Visual Web Tools Reference Guide

ISBN: Publication date: April 2008

Visual Web Tools Reference Guide

PDF version

Visual Web Tools Reference Guide

Copyright $\ensuremath{\mathbb{O}}$ 2007, 2008 JBoss, a division of Red Hat Inc.

1. Visual Web Tools	. 1			
1.1. Key Features of Visual Web Tools 1				
1.2. Other relevant resources on the topic	. 2			
2. Spring Tools	3			
2.1. Spring IDE guide	. 3			
2.1.1. Add Spring Project Nature	3			
2.1.2. Create New Spring Project	. 3			
2.1.3. Add References To Other Spring Projects	. 3			
2.1.4. Add Spring Beans Config Files	. 3			
2.1.5. Create Spring Beans Config Sets	3			
2.1.6. Open Spring Explorer	3			
2.1.7. Validate Spring Beans Config	. 3			
2.1.8. Open Spring Beans Graph	. 3			
2.1.9. Search Spring Beans	. 3			
3. Editors	. 5			
3.1. Editors Features	. 5			
3.1.1. OpenOn	. 5			
3.1.2. Content Assist	11			
3.1.3. Synchronized Source and Visual Editing	24			
3.2. Visual Page Editor	26			
3.2.1. Visual/Source View	27			
3.2.2. Pages Styling	30			
3.2.3. Templating	37			
3.2.4. Advanced Settings	39			
3.2.5. Page Preview	48			
3.2.6. Setup notes for Linux	48			
3.3. More Editors	49			
3.3.1. Graphical Properties Editor	49			
3.3.2. Graphical TLD Editor	51			
3.3.3. Graphical Web Application File (web.xml) Editor	56			
3.3.4. CSS Editor	61			
3.3.5. JavaScript Editor	63			
3.3.6. XSD Editor	65			
3.3.7. Support for XML Schema	71			
4. JBoss Tools Palette	73			
4.1. Palette Options	74			
4.1.1. Palette Editor	74			
4.1.2. Show/Hide	81			
4.1.3. Import	83			
4.2. Using the Palette	83			
4.2.1. Inserting Tags into a JSP File	83			
4.2.2. Adding Custom JSF Tags to the JBoss Tools Palette	86			
4.3. RichFaces Support	91			
4.3.1. Relevant Resources Links	92			

5.	Web Projects View	93
	5.1. Project Organization	93
	5.2. Drag and Drop	
	5.2.1. For a Property	
	5.2.2. For Managed Bean Attributes	
	5.2.3. Navigation Rules	
	5.2.4. For a Tag Library File Declaration	
	5.2.5. For JSP Pages	
	5.3. Developing the Application	100
	5.4. Expanding Tag Library Files	101
	5.5. Drag and Drop Tag Libraries on to JBoss Tools Palette	102
	5.6. Create and Import JSF and Struts Projects	102
6.	JBoss Tools Preferences	105
	6.1. Packaging Archives	106
	6.2. Editors	108
	6.3. Visual Page Editor	110
	6.4. El Variables	113
	6.5. JSF	115
	6.6. JSF Pages	116
	6.7. JSF Project	117
	6.8. JSF Flow Diagram	120
	6.9. Label Decorations	122
	6.10. Seam	124
	6.11. Seam Validator	125
	6.12. Struts	126
	6.13. Struts Automation	127
	6.14. Plug-in Insets	128
	6.15. Resource Insets	130
	6.16. Struts Customization	131
	6.17. Struts Project	132
	6.18. Struts Support	134
	6.19. Struts Pages	135
	6.20. Struts Flow Diagram	136
	6.21. Tiles Diagram	138
	6.22. Verification	139

Visual Web Tools

This guide covers the usage of Visual Web Tools in JBoss Developer Studio and JBoss Tools. The difference between these products is that JBoss Tools are just a set of Eclipse plugins where JBoss Developer Studio adds the following functionality:

- an installer
- Eclipse and Web Tools preconfigured
- JBoss EAP with JBoss AS and Seam preconfigured
- · 3rd party plugins bundled and configured
- · access to RHEL and Red Hat Network
- · access to the JBoss/Red Hat supported software

For additional information, please visit the JBoss Developer Studio home page at: <u>http://</u> <u>www.jboss.com/products/devstudio</u>.

In JBoss Tools there is an extensive collection of specialized wizards, editors and views that can be used in various scenarios while developing Web applications. The following chapters walk through these features.

1.1. Key Features of Visual Web Tools

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

Table 1.1. Key Functionality for Visual Web Too	ls
---	----

Feature	Benefit	Chapter
Visual Page Editor	Powerful and customizable visual page editor. Possibility to develop an application using any web technology: jsf, seam, struts, jsp, html and others. Developing using four tabs: visual/source, visual, source and preview. Fast and easy switching between these tabs. Split screen design of visual and source views. Full and instant synchronization between source and visual views. Integration with properties and outline views. Graphical toolbar to add inline styling to any tag.	<u>visual page editor</u>
JBoss Tools Palette	Organizing various tags by groups, inserting tags into a jsp or xhtml page with one click,	jboss tools palette

Feature	Benefit	Chapter
	adding custom or 3rd party tag libraries into the palette, easy controling the number of tag groups shown on the palette.	
Web Projects View	Visualizing and displaying projects by function. Easy selecting of different 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. Expanding and inspecting tag library files. Selecting custom and third-party tag libraries to drag and drop onto the JBoss Tools Palette.	web projects view
OpenOn	Easy navigation between views and other parts of your projects.	<u>openOn</u>
Content Assist	Code completion proposals while working with html, java, JavaScript , xml, jsp, xhtml, xhtml, seam project and jsf configuration files. Content assist based on project data (dynamic code assist); with graphical editor. Code completion for values from property files, beans attributes and methods, navigation rule outcomes and jsf variables.	content assist
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 attributes, navigation rules, tag library file declarations, jsp files from web projects view by clicking them and draging to source code.	<u>visual page editor</u> <u>drag-and-drop</u>
RichFaces Support	Tight integration between JBDS and <i>RichFaces</i> frameworks. Easy managing RichFaces components in any web application. Support for RichFaces and Ajax4jsf libraries in JBoss Tools Palette. Rendering RichFaces components in Visual Page Editor.	<u>RichFaces support</u>

1.2. Other relevant resources on the topic

All JBoss Developer Studio/JBoss Tools documentation you can find here.

The latest documentation builds are available *here*.

Spring Tools

JBoss Developer Studio is bundled with <u>Spring IDE</u> for Eclipse. Visit Spring IDE site for the latest versions and documentation.

2.1. Spring IDE guide

<u>Spring IDE</u> is a graphical user interface for the configuration files used by the <u>Spring Framework</u>. It's built as a set of plugins for the Eclipse platform.

- 2.1.1. Add Spring Project Nature
- 2.1.2. Create New Spring Project
- 2.1.3. Add References To Other Spring Projects
- 2.1.4. Add Spring Beans Config Files
- 2.1.5. Create Spring Beans Config Sets
- 2.1.6. Open Spring Explorer
- 2.1.7. Validate Spring Beans Config
- 2.1.8. Open Spring Beans Graph
- 2.1.9. Search Spring Beans

Editors

In the <u>JSF Tools Reference Guide</u> and <u>Struts Tools Reference Guide</u> you had possibility to read about Graphical Editor for <u>JSF</u> and <u>Struts</u> configuration files, <u>Graphical Editor for Tiles Files</u>, <u>Graphical Editor for Struts Validation Files</u>. All these editors have <u>OpenOn</u> and <u>Content Assist</u> features, which are described in more details in this document. In addition you get to know a <u>Visual Page Editor</u> for combined visual and source editing of Web pages and many <u>other editors</u> for different types of files.

3.1. 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.

3.1.1. OpenOn

OpenOn lets you easily link directly from one resource to another in your project without using the Package Explorer view (project tree). With OpenOn, you can simply click on a reference to another file and the file will be opened.

OpenOn is available for the following files:

- XML files
- JSP/XHTML Pages
- Java files

3.1.1.1. 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 hover 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.



Figure 3.1. NameBean Managed Bean

This is the result of using OpenOn.



Figure 3.2. NameBean Java Class

You can also use OpenOn with defined attributes.



Figure 3.3. OpenOn With Defined Attributes

You can also open any JSP pages.

1	faces-config.xml 🛱 🕗 User.java	• 🗆				
	<pre>?xml version="1.0" encoding="UTF-8"?></pre>	^				
	<faces-config <="" td="" version="1.2" xmlns="http://java.sun.com/xml/ns/javaee"><td></td></faces-config>					
	<pre>xmlns:xi="http://www.w3.org/2001/XInclude"</pre>					
	xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="htt					
	<man aged-bean=""></man>					
	<description>User Name Bean</description>					
	<man -="" aged="" ame="" bean="" n="">user</man>					
	<man aged-bean-class="">demo.User</man>					
	<managed-bean-scope>session</managed-bean-scope>					
	<man aged-property=""></man>					
	<property-name>name</property-name>	-				
	<property-class>java.lang.String</property-class>					
	<value></value>					
	<navigation-rule></navigation-rule>					
	<from-view-id>/pages/inputUserName.jsp</from-view-id>					
	<navigation-case></navigation-case>					
	<from-outcome>hello</from-outcome>					
	<to-view-id>/pages/hello.jsp</to-view-id>					
Di	agram Tree Source					

Figure 3.4. JSP Page OpenOn

3.1.1.2. 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.



Figure 3.5. OpenOn With Imported Property Files

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



Figure 3.6. OpenOn With CSS File

Open managed beans:

🔊 faces-config.xml 🛛 🕗 User.java 📾 inputUserName.jsp 🕴	- 8			
<hl><h:outputtext value="#{Message.header}"></h:outputtext></hl>	-			
<h:messages style="color: red"></h:messages>				
<h:form id="greetingForm"></h:form>				
<h:outputtext value="#{Message.prompt_message}"></h:outputtext>				
<h:inputtext required="true" value="#{user.name}"></h:inputtext>				
<f:validatelength maximun<sup="">(2) 30 minimum="3"/></f:validatelength>				
<h:commandbutton action="hello" value="Say Hello!"></h:commandbutton>	=			
	-			
	▶			
	×			
Visual/Source Visual Source Preview				

Figure 3.7. OpenOn With Managed 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:

🔊 faces-config.xml 🕢 User.java 🔯 inputUserName.jsp 🛿	- 8
<hl><h:outputtext color:="" red"="" value="#{Message.header}*/></hl></td><td>A_</td></tr><tr><td><h:messages style="></h:outputtext></hl>	
<h:form id="greetingForm"></h:form>	
<h:outputtext value="#{Message.prompt_message}"></h:outputtext>	
<h:inputtext required="true" value="#{user.name}"></h:inputtext>	
<f:validatelength maximum="30" minimum="3"></f:validatelength>	
<h:commandbutton action="hello" value="Say Hello!"></h:commandbutton>	=
	~
	×
Visual/Source Visual Source Preview	

Figure 3.8. OpenOn With JSF Tag

3.1.2. Content Assist

Content assist is available when working with

• Seam project files

- JSF project files
- Struts project files
- JSP files
- <u>RichFaces components</u>
- <u>ESB XML files</u>

3.1.2.1. 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

3.1.2.1.1. 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:



Figure 3.9. Content Assist in JSF Configuration File

Content Assist for JSF JSP file:

🔊 faces-config.xml	🚺 User.java	🐵 *inputUserName	ijsp 🕅		- 0]
<pre>%@ taglib uri=" %@ taglib uri=" <f:loadbundle va<br=""><html></html></f:loadbundle></pre>	http://java.su http://java.su r="Message" ba nput User Name l="styleshell"	<pre>n.com/jsf/core" n.com/jsf/html" sename="demo.Mes: Page type="text/css"</pre>	prefix="f" prefix="h" sages"/> href="st)	' %> ' %> /le.css"/>		
<1. VIEW>					=	
f:actionListe f:attribute f:convertDa f:convertDa f:convertNu f:convertNu f:facet f:facet f:loadBundle f:phaseListe t:selectitem html b f:selectitem	ener IteTime Imber e ener			Register an ActionListener instanc associated with the closest paren	e on the	e UlComponent aponent custom action
Visual/Source Visual S	ource Preview					

Figure 3.10. Content Assist in JSP File

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



Figure 3.11. Content Assist in web.xml File

3.1.2.1.2. 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 figure below 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.



Figure 3.12. Inserting Message

In the following screenshot 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).

