3.4.3. Tree View

In the Tree mode, the different elements of the Tiles file are organized into functional categories on the left-hand side and a form for editing the properties of currently selected items on the right-hand side.

🗄 tiles-defs.xml 🕴				- 0	
Tiles Editor					
✓ tiles-defs	▼ Tiles Cor	nfig Description		<u>^</u>	
🗢 📓 tiles-defs.xml	Name: tiles-defs				
tiles.layout	Encoding				
tiles.layout.weivgroup	Encoung:	L			
tiles.layout.weivitems	- Definitio	ons			
▽ 🔚 tiles.layout.error	name	extends	path	Add	
🔳 title	tiles.layout	1	/jsp/main.jsp	Remove	
action	action tiles.layout.weiv: tiles.layout				
chooser	tiles.layout.weivi tiles.layout				
🔚 tiles.head	tiles.layout.error tiles.layout				
🔚 tiles.chooser	tiles.head		/tiles/header.jsp	Down	
🔚 tiles.action	tiles.choos	er	/tiles/chooser.jsp	Down	
🔚 tiles.foot	tiles.action	1	/tiles/action.jsp		
🔚 tiles.weivgroup	tiles.foot		/tiles/foot.jsp		
🔚 tiles.weivitems	tiles.weivg	roup	/tiles/WeivGroup.jsp		
🔚 tiles.groups	tiles.weivit	ems	/tiles/WeivItems.jsp		
🔚 tiles.items	tiles.group	s	/tiles/Groups.jsp		
🔚 tiles.error	tiles.items		/tiles/items.jsp	_	
<	tiles.error		/tiles/Error.jsp	-	
Tree Diagram Source					

Figure 8.67. Tree View

To edit the file, simply right click any node and select among the available actions:

<ul> <li>tiles.layout</li> <li>tiles.layout.weivgroup</li> <li>tiles.layout.weivitems</li> </ul>	Exte Path	nds	tiles.layout	
V lilec lought organ		rollerClass		
Add PutList		rollerUrl		
Rename		/anced		
🔚 tile 👔 Copy 📿	Ctrl + C			
🔚 tile 👘 Paste 🛛 🖓	Ctrl + V	ay-Name:		
🔚 tile 🔀 Delete	Delete	ription:		
🔚 tile Properties				
🔚 tiles.weivitems 🔚 tiles.groups	Sma	II-Icon:		
🔚 tiles.items	Large-lcon:			
🔚 tiles.error	• Deprecated			
	Page	/jsp/	main.jsp	
Tree Diagram Source				

Figure 8.68. Editing in Tiles Editor

## 8.3.4.4. Diagram View

The Diagram mode is shown below:



Figure 8.69. Diagram View

To create new definition, simply right click anywhere in the diagram:

New Definition	
of Cut	Ctrl + X
📄 Сору	Ctrl + C
💼 Paste	Ctrl + V
🗱 Delete	Delete
Preferences	
Input Methods	,
	New Definition Cut Copy Paste X Delete Preferences Input Methods

Figure 8.70. Creating New Definition

You can also use the Diagram toolbar to make editing easier:



Figure 8.71. Diagram Toolbar

### 8.3.4.5. Source

The Tiles editor also comes with a Source view that gives you full control over the source. Any changes here will immediately appear in when you switch to any of the other viewers.





### Figure 8.72. Source View

Content assist is available in the Source mode.

	<pre><tiles-definitions> <definition name="tiles.layout" path="/j &lt;put name=" title"="" value="Main page"></definition> <put name="header" value="tiles.head"></put> <put <put="" name="footer" value="tiles.foot"></put> </tiles-definitions></pre>	jsp/main.jsp"> > er"/> '/>		
	O definition	Element : definition	yout">	=
	# comment - xml comment	The "definition" element describes a definition that can be inserted in a isp page. This definition		
	# XSL processing instruction - XSL processing in	is identified by its logical name. A definition		
		allows to define all the attributes that can be set		
		In tag from a jsp page. controllerClass The fully qualified lava class name of the controller	yout">	
		subclass to call immediately before the tiles is		
		inserted. Only one of controllerClass or		
		The context-relative path to the resource used as	•>	
		controller called immediately before the tiles is		Н
		inserted. Only one of controllerClass or		
	<pre><definition name="tiles.head" path="/til&lt;/pre&gt;&lt;/td&gt;&lt;td&gt;les/header.jsp"></definition></pre>			
	<pre><definition name="tiles.chooser" path="//&lt;/pre&gt;&lt;/td&gt;&lt;td&gt;&lt;pre&gt;/tiles/chooser.jsp" tiles.action"=""></definition> tiles/cation_isp"/&gt;</pre>			
	<pre><definition =="" name="tites.action" path="//&lt;/pre"></definition></pre>	cites/action.jsp />		⊡
Tre	e Diagram Source		•	
	biagram source			

Figure 8.73. Content Assist



Any errors are immediately reported as shown below:

Figure 8.74. Error Reporting

You can also use the Outline view together with the editor's Source mode. Selecting any node in the Outline view will jump to that place in the source:



Figure 8.75. Outline View

# 8.3.5. Graphical Editor for Struts Validation Files

JBoss Developer Studio comes with a visual validation editor. To create a new validation file, right click any folder and select *File* > *Validation File* from the context menu.

Ne <u>w</u>	Þ	📑 Project
Go Into		🖶 Package
Open in <u>N</u> ew Window		G Class
Ope <u>n</u> Type Hierarchy	F4	😯 Interface
Sho <u>w</u> In	Shift+Alt+W	😂 Source Folder
Conv	Ctrl+C	😂 Folder
Copy Qualified Name	curre	📑 File
Paste	Ctrl+V	Struts Config
X Delete	Delete	Arces Config
	20000	🖹 Tiles File
Build Path	•	💑 Validation File
<u>S</u> ource	Shift+Alt+S	💸 TLD File
Refactor	Shift+Alt+T 🕨	📸 JSP File
and Import		🚵 XHTML File
A Export		📸 HTML File
Exp <u>o</u> rt		🚳 CSS File
🔘 Run XDoclet	Shift+Ctrl+F1	🖄 JS File
🔗 Refresh	F5	📸 Properties File
Clo <u>s</u> e Project Close <u>U</u> nrelated Projects		📑 E <u>x</u> ample
Assign Working Sets		📑 <u>O</u> ther

Figure 8.76. Creating Validation File

The validation editor works through a number of viewers.

The Formsets viewer shows forms and their elements for which to define validation rules:

🗹 *MyValidation.xml 🛙				- 8
Formsets	Page	1		Edit
	Indexed List Prope			Edit
マ 💖 formset (default)	Msg - Corresponde	ed Message Template		
💖 Constants	name	key	resource	Add
AlertWizard				C dit
dashboardManagerForm				Eaic
Ø Ø DashboardWizardNew				Delete
▽ 🥏 dashboardViewForm	Arg - Replacement	t Value for Message Ten	nplate	
▽ 🧿 name	name	arg key	resource	Add
19 required	required	arg0 Name	false	Edit
👺 mask				Delete
	-Var - Validator Para	ameter		
	var-name	var-value		Add
	mask	^[()A-Za-z0-9()_]4	+[A-Za-z0-9']	Edit
				Delete
Formsets Validators Constants Tro	ee Source			

Figure 8.77. Formsets Viewer

The Constants viewer lets you set constant values for your validation rules:

MyValidation.xml 8	3						- 0	
Current Global Sectio	Current Global Section globa ¢ 👔							
constant-name		constant-va	alue				<u>A</u> dd	
standartName		^[()A-Za-z	:0-9_]+[A-Za-z	20-9' _]*[()A-	Za-z0-9_]*\$		Edit	
							Delete	
Formsets Validators	Constants Tre	ee Source						

Figure 8.78. Constants viewer

The validation file also can be viewed in a Tree viewer:

MyValidation.xml 🕴	- C
🗢 🍯 MyValidation.xml	
🎲 global	
🗢 🎲 formset (default)	
🕨 🥔 AlertWizard	
🕨 🥔 dashboardManagerForm	
🕨 🥔 DashboardWizardNew	
ashboardViewForm 🤣	
🗢 🥥 name	
💖 mask	
arg0 for required	
Formsets Validators Constants Tree Source	

### Figure 8.79. Tree Viewer

At any point you have full control over the source by switching to the Source viewer. Any editing in this viewer will immediately be available in the other viewers of this editor.

