Visual Web Tools Reference Guide

ISBN:

Publication date:

1. Visual Web Tools 1
    1.1. Key Features of Visual Web Tools 1
    1.2. Other relevant resources on the topic 3
2. Spring Tools 4
    2.1. Spring IDE guide 4
        2.1.1. Add Spring Project Nature 4
        2.1.2. Create New Spring Project 4
        2.1.3. Add References To Other Spring Projects 4
        2.1.4. Add Spring Beans Config Files 4
        2.1.5. Create Spring Beans Config Sets 4
        2.1.6. Open Spring Explorer 4
        2.1.7. Validate Spring Beans Config 4
        2.1.8. Open Spring Beans Graph 4
        2.1.9. Search Spring Beans 4
3. Editors 5
    3.1. Editors Features 5
        3.1.1. OpenOn 5
        3.1.2. Content Assist 12
        3.1.3. Synchronized Source and Visual Editing 28
    3.2. Visual Page Editor 30
        3.2.1. Visual/Source View 31
        3.2.2. Pages Styling 40
        3.2.3. Templating 51
        3.2.4. VPE Toolbar 54
        3.2.5. Page Preview 66
        3.2.6. Support for Custom Facelets Components 67
        3.2.7. Setup notes for Linux 72
    3.3. More Editors 73
        3.3.1. Graphical Properties Editor 73
        3.3.2. Graphical TLD Editor 76
        3.3.3. Graphical Web Application File (web.xml) Editor 79
        3.3.4. CSS Editor 84
        3.3.5. JavaScript Editor 86
        3.3.6. XSD Editor 87
        3.3.7. Support for XML Schema 92
4. JBoss Tools Palette 94
    4.1. Palette Options 95
        4.1.1. Palette Editor 96
        4.1.2. Show/Hide 104
        4.1.3. Import 105
    4.2. Using the Palette 105
        4.2.1. Inserting Tags into a JSP File 105
        4.2.2. Adding Custom JSF Tags to the JBoss Tools Palette 108
5. CSS Editing Perspective 113

5.1. Outline view 114
5.2. Properties view 115
5.3. CSS Properties view 116
5.4. CSS Preview 120
6. RichFaces Support 121
6.1. Code Assist for RichFaces 121
6.2. OpenOn for RichFaces 122
6.3. RichFaces in the JBoss Tools Palette 123
6.4. Relevant Resources Links 125
7. Web Projects View 126
7.1. Project Organization 126
7.2. Drag and Drop 127
7.2.1. For a Property 127
7.2.2. For Managed Bean Attributes 129
7.2.3. Navigation Rules 129
7.2.4. For a Tag Library File Declaration 131
7.2.5. For JSP Pages 132
7.3. Developing the Application 133
7.4. Expanding Tag Library Files 134
7.5. Drag and Drop Tag Libraries on to JBoss Tools Palette 135
7.6. Create and Import JSF and Struts Projects 135
8. JBoss Tools Preferences 137
8.1. Packaging Archives 138
8.2. Editors 140
8.3. Visual Page Editor 142
8.4. El Variables 145
8.5. JSF 147
8.6. JSF Pages 148
8.7. JSF Project 149
8.8. JSF Validator 151
8.9. JSF Flow Diagram 152
8.10. Label Decorations 154
8.11. Seam 156
8.12. Seam Validator 157
8.13. Seam Pages Diagram 158
8.14. Struts 159
8.15. Struts Automation 160
8.16. Plug-in Insets 161
8.17. Resource Insets 163
8.18. Struts Customization 164
8.19. Struts Project 165
8.20. Struts Support 167
8.21. Struts Pages 168
8.22. Struts Flow Diagram 169

8.23. Tiles Diagram 171
8.24. Verification 172
8.25. Server Preferences 174
8.26. XDoclet 178
9. Context Menu Preferences and Options 182
9.1. Add/Remove Struts Capabilities 182
9.2. Add/Remove JSF Capabilities 182
9.3. Add Custom Capabilities 182
10. FAQ 184
10.1. What should I do if Visual Page Editor does not start under Linux? 184
10.2. How do I change the auto-formating preferences for the Visual Page Editor? 185
11. Conclusion 186

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: *http://www.jboss.com/products/devstudio*.

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 Tools

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.	*visual page editor*
Multiple Editors	An extensive collection of specialized editors for different file types: properties, TLD, web.xml, tiles, and so on: Graphical Properties Editor, Graphical TLD Editor, Graphical Web	*more editors*

Feature	Benefit	Chapter
	Application File (web.xml) Editor, CSS Editor, JavaScript Editor, XSD Editor, Support for XML Schema.	
JBoss Tools Palette	Organizing various tags by groups, inserting tags into a jsp or xhtml page with one click, adding custom or 3rd party tag libraries into the palette, easy controlling the number of tag groups shown on the palette.	*jboss tools 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.	*openOn*
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 dragging to source code.	*visual page editor* *drag-and-drop*
RichFaces Support	Tight integration between JBDS and *RichFaces* 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.	*RichFaces support*

Feature	Benefit	Chapter
Flexible Configuration	Various features of JBoss Developer Studio can be easily configured via the Preferences screen.	*preferences*

1.2. Other relevant resources on the topic

All JBoss Developer Studio/JBoss Tools release documentation you can find at*http://docs.jboss.org/tools* in the corresponding release directory.

The latest documentation builds are available at *http://download.jboss.org/jbosstools/nightly-docs*.

Spring Tools

JBoss Developer Studio is bundled with *Spring IDE* for Eclipse. Visit Spring IDE site for the latest versions and documentation.

2.1. *Spring IDE guide*

Spring IDE is a graphical user interface for the configuration files used by the *Spring Framework*. 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 *JSF Tools Reference Guide* and *Struts Tools Reference Guide* you had possibility to read about Graphical Editor for *JSF* and *Struts* configuration files, *Graphical Editor for Tiles Files*, *Graphical Editor for Struts Validation Files*. All these editors have *OpenOn* and *Content Assist* features, which are described in more details in this document. In addition you get to know a *Visual Page Editor* for combined visual and source editing of Web pages and many *other editors* 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.

The mentioned features are the following:

- *OpenOn*
- *Content Assist*
- *Synchronized Source and Visual Editing*

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 use *F3* or *Ctrl+Click* on a reference to another file and the file will be opened.

OpenOn is available for:

- *XML files*
- *JSP/XHTML Pages*
- Java files
- *CSS classes*
- *Paths to files set using EL variable*

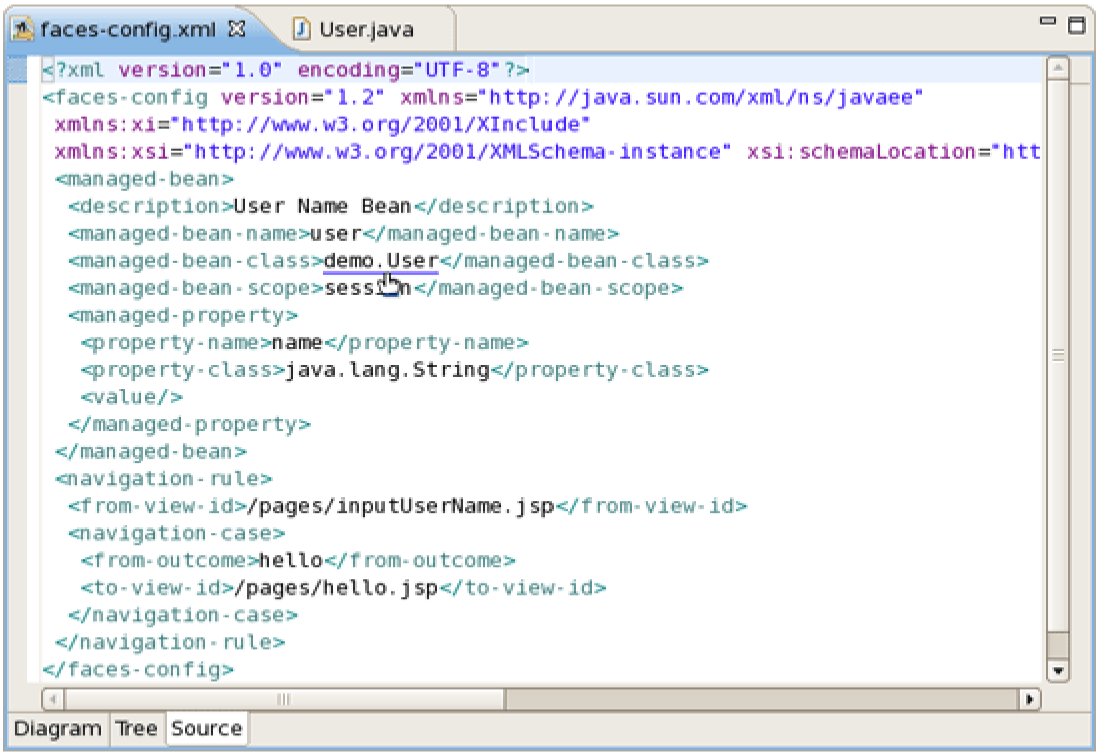
3.1.1.1. XML Files

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.

Use the OpenOn functionality for the next entries defined in XML file:

1. Managed beans

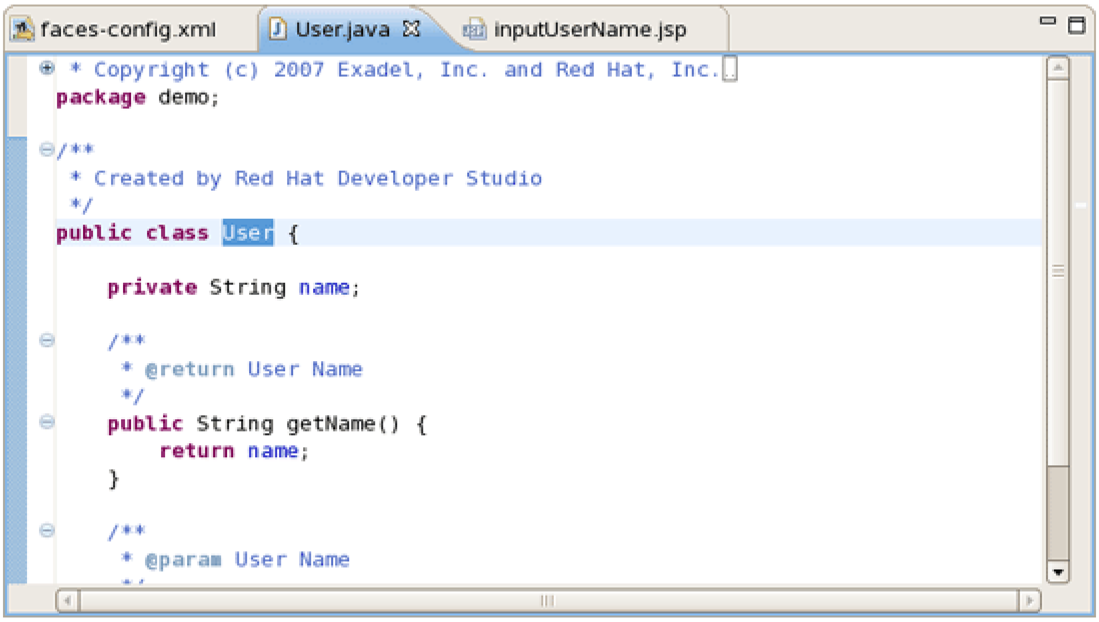
   In this example the managed bean *"User"* will open.



```
<?xml version="1.0" encoding="UTF-8"?>
<faces-config version="1.2" xmlns="http://java.sun.com/xml/ns/javaee"
 xmlns:xi="http://www.w3.org/2001/XInclude"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="htt
 <managed-bean>
  <description>User Name Bean</description>
  <managed-bean-name>user</managed-bean-name>
  <managed-bean-class>demo.User</managed-bean-class>
  <managed-bean-scope>session</managed-bean-scope>
  <managed-property>
   <property-name>name</property-name>
   <property-class>java.lang.String</property-class>
   <value/>
  </managed-property>
 </managed-bean>
 <navigation-rule>
  <from-view-id>/pages/inputUserName.jsp</from-view-id>
  <navigation-case>
   <from-outcome>hello</from-outcome>
   <to-view-id>/pages/hello.jsp</to-view-id>
  </navigation-case>
 </navigation-rule>
</faces-config>
```

Figure 3.1. Opening a Managed Bean

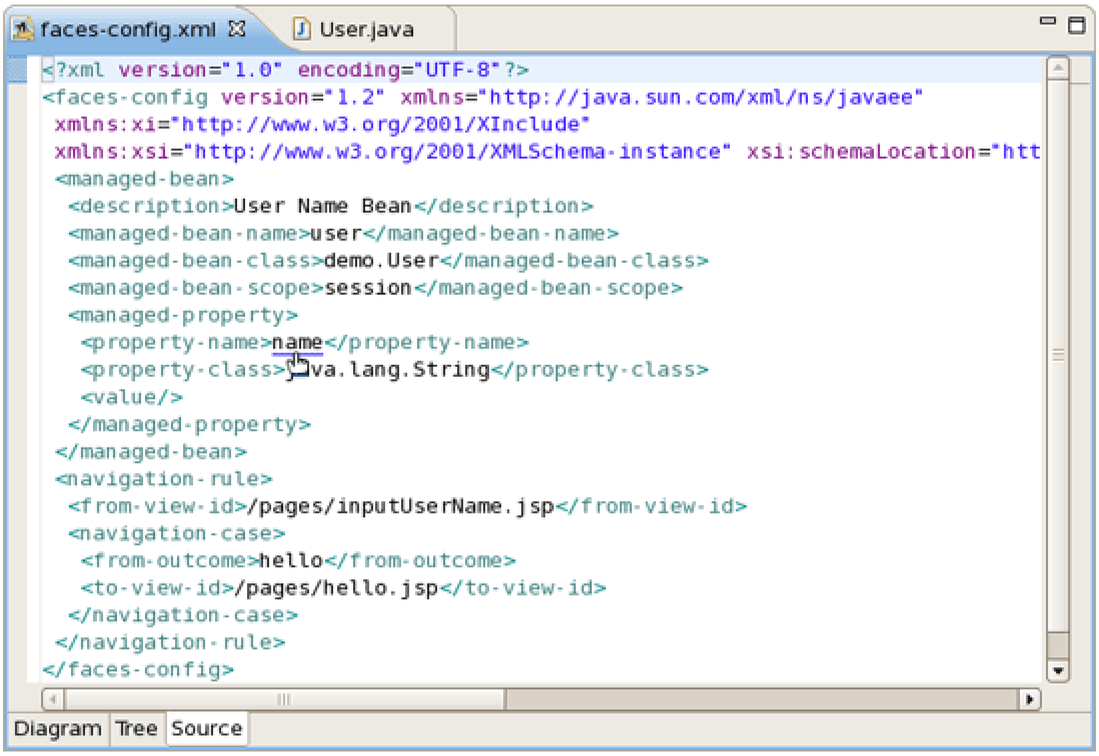
This is the result of using OpenOn.


faces-config.xml
User.java
inputUserName.jsp

```
 * Copyright (c) 2007 Exadel, Inc. and Red Hat, Inc.
package demo;

/**
 * Created by Red Hat Developer Studio
 */
public class User {

    private String name;

    /**
     * @return User Name
     */
    public String getName() {
        return name;
    }

    /**
     * @param User Name
```

Figure 3.2. Opened Managed Bean

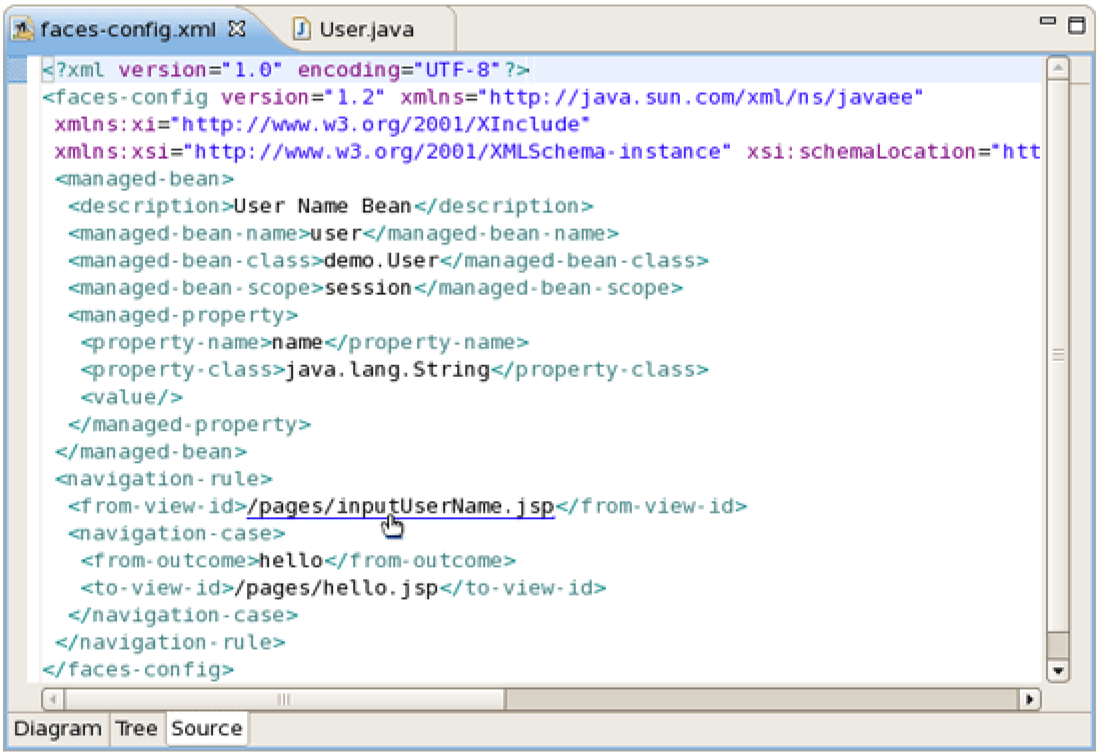
2. Beans properties


faces-config.xml
User.java
Diagram
Tree
Source

```
<?xml version="1.0" encoding="UTF-8"?>
<faces-config version="1.2" xmlns="http://java.sun.com/xml/ns/javaee"
 xmlns:xi="http://www.w3.org/2001/XInclude"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="htt
 <managed-bean>
  <description>User Name Bean</description>
  <managed-bean-name>user</managed-bean-name>
  <managed-bean-class>demo.User</managed-bean-class>
  <managed-bean-scope>session</managed-bean-scope>
  <managed-property>
   <property-name>name</property-name>
   <property-class>java.lang.String</property-class>
   <value/>
  </managed-property>
 </managed-bean>
 <navigation-rule>
  <from-view-id>/pages/inputUserName.jsp</from-view-id>
  <navigation-case>
   <from-outcome>hello</from-outcome>
   <to-view-id>/pages/hello.jsp</to-view-id>
  </navigation-case>
 </navigation-rule>
</faces-config>
```

Figure 3.3. OpenOn for the Bean Property

3. JSP file references



```
<?xml version="1.0" encoding="UTF-8"?>
<faces-config version="1.2" xmlns="http://java.sun.com/xml/ns/javaee"
 xmlns:xi="http://www.w3.org/2001/XInclude"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="htt
 <managed-bean>
  <description>User Name Bean</description>
  <managed-bean-name>user</managed-bean-name>
  <managed-bean-class>demo.User</managed-bean-class>
  <managed-bean-scope>session</managed-bean-scope>
  <managed-property>
   <property-name>name</property-name>
   <property-class>java.lang.String</property-class>
   <value/>
  </managed-property>
 </managed-bean>
 <navigation-rule>
  <from-view-id>/pages/inputUserName.jsp</from-view-id>
  <navigation-case>
   <from-outcome>hello</from-outcome>
   <to-view-id>/pages/hello.jsp</to-view-id>
  </navigation-case>
 </navigation-rule>
</faces-config>
```

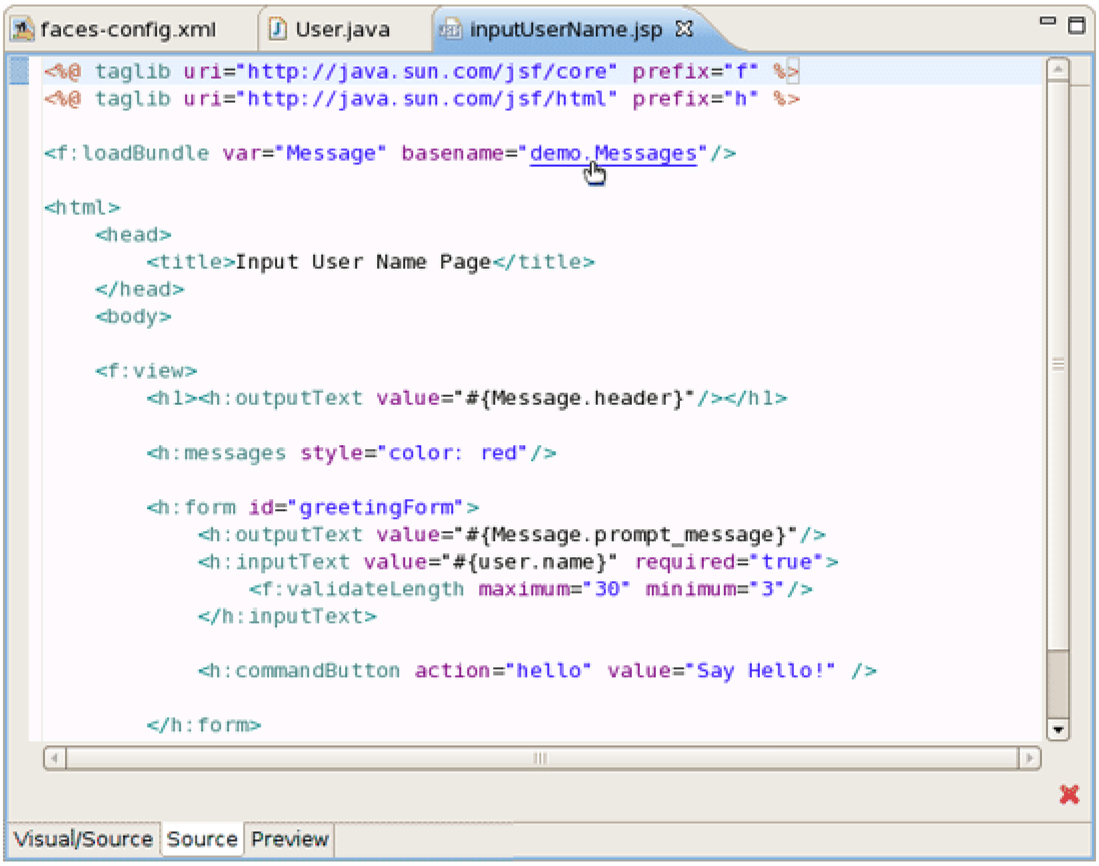
Figure 3.4. OpenOn for JSP Page

3.1.1.2. JSP/XHTML Pages

OpenOn is also available in JSP and XHTML pages edited in the Visual Page Editor. It will allow you to quickly jump to the reference instead of having to hunt around in the project structure.

You can use OpenOn for the following JSP/XHTML file entries:

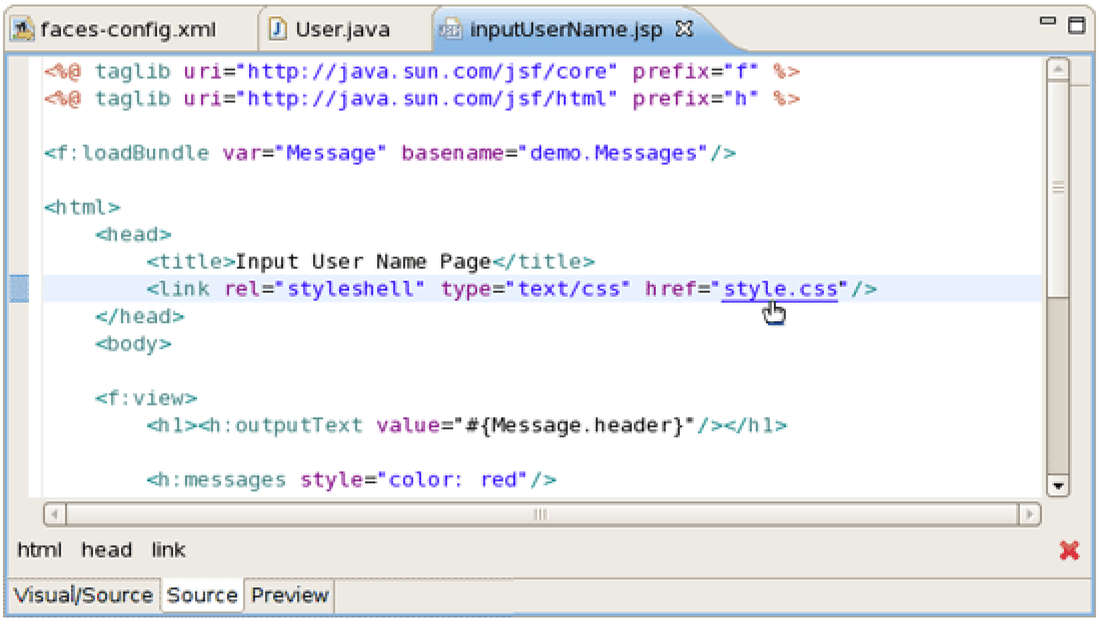
1. Imported property files


faces-config.xml
User.java
inputUserName.jsp
Visual/Source
Source
Preview

```
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>
<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>

<f:loadBundle var="Message" basename="demo.Messages"/>

<html>
    <head>
        <title>Input User Name Page</title>
    </head>
    <body>

    <f:view>
        <h1><h:outputText value="#{Message.header}"/></h1>

        <h:messages style="color: red"/>

        <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>

            <h:commandButton action="hello" value="Say Hello!" />

        </h:form>
```

Figure 3.5. OpenOn for Property File Imported to the JSP Page

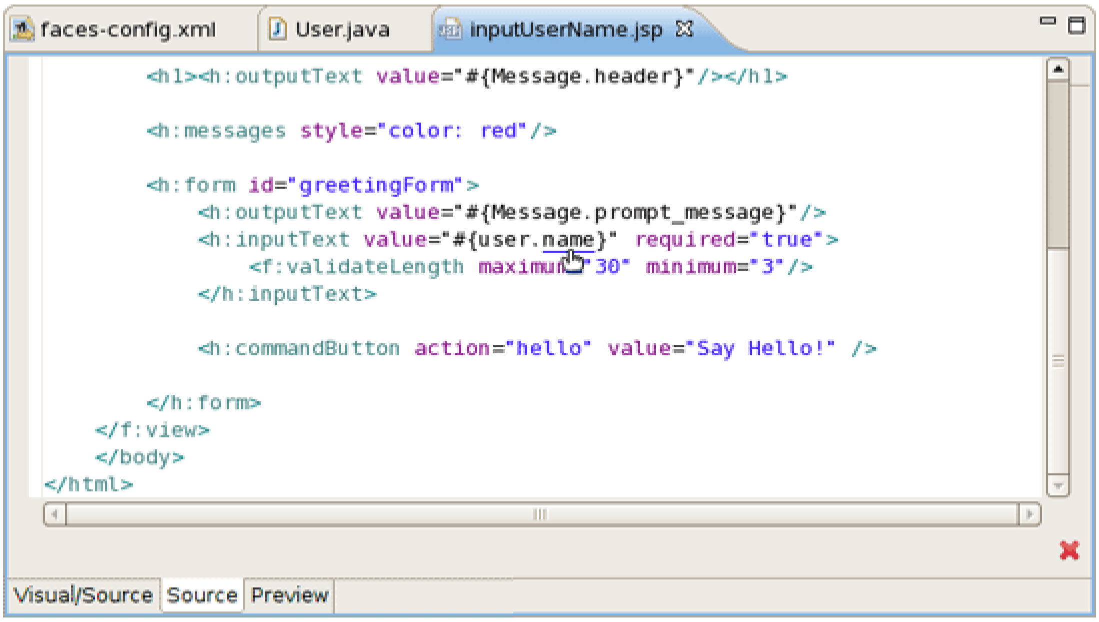
2. CSS files used in a JSP/XHTML page


faces-config.xml
User.java
inputUserName.jsp
html
head
link
Visual/Source
Source
Preview

```
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>
<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>

<f:loadBundle var="Message" basename="demo.Messages"/>

<html>
    <head>
        <title>Input User Name Page</title>
        <link rel="styleshell" type="text/css" href="style.css"/>
    </head>
    <body>

    <f:view>
        <h1><h:outputText value="#{Message.header}"/></h1>

        <h:messages style="color: red"/>
```

Figure 3.6. OpenOn With CSS File

3. Managed beans and their properties

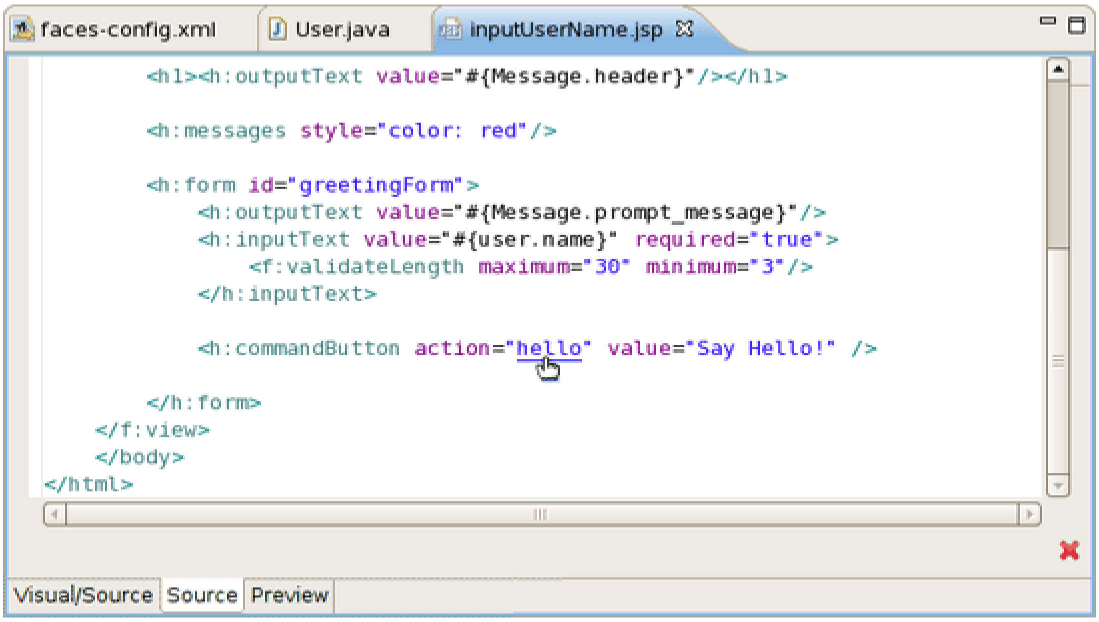

faces-config.xml
User.java
inputUserName.jsp
Visual/Source
Source
Preview

```
        <h1><h:outputText value="#{Message.header}"/></h1>

        <h:messages style="color: red"/>

        <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>

            <h:commandButton action="hello" value="Say Hello!" />

        </h:form>
    </f:view>
    </body>
</html>
```

Figure 3.7. OpenOn With Managed Beans

4. Navigation rules in JSP files

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


faces-config.xml
User.java
inputUserName.jsp
Visual/Source
Source
Preview

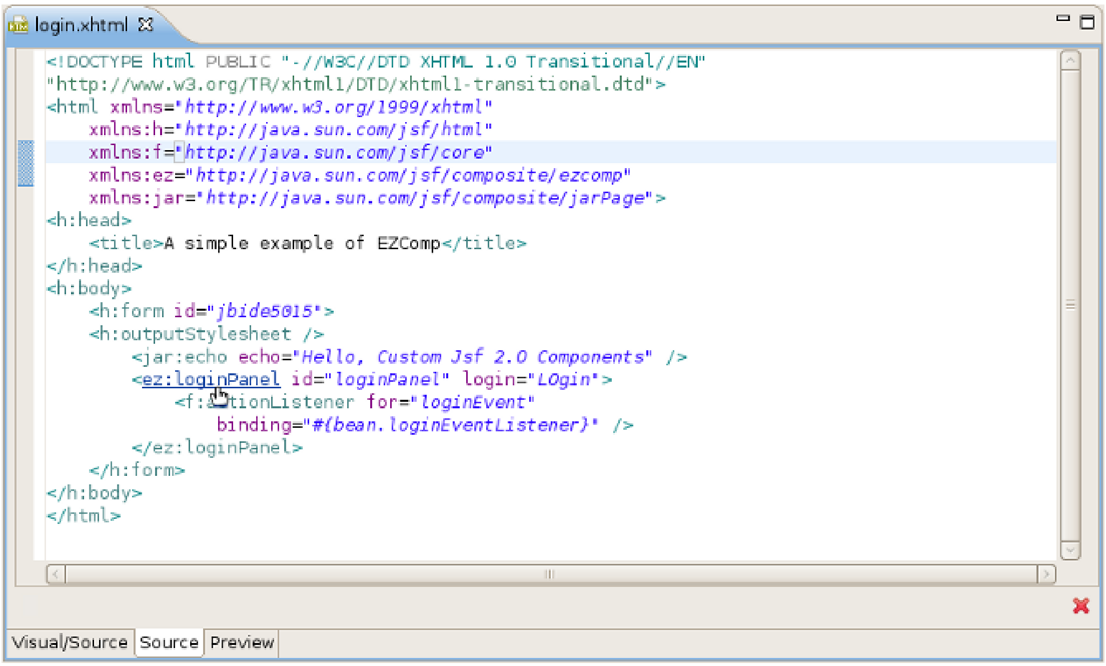
```
        <h1><h:outputText value="#{Message.header}"/></h1>

        <h:messages style="color: red"/>

        <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>

            <h:commandButton action="hello" value="Say Hello!" />

        </h:form>
    </f:view>
    </body>
</html>
```

Figure 3.8. OpenOn with JSF Tag

5. Custom Facelets tag libraries in XHTML pages

For details, see *"OpenOn for Custom Facelets Tag Libraries"* later in this guide.

6. Custom JJSF 2.0 components



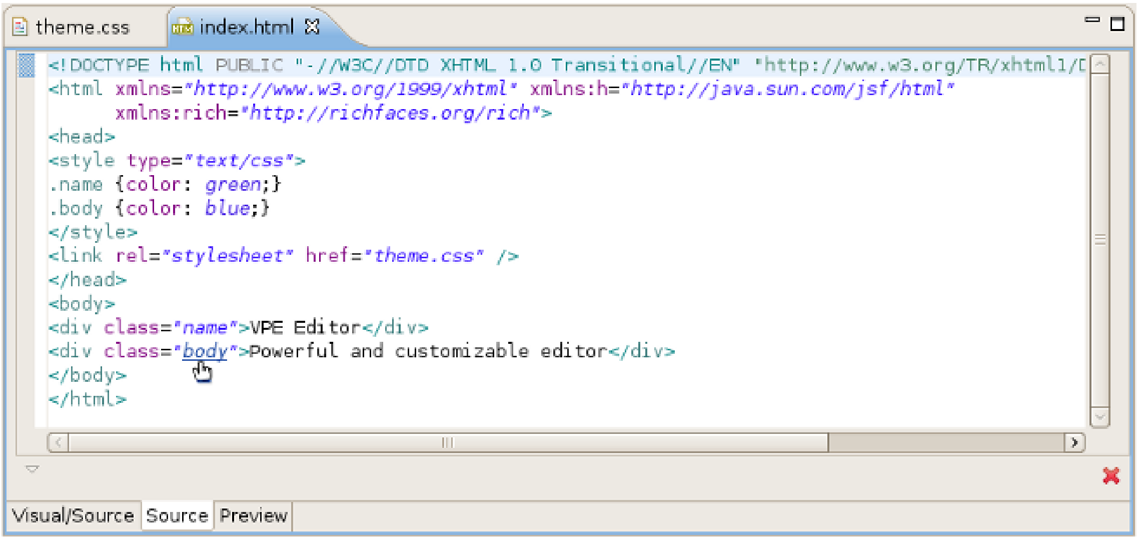
login.xhtml
Visual/Source
Source
Preview

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:ez="http://java.sun.com/jsf/composite/ezcomp"
    xmlns:jar="http://java.sun.com/jsf/composite/jarPage">
<h:head>
    <title>A simple example of EZComp</title>
</h:head>
<h:body>
    <h:form id="jbide5015">
    <h:outputStylesheet />
        <jar:echo echo="Hello, Custom Jsf 2.0 Components" />
        <ez:loginPanel id="loginPanel" login="Login">
            <f:actionListener for="loginEvent"
                binding="#{bean.loginEventListener}" />
        </ez:loginPanel>
    </h:form>
</h:body>
</html>
```

Figure 3.9. OpenOn with JSF 2.0 Component

3.1.1.3. CSS Classes

You can quickly navigate through CSS classes using OpenOn

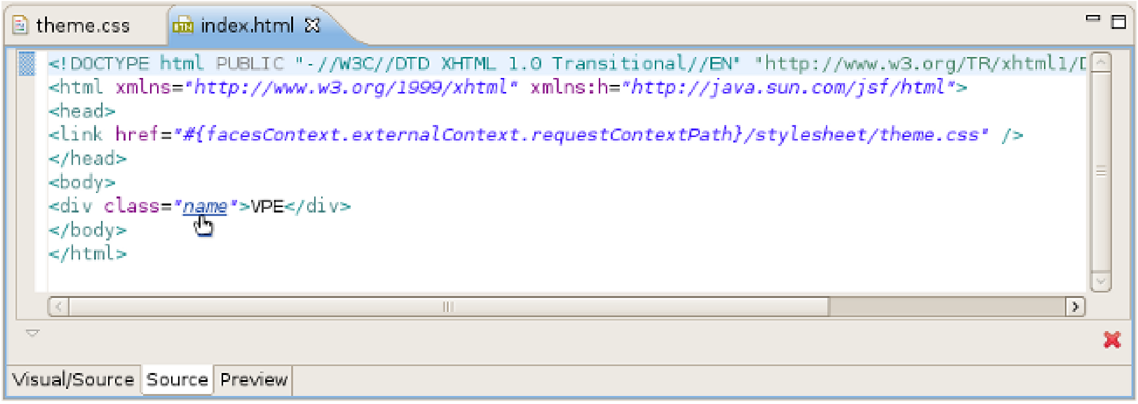


theme.css
index.html
Visual/Source
Source
Preview

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:h="http://java.sun.com/jsf/html"
      xmlns:rich="http://richfaces.org/rich">
<head>
<style type="text/css">
.name {color: green;}
.body {color: blue;}
</style>
<link rel="stylesheet" href="theme.css" />
</head>
<body>
<div class="name">VPE Editor</div>
<div class="body">Powerful and customizable editor</div>
</body>
</html>
```

Figure 3.10. OpenOn With CSS Class

OpenOn is also implemented for css classes added by a complex link.

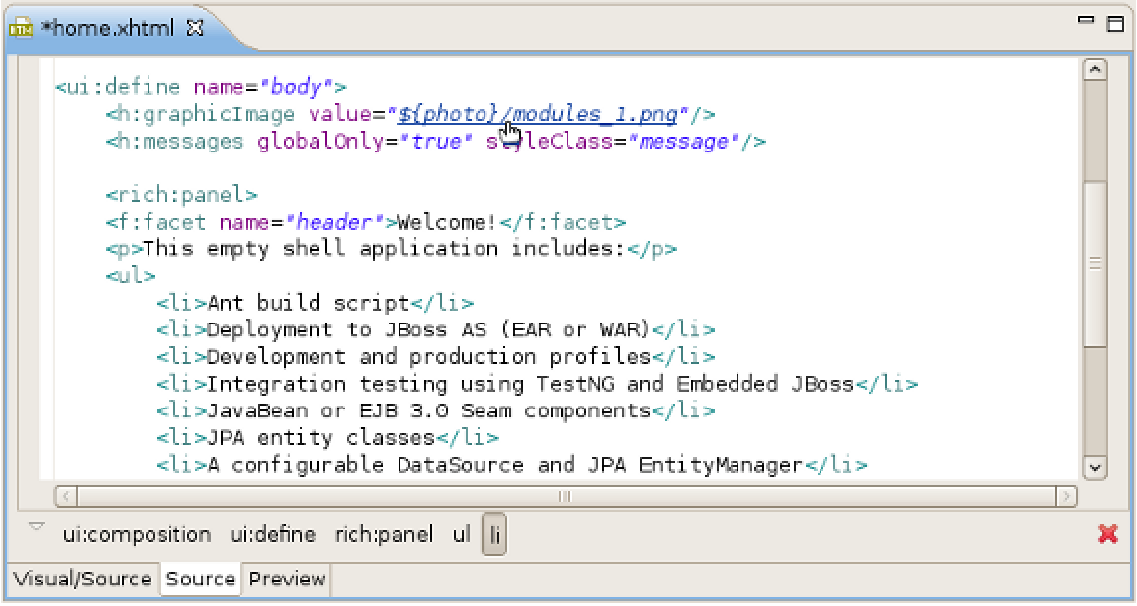

theme.css
index.html
Visual/Source
Source
Preview

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/D
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:h="http://java.sun.com/jsf/html">
<head>
<link href="#{facesContext.externalContext.requestContextPath}/stylesheet/theme.css" />
</head>
<body>
<div class="name">VPE</div>
</body>
</html>
```

Figure 3.11. OpenOn With CSS Class added by a complex link

3.1.1.4. OpenOn for EL variables

OpenOn can be used for paths to files set with EL variable.


*home.xhtml
ui:composition
ui:define
rich:panel
ul
li
Visual/Source
Source
Preview

```
<ui:define name="body">
    <h:graphicImage value="${photo}/modules_1.png"/>
    <h:messages globalOnly="true" styleClass="message"/>

    <rich:panel>
    <f:facet name="header">Welcome!</f:facet>
    <p>This empty shell application includes:</p>
    <ul>
        <li>Ant build script</li>
        <li>Deployment to JBoss AS (EAR or WAR)</li>
        <li>Development and production profiles</li>
        <li>Integration testing using TestNG and Embedded JBoss</li>
        <li>JavaBean or EJB 3.0 Seam components</li>
        <li>JPA entity classes</li>
        <li>A configurable DataSource and JPA EntityManager</li>
```

Figure 3.12. OpenOn for paths to files set with EL variable

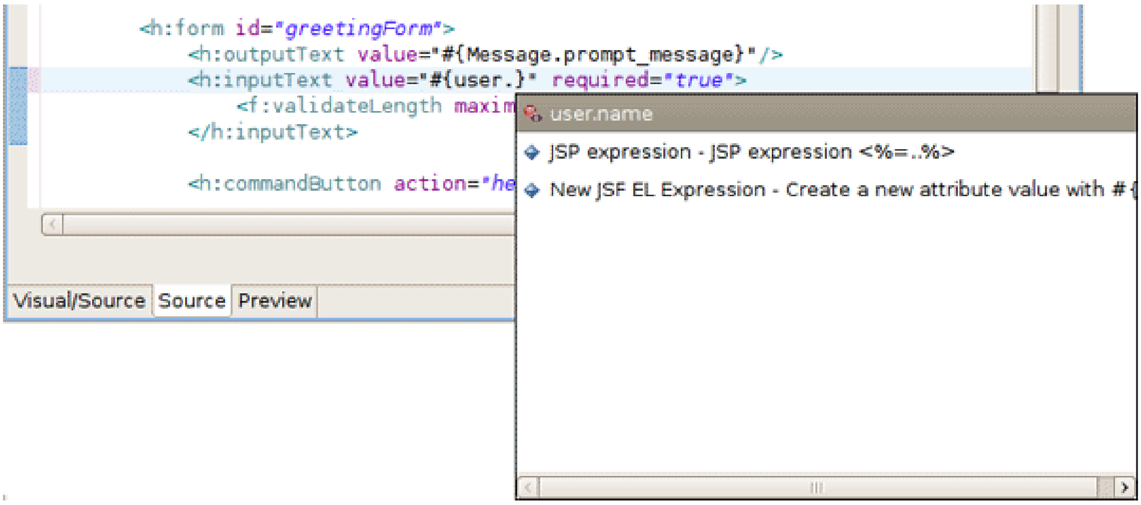
3.1.2. Content Assist

Content assist is available when working with

- *Seam project files*
- *JSF project files*
- *Struts project files*

- *JSP files*
- *XHTML files*
- *RichFaces components*
- *ESB XML files*

Notice, that code completion for EL variables has icons illustrating what they are from. Currently it's performed for resource bundles, JSF and Seam components.



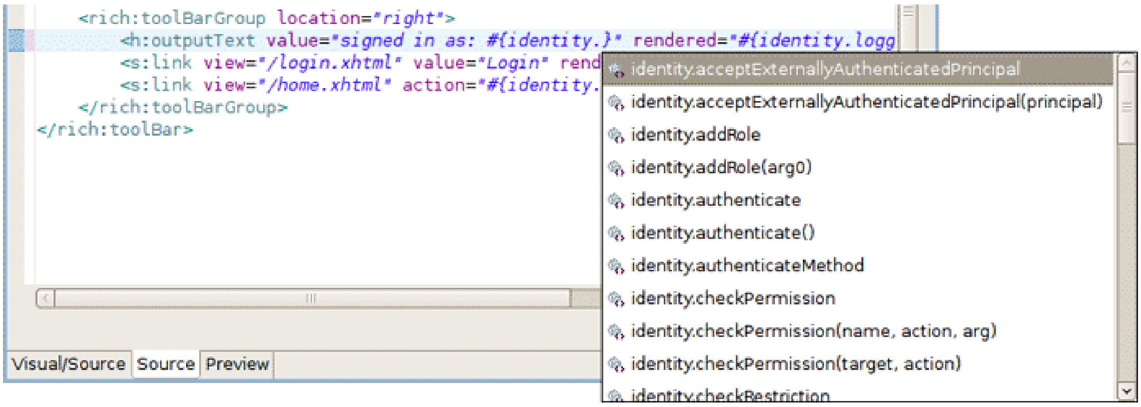
```
        <h:form id="greetingForm">
            <h:outputText value="#{Message.prompt_message}"/>
            <h:inputText value="#{user.}" required="true">
                <f:validateLength maxim
            </h:inputText>

            <h:commandButton action="he
```

user.name

JSP expression - JSP expression <%=..%>

New JSF EL Expression - Create a new attribute value with #{

Figure 3.13. JSF Content Assist



```
    <rich:toolBarGroup location="right">
        <h:outputText value="signed in as: #{identity.}" rendered="#{identity.logg
        <s:link view="/login.xhtml" value="Login" rend
        <s:link view="/home.xhtml" action="#{identity.
    </rich:toolBarGroup>
</rich:toolBar>
```

identity.acceptExternallyAuthenticatedPrincipal
identity.acceptExternallyAuthenticatedPrincipal(principal)
identity.addRole
identity.addRole(arg0)
identity.authenticate
identity.authenticate()
identity.authenticateMethod
identity.checkPermission
identity.checkPermission(name, action, arg)
identity.checkPermission(target, action)
identity.checkRestriction

Figure 3.14. Seam Content Assist

Also, as you can see, the ranking and sorting are available in EL code completions.

3.1.2.1. JSF Project Files

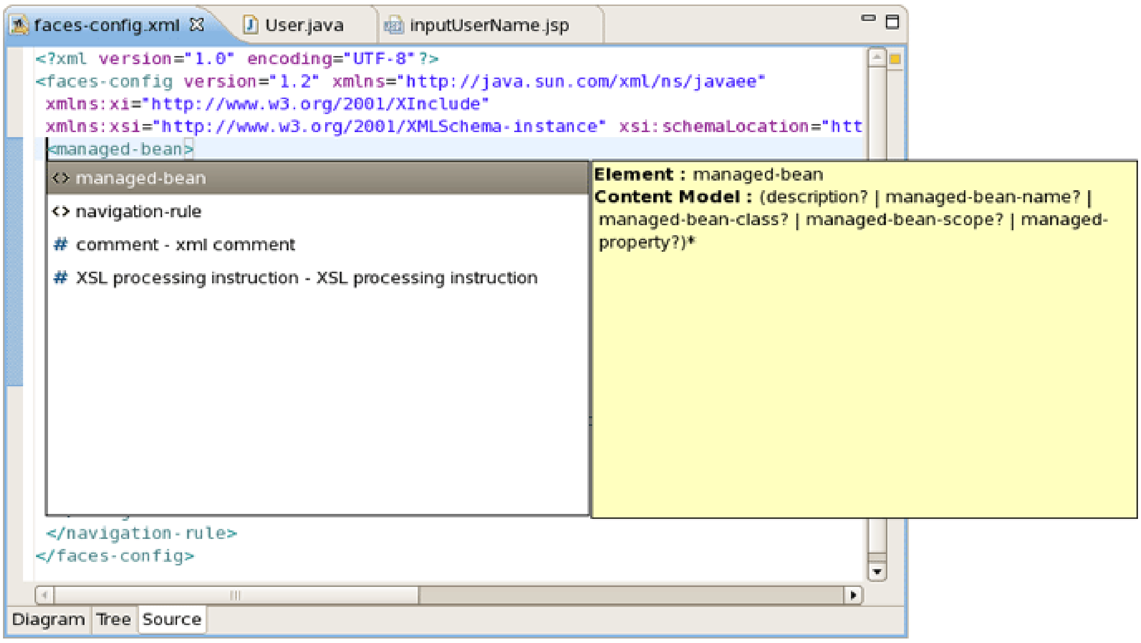
When working with JSF project in JBoss Developer Studio, you can use various Content Assist features while developing:

- Content Assist for XML, XHTML, JSP and JSF configuration files
- Content Assist for Composite Components
- 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:


faces-config.xml
User.java
inputUserName.jsp
managed-bean
navigation-rule
comment - xml comment
XSL processing instruction - XSL processing instruction
Diagram
Tree
Source

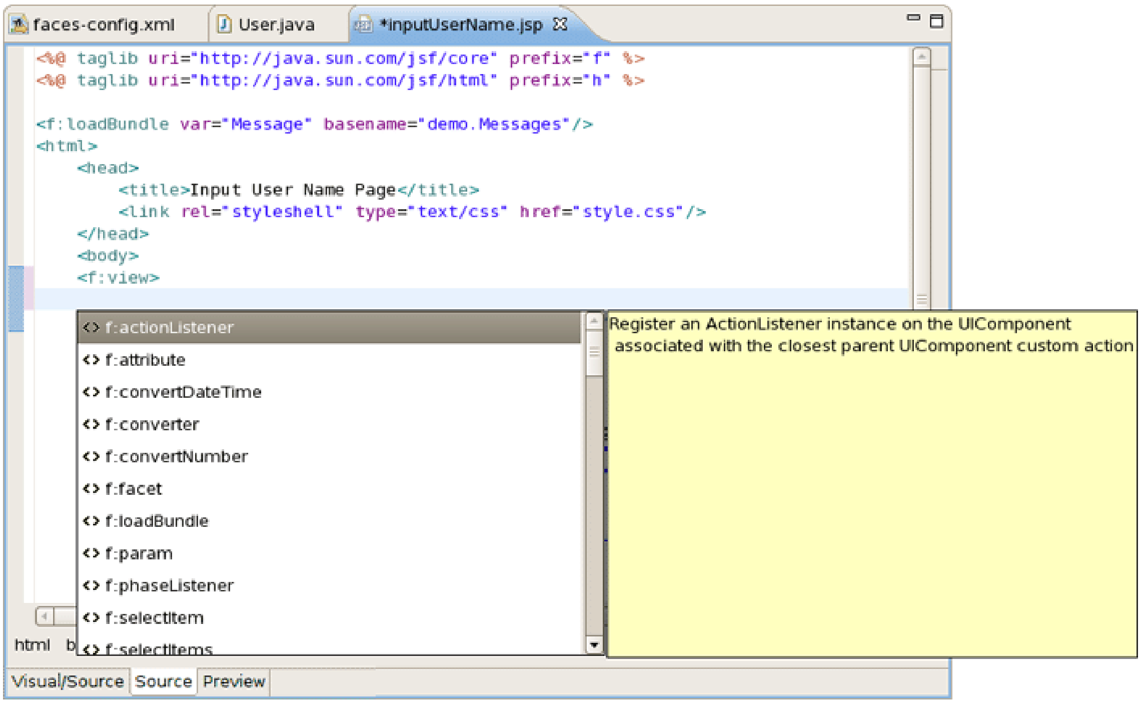
```
<?xml version="1.0" encoding="UTF-8"?>
<faces-config version="1.2" xmlns="http://java.sun.com/xml/ns/javaee"
 xmlns:xi="http://www.w3.org/2001/XInclude"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="htt
 <managed-bean>
```

```
 </navigation-rule>
</faces-config>
```

Element : managed-bean
Content Model : (description? | managed-bean-name? | managed-bean-class? | managed-bean-scope? | managed-property?)*

Figure 3.15. Content Assist in JSF Configuration File

Content Assist for JSF JSP file:


faces-config.xml
User.java
*inputUserName.jsp
html
b
Visual/Source
Source
Preview

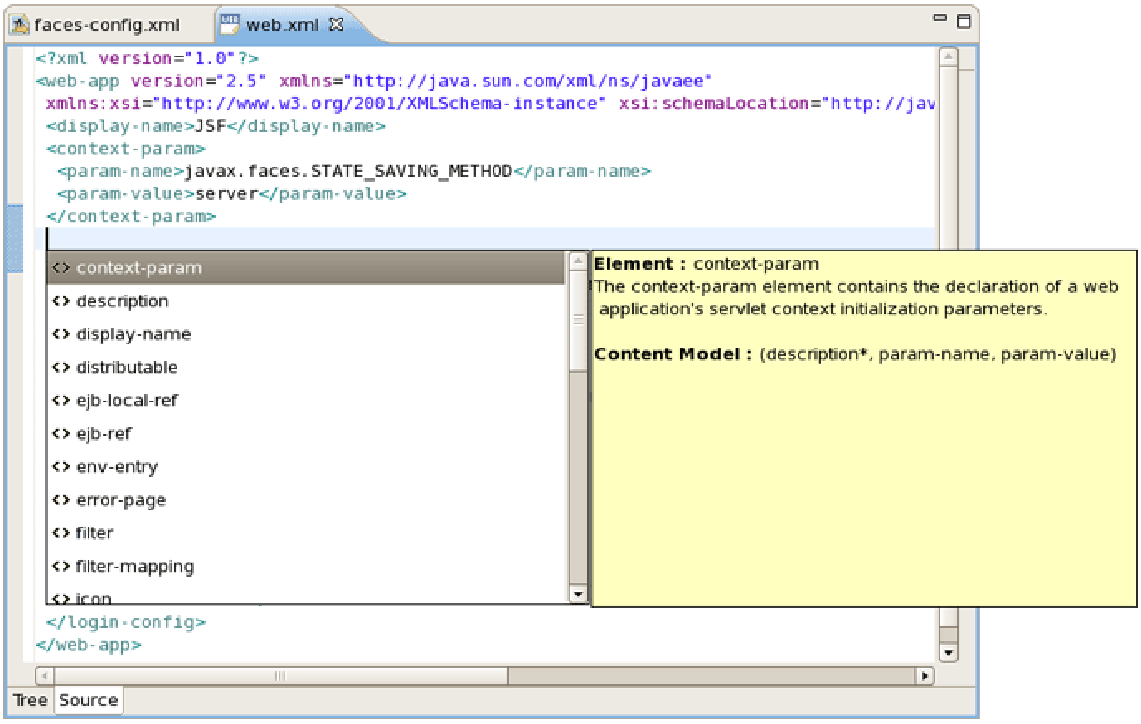
```
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>
<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>

<f:loadBundle var="Message" basename="demo.Messages"/>
<html>
    <head>
        <title>Input User Name Page</title>
        <link rel="styleshell" type="text/css" href="style.css"/>
    </head>
    <body>
    <f:view>
```

- <> f:actionListener
- <> f:attribute
- <> f:convertDateTime
- <> f:converter
- <> f:convertNumber
- <> f:facet
- <> f:loadBundle
- <> f:param
- <> f:phaseListener
- <> f:selectItem
- <> f:selectItems

Register an ActionListener instance on the UIComponent associated with the closest parent UIComponent custom action

Figure 3.16. Content Assist in JSP File

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


faces-config.xml
web.xml
Tree
Source

```
<?xml version="1.0"?>
<web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://jav
 <display-name>JSF</display-name>
 <context-param>
  <param-name>javax.faces.STATE_SAVING_METHOD</param-name>
  <param-value>server</param-value>
 </context-param>
```

- <> context-param
- <> description
- <> display-name
- <> distributable
- <> ejb-local-ref
- <> ejb-ref
- <> env-entry
- <> error-page
- <> filter
- <> filter-mapping
- <> icon