🔊 faces-config.xml 🛛 🕮 web.xml 🔯 *inputUserName.jsp 🕴	- 8
<f:view> <h:outputtext value="#{Message.header}" }=""></h:outputtext> <h:messages style="color: red"></h:messages> <h:form id="greetingForm"> <h:outputtext value="#{Message.prompt_message}"></h:outputtext> <h:inputtext value="#{Message.prompt_message}"></h:inputtext> <h:inputtext value="#{Message.prompt_message}"></h:inputtext> <h:inputText value="#{Me</td><td></td></h:form></f:view>	
Image: transmission of the state of	×

Figure 3.13. Attributes List

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

		_
<h:commandbutton action="</td><td>" value="Say Hello!"></h:commandbutton>		
	© #{user}	
	(a) hello	
	JSP expression - JSP expression <%=%>	
		J
the second secon		, ¹
htmi body r:view h:rorm h:commandButton		
Visual/Source Visual Source Preview		

Figure 3.14. Code Assist

3.1.2.1.3. 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*.

🔉 *faces-config.xml 🛛 🔛 web.x	cml 🔯 inputUserName.jsp	- 8
Faces Config Editor		
	+ Factories	
 Faces-config.xml* Application Components Converters Anaged Beans Referenced Beans Referenced Beans Render Kits Validators Extensions 	Application-Factory:	su Browse
Diaman Tasa Course		
Diagram lifee Source		

Figure 3.15. Content Assist in Tree JSF Configuration Editor

3.1.2.2. Struts Project Files

Content Assist features are available when you work with Struts projects.

3.1.2.2.1. Content Assist for Struts Configuration File

Content Assist helps you in Struts Configuration file.

🐟 *struts-config.xml 🕴		٥
<pre><?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE struts-config PUBLIC "-//Apache Softwa "http://struts.ap"</pre> </pre>	re Foundation//DTD Struts Configuration 1.2//EN	
<pre><struts.config> <data-sources></data-sources> <form-beans> <form-beans> <global.exceptions></global.exceptions> <global.forwards> <forward name="getName" path="/pages/inputname </global.forwards></pre></th><td>imeForm"></forward><td></td></global.forwards></form-beans></form-beans></struts.config></pre>		
<pre><action-mappings> <action <="" name="GetNameForm" path="/greeting" pre=""></action></action-mappings></pre>	<pre>scope="request" type="demo.GreetingAction"></pre>	
Attribute : attribute Data Type : CDATA	 attribute className forward id include input parameter prefix roles suffix 	11

Figure 3.16. Struts Content Assist

3.1.2.2.2. Content Assist for Struts JSP File

Using Code Assist in Struts JSP file is shown below.

<bean :<="" th=""><th>/ 51</th><th></th><th></th></bean>	/ 51		
<table< th=""><th></th><th>É</th><th>Element : bean:cookie</th></table<>		É	Element : bean:cookie
<t< th=""><td><> bean:define</td><td>1</td><td></td></t<>	<> bean:define	1	
</th <th><> bean:header</th> <th></th> <th></th>	<> bean:header		
<t< th=""><th><> bean:include</th><th></th><th></th></t<>	<> bean:include		
	<> bean:message		
	<> bean:page		
</th <td><> bean:parameter</td> <td></td> <td></td>	<> bean:parameter		
<td><> bean:resource</td> <td></td> <td>_</td>	<> bean:resource		_
~	<> bean:size		
	🗘 bean:struts		
		_	
Visual/Source Visua	Source		

Figure 3.17. Struts JSP Content Assist

3.1.2.3. JSP Pages

3.1.2.3.1. Content Assist for JSF Tags

JBDS provides full code completion for JSF tags:



Figure 3.18. 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. At this point you can ask for attribute proposals.

<pre><body> <fiview> <fiview> <fi:inputText value=" </rich:calendar_popup </rich:calendar> <h3> <h3> </h3> </h3></fiview> </fiview></body> </pre>	<pre>> #{Message} #{user} #{user} #{applicationScope} #{cookie} #{facesContext} #{facesContext} #{header} #{header} #{headervalues} #{param} #{paramValues} # #{paramValues} # #{requestScope} # #{requestScope} ##{tequestScope} #{tequestScope} #{tequestScope}</pre>	
(Visual/Source Visual Source Preview	(a) #{requestScope}	2

Figure 3.19. Attributes Content Assist

3.1.2.3.2. Content Assist for JSTL Tags



Figure 3.20. JSTL Tags Content Assist

3.1.2.3.3. Content Assist for HTML Tags

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

can> input ↔ input ↔ ins ↓ JSP expression - JSP expression <%=%>	Defines a form control for user input
(*) III Visual/Source Visual Source	v V X

Figure 3.21. HTML Tags Content Assist

You can use as well attributes proposals for HTML tags:

de dus			
<pre><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></pre>			
sinput type= name=		Attribute : tupe	
O "button"		Data Tupe : ENUM	
<rich:cal "checkt<="" @="" th=""><th>box"</th><th>Enumerated Values :</th><th></th></rich:cal>	box"	Enumerated Values :	
(a) "file"		- text	
<th>·</th> <th>- password</th> <th></th>	·	- password	
<h3> @ "image"</h3>	·	- radio	
▲ a "passw	ord"	- submit - reset	
< @ "radio"		- file	
- (3) "reset"		- hidden - image	
@ "submit	e la	- button	
(a) "text"			
↔ a4i-acti	ionparam .		
(v v K	2
Visual/Source Visual Source Previ	ew		

Figure 3.22. HTML Tags Content Assist

3.1.2.3.4. Content Assist for JavaScript Tags

Figure 3.23. JavaScript Tags Content Assist

3.1.2.4. RichFaces components

JBDS indeed provides code completion for <u>*RichFaces*</u> framework components. All you have to do is to install RichFaces libraries into your project. See <u>here</u> how to install it.

🔋 Pack 🕺 🛞 Sea 🗖 🗖	📾 *helio.jsp 😫 🗟 web.xml	- 0			
 	<pre></pre>				
Web Projects X X	<pre><html></html></pre>				
Element : rich-datal ist	Rrich;	-			
	<>rich:dataFilterSlider				
	c> nch:dataGrid				
	A rich:dataOrderedl.ict				
<> rich:dudi bulc					
<>rich:dragIndicator					
<>rich:dragListener					
<> rich:dragSupport					
L	at data data Parcella are	•			

Figure 3.24. Content Assist for RichFaces Components

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

Attribute name	Value		
ointPoint			
ocalValueSet			
ocale			
monthLabels			
monthLabelsShort			
optionalFooter			
optionalHeader			
oopup	false		
preloadDateRange	Beg		
preloadDateRange	nd		
rendered			

Figure 3.25. Insert Tag

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

🖻 he	llo.js	pΣ	3	æ	web	.xm	I			- 0
• (\$					⇒ ⇒ B I <u>U</u> <u>A</u>	
*	<<	<		July,	2008	1	>	>>		
\$		Sun	Mon	Tue	Wed	Thu	Fri	Sat		
	27			1	2	3	4	5		
	28	6	7	8	9	10	11	12		
1	29	13	14	15	16	17	18	19		
	30	20	21	22	23	24	25	26		
	31	27	28	29	30	31				
	07/	01/200	8		_		Т	xlay		
	He	llo,	#{t	iser	.na	me	}!			
html Visua	bo I/So	dy urce	f:vi	ew sual	Sou	irce	Pre	viev	w	×

Figure 3.26. RichFaces Component

3.1.2.5. 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:

 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

Follow these steps to set what is available for code assist:

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

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

• Add bundle resource (property file) code assist:

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

Add managed bean property code assist:

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

• Add managed bean property but of a specified type:

```
<AttributeType ...>
<proposal type="beanProperty">
<param name="type" value="java.lang.Boolean"/>
</proposal>
</AttributeType>
```

• Add managed bean method with a signature:

```
<AttributeType ...>
<proposal type="beanMethodBySignature">
<param name="paramType" value="javax.faces.context.FacesContext"/>
<param name="paramType" value="javax.faces.component.UIComponent"/>
<param name="paramType" value="java.lang.Object"/>
<param name="returnType" value="void"/>
</proposal>
</AttributeType>
```

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.

3.1.3. 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 a full control over all project source files. Any changes you make in the source view 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.



Figure 3.27. Three Views are Synchronized

The same is relevant to all other JBoss Developer Studio editors.

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

🕎 web.xml 🛙				- 8	
Web XML Editor					
▼ web	▼ Web	Descriptor 2.4		<u>^</u>	
 Web.xml Context Params Filters Exclusion-config Servlets Servlets Servlets Welcome-file-list Error Pages JSP Config Security Constrain Iogin-config Security Roles Env Entries 	Name: Display Descrip web.xml EX <7xml versio <web-app ver<br="">xmlns:xsi=" <display-na <context-pa <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan <param-nan servlet-na</param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </param-nan </context-pa </display-na </web-app>	<pre>value of interval interva</pre>	n.com/xml/ns/j2ee" -instance" xsi:schemaLoc D gureListener <th>cation="http:/</th> <th>//java.sun.c</th>	cation="http:/	//java.sun.c
	Tree Source				

Figure 3.28. Two Views are Synchronized

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



Figure 3.29. Two Views are Synchronized

3.2. 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 three tabs: Visual/Source, Source and Preview. To switch between the views you can use tabs at the bottom of the VPE or the shortcuts *Ctrl* + *PageUp/Ctrl* + *PageDown*.

🖻 login.xhtml 🕴 🧧 🗖
<pre><!DOCTYPE composition PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"</td> </pre>
anssem= >sflafvts_sanssem*ds (
▼ [○]Normal ○] Normal ○] B I U / A
% #{projectName}: Home Welcome, #{identity.username}! Login Logout
Error Messages
Login
Please login using any username and password
Username #{identity.username}
Visual/Source Source Preview

Figure 3.30. Visual Page Editor

3.2.1. Visual/Source View

Using the Visual/Source view you can edit your pages in the Source and Visual modes simultaneously having an instant synchronization between them:



Figure 3.31. Visual/Source View

The view is designed in the form of a split pane with toggle buttons for quickly moving between Source, Visual or Source/Visual modes as shown on the figure above.

One more way to toggle between the various states of the split pane is using the shortcuts *Shift* + *F*6 for maximizing/restoring the Source part and *Shift* + *Alt* + *F*6 for maximizing/restoring the Visual part.



It should be pointed out that, no matter in what mode you are working, you get a full integration with Properties and Outline views:

🔲 Properties 🛿 🖳 🗖	📾 *login.xhtml 🛛 🗖 🗖	🗄 Outline 🕱 📄 🗖 🗖
🗉 🔅 🖪 🗶 🔽	<h:form id="login"></h:form>	G DOCTYPE:composition
Property Value	<pre>crich:papel></pre>	▽ <> ui:composition xmlns=http:/
	<f:facet name="header">Login</f:facet>	▽ 🗘 ui:define name=body
bgcolor	copplease login using any username and passwords (op	<> h:messages styleClass
binding	sing any usering any userinane and password opp	マ <> h:form id=login
border	<pre><div class="dialog"></div></pre>	
captionCl	<pre><h:outputlabel for="username">Username</h:outputlabel></pre>	<> f:facet name=he
captionSt	<pre><h:inputtext <="" id="username" pre=""></h:inputtext></pre>	<> p
cellpaddir	<pre>value="#ildentily.username/"/> <h:outputlabel for="password">Password</h:outputlabel> </pre>	▽ <> div class=dialog
cellspacir		❤ ≪⇒ h:panelGrid cc
columnCl_name,v	▼ ○ Normal ○ Normal ○ B I U	<>> h:outputLal
columns 2	×	<> h:inputText
dir 🔶	Ag Login	<> h:outputLał
footerCla	Please login using any username and password	<> h:inputSecr
frame		<> h:outputLał
headerCl:	Username #{identity.username}	<> h:selectBoc
id	D Password D	I div class=actionBut
lang	Remember me	
onclick		
ondblclick	ui:composition ui:define h:form rich:panel div h:panelGrid 🛛 🗙	
	Visual/Source Source Preview	< III >

Figure 3.32. Integration with Properties and Outline Views

It's also possible to use the <u>JBoss Tools Palette</u> to insert any tag from the list of tag libraries to the page you are editing with just a click or drag-and-drop.