Editors



Figure 8.80. Source Viewer

You can also open your own custom or Struts-standard validation-rules.xml file.

The Validators viewer shows the validation rules for a selected validator. You can of course add your own rules.

MyValidation.xml 🕴		- 0
Current Global Section globa 🗢 📑 🍵		
Validators	Depends C	Change
name	Message Key errors.timeornumber	Change
long	Java Class Name com echopass provisioning rra validator TimeOrNumbe	Change
float	Method validateTimeOrNumber	hange
double	Method Param java Jang Object. org apache commons V	Change
intRange	JavaScript	, nange
creditCard	Function Name	Change
email	Function Body	-
url	var timeMask = "^([0-9]){1,3}(:[0-5][0-9]){0,2}\$";	
time_numberforthresholds	var booleanMask = "^ostivelinactive\$";	_
	var anymask = "^"\$ ; var percentMask = "^[0-9]+.?[0-9]?%?\$";;	
	var statistic = new Array();	~
Formsate Validators Constants Tr		Þ

Figure 8.81. Validation Rules

Here are the validation rules shown in the Source viewer.



Figure 8.82. Validation Rules in Source Viewer

# 8.3.6. Spring IDE

JBoss Developer Studio bundles a Spring Framework editor from Spring IDE for Eclipse [ht-tp://springide.org/project]. Visit this site for the latest versions and documentation.

You can add a Spring Project Nature to an existing project by right-clicking on the project and selecting *Adding Spring Project Nature* from the context menu:
🚦 Package I	Ex 🛿 🔝 Web Project 🗖 🗖	
	수수 🗟 🖻 🕏 🎽	
▼ 🐸 JSFKI Ø Ja	Ne <u>w</u> Go Into	•
Þ <b>⇒</b> jR Þ <b>⇒</b> Al Þ <b>⇔</b> ar	Open in <u>N</u> ew Window Ope <u>n</u> Type Hierarchy Sho <u>w</u> In	F4 Shift+Alt+W ▸
Þ ⊜ w [	Copy Copy Qualified Name	Ctrl+C
<u>د ا</u>	Paste <u>Paste</u> <u>Delete</u>	Ctrl+V Delete
Properti Property	<u>B</u> uild Path <u>S</u> ource Refactor	► Shift+Alt+S Shift+Alt+T
deriv edita	≧n Import ≧ Export	
linke	Run XDoclet Refresh Close Project Assign Working Sets	Shift+Ctrl+F1 F5
	<u>R</u> un As Debug As Profile As Validate	* * *
	Add Spring Project <u>N</u> ature T <u>e</u> am	•
	Comp <u>a</u> re With Restore from Local Histor <u>y</u> JBoss Tools PDE To <u>o</u> ls Iava EE	•
-	Properties	Alt+Enter

#### Figure 8.83. Adding Spring Project Nature

If you need to remove it you should select Remove Spring Project Nature from the context menu of your project.

Once the Nature is added, a Spring project will be decorated with a small "S" in the Package Explorer view.



Figure 8.84. Project with Spring Nature

To add a Spring Configuration File with bean definitions open Properties dialog from the context menu of your project. Then select *Spring > Beans Support* on the left side.

Properties for JSFKickStart							
type filter text	Beans Support	<b>⇔</b> ∙ ⇔≁					
Resource BeanInfo Path Builders	Config Files Config Sets Maintain Spring beans config files.						
FreeMarker Context Hibernate Settings J2EE Module Depend Java Build Path Java Code Style	Config file e <u>x</u> tensions xml	Add Remove					
<ul> <li>Java Compiler</li> <li>Java Editor</li> <li>Javadoc Location</li> <li>JSP Fragment</li> <li>Packaging Archives</li> <li>Profile Compliance a</li> <li>Project Facets</li> <li>Project References</li> <li>Run/Debug Settings</li> <li>Seam Settings</li> <li>Server</li> </ul>							
<ul> <li>✓ Spring</li> <li>Beans Support</li> <li>Web Flow Support</li> <li>Targeted Runtimes</li> <li>Task Tags</li> <li>TestNG</li> <li>Validation</li> <li>Web Content Setting:</li> </ul>	OK	Cancel					

Figure 8.85. Spring Beans

۲		Properties for JSFKickStart	×
type filter to	ext	Beans Support	<b>⇔</b> • ⇔•
Resourc Beaninfo	e 🗾	Config Files Config Sets Maintain Spring beans config files.	
FreeMark Hibernat	e Settings	Config file e <u>x</u> tensions xml	Add
Java Bui Dava Co Dava Co	d Path de Style mpiler		Remove
Java Edit Javadoc JSP Fragi Packagin	tor Location ment		
Profile C Project F Project F Run/Deb	ompliance a acets eferences ug Settings		
Server Server Spring Beans Web f	s Support Flow Support		
Targeted Task Tag TestNG Validatio	n		
(* III	► F	ОК	Cancel

#### Figure 8.86. Adding Configuration Files

The added file will also be marked by an "S" in the Package Explorer view.

To activate the Spring Explorer view, select Window > Show View... > Other and then Spring > Spring Explorer.

۲	Show View	×			
-					
ty	pe filter text				
Þ	🗁 JPA Development				
Þ	🗁 PDE				
Þ	🗁 PDE Runtime				
Þ	🗁 Seam				
Þ	Server				
~	🔻 🗁 Spring				
	🞇 Beans Cross References				
	輵 Spring AOP Event Trace				
	🔎 Spring Explorer	=			
Þ	🗁 SQL Development				
Þ	🗁 Team	•			
	OK Cancel				

Figure 8.87. Selecting Spring Beans View

This view shows a read-only outline view of the Spring Bean Configuration File.

#### 8.3.7. CSS Editor

The CSS editor comes with the same features you will find in all other JBoss Developer Studio editors.

- Content assist
- Validation and error checking

With the CSS (Cascading Style Sheet) editor, you can take advantage of code prompting:

Editors

\*stylesheet.css X



Figure 8.88. CSS Editor

And you can also use the Properties view next to the editor to edit existing stylesheet declaration properties:



Figure 8.89. Properties View

#### 8.3.8. JavaScript Editor

The JavaScript editor includes a Preview viewer and a Source viewer. In the Source viewer, you can use code assist: Editors



Figure 8.90. JavaScript Editor

You can also use the Source viewer with the Outline view to navigate around the file:



Figure 8.91. Source Viewer

#### 8.3.9. XSD Editor

JBoss Developer Studio comes with an XSD Editor for XML Schema files. This editor comes from the Web ToolsProject(WTP)(seeWTPGettingStarted[ht-tp://www.eclipse.org/webtools/testtutorials/gettingstarted/GettingStarted.html]).

To create a new XSD file, right-click a folder in the Package Explorer view, select New > Other... from the context menu and then select XML > XML Schema in the dialog box.