```
 </login-config>
</web-app>
```

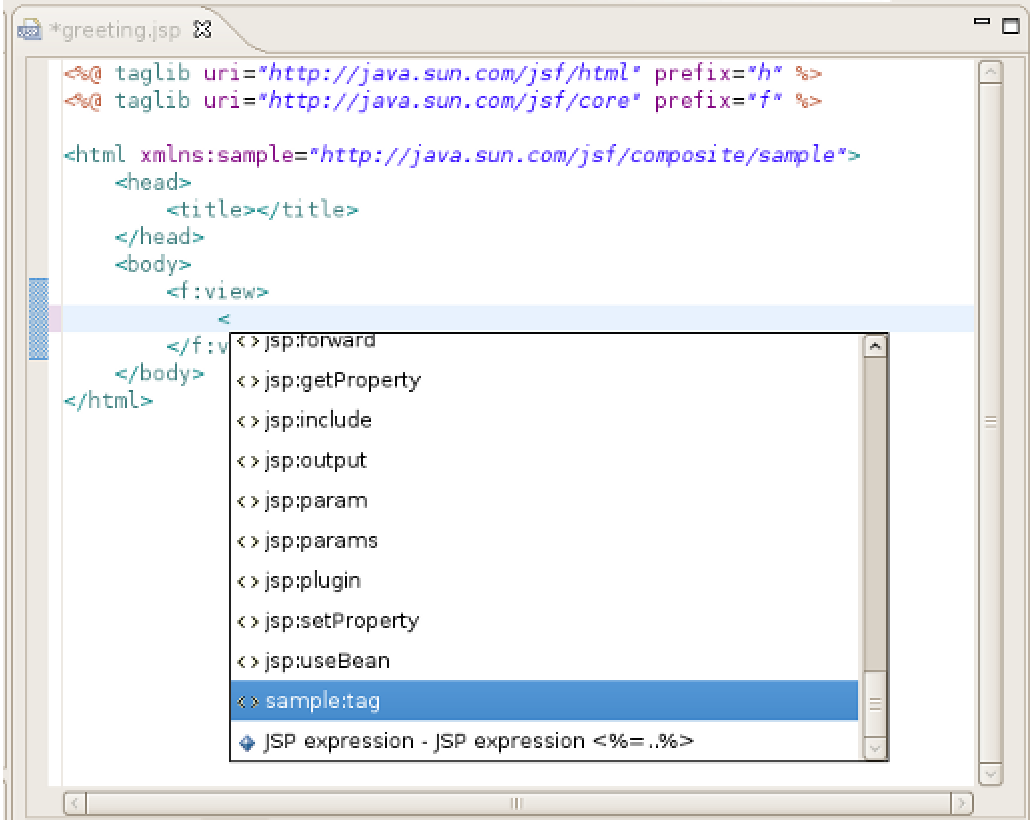
Element : context-param
The context-param element contains the declaration of a web application's servlet context initialization parameters.

Content Model : (description*, param-name, param-value)

Figure 3.17. Content Assist in web.xml File

3.1.2.1.2. Content Assist for Composite Components

Content assist functionality is also available for composite components. On the screen is shown content assist for a composite component file "tag.xhtml" within a JSF 1.2 project with facelets


*greeting.jsp
<> jsp:forward
<> jsp:getProperty
<> jsp:include
<> jsp:output
<> jsp:param
<> jsp:params
<> jsp:plugin
<> jsp:setProperty
<> jsp:useBean
<> sample:tag
JSP expression - JSP expression <%= ..%>

```
<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>

<html xmlns:sample="http://java.sun.com/jsf/composite/sample">
    <head>
        <title></title>
    </head>
    <body>
        <f:view>
            <
        </f:v
    </body>
</html>
```

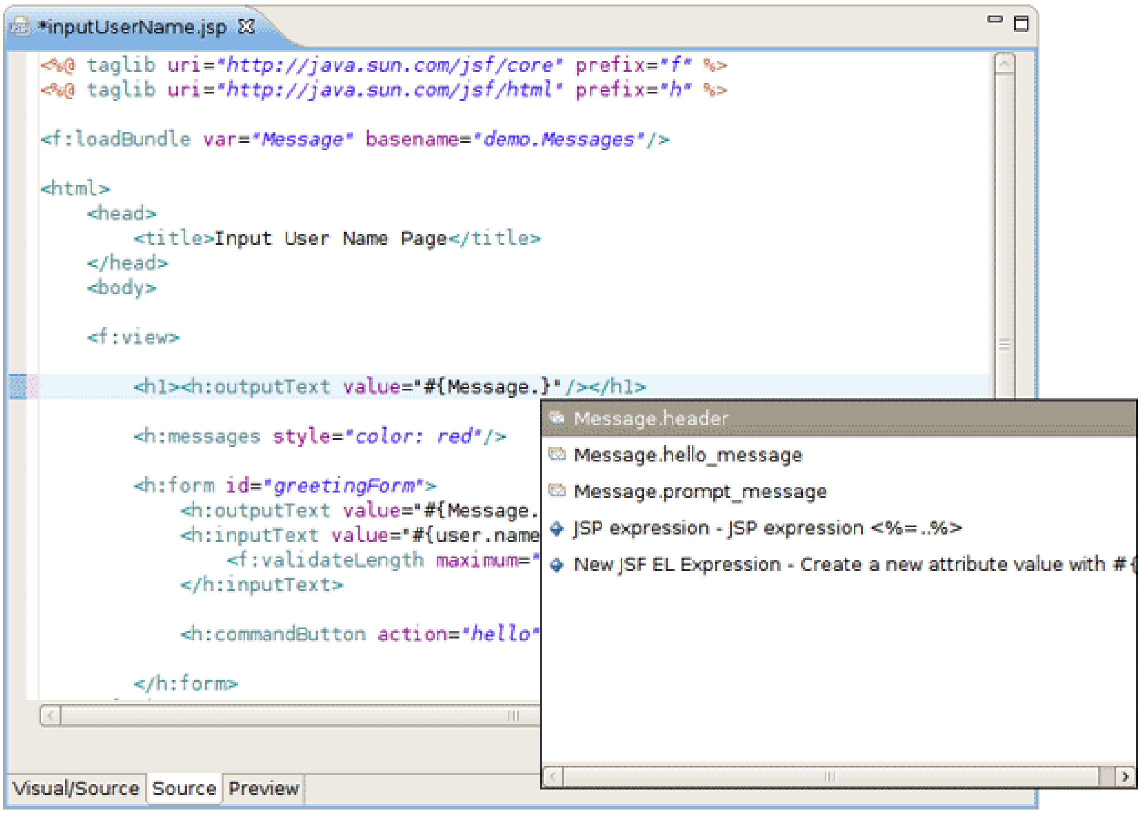
Figure 3.18. Content Assist in web.xml File

3.1.2.1.3. 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.



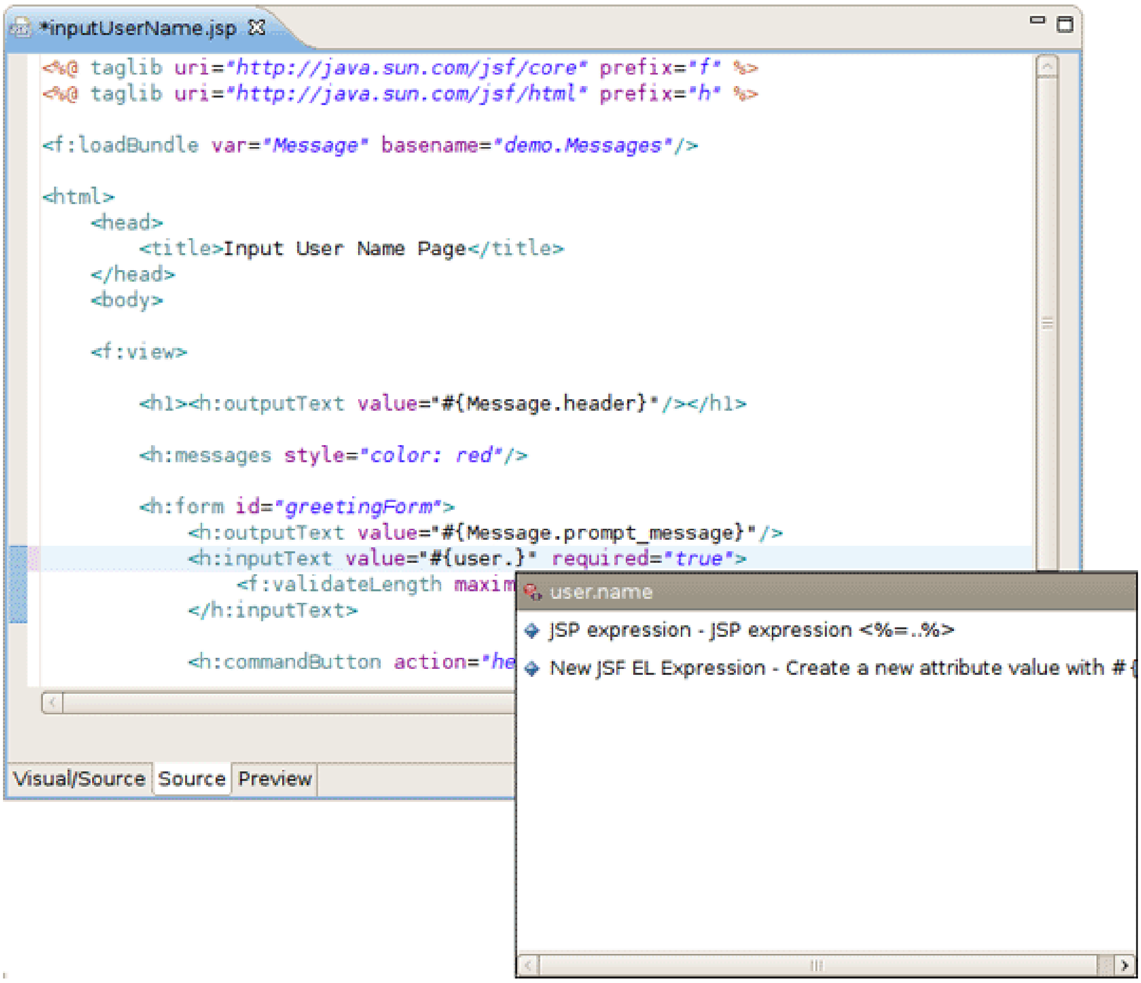
```
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>
<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>

<f:loadBundle var="Message" basename="demo.Messages"/>

<html>
    <head>
        <title>Input User Name Page</title>
    </head>
    <body>

    <f:view>

        <h1><h:outputText value="#{Message.}"/></h1>

        <h:messages style="color: red"/>

        <h:form id="greetingForm">
            <h:outputText value="#{Message.
            <h:inputText value="#{user.name
                <f:validateLength maximum="
            </h:inputText>

            <h:commandButton action="hello"

        </h:form>
```

```
Message.header
Message.hello_message
Message.prompt_message
JSP expression - JSP expression <%=..%>
New JSF EL Expression - Create a new attribute value with #{
```

Figure 3.19. 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).

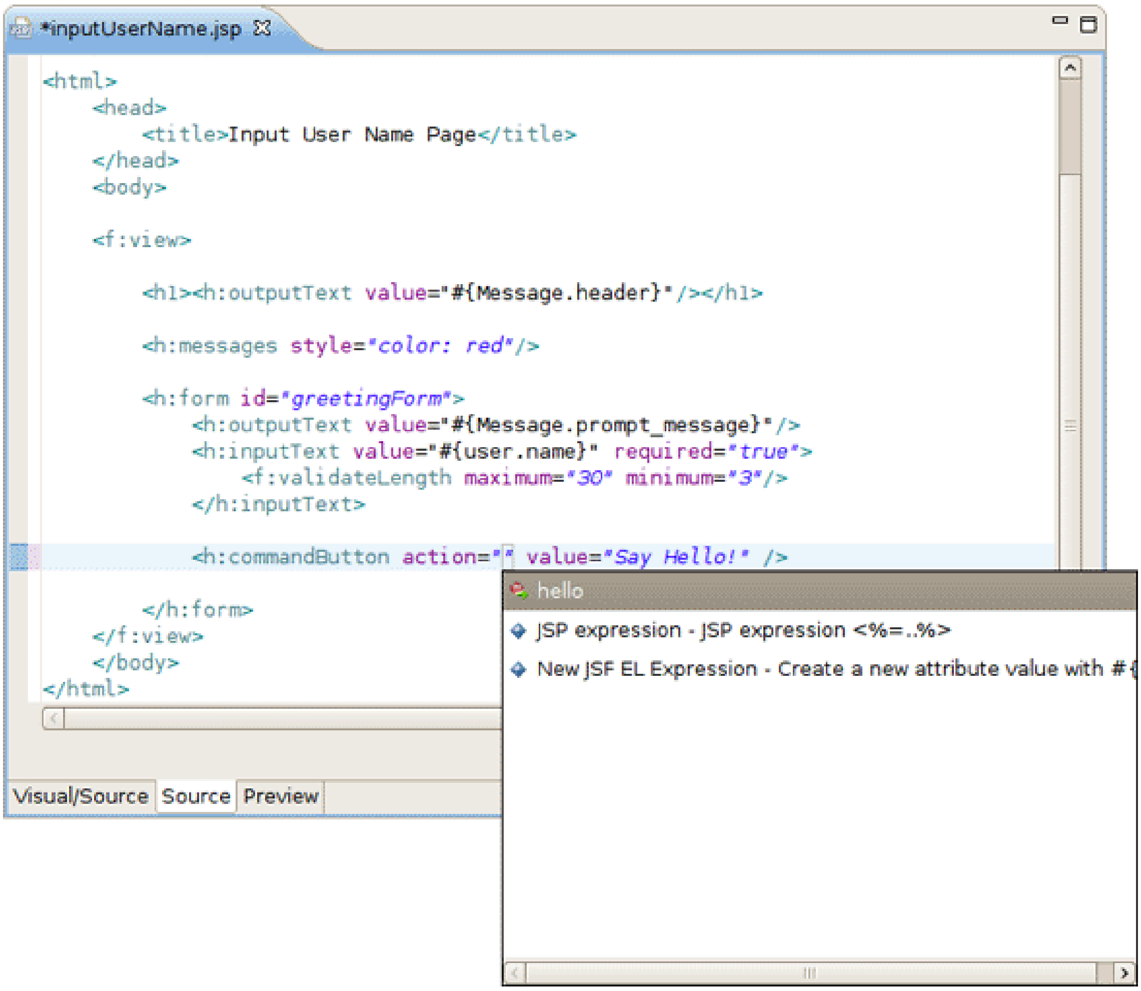

*inputUserName.jsp
Visual/Source
Source
Preview

```
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>
<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>

<f:loadBundle var="Message" basename="demo.Messages"/>

<html>
    <head>
        <title>Input User Name Page</title>
    </head>
    <body>

    <f:view>

        <h1><h:outputText value="#{Message.header}"/></h1>

        <h:messages style="color: red"/>

        <h:form id="greetingForm">
            <h:outputText value="#{Message.prompt_message}"/>
            <h:inputText value="#{user.}" required="true">
                <f:validateLength maxim
            </h:inputText>

            <h:commandButton action="he
```

```
user.name
JSP expression - JSP expression <%=..%>
New JSF EL Expression - Create a new attribute value with #{
```

Figure 3.20. Attributes List

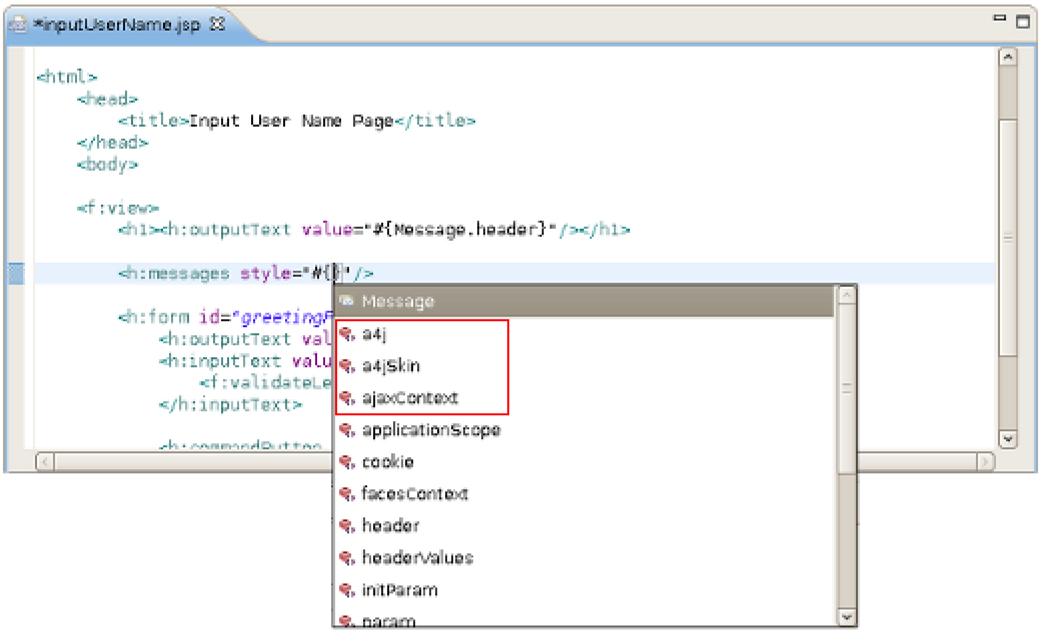
Code Assist based on project data will also prompt you for navigation rules that exist in your JSF configuration file.


*inputUserName.jsp
hello
JSP expression - JSP expression <%=..%>
New JSF EL Expression - Create a new attribute value with #
Visual/Source
Source
Preview

```
<html>
    <head>
        <title>Input User Name Page</title>
    </head>
    <body>

    <f:view>

        <h1><h:outputText value="#{Message.header}"/></h1>

        <h:messages style="color: red"/>

        <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>

            <h:commandButton action="" value="Say Hello!" />

        </h:form>
    </f:view>
    </body>
</html>
```

Figure 3.21. Code Assist

Code Assist can also provide you with access to the beans located in jar archives.

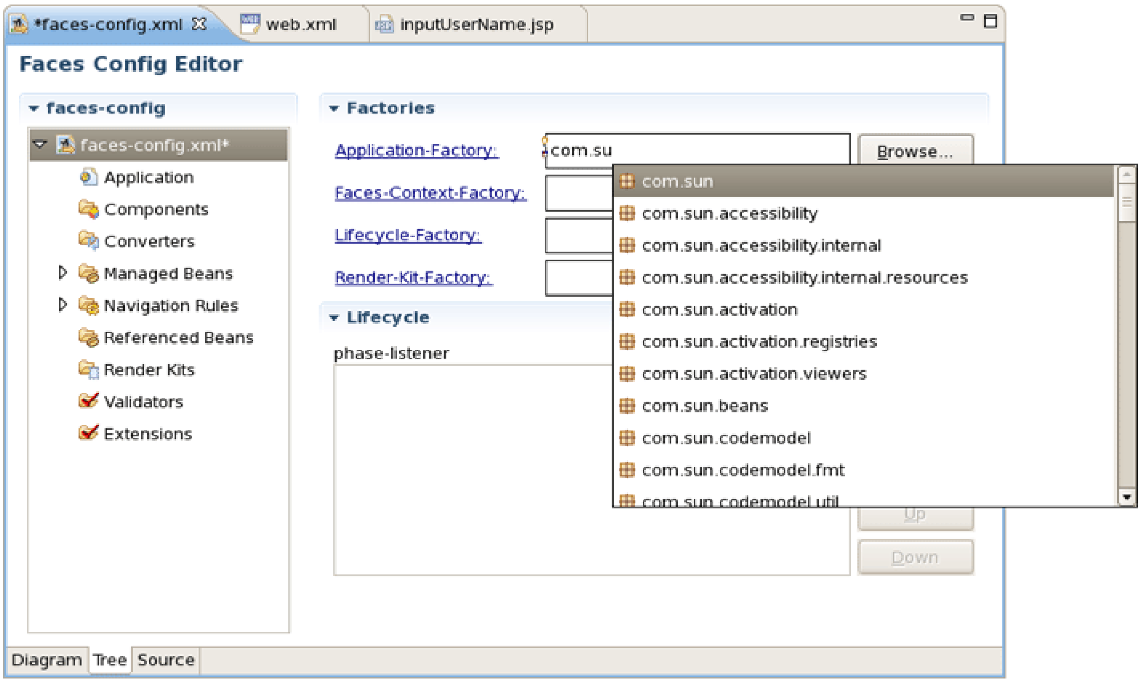

*inputUserName.jsp
Message
a4j
a4jSkin
ajaxContext
applicationScope
cookie
facesContext
header
headerValues
initParam
param

```
<html>
    <head>
        <title>Input User Name Page</title>
    </head>
    <body>

    <f:view>
        <h1><h:outputText value="#{Message.header}"/></h1>

        <h:messages style="#{}"/>

        <h:form id="greetingF
            <h:outputText val
            <h:inputText valu
                <f:validateLe
            </h:inputText>
```

Figure 3.22. Code Assist: accessing beans in jar archives

3.1.2.1.4. 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.xml
inputUserName.jsp
Faces Config Editor
faces-config
faces-config.xml*
Application
Components
Converters
Managed Beans
Navigation Rules
Referenced Beans
Render Kits
Validators
Extensions
Factories
Faces-Context-Factory:
Lifecycle-Factory:
Render-Kit-Factory:
Browse...
com.sun
com.sun.accessibility
com.sun.accessibility.internal
com.sun.accessibility.internal.resources
com.sun.activation
com.sun.activation.registries
com.sun.activation.viewers
com.sun.beans
com.sun.codemodel
com.sun.codemodel.fmt
com.sun.codemodel.util
Lifecycle
phase-listener
Up
Down
Diagram
Tree
Source

Application-Factory: com.su

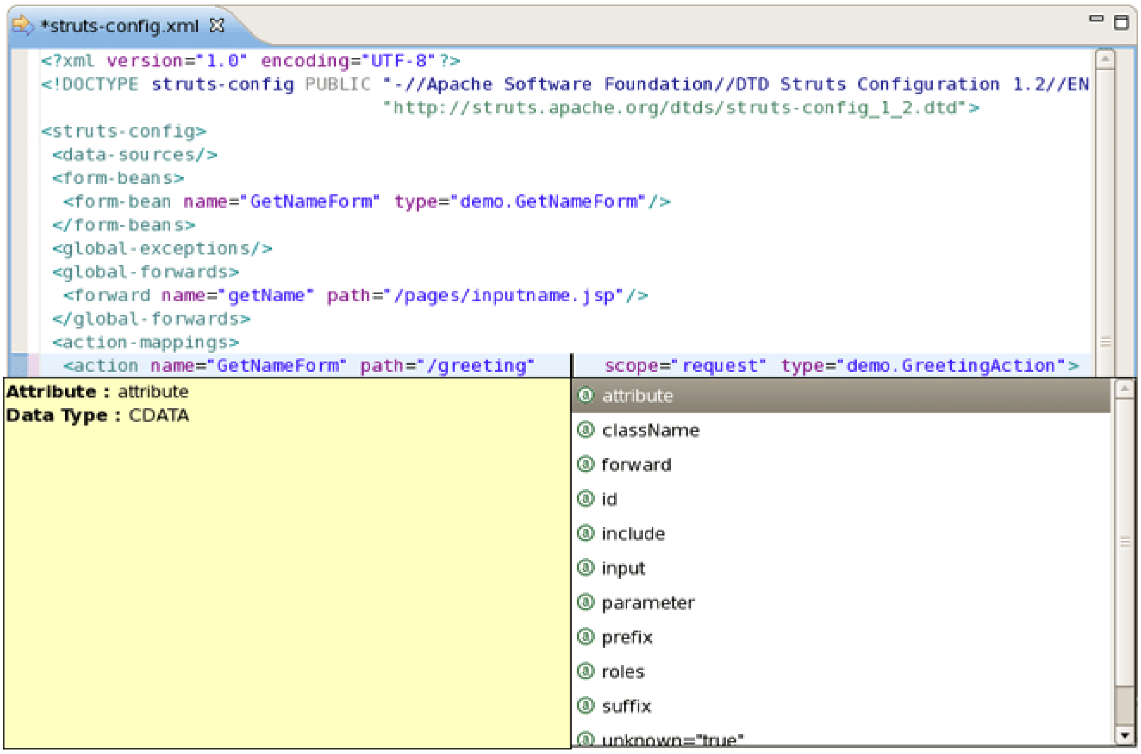
Figure 3.23. Content Assist in Tree JSF Configuration Editor

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

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts-config PUBLIC "-//Apache Software Foundation//DTD Struts Configuration 1.2//EN
                              "http://struts.apache.org/dtds/struts-config_1_2.dtd">
<struts-config>
 <data-sources/>
 <form-beans>
  <form-bean name="GetNameForm" type="demo.GetNameForm"/>
 </form-beans>
 <global-exceptions/>
 <global-forwards>
  <forward name="getName" path="/pages/inputname.jsp"/>
 </global-forwards>
 <action-mappings>
  <action name="GetNameForm" path="/greeting"    scope="request" type="demo.GreetingAction">
```

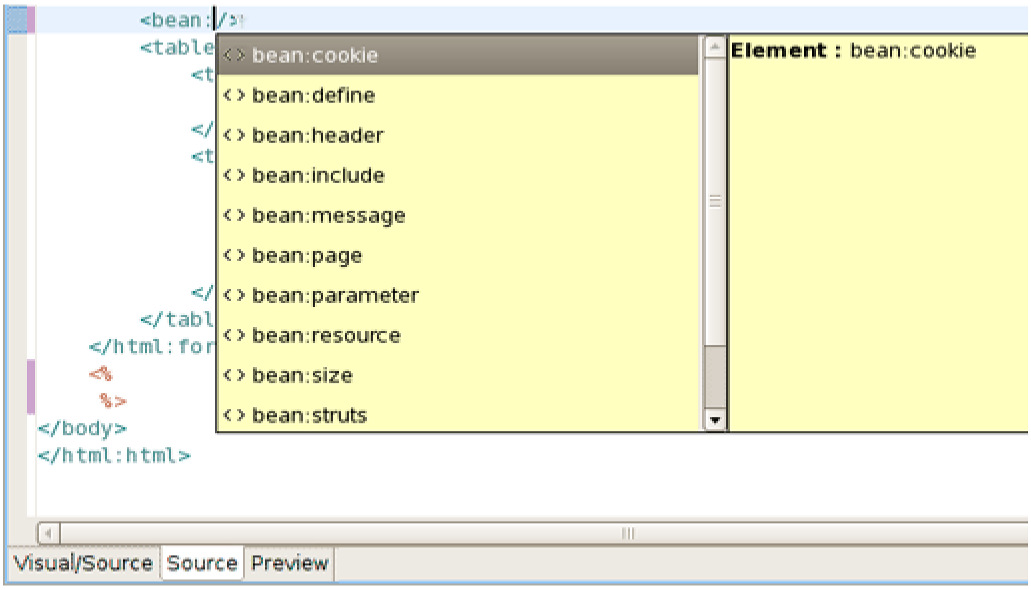
Attribute : attribute
Data Type : CDATA

attribute
className
forward
id
include
input
parameter
prefix
roles
suffix
unknown="true"

Figure 3.24. 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:|/>
        <table
            <t

            </
            <t



            </
        </tabl
    </html:for
    <%
     %>
</body>
</html:html>
```

- bean:cookie
- bean:define
- bean:header
- bean:include
- bean:message
- bean:page
- bean:parameter
- bean:resource
- bean:size
- bean:struts

Element : bean:cookie

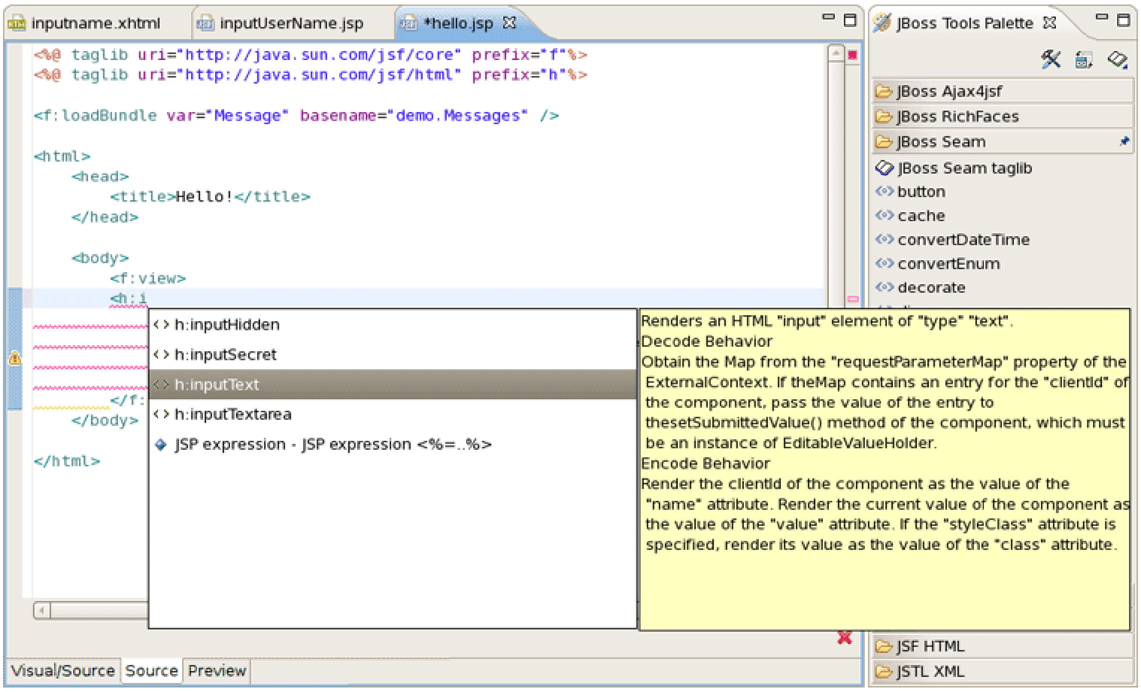
Visual/Source
Source
Preview

Figure 3.25. 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:



```
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f"%>
<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h"%>

<f:loadBundle var="Message" basename="demo.Messages" />

<html>
    <head>
        <title>Hello!</title>
    </head>

    <body>
        <f:view>
        <h:i

        </f:
    </body>

</html>
```

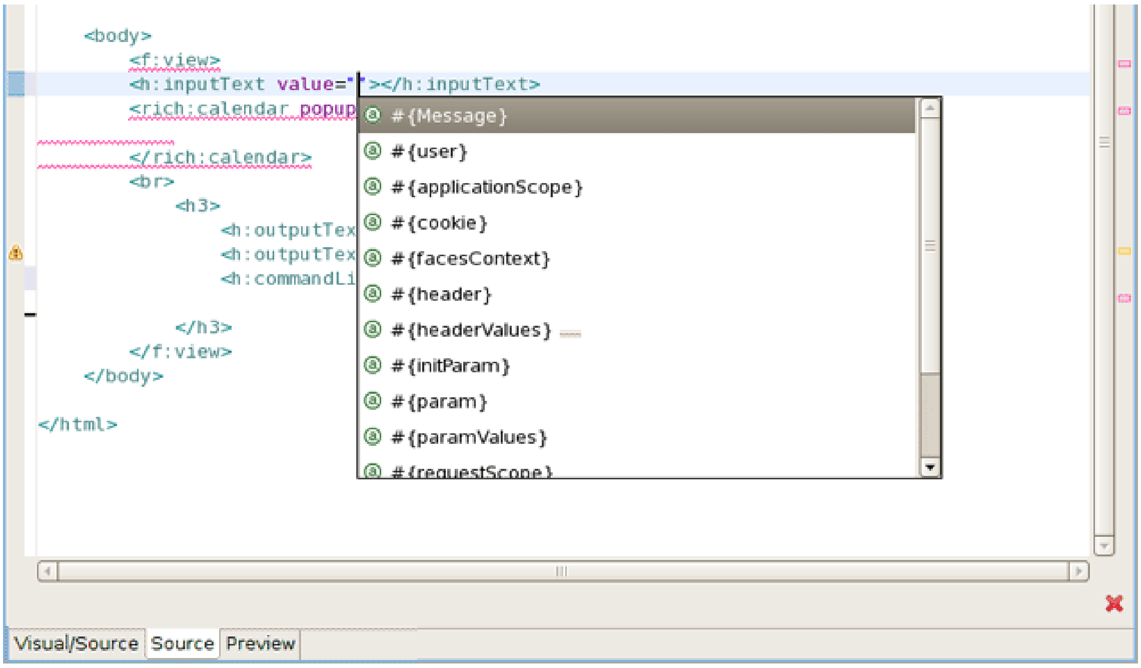
- h:inputHidden
- h:inputSecret
- h:inputText
- h:inputTextarea
- JSP expression - JSP expression <%=..%>

Renders an HTML "input" element of "type" "text".
Decode Behavior
Obtain the Map from the "requestParameterMap" property of the ExternalContext. If theMap contains an entry for the "clientId" of the component, pass the value of the entry to thesetSubmittedValue() method of the component, which must be an instance of EditableValueHolder.
Encode Behavior
Render the clientId of the component as the value of the "name" attribute. Render the current value of the component as the value of the "value" attribute. If the "styleClass" attribute is specified, render its value as the value of the "class" attribute.

inputname.xhtml
inputUserName.jsp
*hello.jsp
JBoss Tools Palette
JBoss Ajax4jsf
JBoss RichFaces
JBoss Seam
JBoss Seam taglib
button
cache
convertDateTime
convertEnum
decorate
JSF HTML
JSTL XML
Visual/Source
Source
Preview

Figure 3.26. 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.

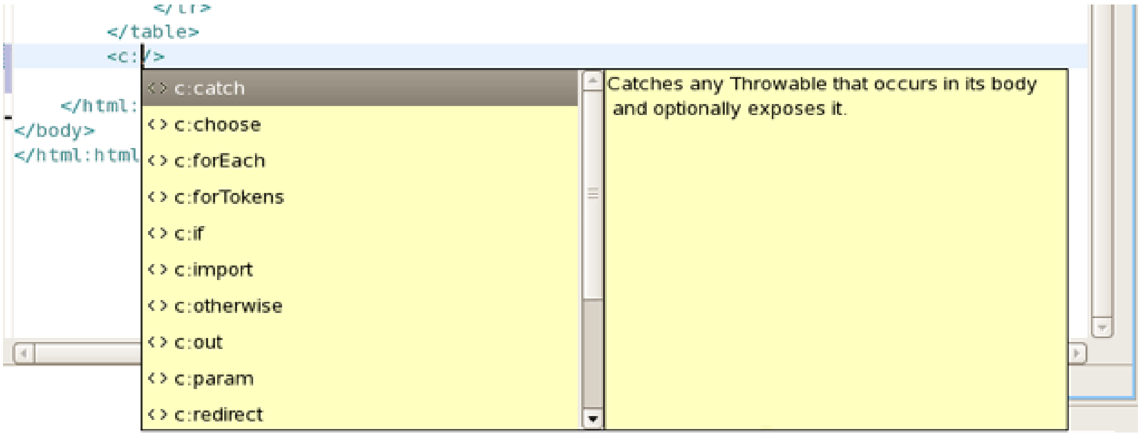


```
    <body>
        <f:view>
        <h:inputText value=""></h:inputText>
        <rich:calendar popup

        </rich:calendar>
        <br>
            <h3>
                <h:outputTex
                <h:outputTex
                <h:commandLi

            </h3>
        </f:view>
    </body>

</html>
```

#{Message}
#{user}
#{applicationScope}
#{cookie}
#{facesContext}
#{header}
#{headerValues}
#{initParam}
#{param}
#{paramValues}
#{requestScope}
Visual/Source
Source
Preview

Figure 3.27. Attributes Content Assist

3.1.2.3.2. Content Assist for JSTL Tags



```
            </tr>
        </table>
        <c:/>

    </html:
</body>
</html:html
```

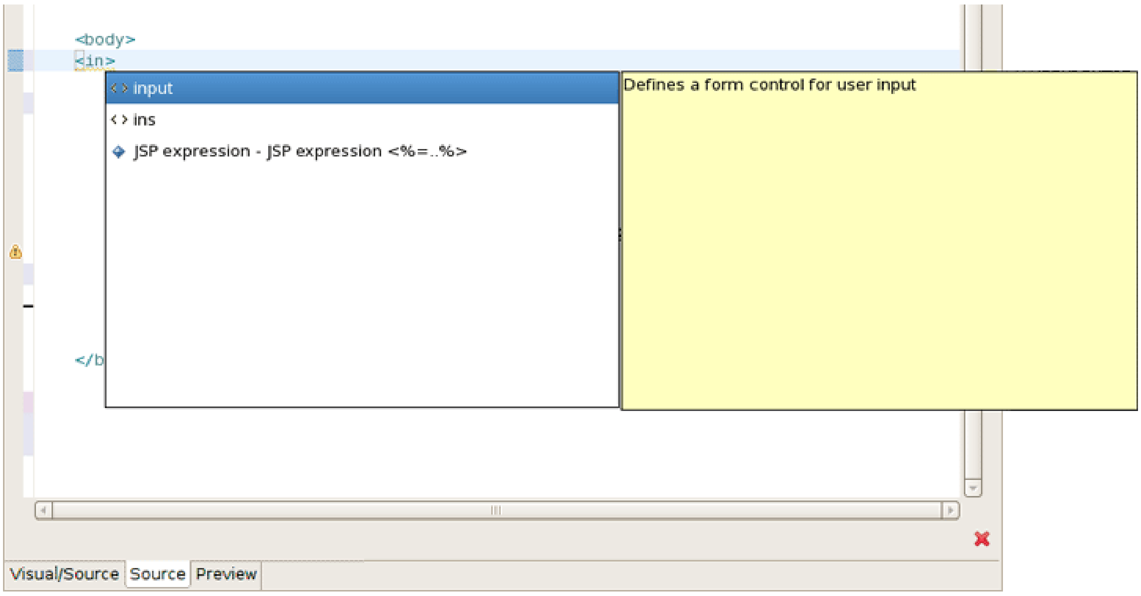
c:catch
c:choose
c:forEach
c:forTokens
c:if
c:import
c:otherwise
c:out
c:param
c:redirect

Catches any Throwable that occurs in its body and optionally exposes it.

Figure 3.28. JSTL Tags Content Assist

3.1.2.3.3. Content Assist for HTML Tags

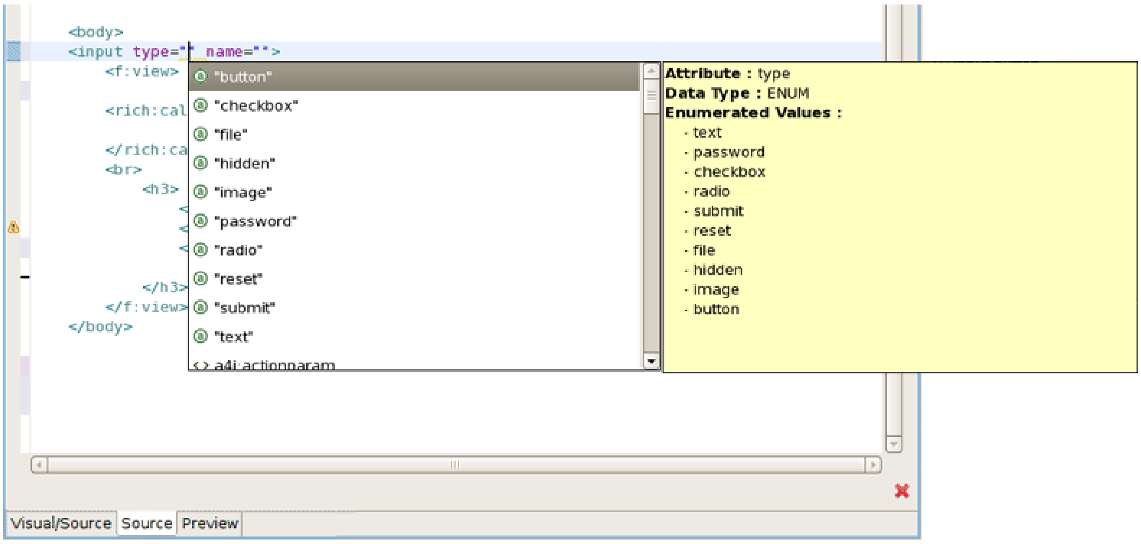
Content assist for HTML tags has the same mechanism as for JSF tags:


<body>
<in>
input
ins
JSP expression - JSP expression <%=..%>
</b
Visual/Source
Source
Preview

Defines a form control for user input

Figure 3.29. HTML Tags Content Assist

You can use as well attributes proposals for HTML tags:

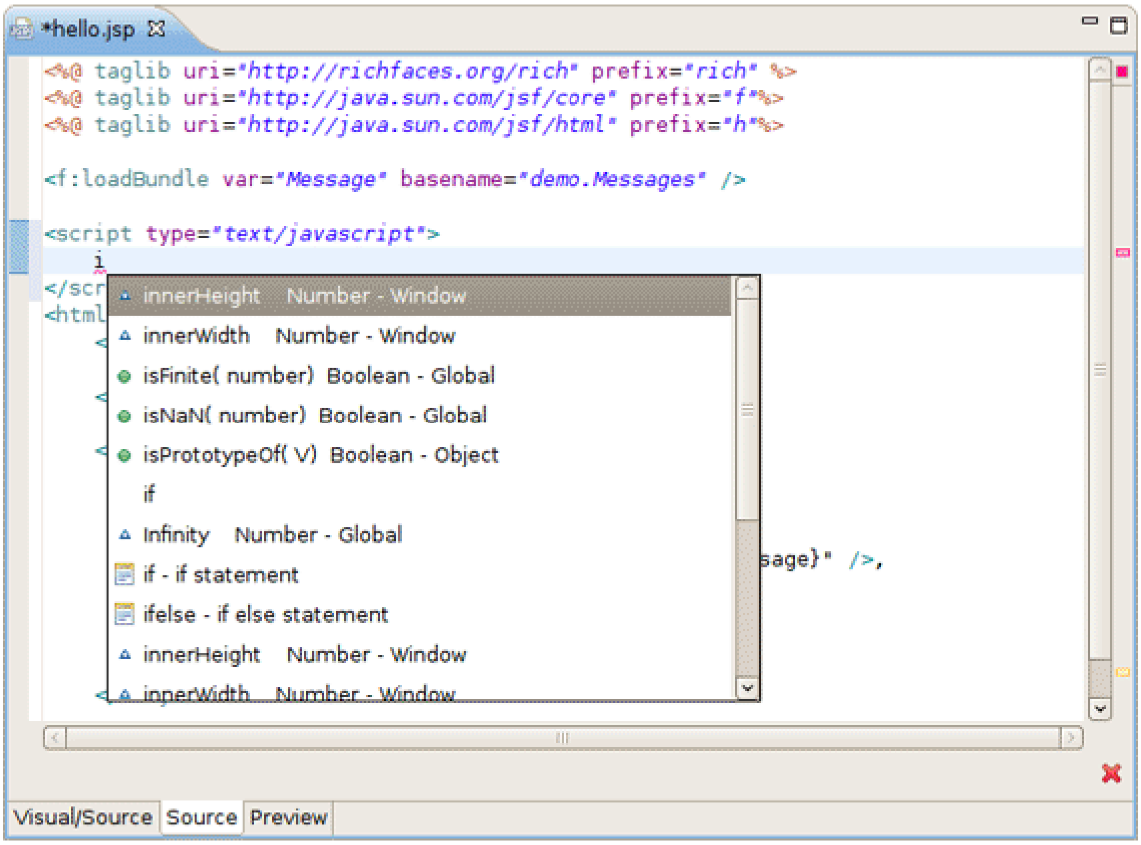

<body>
<input type="" name="">
<f:view>
<rich:cal
</rich:ca

<h3>
</h3>
</f:view>
</body>
"button"
"checkbox"
"file"
"hidden"
"image"
"password"
"radio"
"reset"
"submit"
"text"
a4j:actionparam
Visual/Source
Source
Preview

Attribute : type
Data Type : ENUM
Enumerated Values :
- text
- password
- checkbox
- radio
- submit
- reset
- file
- hidden
- image
- button

Figure 3.30. HTML Tags Content Assist

3.1.2.3.4. Content Assist for JavaScript Tags

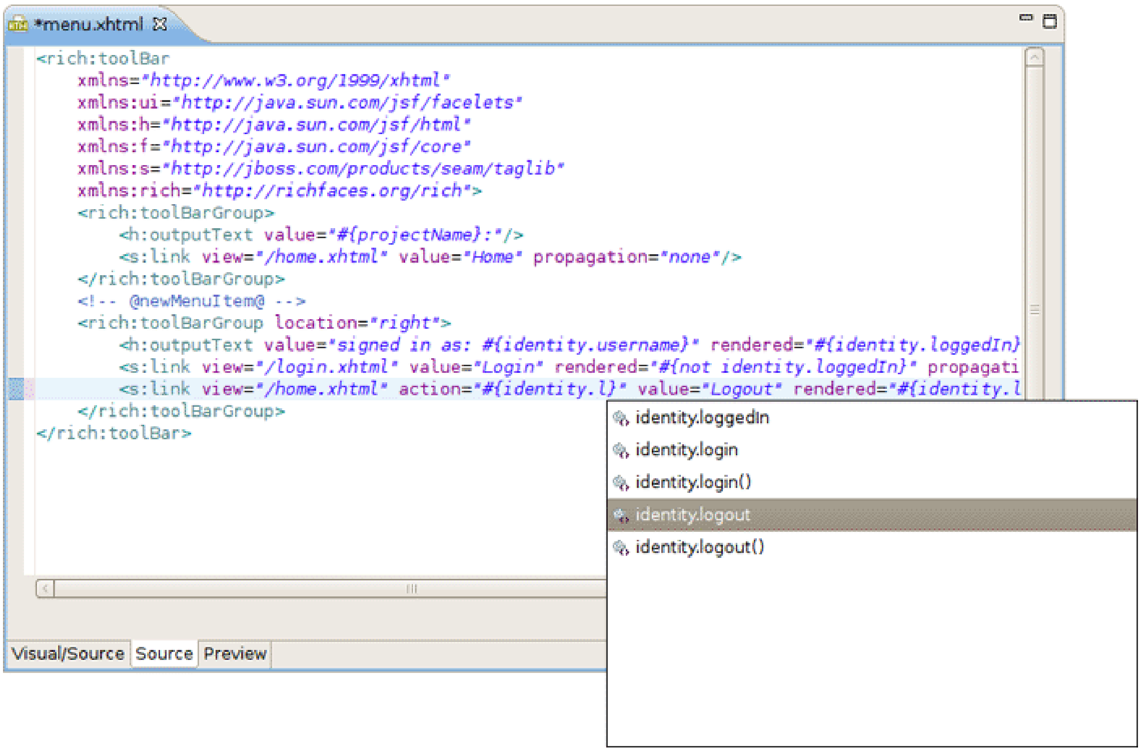

*hello.jsp
innerHeight Number - Window
innerWidth Number - Window
isFinite(number) Boolean - Global
isNaN(number) Boolean - Global
isPrototypeOf(V) Boolean - Object
if
Infinity Number - Global
if - if statement
ifelse - if else statement
innerHeight Number - Window
innerWidth Number - Window
Visual/Source
Source
Preview

```
<%@ taglib uri="http://richfaces.org/rich" prefix="rich" %>
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f"%>
<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h"%>

<f:loadBundle var="Message" basename="demo.Messages" />

<script type="text/javascript">
    i
</scr
<html
```

Figure 3.31. JavaScript Tags Content Assist

3.1.2.4. Content Assist for XHTML Pages

The code completion for the Seam components in a Seam project shows the proposals marked with Seam icon.


*menu.xhtml
identity.loggedIn
identity.login
identity.login()
identity.logout
identity.logout()
Visual/Source
Source
Preview

```
<rich:toolBar
    xmlns="http://www.w3.org/1999/xhtml"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:s="http://jboss.com/products/seam/taglib"
    xmlns:rich="http://richfaces.org/rich">
    <rich:toolBarGroup>
        <h:outputText value="#{projectName}:"/>
        <s:link view="/home.xhtml" value="Home" propagation="none"/>
    </rich:toolBarGroup>
    <!-- @newMenuItem@ -->
    <rich:toolBarGroup location="right">
        <h:outputText value="signed in as: #{identity.username}" rendered="#{identity.loggedIn}
        <s:link view="/login.xhtml" value="Login" rendered="#{not identity.loggedIn}" propagati
        <s:link view="/home.xhtml" action="#{identity.l}" value="Logout" rendered="#{identity.l
    </rich:toolBarGroup>
</rich:toolBar>
```

Figure 3.32. Content Assist for Seam Components in the XHTML Page

If XHTML file uses custom Facelets components, the Content Assist should also be available for them. For details, see *"Content Assist for Custom Facelets Components"* later in this guide.

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:

1. Create a new xml file in *<JBDS_home>studio/eclipse/plugins/org.jboss.tools.common.kb_***/schemas/tld/*. For example call it *JeniaFaces.xml* . The file should be written according to *<JBDS_home>/studio/eclipse/plugins/org.jboss.tools.common.kb/kb.jar/org/jboss/tools/common/kb/kb-schema_1.0.dtd*

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>
```

1. 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"/>
```

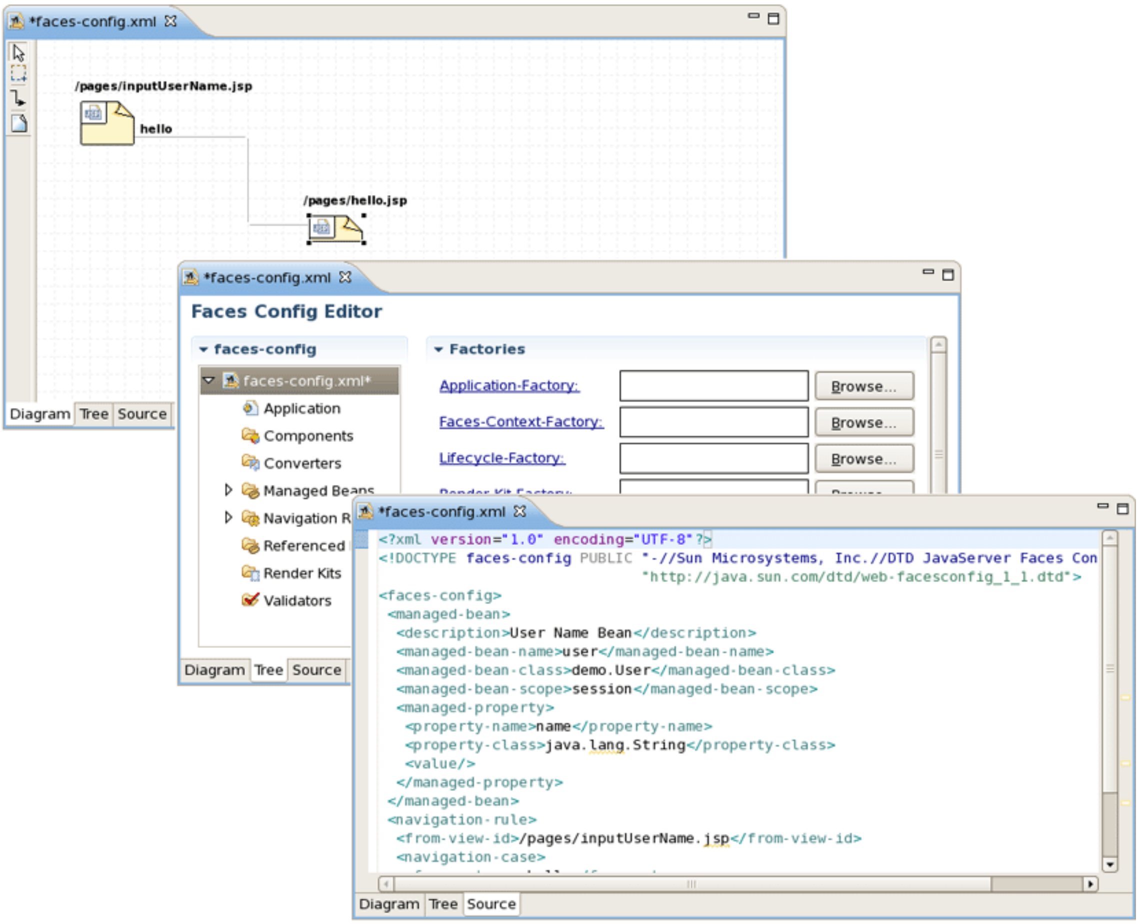
2. Restart Eclipse. You should now have code assist for the component.

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.

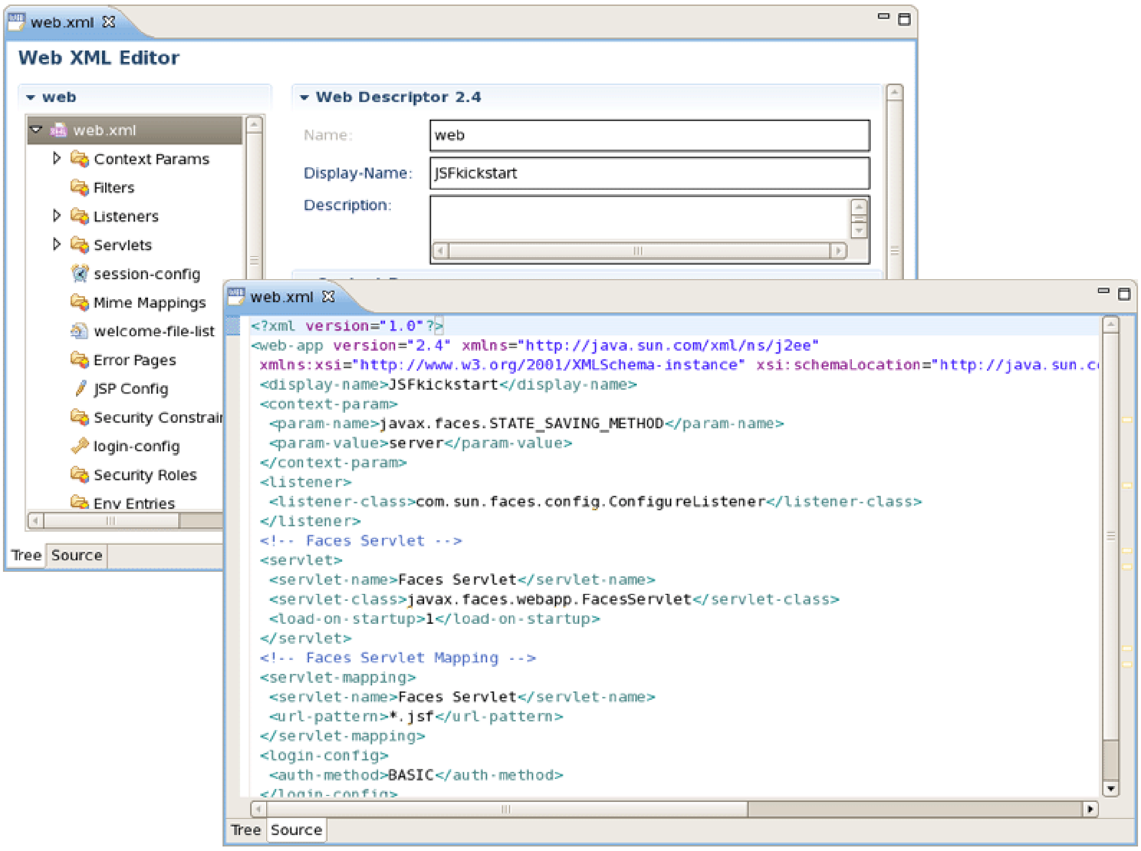

*faces-config.xml
/pages/inputUserName.jsp
hello
/pages/hello.jsp
Diagram
Tree
Source
*faces-config.xml
Faces Config Editor
faces-config
faces-config.xml*
Application
Components
Converters
Managed Beans
Navigation R
Referenced
Render Kits
Validators
Factories
Application-Factory:
Faces-Context-Factory:
Lifecycle-Factory:
Browse...
Browse...
Browse...
Diagram
Tree
Source
*faces-config.xml
Diagram
Tree
Source

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE faces-config PUBLIC "-//Sun Microsystems, Inc.//DTD JavaServer Faces Con
                              "http://java.sun.com/dtd/web-facesconfig_1_1.dtd">
<faces-config>
 <managed-bean>
  <description>User Name Bean</description>
  <managed-bean-name>user</managed-bean-name>
  <managed-bean-class>demo.User</managed-bean-class>
  <managed-bean-scope>session</managed-bean-scope>
  <managed-property>
   <property-name>name</property-name>
   <property-class>java.lang.String</property-class>
   <value/>
  </managed-property>
 </managed-bean>
 <navigation-rule>
  <from-view-id>/pages/inputUserName.jsp</from-view-id>
  <navigation-case>
```

Figure 3.33. 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
Web XML Editor
web
web.xml
Context Params
Filters
Listeners
Servlets
session-config
Mime Mappings
welcome-file-list
Error Pages
JSP Config
Security Constrai
login-config
Security Roles
Env Entries
Web Descriptor 2.4
Tree
Source
web.xml
Tree
Source

Name: web

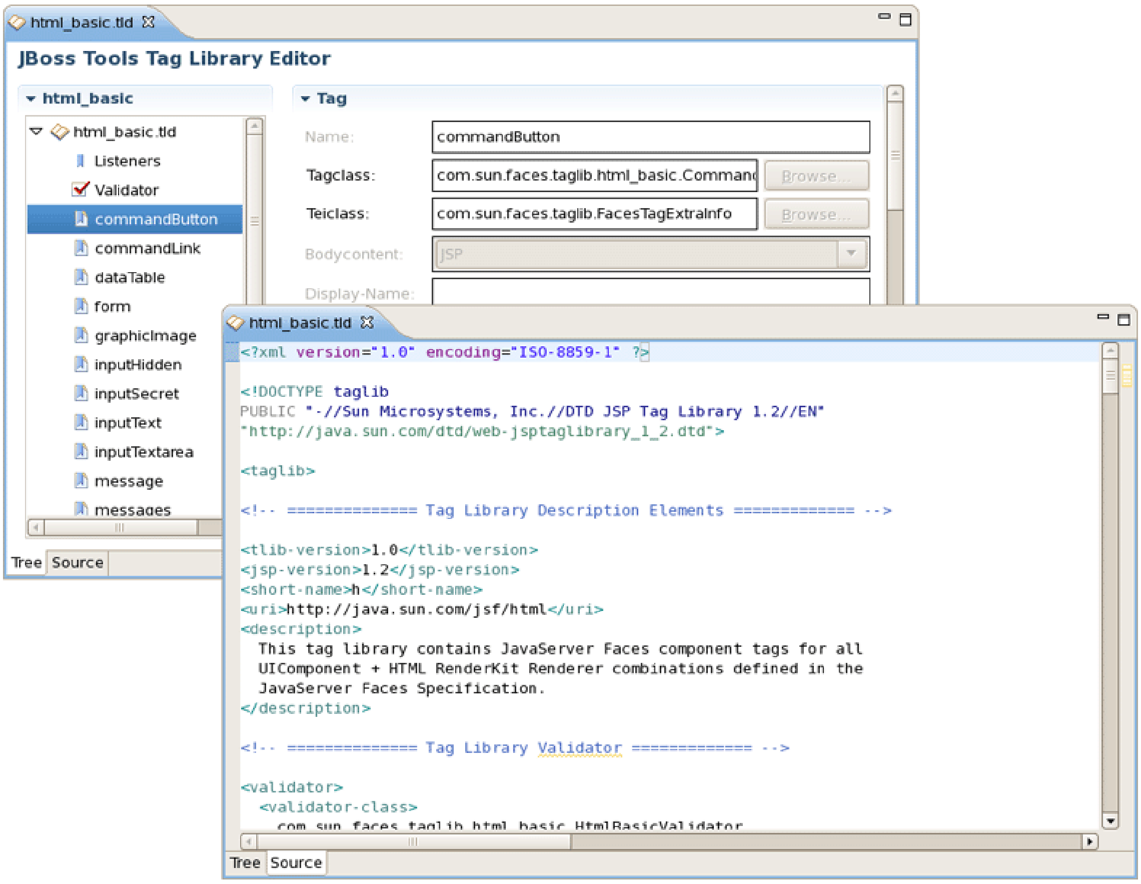
Display-Name: JSFkickstart

Description:

```
<?xml version="1.0"?>
<web-app version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://java.sun.c
 <display-name>JSFkickstart</display-name>
 <context-param>
  <param-name>javax.faces.STATE_SAVING_METHOD</param-name>
  <param-value>server</param-value>
 </context-param>
 <listener>
  <listener-class>com.sun.faces.config.ConfigureListener</listener-class>
 </listener>
 <!-- Faces Servlet -->
 <servlet>
  <servlet-name>Faces Servlet</servlet-name>
  <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
  <load-on-startup>1</load-on-startup>
 </servlet>
 <!-- Faces Servlet Mapping -->
 <servlet-mapping>
  <servlet-name>Faces Servlet</servlet-name>
  <url-pattern>*.jsf</url-pattern>
 </servlet-mapping>
 <login-config>
  <auth-method>BASIC</auth-method>
 </login-config>
```