📾 *login.xhtml 🛛 🦳 🖓	🗆 🚿 jBoss Tool 🛛 🗖 🗖
<h:form id="login"></h:form>	🛠 🗟 🛇
<pre>crich:panel></pre>	🕞 JBoss Ajax4jsf
<f:facet name="*header*">Login</f:facet>	😂 JBoss RichFaces 🛛 🗠
and pace leave using any uperforme and pacewords (as	modalPanel
specease cogin using any username and passwords/pa	(a) nodeSelectListener
<div class="dialog"></div>	confide Select Eistener
<pre><h:panelgrid columnclasses="name, valu</pre></td><td>orderingList</td></tr><tr><td><h:outputLabel for=" columns="2" rowclasses="prop" username"="">Username</h:panelgrid></pre>	paint2D
<h:inputtext <="" id="username" td=""><td>opanel</td></h:inputtext>	opanel
value="#(identity.username)"/>	() nonelPer
<h:outputlabel for="password">Password</h:outputlabel>	panelBar
<pre><n:inputsecret id="password" password)#="" xolue="#fidentity"></n:inputsecret> </pre>	opanelBaritem
	> panelMenu
	panelMenuGroup
Vormal C Normal B I U Z. A.	panelMenultem
🛠 Login	(a) recursive TreeNode
A	ecursive ineervoide
Please login using any username and password	scrollableDataTable
	separator
Usemame #(identity.username)	simpleTogglePanel
Password	spacer
Remember me	subTable
	suggestionbox
	≪>tab
cogn	
sine side the billion	Boss Seam
uncomposition underine hitrorm	▷ JSF Facelets
Visual/Source Source Preview	⇒ JSF HTML

Figure 3.33. Inserting Tag From the Palette

3.2.1.1. JSP Syntax Validation

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

3.2.2. Pages Styling

Most web pages use the cascading style sheets (CSS) to control the way they look. With Visual Page Editor you can easily stylise your pages. In this section we are going to introduce you to a powerful mechanism that VPE provides for a complete control over pages styling.

3.2.2.1. Inline Style Editing

In the Visual part of the VPE there is a graphical toolbar, use it to add inline styling to JSF and Struts tags on your page. The toolbar can be hidden by clicking on arrow sign in the upper left corner.



Figure 3.34. Text Formatting

For editing inline styles for DOM elements VPE also provides CSS Dialog. It can be called from *style* line in the Properties view for a currently selected element.
□ Properties 🛛	🗄 🔅 🛤 🗙 🔻 🗖 🗖
Property	Value
onselect	
readonly	
rendered	
required	
requiredMessag	9
size	
style	()
styleClass	
tabindex	
title	=
validator	
validatorMessa	e 🗸

Figure 3.35. Call the CSS Dialog

CSS Dialog has four tabs where css properties for text, background, borders and others can be specified. A simple preview which is generated at the top of the CSS Dialog allows you to see the changes before you apply them.

	CSS Style Dialog	×
Text for preview	v	
Text/Font Backg	round Boxes Property Sheet	Quick Edit
Font Family:		
Color:	FireBrick 🗸	
Font Size:	18 🗸	px 🗘
Font Style:	~	
Font Weight:	~	
Text Decoration:	~	
Text Align:	~	
	ок	Cancel

Figure 3.36. CSS Dialog

3.2.2.2. External Stylesheets

The pages you are working with in VPE can use external stylesheets. VPE allows you to create new style classes in existing stylesheets and/or edit them as well. For these purposes CSS Style Class Dialog is provided.

Select the element for which you need to create or edit style class and press button next to *styleClass* field in Properties view.



Figure 3.37. Calling the CSS Style Class Dialog

It'll pick up the CSS Style Class Dialog which looks like on the figure below.

😔 CSS Style Class Dialog 🗙			
CSS file : Style class :			
Text/Font Background Boxes Pr	operty Sheet		
Font Family:			
Color:	 III 		
Font Size:	~		
Font Style:	~		
Font Weight:	~		
Text Decoration:	~		
Text Align:	~		
	OK Cancel		

Figure 3.38. CSS Style Class Dialog

First, you should specify the CSS file where you are going to put your style class. Do this by pressing button next to the CSS file field.

SS File Selecti	on 🗙
Select CSS file from the tree:	
🗢 🗁 WebContent	
▽ 🗁 stylesheet	
theme.css	
0	K Cancel

Figure 3.39. CSS File Selection

To create new CSS class write its name in the *Style class* field and then configure style settings switching between the tabs: *Text/Font, Background, Boxes, Property Sheet.* To add existing styling to the chosen element expand the list of the existed style classes and point to the necessary one.

🥮	CSS Style Class Dialog X	J
CSS file :	WebContent/stylesheet/theme.css	
Style class :	~	
	body	
Text/Font E	a:active,a:link,a:visited	
Font Fa	a:hover	
	input,textarea	
C	input[type='submit'],input[type='button']	
Font 5	.tableControl,.actionButtons	
	.tableControl a	
Font S	.tableControl	
Font We	.footer	
.rich-table		
Text Decora	hl	
Text A	body	
	.body	
	.columnHeader:hover	
	.message	
	name	
	.value	
	.error	
	.errors	
	img.errors	
	.errors input	
	.errors textarea	
	.required	
	.rich-stglpanel-body	

Figure 3.40. Style Class Selection

Quick Edit gives a preview of the properties which are set for the existing style class. You can easily modify them with the help of this wizard.

🍪 CSS Style Class Dialog 🛛 🗙				
Text for preview				
<	11			>
CSS file : WebCo	ntent/stylesh	eet/theme.css	5	
Style class : .mess	age			~
Text/Font Backgro	und Boxes	Property Shee	t C	uick Edit
Background Color:	#F0F8FF		~	
Border:	1px solid #F	FCC00		
Font Size:	12		~	px 🗘
Margin Bottom:	5		~	px 🗘
Margin Top:	5		~	px 🗘
Padding:	5			px 🗘
	(ОК		Cancel

Figure 3.41. Quick Edit

Preview at the top of the CSS Style Class Dialog visualizes the result.

The dialog for creating a new CSS class, which is called from *New* > *Other...* > *JBoss Tools Web* > *CSS Class,* looks the same.

9	×
Create New CSS Class	
Create New CSS Class	
CSS file :	
Style class :	~
Text/Font Background Boxes Property Sheet	
Font Family:	
Color:	· 🔛
Font Size:	✓
Font Style:	~
Font Weight:	~
Text Decoration:	T
Text Alian:	T
? Angle Back Next > Finish	Cancel

Figure 3.42. New CSS Class Dialog

3.2.3. Templating

The VPE also makes it possible to create templates for unknown tags.

To call the Template dialog for a tag, right-click on it in Visual mode and select Template option.

💼 *h	📾 *home.xhtml 🕴 📃 🗖			
	<pre>xmlns:rich="http://richtaces.org/rich" template="layout/template.xhtml"></pre>			
<	<pre><ui:define name="body"></ui:define></pre>			
	<h:messages globalonly<="" td=""><td>y="true" styleClass="mes</td><td>sage*/></td><td></td></h:messages>	y="true" styleClass="mes	sage*/>	
1		III		
•	🗘 🛛 Normal	0 Normal 0	BIULA	
×				^
<i>®</i>	h:unknowntag h:messages	<h:unknowntag> Attributes</h:unknowntag>		
	Welcome!	Parent Tag >		
	This empty shell application inc	Insert Around		_
	 Ant build script 	Insert Before >		
	 Deployment to JBoss Development and proc 	Insert After >		
	 Integration testing usin JavaBean or EJB 3.0 JPA entity classes 	Template		
	 A configurable DataSo Templated Facelets vi 	Cut		
	 RichFaces panels and Default CSS styleshee 	Сору		~
ui:co	mposition ui:define h:unkno	Paste		×
Visua	al/Source Source Preview	Re <u>m</u> ove		_

Figure 3.43. Calling Template Dialog

Here is what the Template dialog looks like.

6		Template	×
Tag A	Attributes		
URI: Tag N	http://java Jame: h:unknowr	.sun.com/jsf/html htag	
	Tag for Display	div	
	Children	X	
	lcon		
	Value	{name()}	
	Style		
1		OK Cance	

Figure 3.44. Template Dialog

Tag for Display field in the Template dialog requires specifying a type of tag. It can be SPAN, DIV, TABLE or any other html element. Here it's also possible to mark weather the tag is children

or not (by checking *Children*), whether it contains an IMG tag (by checking *lcon*) and specify a value for it.

What comes to the *Style* field, you can fill it out manually or make use of the button next to the field to bring the <u>CSS Dialog</u> [31] for editing styles.

You can observe all defined templates in the <u>VPE Preferences</u> on the Templates tab which you can quickly access by pressing <u>Preferences button</u>.

Preferences (Filtered)					
type filter text	Visual Page Editor				⇔~ ⇔~ ▼
	General Templates				
∀ Web	URI	Tag for Display	Tag Name	Children	
	http://iava.sun.com/isf/html	div	h:unknowntag	ves	Edit
Visual Page Edi				no	Remove
				no	
				no	
		111		>	
0				ок	Cancel

Figure 3.45. Templates Tab of the VPE Preferences Page

Here it's possible to edit or remove any listed in the table template.

3.2.4. Advanced Settings

In the left vertical pane of the Visual part there are three buttons: *Preferences, Refresh* and *Page Design Options.*

<u> </u>	🗘 🛛 Normal	Normal 🗘 B I 😐 🔏 🛕	
X	Login		^
49 50	Please login using any us	sername and password	
	Username	#{identity.username}	
	Password	*********	=
	Remember me		
	Login		~
ui:compo	osition ui:define h:for	m rich:panel div h:panelGrid h:outputLabel	×
Visual/So	ource Source Preview		

Figure 3.46. Buttons on the Visual Part of VPE

• *Preferences* button provides a quick access to Visual Page Editor preferences.

Preferences (Filtered)			
type filter text	Visual Page Editor	\$~\$~ ~	
✓ JBoss Tools	General Templates		
✓ web ✓ Editors		☑ Show Border for Unknown Tags	
Visual Page Edi		□ 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	
	Default Editor Tab:	Visual/Source	
	Size of Visual Editor Pane 0-100%:	50% (
		Restore Defaults Apply	
0		OK Cancel	

Figure 3.47. Visual Page Editor Preferences Window

- Clicking on *Refresh* button you refresh the displayed information.
- *Page Design Options* button leads to window which helps you to specify necessary references to resources. Here is what this window looks like.

	References	to Resources	X
Page Des	ign Options		I
Actual Run-	Time Absolute Folder		
Path			Browse
Scope:	Page		~
Actual Run	Time Relative Folder		
Path			Browse
Scope:	Page		~
Included c	ss files		
Scope	CSS File Path		Add
			Edit
			Remove
Included to	ag libs		
Scope	URI	Prefix	Add
			Edit
			Remove
Substitute	d El expressions		
Scope	El Expression	Value	Add
			Edit
			Remove
		Ok	Cancel

Figure 3.48. Page Design Options

This dialog lets you set resources which are usually only resolved in runtime. Let's look at what functionality it proposes.

The first two sections of the window let you define actual runtime folders. The example below will help you to clarify how this can be used.

Suppose you have the following project structure:

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

The content of the *header.jsp* is:

My Header

and *main.jsp* content is:

<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 'projectName > WebContent > pages' and you will see the image appeared.

In the bottom part of the window you can set a path to included css files, tag libs and substituted EL expressions.

scope	CSS File Path		Add
			Edit
			Remove
cluded tag	g libs		
Scope	URI	Prefix	Add
			Edit
			Remove
ubstituted	El expressions		Remove
ubstituted Scope	El expressions El Expression	Value	Remove
ubstituted Scope	El expressions El Expression	Value	Add Edit

Figure 3.49. Bottom Part of the Page Design Options

Let' consider an example. For instance, the definition of your CSS on the page is the next:



This will work fine in runtime, but the Visual Page Editor doesn't know what *requestContextPath* in design time is. In order to see the necessary styles applied in design time your should add a path to your stylesheet in the CSS File Path section.

The next URI section lets you add URI taglibs so that the editor knows where to find the tag libraries.

And the last Substituted EL expressions section is provided to specify the values for specific EL variables. It can be useful for a preview generation.

As an example look at the figure below:

📾 hello.jsp 🕱	- 0
<pre></pre>	
html body f:view h3 h:outputText Visual/Source Source Preview	×

Figure 3.50. EL Expression

Here bath in Source and Visual modes you see the EL expression #{user.name}. When you switch to Preview view, you'll also see this expression. Now press *Page Design Options* button and set the value for the #{user.name} as World.

۲	Add El Reference
El Value	#{user.name}
	Scope: Page: Only This Page
	Folder: Any Page at the Same Folder
	Project: Any Page at the Same Project
Value	World
0	<u>E</u> inish Cancel

Figure 3.51. Setting the Value for the EL Expression

As a result in Visual mode and Preview view the word *World* is displayed.

📾 hello.jsp 🕴	- 8
<pre><f:loadbundle basename="demo.Messages" var="Message"></f:loadbundle> <html></html></pre>	
→ O Normal O Normal O B I U A	
★ World!	
html body f:view h3 h:outputText	×
Visual/Source Source Preview	

Figure 3.52. The EL Expression Value

You can find useful one more functionality provided by VPE. At the bottom of the Visual/Source view there is a Selection Tag Bar. It allows to see tags tree for a current component selected in Visual or Source mode.