9	New X
Select a wizard	
Create a new XML schema file	
<u>W</u> izards:	
type filter text	
👂 🗁 Plug-in Development	A
👂 🗁 Red Hat Developer Studio	
👂 🗁 Server	
👂 🗁 Web	
Web Services	
🗢 🗁 XML	
📴 DTD File	
A WSDL	
XML	
S XML Schema	≡
👂 🗁 Other	
Examples	
< Back	Next > Bnish Cancel

Figure 8.92. Creating New XSD file

The XSD Editor includes two viewers for working on the file, a Design viewer and a Source viewer:

#### Editors



Figure 8.93. Source Viewer

In the Design viewer, you can drill down on an element by double-clicking on it:

S cmicm.xsd 🛿				- 8
e arg	*	🖪 (argT	ype)	
	Ø		string	1
	00	— (8) id	idType	
	0	Our line line line line line line line line	namespaceRefType	
	ø	IctRef	namespaceRefType	
	ø	a ref	refType	
		Iname	string	
		③ dataType	namespaceRefType	
		差 atom	(atomType)	
		🛃 scalar	(scalarType)	
		🕖 array	(arrayType)	
		🛃 matrix	(matrixType)	
		🖉 expression	(expressionType)	
				_
Design Source				
Design Source				

#### Figure 8.94. Design Viewer

Various edit options are available when you right-click an element in the diagram:



#### Figure 8.95. Edit Options

You can also use the Properties view to edit a selected element:

		③ name	string				
		③ dataType	namespaceRefType				
		📌 atom	(atomType)				
		🛃 scalar	(scalarType)				
		🛃 array	(arrayType)				
	0*	🕖 matrix	(matrixType)				
		🕫 expression	(expressionType)				
Problems Tasks Servers E Properties 🛛							
attribut	te						
Name: ref							
Type: reft	reflyne						
.,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	<u> </u>						
Usage:							
	rvers Propo attribut Name: ref Type: ref	rvers Properties X attribute Name: ref Type: refType	Image: Second state   Image: Type: Second state				

Figure 8.96. Properties View

You can also use a Source viewer for the file. In this viewer, along with direct editing of the source code, you can also edit the file by using the Properties view on the right:

S cmlcm.xsd 🛛	- 8	Properties 🛙	~ - 8
<pre>="" class="description"/&gt; itation&gt; in&gt; ip&gt;<xsd:attributegroup id="attGp.min" name="min"> ip&gt;<xsd:attributegroup id="attGp.min" name="min"> i="att.min" name="min" type="minType"&gt; i&gt; cation&gt; ="" class="summary"/&gt; ="" class="description"/&gt; itation&gt; on&gt; </xsd:attributegroup></xsd:attributegroup></pre>	=	General Documentation Extensions	attriroup
<pre>ip&gt;<xsd:attributegroup id="attGp.ref" name="ref"> 'att.ref" name="ref" type="refType"&gt; 'att.ref" name="ref" type="refType"/ref" type="reftype"/reftype"/ref" t</xsd:attributegroup></pre>			

Figure 8.97. Source Viewer

## 8.3.10. Support for XML Schema

JBoss Developer Studio fully supports XML files based on schemas as well as DTDs:

WIE	web.xml	23

	🦞 web.xml 🕴 🗖							
	xml</th <th>ver</th> <th>sion="1.0"?&gt;</th> <th><u>ڪ</u></th>	ver	sion="1.0"?>	<u>ڪ</u>				
	<web-< th=""><th>app</th><th><pre>version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee"</pre></th><th></th></web-<>	app	<pre>version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee"</pre>					
	xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://							
	<des< th=""><th>crip</th><th>tion&gt;Facelets StarterKit</th><th></th></des<>	crip	tion>Facelets StarterKit					
	<dis< th=""><th>play</th><th>-name&gt;myFaces</th><th></th></dis<>	play	-name>myFaces					
	<con< th=""><th>text</th><th>-param&gt;</th><th></th></con<>	text	-param>					
	<pa< th=""><th>ram-</th><th>name&gt;javax.faces.DEFAULT_SUFFIX</th><th></th></pa<>	ram-	name>javax.faces.DEFAULT_SUFFIX					
	<pa< th=""><th>ram-</th><th>value&gt;.xhtml</th><th></th></pa<>	ram-	value>.xhtml					
	<th>ntex</th> <th>t-param&gt;</th> <th></th>	ntex	t-param>					
	<con< th=""><th>text</th><th>-param&gt;</th><th>=</th></con<>	text	-param>	=				
	<pa< th=""><th>ram-</th><th>name&gt;facelets.REFRESH_PERIOD</th><th>-</th></pa<>	ram-	name>facelets.REFRESH_PERIOD	-				
	<pa< th=""><th>ram-</th><th>value&gt;2</th><th></th></pa<>	ram-	value>2					
	<th>ntex</th> <th>t-param&gt;</th> <th></th>	ntex	t-param>					
	<con< th=""><th>text</th><th>-param&gt;</th><th></th></con<>	text	-param>					
	<pa< th=""><th>ram-</th><th>name&gt;facelets.DEVELOPMENT</th><th></th></pa<>	ram-	name>facelets.DEVELOPMENT					
	<pa< th=""><th>ram-</th><th>value&gt;true</th><th></th></pa<>	ram-	value>true					
	<th>ntex</th> <th>t-param&gt;</th> <th></th>	ntex	t-param>					
	<con< th=""><th>text</th><th>-param&gt;</th><th></th></con<>	text	-param>					
	<pa< th=""><th>ram-</th><th>name&gt;javax.faces.STATE_SAVING_METHOD</th><th></th></pa<>	ram-	name>javax.faces.STATE_SAVING_METHOD					
	<pa< th=""><th>ram-</th><th>value&gt;client</th><th>H</th></pa<>	ram-	value>client	H				
	<th>ntex</th> <th>t-param&gt;</th> <th></th>	ntex	t-param>					
	<con< th=""><th>text</th><th>-param&gt;</th><th></th></con<>	text	-param>					
	<pa< th=""><th>ram-</th><th>name&gt;com.sun.faces.validateXml</th><th></th></pa<>	ram-	name>com.sun.faces.validateXml					
	<pa< th=""><th>ram-</th><th>value&gt;true</th><th></th></pa<>	ram-	value>true					
	<th>ntex</th> <th>t-param&gt;</th> <th></th>	ntex	t-param>					
	<con< th=""><th>text</th><th>-param&gt;</th><th></th></con<>	text	-param>					
	<pa< th=""><th>ram-</th><th>name&gt;com.sun.faces.verifyObjects</th><th></th></pa<>	ram-	name>com.sun.faces.verifyObjects					
	<param-value>true</param-value>							
	<th>ntex</th> <th>t-param&gt;</th> <th>•</th>	ntex	t-param>	•				
	4							
Tre	e Sou	irce						

Figure 8.98. XML File

# 9

# **JBoss Tools Preferences**

Configuring the various JBoss Developer Studio features is done via the Preferences screen by selecting Window > Preferences > JBoss Tools from the menu bar.

۲	Preferences							
ty	pe filter text		JBoss Tools	(⇒ + ⇒ +				
Þ	General	*	JBossTools preferences					
Þ	Ant							
Þ	Connectivity							
	FreeMarker Editor							
Þ	Help							
	HQL editor							
Þ	Install/Update							
Þ	Internet							
Þ	Java							
Þ	JBoss jBPM							
~	JBoss Tools	=						
-	JBoss Servers							
	Packaging Archives							
	▷ Web							
	XDoclet							
	JPA							
₽	Plug-in Development							
₽	Run/Debug							
₽	Server							
⊳	Spring							
⊳	SQL Development							
⊳	Team							
	TestNG		Restore Defaults	Apply				
	Validation	•	Nestore Deradits	мрру				
0	D		ОК	Cancel				

Figure 9.1. Preferences are included in this dialog.