Figure 3.34. 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.


html_basic.tld
JBoss Tools Tag Library Editor
html_basic
html_basic.tld
Listeners
Validator
commandButton
commandLink
dataTable
form
graphicImage
inputHidden
inputSecret
inputText
inputTextarea
message
messages
Tree
Source
Tag
Browse...
Browse...
html_basic.tld
Tree
Source

Name: commandButton

Tagclass: com.sun.faces.taglib.html_basic.Command

Teiclass: com.sun.faces.taglib.FacesTagExtraInfo

Bodycontent: JSP

Display-Name:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>

<!DOCTYPE taglib
PUBLIC "-//Sun Microsystems, Inc.//DTD JSP Tag Library 1.2//EN"
"http://java.sun.com/dtd/web-jsptaglibrary_1_2.dtd">

<taglib>

<!-- ============== Tag Library Description Elements ============= -->

<tlib-version>1.0</tlib-version>
<jsp-version>1.2</jsp-version>
<short-name>h</short-name>
<uri>http://java.sun.com/jsf/html</uri>
<description>
  This tag library contains JavaServer Faces component tags for all
  UIComponent + HTML RenderKit Renderer combinations defined in the
  JavaServer Faces Specification.
</description>

<!-- ============== Tag Library Validator ============= -->

<validator>
  <validator-class>
    com.sun.faces.taglib.html_basic.HtmlBasicValidator
```

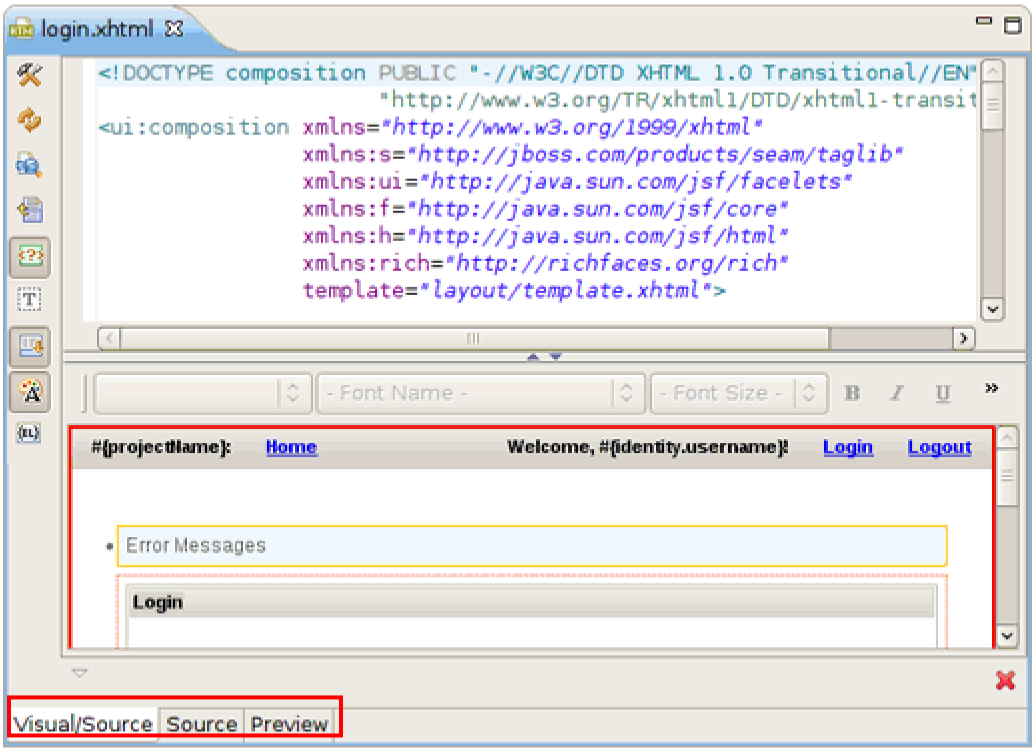
Figure 3.35. Two Views are Synchronized

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. Double-click on the necessary file in the Package Explorer view to open it in the Visual Editor or just drag-and-drop it into perspective (the drag-and-drop feature can be also applied to JSP, XHTML or HTML files created locally).

As a new JSF 2.0 specification has been released, support of new features is now implemented in the Visual Page Editor. The JSF 2.0 tags like `<h:body>`, `<h:head>`, `<h:outputscript>`, `<h:outputstyle>` are supported in the editor as well as the composite components and the resource handling from the expression language. (See the *following link* on how to use composite components and *following blog post* on how to handle the resources form the EL).

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* .



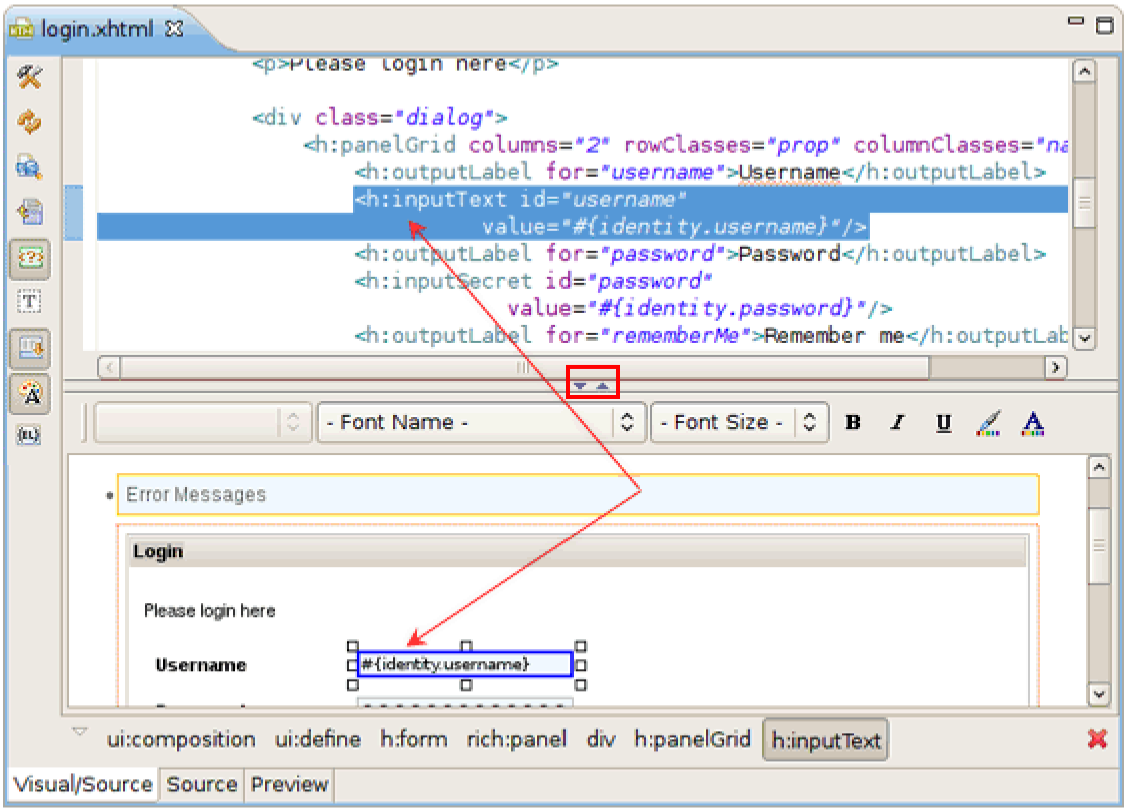
```
<!DOCTYPE composition PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
                      "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transit
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
                xmlns:s="http://jboss.com/products/seam/taglib"
                xmlns:ui="http://java.sun.com/jsf/facelets"
                xmlns:f="http://java.sun.com/jsf/core"
                xmlns:h="http://java.sun.com/jsf/html"
                xmlns:rich="http://richfaces.org/rich"
                template="layout/template.xhtml">
```

login.xhtml
- Font Name -
- Font Size -
B
I
U
#{projectName}:
Home
Welcome, #{identity.username}!
Login
Logout
Error Messages
Login
Visual/Source
Source
Preview

Figure 3.36. 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:



login.xhtml
- Font Name -
- Font Size -
Error Messages
Login
Please login here
Username
#{identity.username}
ui:composition
ui:define
h:form
rich:panel
div
h:panelGrid
h:inputText
Visual/Source
Source
Preview

```
            <p>Please login here</p>

            <div class="dialog">
                <h:panelGrid columns="2" rowClasses="prop" columnClasses="na
                    <h:outputLabel for="username">Username</h:outputLabel>
                    <h:inputText id="username"
                              value="#{identity.username}"/>
                    <h:outputLabel for="password">Password</h:outputLabel>
                    <h:inputSecret id="password"
                                value="#{identity.password}"/>
                    <h:outputLabel for="rememberMe">Remember me</h:outputLab
```

Figure 3.37. Visual/Source View

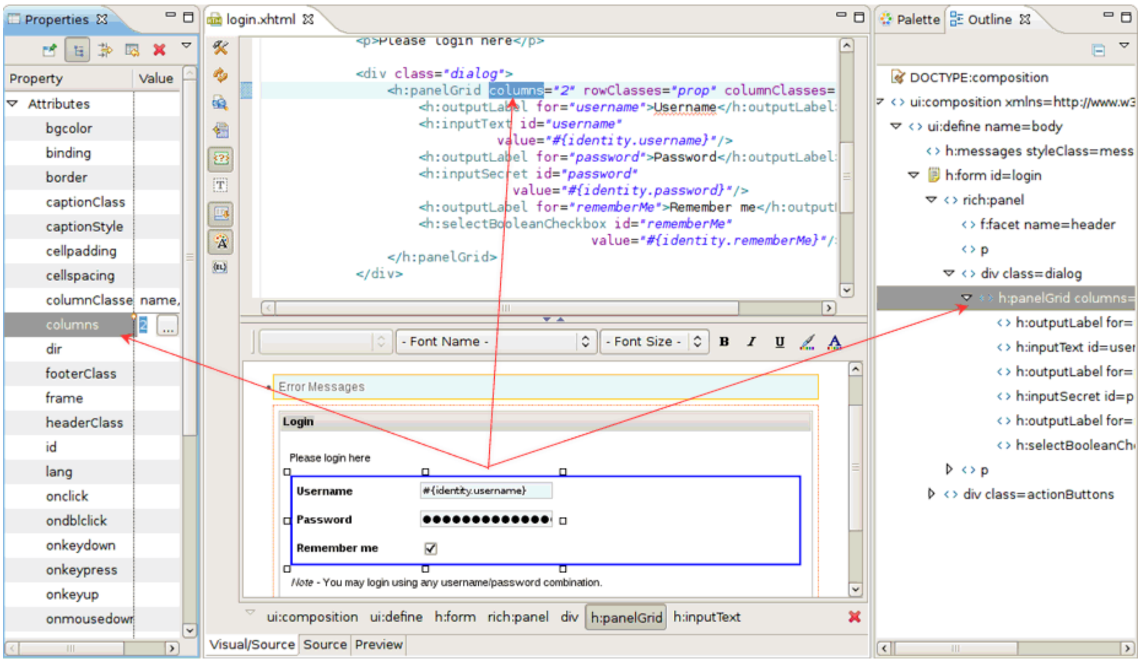
The view is designed in the form of a split pane with toggle buttons for quickly moving between Source, Visual or Source/Visual modes as shown on the figure above.

One more way to toggle between the various states of the split pane is using the shortcuts *Shift + F6* for maximizing/restoring the Source part and *Shift + Alt + F6* for maximizing/restoring the Visual part.

Tip:

When editing large documents hiding the Visual part will speed up the editing.

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



```
<p>Please login here</p>

<div class="dialog">
    <h:panelGrid columns="2" rowClasses="prop" columnClasses=
        <h:outputLabel for="username">Username</h:outputLabel:
        <h:inputText id="username"
                    value="#{identity.username}"/>
        <h:outputLabel for="password">Password</h:outputLabel:
        <h:inputSecret id="password"
                    value="#{identity.password}"/>
        <h:outputLabel for="rememberMe">Remember me</h:output
        <h:selectBooleanCheckbox id="rememberMe"
                    value="#{identity.rememberMe}"/:
    </h:panelGrid>
</div>
```

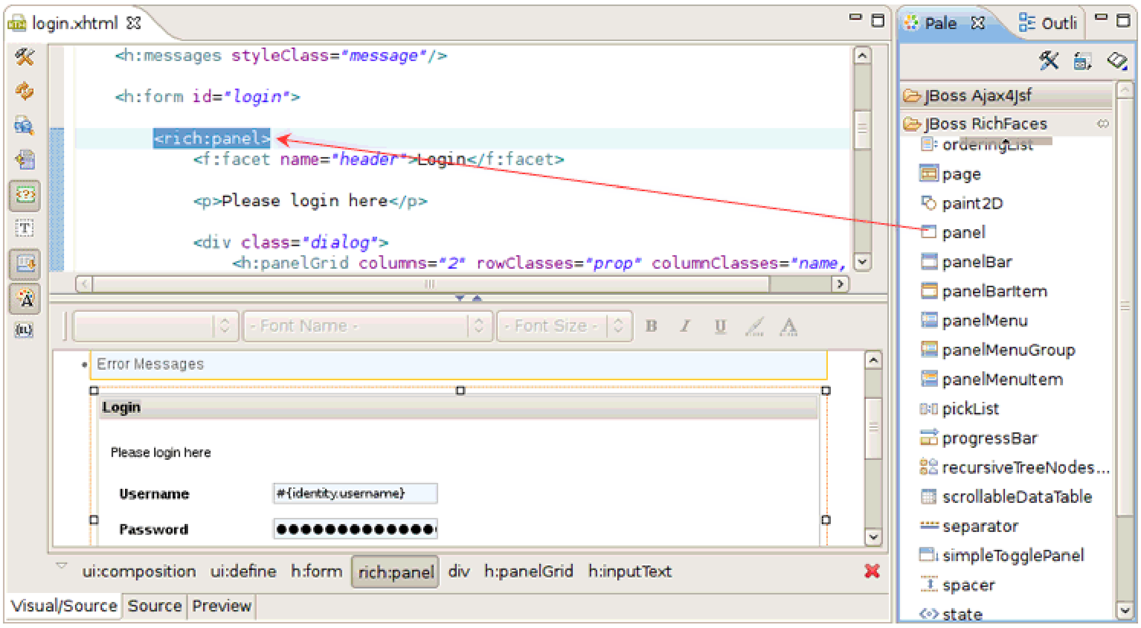
Properties
Property
Value
Attributes
bgcolor
binding
border
captionClass
captionStyle
cellpadding
cellspacing
columnClasse name,
columns 2
dir
footerClass
frame
headerClass
id
lang
onclick
ondblclick
onkeydown
onkeypress
onkeyup
onmousedown
login.xhtml
- Font Name -
- Font Size -
Error Messages
Login
Please login here
Username
#{identity.username}
Password
Remember me
Note - You may login using any username/password combination.
ui:composition ui:define h:form rich:panel div h:panelGrid h:inputText
Visual/Source Source Preview
Palette
Outline
DOCTYPE:composition
ui:composition xmlns=http://www.w3
ui:define name=body
h:messages styleClass=mess
h:form id=login
rich:panel
f:facet name=header
p
div class=dialog
h:panelGrid columns=
h:outputLabel for=
h:inputText id=user
h:outputLabel for=
h:inputSecret id=p
h:outputLabel for=
h:selectBooleanCh
p
div class=actionButtons

Figure 3.38. Integration with Properties and Outline Views

The Outline view displays a specific outline of a structured file that is currently open in the editor area, and lists its structural elements. Right-click on the elements will open additional options that allow adding other specific elements in necessary positions.

The Properties view shows property names and their values for a selected item. The values are editable, just select any and click on the button that appeared to choose a new value. Key combination Ctrl+Z will return the previous value, Ctrl+Y will return the new value again. The Properties view has additional options and can be set up to display categories and advanced properties.

It's also possible to use the *JBoss Tools Palette* 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
Pale
Outli
JBoss Ajax4Jsf
JBoss RichFaces
orderingList
page
paint2D
panel
panelBar
panelBarItem
panelMenu
panelMenuGroup
panelMenuItem
pickList
progressBar
recursiveTreeNodes...
scrollableDataTable
separator
simpleTogglePanel
spacer
state
- Font Name -
- Font Size -
ui:composition
ui:define
h:form
rich:panel
div
h:panelGrid
h:inputText
Visual/Source
Source
Preview

```
    <h:messages styleClass="message"/>

    <h:form id="login">

        <rich:panel>
            <f:facet name="header">Login</f:facet>

            <p>Please login here</p>

            <div class="dialog">
                <h:panelGrid columns="2" rowClasses="prop" columnClasses="name,
```

- Error Messages

Login

Please login here

Username #{identity.username}

Password ●●●●●●●●●●●●●

Figure 3.39. Inserting Tag From the Palette

You can insert a tag/component from the palette into either the Source or the Visual part by calling a context menu and selecting Insert around, Insert before, Insert after or Replace With, pointing to From Palette, picking the type of the tag and finally choosing the tag you want to insert.

The image below illustrates how you can insert a tag into the Source part.

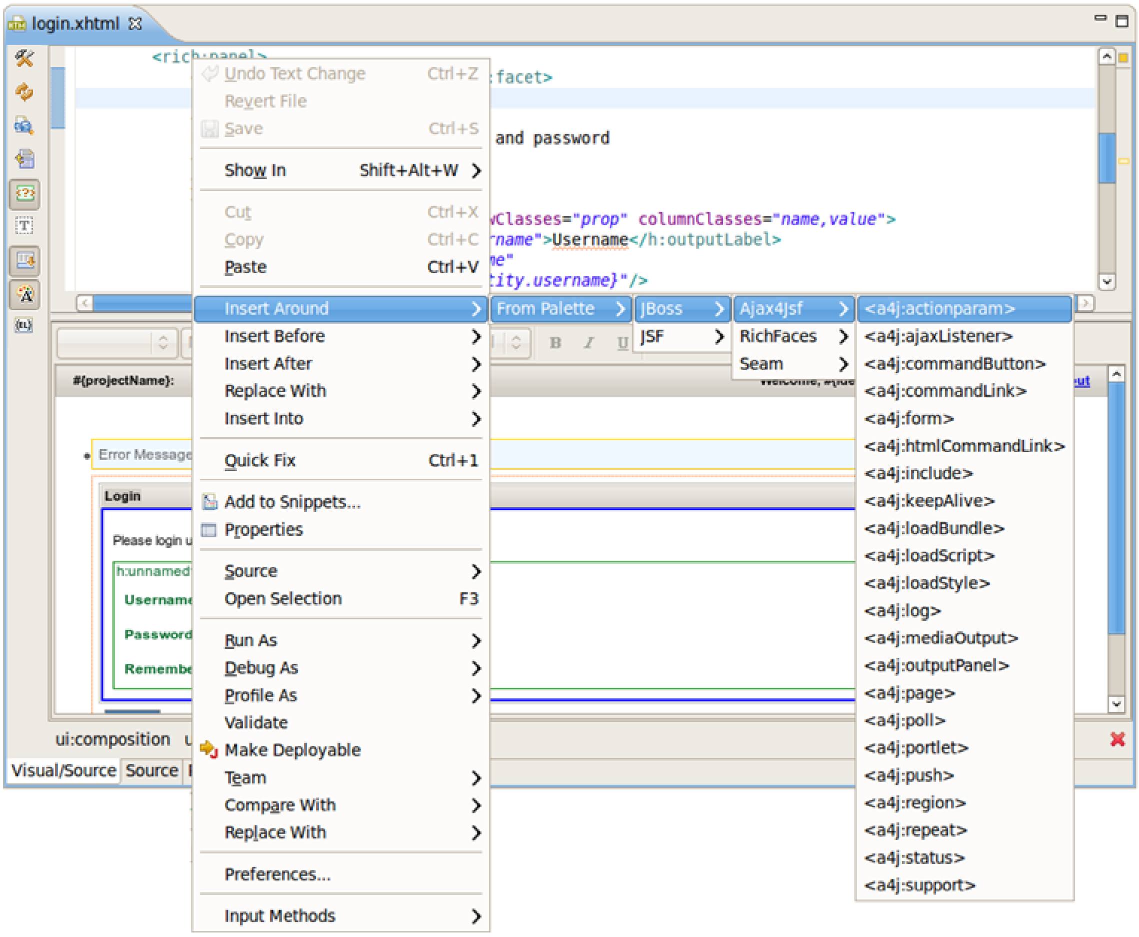
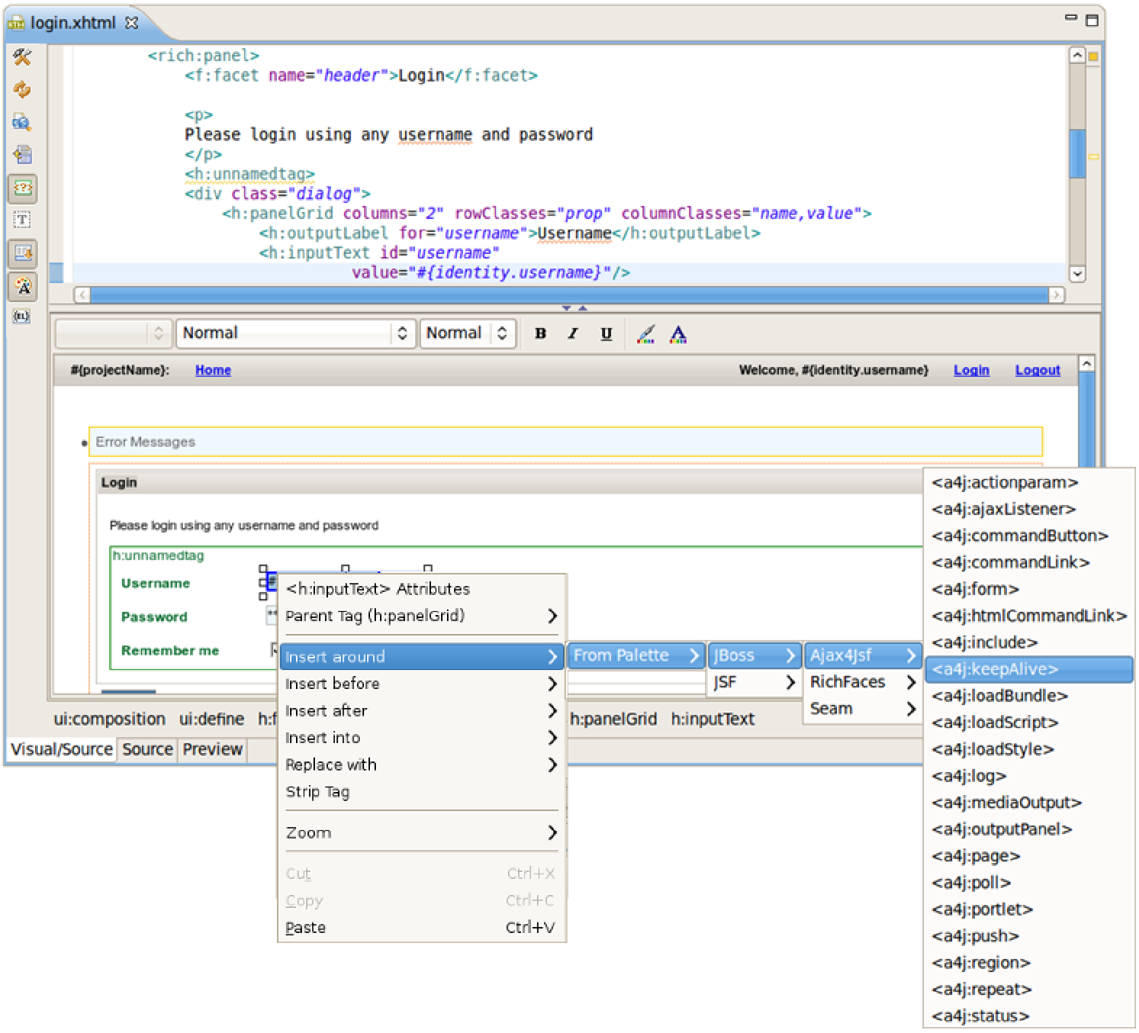

login.xhtml
Undo Text Change Ctrl+Z
Revert File
Save Ctrl+S
Show In Shift+Alt+W
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Insert Around
Insert Before
Insert After
Replace With
Insert Into
Quick Fix Ctrl+1
Add to Snippets...
Properties
Source
Open Selection F3
Run As
Debug As
Profile As
Validate
Make Deployable
Team
Compare With
Replace With
Preferences...
Input Methods
From Palette
JBoss
JSF
Ajax4Jsf
RichFaces
Seam
<a4j:actionparam>
<a4j:ajaxListener>
<a4j:commandButton>
<a4j:commandLink>
<a4j:form>
<a4j:htmlCommandLink>
<a4j:include>
<a4j:keepAlive>
<a4j:loadBundle>
<a4j:loadScript>
<a4j:loadStyle>
<a4j:log>
<a4j:mediaOutput>
<a4j:outputPanel>
<a4j:page>
<a4j:poll>
<a4j:portlet>
<a4j:push>
<a4j:region>
<a4j:repeat>
<a4j:status>
<a4j:support>
#{projectName}:
Error Message
Login
Please login u
h:unnamed
Username
Password
Remembe
ui:composition
Visual/Source
Source

Figure 3.40. Inserting a tag into the Source part

And this is how a tag is inserted using a context menu in the Visual part.

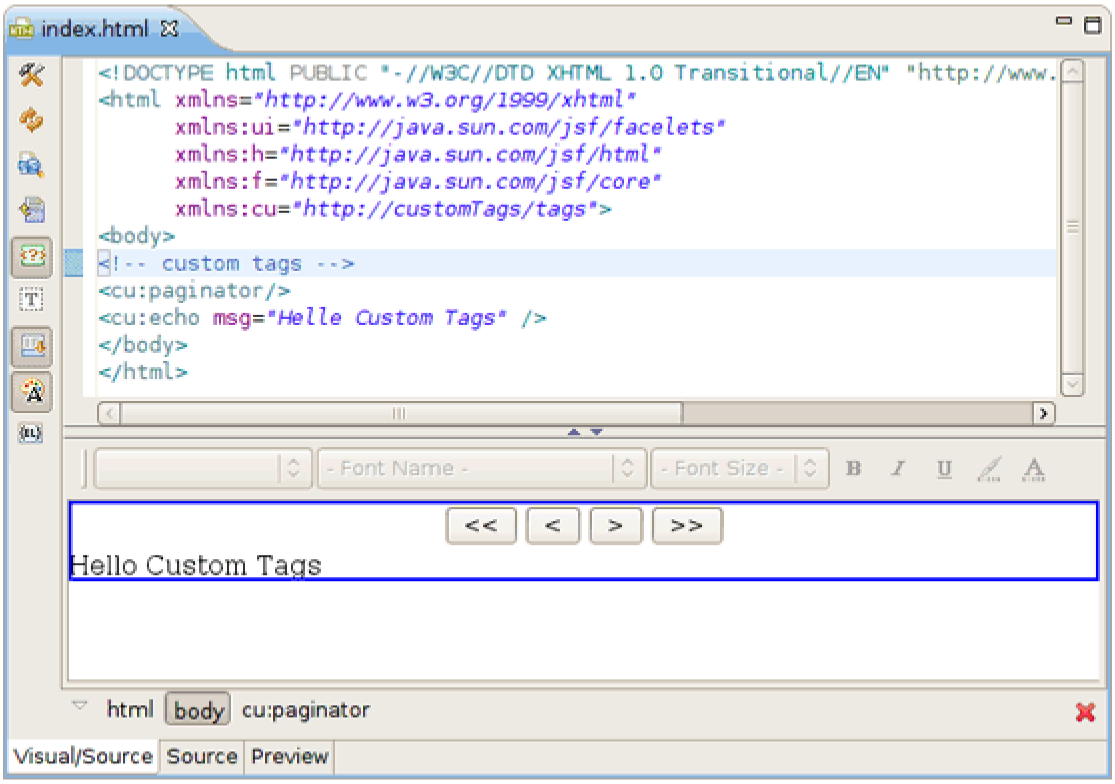


```
        <rich:panel>
            <f:facet name="header">Login</f:facet>

            <p>
            Please login using any username and password
            </p>
            <h:unnamedtag>
            <div class="dialog">
                <h:panelGrid columns="2" rowClasses="prop" columnClasses="name,value">
                    <h:outputLabel for="username">Username</h:outputLabel>
                    <h:inputText id="username"
                              value="#{identity.username}"/>
```

login.xhtml
Normal
Normal
B
I
U
#{projectName}:
Home
Welcome, #{identity.username}
Login
Logout
Error Messages
Login
Please login using any username and password
h:unnamedtag
Username
Password
Remember me
<h:inputText> Attributes
Parent Tag (h:panelGrid)
Insert around
Insert before
Insert after
Insert into
Replace with
Strip Tag
Zoom
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
From Palette
JBoss
JSF
Ajax4Jsf
RichFaces
Seam
<a4j:actionparam>
<a4j:ajaxListener>
<a4j:commandButton>
<a4j:commandLink>
<a4j:form>
<a4j:htmlCommandLink>
<a4j:include>
<a4j:keepAlive>
<a4j:loadBundle>
<a4j:loadScript>
<a4j:loadStyle>
<a4j:log>
<a4j:mediaOutput>
<a4j:outputPanel>
<a4j:page>
<a4j:poll>
<a4j:portlet>
<a4j:push>
<a4j:region>
<a4j:repeat>
<a4j:status>
ui:composition
ui:define
h:f
h:panelGrid
h:inputText
Visual/Source
Source
Preview

Figure 3.41. Inserting a tag into the Visual part

Visual Page Editor also displays custom tags correctly if they are configured properly. The picture below shows an example how custom tags *"pagination"* and *"echo"* will be displayed in VPE.



index.html
- Font Name -
- Font Size -
<<
<
>
>>
html
body
cu:paginator
Visual/Source
Source
Preview

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:cu="http://customTags/tags">
<body>
<!-- custom tags -->
<cu:paginator/>
<cu:echo msg="Helle Custom Tags" />
</body>
</html>
```

Hello Custom Tags

Figure 3.42. Custom Tags in the VPE

The listings of the custom tags implementations will help to clear how VPE works.

- echo.xhtml:

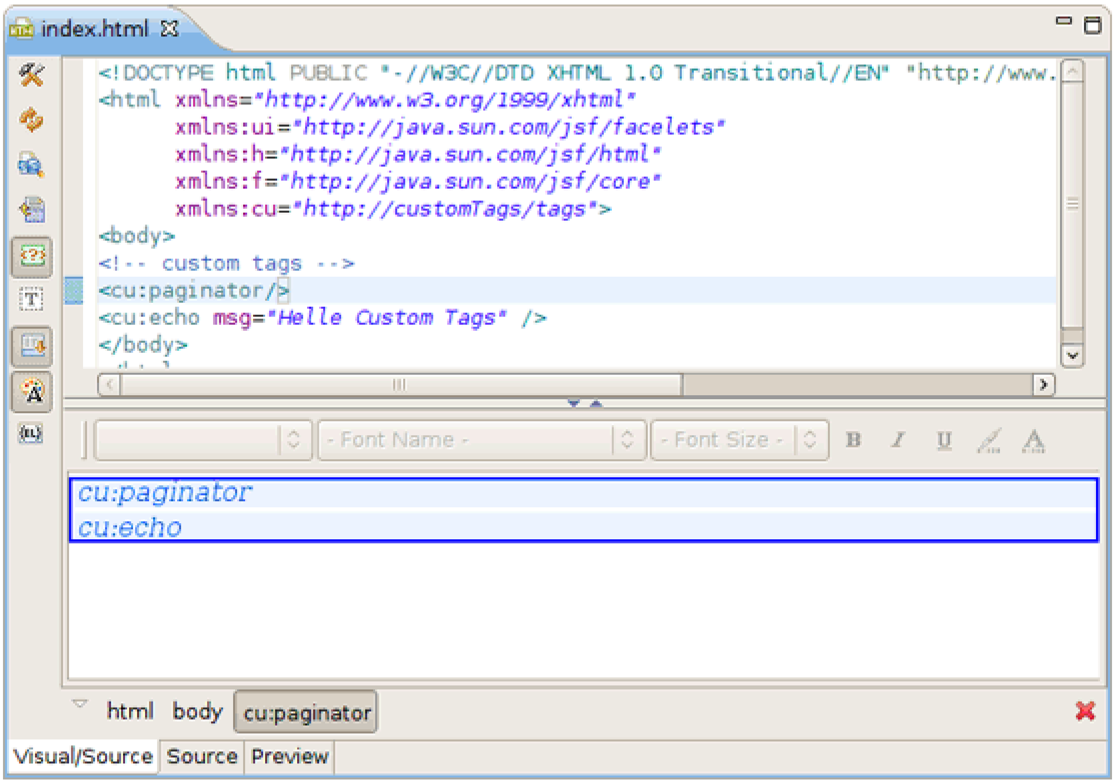
```
<ui:composition xmlns:ui="http://java.sun.com/jsf/facelets">
    <span class="message">#{msg}</span>
</ui:composition>
```

- paginator.xhtml:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:f="http://java.sun.com/jsf/core">
```

```
<ui:component>
<!-- h:inputHidden id="currentPage" replace, because if on page two fields,
two elements with equal id has been used, but should be used only one -->
    <h:panelGrid style="margin-right:auto;margin-left:auto;" columns="4">
        <h:commandButton value="&lt;&lt;" type="submit"
        onclick="document.getElementById('currentPage').value=0" >
        </h:commandButton>
        <h:commandButton value="&lt;" type="submit"
                        onclick="document.getElementById('currentPage').value=#{user.currentPage-
user.rowsPerPage}">
        </h:commandButton>
        <h:commandButton value="&gt;" type="submit"
                         onclick="document.getElementById('currentPage').value=#{user.currentPage
+user.rowsPerPage}">
        </h:commandButton>
        <h:commandButton value="&gt;&gt;" type="submit"
                 onclick="document.getElementById('currentPage').value=#{user.numberOfItems  -
 user.rowsPerPage}">
        </h:commandButton>
    </h:panelGrid>
    <h:inputHidden id="currentPage" value=""/>
</ui:component>
</html>
```

If your custom tags aren't configured correctly your Visual mode will look like this:



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:cu="http://customTags/tags">
<body>
<!-- custom tags -->
<cu:paginator/>
<cu:echo msg="Helle Custom Tags" />
</body>
```

cu:paginator
cu:echo

Figure 3.43. Wrong configured Custom Tags in the VPE

3.2.1.1. Commenting out Code

VPE supports possibility to add comments in files you are working with (JSP, XHTML, etc.):

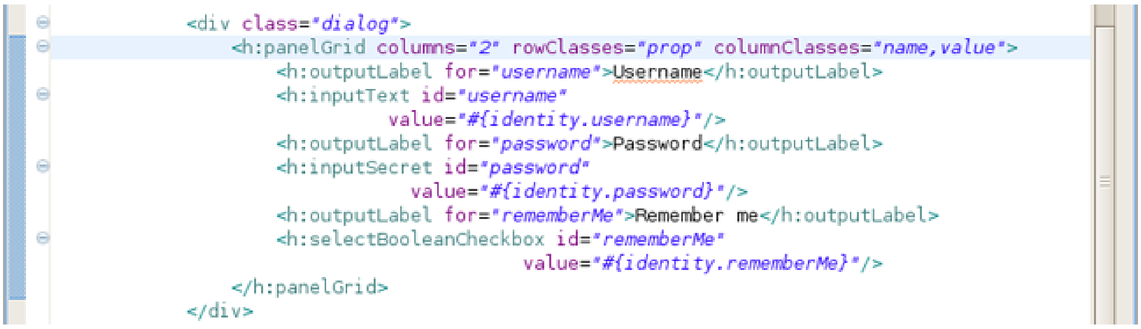
- HTML comments (`<!-- -->`) which are output to the client
- JSP comments (`<%-- --%>`) which are not output to the client as part of the JSP page output

3.2.1.2. Using Code Folding

Visual Page Editor lets you collapse (hide) and expand (show) sections of your code to make it easier to navigate and read.

Code folding can be enabled by right-clicking on the left margin on the Source part of Visual Page Editor, selecting Folding and checking the Enable Folding checkbox or using the *Ctrl +Numpad_Divide* shortcut.

When the code folding is enabled a minus sing (⊖) will appear on the left margin of the editor next to each opening block tag.

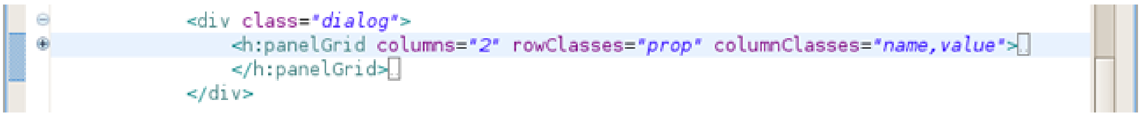


```
<div class="dialog">
    <h:panelGrid columns="2" rowClasses="prop" columnClasses="name,value">
        <h:outputLabel for="username">Username</h:outputLabel>
        <h:inputText id="username"
                  value="#{identity.username}"/>
        <h:outputLabel for="password">Password</h:outputLabel>
        <h:inputSecret id="password"
                    value="#{identity.password}"/>
        <h:outputLabel for="rememberMe">Remember me</h:outputLabel>
        <h:selectBooleanCheckbox id="rememberMe"
                              value="#{identity.rememberMe}"/>
    </h:panelGrid>
</div>
```

Figure 3.44. Enabled Code Folding

Click the minus sign to collapse a block tag.

When the minus sing is clicked on the appropriate tag collapses and a plus sing (⊕) is displayed on the left margin as well as a gray rectangle two dots (▯) appears after opening and closing tags.



```
<div class="dialog">
    <h:panelGrid columns="2" rowClasses="prop" columnClasses="name,value">▯
    </h:panelGrid>▯
</div>
```

Figure 3.45. Collapsed Code

3.2.1.3. JSP Syntax Validation

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

3.2.1.4. Support for custom TagLibs and Taglib versions

VPE templates support custom tag libs, e.g. Seam Mail facelet taglib, RichFaces taglibs or any other created by you.

VPE templates also provide a support for various versions of tag libraries. It means that the VPE takes control over those components which have different parameters or preview according to the framework version (like seam 1.2 and seam 2.0, or JSF 1.1 and JSF 1.2).

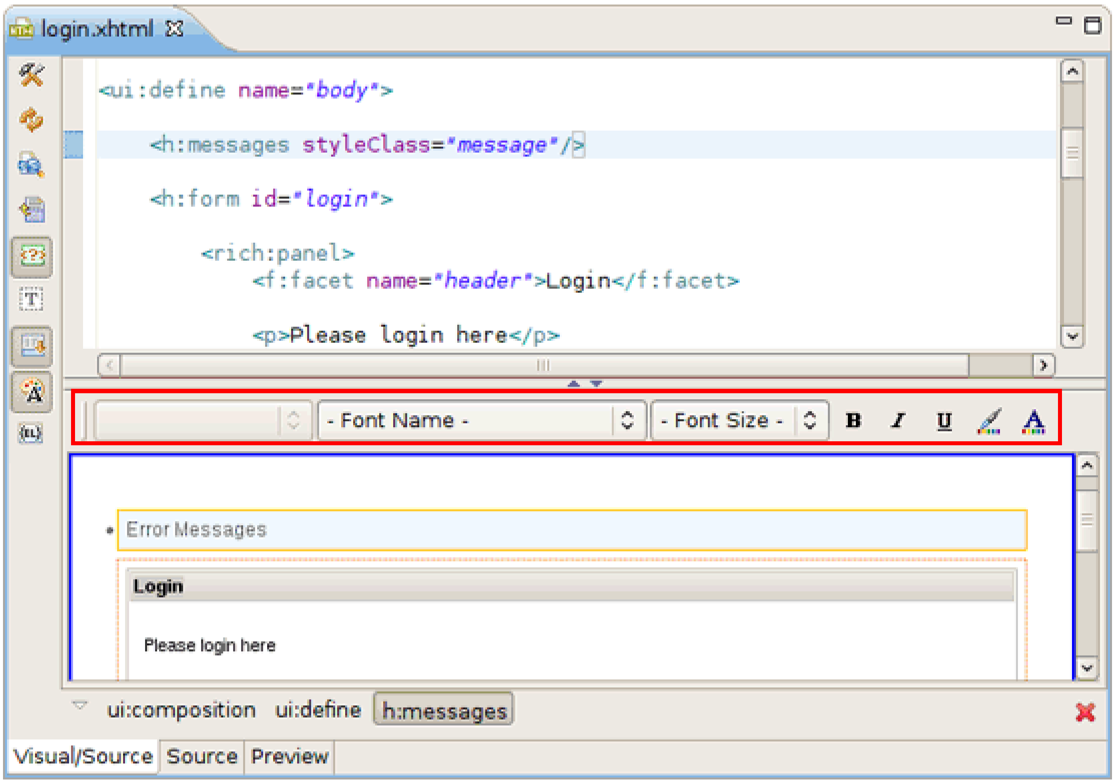
For example, `<s:decorate>` element in seam has different parameters in versions 1.2 and 2.0 as well as `<h:outputLink>` JSF element has different preview in versions 1.1 and 1.2.

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. More helpful information on wok with CSS files can be found in *CSS Perspective chapter*

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 with the help of the special button ( ) on the VPE toolbar.



```
<ui:define name="body">

    <h:messages styleClass="message"/>

    <h:form id="login">

        <rich:panel>
            <f:facet name="header">Login</f:facet>

            <p>Please login here</p>
```

login.xhtml
- Font Name -
- Font Size -
B
I
U
Error Messages
Login
Please login here
ui:composition
ui:define
h:messages
Visual/Source
Source
Preview

Figure 3.46. Text Formatting

For editing inline styles for DOM elements VPE provides the CSS Dialog. It can be called from *style* line in the Properties view for a currently selected element.

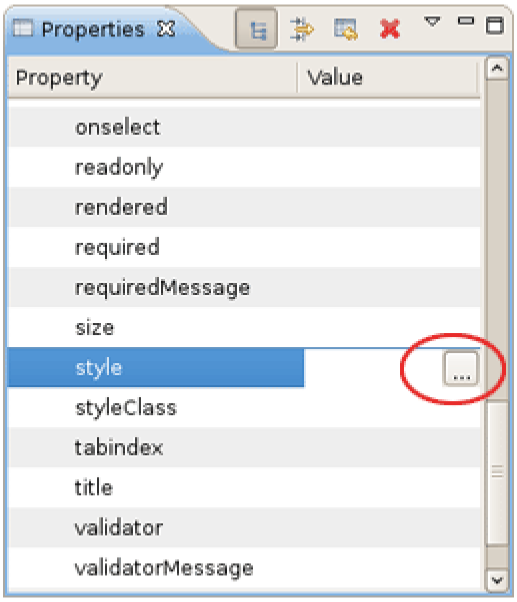
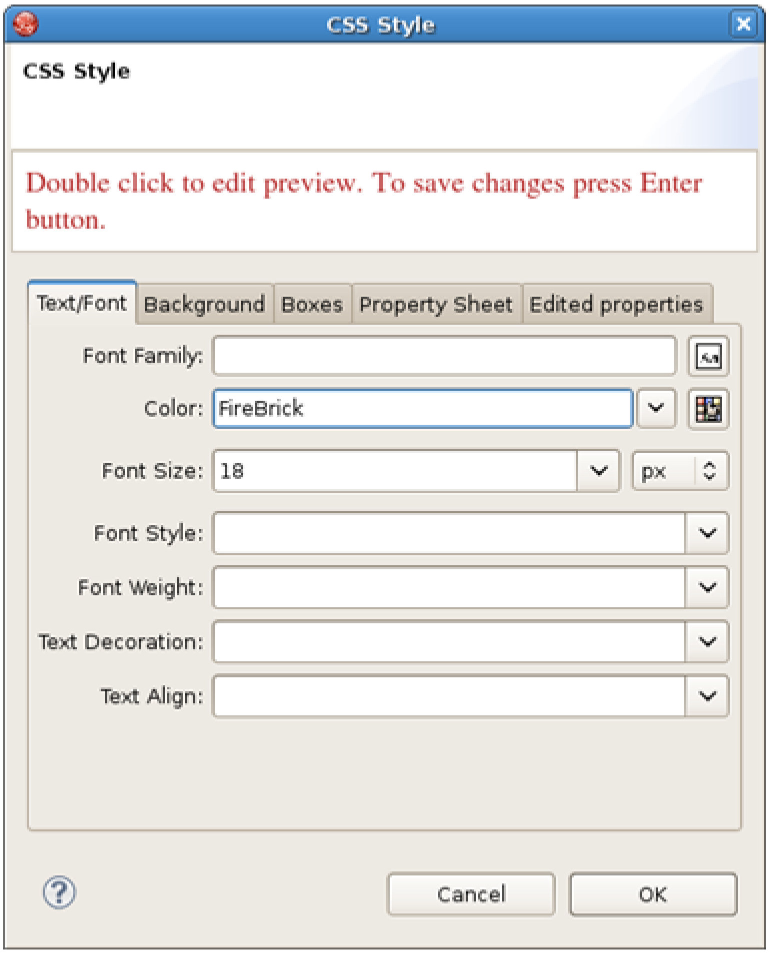

Properties
Property
Value
onselect
readonly
rendered
required
requiredMessage
size
style
...
styleClass
tabindex
title
validator
validatorMessage

Figure 3.47. Call the CSS Dialog

CSS Style Dialog has several 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 Style Dialog allows you to see the changes before you apply them.



CSS Style

Double click to edit preview. To save changes press Enter button.

CSS Style
Text/Font
Background
Boxes
Property Sheet
Edited properties

Font Family:

Color: FireBrick

Font Size: 18 px

Font Style:

Font Weight:

Text Decoration:

Text Align:

Cancel
OK

Figure 3.48. CSS Style 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 Edit 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.

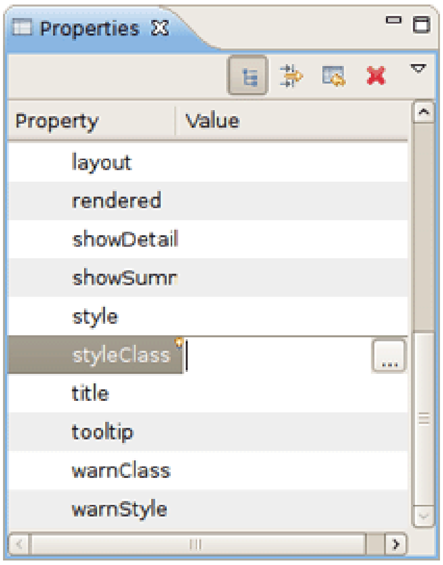

Properties
Property
Value
layout
rendered
showDetail
showSumr
style
styleClass
...
title
tooltip
warnClass
warnStyle

Figure 3.49. Calling the Edit Style Class Dialog

It'll pick up the Edit Style Class Dialog which looks like on the figure below:

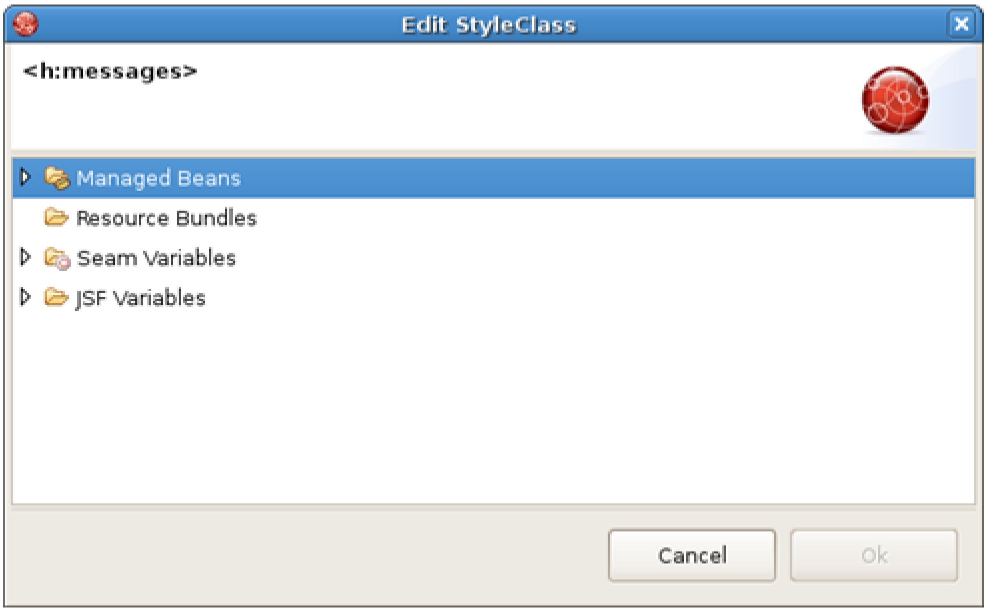

Edit StyleClass
<h:messages>
Managed Beans
Resource Bundles
Seam Variables
JSF Variables
Cancel
Ok

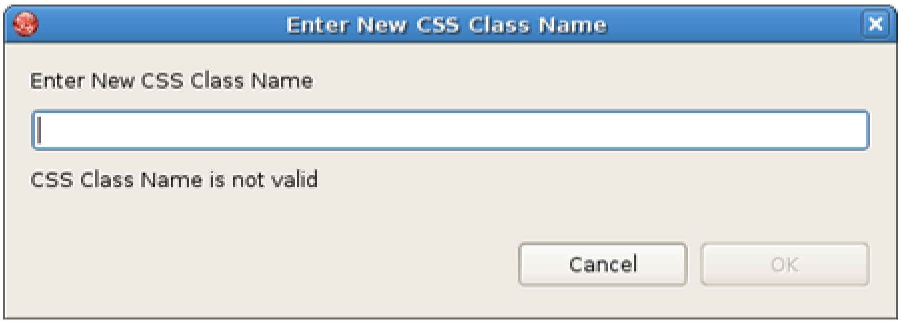
Figure 3.50. Edit Style Class Dialog

Choose a style class from the variants provided and click on the Ok button to apply the changes.

To open a CSS dialog based on the active CSS file click on

in the top panel or use hot-keys (Shift+Ctrl+C).

To create a new CSS class for the file click on the Add CSS Class button, write its name in the field appeared and click on the Ok button:


Enter New CSS Class Name
Enter New CSS Class Name
Cancel
OK

CSS Class Name is not valid

Figure 3.51. Add CSS Class

Then you can configure style settings switching between the tabs: *Text/Font*, *Background*, *Boxes*, *Property Sheet*. The list of already existing classes with names beginning with the symbols printed will be displayed on standard "Ctrl+Space" key combination. To add existing styling to the chosen element just point to the necessary one. Each time you select any class it is displayed in the Preview tab. Click on the *Apply* button will apply the changes without closing the window.

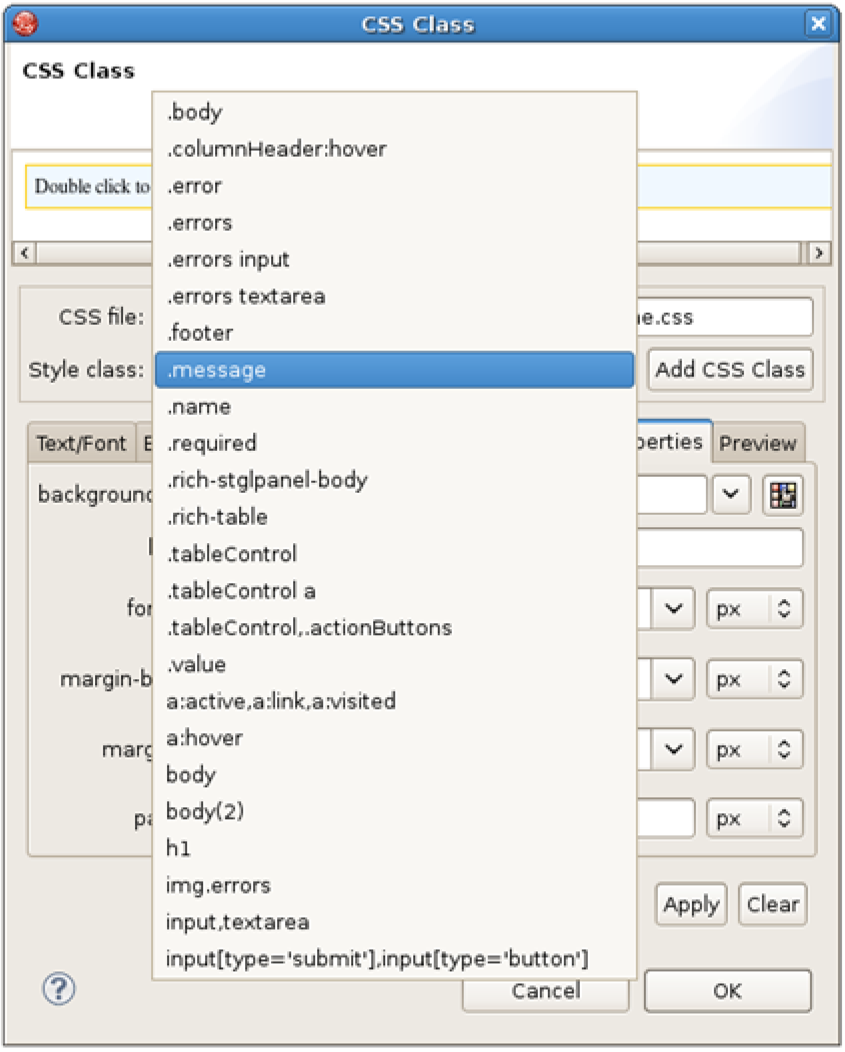
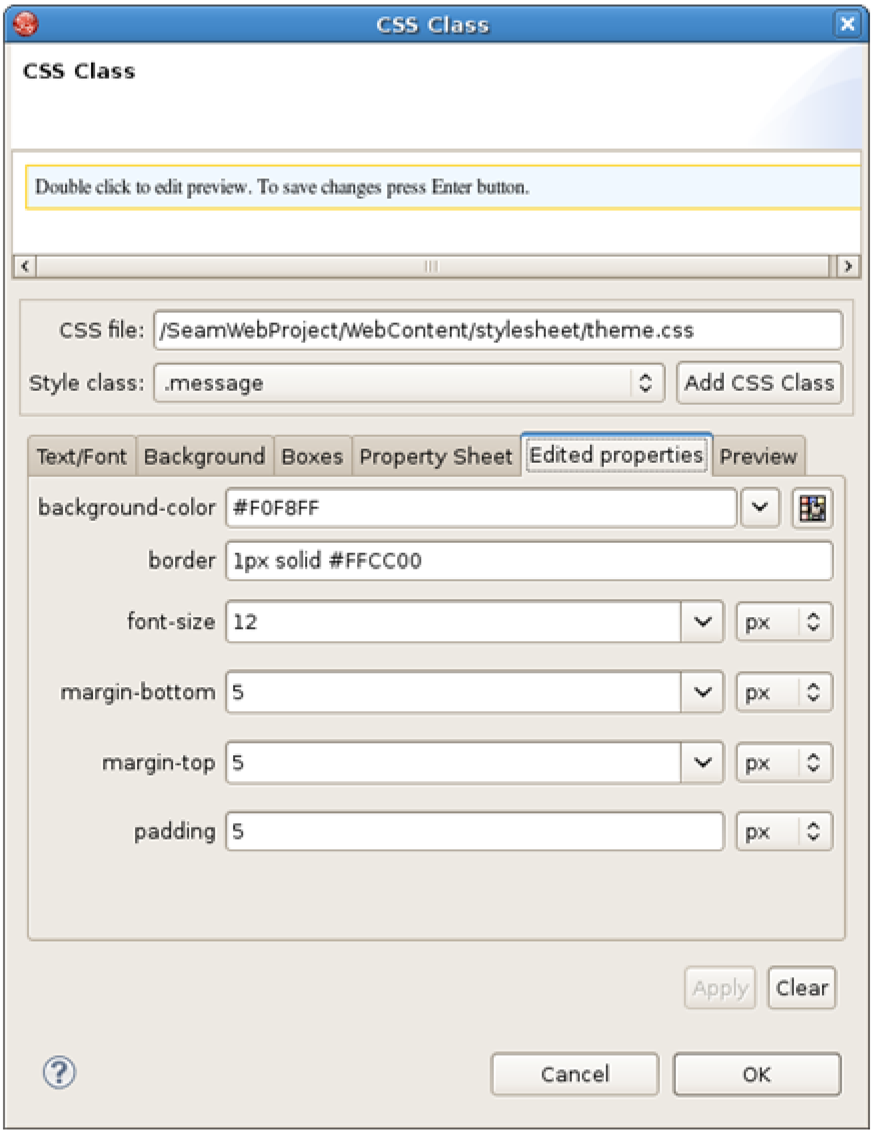

CSS Class
CSS Class
Double click to
CSS file:
e.css
Style class:
Add CSS Class
Text/Font
E
perties
Preview
backgroun
for
margin-b
marg
pa
px
px
px
px
Apply
Clear
Cancel
OK
.body
.columnHeader:hover
.error
.errors
.errors input
.errors textarea
.footer
.message
.name
.required
.rich-stglpanel-body
.rich-table
.tableControl
.tableControl a
.tableControl,.actionButtons
.value
a:active,a:link,a:visited
a:hover
body
body(2)
h1
img.errors
input,textarea
input[type='submit'],input[type='button']

Figure 3.52. Style Class Selection

The *Edited properties* tab 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 Class
CSS Class
Text/Font
Background
Boxes
Property Sheet
Edited properties
Preview
Add CSS Class
Apply
Clear
Cancel
OK

Double click to edit preview. To save changes press Enter button.