• (≎ Normal ≎	Normal 🗘 B I 🗓 🔏 🛕	
X	Login		^
🤣	Please login using any username and password		
	Username #{identity.username}		
	Password ***************		
	Remember me 🔽		
			-
	Login		~
ui:co	omposition ui:define h:form rich:panel div h	:panelGrid h:outputLabel	¢
Visua	al/Source Preview		

Figure 3.53. Selection Tag Bar

If you want to hide the Selection Tag Bar, use the button in the form of a red cross on the lower right side. To reset it again you should check the proper option in the <u>VPE Preferences</u>.

3.2.5. Page Preview

VPE comes with design-time preview feature which is available for:

- Struts Pages
- JSF Pages

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

ogin.xhtml 🛙					
projectName}:	Home	We	lcome, #{identity.username}!	Login	Logout
	_				
Error Message	2S				
Please login u	sing any use	rname and password			
Username		#{identity.username}			
Password		•••••			
Remember	me	<u>v</u>			
Login					
		Powered by Seam. Generate	d by seam-gen.		
al/Source So	Irce Prev	iew			

Figure 3.54. Preview View

3.2.6. Setup notes for Linux

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

The Visual Page Editor requires the library libstdc++.so.5. This library is contained in the compatlibstdc++-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.

3.3. More Editors

Besides Visual Page Editor JBDS is supplied with a huge range of various editors for different file types: properties, TLD, web.xml, tiles, and so on.

3.3.1. 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.

Ne <u>w</u>	×	🟦 JSF Project
Go Into		🗴 Struts Project
Open in <u>N</u> ew Window		🎦 Project
Ope <u>n</u> Type Hierarchy	F4	🍄 Package
Sho <u>w</u> In	Shift+Alt+W 🕨	🞯 Class
і ⊆ору	Ctrl+C	 Interface Source Folder
Copy Qualified Name Paste <u>Q</u> elete	Ctrl+V Delete	C Folder File
Build Path	•	Faces Config
<u>S</u> ource	Shift+Alt+S 🔸	Tiles File
Refactor	Shift+Alt+T 🕨	Validation File
2 Import		TLD File
A Export		i JSP File
		M XHTML File
Run XDoclet	Shift+Ctrl+F1	M HTML File
🖑 Re <u>f</u> resh	F5	CSS File
Clo <u>s</u> e Project		🧏 JS File
Close Unrelated Projects		Properties File
Assign Working Sets		📋 E <u>x</u> ample
<u>R</u> un As <u>D</u> ebug As <u>P</u> rofile As	•	📑 Other
		2

Figure 3.55. Selecting Properties File

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

🔲 message.propertie	es 🛙	- 6
message.properties		
name	value	Add
header	Hello Demo Application	
prompt_message	Name:	Edit
hello_message	Hello	Delete
		Цр
		Do <u>w</u> n
Properties Source		

Figure 3.56. "Properties" Viewer

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

🖬 message.properties 😫	- 0
message.properties	
<pre>header=Hello Demo Application prompt_message=Name: hello_message=Hello</pre>	
	=
	~
Properties Source	

Figure 3.57. Source Viewer

3.3.2. 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

3.3.2.1. Tree view

oss Tools Tag Library E	dito	r		
html_basic		▼ Tag Library		
💠 html_basic.tld	Â	Tlibversion:	1.2	ור
Listeners		Shortname		31
🗹 Validator			···	=
👂 🧾 commandButton		URI:	http://java.sun.com/jsf/html	
👂 🧾 commandLink	=	Display-Name:		
👂 🖹 dataTable		Crossill Learns	[=
👂 📃 form		Small-Icon:		\exists
👂 🧾 graphicImage		Large-Icon:		
👂 🖹 inputHidden		Description:	his tag library contains JavaServer Faces	<u>^</u>
👂 🧾 inputSecret			UIComponent + HTML RenderKit Rend	•
👂 🖹 inputText				
👂 🧾 inputTextarea		▼ Defined		
👂 🥂 message		name	tagclass Add	
👂 🧾 messages		commandButto	com.sun.faces.taglib.html_b	5
👂 🧾 outputFormat		commandLink	com.sun.faces.taglib.html_b	
			Edit	

Figure 3.58. Tree View

3.3.2.2. Source view

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



Figure 3.59. Source View

You can easily add a new tag:

⊘ MyTLD.tld ⊠		- 8
JBoss Tools Tag Li	orary Editor	
▼ MyTLD	▼ Tag Library	<u> </u>
✓ ♦ MyTLD.tld Listeners	Rename 1.1	
Validator	New Validator	
🖹 choose	Properties //java.sun.com/jsp/jstl/xml1	
I out	Displa Tag Tag Lib	
Þ 🖪 if	Tag File	
👂 🛃 forEach		
👂 🖪 param	Large-Icon:	
👂 🖪 set	Description: My Tag Library	
When	4	
Functions	▼ Defined	
	name tagclass	<u>A</u> dd
	choose org.apache.taglibs.standard.tag.common.c	ore.Chor
	out org.apache.taglibs.standard.tag.rt.xml.Expr	Tag
	if org.apache.taglibs.standard.tag.common.x	ml.lfTag
4 111	forEach org.apache.taglibs.standard.tag.common.x	ml.ForEa
Tree Source		

Figure 3.60. Adding a New Tag

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

♦ MyTLD.tld 🛛			- 0
JBoss Tools Tag Library Edi	itor		
▼ MyTLD	▼ Tag		Ĥ
▽ ⊘ MyTLD.tld	Name:	if	
Listeners Validator	Tagclass:	org.apache.taglibs.standard.tag.common.xml.lfTa	
🔳 choose	Teiclass:	<u>B</u> rowse	
👂 📕 out	Bodycontent:	JSP 💌	=
Add Variable Add Attribute As Fra Add Attribute As Fra whe Funce You a find	agment Ctrl + 0 Ctrl + 1 Delet		_
Verify	• ray name	required Add	Ţ
Tree Source			

Figure 3.61. Adding a New Attribute

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

Ø 1	MyTLD.tld 🛿 🔗 html_basic.tld			- 0	
	<pre>?xml version="1.0" encoding="L ttagLib xmlns="http://java.sun. xmlns:xsi="http://www.w3.or xsi:schemaLocation="http:// version="2.0"> <description>My Tag Library<!--<br--><disptay-name>My Tag Lib<tlib-version>1.1</tlib-version>1.1</disptay-name></description></pre>	TTF-8" ?> com/xml/ns/j2ee" rg/2001/XMLSchema-inst java.sun.com/xml/ns/j /description> splay-name> ion>	ance" 2ee http://java.sun.co	m/xml/ns/j2ee/w	
	 description display-name function icon listener short-name tag tag-file taglib-extension 		Element : short-name Defines a simple default authoring tool to create r example, it may be used directives. Do not use wi or underscore. Data Type : string	name that could be used armes with a mnemonic f as the preferred prefix w hite space, and do not st	by a JSP /alue; for /alue in taglit art with digits
	<> tlib-version				
ſ	d				
Tree	Source				

Figure 3.62. 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 3.63. Error Reporting

3.3.3. 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

3.3.3.1. Tree View

eb XML Editor			
web	✓ Servlet		
👜 web.xml	Servlet-Name:	Faces Servlet	
Context Params Filters	Servlet-Class:	javax.faces.webapp.f	aces Browse
Listeners	Load-on-Startup:	1	
▽ 🔄 Servlets	▼ Init Params	-	
Faces Servlet:javax.faces.	param-name	param-value	Add
ୠ Faces Servlet:*.jsf	paratiti	paratiti faloc	
🞯 session-config			<u>R</u> emove
😋 Mime Mappings			Edit
🚳 welcome-file-list			
襑 Error Pages			<u>D</u> P
🖉 JSP Config			Down
🔄 Security Constraints	- Security Roles		
🧈 login-config	role-name	role-link	Add
a Security Roles			
a Env Entries			Remove
🧠 EJB			<u>E</u> dit
🔄 Services			Up
襑 Resources			
裿 Message Destinations			Down
🖉 locale-encoding-mapping-list	▼ Advanced		
	Small-Icon:		Change

Figure 3.64. Tree View

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

♦ MyTLD.tld			- 0		
Web XML Editor					
▼ web	▼ Web Des	riptor 2.5			
🔻 💩 web.xml	Mana	luch			
🕨 😋 Context F	•	🗡 Context Param			
🗟 Filters 🛛 Rename		Filter			
👂 🔄 Listeners 🖾 Change Timesta	2000	M Listener			
		Serviet			
Faces Copy	Ctrl + C	Servlet Mapping			
Saces of Cut	Ctrl + X	🕈 Mime Mapping			
😭 session-c	Ctrl + V	🕙 Error Page	<u>A</u> dd		
🏹 Mime Maj 🛛 Properties		🕈 Security Constraint	<u>B</u> emove		
a welcome-file-list		🔉 Security Role	Edit		
😂 Error Pages		🔉 Env Entry	Earch		
🖉 JSP Config		🕉 Ejb Ref	Цр		
a Security Constraints		Ejb Local Ref	Down		
🧈 login-config	* Advanced	Service Ref			
a Security Roles	Advanced	Resource Env Ref			
a Env Entries	Small-Icon:	Message Destination Ref	<u>C</u> hange		
🔄 EJB	Large-lcon:	Message Destination	<u>C</u> hange		
🔄 Services	Distributable				
a Resources	Distributable				
🍋 Message Destinations	Metadata-Co	mplete:	-		
🖉 locale-encoding-mapping-list	Encoding:		•		
Tree Source					

Figure 3.65. Adding New Elements

3.3.3.2. Source View

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



Figure 3.66. Source View

3.3.3.3. Content Assist

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



Figure 3.67. Content Assist

3.3.3.4. 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 3.68. Errors Reporting

3.3.4. 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:



Figure 3.69. CSS Editor

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

-	inpu	utUserName.jsp 💿 *stylesheet.css 🕱	- 8	Properties 🛛 🗖 🗖
	}	text-align: center;	_ _	□
	. me }	<pre>ediaobject{ padding : 5px 10px 5px 35px; </pre>		Aural Box model
	H1 }	{ MARGIN: 0px; FONT-SIZE: 22px; COLOR: #ff6600; PADDING: 45px 0pr	x 10px	Colors and Background: Fonts
	H2 }	<pre>{ MARGIN: 0px; FONT-SIZE: 18px; COLOR: #2a7bd4; PADDING: 25px 0p; </pre>	к 10рх	font font-family
	H3 }	<pre>MARGIN: 0px; FONT-SIZE: 15px; COLOR: #000000; PADDING: 20px 0px</pre>	x 10px	font-size 15px font-size-adjust
	}	MARGIN: 0px; FONT-SIZE: 12px; COLOR: #000000; PADDING: 15px 0p	к 10рх	font-stretch
	} H6	MARGIN: 0px; FONT-SIZE: 12px; COLOR: #000000; PADDING: 15px 0p	x 10px	font-weight
	}	MARGIN: 0px; FONT-SIZE: 11px; COLOR: #000000; PADDING: 5px 0px	өрх өр	 Paged media Tables
	div tex }	<pre>/.book div.section div.mediaobject{ <t-align:left;< pre=""></t-align:left;<></pre>	•	 ▷ Text ▷ User interface
	4		Þ	Visual

Figure 3.70. Properties View

3.3.5. JavaScript Editor

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



Figure 3.71. JavaScript Editor

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



Figure 3.72. Source Viewer

3.3.6. XSD Editor

JBoss Developer Studio comes with an XSD Editor for XML Schema files. This editor comes from the Web Tools Project (WTP) (see <u>WTP Getting Started</u>).

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.