From this screen, you can select these more specific sets of JBoss Tools preferences:

- Code Assist
- Editors

- JBoss Servers
- JSF
- JSF Flow Diagram
- JSF Page
- JSF Propject
- Packaging Archives
- Plug-in Insets
- Resource Insets
- Seam
- Seam Validator
- Struts
- Struts Automatic
- Struts Customization
- Struts Flow Diagram
- Struts Pages
- Struts Project
- Struts Support
- Title Diagram
- Verification
- View
- Visual Page Editor
- XDoclet
- XDoclet Templets
- XDoclets Variables

## 9.1. CodeAssist

Select *JBoss Tools* > *XDoclet* > *Code Assist* to see Code Assist preference page.

Here is what the Code Assist preference page looks like:

۲			Preferences	×
ty	pe filter text		Code Assist	<b>⇔</b> ∙ ⇔⇒
Þ	General	-	Allows to refresh the XTags from the XDoclet modules	
⊳	Ant		Befresh XDoclet Data	
₽	Connectivity		Neiresii Abociet bata	
	FreeMarker Editor			
⊳	Help			
	HQL editor			
⊳	Install/Update			
⊳	Internet			
⊳	Java			
⊳	JBoss jBPM	=		
~	JBoss Tools			
	JBoss Servers			
	Packaging Archives			
	◊ Web			
	Code Assist			
	JPA			
⊳	Plug-in Development			
⊳	Run/Debug	Н		
₽	Server			
⊳	Spring			
⊳	SQL Development			
⊳	Team	•	Restore Defaults	Apply
4		)		EPP19
¢	D		ОК	Cancel

Figure 9.2. CodeAssist

## 9.2. Editors

You can set various preferences for the editors that JBoss Developer Studio adds to the Eclipse environment by se-

lectind *JBoss Tools* > *Web* > *Editors*.

۲		Preferences	×
ty	pe filter text	Editors	<b>⇔</b> ∙ ⇔⇒
Þ	General	A	
⊳	Ant	Always use JBoss Tools editors with Open opt	on
⊳	Connectivity	Show warning when project has no Red Hat of	apabilities
	FreeMarker Editor	Use Source tab as a default for mutli-tab edit	ors
⊳	Help		
	HQL editor		
⊳	Install/Update		
⊳	Internet		
⊳	Java	=	
⊳	JBoss jBPM		
$\nabla$	JBoss Tools		
	JBoss Servers		
	Packaging Archives		
	⊽ Web		
	Editors		
	▷ JSF		
	Seam		
	Struts		
	Verification		
	XDoclet		
	JPA		
⊳	Plug-in Development		
⊳	Run/Debug	Restore Defaults	Apply
4			
0	D	ОК	Cancel

Figure 9.3. Editors

In the initial Editors screen, you can decide on some global settings for JBoss Tools editors. You can select whether an available JBoss Tools editor should always be the default editor for a type of file, whether the user should be warned that making a project an Red Hat project will make an JBoss Tools editor fully available for a particular type of file, and whether, for JBoss Tools editors, the Source mode should be the default instead of a visual mode.

## 9.3. JBoss Servers

The following preferences can be changed on the *JBoss Tools* > *JBoss Servers* preference page.

Here is what the JBoss Servers preference page looks like:

٠	Preferences	×
type filter text	JBoss Servers	<b>⇔</b> • ⇔∗
type filter text <ul> <li>General</li> <li>Ant</li> <li>Connectivity</li> <li>FreeMarker Editor</li> <li>Help</li> <li>HQL editor</li> <li>Install/Update</li> <li>Internet</li> <li>Java</li> <li>JBoss jBPM</li> <li>JBoss Tools</li> <li>JBoss Tools</li> <li>JBoss Servers</li> <li>Packaging Archives</li> <li>Packaging Archives</li> <li>Veb</li> <li>Editors</li> <li>JSF</li> <li>Seam</li> <li>Struts</li> <li>Verification</li> <li>XDoclet</li> <li>JPA</li> <li>Plug-in Development</li> <li>Run/Debug</li> </ul>	Servers Servers Server Timeouts Start Timeout 450 Stop Timeout 450 Upon Timeout: Abort Server Transition Set Server State to Target State Startup Poller MX Poller \$ Shutdown Poller Process Terminated Poller \$ Shutdown Poller Process Terminated Poller \$	
0	ОК	Cancel

Figure 9.4. JBoss Servers

# 9.4. JSF

Select *JBoss Tools* > *Web* > *JSF* to get to the JSF Project specific preferences.

9	Preferences	×
type filter text	JSF	<b>⇔</b> • ⇔∗
♦ General	JSF Projects specific preferences.	
▶ Ant		
Connectivity		
FreeMarker Editor		
▶ Help		
HQL editor		
Install/Update		
▶ Internet		
⊅ Java		
▷ JBoss jBPM		
JBoss Servers		
Packaging Archives		
⊽ Web		
Editors		
▶ JSF		
Seam		
Struts		
Verification		
XDoclet		
JPA		
Plug-in Development		
▶ Run/Debug		
0	ок	Cancel



## 9.5. JSF Flow Diagram

Selecting JBoss Tools > Web > Editors > JSF Flow Diagram allows you to specify some aspects of the Diagram mode of the JSF configuration file editor.

۲				Preferences X
ty	pe filter text		JSF Flow Diagram	n 🗘 🗸 🗘 🗸
Þ	General Ant	<u>^</u>	JSF Flow Diagram	Add View
Þ	Connectivity			Show Grid
	FreeMarker Editor		Grid Step:	16
Þ	Help			
	HQL editor		Link Path Font:	default,size=8,style=1
Þ	Install/Update		View Path Font:	default,size=8,style=1
Þ	Internet	=		Do not create a navigation rule for a view that has no navigation case
Þ	Java			Switch to standard central mode for surger after transition is made
Þ	JBoss jBPM			Switch to standard control mode for cursor after transition is made
~	JBoss Tools			Show shortcut icon
	JBoss Servers			Show shortcut path
	Packaging Archive	es l		
	✓ Web			
	✓ Editors			
	JSF Flow Dia	gram		
	Struts Flow L	magram		
	Visual Page I	Editor		
	Þ ISF			
	Seam			
	Struts			
	Verification	-		Parter Defente
4				Restore Deraults Apply
(	D			OK Cancel

Figure 9.6. JSF Flow Diagram

The first two items control the background grid for the diagram. The next two items allow you to control the appearance of the labels for views (pages) and the transitions between views. For these two items, clicking the *Change...* button allows you to assign a font with a dialog box.

The first check box determines whether a view in the diagram that doesn't have a transition connecting it to another view yet should be written to the source code as a partial navigation rule. The next check box determines whether the diagram cursor reverts immediately to the standard selection mode after it's used in the transition-drawing mode to draw a transition. Finally, the last two check boxes concern shortcuts. A shortcut is a transition that is there but isn't actually displayed in the diagram as going all the way to the target view it's connected to, in order to make the

diagram clearer. With the check boxes, you can decide whether to display a small shortcut icon as part of the shortcut and also whether to display the target view as a label or not.

<b>19</b>	Preferences	×
type filter text	JSF Flow Diagram	<b>⇔</b> • ⇔•
Ø General	ISE Flow Diagram Add View	
▶ Ant		
Connectivity	Page Template: JSFBasePage	-
FreeMarker Editor	Extension: isp	<b>_</b>
▷ Help	[]	
HQL editor		
Install/Update		
Internet		
⊅ Java		
JBoss jBPM		
JBoss Servers		
Packaging Archives		
⊽ Web		
JSF Flow Diagram		
Struts Flow Diagram		
Tiles Diagram		
Visual Page Editor		
▷ JSF		
Seam		
Struts		
Verification	Restore Defaults	Apply
٢	ОК	Cancel

Figure 9.7. Add View

Selecting the Add Page tab in the JSF Flow Diagram screen allows you to determine the default template and file extension for views (pages) you add directly into the diagram using a context menu or the view-adding mode of the diagram cursor.