CSS file: /SeamWebProject/WebContent/stylesheet/theme.css

Style class: .message

background-color #F0F8FF

border 1px solid #FFCC00

font-size 12 px

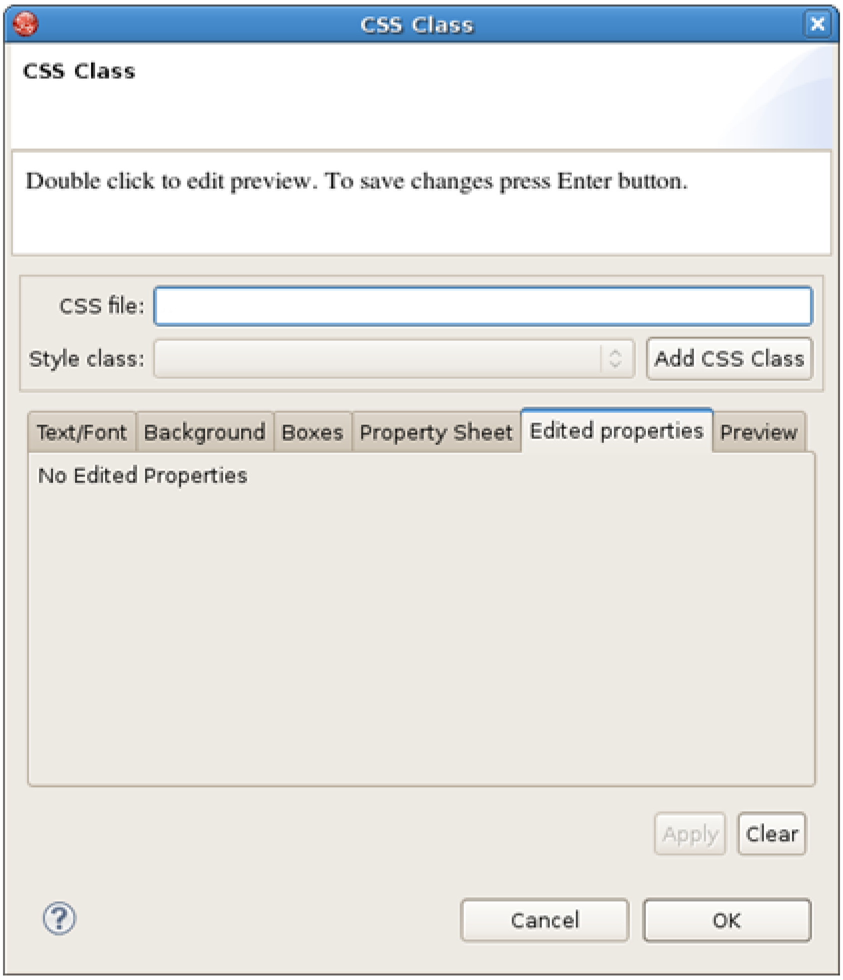
margin-bottom 5 px

margin-top 5 px

padding 5 px

Figure 3.53. Edited Properties

If the style class isn't chosen, the tab doesn't show any properties.


CSS Class
CSS Class

Double click to edit preview. To save changes press Enter button.

CSS file:

Style class:

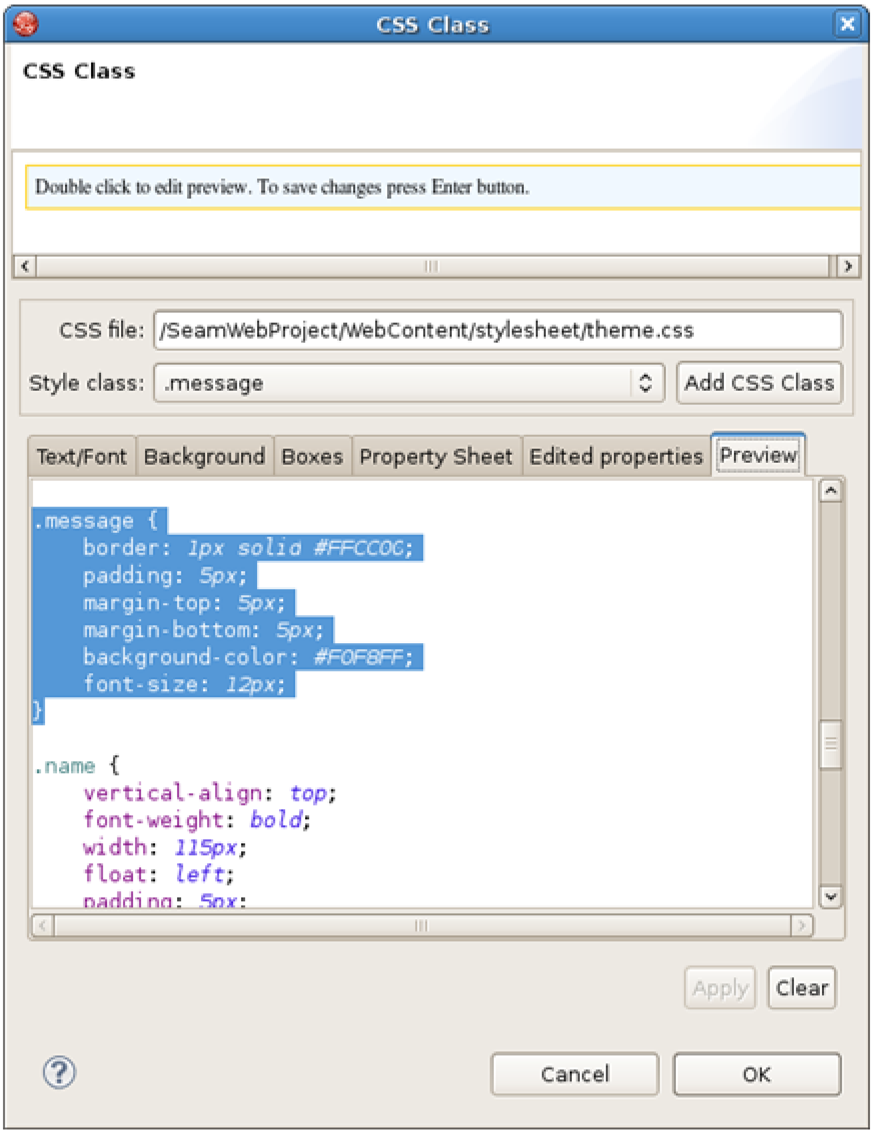
Add CSS Class
Text/Font
Background
Boxes
Property Sheet
Edited properties
Preview

No Edited Properties

Apply
Clear
Cancel
OK

Figure 3.54. Edited Properties when the style class isn't chosen

The *Preview tab* is for observing the content of the chosen CSS file. This tab is hidden if no CSS file is chosen.


CSS Class
CSS Class
Text/Font
Background
Boxes
Property Sheet
Edited properties
Preview
Add CSS Class
Apply
Clear
Cancel
OK

Double click to edit preview. To save changes press Enter button.

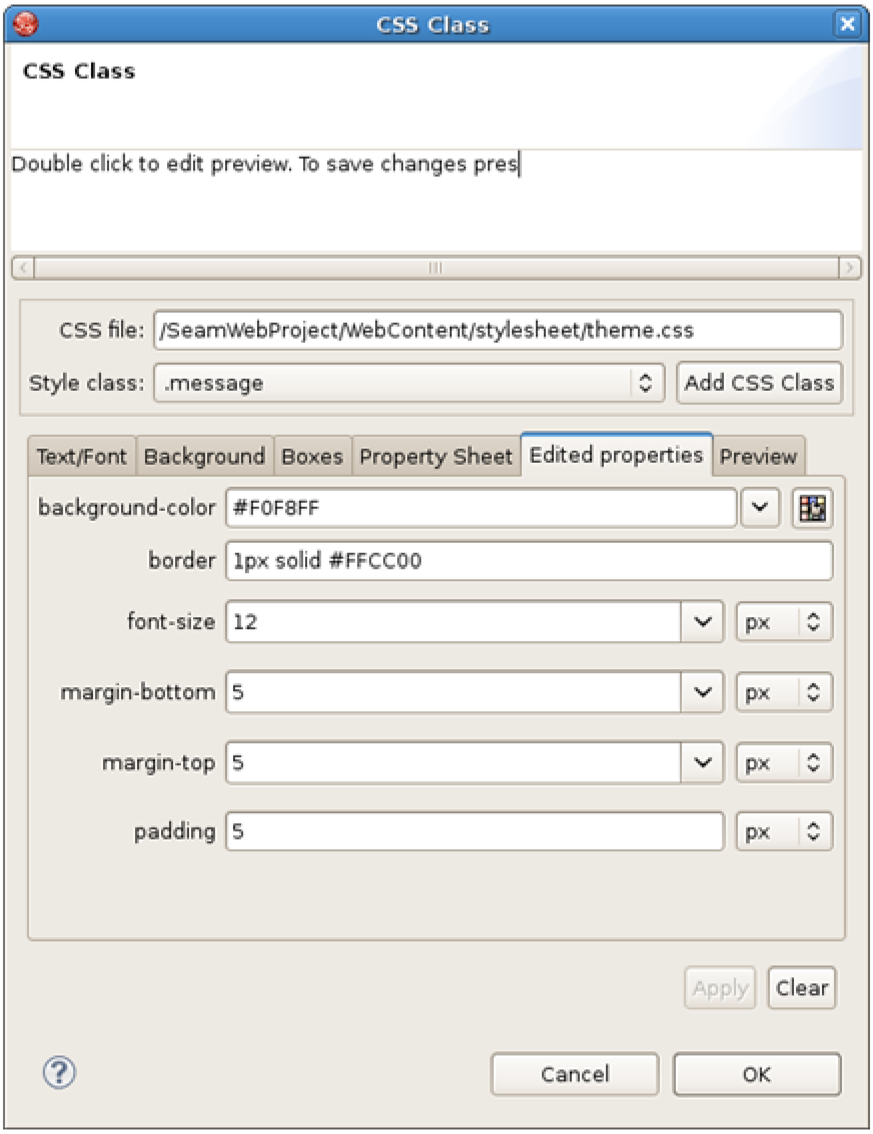
CSS file: /SeamWebProject/WebContent/stylesheet/theme.css

Style class: .message

```
.message {
    border: 1px solid #FFCC0C;
    padding: 5px;
    margin-top: 5px;
    margin-bottom: 5px;
    background-color: #F0F8FF;
    font-size: 12px;
}

.name {
    vertical-align: top;
    font-weight: bold;
    width: 115px;
    float: left;
    padding: 5px;
```

Figure 3.55. Preview Tab

At the top of the CSS Class Dialog you can see a preview box which visualizes the result. To edit the preview you should double click in the box. To leave the focus, use *Ctrl + Tab*.



CSS Class

CSS Class

Double click to edit preview. To save changes pres

CSS file: /SeamWebProject/WebContent/stylesheet/theme.css

Style class: .message

Add CSS Class
Text/Font
Background
Boxes
Property Sheet
Edited properties
Preview

background-color: #F0F8FF

border: 1px solid #FFCC00

font-size: 12 px

margin-bottom: 5 px

margin-top: 5 px

padding: 5 px

Apply
Clear
Cancel
OK

Figure 3.56. Editing the Preview

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

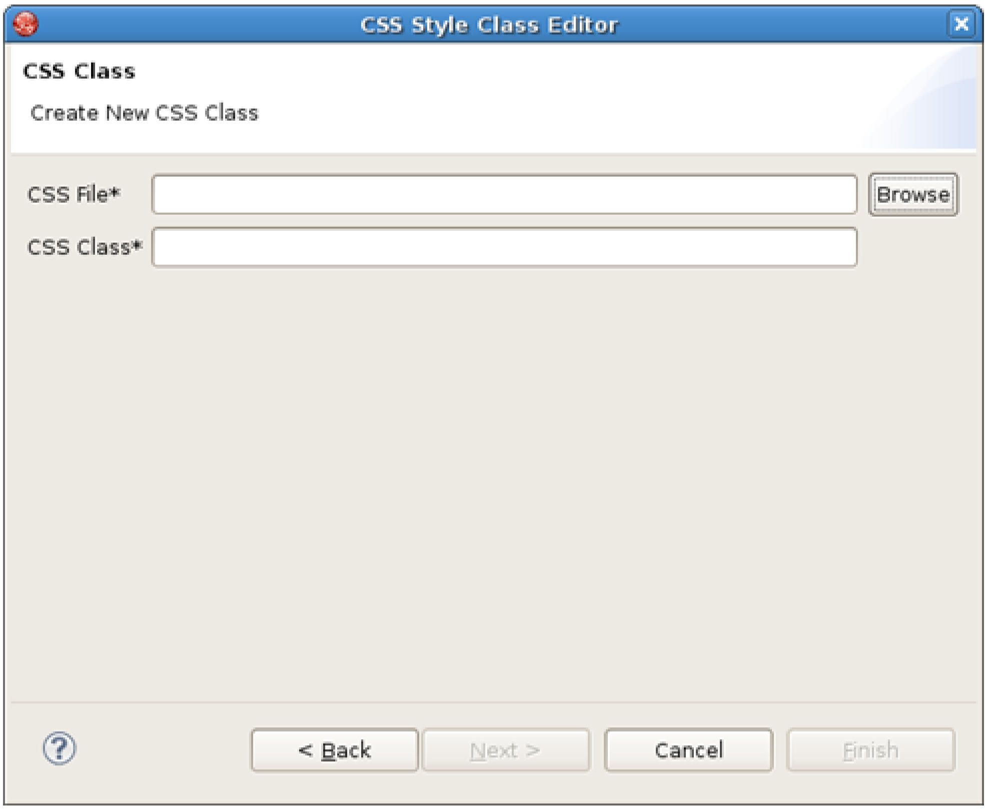

CSS Style Class Editor
CSS Class
Create New CSS Class
CSS File*
Browse
CSS Class*
< Back
Next >
Cancel
Finish

Figure 3.57. New CSS Class Dialog

Click on the Browse button to open a dialog to select the CSS file to create a CSS class for:

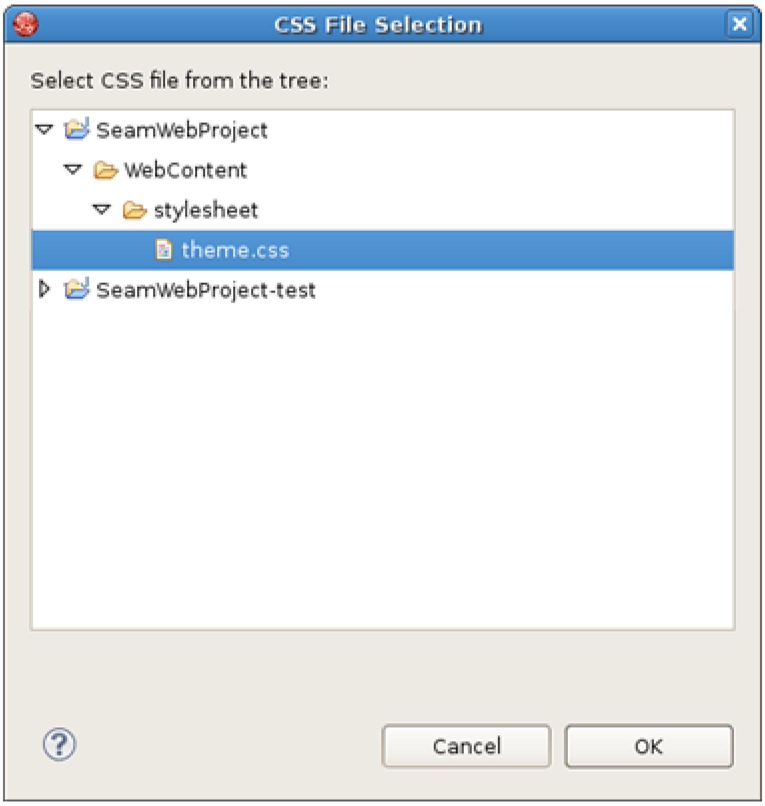

CSS File Selection
Select CSS file from the tree:
SeamWebProject
WebContent
stylesheet
theme.css
SeamWebProject-test
Cancel
OK

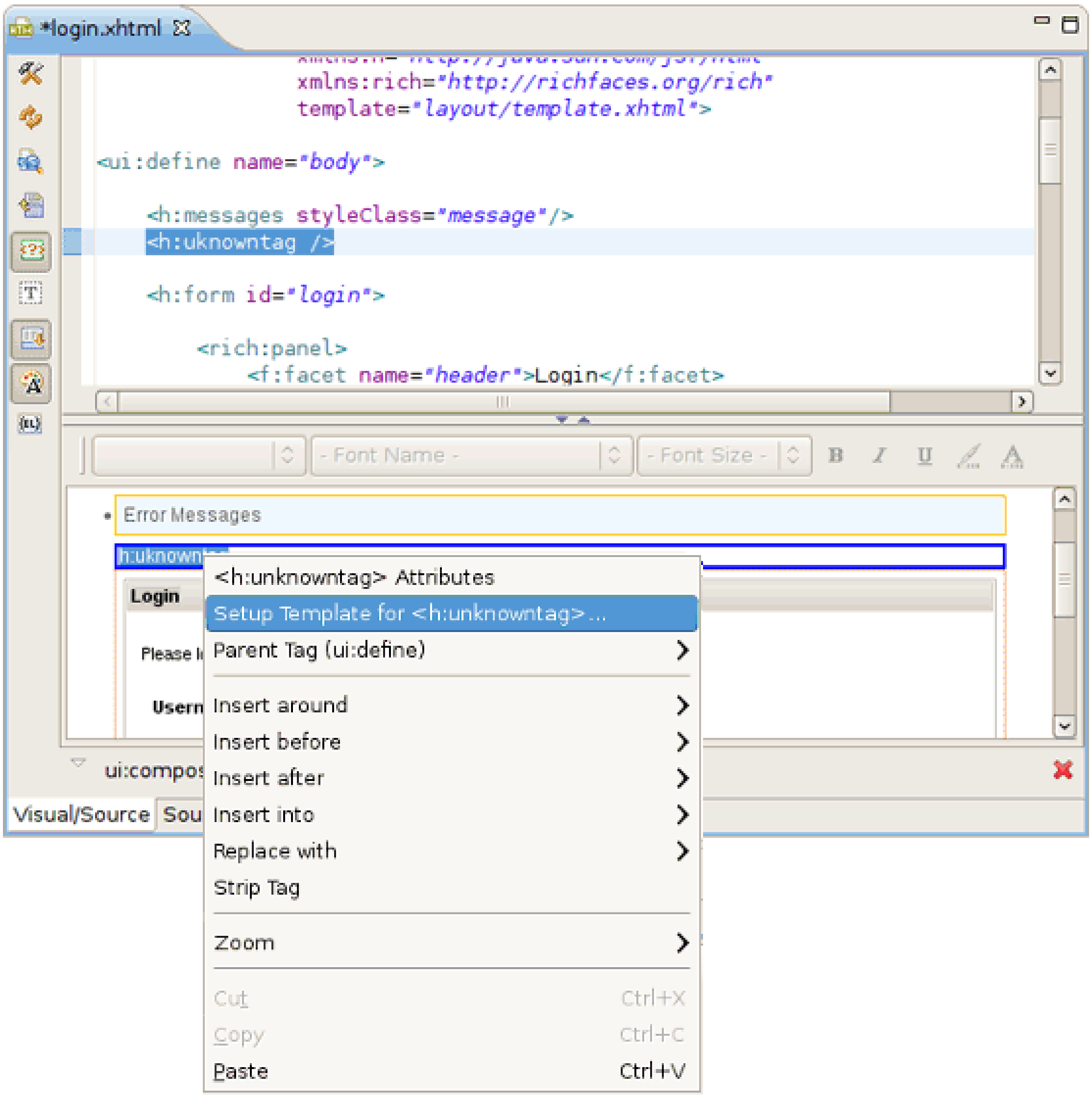
Figure 3.58. CSS File Selection

Choose the necessary CSS file and click on the Ok button.

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 *Setup Template for <tag name>* option.

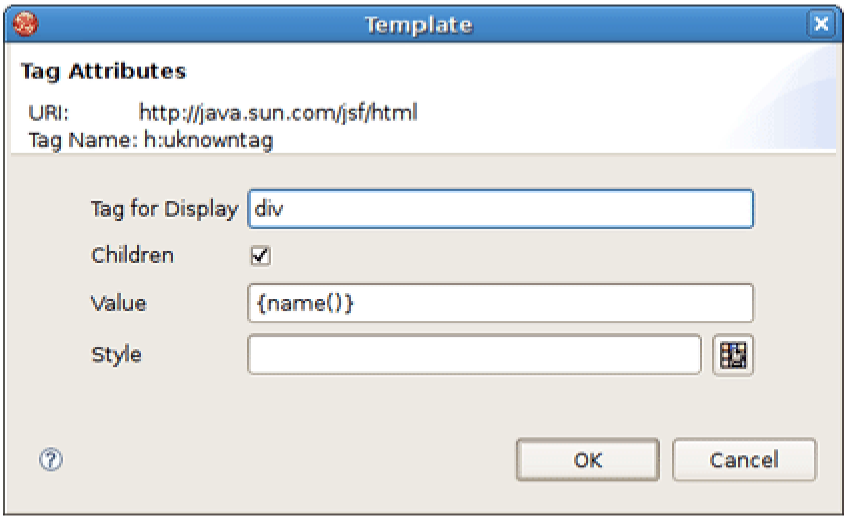


```
                  xmlns:rich="http://richfaces.org/rich"
                  template="layout/template.xhtml">

<ui:define name="body">

    <h:messages styleClass="message"/>
    <h:uknowntag />

    <h:form id="login">

        <rich:panel>
            <f:facet name="header">Login</f:facet>
```

*login.xhtml
- Font Name -
- Font Size -
Error Messages
h:uknown
Login
Please l
Usern
ui:compos
Visual/Source
Sou
<h:unknowntag> Attributes
Setup Template for <h:unknowntag>...
Parent Tag (ui:define)
Insert around
Insert before
Insert after
Insert into
Replace with
Strip Tag
Zoom
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V

Figure 3.59. Calling Template Dialog

Here is what the Template dialog looks like.



URI: http://java.sun.com/jsf/html
Tag Name: h:uknowntag

Tag for Display: div
Children: ☑
Value: {name()}
Style:

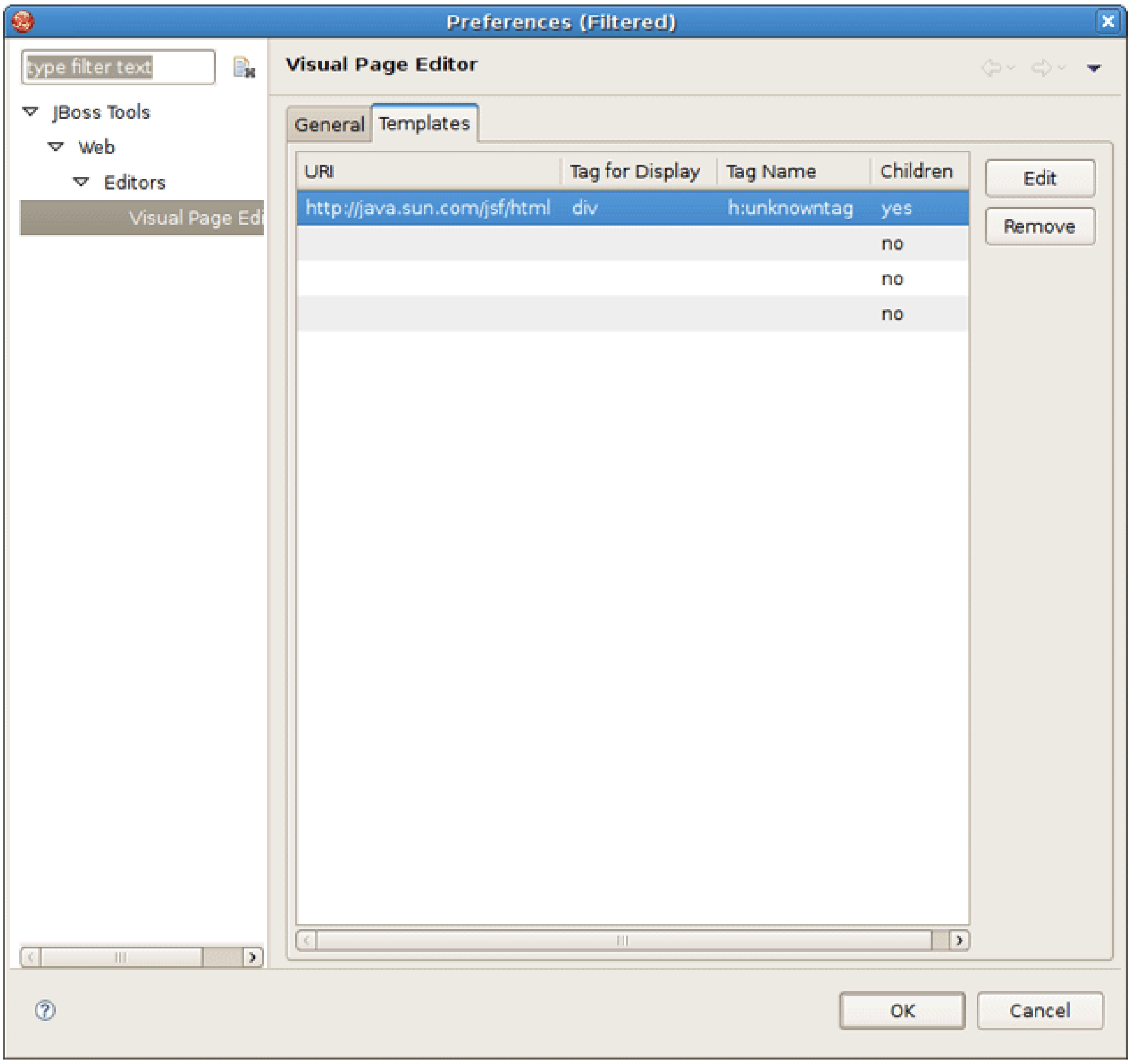
Figure 3.60. Template Dialog

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. Check *Children* , if you want to mark a tag as a child element.

The *Value* field is for setting a tag value.

As for the *Style* field, you can fill it out manually or make use of the button next to the field to bring the *CSS Dialog [42]* for editing styles.

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


Preferences (Filtered)
type filter text
JBoss Tools
Web
Editors
Visual Page Edi
Visual Page Editor
General
Templates
Edit
Remove
OK
Cancel

URI	Tag for Display	Tag Name	Children
http://java.sun.com/jsf/html	div	h:unknowntag	yes
			no
			no
			no

Figure 3.61. Templates Tab of the VPE Preferences Page

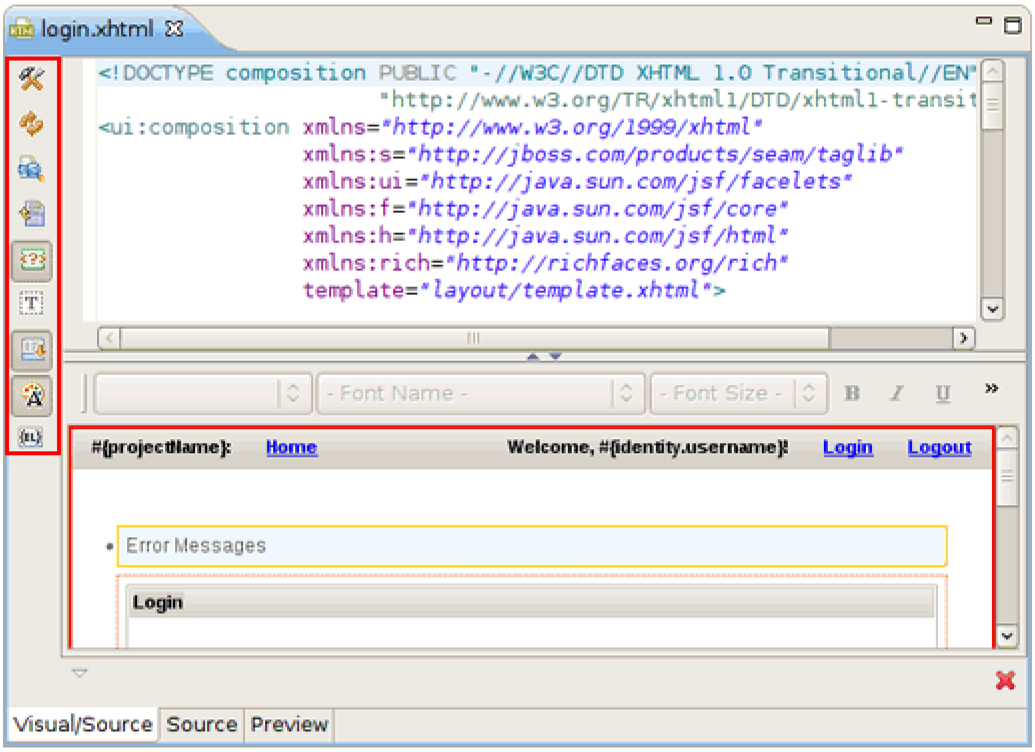
Here it's possible to edit or remove any listed in the table template.

3.2.4. VPE Toolbar

The Visual Page Editor toolbar includes the next buttons:

- *Preferences* ( )
- *Refresh* ( )

- Page Design Options ( )
- the button to switch the current Visual/Source layout ( )
- Show border for unknown tags ( )
- Show non-visual tags ( )
- Show selection bar ( )
- Show text formatting bar ( )
- Show bundle's messages as EL expressions ( )



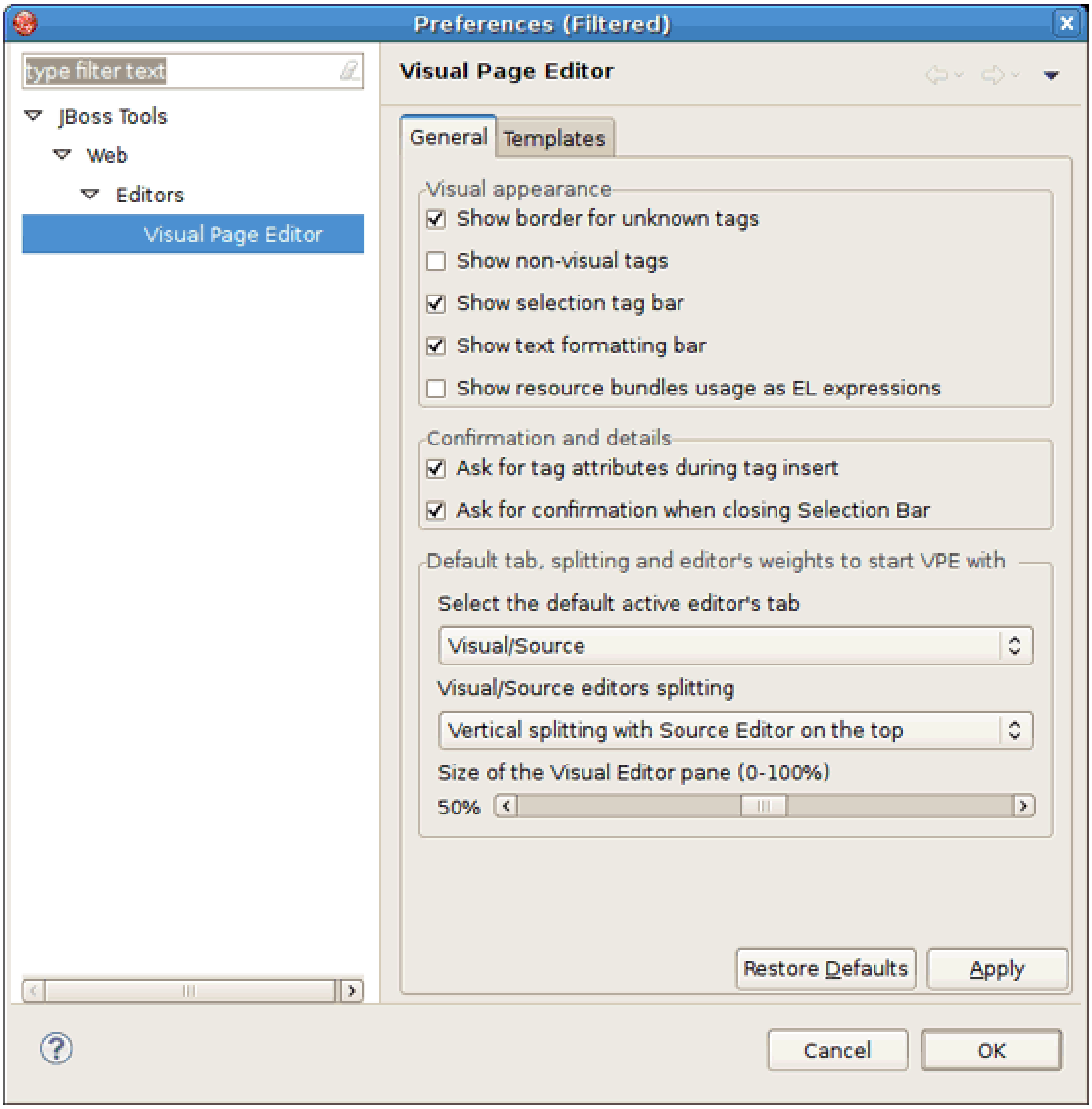
login.xhtml
- Font Name -
- Font Size -
#{projectName}:
Home
Welcome, #{identity.username}!
Login
Logout
Error Messages
Login
Visual/Source
Source
Preview

```
<!DOCTYPE composition PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
                  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transit
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
               xmlns:s="http://jboss.com/products/seam/taglib"
               xmlns:ui="http://java.sun.com/jsf/facelets"
               xmlns:f="http://java.sun.com/jsf/core"
               xmlns:h="http://java.sun.com/jsf/html"
               xmlns:rich="http://richfaces.org/rich"
               template="layout/template.xhtml">
```

Figure 3.62. Buttons on the VPE Toolbar

3.2.4.1. Preferences

The *Preferences* button ( ) provides a quick access to the Visual Page Editor preferences.


Preferences (Filtered)
type filter text
JBoss Tools
Web
Editors
Visual Page Editor
General
Templates
Restore Defaults
Apply
Cancel
OK

Visual Page Editor

Visual appearance
☑ Show border for unknown tags
☐ Show non-visual tags
☑ Show selection tag bar
☑ Show text formatting bar
☐ Show resource bundles usage as EL expressions

Confirmation and details
☑ Ask for tag attributes during tag insert
☑ Ask for confirmation when closing Selection Bar

Default tab, splitting and editor's weights to start VPE with
Select the default active editor's tab: Visual/Source
Visual/Source editors splitting: Vertical splitting with Source Editor on the top
Size of the Visual Editor pane (0-100%): 50%

Figure 3.63. Visual Page Editor Preferences Window

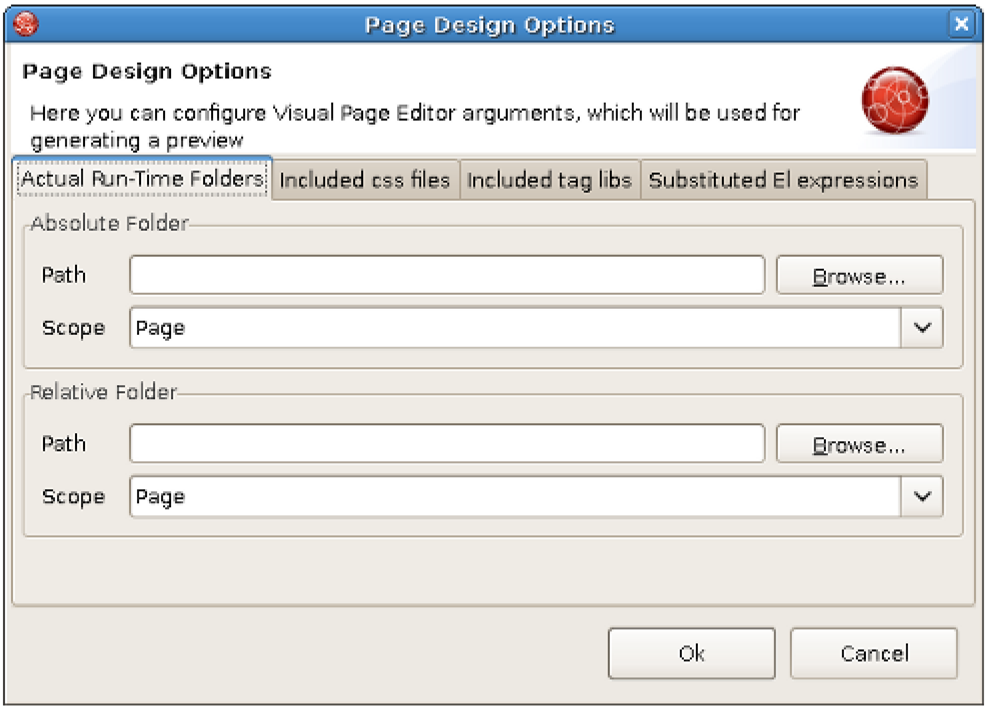
This page provides a number of options associated with the editor representation. The more detailed description on each one you can find in the "JBoss Tools Preferences" chapter under *Visual Page Editor*.

3.2.4.2. Refresh

Clicking on the *Refresh* button ( ) refreshes the displayed information.

3.2.4.3. Page Design Options

The *Page Design Options* button ( ) leads to a window which helps you specify necessary references of the resources. It is represented by a window with 4 tabs. The first one, *Actual Run-Time folders*, is used to replace absolute and relative path values when generating a preview:


Page Design Options

Here you can configure Visual Page Editor arguments, which will be used for generating a preview

Absolute Folder

Path:

Scope: Page

Relative Folder

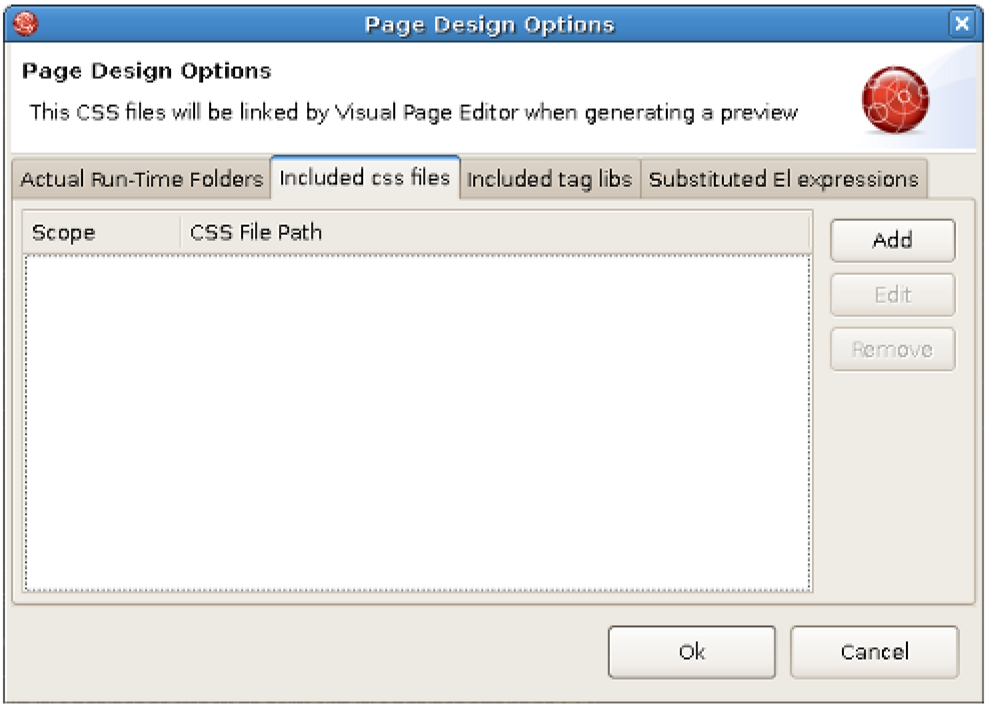
Path:

Scope: Page

Page Design Options
Actual Run-Time Folders
Included css files
Included tag libs
Substituted El expressions
Browse...
Browse...
Ok
Cancel

Figure 3.64. Page Design Options: Actual Run-Time folders

The second tab, *Included CSS files* , is used to add CSS files to be linked by Visual Page Editor when generating a preview:

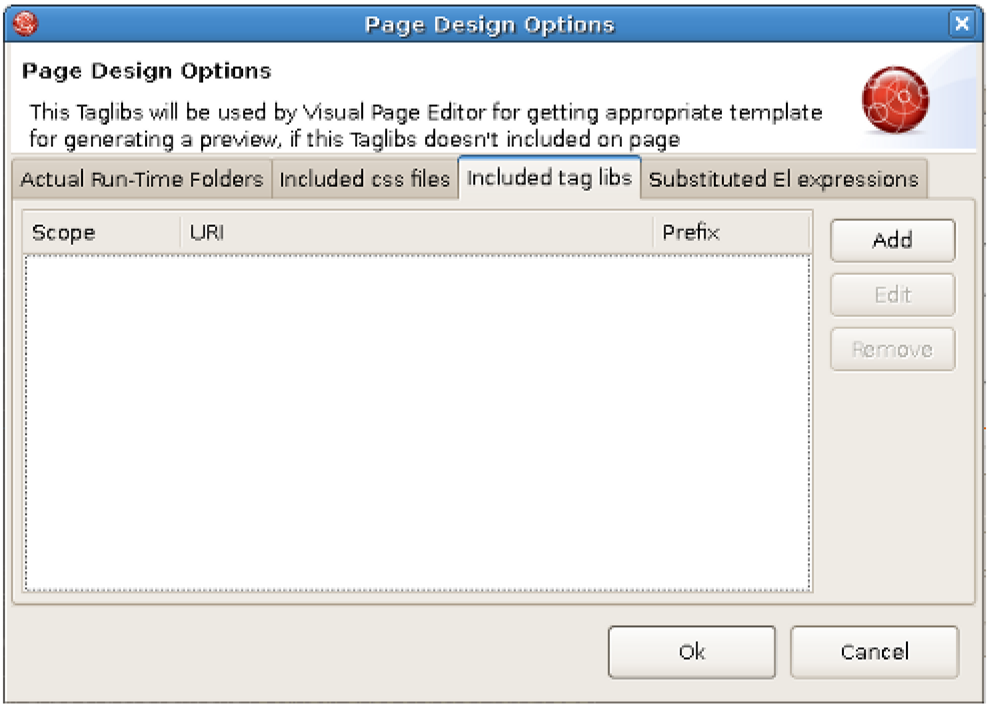

Page Design Options
Actual Run-Time Folders
Included css files
Included tag libs
Substituted El expressions
Scope
CSS File Path
Add
Edit
Remove
Ok
Cancel

Page Design Options

This CSS files will be linked by Visual Page Editor when generating a preview

Figure 3.65. Page Design Options: Included CSS files

The third one, *Included tag libs*, can be used to add Taglibs that can be used by the editor for getting appropriate templates to generate a preview:


Page Design Options
Actual Run-Time Folders
Included css files
Included tag libs
Substituted El expressions
Scope
URI
Prefix
Add
Edit
Remove
Ok
Cancel

Page Design Options

This Taglibs will be used by Visual Page Editor for getting appropriate template for generating a preview, if this Taglibs doesn't included on page

Figure 3.66. Page Design Options: Included tag libs

And finally, the *Substituted El expressions* tab is used to add El expressions that will be substituted by the editor when generating a preview:

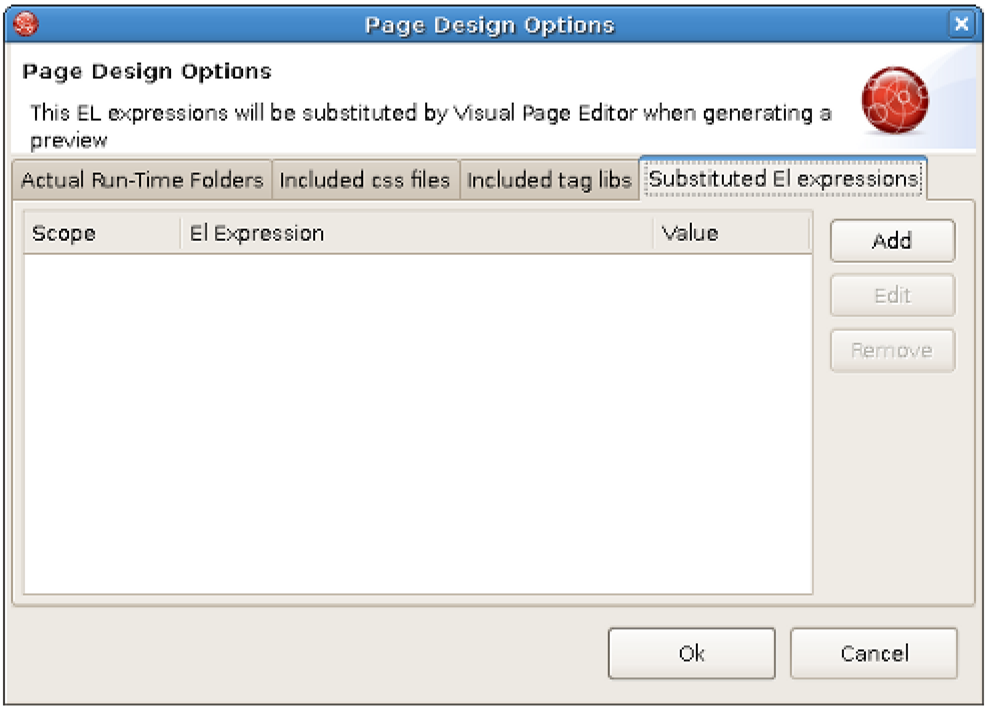

Page Design Options
Page Design Options
This EL expressions will be substituted by Visual Page Editor when generating a preview
Actual Run-Time Folders
Included css files
Included tag libs
Substituted El expressions
Scope
El Expression
Value
Add
Edit
Remove
Ok
Cancel

Figure 3.67. Page Design Options: Substituted El expressions

The first two tabs of the window let you define actual runtime folders. The example below will help you understand how this can be done.

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

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.

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

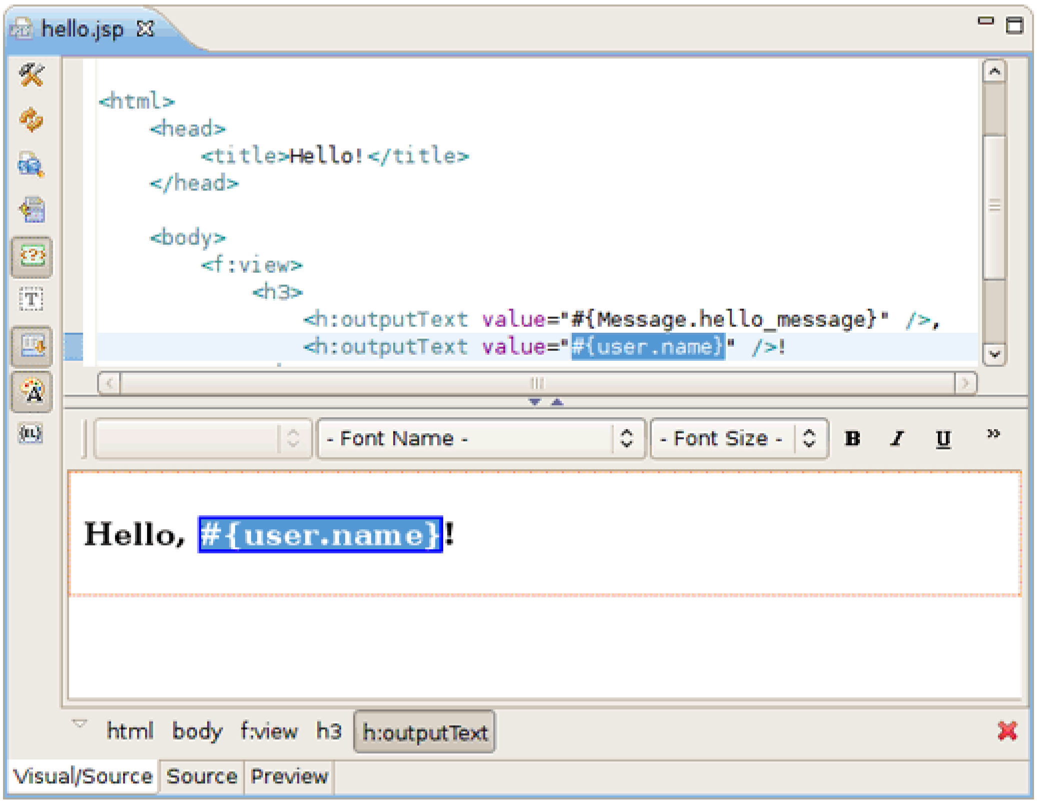
```
<link rel="stylesheet" type="text/css"
   href="#{facesContext.externalContext.requestContextPath}/style.css"/>
```

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:

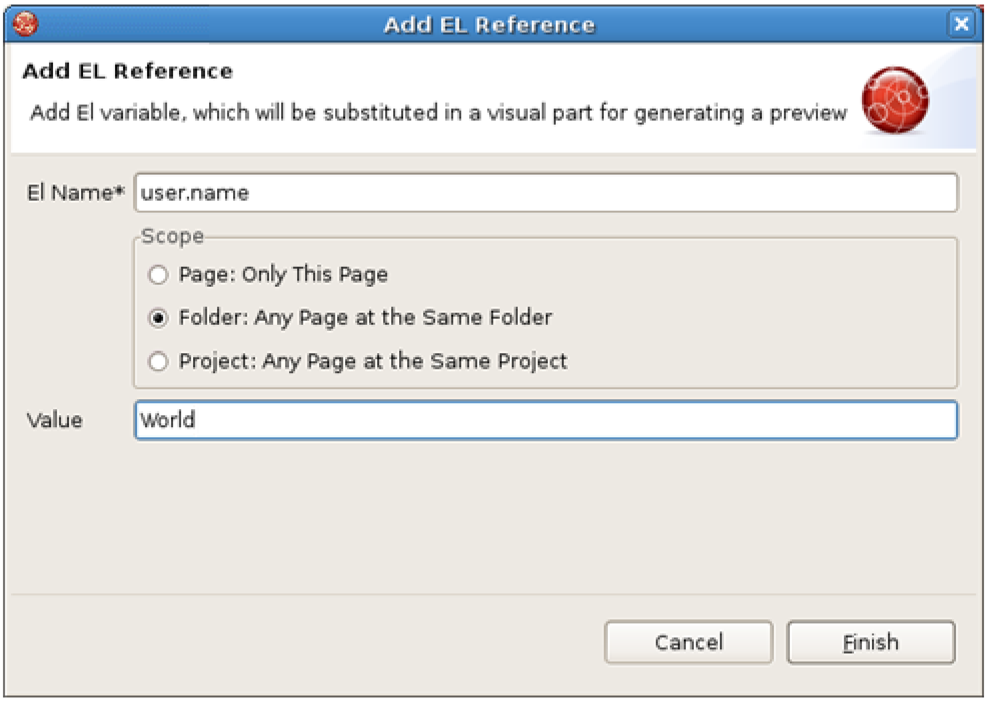

hello.jsp
- Font Name -
- Font Size -
B
I
U
html
body
f:view
h3
h:outputText
Visual/Source
Source
Preview

```
<html>
    <head>
        <title>Hello!</title>
    </head>

    <body>
        <f:view>
            <h3>
                <h:outputText value="#{Message.hello_message}" />,
                <h:outputText value="#{user.name}" />!
```

Hello, #{user.name}!

Figure 3.68. 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.*



Add EL Reference

Add El variable, which will be substituted in a visual part for generating a preview

El Name* user.name

Scope

○ Page: Only This Page

● Folder: Any Page at the Same Folder

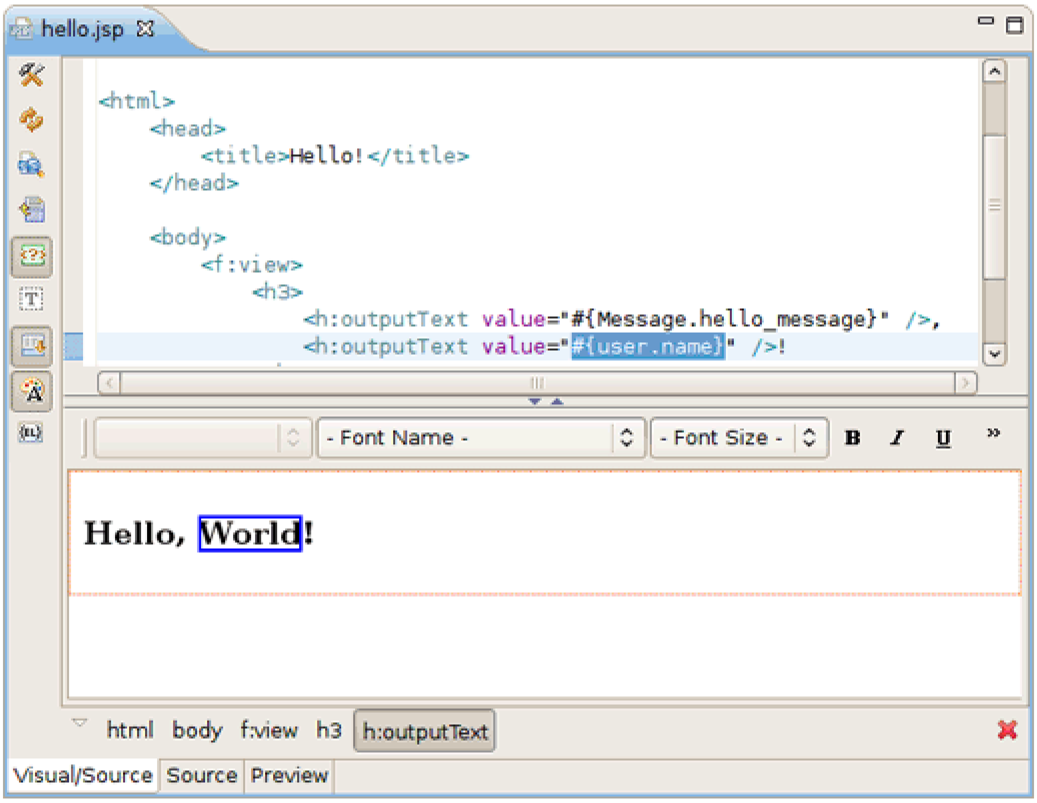
○ Project: Any Page at the Same Project

Value World

Add EL Reference
Cancel
Finish

Figure 3.69. Setting the Value for the EL Expression

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



```
<html>
    <head>
        <title>Hello!</title>
    </head>

    <body>
        <f:view>
            <h3>
                <h:outputText value="#{Message.hello_message}" />,
                <h:outputText value="#{user.name}" />!
```

Hello, World!

hello.jsp
- Font Name -
- Font Size -
B
I
U
html
body
f:view
h3
h:outputText
Visual/Source
Source
Preview

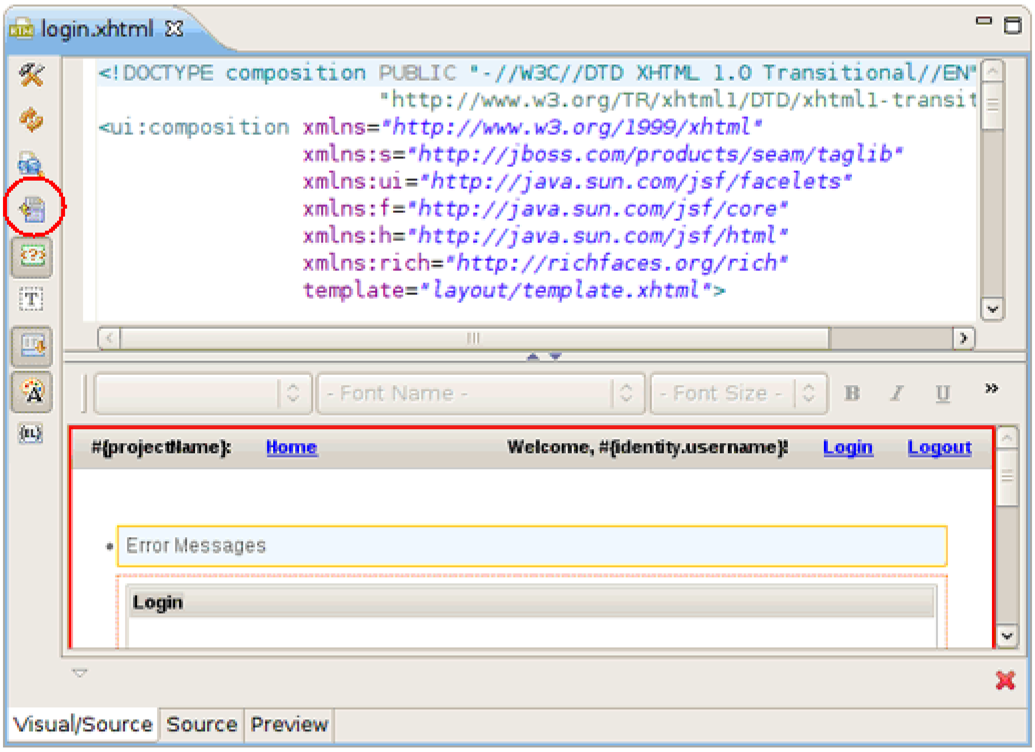
Figure 3.70. The EL Expression Value

3.2.4.4. Visual/Source Editors splitting buttons

The *Visual/Source Editors splitting buttons* provide the possibility to choose one of the four possible layouts for the Visual/Source Editor.