@	New	×
Select a wizard		
Create a new XML schema file		
<u>W</u> izards:		
type filter text		
· · · · · · · · · · · · · · · · · · ·		
User Assistance		
👂 🗁 Web		
Web Services		
🗢 🗁 XML		
DTD		
XML		
S XML Schema		=
👂 🗁 Examples		-
	Nexts	Cancel
V <u>Back</u>	<u>Mext</u> > Emisn	Cancel

Figure 3.73. Creating New XSD file

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


Figure 3.74. Source Viewer

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

S XMLSchema.xsd 🛙						- 0
			1.0		l	
c choice		🗈 explic	tGroup)		
		a name		NCName		
		③ ref		QName		
		🚳 anyAttribute				
Γ		🕖 annotation	[01]	(annotationType)		
		e group		groupRef		
		c element		localElement		
	" <u>_</u>	🛃 choice		explicitGroup		
	0*	🛃 sequence		explicitGroup		
		🖉 any		(anyType)		
L						
Design Source						

Figure 3.75. Design Viewer

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

Figure 3.76. Edit Options

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

S XMLSchema.xsd	x							- 0
E choice				la explicitGroup				Ê
				aname		NCName	7	
				In the second		QName		
				🔞 anyAttribute				
				🛃 annotation	[01]	(annotationType)		
			Г	e group		groupRef		
				element 🕘		localElement		=
				🛃 choice		explicitGroup		
			-	🔊 sequence		explicitGroup		
				🐙 any		(anyType)		
								_
Design Source								<u></u>
Properties	_							~ - 0
③ attribute								
General	Name:	ref						Ç
Constraints	Type:	xs:QI	Name					•
Documentation	Usage:							•
Extensions	See ge.							

Figure 3.77. 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 XMLSchema.xsd 🛛 🗖 🗖	3	E Properties 🕱			~ ·	- 8
<pre><xs:attribute <="" name="maxOccurs" type="xs:allNNI" use="optional" xs:attributegroup=""></xs:attribute></pre>		③ attribute				
<pre><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></pre>		(a) attribute General Constraints Documentation Extensions	Name: Type: Usage:	ref XS:QName		

Figure 3.78. Source Viewer

3.3.7. Support for XML Schema

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



Figure 3.79. XML File

JBoss Tools Palette

This chapter will introduce you to the functionality provided by JBoss Tools Palette. The Palette allows you to quickly and easily create your JSP or JSF pages. Now you can do it more faster without additional knowledge.

The JBoss Tools Palette allows you to:

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

The JBoss Tools Palette contains a developer's project tag libraries and provides possibility to add any tag libraries to it. Also you can choose a necessary one from the list of already existed tag libraries:

- HTML
- JBoss
- JSF
- JSTL
- MyFaces
- Oracle ADF Faces
- Struts
- XHTML



Figure 4.1. Default View of The JBoss Tools Palette

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

4.1. Palette Options

To facilitate your work, you can configure the Palette in your own way, by selecting the corresponding icon on the Palette toolbar.

There is a 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

🧭 JBoss Tools Palette 😫 🦳 🗖
🕺 🖩 🥎
🕞 JBoss Ajax4jsf
🕞 JBoss RichFaces
🕞 JBoss Seam
▷ JSF Core
▷ JSF Facelets
▷ JSF HTML
🕞 JSF html 🔹 🖈
column
commandButton
commandLink
🐼 dataTable
I form
graphicImage
inputHidden
inputSecret
inputText
inputTextarea
message
🐼 messages 🔻

Figure 4.2. Palette Buttons

4.1.1. Palette Editor

JBoss Tools Palette contains existing libraries of tags, thus the Palette editor is intended to work with them or create your new one, as well.

To open the editor, click on the Palette Editor icon:



Figure 4.3. Palette Editor Icon

The window has two parts. There is a reflected grouped list of components on the left side of the palette editor. Each group is divided into multiple groups, every of which is a tag library. The right side of the palette editor is an editing window where it's possible to change values of group or tag library attributes that you've chosen on the left part of the window.

It can also be done by right click and using *Edit...* option.

For example, JSF group consists of Core, Facelets, HTML tag libraries and the attributes as name, description and hidden which are available for editing:

9		Palette Edit	or		X
▼ 🗁 XStudio	*	name	value		
👂 🆏 Icons		element type	group		
		name	JSF		
HTML		description			
IBoss	=	hidden	no		
🔻 🚔 JSF					
🕨 🐸 Core					
Facelets					
Þ 🐸 HTML					
ISTL					
MyFaces					
	•				
					_
				OK Cancel	

Figure 4.4. Tag Libraries of the JSF Group

The Palette Editor provides the following possibilities when working with existing tags or icons:

· to work with a set of icons

lcons is the root folder for the icon sets. The first step is creating the icon set. Right click on the *lcons* folder and select *Create > Create Set...*

Set the value of the name in the *Add lcons* window and click *Finish* button. A new element will appear in the list.

9	Palette	Editor	6
▼ 😂 XStudio	name	value	
≂ 🦓 Icons	element type	partition	
<mark>ک</mark> ۲ Create	•	N Create Set	
Paste	Ctrl + V		
👂 🌇 Frames 👘			
👂 鞼 JSF Core			
Lists			
👂 鞼 Struts Bean			
👂 鞼 Struts Common 🔄			
Struts HTML			
👂 鞼 Struts Logic			
			OK Cancel

Figure 4.5. Creating a Set of Icons

Also you can delete the set. Right click on the set of icons that you wish to remove and chose the *Delete Set* option from the pop-up menu or click the *Delete* keyboard button.

• to edit icons in the chosen set

When the set of icons is created, new icons can be imported to it. Choose the required set and select the option *Create > Import Icon...* from the pop-up menu that appears after you right-click on a folder.

9		Palette Edito	or		×
マ 🗁 XStudio マ 🖏 Icons ▷ 🏂 Commo	on a	ame ement type	va se	lue t of icons	
 Tonts Tonts Trames Trames Torts Torts	Create Copy Set Cut Set Paste X Delete Set TML ogic	Ctrl + Ctrl + Ctrl +	C X V te	₩업 Create Set Normal Import Icon	
				ОК	Cancel

Figure 4.6. Creating Icons

Set the name of the icon and the path and click *Finish* button.

• to work with a group of tag libraries

The first step in work with the editor is creating a group of libraries. It's very easy to do, right mouse button click on the *Palette* folder and select *Create > Create Group...*

Set a name of a group in the Create Group window and click *OK* button. A new element will appear at the end of the list.

	Palette E	ditor	×
▼ 🗁 XStudio	name	value	
🕨 🖏 Icons	element type	partition	
▼ 9 Palette		Delete-	
Create		 Create Group 	
🕨 🔒 JBoss 💼 Paste	Ctrl	+ V	
D 🔒 JSF	=		
Þ 🔒 JSTL			
MyFaces			
👂 🔒 Oracle ADF Fac			
👂 🚊 Some Group			
D 🔒 Struts			
	L		
		ОК	Cancel

Figure 4.7. Creating a Group of Tag Libraries

You are allowed to edit or delete a group, as well. If you'd like to change attributes of a group, use the right editing window of the palette editor or the *Edit...* option, like it was mentioned before. In order to remove the group, right click on the group that you wish to remove and chose the *Delete* option or click the *Delete* keyboard button.



to work with a tag library

The group maintains a list of tag libraries. If you'd like to create your own library, click right mouse button on the group and choose *Create Group...* option.

9		Palette	Editor		×
▽ 🗁 XStudio	Á	name	value		
👂 鞼 Icons		element type	group		
		name	HTML		
🕨 🚔 HTML	Croate	description			
🕨 🗎 JBoss	G Create	Group	res		
D 🗎 JSF	Edit				
D 🗎 JSTL	Copy	Ctr	1 + C		
MyFace	of Cut	Ctr	1 + X		_
👂 🗎 Oracle	💼 Paste	Ctr	1 + V		
D 📄 Some (1 Delete	D			
D 🗎 Struts	. Delete		elete		
(III					
				ок	Cancel

Figure 4.8. Creating a tag library

After setting the attribute name and the path of the icon, click *Ok* button.



You are allowed to edit or delete the tag library, as well. If you'd like to change attributes of the library or choose another icon, use the right editing window of the palette editor or the *Edit...* option. In order to remove the tag library, right click on the library that you wish to remove and chose the *Delete* option or click the *Delete* keyboard button.



· to work with a tag element

When the library folder is created, new tags can be added to it. Choose the required library and select the option *Create > Create Macro...* from the pop-up menu that appears after you right-click on a folder.

•	Palette Editor X							
▽ 🗁 XStudio	Á	name	value	2				
👂 🖏 Icons		element type	sub-	group				
▽ 🚿 Palette		name	Block	ĸ				
🗢 😑 HTML	=	icon						
🕨 🐸 Block	Edit	Ob a man a cond	to Adver	//www.w3.org/TR/REC-html40				
🕨 🐸 Core	Crea	te	•	N Create Macro				
🕨 🐸 Form	crea							
👂 🐸 Frame:	[Сору	/	Ctrl + C					
👂 🐸 Scripts	of Cut		Ctrl + X					
🕨 📂 Table	Paste	2	Ctrl + V					
👂 🐸 Text	X Dele	te	Delete					
				OK Cancel				

Figure 4.9. Creating a tag element

In the Add Palette Macro window, you can configure the tag element. Attribute *Name* is mandatory to fill and it will be the name of the tag element. Other settings are optional. You can choose the icon and set the *Start Text* and the *End Text* for your tag element. If your tag text is too long, use the *Change...* button to see it all. For *start text* and *end text* there is a possibility to control the cursor position by using "|" symbol.

9		Palette Ed	lite	or	X
▽ 🗁 XStudio	A	name	1	value	1
👂 🌇 Icons		element type		macro	
⊽ 🚿 Palette	=	name		a	
🗢 😑 HTML		icon		%lcons%/Struts HTML/link	
👂 🐸 Block		large icon			
🗢 📂 Core		description		<pre><html> </html></pre>	•
⊛ a		start text	(
address		end text			
🐼 area		automatically ref	on	yes	
body					
h1	_				
	Ű				
				OK Cancel	

Figure 4.10. Parameters of the Palette element

After all the attributes are set, click *Finish* button.



You are also allowed to edit or delete the tag. If you'd like to change the attributes of the tag or choose another icon for it, use the right editing window of the palette editor or the *Edit...* option from the pop-up menu. In order to remove the tag, right click on the tag that you wish to remove and chose the *Delete* option or click the *Delete* keyboard button.



Important:

The removal option is enabled only for custom tags. JBoss Palette tags can not be removed but can be modified.

4.1.2. 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, at the top right side of the JBoss Tools Palette.

🧭 JBoss Tools Palette 🕴 🦳 🗖
🛠 🗐 🔗
Boss Ajax4jsf
Boss RichFaces
🗁 JBoss Seam
▷ JSF Core
▷ JSF Facelets
▷ JSF HTML

Figure 4.11. Show/Hide Button

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

Show/Hide Drawers	×
Customize Palette	I
 HTML JBoss JSF JSTL MyFaces Oracle ADF Faces Struts XHTML 	Show All Hide All
Ok	Cancel

Figure 4.12. Show/Hide Drawers

If libraries are not displayed in the palette, check whether they are selected. Click the plus sign to expand the libraries of the group and make sure that a tick is put next to the wanted libraries.

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

🚿 JBoss Tools Palette 🛿 🗖 🗖
🛠 🗟 🔗
C HTML Block
Core HTML Core
C HTML Form
🗁 HTML Frames
C HTML Scripts
🔁 HTML Table
C HTML Text
🔁 JBoss Ajax4jsf
JBoss RichFaces
궏 JBoss Seam
≥ JSF Core
🔁 JSF Facelets
≥ JSF HTML
MyFaces Extensions
MyFaces Sandbox
궏 MyFaces Tomahawk
C XHTML List
XHTML Object
XHTML Structural
🔁 XHTML Table
🔁 XHTML Text
C XHTML XForms Basic

Figure 4.13. New Added Groups

The names of the elements are compound. The first part is the group name and the second is the library name.

4.1.3. Import

The Import button lets you add a custom or 3rd party tag library to JBoss Tools Palette. Find out more information on how to add particular tags see the <u>Adding Custom JSF Tags</u> section.

4.2. Using the Palette

4.2.1. Inserting Tags into a JSP File

A new tag can be added into any text file including jsp, htm, html and xhtml.

Let's do it. Open your JSP file and place the cursor in a place where you'd like to add a tag and then click that tag in the palette. In the Insert Tag window, that appears, you can set the value of *general* and *advanced* attributes of the tag that you chose.

•	Insert Tag	×		
ch:commandButton> attributes				
General Advanced				
Attribute name	Value	<u>^</u>		
action				
value		=		
accesskey				
actionListener				
alt				
binding				
dir				
disabled				
id				
image				
3		Finish Cancel		
U		Enish		

Figure 4.14. Inserting Tag

In the example below, the *commandButton* tag has been inserted.



Figure 4.15. 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.

9		Palette Edito	r	×
🕨 🐸 Core		name	value	
🗢 😂 Form		element type	macr	0
button		name	input	
checkbox	=	icon		
choosing		large icon		
fieldset		description	<htm< td=""><td>nl></td></htm<>	nl>
 file 		start text	<inpu< td=""><td>ut type=" " name=""></td></inpu<>	ut type=" " name="">
 form 		end text	-	
hidden		automatically re	efc no	
input				
input button				
)			
	-	-		
				OK Cancel

Figure 4.16. Palette Editor

Above you can see where the cursor position for HTML > Form > input is set. Thus, 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.