## 9.6. JSF Page

By selecting JBoss Tools > Web > JSF >> JSF Pages you can add jsf pages or remove existing ones.

<b>9</b>	Preferences	×
type filter text	JSF Pages	<b>⇔</b> ▼ ⇔⇒
▶ General ▶ Ant	Blank [default]	Add
Connectivity FreeMarker Editor	JSPRedirect	Remove
Help HQL editor		
▷ Install/Update ▷ Internet		
Þ Java Þ JBoss jBPM		
<ul> <li>✓ JBoss Tools</li> <li>▷ JBoss Servers</li> </ul>		
Packaging Archives Veb		
▷ Editors ✓ JSF		
JSF Pages Project		
<ul> <li>Seam</li> <li>Struts</li> </ul>		
Verification		
JPA		
Run/Debug		
▶ Spring		Set Default
0	ОК	Cancel

Figure 9.8. JSF Page

# 9.7. JSF Propject

Select *JBoss Tools* > *Web* > *JSF* > *Project* to see JSF Project preference page.

On *Project* panel you define a template for a new created project: servlet version, page template and so on.

9	Preferences	×
type filter text	Project	⇔• ⇔∗
General	New Project Import Project	
▶ Ant		
Connectivity	Version:	÷
FreeMarker Editor	Project Template:	
▶ Help		
HQL editor	Use Default Path	
Install/Update	Projects Root: /opt/work/workspace_beta	Change
Internet		
▷ Java	Serviet Version: 2.4	
JBoss jBPM	Register Web Context in	server.xml
	Page Template:	
JBoss Servers		
Packaging Archives		
⊽ Web		
Editors		
⊽ JSF		
JSF Pages		
Project		
Seam		
Struts		
Verification		
XDoclet		
JPA		
Plug-in Development		
Run/Debug		
Server		
Spring	Restore Defaul	ts Apply
0	ОК	Cancel

#### Figure 9.9. JSF Propject

Selecting the Import Project tab in the JSF Project screen allows you to determine the default servlet version and whether to register Web Context in server.xml.

	Preferences	×
type filter text	Project 💝	• \$*
General	New Project Import Project	
▶ Ant		
Connectivity	Servlet Version: 2.4	-
FreeMarker Editor	Register Web Context in server.xml	
▶ Help		
HQL editor		
Install/Update		
Internet		
▷ Java	=	
JBoss jBPM		
JBoss Servers		
Packaging Archives		
∀ Web		
Editors		
⊽ JSF		
JSF Pages		
Project		
Seam		
Struts		
Verification		
XDoclet		
JPA	Restore Defaults Ar	vlac
Plug-in Development		
0	OK Car	ncel

Figure 9.10. Import JSF Propject

# 9.8. Packaging Archives

The following preferences can be changed on the *JBoss Tools* > *Packaging Archives* page.

On Packaging Archives panel you determine settings for Project Packages view and core preferences.

9	Preferences	×
type filter text	Packaging Archives	<b>⇔</b> • ⇔•
<ul> <li>▷ General</li> <li>▷ Ant</li> <li>▷ Connectivity</li> <li>FreeMarker Editor</li> <li>▷ Help</li> <li>HQL editor</li> <li>▷ Install/Update</li> <li>▷ Internet</li> <li>▷ Java</li> <li>▷ JBoss jBPM</li> <li>▽ JBoss Tools</li> </ul>	Core Preferences   Enable incremental builder  Project Packages View  Show full output path next to packages.  Show the full root directory of filesets.  Show project at the root  Show all projects that contain packages	
<ul> <li>JBoss Servers</li> <li>Packaging Archives</li> <li>Web</li> <li>XDoclet</li> <li>JPA</li> <li>Plug-in Development</li> <li>Run/Debug</li> <li>Server</li> <li>Spring</li> <li>SQL Development</li> <li>Team</li> <li>TestNG</li> <li>Validation</li> <li>Web and XML</li> <li>Web Services</li> <li>XDoclet</li> </ul>	Restore Defaults	Apply
0	ОК	Cancel

Figure 9.11. Packaging Archives

# 9.9. Plug-in Insets

By selecting *Web* > *Struts* > *Automation* > *Plug-in Insets* on tab Tiles you can define a default text for tiles plugin.

<b>9</b>	Preferences	×
type filter text	Packaging Archives	<b>⇔</b> • ⇔⇒
<ul> <li>General</li> <li>Ant</li> <li>Connectivity</li> <li>FreeMarker Editor</li> <li>Help</li> <li>HQL editor</li> <li>Install/Update</li> <li>Internet</li> <li>Java</li> <li>IBoss iBPM</li> </ul>	Core Preferences Enable incremental builder  Project Packages View  Show full output path next to packages.  Show the full root directory of filesets.  Show project at the root  Show all projects that contain packages	
<ul> <li>✓ JBoss Tools</li> <li>▷ JBoss Servers</li> <li>Packaging Archives</li> <li>▷ Web</li> </ul>		
<ul> <li>XDoclet</li> <li>JPA</li> <li>Plug-in Development</li> </ul>		
<ul> <li>▶ Run/Debug</li> <li>▶ Server</li> <li>▶ Spring</li> </ul>		
<ul> <li>SQL Development</li> <li>Team</li> <li>TestNG</li> <li>Validation</li> </ul>		
<ul> <li>Web and XML</li> <li>Web Services</li> <li>XDoclet</li> </ul>	Restore Defaults	Арріу
0	ОК	Cancel

#### Figure 9.12. Plug-in Insets

The same is done but for validator plugin on the tab Validators.

Figure 9.13. Plug-in Insets of Validators

## 9.10. Resource Insets

To see Resource Insets preference page select *JBoss Tools* > *Web* > *Strats* > *Automation* > *Resource Insets*.

On Resource Insets panel you determine default error messages for error resource files.



Figure 9.14. Resource Insets

## 9.11. Seam

The following preferences can be changed on the *JBoss Tools* > *Web* > *Seam* page.

On Seam screen you can add and remove Seam runtimes.

Here is what Seam preference page looks like:

۲					Preferen	ces		×
type filter text			s	ear	<b>⇔</b> ▼ ⇔⇒			
⊳	General	*						
Þ	Ant				Name	Versio	Path	Add
Þ	Connectivity			•	Seam 1.2.AP	1.2	/opt/work/rhds_rc	Edit
	FreeMarker Editor							
⊳	Help							Remove
	HQL editor							
Þ	Install/Update							
Þ	Internet							
⊳	Java	Ξ						
Þ	JBoss jBPM							
$\nabla$	JBoss Tools							
	JBoss Servers							
	Packaging Archive							
	⊽ Web							
	Editors							
	▷ JSF							
	Seam							
	Struts							
	Verification							
	XDoclet							
	JPA							
Þ	Plug-in Development							
Þ	Run/Debug	•					Restore Defau	Its Apply
4								
G	D						ОК	Cancel

#### Figure 9.15. Seam

## 9.12. Seam Validator

The following preferences can be changed on the *JBoss Tools* > *Web* > *Seam* > *Validator* page.

In Validator panel you configure seam problems that will be processed by validator.

<b>9</b>		Preferences	×		
type filter text		Validator	<b>⇔</b> • ⇔•		
General	-	Configure Project Specific			
▶ Ant		Select the severity level for the following			
Connectivity		optional Seam Validator problems:			
FreeMarker Editor		← Components			
▶ Help		Duplicate component name:	Error 🗘		
HQL editor		Stateful component does not contain @Remove method:	Error 😫		
Install/Update		Stateful component does not contain @Destroy method:	Error \$		
Internet					
⊅ Java	=	Stateful component has wrong scope:	Error		
JBoss jBPM		Component class name cannot be resolved to a type:	Error 🗘		
		Component class does not contain setter for property:	Error \$		
JBoss Servers		• Entities			
Packaging Archives		Component life-cycle methods:			
⊽ Web		Factories			
Editors		Bilections			
▷ JSF		Context variables			
▽ Seam					
Validator		r Expression language			
Struts					
Verification					
XDoclet					
JPA					
Plug-in Development	•	Postore Defaulte	Apply		
	Ð	Restore <u>D</u> eraults	Арріу		
0		ОК	Cancel		



## 9.13. Struts

By selecting JBoss Tools > Web > Struts you can configure Struts projects specific preferences.