The available layouts and corresponding buttons are as follows:

- Vertical Source on top( )
- Vertical Visual on top ( )
- Horizontal Source to the left ( )
- Horizontal Visual to the left ( )



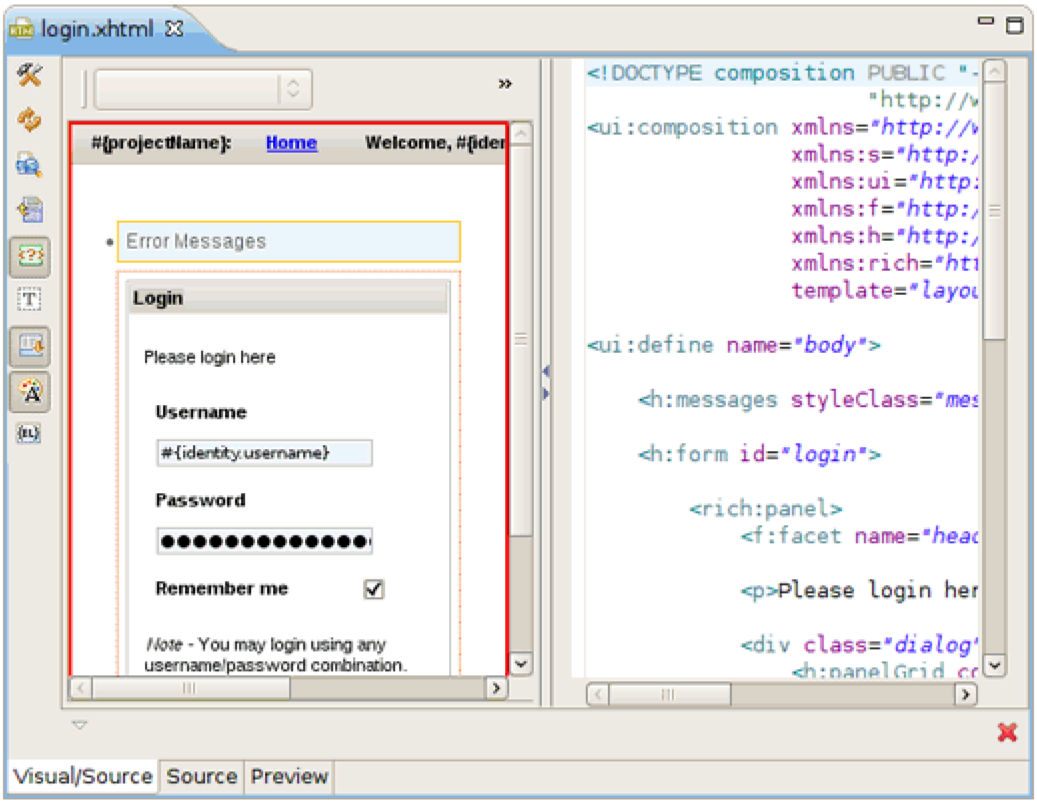
login.xhtml

```
<!DOCTYPE composition PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
                 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transit
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
                xmlns:s="http://jboss.com/products/seam/taglib"
                xmlns:ui="http://java.sun.com/jsf/facelets"
                xmlns:f="http://java.sun.com/jsf/core"
                xmlns:h="http://java.sun.com/jsf/html"
                xmlns:rich="http://richfaces.org/rich"
                template="layout/template.xhtml">
```

- Font Name -
- Font Size -
B
I
U
#{projectName}: Home Welcome, #{identity.username}! Login Logout
Error Messages
Login
Visual/Source
Source
Preview

Figure 3.71. Visual Page Editor Before Layout Changing

Note, at the current view there is only *one* button, that proposes the possibility to change it in order the Source and the View are moved *in a clockwise direction*.



#{projectName}: Home Welcome, #{ider

Error Messages

Login

Please login here

Username

#{identity.username}

Password

●●●●●●●●●●●●●●

Remember me ☑

Note - You may login using any username/password combination.

```
<!DOCTYPE composition PUBLIC "-
                        "http://w
<ui:composition xmlns="http://w
                xmlns:s="http:/
                xmlns:ui="http:
                xmlns:f="http:/
                xmlns:h="http:/
                xmlns:rich="htt
                template="layou

<ui:define name="body">

    <h:messages styleClass="mes

    <h:form id="login">

        <rich:panel>
            <f:facet name="head

            <p>Please login her

            <div class="dialog"
                <h:panelGrid co
```

Figure 3.72. Visual Page Editor After Layout Changing

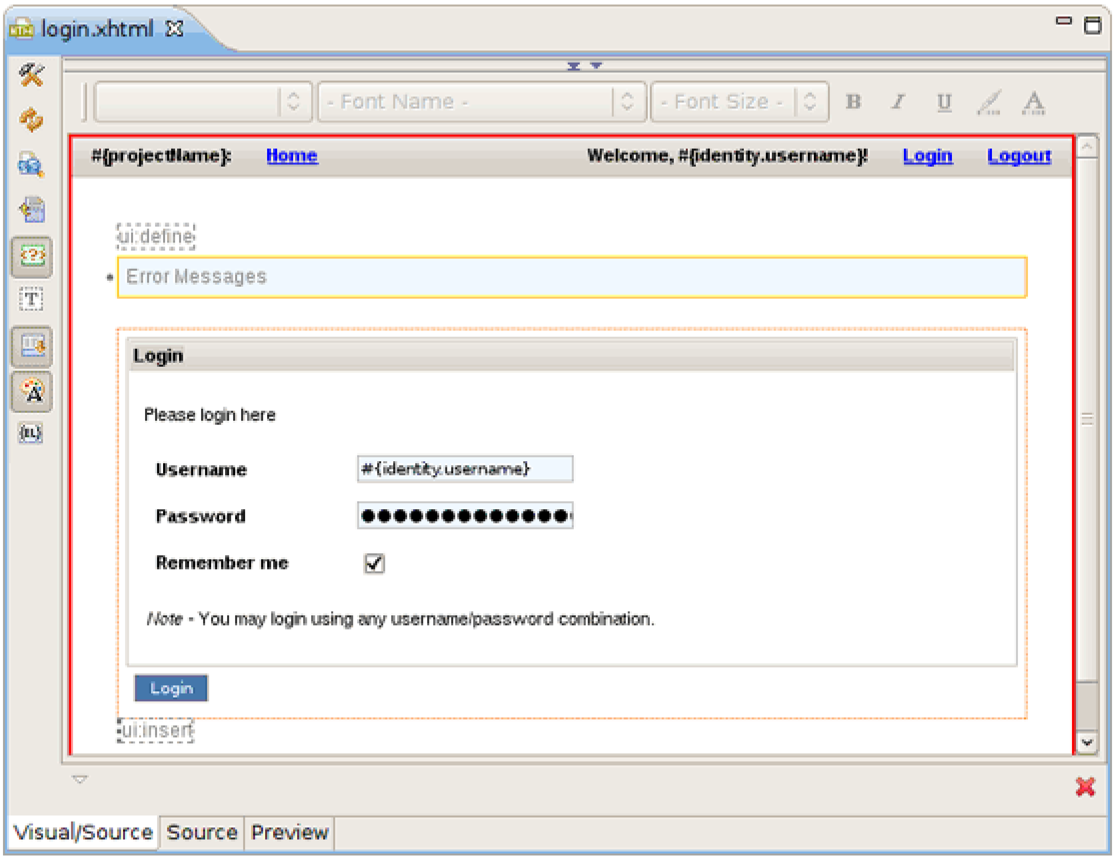
3.2.4.5. Show Border for Unknown Tags

The option is a self explanatory, i. e. if you want unknown tags to be wrapped in a border in the VPE visual part, just press the *Show border for unknown tags* button ( ) on the toolbar.

3.2.4.6. Show Non-visual Tags

Visual Page Editor provides the option for displaying non-visual tags in Visual mode of the editor. To enable it select the *Show non-visual tags* button ( ) on the VPE toolbar.

On the figure you can see non-visual elements with gray dashed borders.


login.xhtml
- Font Name -
- Font Size -
B
I
U
A
#{projectName}:
Home
Welcome, #{identity.username}!
Login
Logout
ui:define
Error Messages
Login
Username
Password
Remember me
Login
ui:insert
Visual/Source
Source
Preview

Please login here

Username #{identity.username}

Password ●●●●●●●●●●●●●●

Remember me ☑

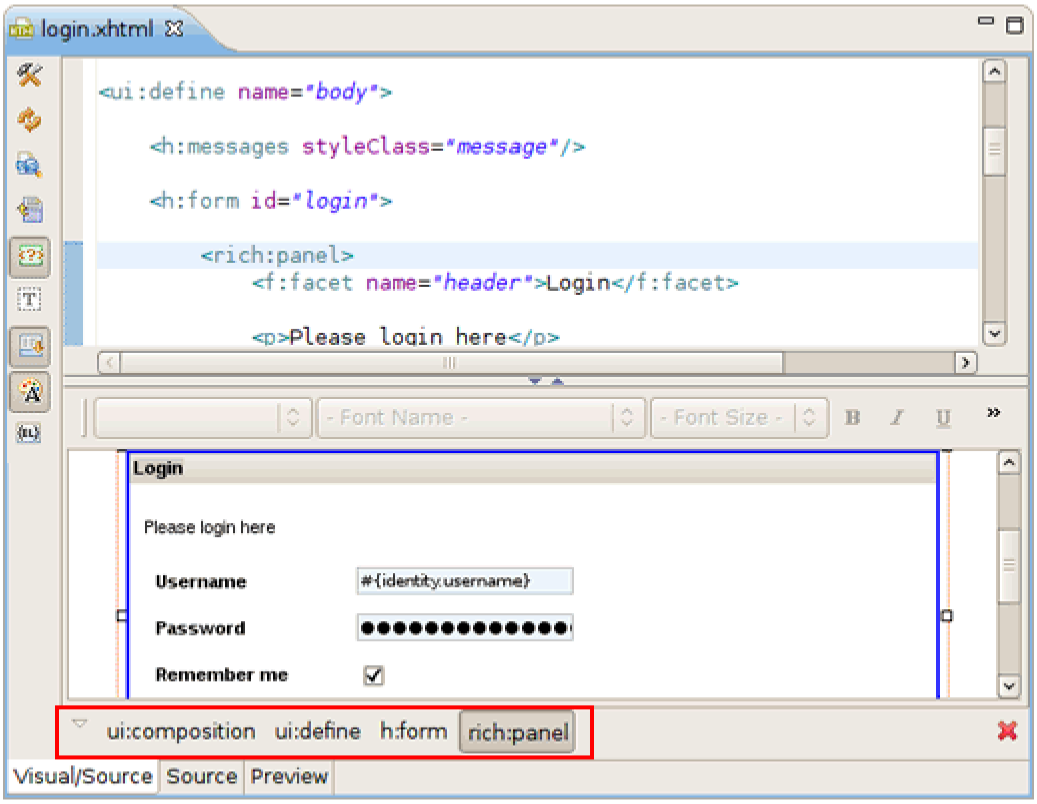
Note - You may login using any username/password combination.

Figure 3.73. Non-visual Tag in the VPE

You can also switch on this option in the VPE preferences, having clicked on the *Preferences* button (🛠).

3.2.4.7. Show Selection Bar

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's updated automatically allowing to see tags tree for a current component selected in Visual or Source mode, also it allows to select tags back and forward.


login.xhtml
- Font Name -
- Font Size -
B
I
U
ui:composition
ui:define
h:form
rich:panel
Visual/Source
Source
Preview

```
<ui:define name="body">

    <h:messages styleClass="message"/>

    <h:form id="login">

        <rich:panel>
            <f:facet name="header">Login</f:facet>

            <p>Please login here</p>
```

Login

Please login here

Username	#{identity.username}
Password	●●●●●●●●●●●●●●
Remember me	☑

Figure 3.74. Selection Tag Bar

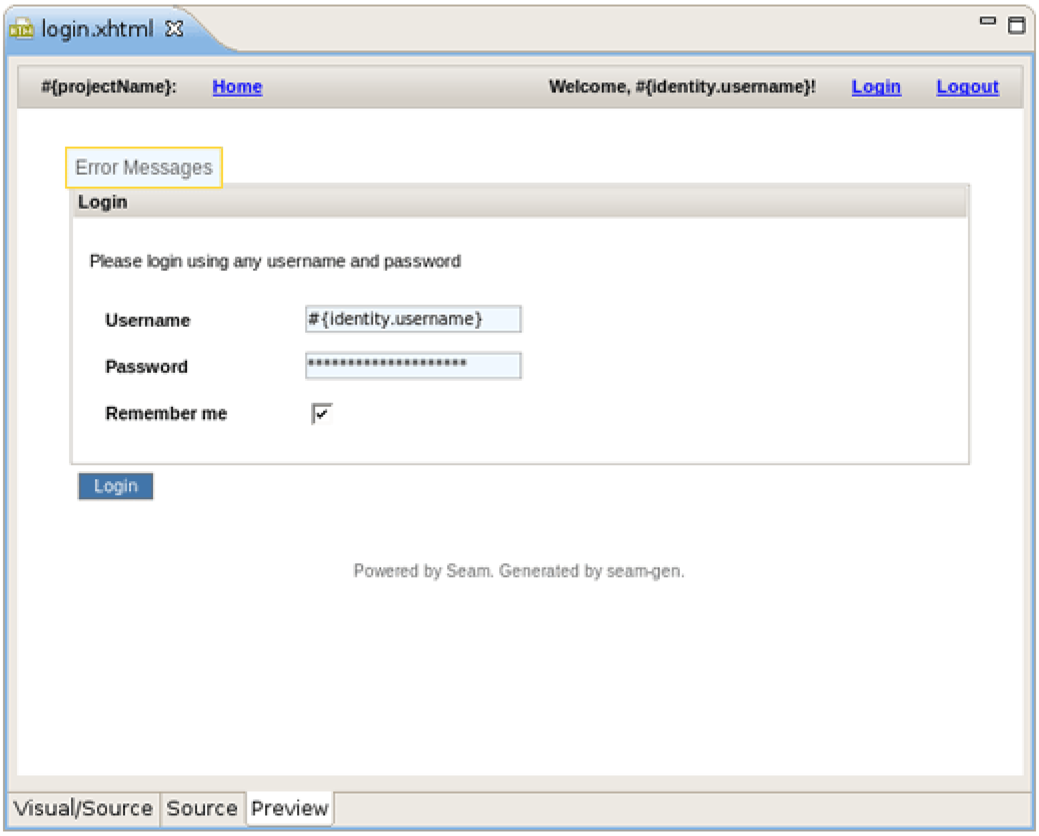
If you want to hide the Selection Tag Bar, use the *Show Selection Bar* button ( ) on the VPE toolbar.

3.2.5. Page Preview

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

- Struts Pages
- JSF Pages
- Seam Pages

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


login.xhtml
#{projectName}:
Home
Welcome, #{identity.username}!
Login
Logout
Error Messages
Login
Login
Visual/Source
Source
Preview

Please login using any username and password

Username #{identity.username}

Password ********************

Remember me ☑

Powered by Seam. Generated by seam-gen.

Figure 3.75. Preview View

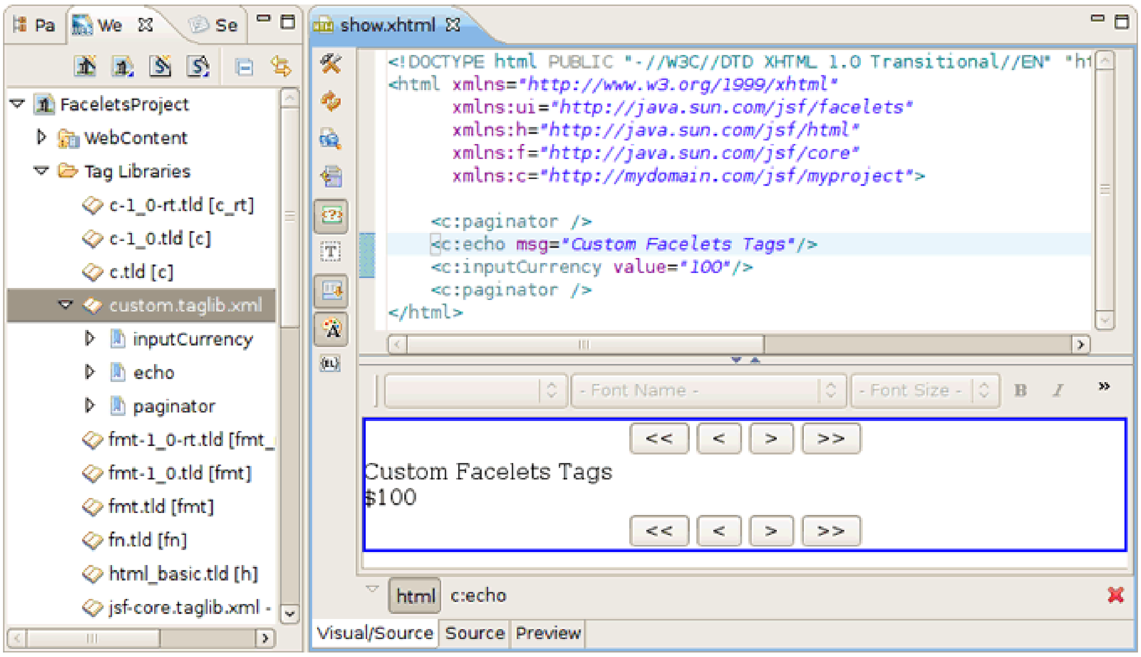
3.2.6. Support for Custom Facelets Components

Visual Page Editor (starting from 3.0.0.M3 version of JBoss Tools) supports custom Facelets tag libraries both declared in the `web.xml` file (for details, see *Creating a component*) and packed into the JAR file.

Tip:

In case of Facelets tag library packed in `.jar` , remember to put `*.taglib.xml` in right place: `[filename].jar/META-INF/*.taglib.xml`

Visual Page Editor recognizes the tags from the custom Facelets tag library and correctly renders them both in source and visual view of the editor.


Pa
We
Se
FaceletsProject
WebContent
Tag Libraries
c-1_0-rt.tld [c_rt]
c-1_0.tld [c]
c.tld [c]
custom.taglib.xml
inputCurrency
echo
paginator
fmt-1_0-rt.tld [fmt_
fmt-1_0.tld [fmt]
fmt.tld [fmt]
fn.tld [fn]
html_basic.tld [h]
jsf-core.taglib.xml -
show.xhtml
- Font Name -
- Font Size -
B
I
<<
<
>
>>
html
c:echo
Visual/Source
Source
Preview

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "ht
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:c="http://mydomain.com/jsf/myproject">

    <c:paginator />
    <c:echo msg="Custom Facelets Tags"/>
    <c:inputCurrency value="100"/>
    <c:paginator />
</html>
```

```
Custom Facelets Tags
$100
```

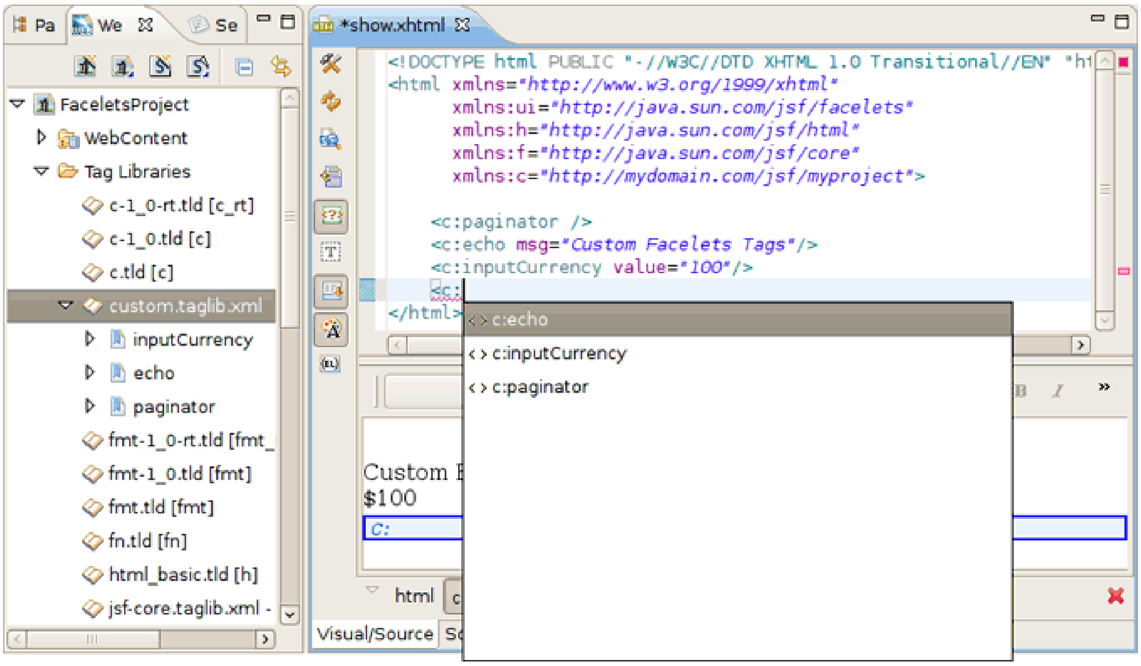
Figure 3.76. Custom Facelets Tags in the VPE

While editing an XHTML file that uses a custom Facelets components you can always make use of the following editor's features:

- *Content Assist for Custom Facelets Components*
- *OpenOn for Custom Facelets Components*

3.2.6.1. Content Assist for Custom Facelets Components

Call the content assist as usual by using *Ctrl+Space* when typing a tag. As proposals you should see custom Facelets tags defined in your Facelets tag library.



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "ht
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:c="http://mydomain.com/jsf/myproject">

   <c:paginator />
   <c:echo msg="Custom Facelets Tags"/>
   <c:inputCurrency value="100"/>
   <c:
</html>
```

Pa
We
Se
FaceletsProject
WebContent
Tag Libraries
c-1_0-rt.tld [c_rt]
c-1_0.tld [c]
c.tld [c]
custom.taglib.xml
inputCurrency
echo
paginator
fmt-1_0-rt.tld [fmt_
fmt-1_0.tld [fmt]
fmt.tld [fmt]
fn.tld [fn]
html_basic.tld [h]
jsf-core.taglib.xml -
*show.xhtml
<> c:echo
<> c:inputCurrency
<> c:paginator
Custom F
$100
c:
html
Visual/Source
Sc

Figure 3.77. Content Assist for Custom Facelets Tags

3.2.6.2. OpenOn for Custom Facelets Components

While developing using Facelets you can make use of:

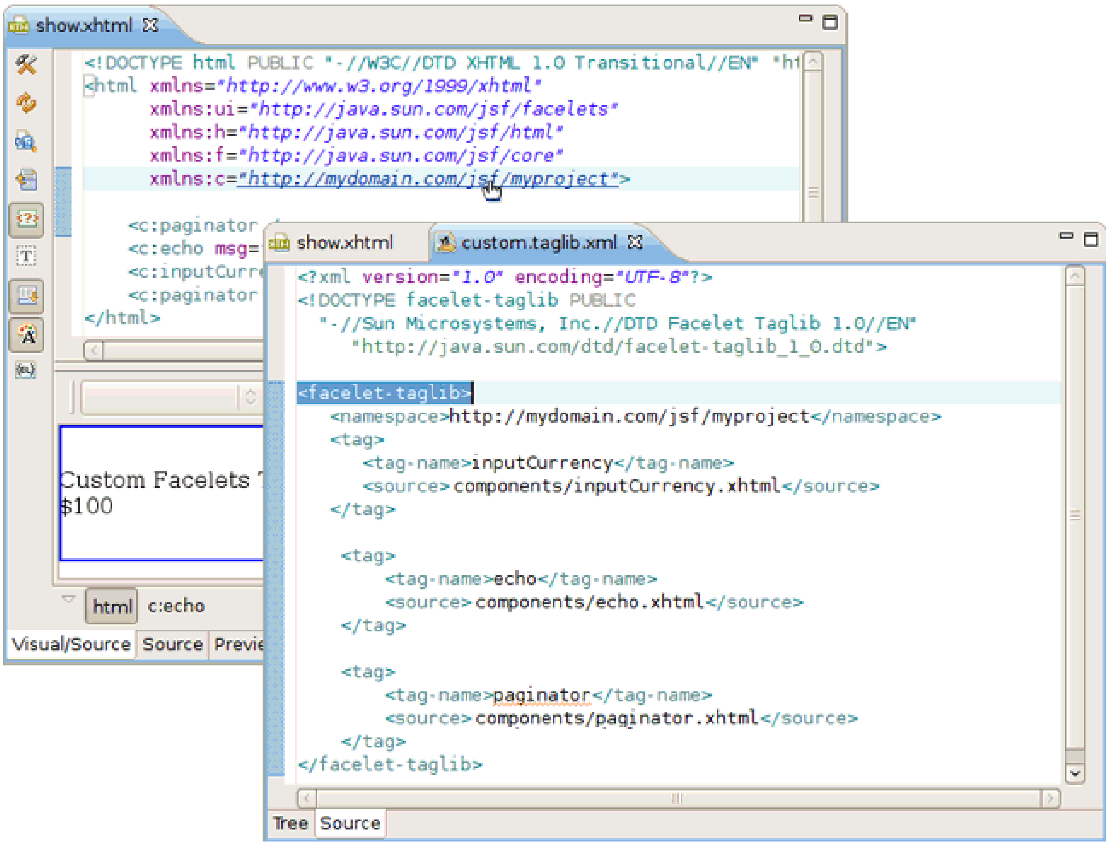
- *OpenOn in XHTML Files That Use Custom Facelets Components*
- *OpenOn in Custom Facelets Tag File (`*.taglib.xml`)*

3.2.6.2.1. OpenOn in XHTML Files That Use Custom Facelets Components

OpenOn functionality in XHTML files is available in two views of the Visual Page Editor:

1. Source view

   Ctrl+Click on the namespace will open the Facelets tag file in a separate window.



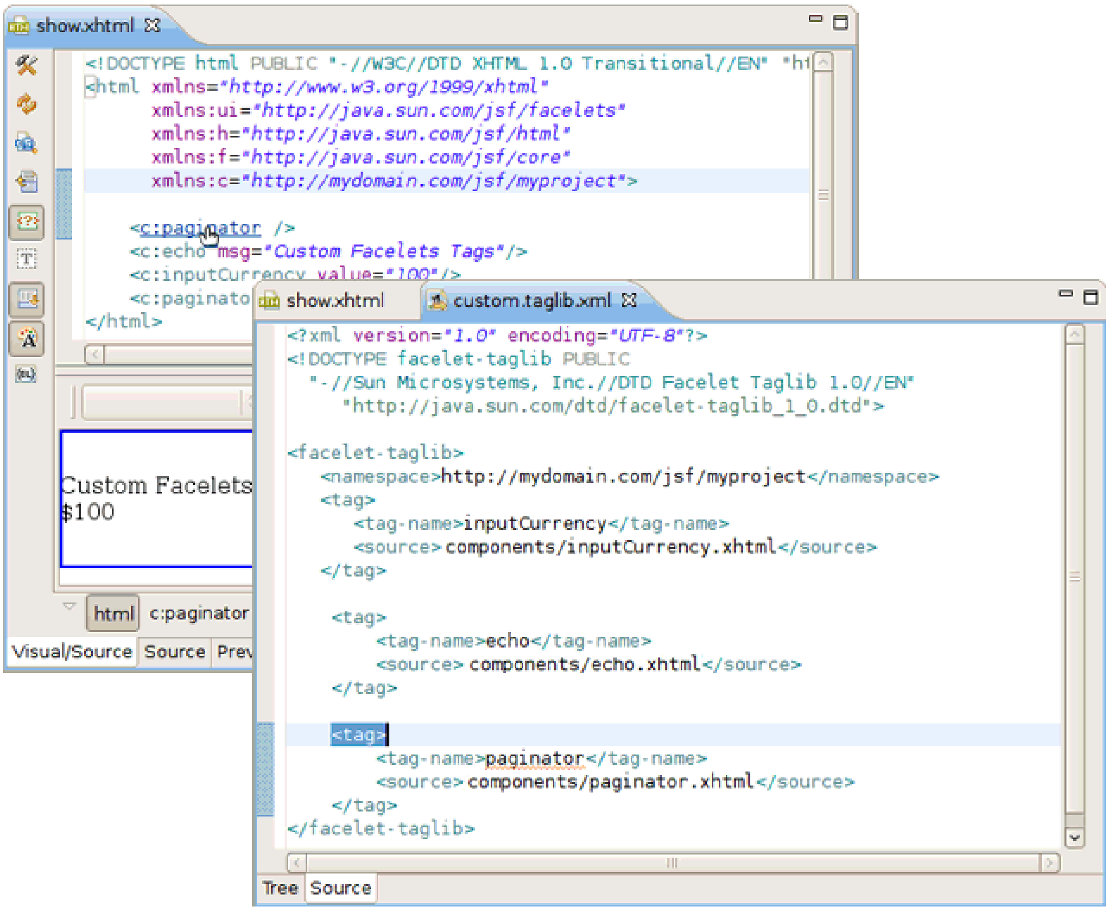
show.xhtml
Custom Facelets
$100
html
c:echo
Visual/Source
Source
Previe
show.xhtml
custom.taglib.xml
Tree
Source

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "ht
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:c="http://mydomain.com/jsf/myproject">

    <c:paginator
    <c:echo msg=
    <c:inputCurr
    <c:paginator
</html>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE facelet-taglib PUBLIC
  "-//Sun Microsystems, Inc.//DTD Facelet Taglib 1.0//EN"
    "http://java.sun.com/dtd/facelet-taglib_1_0.dtd">

<facelet-taglib>
   <namespace>http://mydomain.com/jsf/myproject</namespace>
   <tag>
      <tag-name>inputCurrency</tag-name>
      <source>components/inputCurrency.xhtml</source>
   </tag>

    <tag>
        <tag-name>echo</tag-name>
        <source>components/echo.xhtml</source>
    </tag>

    <tag>
        <tag-name>paginator</tag-name>
        <source>components/paginator.xhtml</source>
    </tag>
</facelet-taglib>
```

Figure 3.78. Opening a Custom Facelets Tag File

Ctrl+Click on any custom Facelets tag declared on the page will do the same. The selected tag will be highlighted in the opened file.



show.xhtml
Custom Facelets
$100
html
c:paginator
Visual/Source
Source
Prev
show.xhtml
custom.taglib.xml
Tree
Source

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "ht
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:c="http://mydomain.com/jsf/myproject">

    <c:paginator />
    <c:echo msg="Custom Facelets Tags"/>
    <c:inputCurrency value="100"/>
    <c:paginato
</html>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE facelet-taglib PUBLIC
  "-//Sun Microsystems, Inc.//DTD Facelet Taglib 1.0//EN"
     "http://java.sun.com/dtd/facelet-taglib_1_0.dtd">

<facelet-taglib>
   <namespace>http://mydomain.com/jsf/myproject</namespace>
   <tag>
      <tag-name>inputCurrency</tag-name>
      <source>components/inputCurrency.xhtml</source>
   </tag>

    <tag>
        <tag-name>echo</tag-name>
        <source> components/echo.xhtml</source>
    </tag>

    <tag>
        <tag-name>paginator</tag-name>
        <source>components/paginator.xhtml</source>
    </tag>
</facelet-taglib>
```

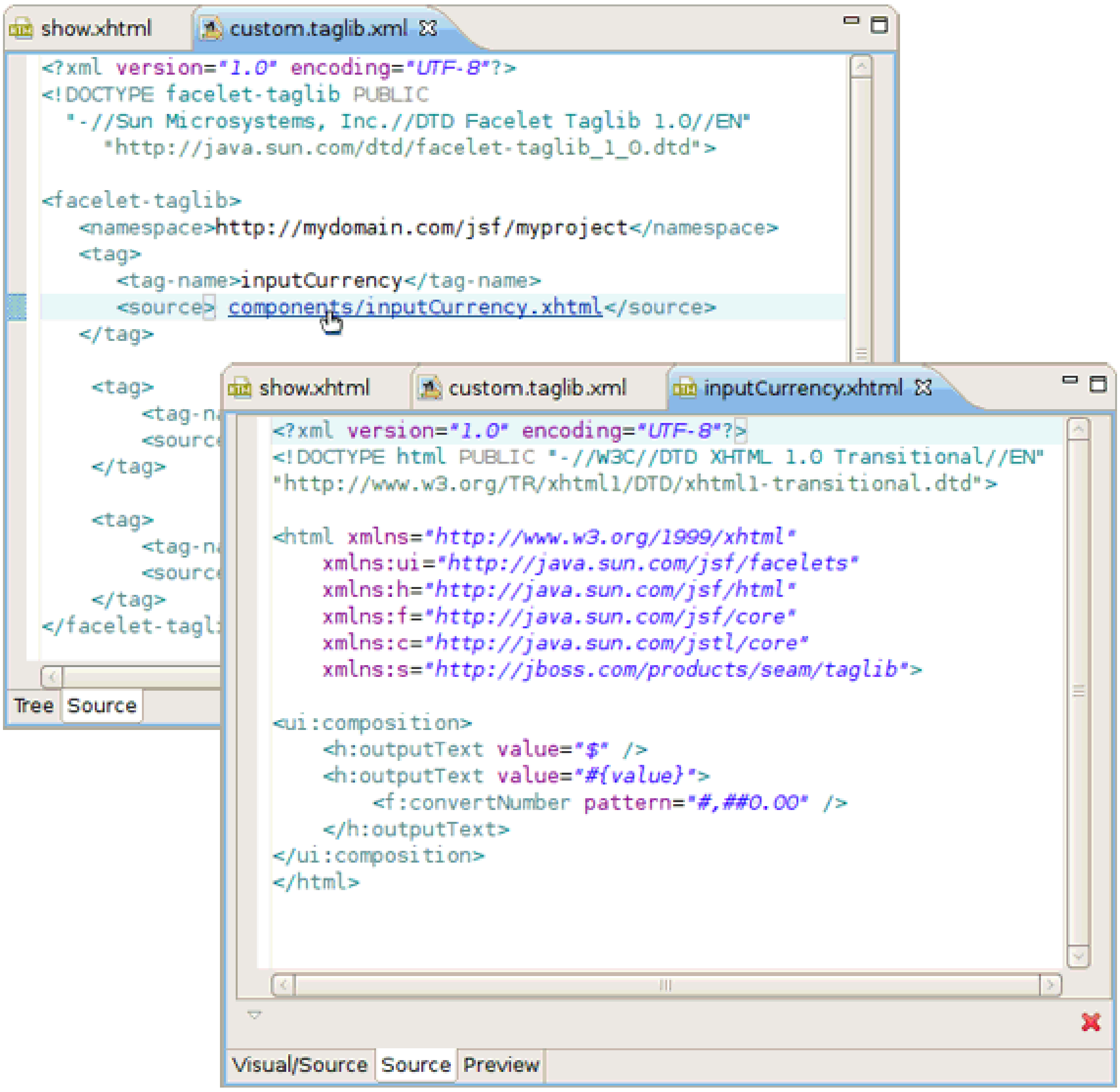
Figure 3.79. Opening a Custom Facelets Tag File

2. Visual view

   In the visual view of the VPE, double-click a custom component and the Facelets tag file (`*.taglib.xml`) where it is declared will be opened.

3.2.6.2.2. OpenOn in Custom Facelets Tag File (`*.taglib.xml`)

Ctrl+Click on the path to source of the Facelets tag will open the component in its own editor.


show.xhtml
custom.taglib.xml
Tree
Source
show.xhtml
custom.taglib.xml
inputCurrency.xhtml
Visual/Source
Source
Preview

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE facelet-taglib PUBLIC
  "-//Sun Microsystems, Inc.//DTD Facelet Taglib 1.0//EN"
     "http://java.sun.com/dtd/facelet-taglib_1_0.dtd">

<facelet-taglib>
   <namespace>http://mydomain.com/jsf/myproject</namespace>
   <tag>
      <tag-name>inputCurrency</tag-name>
      <source> components/inputCurrency.xhtml</source>
   </tag>

    <tag>
        <tag-n
        <sourc
    </tag>

    <tag>
        <tag-n
        <sourc
    </tag>
</facelet-tagl
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:c="http://java.sun.com/jstl/core"
    xmlns:s="http://jboss.com/products/seam/taglib">

<ui:composition>
    <h:outputText value="$" />
    <h:outputText value="#{value}">
        <f:convertNumber pattern="#,##0.00" />
    </h:outputText>
</ui:composition>
</html>
```

Figure 3.80. Opening a Custom Facelets Component

3.2.7. Setup notes for Linux

Linux users who are going to use earlier then JBoss Tools 3.1.0.M4 versions 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 compat-libstdc++-33.i386 package.

Note

Starting from JBoss Tools 3.1.0.M4 the libstdc++.so.5 library isn't required.

- 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.

If it doesn't help and you use Fedora Core Linux and Eclipse Version: 3.4.1,the issue can be produced because libswt-xulrunner-gtk-3449.so file doesn't present in eclipse-swt-3.4.1-5.fc10.x86_64.rpm/eclipse/plugins/org.eclipse.swt.gtk.linux.x86_64_3.4.1.v3449c.jar.To add this file to eclipse you should:

- Decompress eclipse/plugins/org.eclipse.swt.gtk.linux.x86_3.4.1.v3449c.jar form eclipse-SDK-3.4.1-linux-gtk-x86_64.tar.gz
- Copy libswt-xulrunner-gtk-3449.so file to your Fedora Eclipse location.
- Open the file eclipse.ini,which can be found in your Fedora Eclipse location and add the following line:

```
-Dswt.library.path=/usr/lib/eclipse
```

,where `/usr/lib/eclipse` is the path to your eclipse folder.

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.

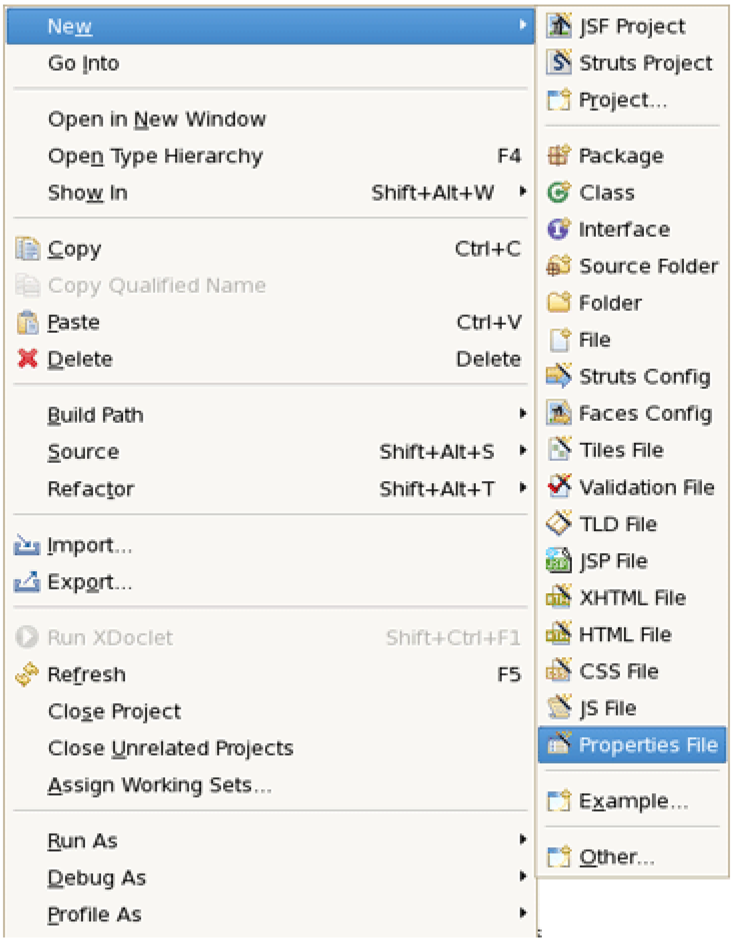
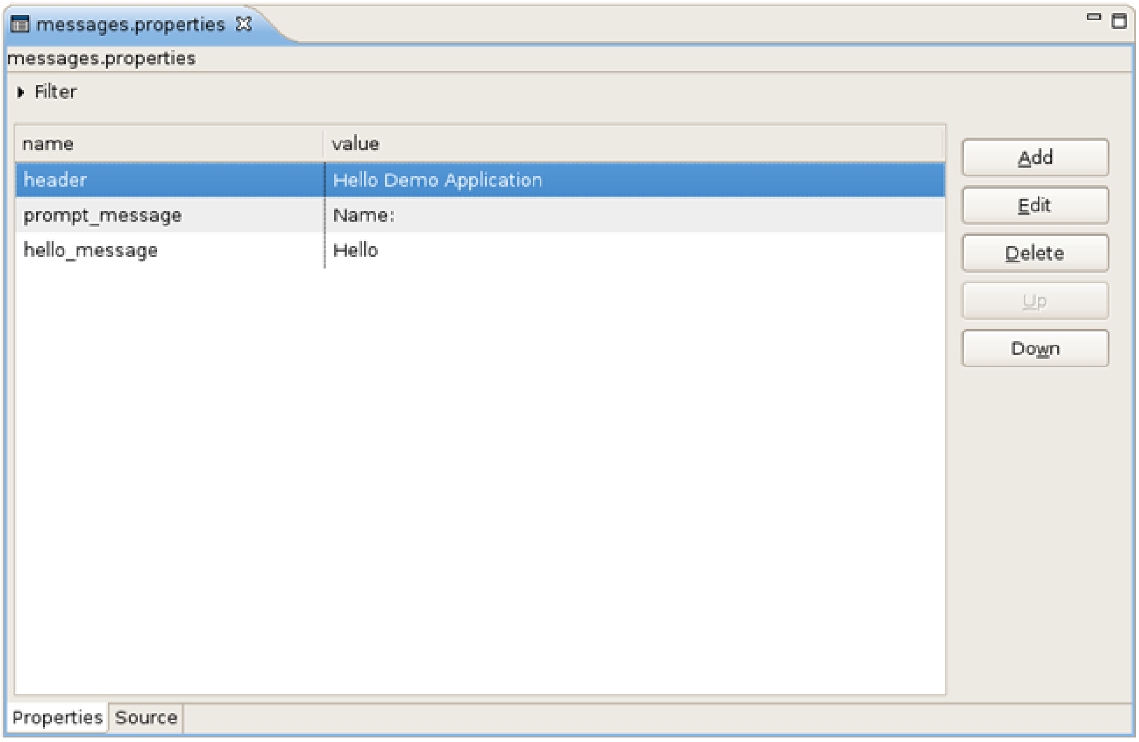

New
Go Into
Open in New Window
Open Type Hierarchy F4
Show In Shift+Alt+W
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Build Path
Source Shift+Alt+S
Refactor Shift+Alt+T
Import...
Export...
Run XDoclet Shift+Ctrl+F1
Refresh F5
Close Project
Close Unrelated Projects
Assign Working Sets...
Run As
Debug As
Profile As
JSF Project
Struts Project
Project...
Package
Class
Interface
Source Folder
Folder
File
Struts Config
Faces Config
Tiles File
Validation File
TLD File
JSP File
XHTML File
HTML File
CSS File
JS File
Properties File
Example...
Other...

Figure 3.81. Selecting Properties File

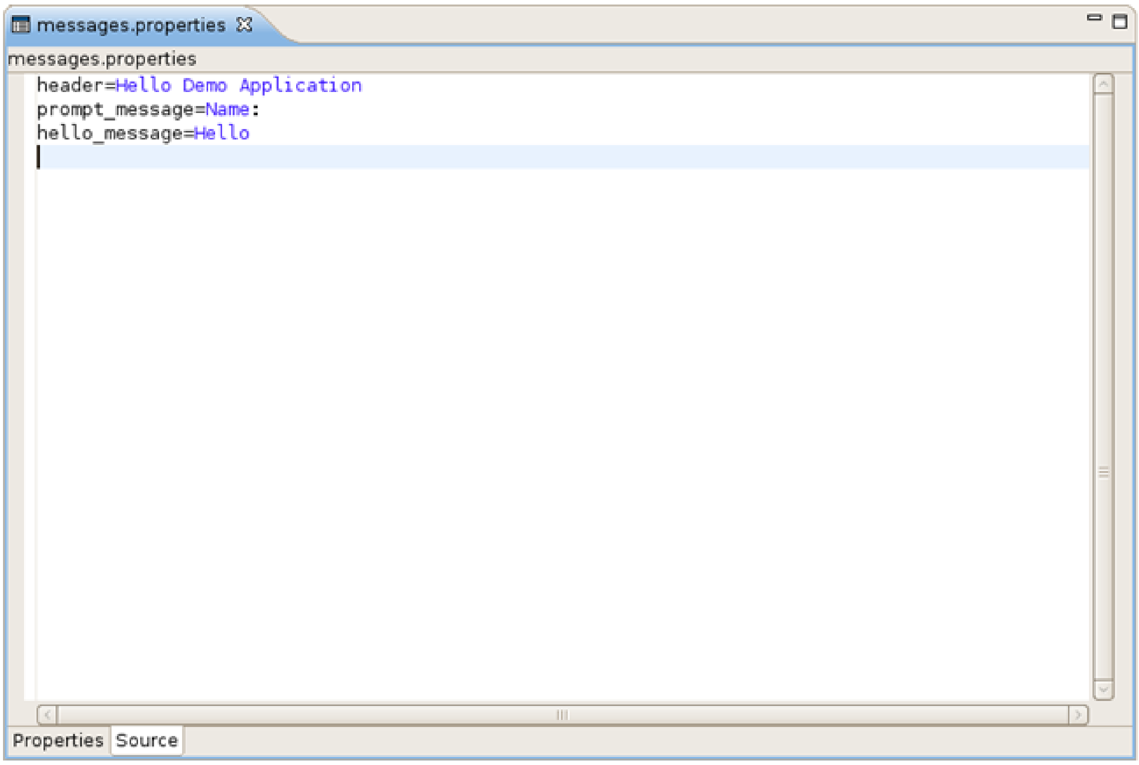
You can edit the file using a table-oriented "Properties" viewer:


messages.properties
messages.properties
Filter
Add
Edit
Delete
Up
Down
Properties
Source

name	value
header	Hello Demo Application
prompt_message	Name:
hello_message	Hello

Figure 3.82. "Properties" Viewer

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


messages.properties
messages.properties
Properties
Source

```
header=Hello Demo Application
prompt_message=Name:
hello_message=Hello
```

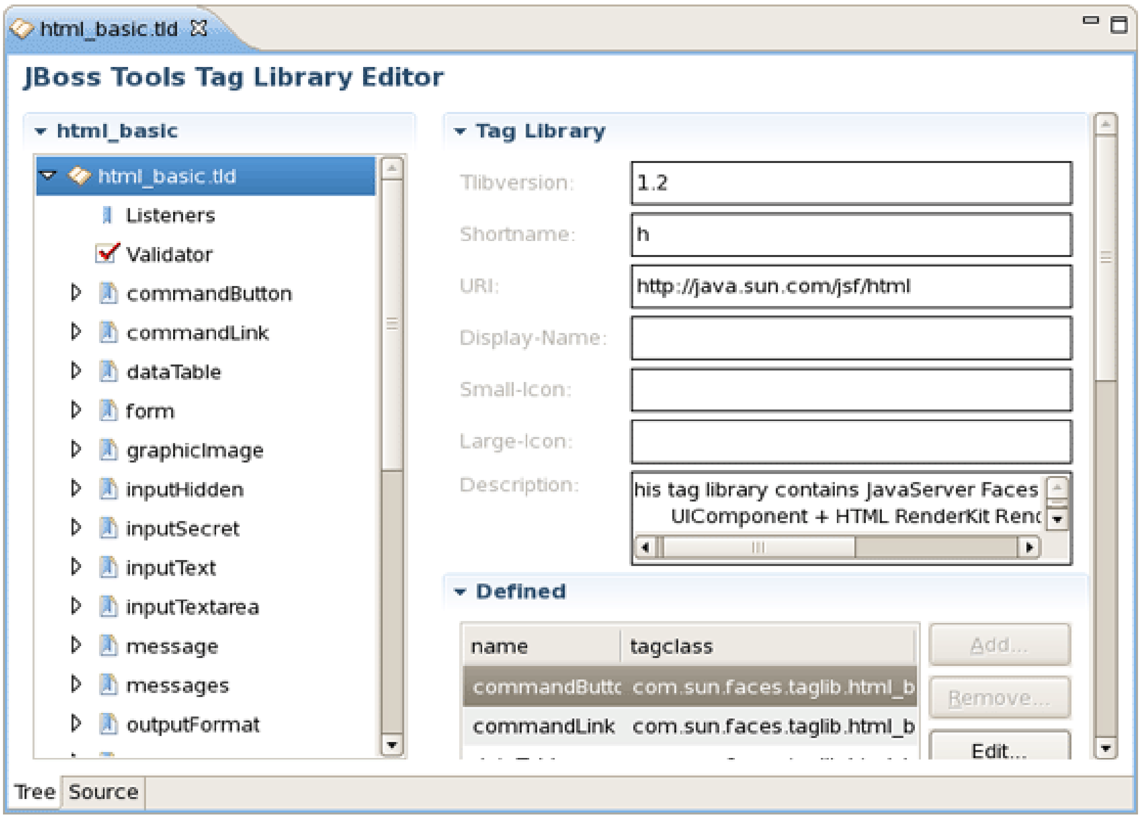
Figure 3.83. Source Viewer

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


html_basic.tld
JBoss Tools Tag Library Editor
html_basic
html_basic.tld
Listeners
Validator
commandButton
commandLink
dataTable
form
graphicImage
inputHidden
inputSecret
inputText
inputTextarea
message
messages
outputFormat
Tag Library
Defined
Add...
Remove...
Edit...
Tree
Source

Tlibversion: 1.2

Shortname: h

URI: http://java.sun.com/jsf/html

Display-Name:

Small-Icon:

Large-Icon:

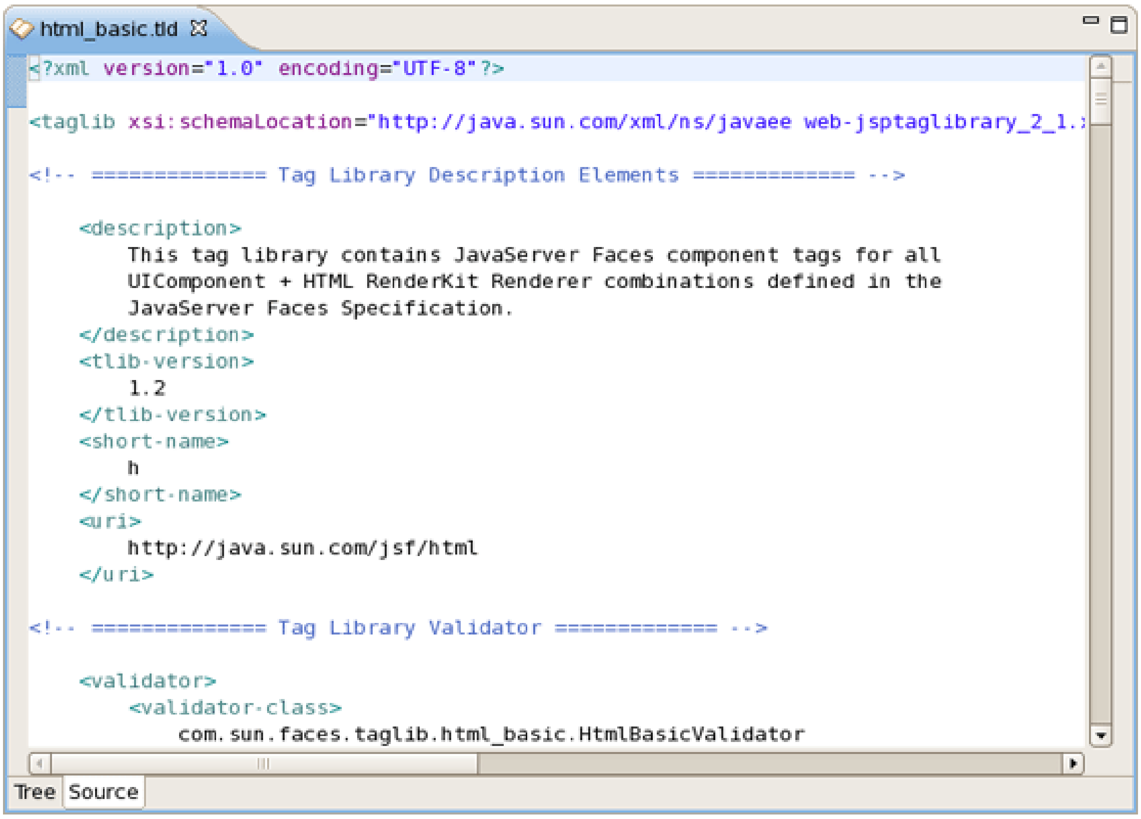
Description: his tag library contains JavaServer Faces UIComponent + HTML RenderKit Renc

name	tagclass
commandButtc	com.sun.faces.taglib.html_b
commandLink	com.sun.faces.taglib.html_b

Figure 3.84. Tree View

3.3.2.2. Source view

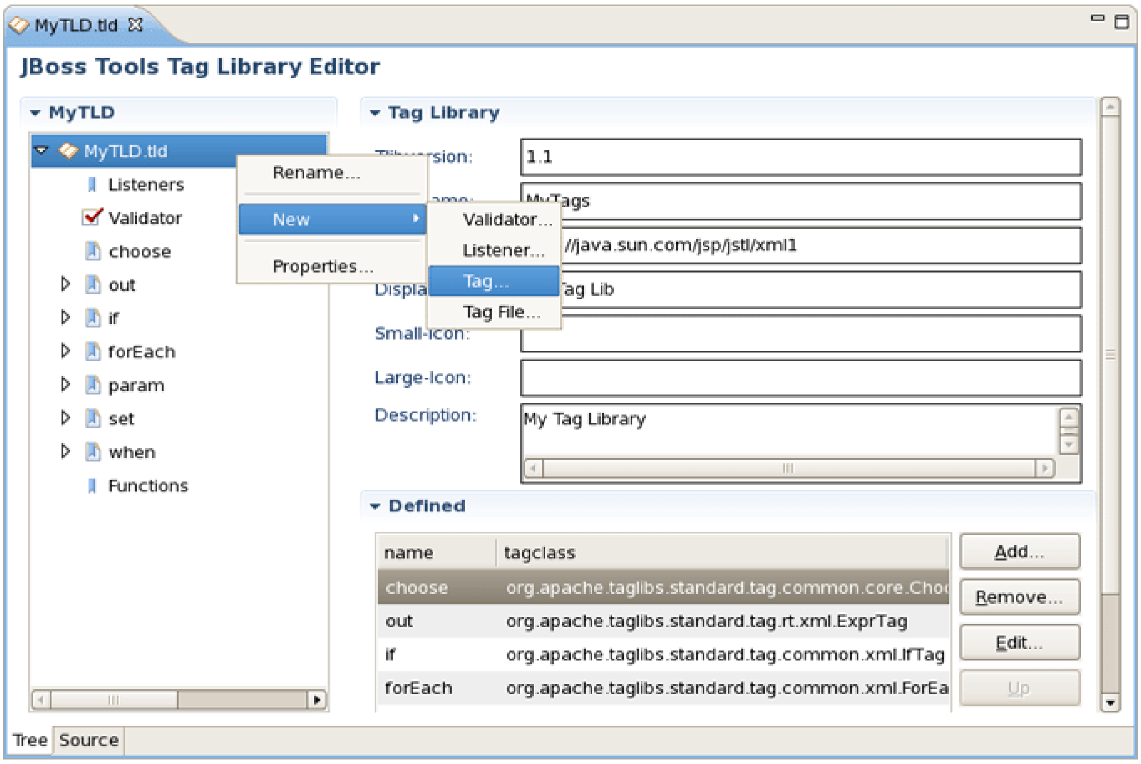
You can easily switch from Tree to Source by selecting the Source tab at the bottom of the editor.



```
<?xml version="1.0" encoding="UTF-8"?>

<taglib xsi:schemaLocation="http://java.sun.com/xml/ns/javaee web-jsptaglibrary_2_1.