<pre>-ul:composition template='/templates/common.xhtml'> -ul:composition templates'/templates/common.xhtml'> -ul:composition templates/common.xhtml'> -ul:composition templates</pre>	ome.xhtml 🚾 *inputname.xhtml 🔯	- c	
<pre> <pre> </pre></pre>	<pre><ui:composition template="/templates/common.xhtml"></ui:composition></pre>		
<pre> </pre>	<pre><u::define name="pageTitle">JSF 1.2 and Facelets under Tomcat. KickStart Application</u::define></pre>		
<pre>sul:define name="body"></pre>	<pre><ui:define name="pageHeader">JSF 1.2 Hello Application</ui:define></pre>		
<pre></pre>	<ui:define name="body"></ui:define>		
<pre></pre>	<h:message <="" showsummary="true" td=""><td><pre>showDetail="false" style="color: red; font-weight: bold;</pre></td></h:message>	<pre>showDetail="false" style="color: red; font-weight: bold;</pre>	
<pre>chapt type= name='> Chapt type= name='> Chapt type= name='> Chapt type= (NUM) Chapt type: (NUM) C</pre>	<form id="hellof</td><td>'orm" jsfc="h:form"></form>		
Attribute : type The second se	<input type=" name="/>		
Immerated Values: *checkbox* *idem* *password *Tildem* *checkbox *inidem* *cradio *image* *submit *file * Tailo* * Tradio* 	Tupe : ENIIM	O "button"	
<pre>text</pre>	nerated Values :	(a) "checkbox"	
- password - checkbox - radio - radio - submit - reset - riset - file - hidden - reset - file - hidden - reset - file - hidden - reset - hidden - reset - file - hidden - reset - file - hidden - reset - file - fi	xt	(i) "file"	
-checkbox -checkbox -checkbox -radio @ 'mage" submit @ 'password" -riset @ 'radio" hidden @ 'radio"	assword	() "hidden"	
submit submit reset file indef indef indef indef indef indef	1eckbox		
- reset	ulo	@ "image"	
• file () *radio*	set	(a) "password"	
- hidden	e	(8) "radio"	
(a) reset	dden	(a) "reset"	
- image	hage		
- bullon	1001		
() "text"		(a) "text"	
(3) #{person}		@ # {nerson}	

Figure 4.17. Cursor position

4.2.2. 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.

4.2.2.1. Drag-and-Drop

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



Figure 4.18. Web Projects View

Also make sure that the JBoss Tools Palette is open. Select the tag library that you want to add and simply drag-and-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. Chosen *TLD file*, *name* and *prefix* of the library and *Library URL* are detected, thus just need to set the *Group* name to which you wish to place this tag library. You can either add this tag library to an existing Group or just create a new one.

9	Import Tags from TLD File	×
Create Macro Parent group mi	from Tags In TLD file Instance	
TLD File*	/JSFHello//META-INF/html_basic.tld	
Name:*	html	
Default Prefix	h	
Library URI	http://java.sun.com/jsf/html	
 Add to Exist Create New 	Group	ן
	OK Cancel)

Figure 4.19. Import Tags From TLD File Form

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



Figure 4.20. JBoss Tools Palette with New Tag Library

4.2.2.2. Import Button

The same you can do with *Import* button. You can see this button at the top right side of the JBoss Tools Palette.

🧭 JBoss Tools Palette 🕴 🦳 🗖
🛠 🗟 👰
Boss Ajax4jsf
Boss RichFaces
🕞 JBoss Seam
▷ JSF Core
▷ JSF Facelets
JSF HTML
▷ JSF JSTL core
🕞 Struts Bean
🕞 Struts Common
🔁 Struts Form
C Struts HTML
Struts Logic
Struts Nested
🕞 Struts Tiles

Figure 4.21. Import Button

By clicking on the *Import button* you will see the Import Tag window a similar like in the <u>Drag-and-Drop</u> method. Set the name and prefix of the library and Library URL. Also you need to set the Group name to which you'd like to add your tag library. And like in the previous method you can add it to an existing Group or create a new one. On this Import Tag form you can use *Browse...* button to locate the tag library that you want to add:

9	Import Tags from TLD File	×
Create M	facro from Tags in TLD file	
🙆 Attrit	Edit TLD	
	TLD File*	
TLD File	🗢 🗁 JSFHello	
Name:	◊ x-1_0-rt.tid	h
	⊘ c.tid	E
Default	🗇 permittedTaglibs.tld	μ
Library	♦ sql-1_0-rt.tid	h
	🗇 sql.tid	E
Ade		h
	🗇 scriptfree.tld	H
O Cre	<pre> fmt-1_0-rt.tid </pre>	μ
	<pre> fmt.tld </pre>	Ь
	🗇 fn.tld	μ
	♦ sql-1_0.tid	
	<pre> fmt-1_0.tid fmt</pre>	
	html_basic.tld	
	<pre>⟨</pre> <pre>Ist_core.tid</pre>	
	OK Cancel	

Figure 4.22. Select TLD File

4.3. RichFaces Support

JBoss Developer Studio comes with a tight integration with <u>*RichFaces component framework.*</u> RichFaces and Ajax4jsf tag libraries in <u>*JBoss Tools Palette*</u> always exist.



Figure 4.23. RichFaces Components

To start using RichFaces components as well as Ajax4jsf ones in JBDS you should first put *richfaces-*.jar* files into the */lib* folder of your project.



Note:

Currant version of JBoss Developer Studio (i. e. 1.1.0GA) includes <u>RichFaces</u> 3.2.2. The JBoss Tools 3.0.0.beta1 comes with <u>RichFaces 3.1.3</u> and partly support 3.2 version of the component framework. If you need to use the latest version of the component framework you should import it into the Palette like any other <u>custom</u> tag library.

4.3.1. Relevant Resources Links

It may be helpful for you to look through the *movie* which covers a creation of a jsf application with simple content using the RichFaces components.

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* > *Other* > *JBoss Tools Web* > *Web Projects* from the menu bar.

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

5.1. 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 5.1. Web Projects View

5.2. Drag and Drop

Web Projects View has a drag and drop option that can be used for property, managed bean attributes, navigation rules, tag library file declaration and JSP Pages.

5.2.1. 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.

```
<html>
<head>
<title>Input User Name Page</title>
</head>
<body>
<f:view>
<hl><h:outputText value=""/></hl>
```

Figure 5.2. OutputText Tag

Select the property:



Figure 5.3. 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 5.4. 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 <<u>h:outputText value=</u>"#{Message.header}"/> in the page.

5.2.2. 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 5.5. Selecting Managed Bean Attribute

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

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

Figure 5.6. Added Expression

5.2.3. Navigation Rules

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



Figure 5.7. Selecting Navigation Rule

Drag and drop it inside the commandButton tag:



Figure 5.8. Naviagation Rule in CommandButton Tag

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



Figure 5.9. Navigation Rule in Action Method

Here is how it would look after drag and drop:

```
<f:validateLength maximum="30" minimum="3"/>
</h:inputText>
<h:commandButton action="#{user.name}" value="Say Hello!" />
</h:form>
```

Figure 5.10. Inserted Navigation Rule

5.2.4. For a Tag Library File Declaration

Select a TLD file:



Figure 5.11. Selecting TLD File

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

Figure 5.12. Inserted TLD File

5.2.5. For JSP Pages

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



Figure 5.13. Creating JSP Forward

5.3. 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 Proje 🕴 😫 Package B	=x	🏂 *faces-config.xml 🕱	• 8
 ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲		/pages/inputUser/	Vame.jsp /pages/hello.jsp
Navigation Rules	New	•	Rule
Referenced Beans			
🔄 Render Kits	📌 Cut	Ctrl + X	
✓ Validators	[Copy	Ctrl + C	
S Extensions	💼 Paste	Ctrl + V	
👂 🗁 Beans		e Delete	
🕨 🗁 Tiles	- Delet	Delete	
👂 👜 web.xml	Properties		
StrutsHello	💛 Verify	/	
		Diagram Iree Source	

Figure 5.14. Creating New Navigation Rule

5.4. 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.



Figure 5.15. Expanding Tag Library File

5.5. Drag and Drop Tag Libraries on to JBoss Tools Palette

Read Adding Tag Libraries to learn about this.

5.6. 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 5.16. Web Projects View Buttons

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.

1		Preferences	×
ty	pe filter text	JBoss Tools	⇔ • ⇔∘ •
⊳	General	JBossTools preferences	
⊳	Agent Controller		
⊳	Ant		
⊳	Data Management		
	FreeMarker Editor		
Þ	Help		
	HQL editor		
Þ	Install/Update		
⊳	Java		
₽	JavaScript		
₽	JBoss jBPM		
~	JBoss Tools		
	Packaging Archives		
	⊽ Web		
	▷ Editors		
	El Variables		
	▷ JSF		
	Label Decorations		
	Seam		
	Struts		
	Verification		
	JPA		
Þ	Plug-in Development		
Þ	Profiling and Logging		
Þ	Run/Debug		
Þ	Server		
	Service Policies		
(D	ОК	Cancel

Figure 6.1. Preferences are included in this dialog.

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

Packaging Archives

- Editors
- <u>Visual Page Editor</u>
- <u>El Variables</u>
- <u>JSF</u>
- JSF Page
- JSF Propject
- JSF Flow Diagram
- <u>Seam</u>
- Seam Validator
- <u>Struts</u>
- <u>Struts Automatic</u>
- Plug-in Insets
- <u>Resource Insets</u>
- Struts Customization
- Struts Project
- <u>Struts Support</u>
- Struts Pages
- Struts Flow Diagram
- <u>Tiles Diagram</u>
- Verification

6.1. Packaging Archives

Fallow to *JBoss Tools > Packaging Archives* to open the page for changing Packaging Archives preferences.

Here you can determine settings for Project Packages view and core preferences.

-		Preferences 🗙
ty	pe filter text	📋 Packaging Archives 🛛 😓 🗸 🚽
	General Agent Controller Ant Data Management FreeMarker Editor Help HQL editor	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
	Java JavaScript JBoss jBPM JBoss Tools Packaging Archives Þ Web	
	Plug-in Development Profiling and Logging Run/Debug Server Service Policies Team Test Validation Web Web Services	
۵ ۵	XDoclet XML	Restore Defaults Apply OK Cancel

Figure 6.2. Packaging Archives

The next table lists all available preferences for Packaging Archives and their description.

Table 6.1. Packaging Archives Preferences

Option	Description	Default
Enable incremental builder	Uncheck this option if you don't want to enable incremental builder for your resources	On
Show full output path next to packages	This option allows you to show or hide an output path next to packages .	On
		On

Option	Description	Default
Show the full root directory of filesets	If on, the full root directory is displayed next to filesets. Otherwise, it's hidden .	
Show project at the root	This option allows you to choose whether to display a project name at the root of the packages or not. When checked, 'Show all projects that contain packages' is enabled .	On
Show all projects that contain packages	Selecting this setting enables the Projects Archiving view to show or hide all projects that contain packages. The option is available when the previous one is checked.	Off

6.2. Editors

To adjust settings common for all editors supplied with JBoss Developer Studio you should select JBoss Tools > Web > Editors.

2	Preferences
type filter text	Editors 🔶
 ▷ General ▷ Agent Controller ▷ Ant ▷ Data Management FreeMarker Editor ▷ Help HQL editor ▷ Install/Update ▷ Java ▷ Java ▷ JavaScript ▷ JBoss jBPM ▼ JBoss Tools 	 ✓ Always use JBoss Tools editors with Open option ✓ Show warning when project has no JBoss Tools capabilitie □ Use Source tab as a default for mutli-tab editors
 ✓ Web Editors El Variables JSF Label Decorations Seam Struts Verification 	
 Plug-in Development Profiling and Logging Run/Debug Server Service Policies 	Restore Defaults

Figure 6.3. Editors

On the Editors page the following preferences are available:

Table 6.2. Editors Preferences

Option	Description	Default
Always use JBoss		On
Tools editors with		
Open option		
		On

Option	Description	Default
Show warning when project has no JBoss Tools capabilities	Check this option to be sure that any JBoss Tools editor fully available for a particular type of file. If no, you'll be warned about this.	
Use Source tab as a default for multi-tab editors	If on, an editor will open the files in the Source view by default	Off

6.3. 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 JSF/HTML files.