<b>9</b>	Preferences	×
type filter text	Struts	<b>⇔</b> • ⇔•
Deneral	Struts Projects specific preferences.	
▶ Ant		
Connectivity		
FreeMarker Editor		
▷ Help		
HQL editor		
Install/Update		
Internet		
⊅ Java		
▷ JBoss jBPM		
JBoss Servers		
Packaging Archives		
⊽ Web		
Editors		
▷ JSF		
▷ Seam		
Struts		
Verification		
XDoclet		
JPA		
Plug-in Development		
Run/Debug		
0	ОК	Cancel



## 9.14. Struts Automatic

On *Automation* panel you can modify default text for the Tilel Struts plug-in element, the Validator Struts plug-in element, and error message resource files.
<b>9</b>	Preferences	×
type filter text	Automation	<b>⊳</b> • ⇔•
Deneral	Here you can modify default text for the Tiles Struts plug-in element,	
▷ Ant	the Validator Struts plug-in element, and error message resource file	5.
Connectivity		
FreeMarker Editor		
▷ Help		
HQL editor		
Install/Update		
▶ Internet =		
▷ Java		
▷ JBoss jBPM		
JBoss Servers		
Packaging Archives		
∀ Web		
▶ Editors		
Þ jsF		
Seam		
Automation		
Customization		
Project		
Struts Pages		
Verification		
0	ОК Са	ancel

Figure 9.18. Struts Automatic

## 9.15. Struts Customization

The following preferences can be changed on the *JBoss Tools* > *Web* > *Struts* > *Customization* page.

In the Customization screen you configure Link Recognizer for Struts tags.

<b>9</b>		Pre	eferences			×
type filter text		Customization	n			<b>⇔</b> • ⇔-
General	-	Link Recognize	r			
▶ Ant		Tag	Attribute	Refer to	Link Type	Add
Connectivity		http://ink	action	action	Starte	Add
FreeMarker Editor		http://ink	nacuon	nacuon	Strute	Edit
▶ Help		http://ink	forward	forward	Strute	Delete
HQL editor		http://frame.	action	action	Strute	
Install/Update		html:frame	action	action	Struts	
Internet	=	html:frame	page	page	Struts	
▷ Java		html:frame	forward	forward	Struts	
JBoss jBPM		html:form	action	action	Struts	
		logic:forward	name	forward	Struts	
JBoss Servers		logic:redirect	forward	forward	Struts	
Packaging Archives						
▽ Web						
Editors						
▷ JSF						
Seam						
Automation						
Customization						
Project						
Struits Pages						
Verification						
Venication	•				Restore Defaults	Apply
0					ОК	Cancel

**Figure 9.19. Struts Customization** 

#### 9.16. Struts Flow Diagram

Similarly to the JSF Flow Diagram screen, selecting *JBoss Tools* > *Web* > *Editor* > *Struts Flow Diagram* page allows you to specify aspects of the Diagram mode of the Struts configuration file editor. The Struts Flow Diagram screen adds an option to hide the Diagram tab and labeling settings for additional artifacts.

<b>9</b>	Prefe	ences	×
type filter text	Struts Flow Diag	ram	<b>⇔</b> • ⇔∗
D General	Struts Flow Diagram	n Add Page	
▶ Ant			
Connectivity		Do not show diagram tab	
FreeMarker Editor		Show grid	
▶ Help	Grid Sten	16	<b></b>
HQL editor	ond step.		
▷ Install/Update	Action Font:	default,size=8,style=1	<u>C</u> hange
▶ Internet =	Forward Font:	default size=7 style=1	Change
▷ Java	rorward ronc.		<u>c</u> nange
JBoss jBPM	Path Font:	default,size=8,style=1	<u>C</u> hange
	Comment Font	default size=10 style=0	Change
JBoss Servers	Comment Point.		<u>c</u> nange
Packaging Archives		Switch to control mode after transition is	s made
∀ Web		Show shortcut icon	
✓ Editors     ✓		Show shortcut path	
JSF Flow Diagram			
Struts Flow Diagram			
Tiles Diagram			
Visual Page Editor			
Þ js⊧			
Seam			
Struts			
Verification		Restere Defaults	Annh
		Restore Defaults	Арріу
٢		ОК	Cancel

Figure 9.20. Struts Flow Diagram

Selecting the Add Page tab in the Struts Flow Diagram screen allows you to determine the default template and file extension for views (pages) you add directly into the diagram using a context menu or the view-adding mode of the diagram cursor.

<b>8</b>	Preferences	×
type filter text	Struts Flow Diagram	<b>⇔</b> + ⇔+
General	Struts Flow Diagram Add Page	
▷ Ant		
Connectivity	Page Template: StrutsForm	•
FreeMarker Editor	Extension: isp	<b></b>
▷ Help	[]] Extension:	
HQL editor		
Install/Update		
Internet	_	
▷ Java		
▷ JBoss jBPM		
JBoss Servers		
Packaging Archives		
∀ Web		
JSF Flow Diagram		
Struts Flow Diagram		
Tiles Diagram		
Visual Page Editor		
Þ JSF		
Seam		
Struts		
Verification	Restore Defaults	Apply
▷ XDoclet		
0	ок	Cancel

Figure 9.21. Adding Page

## 9.17. Struts Pages

You can change the following preferences on the JBoss Tools > Web > Struts > Struts Pages preference page.

On Struts Pages panel you can add or remove Struts pages.

type filter text     Struts Pages       > General     Blank [default]       > Ant     StrutsForm	r ⇔ * 1 Ive
<ul> <li>◊ General</li> <li>◊ Ant</li> <li>Blank [default]</li> <li>Add</li> <li>StrutsForm</li> </ul>	t ve
▷ Ant StrutsForm	ve
StrutsForm	ve
Connectivity Remo	
FreeMarker Editor	
▷ Help	
HQL editor	
Install/Update	
▶ Internet =	
D Java	
▷ JBoss jBPM	
✓ JBoss Tools	
▷ JBoss Servers	
Packaging Archives	
∀ Web	
▷ Editors	
▷ JSF	
D Seam	
⊽ Struts	
Automation	
Customization	
▷ Project	
Struts Pages	
Verification Set Der	fault
OK   Cancel	el

Figure 9.22. Struts Pages

### 9.18. Struts Project

You can change the following preferences on the *JBoss Tools* > *Web* > *Struts* > *Project* preference page:

On Project panel you define a template for a new Struts created project: servlet version, page template and so on.

		Prefere	ences 🛛
type filter text		Project	<b>⇔</b> • ⇔∗
General	^	Now Project Import	Project
▶ Ant		New Project Import	riget
Connectivity		Struts Version:	Struts 1.1
FreeMarker Editor		Project Template	Riank (*
▶ Help		rioject lemplate.	
HQL editor			✓ Use Default Path
Install/Update		Projects Root:	/opt/work/workspace_beta
Internet	=		
⊅ Java		Serviet Version:	2.3
JBoss jBPM			Register Web Context in server.xml
		Page Template:	
JBoss Servers		rage remplater	
Packaging Archives			
⊽ Web			
Editors	Н		
▷ JSF			
Seam			
Automation			
Customization			
Project			
Struts Pages			
Verification	•		Restore Defaults Apply
< III •	)		
0			OK Cancel

Figure 9.23. Struts Project

Selecting the Import Project tab in the Struts Project screen allows you to determine the default servlet version and whether to register Web Context in server.xml.