<!-- ============== Tag Library Description Elements ============= -->

    <description>
        This tag library contains JavaServer Faces component tags for all
        UIComponent + HTML RenderKit Renderer combinations defined in the
        JavaServer Faces Specification.
    </description>
    <tlib-version>
        1.2
    </tlib-version>
    <short-name>
        h
    </short-name>
    <uri>
        http://java.sun.com/jsf/html
    </uri>

<!-- ============== Tag Library Validator ============= -->

    <validator>
        <validator-class>
            com.sun.faces.taglib.html_basic.HtmlBasicValidator
```

Figure 3.85. Source View

You can easily add a new tag:

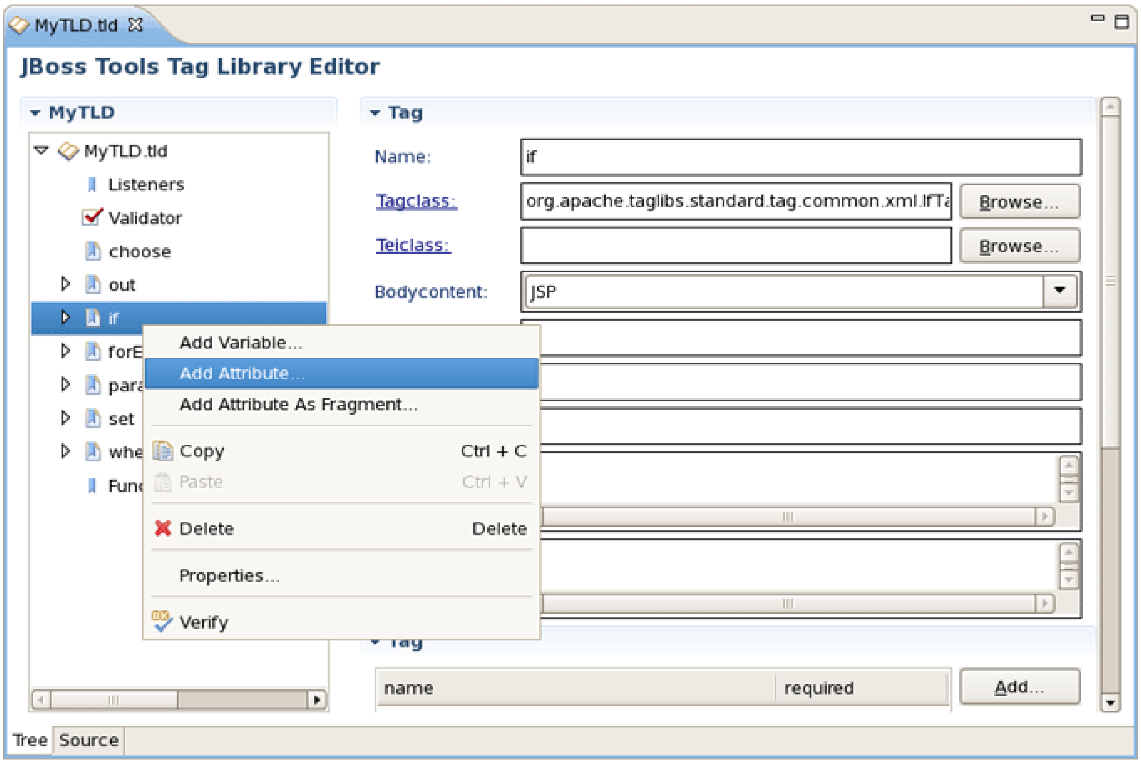

MyTLD.tld
JBoss Tools Tag Library Editor
MyTLD
MyTLD.tld
Listeners
Validator
choose
out
if
forEach
param
set
when
Functions
Rename...
New
Properties...
Validator...
Listener...
Tag...
Tag File...
Tag Library
Defined
Add...
Remove...
Edit...
Up
Tree
Source

Tlib-version: 1.1
MyTags
//java.sun.com/jsp/jstl/xml1
Display-name: ...ag Lib
Small-Icon:
Large-Icon:
Description: My Tag Library

name	tagclass
choose	org.apache.taglibs.standard.tag.common.core.Cho
out	org.apache.taglibs.standard.tag.rt.xml.ExprTag
if	org.apache.taglibs.standard.tag.common.xml.IfTag
forEach	org.apache.taglibs.standard.tag.common.xml.ForEa

Figure 3.86. Adding a New TLD Tag

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

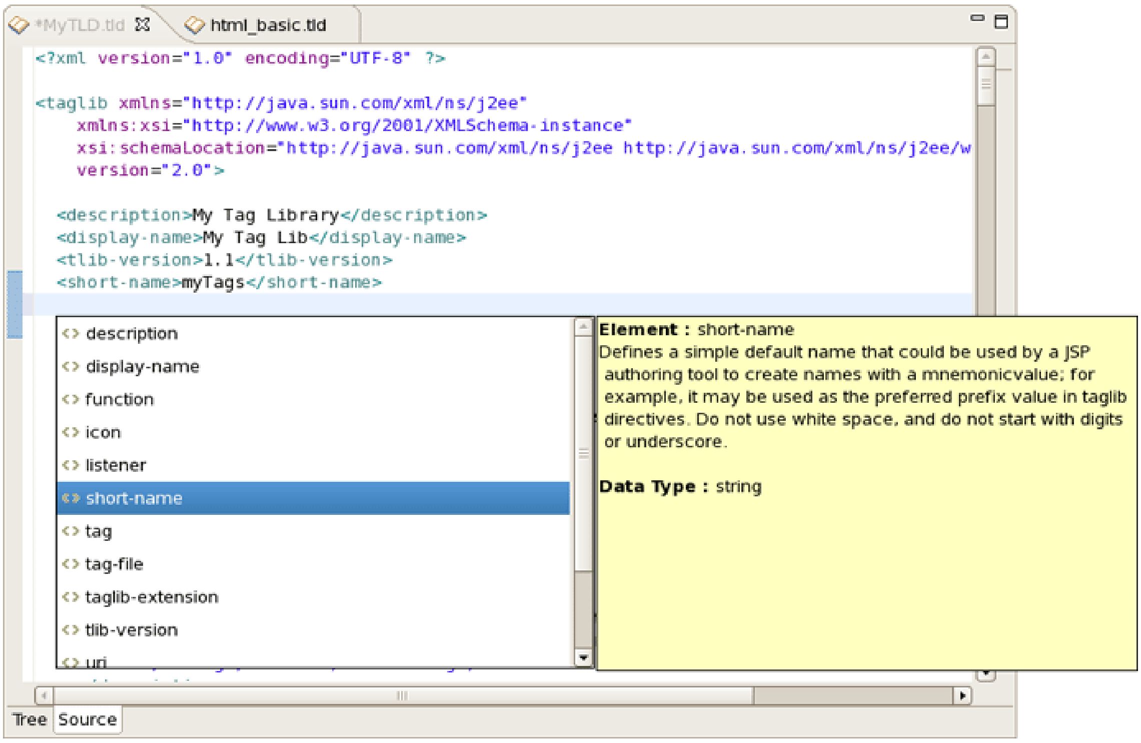

MyTLD.tld
JBoss Tools Tag Library Editor
MyTLD
MyTLD.tld
Listeners
Validator
choose
out
if
forE
para
set
whe
Func
Add Variable...
Add Attribute...
Add Attribute As Fragment...
Copy Ctrl + C
Paste Ctrl + V
Delete Delete
Properties...
Verify
Tag
Browse...
Browse...
Tag
Add...
Tree
Source

Name: if
Tagclass: org.apache.taglibs.standard.tag.common.xml.IfTa
Teiclass:
Bodycontent: JSP

name	required

Figure 3.87. Adding a New Attribute to TLD tag

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


*MyTLD.tld
html_basic.tld
Tree
Source

```
<?xml version="1.0" encoding="UTF-8" ?>

<taglib xmlns="http://java.sun.com/xml/ns/j2ee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/w
    version="2.0">

  <description>My Tag Library</description>
  <display-name>My Tag Lib</display-name>
  <tlib-version>1.1</tlib-version>
  <short-name>myTags</short-name>
```

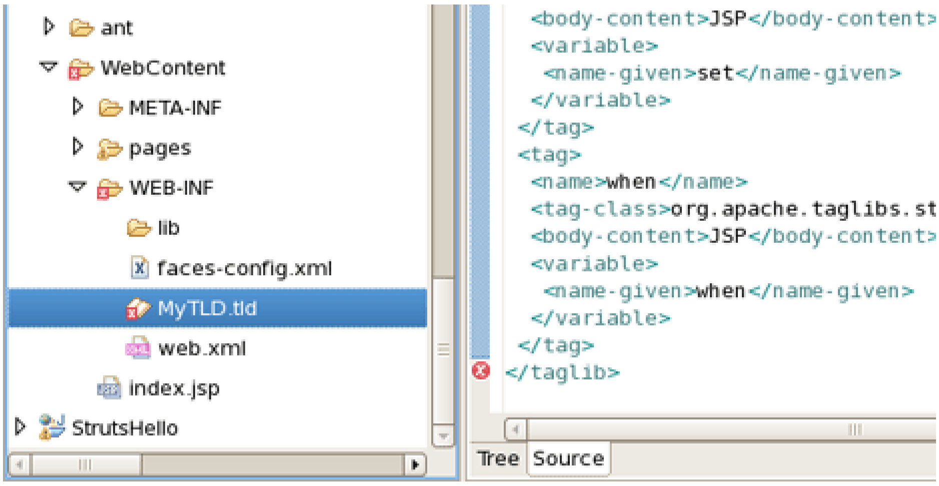
description
display-name
function
icon
listener
short-name
tag
tag-file
taglib-extension
tlib-version
uri

Element : short-name
Defines a simple default name that could be used by a JSP authoring tool to create names with a mnemonicvalue; for example, it may be used as the preferred prefix value in taglib directives. Do not use white space, and do not start with digits or underscore.

Data Type : string

Figure 3.88. TLD 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.


ant
WebContent
META-INF
pages
WEB-INF
lib
faces-config.xml
MyTLD.tld
web.xml
index.jsp
StrutsHello
Tree
Source

```
    <body-content>JSP</body-content>
    <variable>
      <name-given>set</name-given>
    </variable>
  </tag>
  <tag>
    <name>when</name>
    <tag-class>org.apache.taglibs.st
    <body-content>JSP</body-content>
    <variable>
      <name-given>when</name-given>
    </variable>
  </tag>
</taglib>
```

Figure 3.89. Error Reporting

3.3.3. Graphical Web Application File (web.xml) Editor

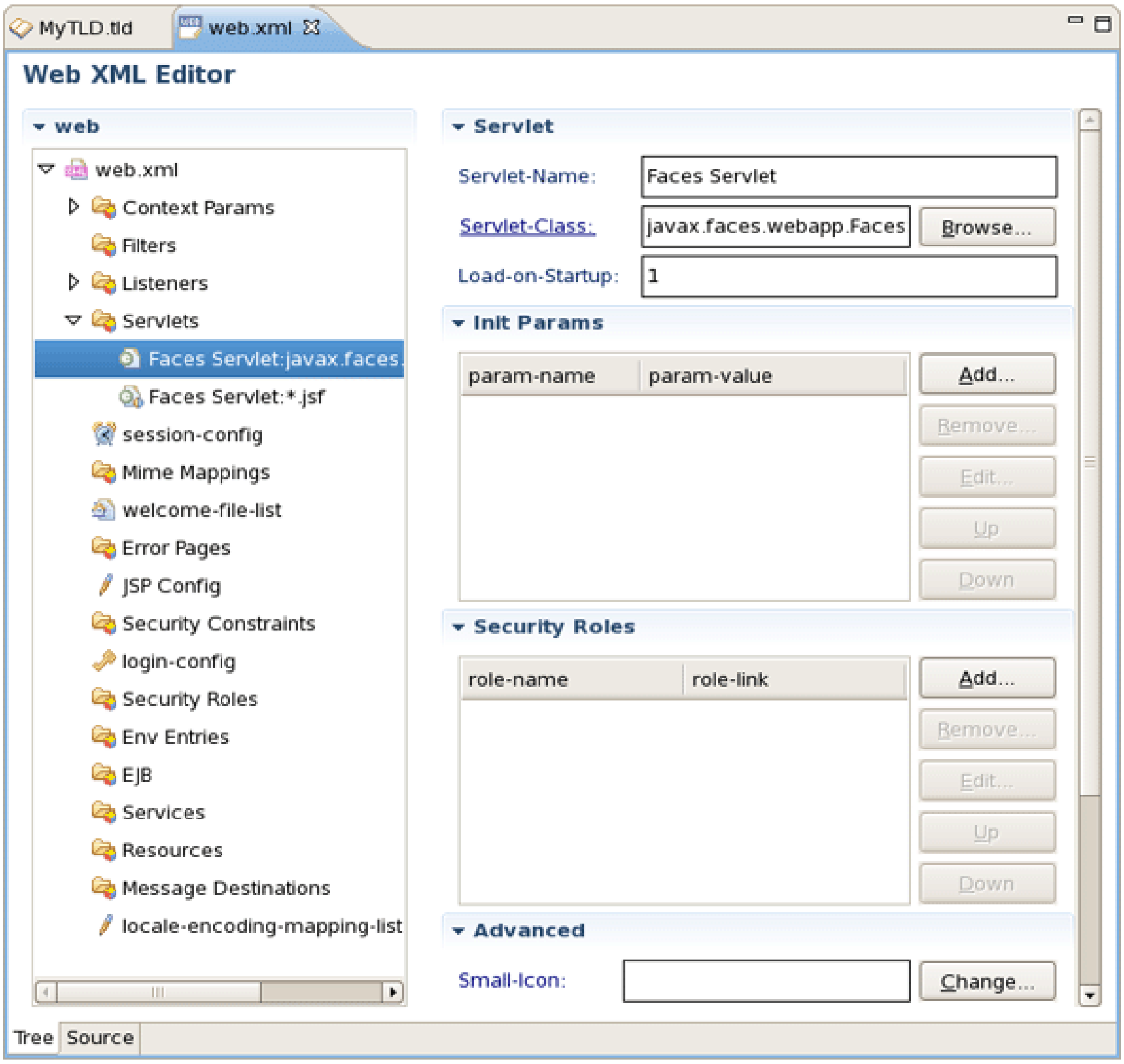
The deployment descriptor *web.xml* file is intended for describing the servlets, container-managed security constraints and various deployment properties specific for your Web Application.

To edit the deployment descriptor JBoss Developer Studio provides its own web.xml editor that comes with the same features you will find in all other JBDS editors:

- Graphical and source edit modes
- Validation and error checking

3.3.3.1. Tree View

Switch to the Tree view if you want to edit *web.xml* in a graphical mode. All elements that *web.xml* could include are located in the left area of the editor in a tree format. Click a node on the left to display and edit its properties that will appear in the right-hand area.


MyTLD.tld
web.xml
Web XML Editor
web
web.xml
Context Params
Filters
Listeners
Servlets
Faces Servlet:javax.faces.
Faces Servlet:*.jsf
session-config
Mime Mappings
welcome-file-list
Error Pages
JSP Config
Security Constraints
login-config
Security Roles
Env Entries
EJB
Services
Resources
Message Destinations
locale-encoding-mapping-list
Servlet
Browse...
Init Params
param-name
param-value
Add...
Remove...
Edit...
Up
Down
Security Roles
role-name
role-link
Add...
Remove...
Edit...
Up
Down
Advanced
Small-Icon:
Change...
Tree
Source

Servlet-Name: Faces Servlet

Servlet-Class: javax.faces.webapp.Faces

Load-on-Startup: 1

Figure 3.90. Tree View for editing web.xml in a graphical mode

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

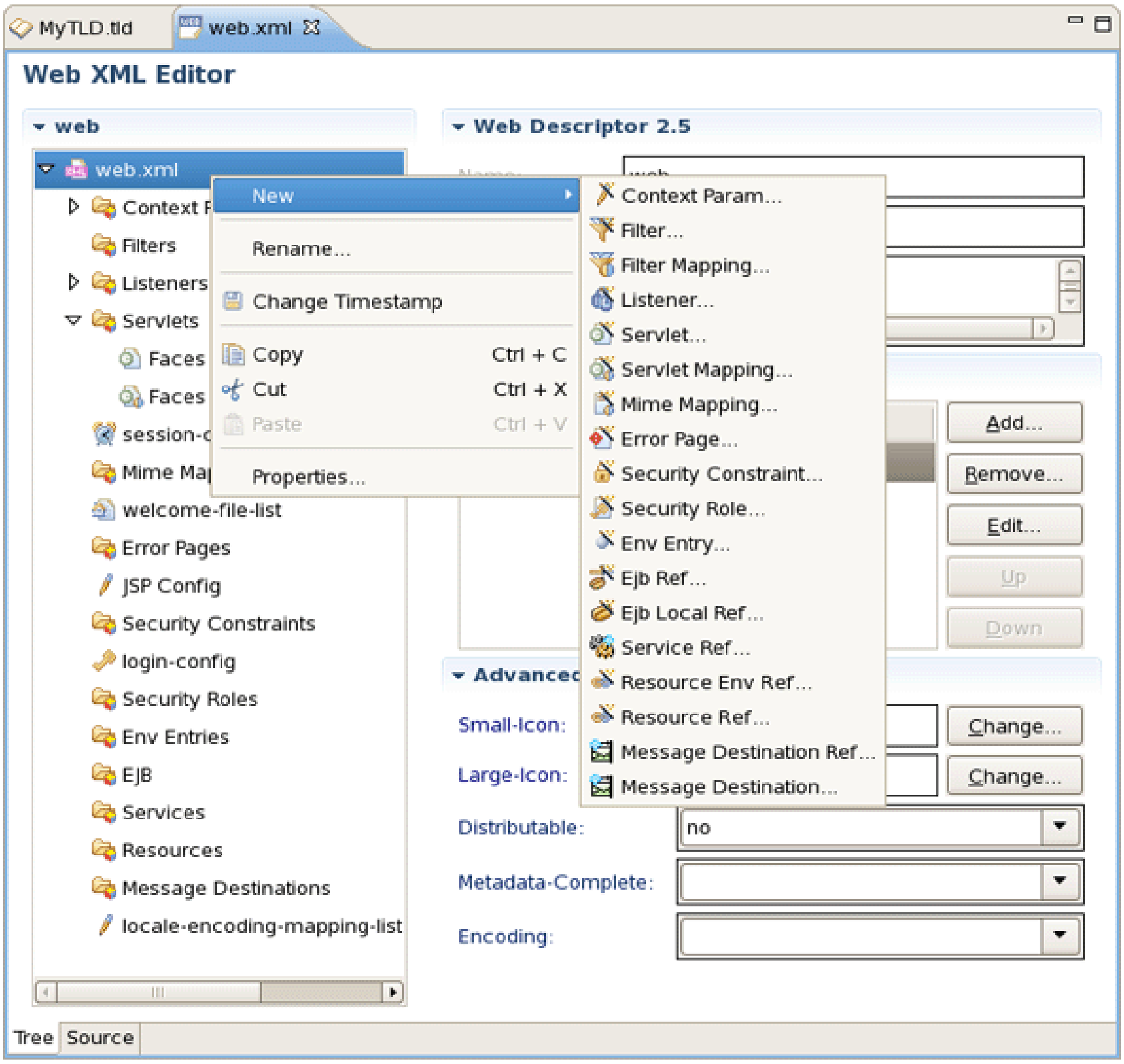
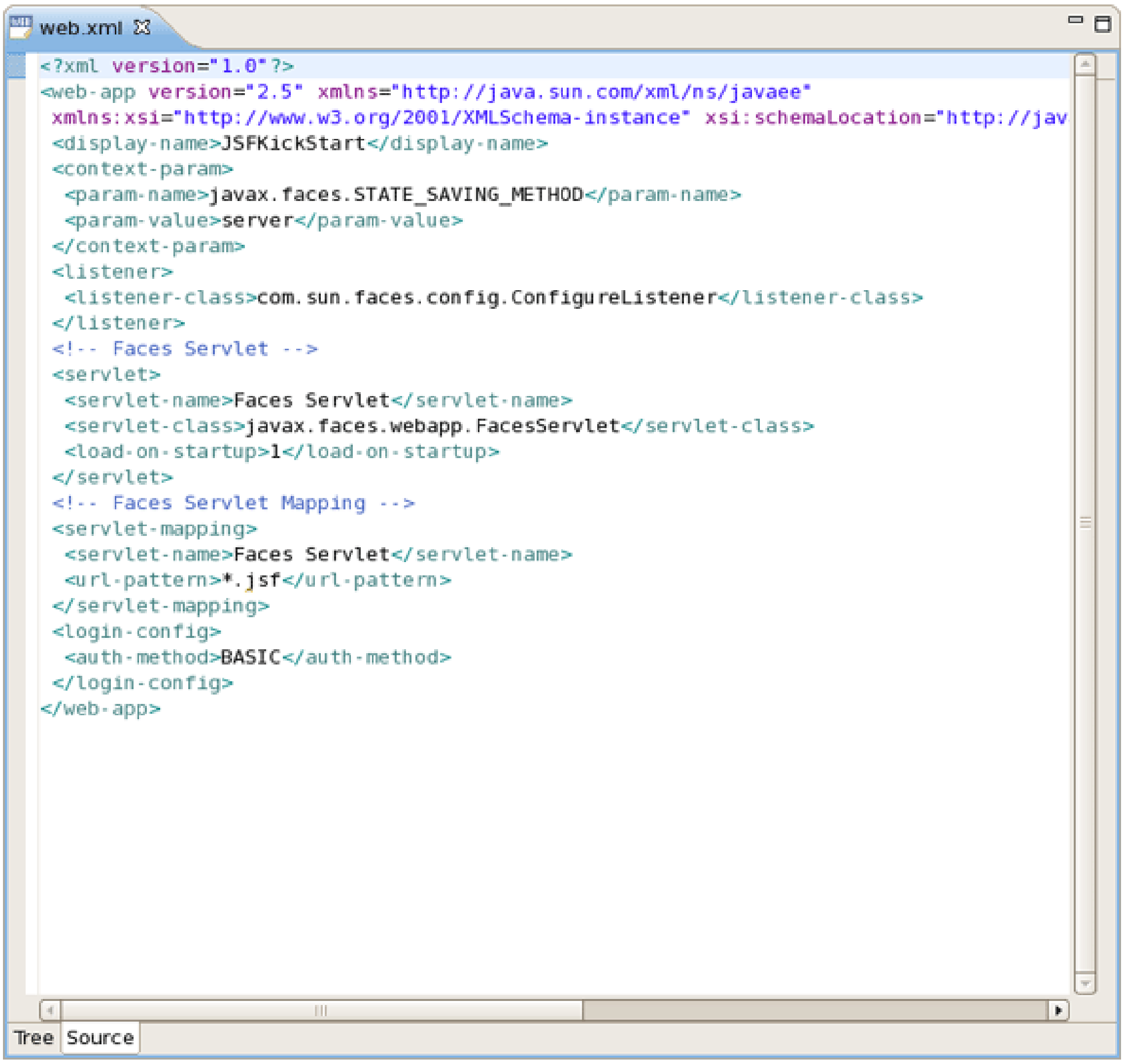

MyTLD.tld
web.xml
Web XML Editor
web
Web Descriptor 2.5
web.xml
Context P
Filters
Listeners
Servlets
Faces
Faces
session-c
Mime Ma
welcome-file-list
Error Pages
JSP Config
Security Constraints
login-config
Security Roles
Env Entries
EJB
Services
Resources
Message Destinations
locale-encoding-mapping-list
New
Rename...
Change Timestamp
Copy Ctrl + C
Cut Ctrl + X
Paste Ctrl + V
Properties...
Context Param...
Filter...
Filter Mapping...
Listener...
Servlet...
Servlet Mapping...
Mime Mapping...
Error Page...
Security Constraint...
Security Role...
Env Entry...
Ejb Ref...
Ejb Local Ref...
Service Ref...
Resource Env Ref...
Resource Ref...
Message Destination Ref...
Message Destination...
Add...
Remove...
Edit...
Up
Down
Advanced
Small-Icon:
Large-Icon:
Change...
Change...
Distributable:
no
Metadata-Complete:
Encoding:
Tree
Source

Figure 3.91. Adding New Elements in Web XML Editor

3.3.3.2. Source View

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

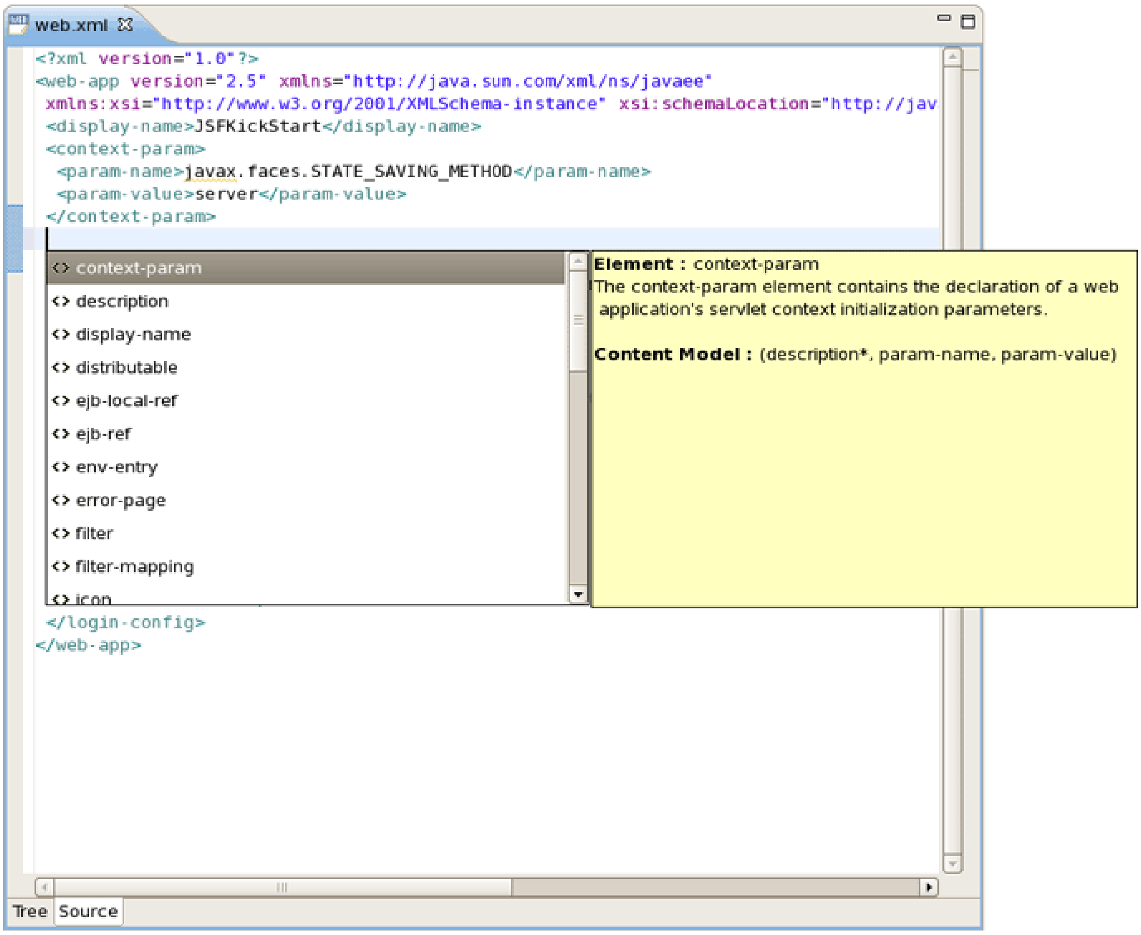


```
<?xml version="1.0"?>
<web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://jav
 <display-name>JSFKickStart</display-name>
 <context-param>
  <param-name>javax.faces.STATE_SAVING_METHOD</param-name>
  <param-value>server</param-value>
 </context-param>
 <listener>
  <listener-class>com.sun.faces.config.ConfigureListener</listener-class>
 </listener>
 <!-- Faces Servlet -->
 <servlet>
  <servlet-name>Faces Servlet</servlet-name>
  <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
  <load-on-startup>1</load-on-startup>
 </servlet>
 <!-- Faces Servlet Mapping -->
 <servlet-mapping>
  <servlet-name>Faces Servlet</servlet-name>
  <url-pattern>*.jsf</url-pattern>
 </servlet-mapping>
 <login-config>
  <auth-method>BASIC</auth-method>
 </login-config>
</web-app>
```

Figure 3.92. Web XML Source View

3.3.3.3. Content Assist

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



web.xml
context-param
description
display-name
distributable
ejb-local-ref
ejb-ref
env-entry
error-page
filter
filter-mapping
icon
Tree
Source

```
<?xml version="1.0"?>
<web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://jav
 <display-name>JSFKickStart</display-name>
 <context-param>
  <param-name>javax.faces.STATE_SAVING_METHOD</param-name>
  <param-value>server</param-value>
 </context-param>

 </login-config>
</web-app>
```

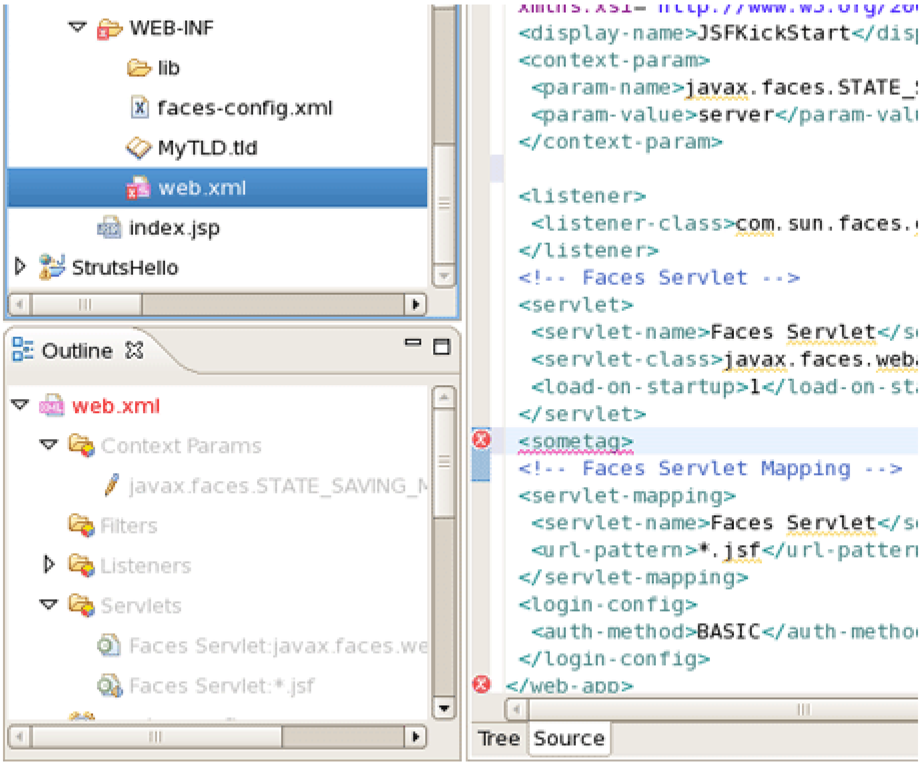
Element : context-param
The context-param element contains the declaration of a web application's servlet context initialization parameters.

Content Model : (description*, param-name, param-value)

Figure 3.93. Web XML 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.


WEB-INF
lib
faces-config.xml
MyTLD.tld
web.xml
index.jsp
StrutsHello
Outline
web.xml
Context Params
javax.faces.STATE_SAVING_M
Filters
Listeners
Servlets
Faces Servlet:javax.faces.we
Faces Servlet:*.jsf
Tree
Source

```
<display-name>JSFKickStart</dis
<context-param>
 <param-name>javax.faces.STATE_
 <param-value>server</param-val
</context-param>

<listener>
 <listener-class>com.sun.faces.
</listener>
<!-- Faces Servlet -->
<servlet>
 <servlet-name>Faces Servlet</s
 <servlet-class>javax.faces.web
 <load-on-startup>1</load-on-st
</servlet>
<sometag>
<!-- Faces Servlet Mapping -->
<servlet-mapping>
 <servlet-name>Faces Servlet</s
 <url-pattern>*.jsf</url-patter
</servlet-mapping>
<login-config>
 <auth-method>BASIC</auth-metho
</login-config>
</web-app>
```

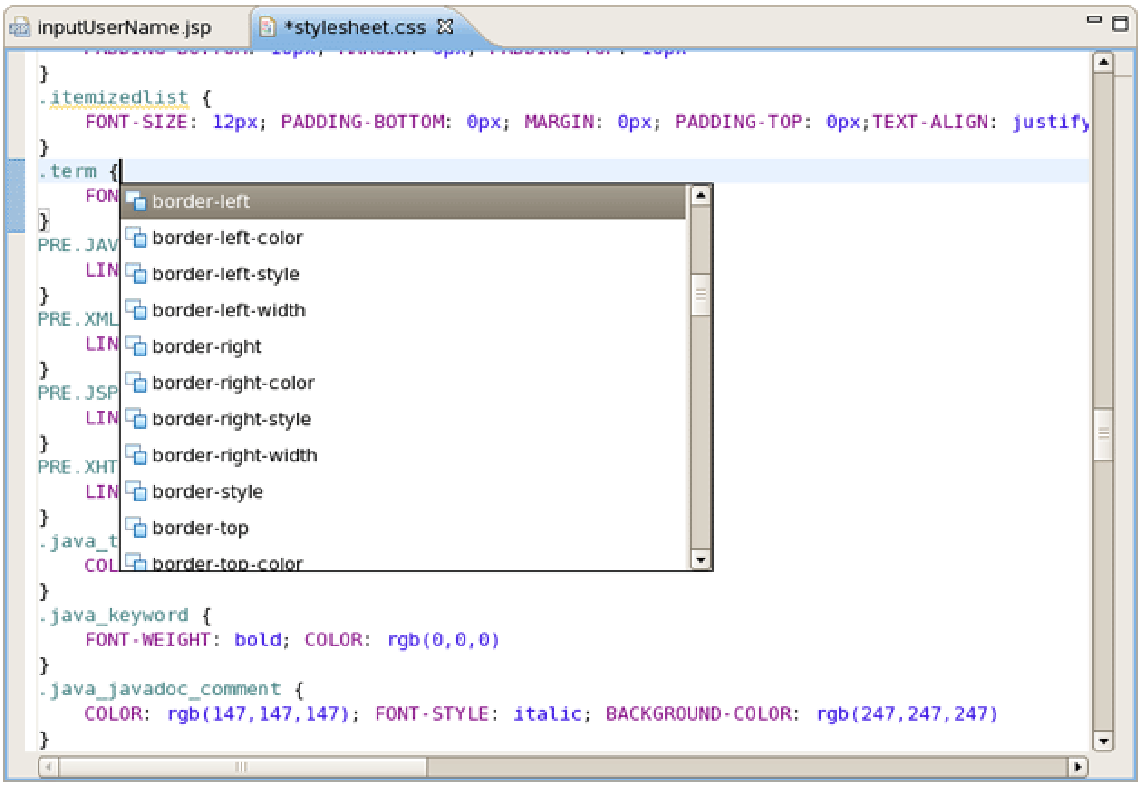
Figure 3.94. Errors Reporting

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

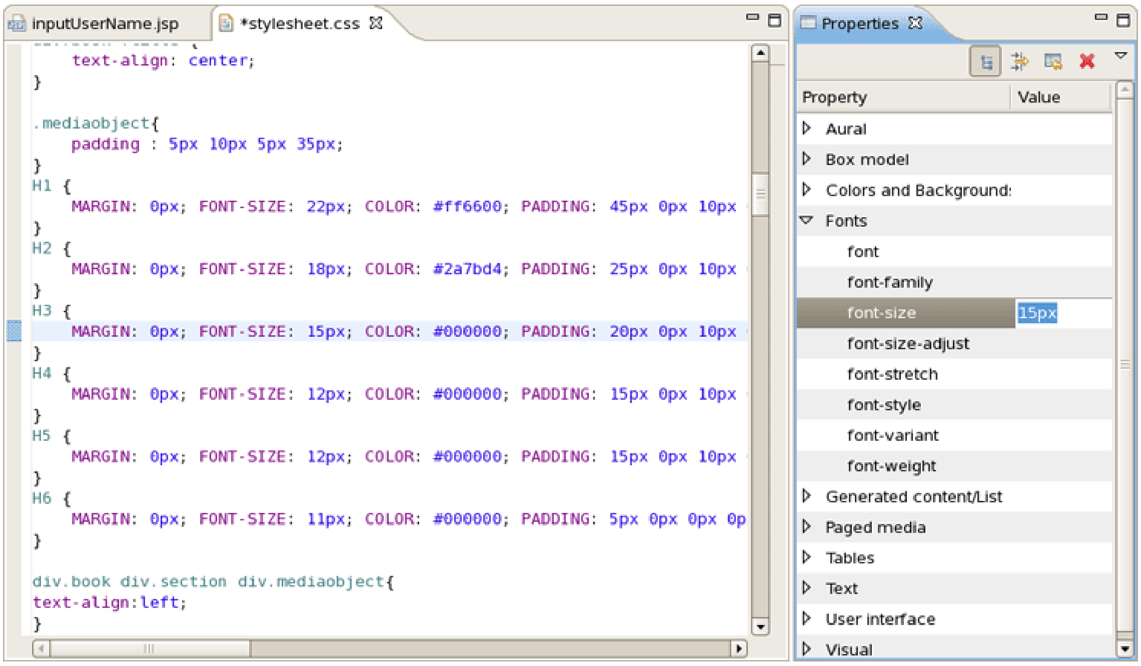
With the CSS (Cascading Style Sheet) editor, you can take advantage of code prompting:


inputUserName.jsp
*stylesheet.css
border-left
border-left-color
border-left-style
border-left-width
border-right
border-right-color
border-right-style
border-right-width
border-style
border-top
border-top-color

```
}
.itemizedlist {
    FONT-SIZE: 12px; PADDING-BOTTOM: 0px; MARGIN: 0px; PADDING-TOP: 0px;TEXT-ALIGN: justify
}
.term {
    FON
}
PRE.JAV
    LIN
}
PRE.XML
    LIN
}
PRE.JSP
    LIN
}
PRE.XHT
    LIN
}
.java_t
    COL
}
.java_keyword {
    FONT-WEIGHT: bold; COLOR: rgb(0,0,0)
}
.java_javadoc_comment {
    COLOR: rgb(147,147,147); FONT-STYLE: italic; BACKGROUND-COLOR: rgb(247,247,247)
}
```

Figure 3.95. CSS Editor

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


inputUserName.jsp
*stylesheet.css
Properties

```
    text-align: center;
}

.mediaobject{
    padding : 5px 10px 5px 35px;
}
H1 {
    MARGIN: 0px; FONT-SIZE: 22px; COLOR: #ff6600; PADDING: 45px 0px 10px
}
H2 {
    MARGIN: 0px; FONT-SIZE: 18px; COLOR: #2a7bd4; PADDING: 25px 0px 10px
}
H3 {
    MARGIN: 0px; FONT-SIZE: 15px; COLOR: #000000; PADDING: 20px 0px 10px
}
H4 {
    MARGIN: 0px; FONT-SIZE: 12px; COLOR: #000000; PADDING: 15px 0px 10px
}
H5 {
    MARGIN: 0px; FONT-SIZE: 12px; COLOR: #000000; PADDING: 15px 0px 10px
}
H6 {
    MARGIN: 0px; FONT-SIZE: 11px; COLOR: #000000; PADDING: 5px 0px 0px 0p
}

div.book div.section div.mediaobject{
text-align:left;
}
```

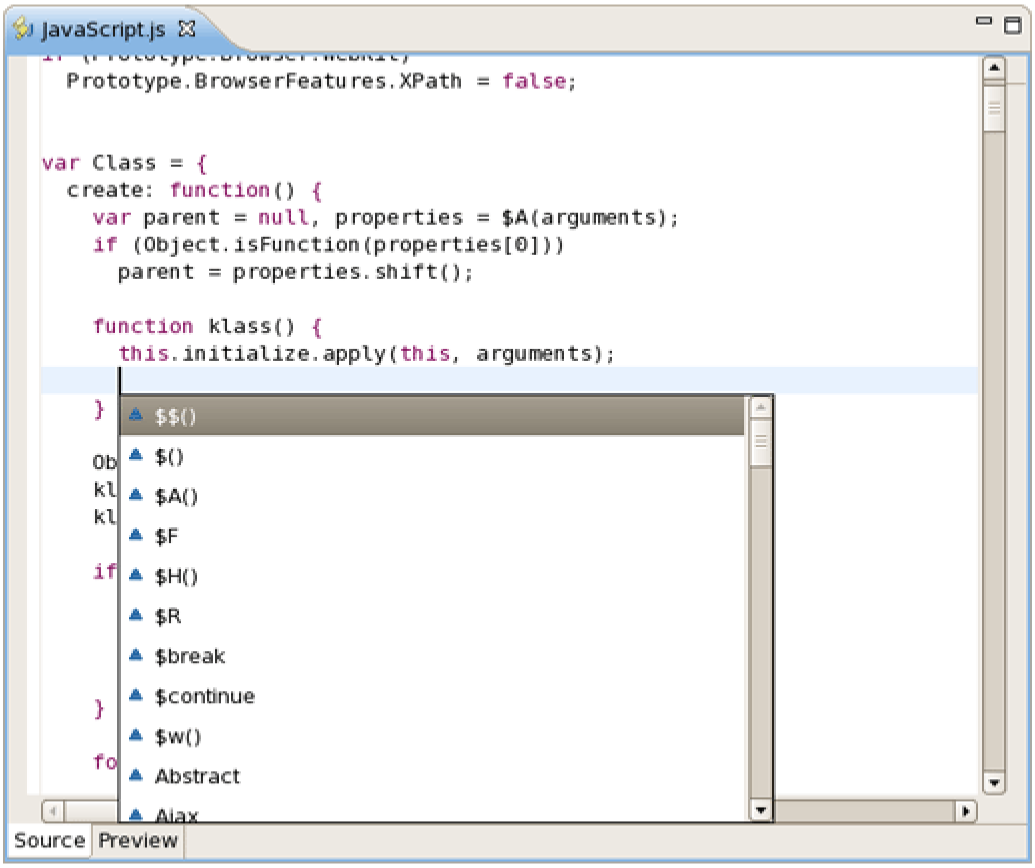
Property	Value
Aural	
Box model	
Colors and Background:	
Fonts	
font	
font-family	
font-size	15px
font-size-adjust	
font-stretch	
font-style	
font-variant	
font-weight	
Generated content/List	
Paged media	
Tables	
Text	
User interface	
Visual	

Figure 3.96. Properties View in CSS Editor

To make you work on CSS files more comfortable, CSS perspective is available, read more about it in *CSS Editing Perspective chapter*

3.3.5. JavaScript Editor

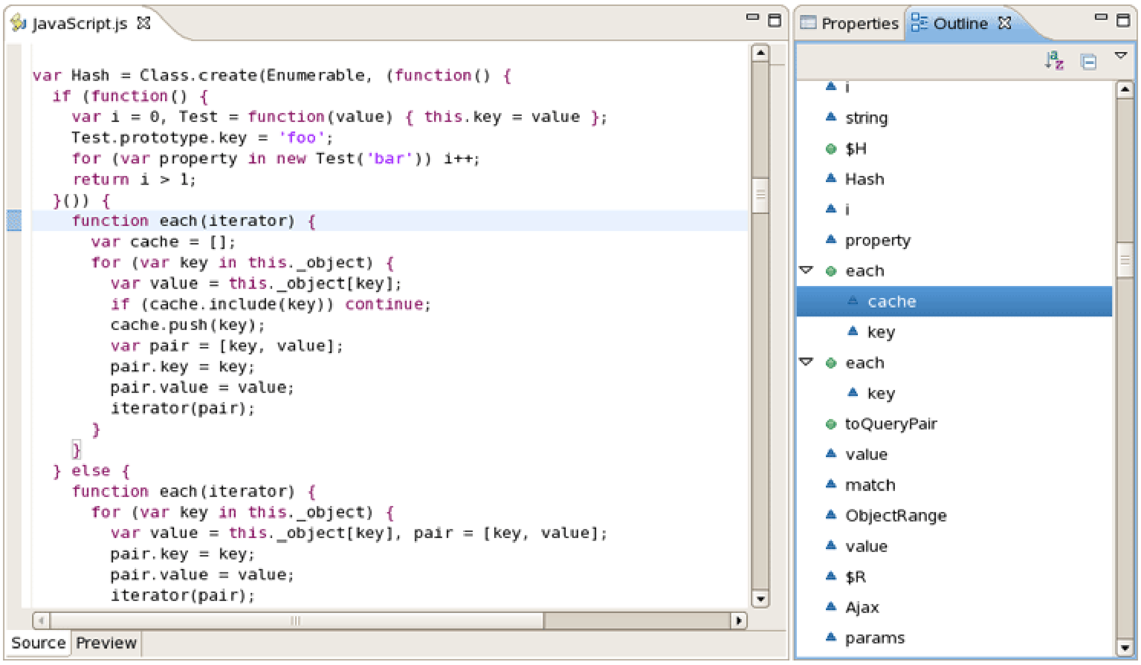
The JavaScript editor includes a Preview viewer and a Source viewer. In the Source viewer, you can use code assist:



```
  Prototype.BrowserFeatures.XPath = false;


var Class = {
  create: function() {
    var parent = null, properties = $A(arguments);
    if (Object.isFunction(properties[0]))
      parent = properties.shift();

    function klass() {
      this.initialize.apply(this, arguments);

    }

    Ob
    kl
    kl

    if




    }

    fo
```

Figure 3.97. JavaScript Editor

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


JavaScript.js
Properties
Outline
i
string
$H
Hash
i
property
each
cache
key
each
key
toQueryPair
value
match
ObjectRange
value
$R
Ajax
params
Source
Preview

```
var Hash = Class.create(Enumerable, (function() {
  if (function() {
    var i = 0, Test = function(value) { this.key = value };
    Test.prototype.key = 'foo';
    for (var property in new Test('bar')) i++;
    return i > 1;
  }()) {
    function each(iterator) {
      var cache = [];
      for (var key in this._object) {
        var value = this._object[key];
        if (cache.include(key)) continue;
        cache.push(key);
        var pair = [key, value];
        pair.key = key;
        pair.value = value;
        iterator(pair);
      }
    }
  } else {
    function each(iterator) {
      for (var key in this._object) {
        var value = this._object[key], pair = [key, value];
        pair.key = key;
        pair.value = value;
        iterator(pair);
```

Figure 3.98. Source Viewer in JavaScript Editor

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 *WTP Getting Started*).

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.

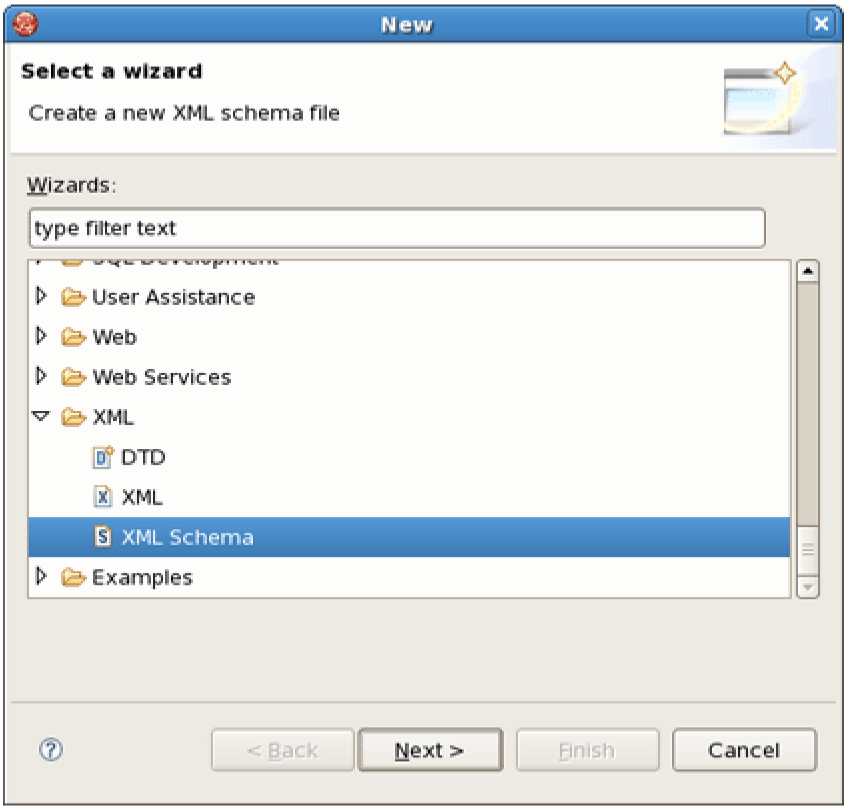
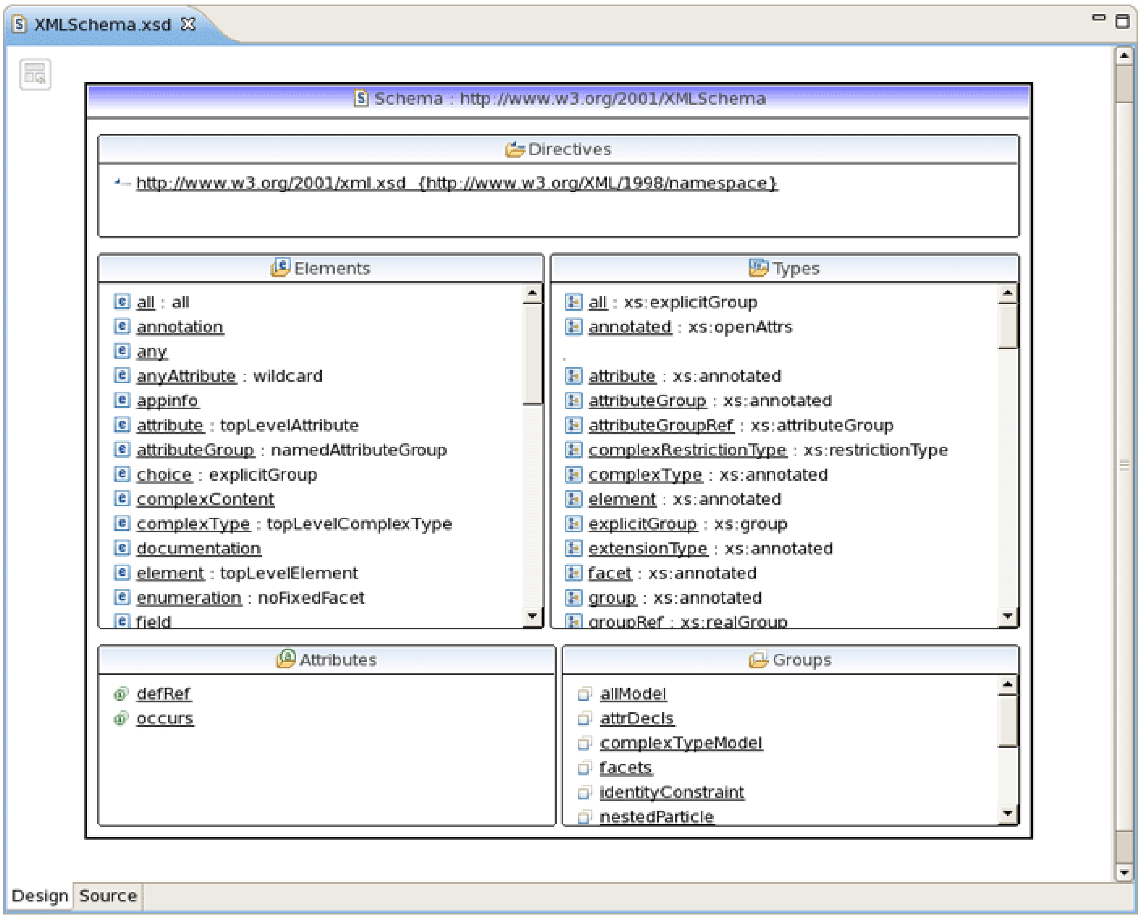

New
Select a wizard
Create a new XML schema file
Wizards:
type filter text
User Assistance
Web
Web Services
XML
DTD
XML
XML Schema
Examples
< Back
Next >
Finish
Cancel

Figure 3.99. Creating New XSD file

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



Schema : http://www.w3.org/2001/XMLSchema

Directives

http://www.w3.org/2001/xml.xsd {http://www.w3.org/XML/1998/namespace}

Elements

all : all
annotation
any
anyAttribute : wildcard
appinfo
attribute : topLevelAttribute
attributeGroup : namedAttributeGroup
choice : explicitGroup
complexContent
complexType : topLevelComplexType
documentation
element : topLevelElement
enumeration : noFixedFacet
field

Types

all : xs:explicitGroup
annotated : xs:openAttrs
attribute : xs:annotated
attributeGroup : xs:annotated
attributeGroupRef : xs:attributeGroup
complexRestrictionType : xs:restrictionType
complexType : xs:annotated
element : xs:annotated
explicitGroup : xs:group
extensionType : xs:annotated
facet : xs:annotated
group : xs:annotated
groupRef : xs:realGroup

Attributes

defRef
occurs

Groups

allModel
attrDecls
complexTypeModel
facets
identityConstraint
nestedParticle

Figure 3.100. Source Viewer in XSD Editor

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

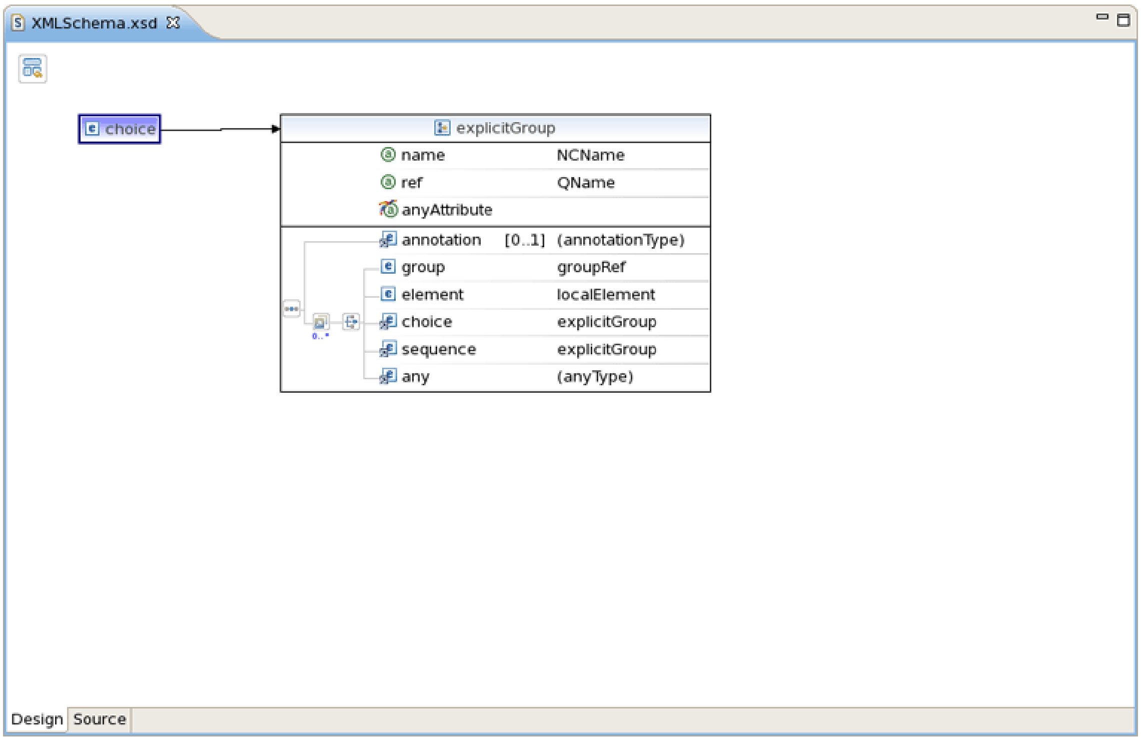

XMLSchema.xsd
choice
explicitGroup
name NCName
ref QName
anyAttribute
annotation [0..1] (annotationType)
group groupRef
element localElement
choice explicitGroup
sequence explicitGroup
any (anyType)
0..*
Design
Source

Figure 3.101. Design Viewer in XSD Editor

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

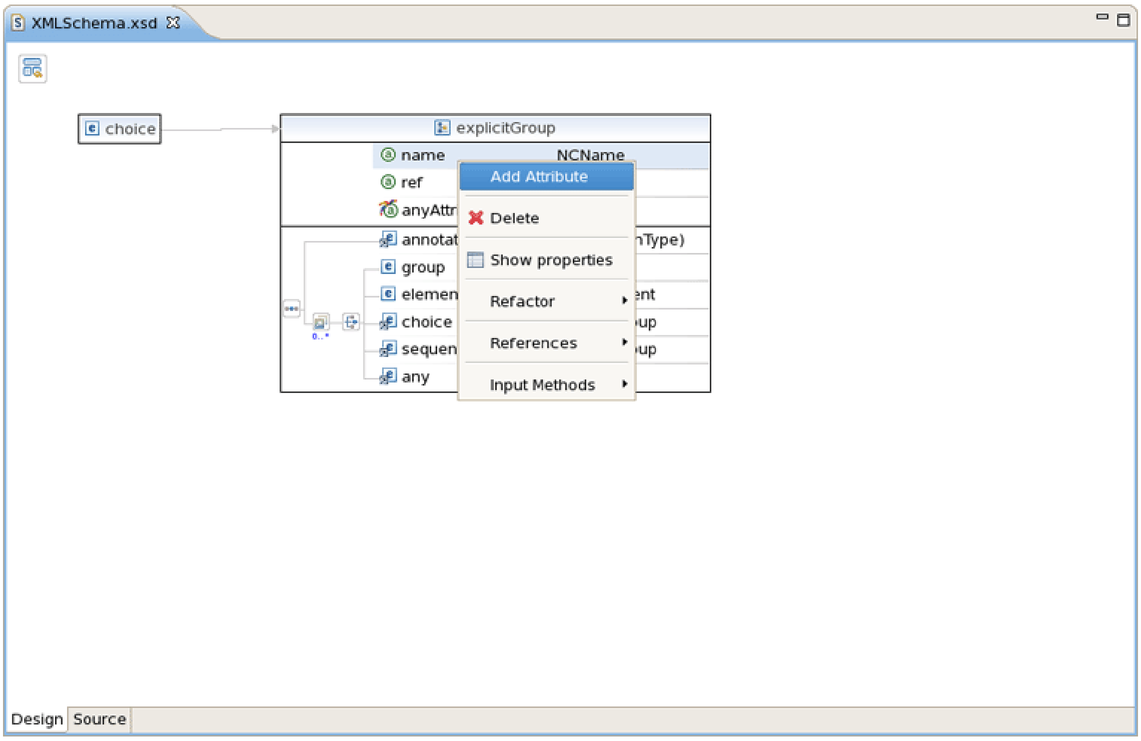
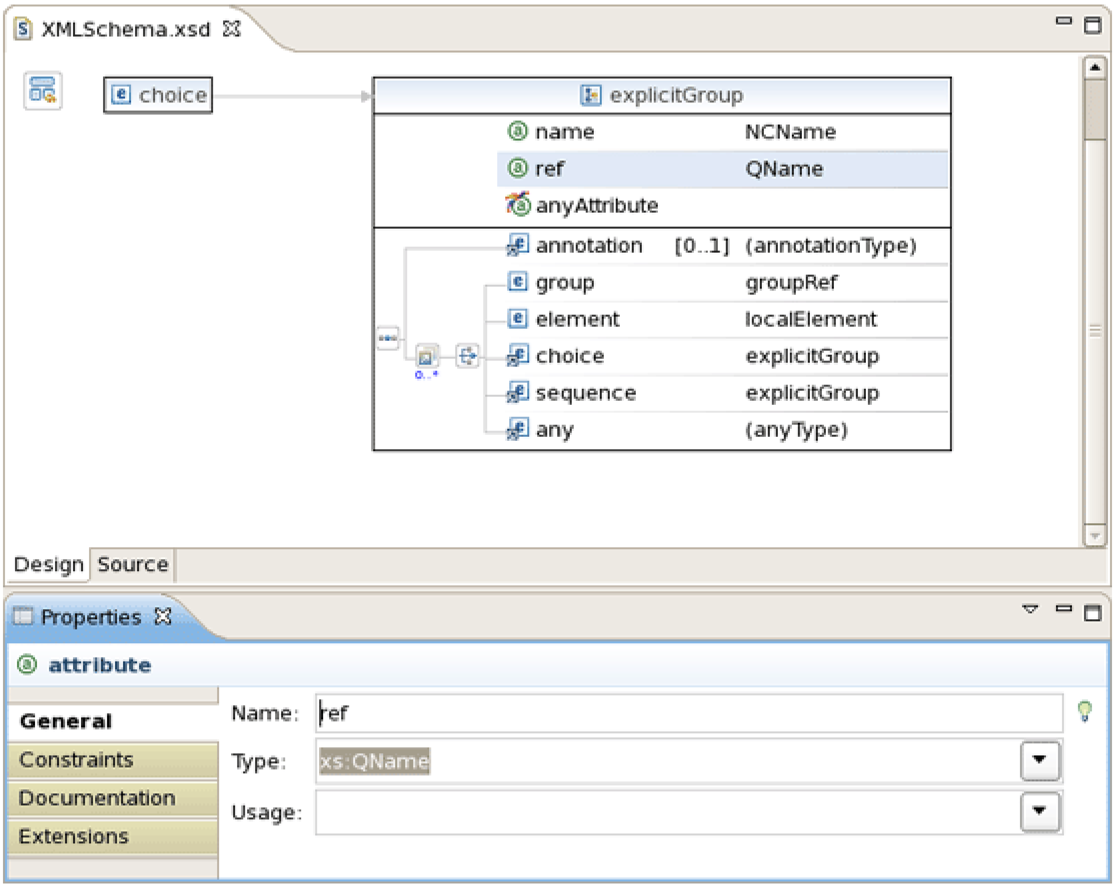

XMLSchema.xsd
choice
explicitGroup
name NCName
ref
anyAttr
annotat
group
elemen
choice
sequen
any
Add Attribute
Delete
Show properties
Refactor
References
Input Methods
Design
Source

Figure 3.102. Edit Options in XSD Editor Context Menu.

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


XMLSchema.xsd
choice
explicitGroup
name
NCName
ref
QName
anyAttribute
annotation
[0..1]
(annotationType)
group
groupRef
element
localElement
choice
explicitGroup
sequence
explicitGroup
any
(anyType)
0..*
Design
Source
Properties
attribute
General
Constraints
Documentation
Extensions

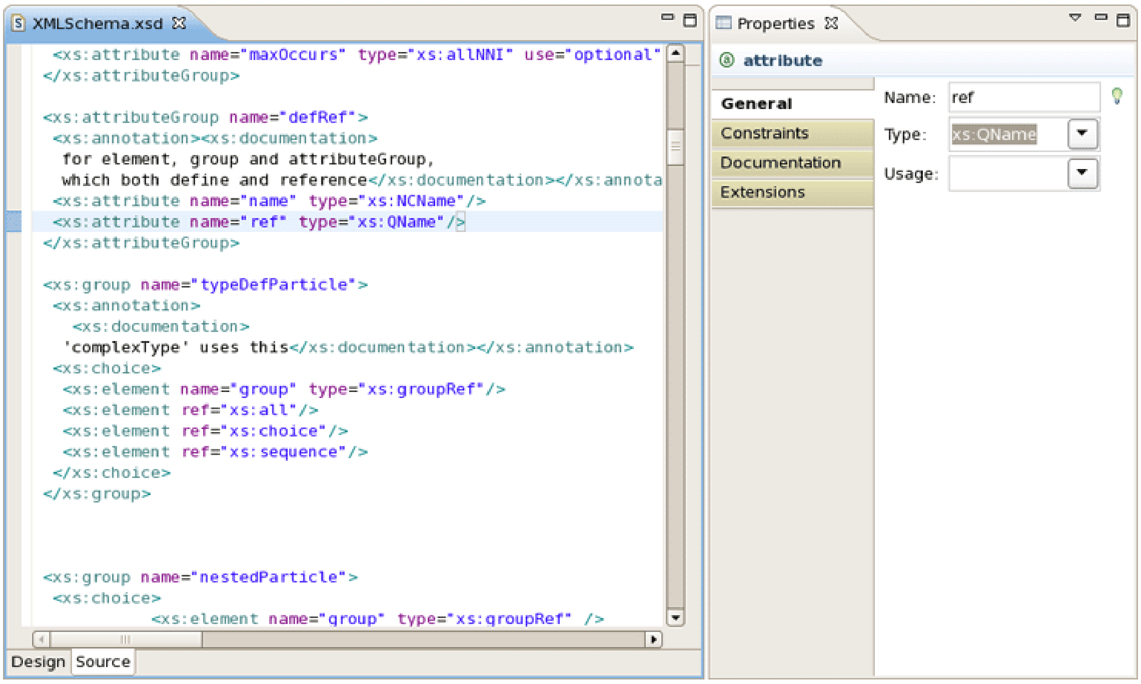
Name: ref

Type: xs:QName

Usage:

Figure 3.103. Properties View in XSD Editor

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:


XMLSchema.xsd
Design
Source
Properties
attribute
General
Constraints
Documentation
Extensions

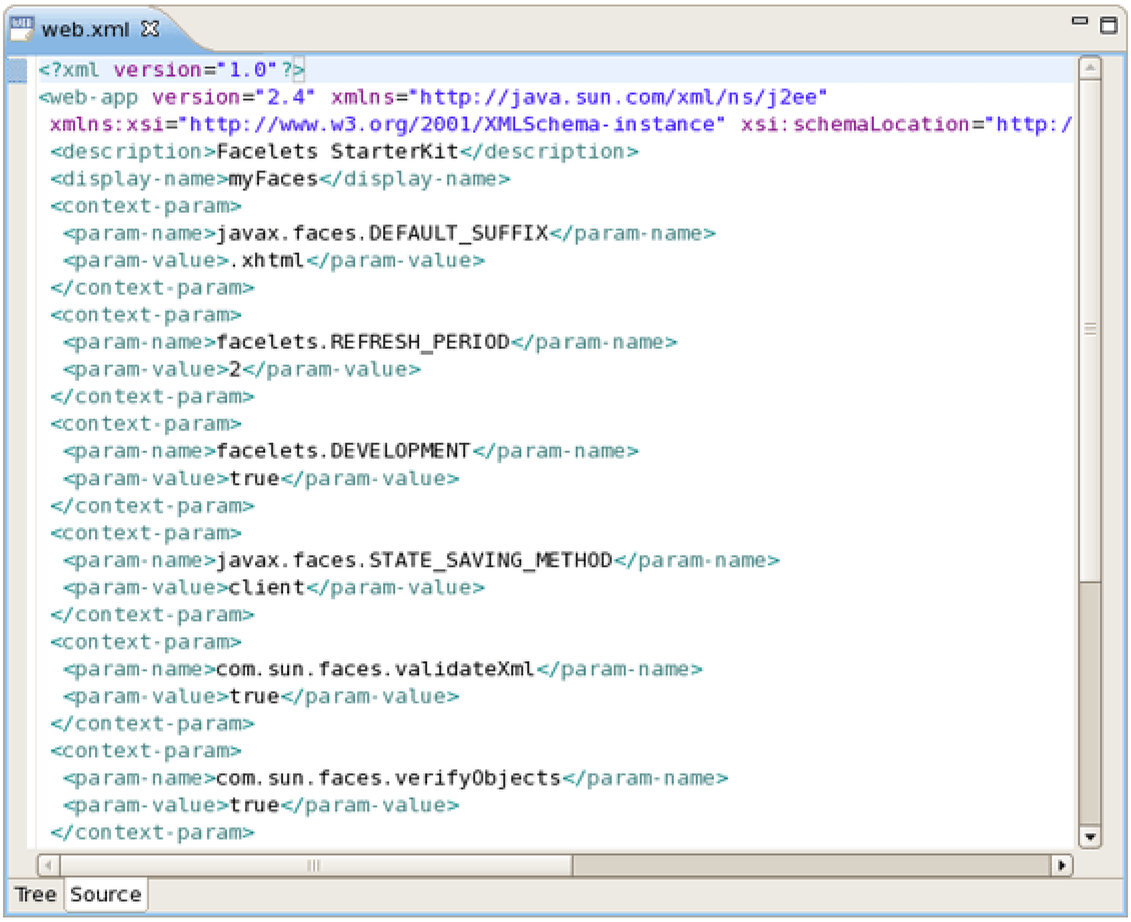
```
  <xs:attribute name="maxOccurs" type="xs:allNNI" use="optional"
 </xs:attributeGroup>