1	Preferences	×
type filter text	Visual Page Editor	⇔ < ⇔ - →
▶ General	General Templates	
Agent Controller		
▶ Ant		Show Border for Unknown Tags
Data Management		Show Invisible Tags
FreeMarker Editor		Show Resource Bundles Usage as EL Expressions
▷ Help		Z Always Prompt for Tag Attributes During Tag Insert
HQL editor		Aways Fromption ag Attributes During lag insert
Install/Update		Show Selection Tag Bar
▶ Java		Always Hide Selection Bar Without Prompt
JavaScript	Default Editor Tab:	Visual/Source
JBoss jBPM		
	Size of Visual Editor Pane 0-100%:	50% <
Packaging Archives		
⊽ Web		
JSF Flow Diagram		
Struts Flow Diagram		
Tiles Diagram		
Visual Page Editor		
El Variables		
V JSF		
Label Decorations		
V Seam		
Varification		
IDA		
D Plug in Development		
 Profiling and Logging 		
Run/Debug		
▶ Server		
<		Restore Defaults Apply
Ô		OK Cancel

Figure 6.4. Visual Page Editor

The next table lists the possible settings that you can adjust on the General tab of the VPE Preferences page.

Table 6.3. VPE Preferences

Option	Description	Default
		On

Option	Description	Default
Show Border for Unknown Tags	The option allows to place the border around unknown tags or undo this	
Show Non-Visual Tags	Check this box, if you want the editor shows non-visual elements on the page you're editing	Off
Show ResourceIf the option is checked, the editor will show ELBundles Usage as ELexpressions instead of the resource valuesExpressionsexpressions instead of the resource values		Off
Always Prompts for Tag Attributes During Tag Insert	Having this option off, the dialog with possible attributes for inserting tag won't appear if all its attributes are optional	On
Show Selection Tag Bar	This option allows to show or hide the Selection Bar	On
Always Hide Selection Bar Without Prompt	Check this box if you don't want the confirmation window appears when closing the Selection Bar	Off
Default Editor Tab	The option provides with a possibility to choose one of the following views - Visual/Source, Source or Preview, as default when opening the editor	Visual/Source
Size of Visual Editor Pane 0 – 100%	With the help of this scroll bar you can adjust the percentage rating between the Source and Visual modes of the Visual/Source view	50%

On the Templates tab you can edit or remove <u>VPE templates</u>.

	Pre	ferences			×
type filter text	Visual Page Editor				⇔-⇔
▶ General	General Templates				
Agent Controller		Tes bleme	Disalau	Children	
▶ Ant		lag Name	Display	Children	Edit
Data Management			tag-name2	no	Remove
FreeMarker Editor			tag-name1	no	
▶ Help			tag-name2	no	
HQL editor			tag-name1	no	
▷ Install/Update					
▶ Java					
▶ JavaScript					
JBoss jBPM					
▼ JBoss Tools					
Packaging Archives					
∀ Web					
JSF Flow Diagram					
Struts Flow Diagram					
Tiles Diagram					
Visual Page Editor					
El Variables					
Þ js⊧					
Label Decorations					
▷ Seam					
Struts					
Verification					
JPA					
Plug-in Development					
Profiling and Logging					
▶ Run/Debug					
♦ Server					
< III >					
					
Ø				OK	Cancel

Figure 6.5. Visual Page Editor Templates

Select a template for editing from the available list and press *Edit* button. It will pick up the *Template dialog* [38] where you can adjust new settings.

6.4. El Variables

To specify necessary EL variables globally, i. e. for all projects and resources in your workspace, you should go to *JBoss Tools > Web > El Variables*.

8	Preferences	×
type filter text	El Variables	⇔ • ⇔∘ •
▶ General		
Agent Controller	Scope El Expression	Value Add
▶ Ant		Edit
Data Management		
FreeMarker Editor		Remove
▶ Help		
HQL editor		
▷ Install/Update		
⊅ java		
▶ JavaScript		
JBoss jBPM		
Packaging Archives		
⊽ Web		
Editors		
El Variables		
⊅ jsf		
Label Decorations		
▷ Seam		
Struts		
Verification		
JPA		
Plug-in Development		
Profiling and Logging		
▷ Run/Debug		
♦ Server		
Service Policies		
⊅ Team		
⊅ Test		
Validation		
⊅ Web 👻	ſ	Restore Defaults Apply
< III >	l	
0		OK Cancel

Figure 6.6. El Variables

Click *Add...* to set value for a new EL variable. In the appeared wizard you should specify the global values and press *Finish*.

	Add El Reference	×
Attribute El	Value must be set.	
El Value*	Scope: Global: For all projects in workspace	
Value		
0	Einish	Cancel

Figure 6.7. Adding a Global El Variable

Tip:
If you specify an equal variable in <u>VPE EL dialog</u> [43] and in Preference EL dialog, variable from preference dialog will have priority.

6.5. JSF

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

8	Preferences
type filter text	JSF ⇔ ⇒ →
Agent Controller	JSF Projects specific preferences.
▶ Ant	
Data Management	
FreeMarker Editor	
▶ Help	
HQL editor	
▷ Install/Update	
⊅ Java	
JavaScript	
JBoss jBPM	
Packaging Archives	
⊽ Web	
Editors	=
El Variables	
▶ JSF	
Label Decorations	
D Seam	
D Struts	
Verification	
JPA	
P Plug-in Development	
P Profiling and Logging	
P Run/Debug	
V Server	
Service Policies	_
D Test	
Validation	
D Web	
v web	
0	OK Cancel

Figure 6.8. JSF

6.6. JSF Pages

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

8	Preferences	×
type filter text	JSF Pages	⇔-⇔
Agent Controller	Blank	Add
▶ Ant	FaceletBlank.xhtml	
Data Management	FaceletCommon.xhtml	Remove
FreeMarker Editor	FaceletForm.xhtml	
▶ Help	ISFBasePage	
HQL editor	ISPRedirect	
▶ Install/Update	,	
▶ Java		
▶ JavaScript		
JBoss jBPM		
⊽ JBoss Tools		
Packaging Archives		
⊽ Web		
Editors		
El Variables		
⊽ JSF		
JSF Pages		
Project		
Label Decorations		
▷ Seam		
Struts		
Verification		
JPA		
Plug-in Development		
Profiling and Logging		
▶ Run/Debug		
▶ Server		
Service Policies		
▶ Team		
▶ Test		Set Default
<		
0	ОК	Cancel
-		

Figure 6.9. JSF Page

6.7. JSF Project

Select JBoss Tools > Web > JSF > Project to see JSF Project preferences page.

On the *New Project* tab you can set default values for <u>New JSF Project</u> wizard:

• Version for setting the default JSF Environment

- *Project Template* so as New JSF Project wizard shows this template as default for the chosen JSF Environment
- *Project Root* for specifying default location for a new JSF project

If you check Use Default Path here, this box will be also checked in the New JSF Project wizard.

• Servlet Version for setting the default Servlet version of a new JSF project

Here it's also possible to define whether to register Web Context in *server.xml* while organizing a new project or not. Check the proper box in order to do that.

Preferences X			
type filter text	Project	\$ - \$	
Agent Controller	New Project Import	Project	
▶ Ant			
Data Management	Version:		
FreeMarker Editor	Project Template:		
▶ Help		☑ Use Default Path	
HQL editor			
Install/Update	Projects Root:	/home/ochikvina/java/eclipse+tools/workspace3 Browse	
⊅ Java	Servlet Version:	2.4	
D JavaScript			
P JBoss jBPM		Register web Context in server.xmi	
Packaging Archives			
✓ Web	_		
El Variables			
✓ jor ISE Deges			
JSF Pages			
Label Decorations			
D Seam			
Struts			
Verification			
IPA			
Plug-in Development			
Profiling and Logging			
▶ Run/Debug			
▶ Server			
Service Policies			
▶ Team			
Þ Test	v	Restore Defaults Apply	
		heatore geradica	
Ø		OK Cancel	

Figure 6.10. New JSF Propject Preferences

On the *Import Project* tab in the JSF Project screen you can determine the default Servlet version for the *Import JSF Project* wizard and also whether to register Web Context in *server.xml* or not.

8		Preferences	×
type filter text	Project		⇔-⇔
♦ General	New Project Impo	ort Project	
Agent Controller			
▶ Ant	Servlet Version:	2.4	~
Data Management		Register Web Context in server.xml	
FreeMarker Editor			
▶ Help			
HQL editor			
▷ Install/Update			
∮ Java			
▶ JavaScript			
JBoss jBPM	_		
	=		
Packaging Archives			
⊽ Web			
▷ Editors			
El Variables			
⊽ JSF			
JSF Pages			
Project			
Label Decorations			
▷ Seam			
Struts			
Verification			
JPA			
Plug-in Development			
Profiling and Logging			
▶ Run/Debug			
Server	~	Restore Default	Apply
0		ОК	Cancel
		L	

Figure 6.11. Import JSF Propject Preferences

6.8. 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.

19		Preferences
type filter text	JSF Flow Diagram	n 🔶 🗘 🚽
Agent Controller	JSF Flow Diagram	Add View
▶ Ant		R show and
Data Management		V Show Grid
FreeMarker Editor	Grid Step:	16
▷ Help	Link Dath Eants	default cite=0 ctule=1
HQL editor	Link Facti Font.	
▷ Install/Update	View Path Font:	default,size=8,style=1
▷ Java		Do not create a navigation rule for a view that has no navigation case
JavaScript		Q Switch to standard control mode for surger after transition is mode.
JBoss jBPM		Switch to standard control mode for cursor after transition is made
		✓ Show shortcut icon
Packaging Archives		✓ Show shortcut path
▼ Web		
JSF Flow Diagram		
Struts Flow Diagram		
Tiles Diagram		
Visual Page Editor		
El Variables		
▼ JSF		
JSF Pages		
Project		
Label Decorations		
▷ Seam		
Struts		
Verification		
JPA		
Plug-in Development		
P Profiling and Logging		
P Run/Debug		
P Server		Restore <u>D</u> efaults <u>Apply</u>
0		OK Cancel

Figure 6.12. 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.

8		Preferences	×
type filter text	JSF Flow Diagra	n	⇔ • ⇔∘ •
Agent Controller	JSF Flow Diagram	Add View	
▶ Ant	Page Template:	ISFBasePage	~
P Data Management		()	
FreeMarker Editor	Extension:	jsp	~
V Help			
HQL editor			
P Install/Opdate			
P java			
P JavaScript			
P JBoss JBPM			
✓ JBoss loois			
Packaging Archives			
∀ Web			
✓ Editors =			
JSF Flow Diagram			
Struts Flow Diagram			
Hies Diagram			
Visual Page Editor			
El variables			
Label Decorations			
> Seam			
Verification			
verification			
Division Devisionment			
 Profiling and Logging 			
 Bun/Debug 			
b Server			
Service Policies			
> Team			
			Restore Defaults Apply
0			OK Cancel

Figure 6.13. 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.

6.9. Label Decorations

The Label Decorations page is opened from *JBoss Tools > Web > Label Decorations*.

8	Preferences	×
type filter text	Label Decorations	⇔-⇔
Agent Controller Ant	Text	
Data Management	Select decorator:	
FreeMarker Editor	Attribute	
▶ Help	🔷 File TLD	_
HQL editor	I Function	
 InstallyOpdate Iava 	🖪 Tag	
JavaScript	I Tag File	
JBoss jBPM	Validator	
	Format:	Add Variable
Packaging Archives	Preview:	
✓ Web	=	
Fl Variables		
▼ JSF		
JSF Pages		
Project		
Label Decorations		
▷ Seam		
P Struts		
IPA		
Plug-in Development		
Profiling and Logging		
▶ Run/Debug		
♦ Server		
Service Policies		
v learn D Test		
< III >>	Restore D	efaults <u>Apply</u>
0	0	K Cancel

Figure 6.14. Label Decorations

On this page you can determine the format for a text output near the decoration label for different Web resources. To change the value for selected element, click *Add Variable...* button next to *Format* field. Appeared wizard will prompt you to select one from the available list.



Figure 6.15. Label Decoration for Validator

6.10. 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:

	Preferences	×
type filter text	Seam	⇔ < ⇒ < ▼
type filter text ▷ General ▷ Agent Controller ▷ Ant ▷ Data Management FreeMarker Editor ▷ Help HQL editor ▷ Install/Update ▷ Java ▷ JavaScript ▷ JBoss jBPM ▽ JBoss Tools Packaging Archives	Seam Name Version Path ✓ jboss-seam-2.0.0.GA 2.0 /hom	
Packaging Archives		
3	L	OK Cancel

Figure 6.16. Seam

6.11. 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.