<b>9</b>	Preferences	×
type filter text	Project	<b>⇔</b> • ⇔•
type filter text         ▷       General         ▷       Ant         ▷       Connectivity         FreeMarker Editor         ▷       Help         HQL editor         ▷       Install/Update         ▷       Internet         ▷       Java         ▷       JBoss jBPM         ▽       JBoss Tools         ▷       JBoss Servers         Packaging Archives         ▽       Web         ▷       Editors         ▷       JSF         ▷       Seam         ▽       Struts         ▷       Automation         Customization	Project New Project Import Project Servlet Version: 2.3  Register Web Context in server.xml	
Project		
Struts Pages		
Verification	Restore <u>D</u> efaults	Apply
7 XDociet	ОК	Cancel

Figure 9.24. Import Struts Pages

## 9.19. Struts Support

The following preferences can be changed on the JBoss Tools > Web > Struts > Project > Struts Support page.

Select Struts Support screen if you want to configure Struts versions support settings.

<b>9</b>	Pref	erences		×
type filter text	Struts Support			<b>⇔</b> ≁ ⇔≁
▶ General	Struts Support 1.2	Struts Support 1.1	Struts Support 1.0	
<ul> <li>Ant</li> <li>Connectivity</li> </ul>	Servlet Class:	org.apache.struts.a	ction.ActionServlet	<u>C</u> hange
FreeMarker Editor	URL Pattern:	*.do		
HQL editor	TLD Files:	struts-bean.tld;strut	s-logic.tld;struts-html.tld	<u>C</u> hange
<ul> <li>▷ Install/Update</li> <li>▷ Internet</li> </ul>				
▶ Java				
<ul> <li>✓ JBoss JBPM</li> <li>✓ JBoss Tools</li> </ul>				
JBoss Servers				
✓ Web				
Editors				
Seam				
Customization				
✓ Project Struts Support				
Struts Pages			Restore <u>D</u> efaults	Apply
0			ОК	Cancel

Figure 9.25. Struts Support

# 9.20. Title Diagram

JBoss Tools > Web > Editors > Title Diagram screen allows you control some settings for the placement of Tiles definitions in the Diagram mode of the JBoss Tools Tiles editor.

<b>19</b>	Preferences	×
type filter text	Tiles Diagram	<b>⇔</b> • ⇔≁
General	Tiles Diagram	
<ul> <li>Ant</li> <li>Connectivity</li> </ul>	Vertical Spacing: default(20)	•
FreeMarker Editor	Horizontal Spacing: default(185)	•
HQL editor	Definition Name Font: default,size=7,style=1	<u>C</u> hange
Install/Update	Layout Animation	
Internet		
⊅ Java		
JBoss jBPM		
JBoss Servers		
Packaging Archives		
⊽ Web		
JSF Flow Diagram		
Struts Flow Diagram		
Tiles Diagram		
Visual Page Editor		
▷ JSF		
Seam		
Struts		
Verification	Restore Default	S Apply
0	ОК	Cancel



### 9.21. Verification

The following preferences can be changed on the *JBoss Tools* > *Web* > *Verification* page.

On Rules Configuration tab of Verifcation panel you can determine JSF and Struts rules.

<b>9</b>	Preferences	×
type filter text	Verification	<b>⇔</b> • ⇔∗
General	Pules Configuration Ontions	Â
▶ Ant	Rules Configuration Options	
Connectivity	Verification Level: any	
FreeMarker Editor	▼ 🗹 JSF Rules	
▶ Help	Check Faces Config Application Attributes	
HQL editor	Check Faces Config Component Attributes	
Install/Update	Check Faces Config Converter Attributes	
Internet	Check Faces Config Factory Attributes	
Þ Java	Check List Entries Attributes	
JBoss jBPM	Check Managed Bean Attributes	
	Check Managed Property Attributes	
JBoss Servers	Check Map Entries Attributes	
Packaging Archives	Check Navigation Rules	
⊽ Web	Check Phase Listener Attributes	
Editors	Check Referenced Bean Attributes	
▷ JSF	Check Render Kit Attributes	
▷ Seam	Check Renderer Attributes	
Struts	Check Validator Attributes	
Verification	Check Web Descriptor	
XDoclet		
JPA	Check Web Descriptor	
Plug-in Development	Check Struts Message Resources	
Run/Debug	Check Struts configuration controller	
▷ Server	Check Struts configuration file	
▷ Spring	Check action forward	
SQL Development	Check action name	
▶ Team 💌		
0	ОК	Cancel

#### Figure 9.27. Verification

On Options tab you can define a limit for the reported errors number.

Figure 9.28. Options of Verification

## 9.22. View

The following preferences can be changed on the *JBoss Tools* > *JBoss Servers* > *View* page.

The View shows you preferences for JBoss Servers view.

۲					Prefe	rences	×
ty	rpe filter text		View				<b>⇔</b> • ⇔•
Þ	General	*					
Þ	Ant				1		
Þ	Connectivity			Name	Enabled	Description	
	FreeMarker Editor			Modules		Allow modules to be included in the tree.	
Þ	Help		Move Up	i Event Log		A place to view server-related events	
	HQL editor		Move Down	XML Configuration		An extension that lets you quickly change often-used fields or ports	
Þ	Install/Update			🗎 File Filters		A front end to open, edit, or view files in your server's directories.	
Þ	Internet			Inactive Categories		Lists inactive categories and allows you to turn them back on.	
Þ	lava						
Þ	IBoss iBPM			L			
~	IBoss Tools	=	Rolow, you ma	av oot or cloar any prof	oroncos en	acific to aach view extension above	
	✓ IBoss Servers		Some extensio	ons may have no prefe	rences to se	et.	
	View						
	Packaging Archives						
	♦ Web						
	> XDoclet						
	ΙΡΔ						
Þ	Plug-in Development						
Þ	Run/Debug						
,	Server						
Þ	Spring						
Þ	SOL Development						
6	Team						
ľ	TestNG	•				Restore Defaults	Apply
	IC SUI O	0					
(	0					ОК	Cancel

Figure 9.29. View

## 9.23. Visual Page Editor

JBoss Tools > Web > Editors > Visual Page Editor screen allows you to control some aspects of the behavior of the Visual Page Editor (VPE) for JSP files. Also you can define a default editor tab.

9	Prefere	nces 🗙
type filter text	Visual Page Editor	<b>⇔</b> • ⇔-
General	General Templates	
▶ Ant		Show Border for Linknown Tags
P Connectivity		- show border for onknown lags
FreeMarker Editor		Show Resource Bundles Usage as EL Expressions
P Help		☑ Always Prompt for Tag Attributes During Tag Insert
HQL editor		Show Selection Tag Bar
P Install/Update		Always Hide Selection Par Without Promot
P Internet		Always hide selection bar without Frompt
Þ Java	Default Editor Tab:	Visual/Source
P JBoss jBPM		
D JBoss Servers		
Packaging Archives		
✓ Web		
✓ Editors		
JSF Flow Diagram		
Struts Flow Diagram		
Tiles Diagram		
Visual Page Editor		
₽ js⊧		
v Seam		
V Struts		
verification		Restore Defaults Apply
0		OK Cancel

Figure 9.30. Visual Page Editor

On the Templates tab you can edit or remove VPE templates.