 <xs:attributeGroup name="defRef">
  <xs:annotation><xs:documentation>
   for element, group and attributeGroup,
   which both define and reference</xs:documentation></xs:annota
  <xs:attribute name="name" type="xs:NCName"/>
  <xs:attribute name="ref" type="xs:QName"/>
 </xs:attributeGroup>

 <xs:group name="typeDefParticle">
  <xs:annotation>
    <xs:documentation>
   'complexType' uses this</xs:documentation></xs:annotation>
  <xs:choice>
   <xs:element name="group" type="xs:groupRef"/>
   <xs:element ref="xs:all"/>
   <xs:element ref="xs:choice"/>
   <xs:element ref="xs:sequence"/>
  </xs:choice>
 </xs:group>



 <xs:group name="nestedParticle">
  <xs:choice>
            <xs:element name="group" type="xs:groupRef" />
```

Name: ref
Type: xs:QName
Usage:

Figure 3.104. Using Source Viewer and Properties View together for source code editing

3.3.7. Support for XML Schema

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



```
<?xml version="1.0"?>
<web-app version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http:/
 <description>Facelets StarterKit</description>
 <display-name>myFaces</display-name>
 <context-param>
  <param-name>javax.faces.DEFAULT_SUFFIX</param-name>
  <param-value>.xhtml</param-value>
 </context-param>
 <context-param>
  <param-name>facelets.REFRESH_PERIOD</param-name>
  <param-value>2</param-value>
 </context-param>
 <context-param>
  <param-name>facelets.DEVELOPMENT</param-name>
  <param-value>true</param-value>
 </context-param>
 <context-param>
  <param-name>javax.faces.STATE_SAVING_METHOD</param-name>
  <param-value>client</param-value>
 </context-param>
 <context-param>
  <param-name>com.sun.faces.validateXml</param-name>
  <param-value>true</param-value>
 </context-param>
 <context-param>
  <param-name>com.sun.faces.verifyObjects</param-name>
  <param-value>true</param-value>
 </context-param>
```

Figure 3.105. 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

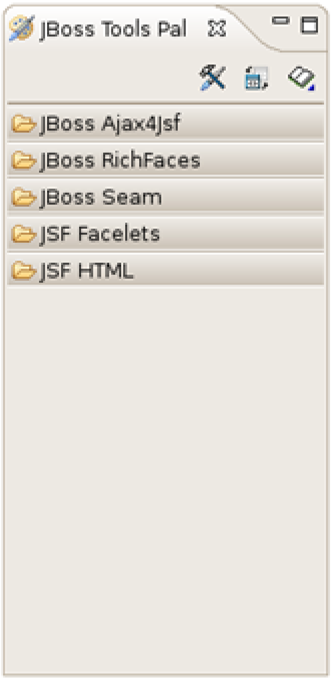


Figure 4.1. Default View of The JBoss Tools Palette

By default JBoss Tools Palette is not displayed. If you want to use it select *Window > Show View Other... > JBoss Tools Web > JBoss Tools Palette* from the menu bar.

The standard Eclipse Palette is displayed by default in both Web Development and Seam perspectives. Now, the standard Eclipse Palette is featured with all JBoss Tools Palette options and capabilities.

To open the standard Eclipse Palette navigate to *Window->Show View->Others->General->Palette*.

The differences between the two palettes are as follows:

- The standard Eclipse Palette is blank by default; the content of the palette is available only if Visual Page Editor is open and active, while JBoss Tools Palette always contains a predefined set of components.
- The Expanded/Collapsed state of components in the standard Eclipse Palette is not global as in JBoss Tools Palette: the state is associated with an instance of Visual Page Editor. It means that the state can be different for various files, and each new file opened in Visual Page Editor will have the default state of Palette with all components collapsed.

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

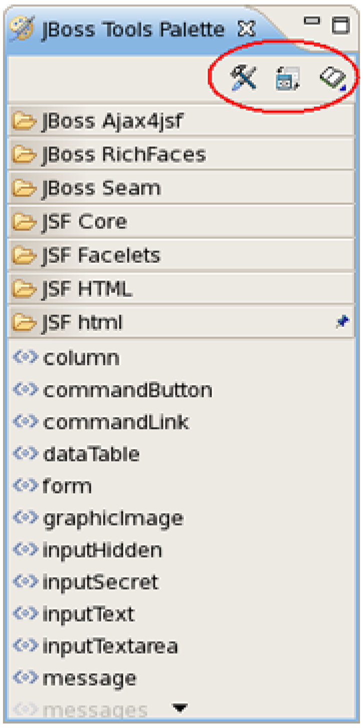

JBoss Tools Palette
JBoss Ajax4jsf
JBoss RichFaces
JBoss Seam
JSF Core
JSF Facelets
JSF HTML
JSF html
column
commandButton
commandLink
dataTable
form
graphicImage
inputHidden
inputSecret
inputText
inputTextarea
message
messages

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.

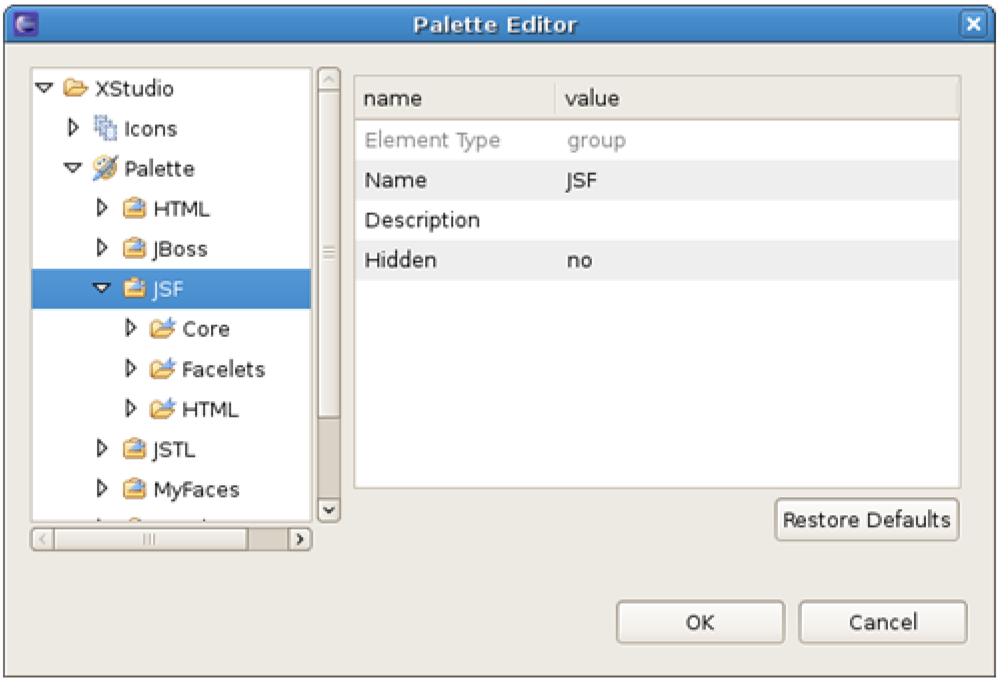
To open the editor, click on the *Palette Editor* icon(🛠).

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.

[redacted]

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


Palette Editor
XStudio
Icons
Palette
HTML
JBoss
JSF
Core
Facelets
HTML
JSTL
MyFaces
Restore Defaults
OK
Cancel

name	value
Element Type	group
Name	JSF
Description	
Hidden	no

Figure 4.3. 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

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

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

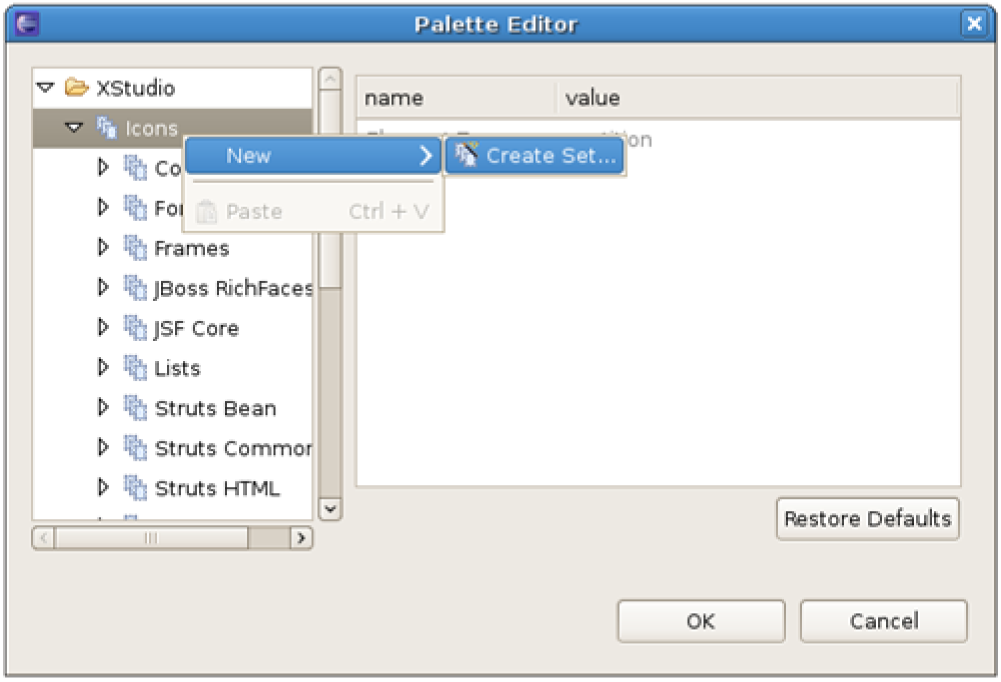

Palette Editor
XStudio
Icons
Co
Foi
Frames
JBoss RichFaces
JSF Core
Lists
Struts Bean
Struts Commor
Struts HTML
New
Paste Ctrl + V
Create Set...
name
value
Restore Defaults
OK
Cancel

Figure 4.4. 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.

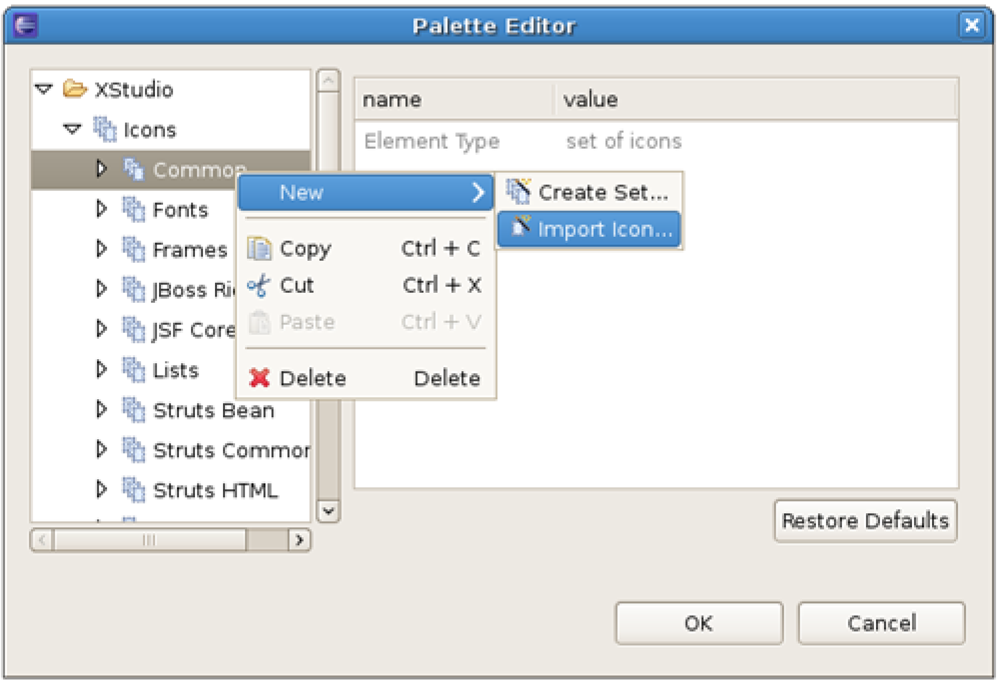

Palette Editor
XStudio
Icons
Common
Fonts
Frames
JBoss Ri
JSF Core
Lists
Struts Bean
Struts Commor
Struts HTML
New
Copy Ctrl + C
Cut Ctrl + X
Paste Ctrl + V
Delete Delete
Create Set...
Import Icon...
name
value
Element Type
set of icons
Restore Defaults
OK
Cancel

Figure 4.5. 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.

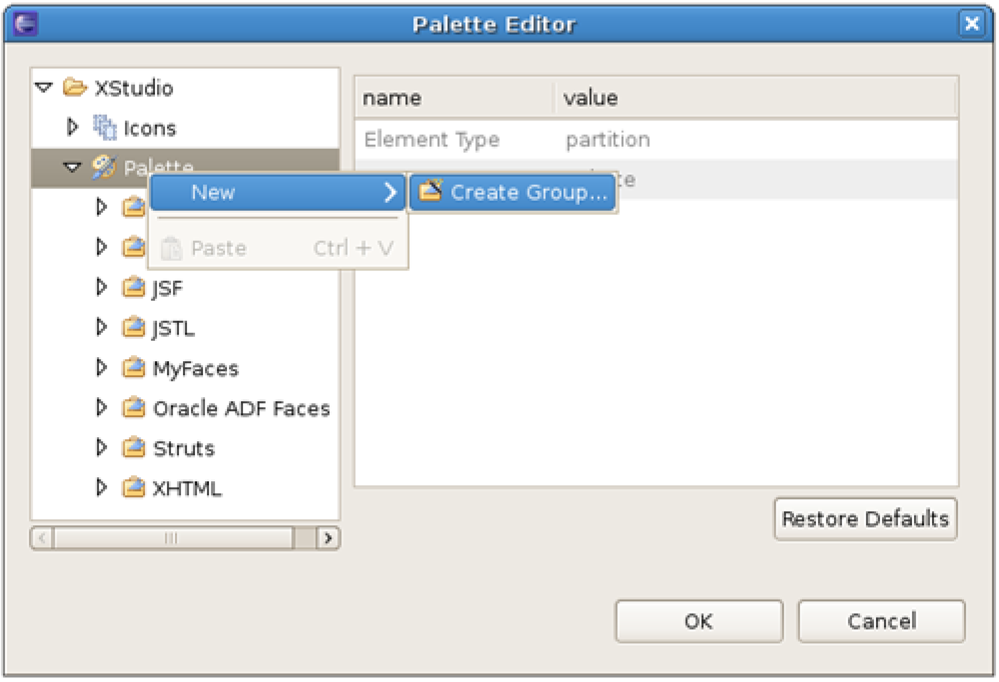

Palette Editor
XStudio
Icons
Palette
JSF
JSTL
MyFaces
Oracle ADF Faces
Struts
XHTML
New
Create Group...
Paste Ctrl + V
name
value
Element Type
partition
Restore Defaults
OK
Cancel

Figure 4.6. Creating a Group of Tag Libraries

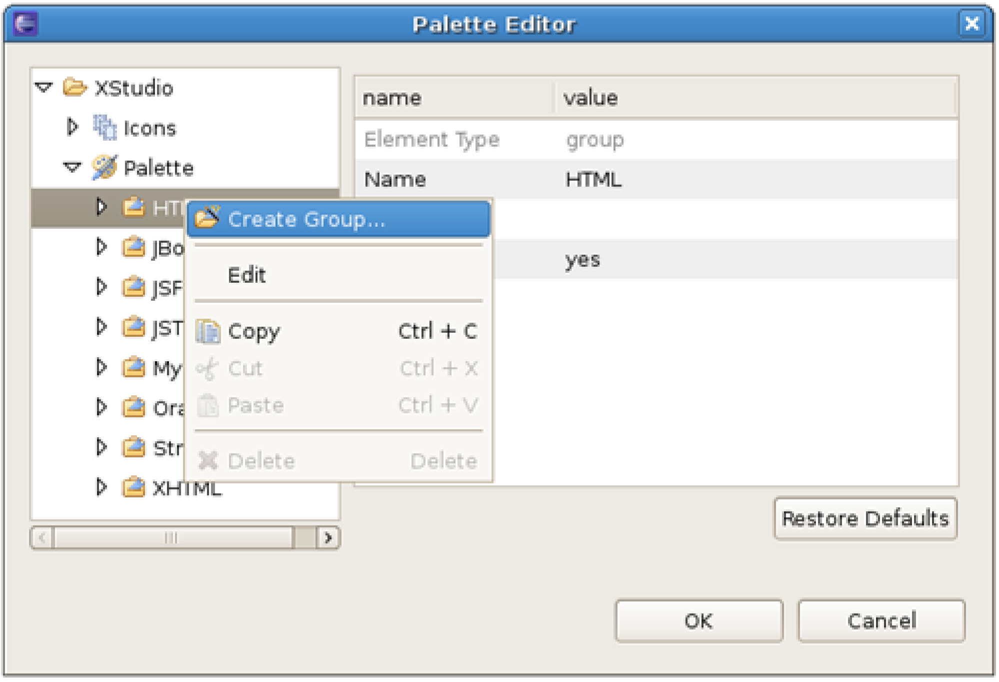
You are allowed to edit or delete a group, as well. If you'd like to change attributes of a group, use the right editing window of the palette editor or the *Edit...* option, like it was mentioned before. In order to remove the group, right click on the group that you wish to remove and choose the *Delete*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

Important:

The removal option is enabled only for custom folders.

- 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.


Palette Editor
XStudio
Icons
Palette
HTI
JBo
JSF
JST
My
Ora
Str
XHTML
Create Group...
Edit
Copy Ctrl + C
Cut Ctrl + X
Paste Ctrl + V
Delete Delete
Restore Defaults
OK
Cancel

name	value
Element Type	group
Name	HTML
	yes

Figure 4.7. Creating a tag library

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

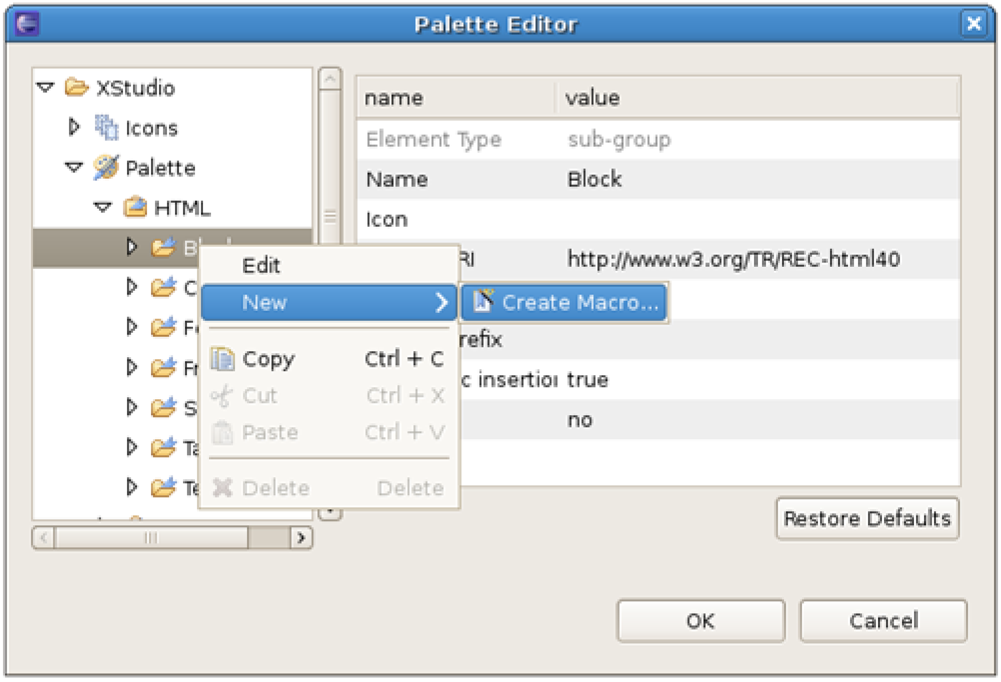
> **Note:**
>
> If you do not choose an icon the default one will be assigned.

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.

> **Important:**
>
> The removal option is enabled only for custom tag libraries.

- to work with a tag element

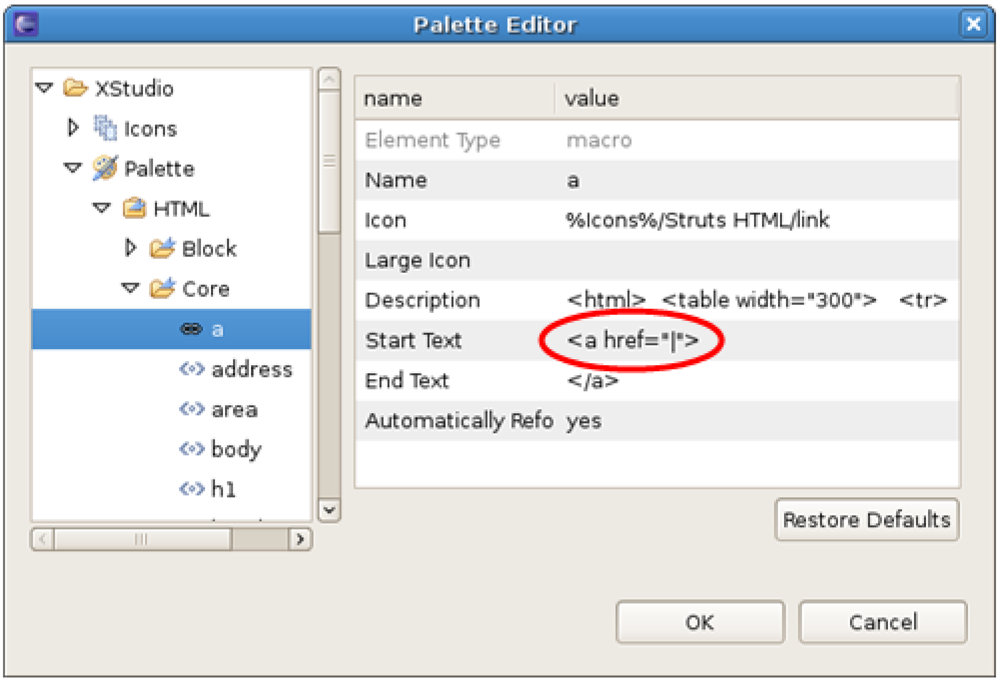
  When the library folder is created, new tags can be added to it. Choose the required library and select the option *Create > Create Macro…* from the pop-up menu that appears after you right-click on a folder.


Palette Editor
XStudio
Icons
Palette
HTML
Edit
New
Create Macro...
Copy Ctrl + C
Cut Ctrl + X
Paste Ctrl + V
Delete Delete
Restore Defaults
OK
Cancel

name	value
Element Type	sub-group
Name	Block
Icon	
RI	http://www.w3.org/TR/REC-html40
refix	
c insertioı	true
	no

Figure 4.8. 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.



name	value
Element Type	macro
Name	a
Icon	%Icons%/Struts HTML/link
Large Icon	
Description	<html> <table width="300"> <tr>
Start Text	
End Text	
Automatically Refo	yes

Figure 4.9. Parameters of the Palette element

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

Note:

If you do not choose an icon the default one will be assigned.

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.

If you have changed any abject in the tree view and you don't like the final result you can always use the *Restore Defaults* button. Click on it will restore defaults for the object selected and for its children elements. Please remember that the button will only restore data for objects defined in the default palette. If selected object is created by you, the button will be disabled. Child objects added by you will not be removed.

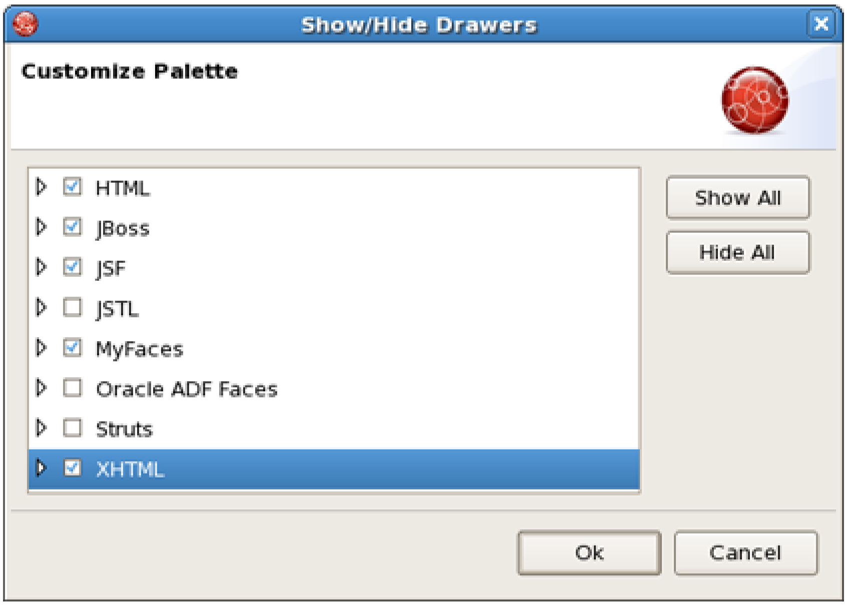
When updating JBoss Tools the palette content is not updated.

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.

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

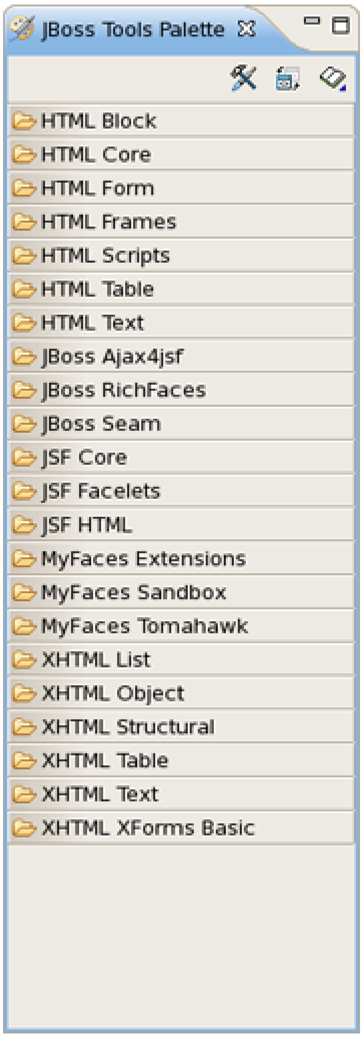


- ☑ HTML
- ☑ JBoss
- ☑ JSF
- ☐ JSTL
- ☑ MyFaces
- ☐ Oracle ADF Faces
- ☐ Struts
- ☑ XHTML

Figure 4.10. 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
HTML Block
HTML Core
HTML Form
HTML Frames
HTML Scripts
HTML Table
HTML Text
JBoss Ajax4jsf
JBoss RichFaces
JBoss Seam
JSF Core
JSF Facelets
JSF HTML
MyFaces Extensions
MyFaces Sandbox
MyFaces Tomahawk
XHTML List
XHTML Object
XHTML Structural
XHTML Table
XHTML Text
XHTML XForms Basic

Figure 4.11. 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

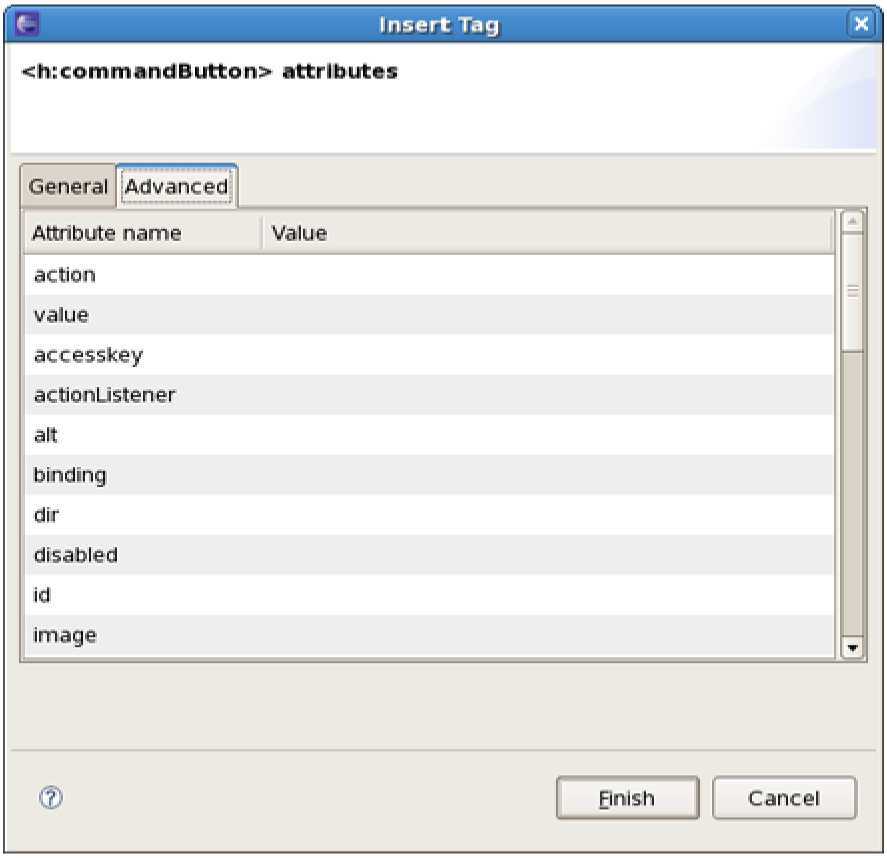
The Import button lets you add a custom or 3rd party tag library to JBoss Tools Palette. Find out more information on how to add particular tags see the *Adding Custom JSF Tags* section.

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.

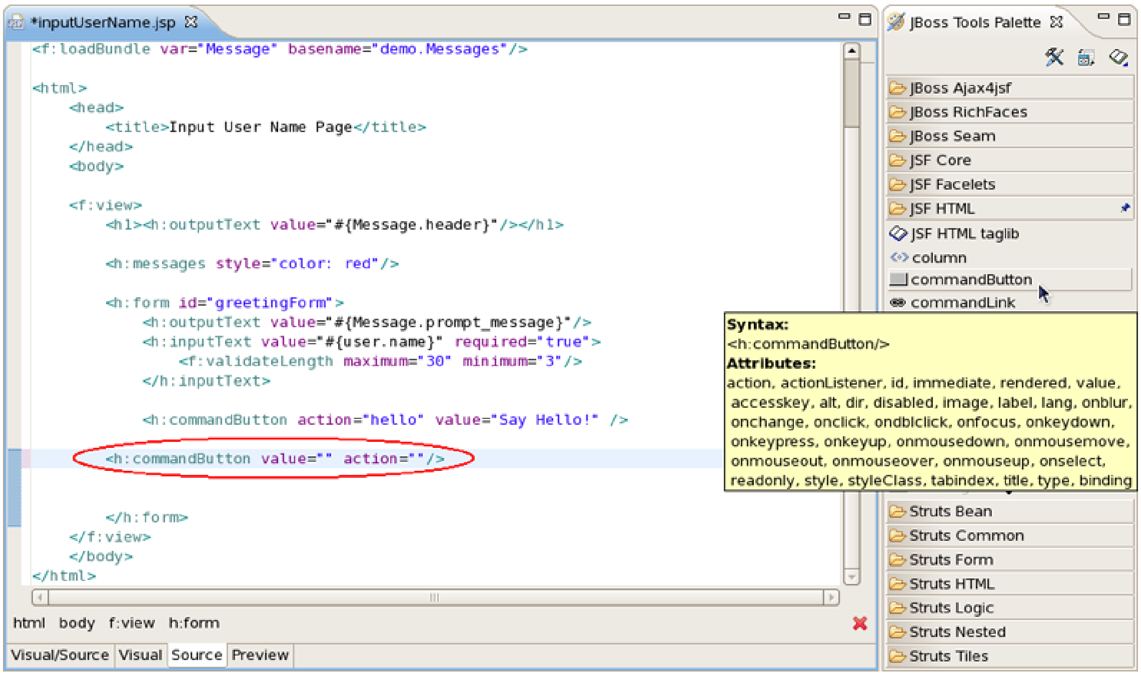

Insert Tag
General
Advanced
Finish
Cancel

<h:commandButton> attributes

Attribute name	Value
action	
value	
accesskey	
actionListener	
alt	
binding	
dir	
disabled	
id	
image	

Figure 4.12. Inserting Tag

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


*inputUserName.jsp
html body f:view h:form
Visual/Source
Visual
Source
Preview
JBoss Tools Palette
JBoss Ajax4jsf
JBoss RichFaces
JBoss Seam
JSF Core
JSF Facelets
JSF HTML
JSF HTML taglib
column
commandButton
commandLink
Struts Bean
Struts Common
Struts Form
Struts HTML
Struts Logic
Struts Nested
Struts Tiles

```
<f:loadBundle var="Message" basename="demo.Messages"/>

<html>
    <head>
        <title>Input User Name Page</title>
    </head>
    <body>

    <f:view>
        <h1><h:outputText value="#{Message.header}"/></h1>

        <h:messages style="color: red"/>

        <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>

            <h:commandButton action="hello" value="Say Hello!" />

        <h:commandButton value="" action=""/>


        </h:form>
    </f:view>
    </body>
</html>
```

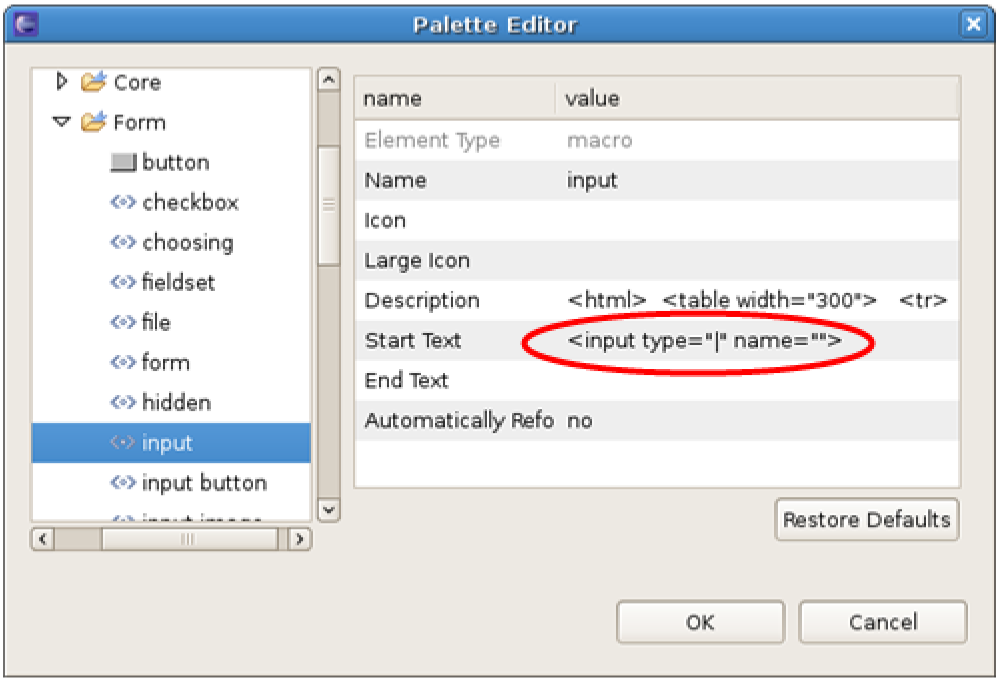
Syntax:
<h:commandButton/>
Attributes:
action, actionListener, id, immediate, rendered, value, accesskey, alt, dir, disabled, image, label, lang, onblur, onchange, onclick, ondblclick, onfocus, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, onselect, readonly, style, styleClass, tabindex, title, type, binding

Figure 4.13. Inserting Tag

Tip:

if you place the cursor over any tag, a balloon hint is shown with all the *"tag"* attributes.

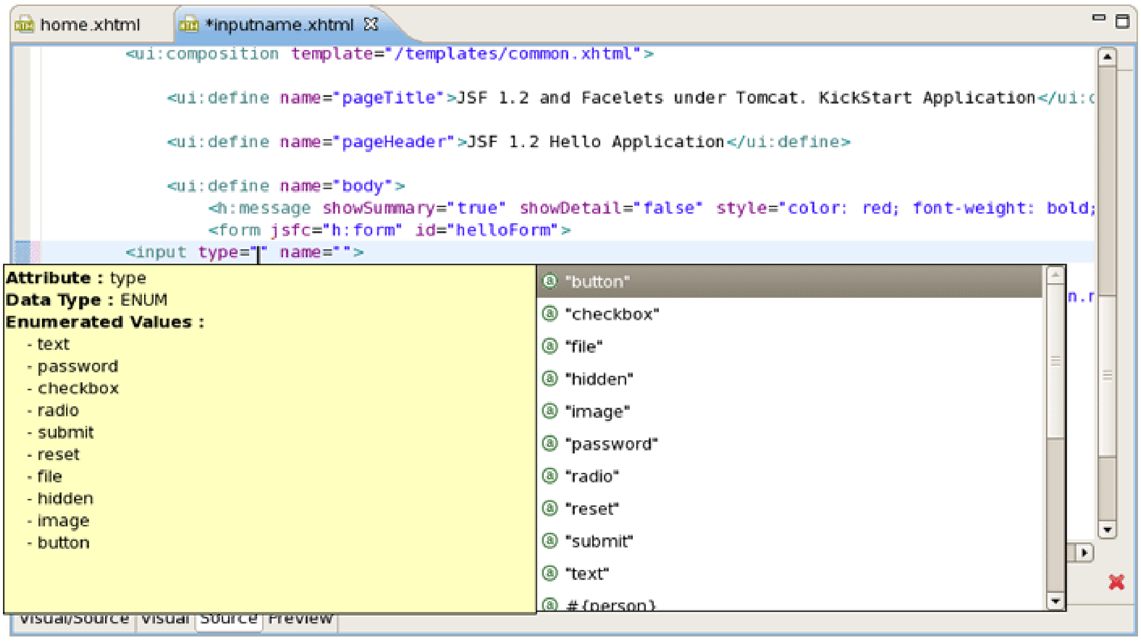
The cursor position after adding a tag into a file is specified by "|" symbol in the tag template on the right in the Palette Editor window.


Palette Editor
Core
Form
button
checkbox
choosing
fieldset
file
form
hidden
input
input button
Restore Defaults
OK
Cancel

name	value
Element Type	macro
Name	input
Icon	
Large Icon	
Description	`<html> <table width="300"> <tr>`
Start Text	`<input type="|" name="">`
End Text	
Automatically Refo	no

Figure 4.14. 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.


home.xhtml
*inputname.xhtml
"button"
"checkbox"
"file"
"hidden"
"image"
"password"
"radio"
"reset"
"submit"
"text"
#{person}
Visual/Source
Visual
Source
Preview

```
        <ui:composition template="/templates/common.xhtml">

            <ui:define name="pageTitle">JSF 1.2 and Facelets under Tomcat. KickStart Application</ui:c

            <ui:define name="pageHeader">JSF 1.2 Hello Application</ui:define>

            <ui:define name="body">
                <h:message showSummary="true" showDetail="false" style="color: red; font-weight: bold;
                <form jsfc="h:form" id="helloForm">
        <input type="" name="">
```

Attribute : type
Data Type : ENUM
Enumerated Values :
- text
- password
- checkbox
- radio
- submit
- reset
- file
- hidden
- image
- button

Figure 4.15. Cursor position

4.2.2. Adding Custom JSF Tags to the JBoss Tools Palette

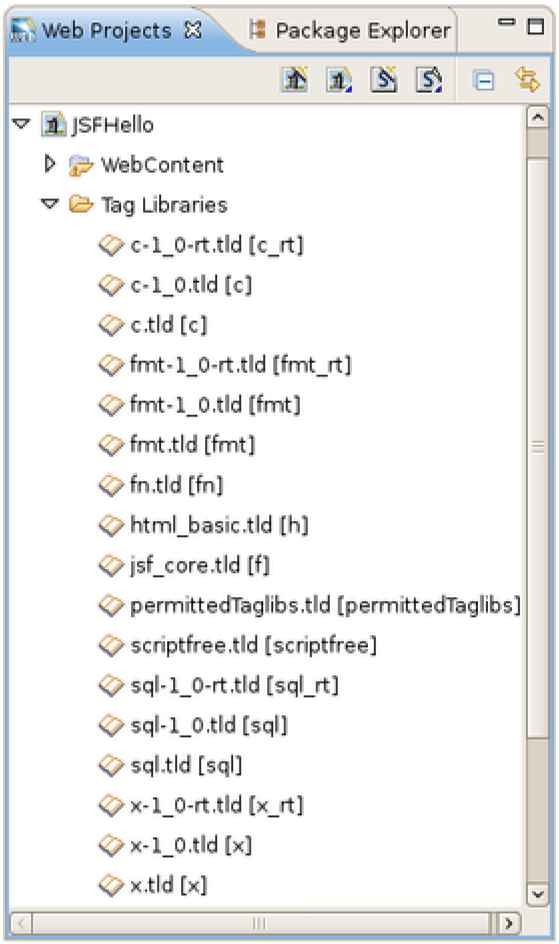
There are two ways to add any custom (including custom Facelets libraries) or 3rd party tag library to the JBoss Tools Palette:

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

Before you add your custom component library, you need to make sure it is included in your project. You need to either place the *".tld"* file or the *".jar"* that includes your tag library under the lib folder in your project. Or you can just add *".tld"* or *".jar"* file to the classpath , and the library will be added to the Tag Library List in Web Projects View.

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.

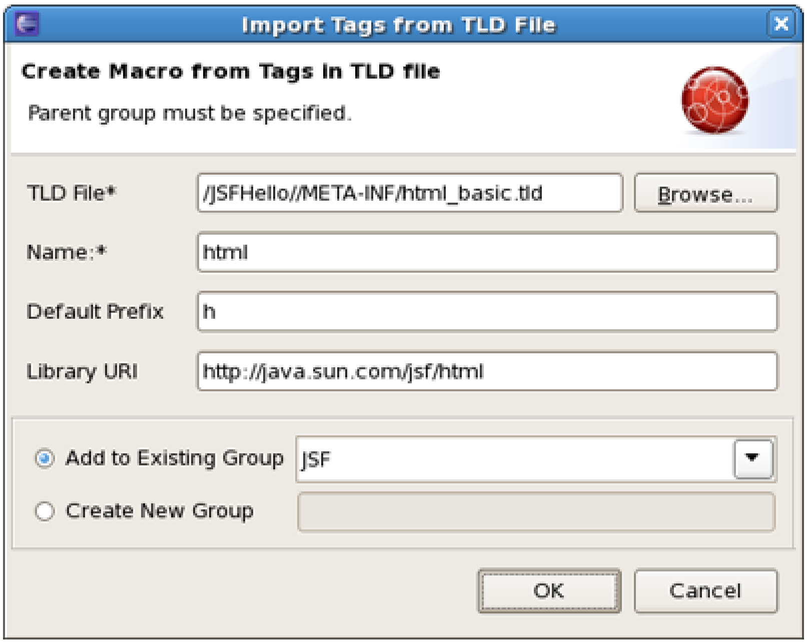


- JSFHello
  - WebContent
  - Tag Libraries
    - c-1_0-rt.tld [c_rt]
    - c-1_0.tld [c]
    - c.tld [c]
    - fmt-1_0-rt.tld [fmt_rt]
    - fmt-1_0.tld [fmt]
    - fmt.tld [fmt]
    - fn.tld [fn]
    - html_basic.tld [h]
    - jsf_core.tld [f]
    - permittedTaglibs.tld [permittedTaglibs]
    - scriptfree.tld [scriptfree]
    - sql-1_0-rt.tld [sql_rt]
    - sql-1_0.tld [sql]
    - sql.tld [sql]
    - x-1_0-rt.tld [x_rt]
    - x-1_0.tld [x]
    - x.tld [x]

Figure 4.16. 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.


Import Tags from TLD File
Browse...
OK
Cancel

Create Macro from Tags in TLD file

Parent group must be specified.

TLD File* /JSFHello//META-INF/html_basic.tld

Name:* html

Default Prefix h

Library URI http://java.sun.com/jsf/html

● Add to Existing Group JSF

○ Create New Group

Figure 4.17. Import Tags From TLD File Form

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

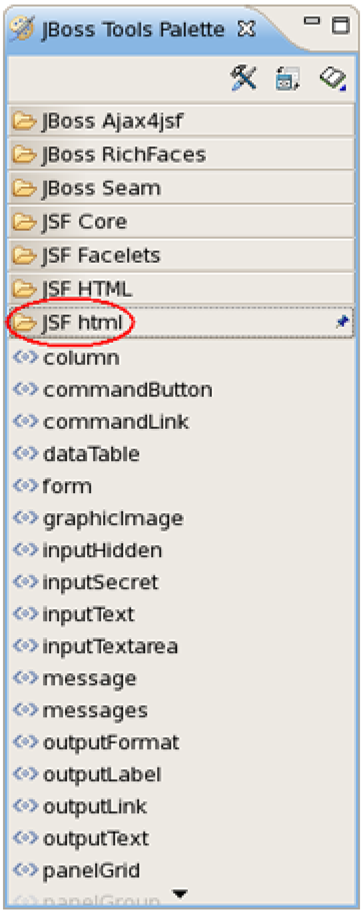
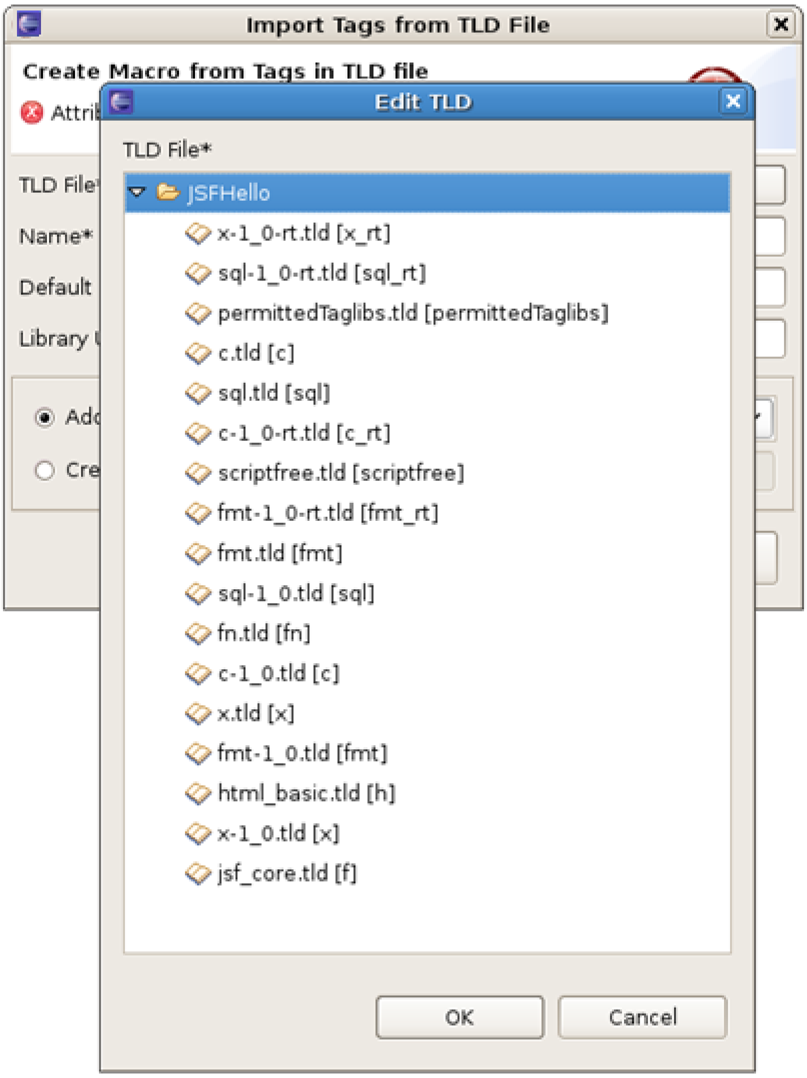

JBoss Tools Palette
JBoss Ajax4jsf
JBoss RichFaces
JBoss Seam
JSF Core
JSF Facelets
JSF HTML
JSF html
column
commandButton
commandLink
dataTable
form
graphicImage
inputHidden
inputSecret
inputText
inputTextarea
message
messages
outputFormat
outputLabel
outputLink
outputText
panelGrid
panelGroup

Figure 4.18. 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.

By clicking on the *Import button* you will see the Import Tag window a similar like in the *Drag-and-Drop* 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:


Import Tags from TLD File
Create Macro from Tags in TLD file
Attrib
TLD File
Name*
Default
Library
Add
Cre
Edit TLD
TLD File*
OK
Cancel

- JSFHello
  - x-1_0-rt.tld [x_rt]
  - sql-1_0-rt.tld [sql_rt]
  - permittedTaglibs.tld [permittedTaglibs]
  - c.tld [c]
  - sql.tld [sql]
  - c-1_0-rt.tld [c_rt]
  - scriptfree.tld [scriptfree]
  - fmt-1_0-rt.tld [fmt_rt]
  - fmt.tld [fmt]
  - sql-1_0.tld [sql]
  - fn.tld [fn]
  - c-1_0.tld [c]
  - x.tld [x]
  - fmt-1_0.tld [fmt]
  - html_basic.tld [h]
  - x-1_0.tld [x]
  - jsf_core.tld [f]

Figure 4.19. Select TLD File

CSS Editing Perspective

In this chapter we will discuss CSS Editing Perspective views, more information about style sheets can be found in *Page Styling section* of Editor chapter

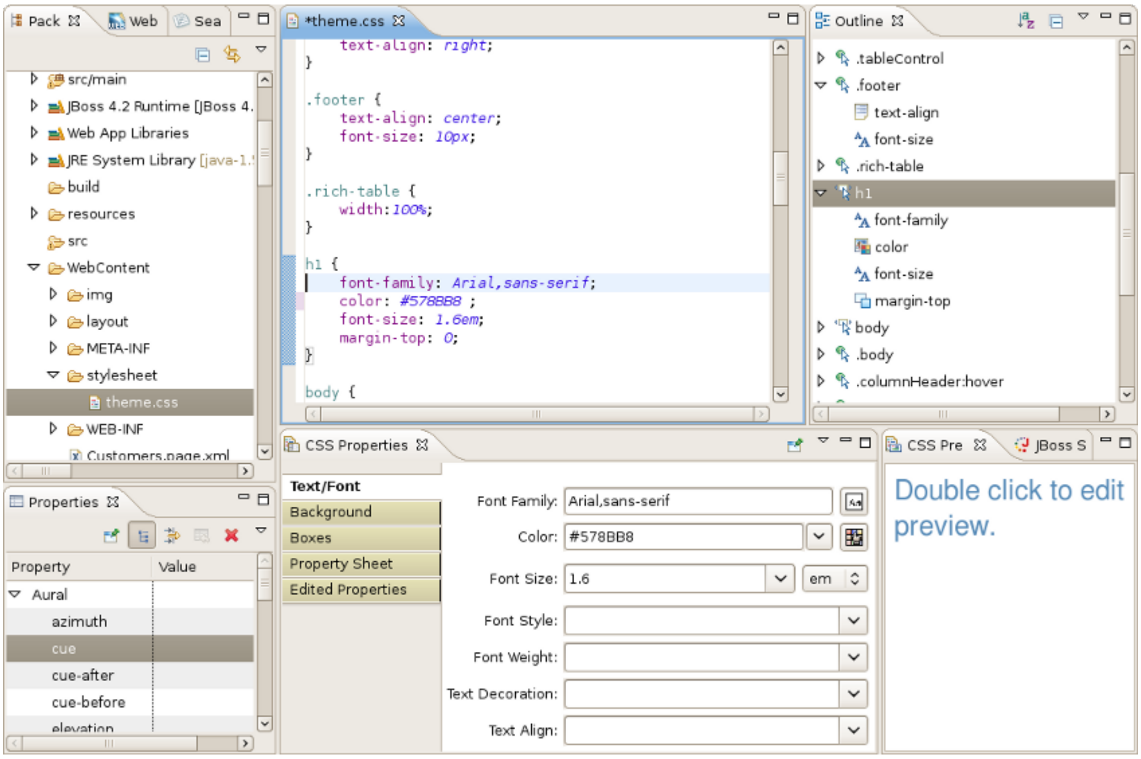
The CSS Editing Perspective combines a set of views which allow you to see the structure of your css files,edit them and see the results. To use this perspective you need to choose *Window >Open Perspective> CSS Editing* .All of the views are fully synchronized with each other:the changes being made in one view are reflected in othes at once.

As you know, there are three ways of inserting a style sheet:

- External style sheet (.css file)
- Internal style sheet (using the *<style>* tag in the head section of an HTML/XHTML/JSP page)
- Inline style (using style attribute)

Using CSS Editing Perspective you can change your style sheet, inserted in any of the possible places described before, in three ways:

- directly in your Editor
- using *CSS Properties view*
- using *Properties view*


Pack
Web
Sea
src/main
JBoss 4.2 Runtime [JBoss 4.
Web App Libraries
JRE System Library [java-1.
build
resources
src
WebContent
img
layout
META-INF
stylesheet
theme.css
WEB-INF
Customers.page.xml
*theme.css
Outline
.tableControl
.footer
text-align
font-size
.rich-table
h1
font-family
color
font-size
margin-top
body
.body
.columnHeader:hover
Properties
Property
Value
Aural
azimuth
cue
cue-after
cue-before
elevation
CSS Properties
Text/Font
Background
Boxes
Property Sheet
Edited Properties
Font Style:
Font Weight:
Text Decoration:
Text Align:
CSS Pre
JBoss S