3	Preferences	
type filter text	Validator	⇔ - ⇔-
 Agent Controller Ant Data Management FreeMarker Editor 	Select the severity level for the following optional Seam Validator problems:	Configure Project Specific Setting
▷ Help	Duplicate component name:	Error
HQL editor Install/Update	Stateful component does not contain @Remove method:	Error
Þ Java	Stateful component does not contain @Destroy method:	Error
JavaScript	Stateful component has wrong scope:	Error
P JBoss jBPM	Component class name cannot be resolved to a type:	Error
Packaging Archives	Component class does not contain setter for property:	Error
⊽ Web	Entities	
Editors	Component life-cycle methods:	
El Variables	E Factories	
▷ JSF	 Bijections 	
Label Decorations	Context variables	
⊽ Seam	Expression language	
Validator Struts Verification JPA Plug-in Development	• Project Settings	
Profiling and Logging		
Run/Debug		
Server		
Service Policies		
P Team		
P lest		
Validation		
< III >		Restore Defaults Apply
0		OK Cancel
·		Cancer

Figure 6.17. Seam Validator

6.12. Struts

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

8	Preferences	×
type filter text	Struts	⇔-⇔
 type filter text Agent Controller Ant Data Management FreeMarker Editor Help HQL editor Install/Update Java 	Struts Struts Projects specific preferences.	
 JavaScript JBoss jBPM JBoss Tools Packaging Archives ✓ Web Editors El Variables JSF Label Decorations ✓ Seam Validator 	1	
 Struts Verification JPA Plug-in Development Profiling and Logging Run/Debug Server 	▼ OK	Cancel

Figure 6.18. Struts

6.13. Struts Automation

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.

8	Preferences	×		
type filter text	Automation 🔶 -	- -		
type filter text ▷ Install/Update ▷ Java ▷ JavaScript ▷ JBoss jBPM ▷ JBoss Tools Packaging Archives ▷ Web ▷ Editors El Variables ▷ ▷ JSF Label Decorations ▽ Seam Validator ▽ Struts ▷ Automation ○ Project Struts Pages Verification JPA Plug-in Development ▷ Profiling and Logging ▷ Server Service Policies	Automation Here you can modify default text for the Tiles Struts plug-in element the Validator Struts plug-in element, and error message resource	ent, files.		
▶ Team				
0	OK Cance	el		

Figure 6.19. Struts Automatic

6.14. Plug-in Insets

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

	Preferences	
type filter text	Plug-in Insets	¢
type filter text ↓ Install/Update ↓ Java ↓ JavaScript ↓ JBoss jBPM ↓ JBoss jBPM ↓ JBoss Tools Packaging Archives ↓ JSF Label Decorations ↓ JSF Label Decorations ↓ JSF Label Decorations ↓ SF Label Decorations ↓ Seam ↓ Validator ↓ Struts ↓ Automation ↓ Plug-in Insets Customization ↓ Project Struts Pages	Plug-in Insets Plug-in Insets Plug-in Insets Plug-in Insets Plug-in className="org.apache.struts.tiles.TilesPlugin"> <pre> </pre> <pre></pre>	
Verification JPA ▷ Plug-in Development ▷ Profiling and Logging ▷ Run/Debug ▷ Server	Res	tore <u>D</u> efaults
0	[ок

Figure 6.20. Plug-in Insets

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

8	Preferences	
type filter text	Plug-in Insets	\$~
▶ General	Tiles Validators	
Agent Controller	Test	
⊅ Ant	lext:	
Data Management	<pre><pre><pre><pre><pre><pre><pre>condition</pre><pre><pre><pre><pre><pre><pre><pre><</pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre>	ugin"> dator-rules.xml, WEB-INE/validation.x
FreeMarker Editor		
▷ Help		
HQL editor		
▷ Install/Update		
⊅ Java		
JavaScript		
JBoss jBPM		
▼ JBoss Tools		
Packaging Archives		
⊽ Web		
Editors		
El ∨ariables		
Þ js⊧		
Label Decorations		
▷ Seam		
Plug-in Insets		
Resource Insets		
Customization		
▷ Project		
Struts Pages		
Verification		
JPA		
Plug-in Development		
Profiling and Logging		
▷ Run/Debug		
D Server		Restore Defaults
0		ОК

Figure 6.21. Plug-in Insets of Validators

6.15. 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.

8	Preferences	
type filter text	Resource Insets	\$
▷ Install/Update ▷ Java	Default Error Messages Text:	
 JBoss jBPM ✓ JBoss jBPM ✓ JBoss Tools Packaging Archives ✓ Web ♦ Editors El Variables ♦ JSF Label Decorations ✓ Seam 	<pre>errors.required= {0} is required. errors.minlength= {0} can not be less than {1} characters. errors.maxlength= {0} can not be greater than {1} characters. errors.invalid= {0} is invalid. errors.byte= {0} must be a byte. errors.short= {0} must be a short. errors.integer= {0} must be an integer. errors.long= {0} must be a long. errors.float= {0} must be a float. errors.double= {0} must be a double. errors.date= {0} is not a date. errors.range= {0} is not in the range {1} through {2}.</pre>	
Validator ♥ Struts ♥ Automation Plug-in Insets Customization ↓ Project Struts Pages Verification JPA ↓ Plug-in Development	<pre>errors.creditcard= {0} is an invalid credit card number. errors.email= {0} is an invalid e-mail address.</pre>	
Profiling and Logging Bun/Debug		
Server		Restore <u>D</u> efaults
0		ок

Figure 6.22. Resource Insets

6.16. 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.

8			Preferences			
type filter text		Customization				\$~ \$
▷ Install/Update	^	Link Recognizer				
⊅ Java		Tag	Attribute	Refer to	Link Type	Add
JavaScript		html:link	action	action	Struts	Edit
JBoss jBPM		html:link	page	page	Struts	Ean
▽ JBoss Tools		html:link	forward	forward	Struts	Delet
Packaging Archives		html:frame	action	action	Struts	
⊽ Web		html:frame	page	page	Struts	
Editors		html:frame	forward	forward	Struts	
El Variables		html:form	action	action	Struts	
▷ JSF		logic:forward	name	forward	Struts	
Label Decorations		logic:redirect	forward	forward	Struts	
⊽ Seam						
Validator	_					
Plug-in Insets						
Resource Insets						
Customization						
Project	11					
Struts Pages						
Verification						
JPA	H					
Plug-in Development						
Profiling and Logging						
▶ Run/Debug						
♦ Server		<	1	II)	0
					Restore <u>D</u> efaults	s <u>A</u> pply
0					OK	Cance
U					UK	

Figure 6.23. Struts Customization

6.17. 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.

8		Preferences	
type filter text	Project		⇔ • ⇔∘
▷ Install/Update	New Project Import	t Project	
▶ Java	Struts Version:	Struts 1.1	
JavaScript	Strata version.	5000 1.1	
JBoss jBPM	Project Template:	Blank	
	-	☑ Use Default Path	
Packaging Archives			
⊽ Web	Projects Root:	/home/ochikvina/java/eclipse+tools/workspace3	Browse.
▷ Editors	Servlet Version:	2.3	
El Variables			
Þ jsf		Register Web Context in server.xml	
Label Decorations			
▽ Seam			
Validator			
Plug-in Insets			
Resource Insets			
Customization			
Project			
Struts Pages			
Verification			
JPA			
Plug-in Development			
Profiling and Logging			
Run/Debug			
Server			
		Restore	<u>D</u> efaults <u>Apply</u>
0			OK Cance

Figure 6.24. 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.

1	8		Preferences	
	type filter text	Project		⇔
	type filter text Install/Update Java JavaScript JBoss jBPM JBoss Tools Packaging Archives ✓ Web Editors El Variables JSF Label Decorations ✓ Seam Validator 	Project Impo Servlet Version:	rt Project 2.3 ✓ Register Web Context in server.xml	
	 ✓ Struts ✓ Automation Plug-in Insets Resource Insets Customization Project Struts Pages Verification JPA Plug-in Development Profiling and Logging 	≡		
	 Run/Debug Server 			Restore Defaults Apply
-	(7)			OK Cance

Figure 6.25. Import Struts Pages

6.18. 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.

Struts Pages

19		Preferences		
type filter text	Struts Support			⇔ • ⇔∘
▶ install/Update	Struts Support 1	.2 Struts Support 1.1 Struts Support 1.0		
▶ Java D JavaScript	Servlet Class:	org.apache.struts.action.ActionServlet		Browse
D IBoss iBPM	LIDI Dattore	la da		
	URL Pattern:	*.do		
Packaging Archives	TLD Files:	struts-bean.tld;struts-logic.tld;struts-html.tld		Browse
✓ Web				
Editors				
El Variables				
▶ ISF				
Label Decorations				
⊽ Seam				
Validator	=			
Plug-in Insets				
Resource Insets				
Customization				
Struts Support				
Struts Pages	_			
Verification				
JPA				
Plug-in Development				
Profiling and Logging				
▶ Run/Debug				
			Restore Defaults	Apply
0			ОК	Cance

Figure 6.26. Struts Support

6.19. 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.

1	Preferences	×
type filter text	Struts Pages	⇔ ⊲,
type filter text Install/Update Java JavaScript JBoss jBPM JBoss Tools Packaging Archives Veb Editors El ∨ariables JSF Label Decorations	Struts Pages Blank [default] FaceletBlank.xhtml FaceletCommon.xhtml FaceletForm.xhtml StrutsForm	
 ▷ Seam ▽ Struts ▽ Automation Plug-in Insets Resource Insets Customization ▷ Project Struts Pages Verification 		
JPA Plug-in Development Profiling and Logging Run/Debug Server Service Policies 		Set Default
0	ок	Cancel

Figure 6.27. Struts Pages

6.20. 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.



Figure 6.28. 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.

8	Preferences	
type filter text	Struts Flow Diagram	
▶ General	Struts Flow Diagram Add Page	
Agent Controller		
D Ant	Page lemplate: StrutsForm	
Data Management	Extension: jsp	
FreeMarker Editor		
▷ Help		
HQL editor		
▶ Install/Update		
⊅ java		
▶ JavaScript		
JBoss jBPM		
Packaging Archives		
∀ Web		
✓ Editors		
JSF Flow Diagram		
Struts Flow Diagram		
Tiles Diagram		
Visual Page Editor		
El Variables		
Þ jsf		
Label Decorations		
▷ Seam		
▷ Struts		
Verification		
JPA		
Plug-in Development		
Profiling and Logging		
▷ Run/Debug		
▷ Server		Pastara Dafaulta
		Restore Defaults
Ø		OK

Figure 6.29. Adding Page

6.21. Tiles 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.
Preferences			
type filter text	Tiles Diagram	, ⇒ , , , , , , , , , , , , , , , , , ,	
 General Agent Controller 	Tiles Diagram		
▶ Ant	Vertical Spacing: default(20)	`	
 Data Management FreeMarker Editor 	Horizontal Spacing: default(185)	~	
▷ Help	Definition Name Font: default,size=7,style=1	<u>B</u> rowse	
HQL editor Install/Update	Layout Animation		
P JavaScript			
D IBoss iBPM			
✓ IBoss Tools			
Packaging Archives			
⊽ Web			
JSF Flow Diagram			
Struts Flow Diagram			
Tiles Diagram			
Visual Page Editor			
El Variables			
▶ JSF			
Label Decorations			
▷ Seam			
P Struts			
Verification			
JPA b. Blug.in Douglonmont			
 Program Development Profiling and Logging 			
 Proming and Logging Bun/Debug 			
D Server			
	Restore Defa	ults <u>A</u> pply	
0	ок	Cancel	

Figure 6.30. Title Diagram

6.22. 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.



Figure 6.31. Verification

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

	Preferences	×
type filter text	Verification	⇔ → → →
▶ General	Rules Configuration Options	<u></u>
 Agent Controller Ant 	Reported Errors Number Limit: unlimited	_
Data Management		
FreeMarker Editor		
▶ Help		
HQL editor		
▷ Install/Update		
⊅ Java		
▶ JavaScript		
JBoss jBPM		=
Packaging Archives		
⊽ Web		
▷ Editors		
El Variables		
Þ jsf		
Label Decorations		
⊅ Seam		
♦ Struts	-	
Verification		
JPA		
Plug-in Development		
Profiling and Logging		
▶ Run/Debug		
♦ Server		
< III >>		
0	ОК	Cancel

Figure 6.32. Options of Verification

In summary, this document should guide you to those parts of JBoss Tools which you specifically need to develop Web Applications. It coves different aspects of visual components such as editors, views, etc. for browsing, representing and editing web resources you are working with.

If there's anything we didn't cover or you can't figure out, please feel free to visit our <u>JBoss</u> <u>Developer Studio Forum</u> to ask questions. There we are also looking for your suggestions and comments.