<b>9</b>	Preferences	×
type filter text	Visual Page Editor	<b>⇔</b> + ⇔+
General	Conoral Templates	
▶ Ant	General Templates	
Connectivity	URI Tag Name Display Chil	dren Edit
FreeMarker Editor		Remove
▷ Help		- Hernove
HQL editor		
Install/Update		
Internet	_	
⊅ Java		
JBoss jBPM		
JBoss Servers		
Packaging Archives		
∀ Web		
✓ Editors		
JSF Flow Diagram		
Struts Flow Diagram		
Tiles Diagram		
Visual Page Editor		
Þ jsf		
Þ Seam		
Struts		
Verification		
0	OK	Cancel

Figure 9.31. Visual Page Editor Templates

# 9.24. XDoclet

The following preferences can be changed on the *JBoss Tools* > *XDoclet* page.

On XDoclet screen you determine XDoclet module versions.

Preferences			
type filter text		XDoclet	<b>⇔</b> • ⇔•
<ul> <li>General</li> <li>Ant</li> <li>Connectivity</li> <li>FreeMarker Editor</li> <li>Help</li> <li>HQL editor</li> <li>Install/Update</li> <li>Internet</li> <li>Java</li> </ul>		Allows to refresh the modules list from the disk JBoss.net xdoclet module version <ul> <li>4.0.x</li> <li>3.2.x</li> </ul> Refresh XDoclet Modules	
<ul> <li>▷ JBoss jBPM</li> <li>▽ JBoss Tools</li> <li>▷ JBoss Servers</li> <li>Packaging Archives</li> <li>▷ Web</li> <li>▷ XDoclet</li> <li>IPA</li> </ul>	=		
<ul> <li>Plug-in Development</li> <li>Run/Debug</li> <li>Server</li> <li>Spring</li> <li>Col. Development</li> </ul>			
<ul> <li>SQL Development</li> <li>Team</li> <li>TestNG</li> <li>Validation</li> <li>(2)</li> </ul>	•	Restore Defaults	<u>A</u> pply Cancel



## 9.25. XDoclet Templates

To see what XDoclet tags templates available for completion select *JBoss Tools* > *XDoclet* > *Code Assist* > *Templates*.



Figure 9.33. XDoclet Templates

By right clicking on any tree element you can add a new template, remove or rename it and also add a new tree.

Select, for example, Add and you'll be prompted by a list of available elements to add.

<b>9</b>	Preferences	×
type filter text	Templates	<b>⇔</b> • ⇔•
▼       XDoclet         ▼       Code Assist         Templates         Variables         JPA         ▶       Plug-in Development         ▼       Run/Debug         Console         External Tools         ▶       Launching         Perspectives         String Substitution         TCP/IP Monitor         View Managemen	XDoclet Tags templates available for completion   Image: Stateful Session EJB   Image: Stateless Session EJB	
<ul> <li>▷ Spring</li> <li>▷ SQL Development</li> <li>▷ Team         <ul> <li>TestNG</li> <li>Validation</li> <li>▷ Web and XML</li> <li>♥ Web Services</li> <li>Axis Emitter</li> <li>Axis2 Preferences</li> <li>▼</li> </ul> </li> </ul>	СК	Cancel

Figure 9.34. Add New XDoclet Template

By double-clicking some element a new list will be shown: now for available methods.

<b>(</b>	Preferences	×
type filter text	Templates	<b>⇔</b> • ⇔∗
▼ XDoclet         ▼ Code Assist         Templates         Variables         JPA         ▶ Plug-in Development         ▼ Run/Debug         Console         External Tools         ▶ Launching         Perspectives         String Substitution	XDoclet Tags templates available for completion     Image: Stateful Session EJB     Image: Stateless Session EJB	
TCP/IP Monitor View Managemen ▷ Server = ▷ Spring ▷ SQL Development ▷ Team TestNG Validation ▷ Web and XML ♡ Web Services Axis Emitter Axis2 Preferences		•
(3)	OK	Cancel

Figure 9.35. Select New Element

Choose any element from the list, then select an attribute.



Figure 9.36. Select New Element

It will be added to the available templates.

<b>9</b>	Preferences	×
type filter text	Templates	<b>⇔</b> • ⇔•
type filter text	Image: Structure in the s	
String Substitution TCP/IP Monitor View Managemen ▷ Server ▷ Spring ▷ SQL Development ▷ Team	<ul> <li>▷ I Stateful Session EJB</li> <li>▷ I Stateless Session EJB</li> </ul>	
TestNG Validation ▷ Web and XML ♥ Web Services Axis Emitter Axis2 Preferences ♥	к К	Cancel

Figure 9.37. Select New Element

# 9.26. XDoclets Variables

By selecting *JBoss Tools* > *XDoclet* > *Variables* you define variables used in templates.

<b>9</b>	Preferences	×
type filter text	Variables	⇔- ⇔-
Þ Help	Definition of the variab	les used in templates
HQL editor	Classname	2(\w*)\$
Install/Update	classifiante.	
Internet	Classname Subset:	^(\w*)Bean\$
⊅ Java	Packagename:	(\S*)\$
JBoss jBPM	Parent Packagename:	^(\S*)\.\w*\$
JBoss Servers		
Packaging Archives		
Þ Web		
✓ XDoclet		
Templates		
Variables		
JPA		
Plug-in Development		
Run/Debug		
Server		
Spring		
SQL Development		
▶ Team	-	
TestNG		
Validation		
· · · · · · · · · · · · · · · · · · ·		Restore Defaults Apply
0		OK Cancel

Figure 9.38. XDoclets Variables

# 9.27. Changing Default Environment During Project Creation

To change the default environment and project template for either JSF or Struts new project creation:

- 1. Select Window > Preferences > JBoss Tools > Web > {JSF or Struts} > Project
- 2. For Version set the environment you want to be the default one
- 3. For Project Template set the template you want to be the default one

<b>9</b>	Prefere	nces X
type filter text	Project	<b>⇔</b> = ⇔ =
type filter text ▷ Install/Update ▷ Internet ▷ Java ▷ JBoss jBPM ♡ JBoss Tools ▷ JBoss Servers Packaging Archive ♡ Web	Project Import Version: Project Template: Projects Root:	Project          JSF 1.2       ↓         JSFBlankWithoutLibs       ↓         ✓       Use Default Path         /opt/work/workspace_rc1       Change
<ul> <li>▷ Editors</li> <li>▽ JSF</li> <li>JSF Pages</li> <li>Project</li> <li>▷ Seam</li> </ul>	Servlet Version: Page Template:	2.4   ▼     ✓ Register Web Context in server.xml     ↓
<ul> <li>Struts         <ul> <li>Verification</li> <li>XDoclet</li> <li>JPA</li> <li>Plug-in Development</li> <li>Run/Debug</li> <li>Server</li> <li>Spring</li> <li>SQL Development</li> <li>Team                  TestNG                  Validation</li> <li>Web and XML</li> </ul> </li> </ul>		Restore Defaults Apply
0		OK Cancel

Figure 9.39. Changing Enviroment Template

## 9.28. Changing Default Project Template During Project Creation

To change the default project template for either JSF or Struts new project creation:

- 1. Select Window > Preferences > JBoss Tools > Web > {JSF or Struts} > Project
- 2. For Project Template set the template you want to be the default one

	Prefere	nces 🗙
type filter text	Project	<b>⇔</b> • ⇔•
Install/Update	New Project Import	Project
Internet	International International	
⊅ Java	Struts Version:	Struts 1.1
JBoss jBPM	Project Template:	Blank
	, , , , , , , , , , , , , , , , , , , ,	
JBoss Servers		Use Default Path
Packaging Archive	Projects Root:	/opt/work/workspace_rc1
⊽ Web	Servlet Version	23
▶ Editors	Service version.	
▷ JSF		Register Web Context in server.xml
Þ Seam	Page Template:	÷
Automation		
Customizati		
Project		
Struts Pages		
Verification		
P XDoclet		
JPA		
Plug-In Development		
V Run/Debug		
b Spring		
b SOL Development		
D Team		
TestNG		
		Restore Defaults Apply
0		OK Cancel

Figure 9.40. Changing Project Template