```
    text-align: right;
}

.footer {
    text-align: center;
    font-size: 10px;
}

.rich-table {
    width:100%;
}

h1 {
    font-family: Arial,sans-serif;
    color: #578BB8 ;
    font-size: 1.6em;
    margin-top: 0;
}

body {
```

Font Family: Arial,sans-serif

Color: #578BB8

Font Size: 1.6 em

Double click to edit preview.

Figure 5.1. CSS Editing Perspective

5.1. Outline view

Using this view you can easily skip between the selectors described in the source files, see the list of properties in any selector just by clicking the triangle near it.

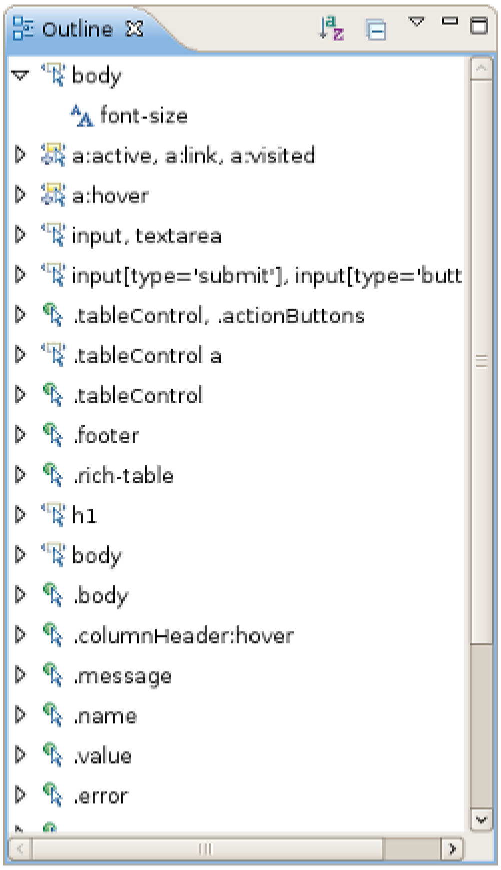
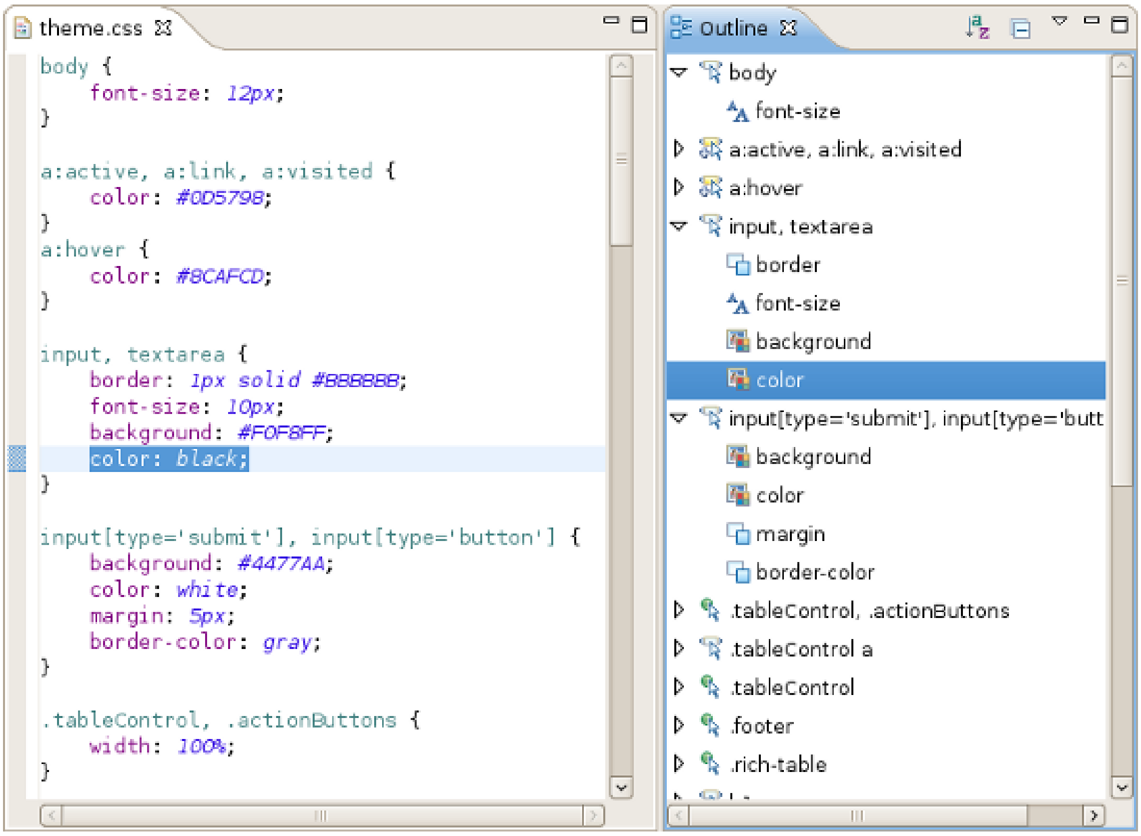

Outline
body
font-size
a:active, a:link, a:visited
a:hover
input, textarea
input[type='submit'], input[type='butt
.tableControl, .actionButtons
.tableControl a
.tableControl
.footer
.rich-table
h1
body
.body
.columnHeader:hover
.message
.name
.value
.error

Figure 5.2. Outline view

You can use the Source viewer with the Outline view to navigate around the file. To do this you should just left click the selector or property you want and it will be automatically highlighted in the source code:



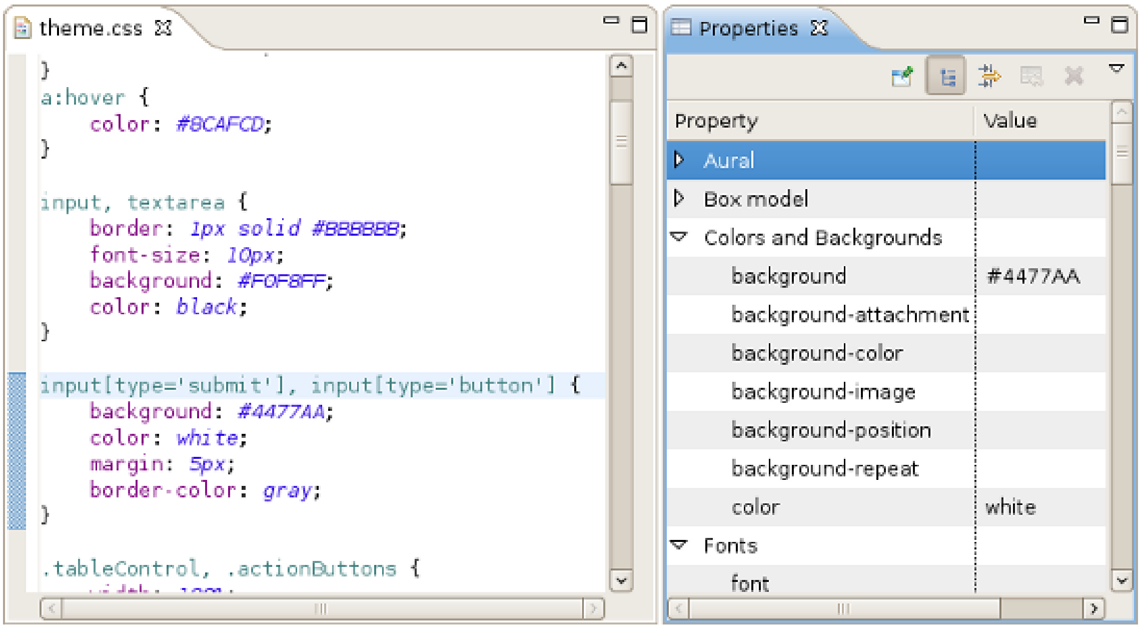
theme.css
Outline
body
font-size
a:active, a:link, a:visited
a:hover
input, textarea
border
font-size
background
color
input[type='submit'], input[type='butt
background
color
margin
border-color
.tableControl, .actionButtons
.tableControl a
.tableControl
.footer
.rich-table

```
body {
    font-size: 12px;
}

a:active, a:link, a:visited {
    color: #0D5798;
}
a:hover {
    color: #8CAFCD;
}

input, textarea {
    border: 1px solid #BBBBBB;
    font-size: 10px;
    background: #F0F8FF;
    color: black;
}

input[type='submit'], input[type='button'] {
    background: #4477AA;
    color: white;
    margin: 5px;
    border-color: gray;
}

.tableControl, .actionButtons {
    width: 100%;
}
```

Figure 5.3. Navigating around the file

5.2. Properties view

Properties view provides a full list of properties of a chosen selector. The properties are divided into logic groups for better navigation.

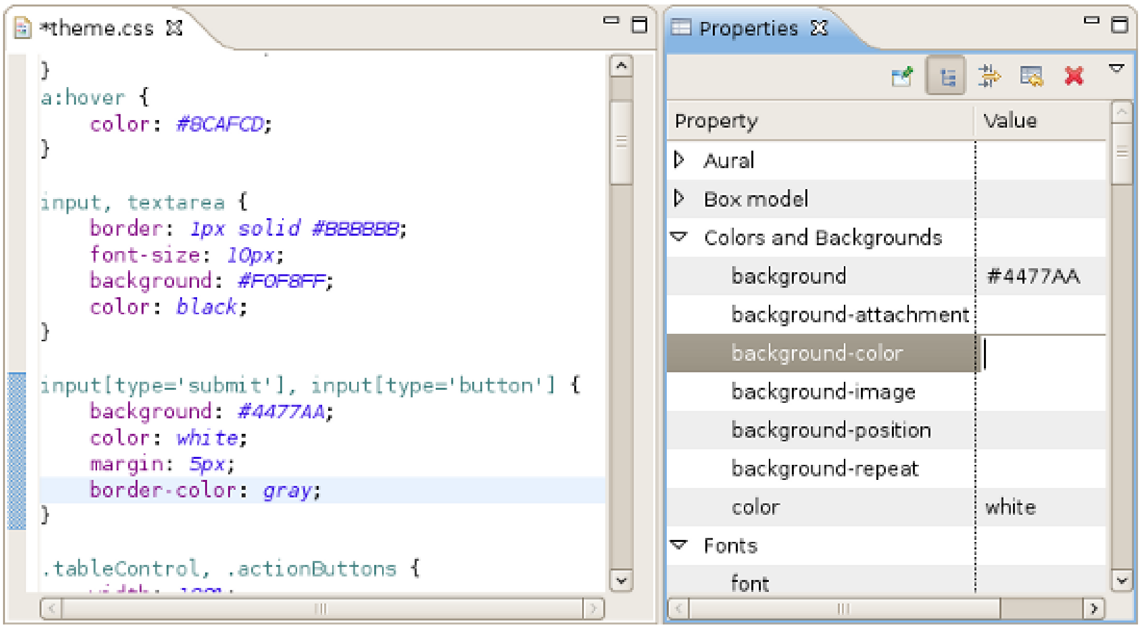


```
}
a:hover {
    color: #8CAFCD;
}

input, textarea {
    border: 1px solid #BBBBBB;
    font-size: 10px;
    background: #F0F8FF;
    color: black;
}

input[type='submit'], input[type='button'] {
    background: #4477AA;
    color: white;
    margin: 5px;
    border-color: gray;
}

.tableControl, .actionButtons {
```

Property	Value
Aural	
Box model	
Colors and Backgrounds	
background	#4477AA
background-attachment	
background-color	
background-image	
background-position	
background-repeat	
color	white
Fonts	
font	

Figure 5.4. Properties view

With the help of Properties view you have also the possibility to edit the css file by adding/editing/removing properties in the selector. Left click the "*Value*" field near the property you want to edit and write the changes in the text field.



```
}
a:hover {
    color: #8CAFCD;
}

input, textarea {
    border: 1px solid #BBBBBB;
    font-size: 10px;
    background: #F0F8FF;
    color: black;
}

input[type='submit'], input[type='button'] {
    background: #4477AA;
    color: white;
    margin: 5px;
    border-color: gray;
}

.tableControl, .actionButtons {
```

Property	Value
Aural	
Box model	
Colors and Backgrounds	
background	#4477AA
background-attachment	
background-color	
background-image	
background-position	
background-repeat	
color	white
Fonts	
font	

Figure 5.5. Updating css using Properties view

5.3. CSS Properties view

CSS Properties view has five tabs:

- *Text/Font [117]*
- *Background [118]*
- *Boxes [118]*
- *Property Sheet [119]*
- *Edited Properties [119]*

- CSS Text/Font properties define the appearance of text,its font family, boldness, size, and the style.

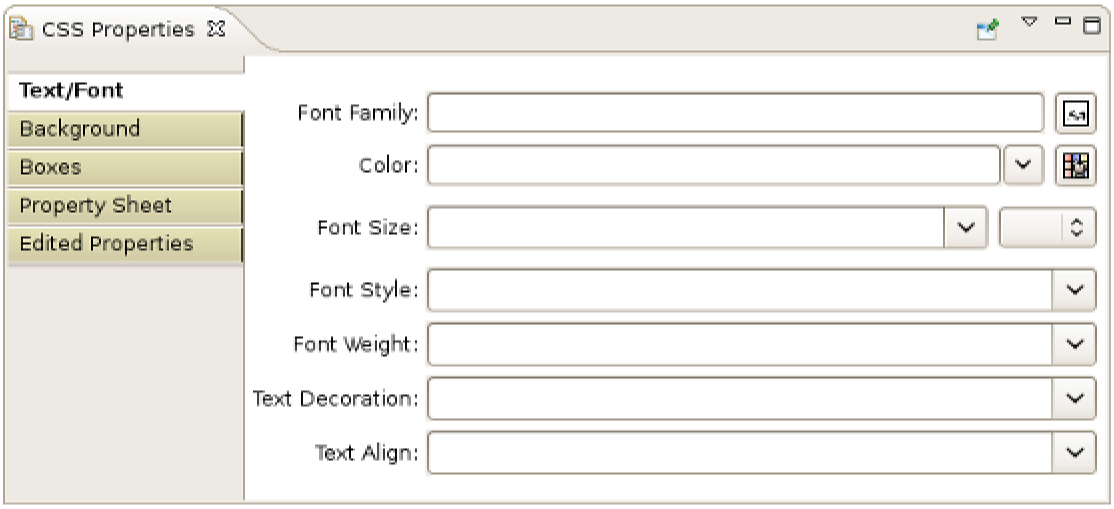

CSS Properties
Text/Font
Background
Boxes
Property Sheet
Edited Properties
Font Family:
Color:
Font Size:
Font Style:
Font Weight:
Text Decoration:
Text Align:

Figure 5.6. Text/Font tab

For example, to define the "*font-family*" property you should click *Choose font family* button( ) near *Font Family* text field and select the fonts you want to use from the list.

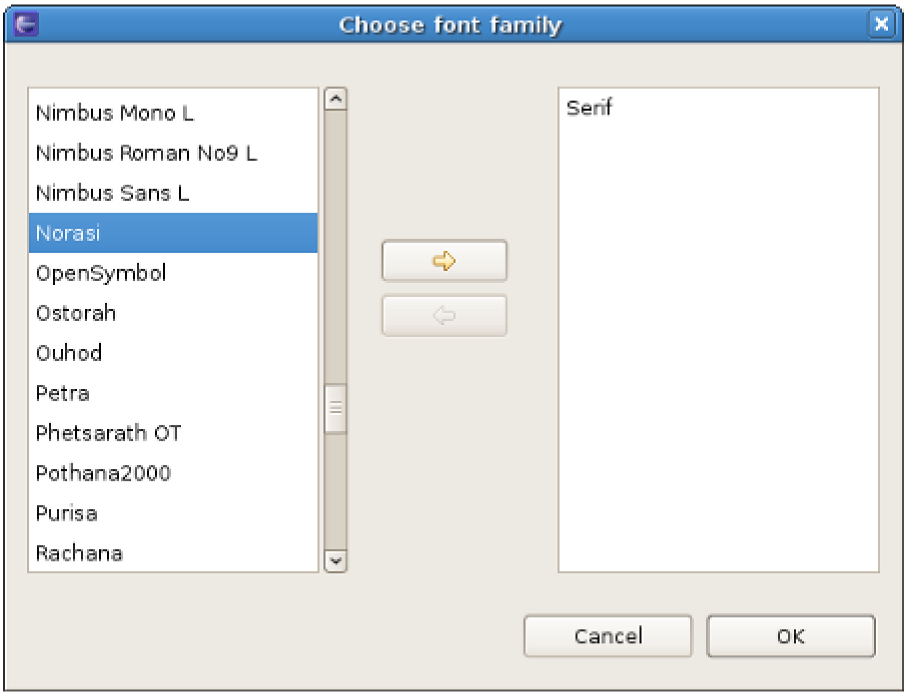

Choose font family
Nimbus Mono L
Nimbus Roman No9 L
Nimbus Sans L
Norasi
OpenSymbol
Ostorah
Ouhod
Petra
Phetsarath OT
Pothana2000
Purisa
Rachana
Serif
Cancel
OK

Figure 5.7. CSS Text/Font tab

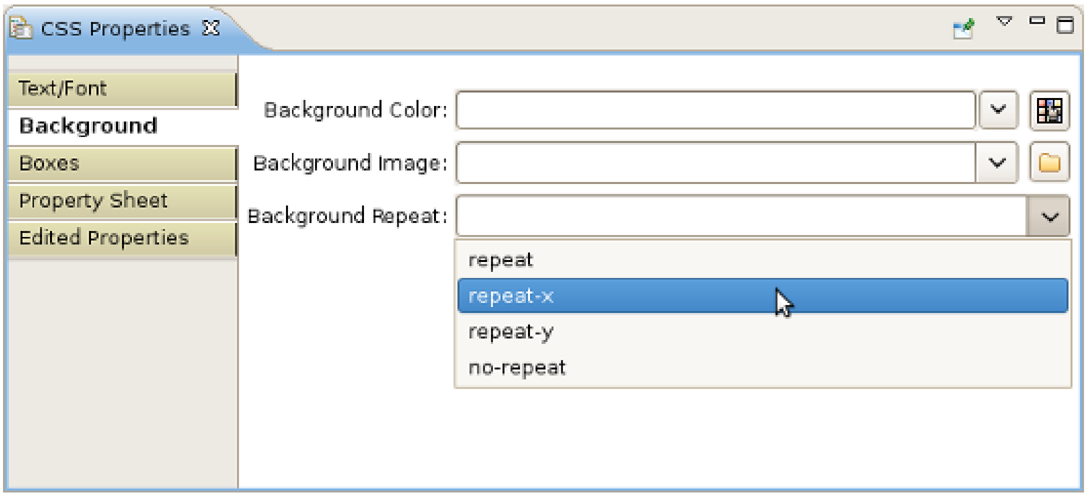
When you click *Ok* the chosen fonts should appear in *Font Family* text field and in the sourse css file. To define other properties in CSS *Text/Font* tab you should just click button near the corresponding field you want and select the appropriate option in the list. Or if you are absolutely sure of the property's value to use you can just write it in the text field.

- You should use CSS background properties and *Background* tab to define the background effects of an element.

Boxes tab is used to define CSS border properties and the box model. The CSS border properties allow you to specify the style and color of an element's border.

As well as in Text/Font tab, it's also possible to define the property in two ways:

- clicking ⌄ and choosing it from the list of options:


CSS Properties
Text/Font
Background
Boxes
Property Sheet
Edited Properties

Background Color:

Background Image:

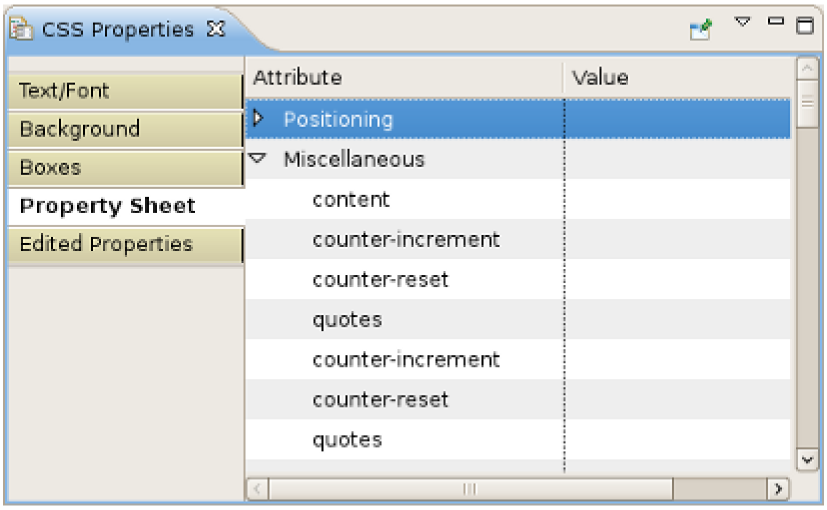
Background Repeat:

repeat
repeat-x
repeat-y
no-repeat

Figure 5.8. Defining the property

- writing the property in the appropriate text field

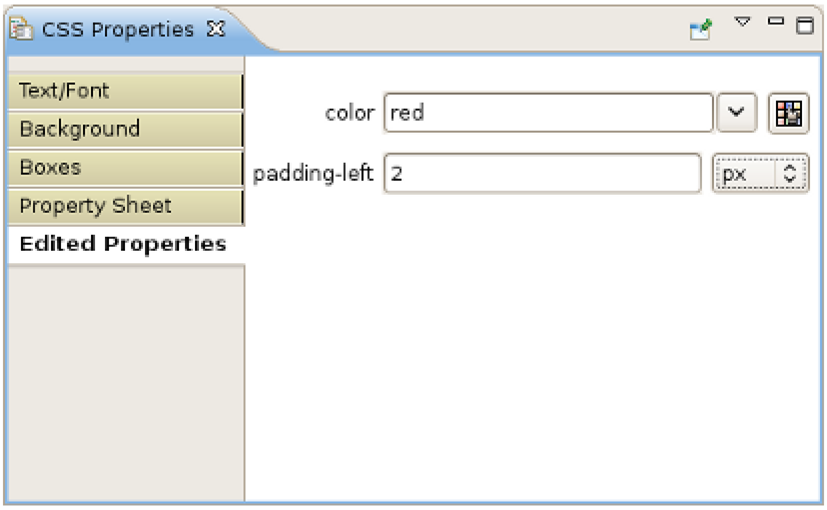
- *Property Sheet* tab contains the categorized list of properties. Like in *Properties view* it's possible to edit the properties values.


CSS Properties
Text/Font
Background
Boxes
Property Sheet
Edited Properties

Attribute	Value
Positioning	
Miscellaneous	
content	
counter-increment	
counter-reset	
quotes	
counter-increment	
counter-reset	
quotes	

Figure 5.9. Property Sheet tab

- *Edited Properties* tab contains only the properties,defined in the selector.


CSS Properties
Text/Font
Background
Boxes
Property Sheet
Edited Properties

color red

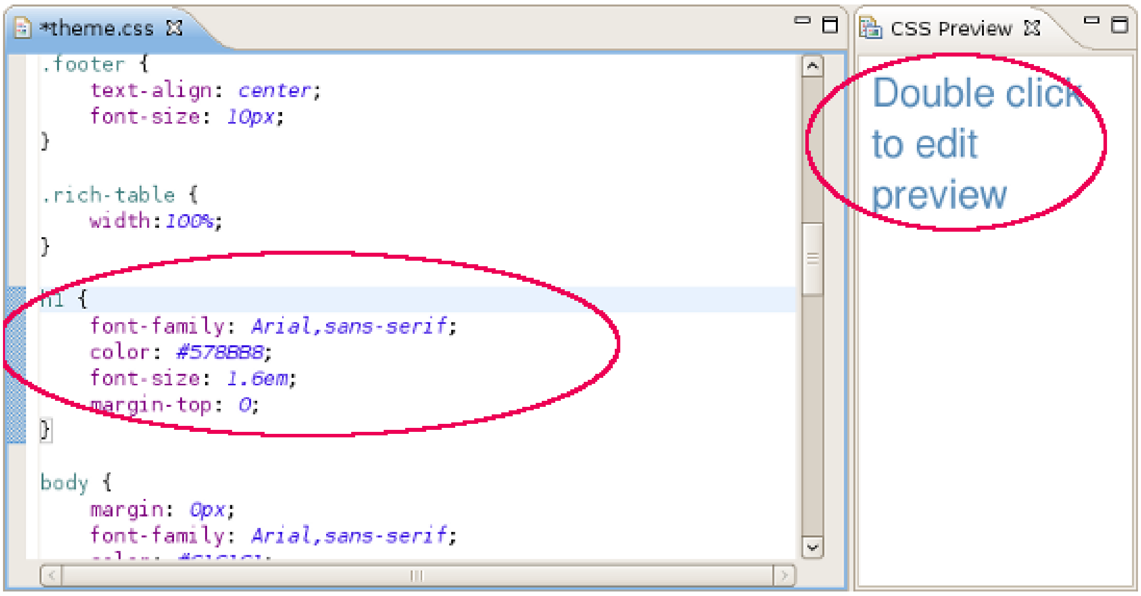
padding-left 2 px

Figure 5.10. Property Sheet tab

It's also possible to edit the properties in the tab.

5.4. CSS Preview

Using CSS Preview you can see how a selector affects any text.


*theme.css
CSS Preview

```
.footer {
    text-align: center;
    font-size: 10px;
}

.rich-table {
    width:100%;
}

h1 {
    font-family: Arial,sans-serif;
    color: #578BB8;
    font-size: 1.6em;
    margin-top: 0;
}

body {
    margin: 0px;
    font-family: Arial,sans-serif;
```

Double click to edit preview

Figure 5.11. Property Sheet tab

The preview is also edited by double click.You can write instead of the default text any text you want,including html tags.

RichFaces Support

JBoss Developer Studio comes with a tight integration with *RichFaces component framework*.

Note:

RichFaces 3.3.X is fully supported in the current version of JBoss Developer Studio (i. e. 2.1.0.GA) and JBoss Tools 3.0.1.GA.

The following features are implemented and fully supported for the current version of the RichFaces components:

- *Content Assist*
- *OpenOn*
- *Representing in JBoss Tools Palette*

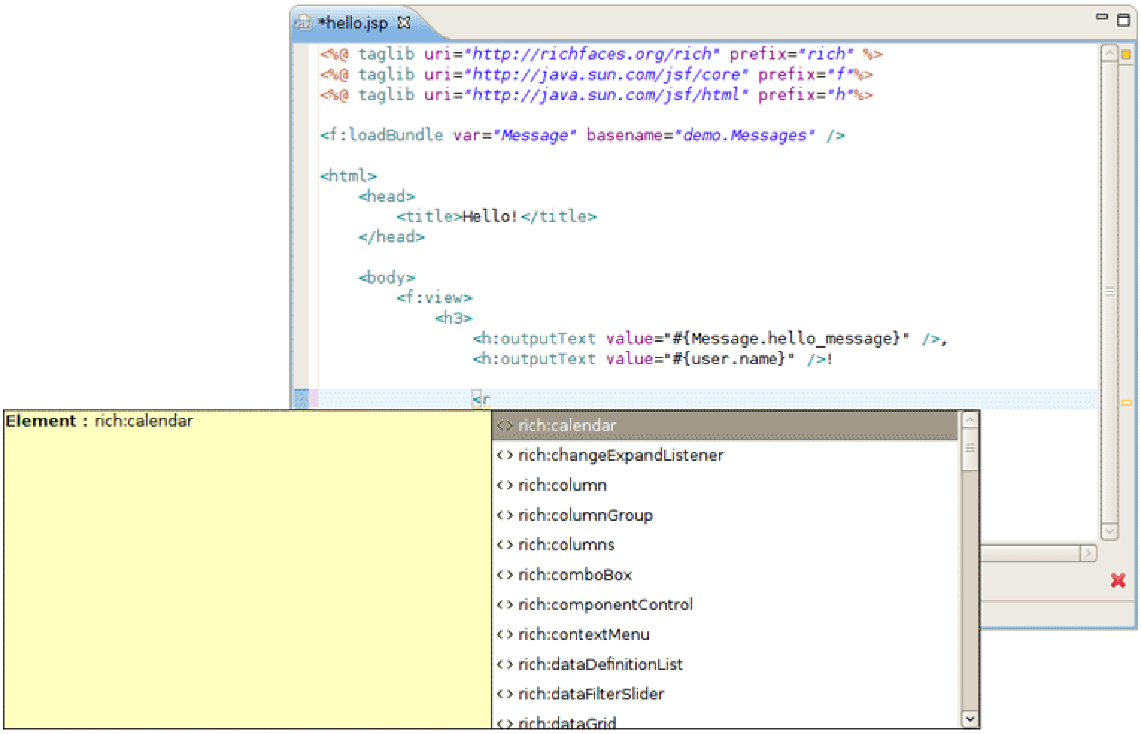
All you have to do is to *download* and install RichFaces libraries into your project, i. e. just put *richfaces-*.jar* files into the */lib* project folder. Also how to get started with RichFaces you can find in *RichFaces documentation*.

6.1. Code Assist for RichFaces

JBDS/JBoss Tools indeed provide code completion for *RichFaces* framework components.

Tip:

RichFaces 3.3.X is now fully supported in code completion.


*hello.jsp
Element : rich:calendar
rich:calendar
rich:changeExpandListener
rich:column
rich:columnGroup
rich:columns
rich:comboBox
rich:componentControl
rich:contextMenu
rich:dataDefinitionList
rich:dataFilterSlider
rich:dataGrid

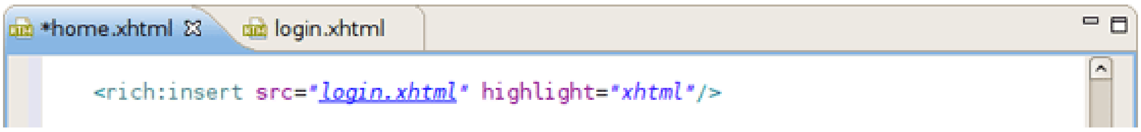
```
<%@ taglib uri="http://richfaces.org/rich" prefix="rich" %>
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f"%>
<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h"%>

<f:loadBundle var="Message" basename="demo.Messages" />

<html>
    <head>
        <title>Hello!</title>
    </head>

    <body>
        <f:view>
            <h3>
                <h:outputText value="#{Message.hello_message}" />,
                <h:outputText value="#{user.name}" />!

                <r
```

Figure 6.1. Content Assist for RichFaces Components

6.2. OpenOn for RichFaces

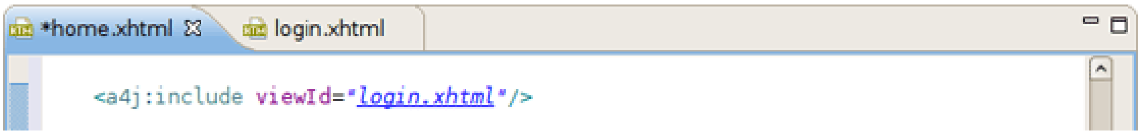
Working with *.jsp/.xhtml* pages in VPE you can also take the advantage of *OpenOn* feature for the RichFaces components.

For example, Richfaces tags `<rich:insert>` and `<a4j:include>>` has OpenOn support.


*home.xhtml
login.xhtml

```
<rich:insert src="login.xhtml" highlight="xhtml"/>
```

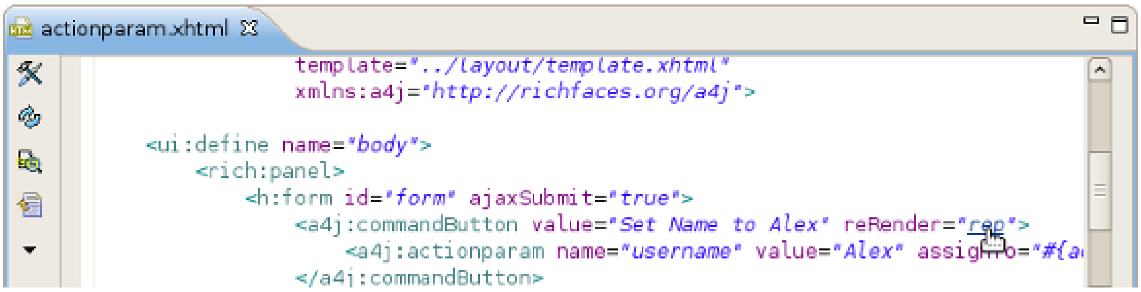
Figure 6.2. OpenOn With Richfaces Tag


*home.xhtml
login.xhtml

```
<a4j:include viewId="login.xhtml"/>
```

Figure 6.3. OpenOn With A4j Tag

OpenOn is also supported in "ForID"-like attributes (the attributes, where the value should be ID or the list of IDs) in RichFaces.


actionparam.xhtml

```
                template="../layout/template.xhtml"
                xmlns:a4j="http://richfaces.org/a4j">

    <ui:define name="body">
        <rich:panel>
            <h:form id="form" ajaxSubmit="true">
                <a4j:commandButton value="Set Name to Alex" reRender="rep">
                    <a4j:actionparam name="username" value="Alex" assignTo="#{a
                </a4j:commandButton>
```

Figure 6.4. OpenOn With "ForID"-like attributes

6.3. RichFaces in the JBoss Tools Palette

RichFaces and *JBoss Tools Palette*.

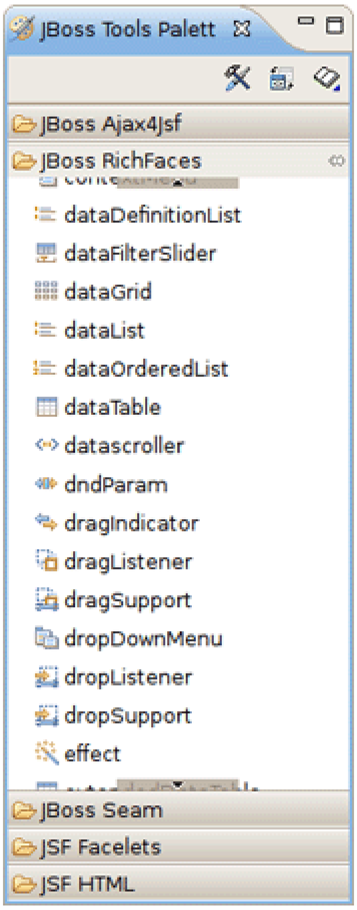
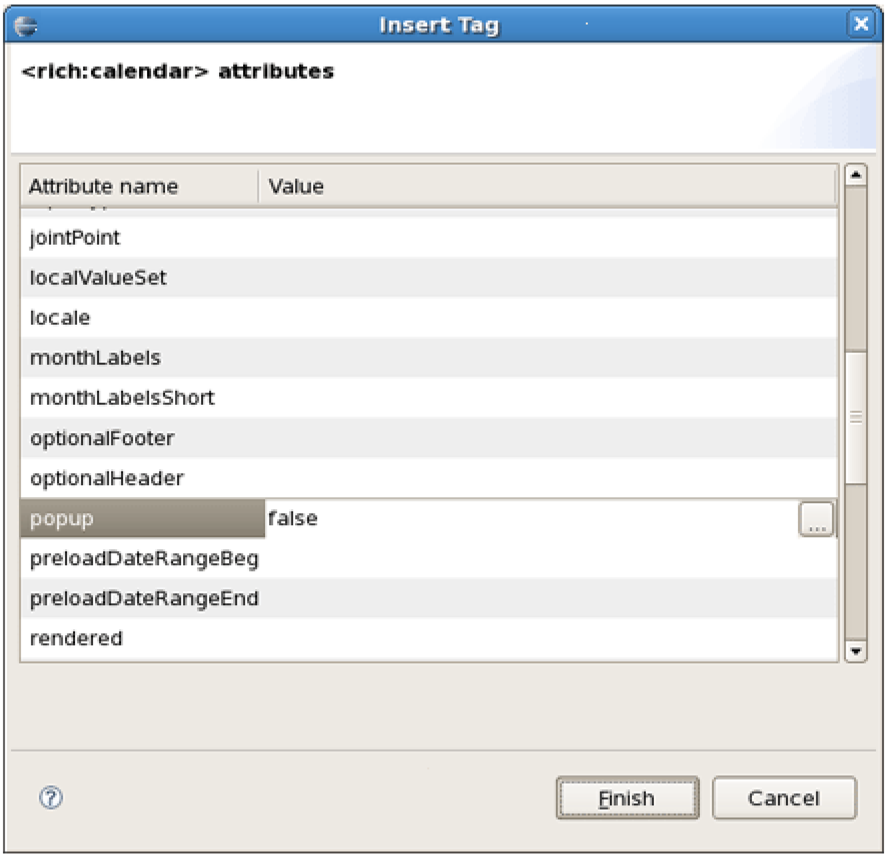

JBoss Tools Palett
JBoss Ajax4Jsf
JBoss RichFaces
dataDefinitionList
dataFilterSlider
dataGrid
dataList
dataOrderedList
dataTable
datascroller
dndParam
dragIndicator
dragListener
dragSupport
dropDownMenu
dropListener
dropSupport
effect
JBoss Seam
JSF Facelets
JSF HTML

Figure 6.5. 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

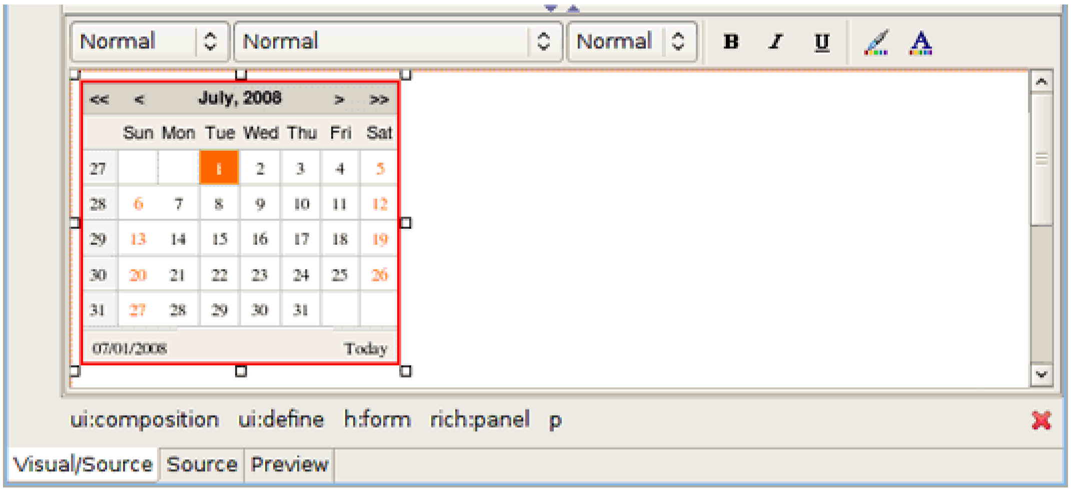

Insert Tag
Finish
Cancel

<rich:calendar> attributes

Attribute name	Value
jointPoint	
localValueSet	
locale	
monthLabels	
monthLabelsShort	
optionalFooter	
optionalHeader	
popup	false
preloadDateRangeBeg	
preloadDateRangeEnd	
rendered	

Figure 6.6. Inserting Tag

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


Normal
Normal
Normal
B
I
U
ui:composition
ui:define
h:form
rich:panel
p
Visual/Source
Source
Preview

<< < July, 2008 > >>

	Sun	Mon	Tue	Wed	Thu	Fri	Sat
27			1	2	3	4	5
28	6	7	8	9	10	11	12
29	13	14	15	16	17	18	19
30	20	21	22	23	24	25	26
31	27	28	29	30	31		

07/01/2008 Today

Figure 6.7. RichFaces Component

6.4. Relevant Resources Links

To get more in-depth information on RichFaces framework refer to *RichFaces Developer Guide*.

It may be also helpful for you to look through the *movies* where there are ones that demonstrate the usage of 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, Struts and Seam 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*

7.1. 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.

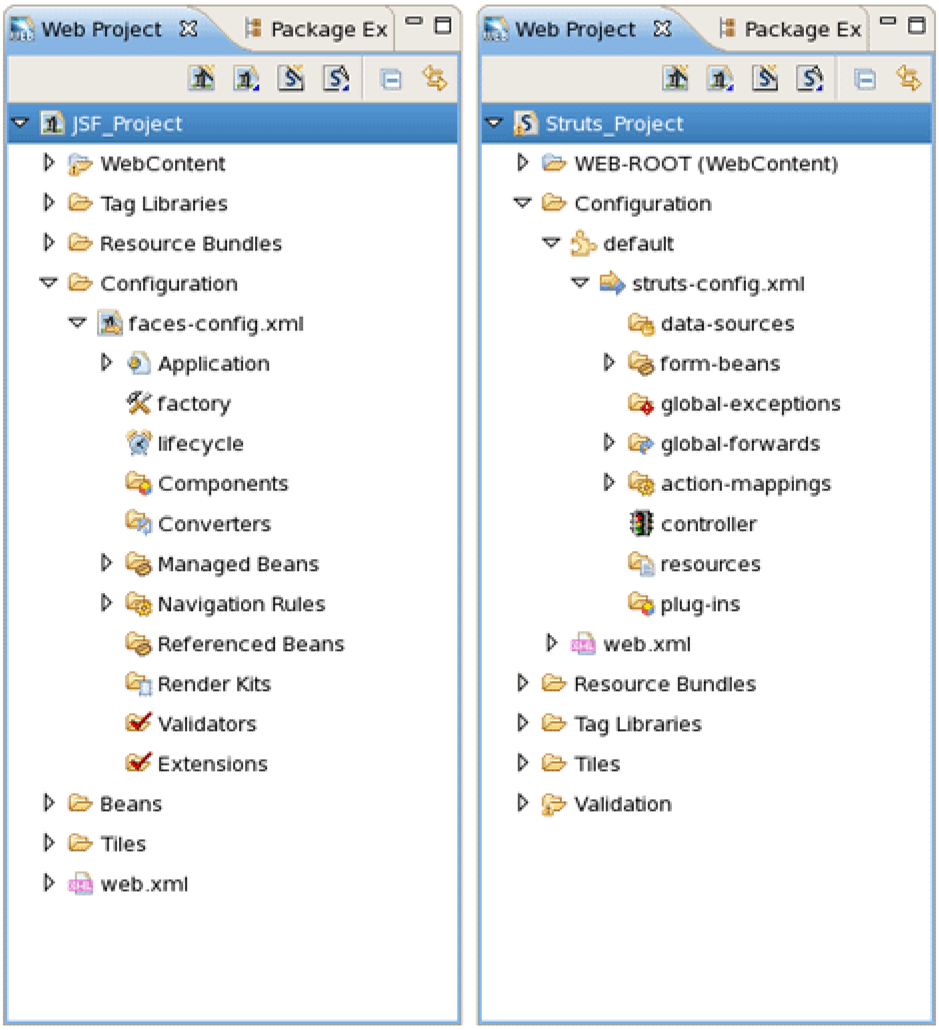

Web Project
Package Ex
JSF_Project
WebContent
Tag Libraries
Resource Bundles
Configuration
faces-config.xml
Application
factory
lifecycle
Components
Converters
Managed Beans
Navigation Rules
Referenced Beans
Render Kits
Validators
Extensions
Beans
Tiles
web.xml
Web Project
Package Ex
Struts_Project
WEB-ROOT (WebContent)
Configuration
default
struts-config.xml
data-sources
form-beans
global-exceptions
global-forwards
action-mappings
controller
resources
plug-ins
web.xml
Resource Bundles
Tag Libraries
Tiles
Validation

Figure 7.1. Web Projects View

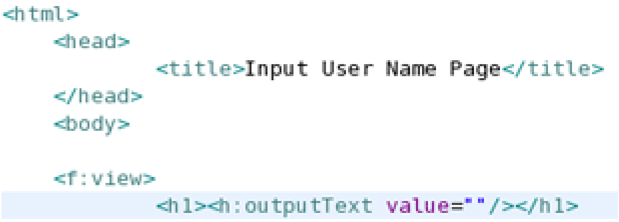
7.2. Drag and Drop

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.

7.2.1. 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>
            <h1><h:outputText value=""/></h1>
```

Figure 7.2. OutputText Tag

Select the property:

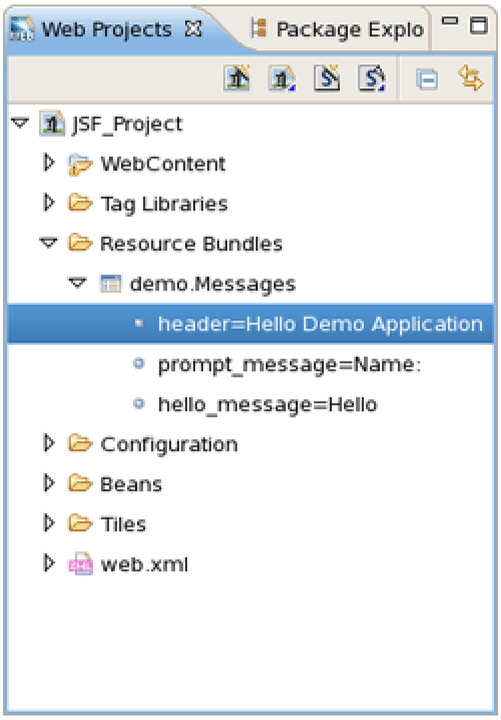
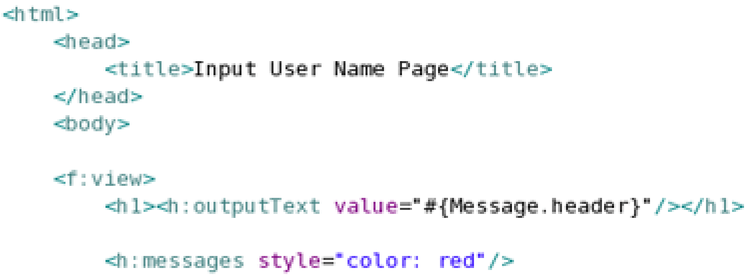

Web Projects
Package Explo
JSF_Project
WebContent
Tag Libraries
Resource Bundles
demo.Messages
header=Hello Demo Application
prompt_message=Name:
hello_message=Hello
Configuration
Beans
Tiles
web.xml

Figure 7.3. 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>
        <h1><h:outputText value="#{Message.header}"/></h1>

        <h:messages style="color: red"/>
```

Figure 7.4. Inserted Property

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

7.2.2. 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.

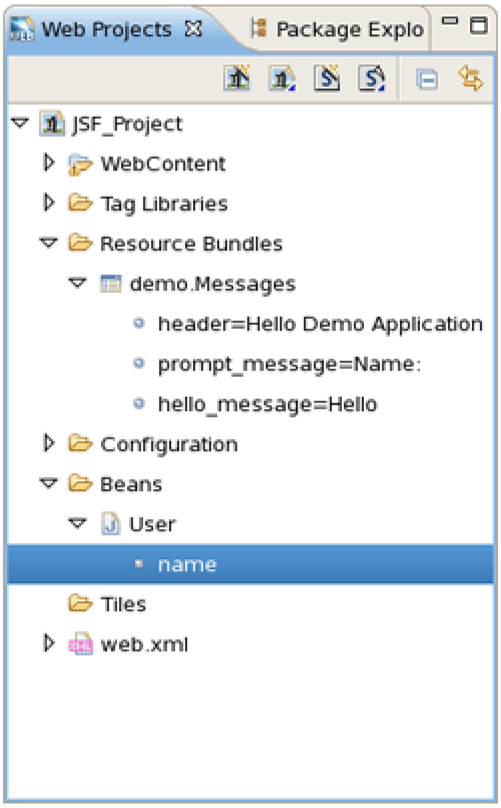
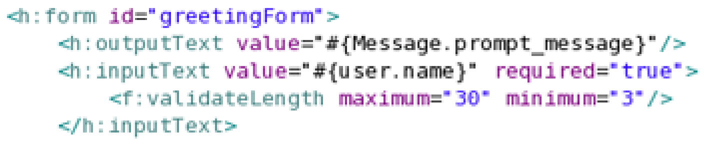

Web Projects
Package Explo
JSF_Project
WebContent
Tag Libraries
Resource Bundles
demo.Messages
header=Hello Demo Application
prompt_message=Name:
hello_message=Hello
Configuration
Beans
User
name
Tiles
web.xml

Figure 7.5. 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 7.6. Added Expression

7.2.3. Navigation Rules

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

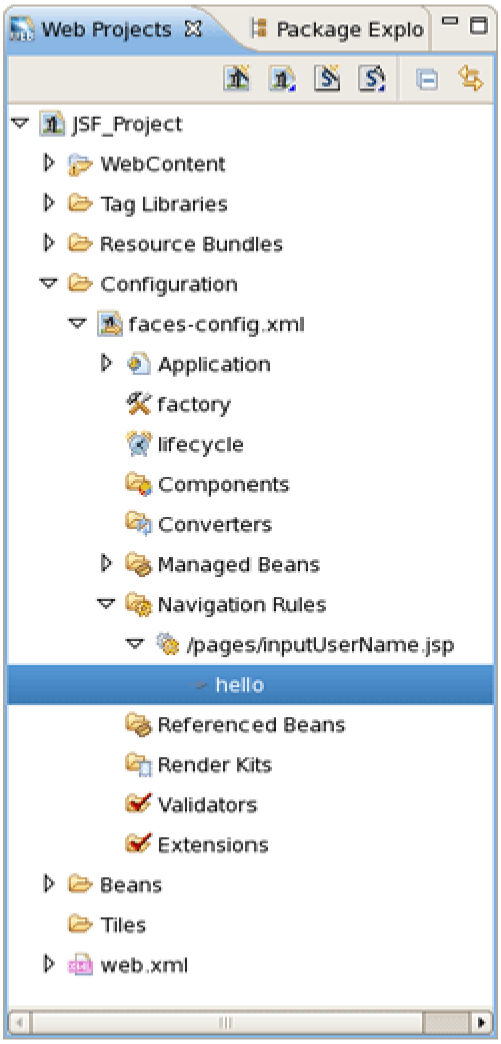
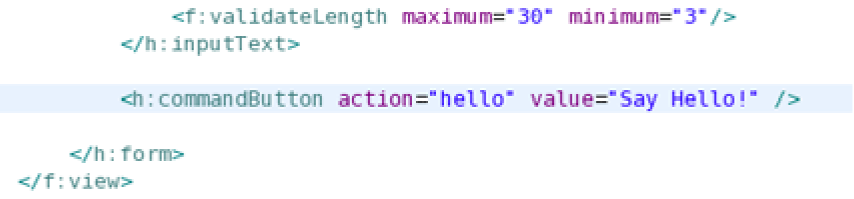

Web Projects
Package Explo
JSF_Project
WebContent
Tag Libraries
Resource Bundles
Configuration
faces-config.xml
Application
factory
lifecycle
Components
Converters
Managed Beans
Navigation Rules
/pages/inputUserName.jsp
hello
Referenced Beans
Render Kits
Validators
Extensions
Beans
Tiles
web.xml

Figure 7.7. Selecting Navigation Rule

Drag and drop it inside the commandButton tag:



```
            <f:validateLength maximum="30" minimum="3"/>
        </h:inputText>

        <h:commandButton action="hello" value="Say Hello!" />

    </h:form>
</f:view>
```

Figure 7.8. Navigation Rule in CommandButton Tag

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

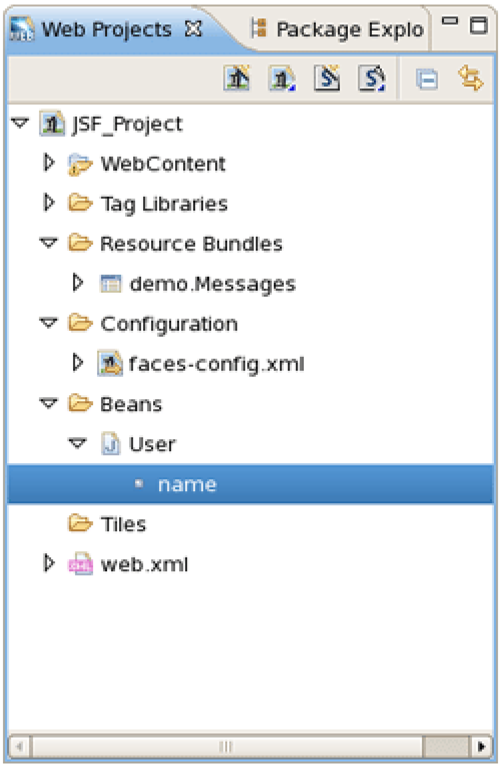
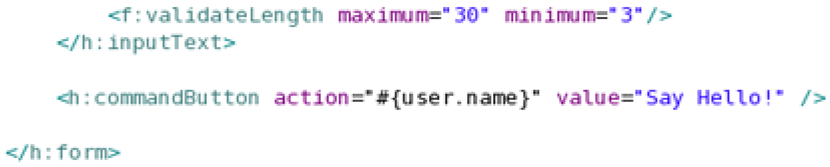

Web Projects
Package Explo
JSF_Project
WebContent
Tag Libraries
Resource Bundles
demo.Messages
Configuration
faces-config.xml
Beans
User
name
Tiles
web.xml

Figure 7.9. 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 7.10. Inserted Navigation Rule

7.2.4. For a Tag Library File Declaration

Select a TLD file:

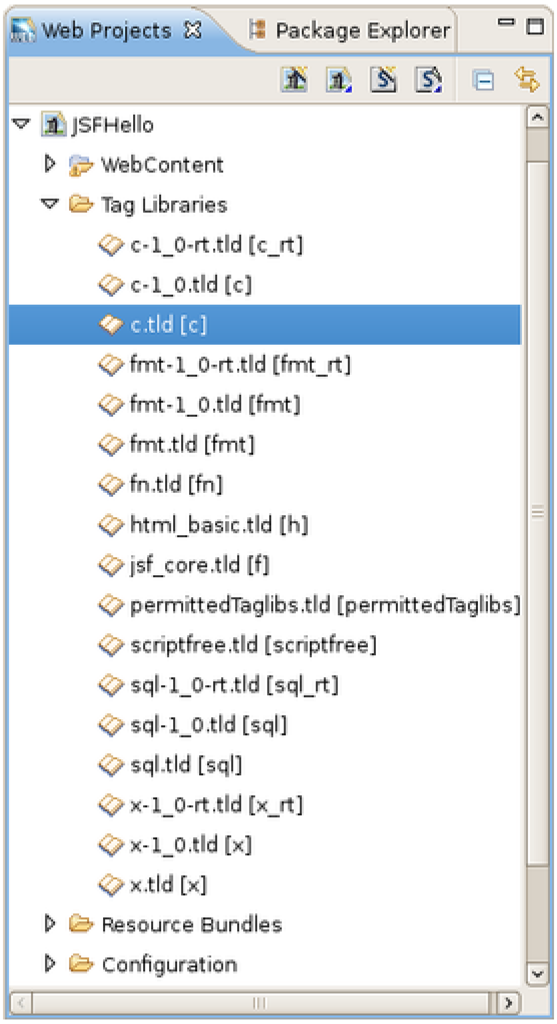
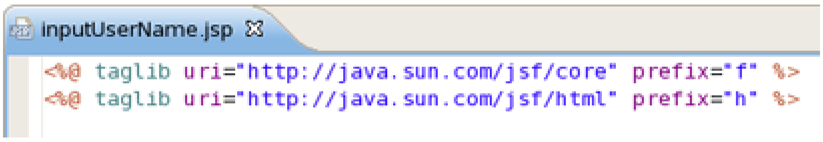

Web Projects
Package Explorer
JSFHello
WebContent
Tag Libraries
c-1_0-rt.tld [c_rt]
c-1_0.tld [c]
c.tld [c]
fmt-1_0-rt.tld [fmt_rt]
fmt-1_0.tld [fmt]
fmt.tld [fmt]
fn.tld [fn]
html_basic.tld [h]
jsf_core.tld [f]
permittedTaglibs.tld [permittedTaglibs]
scriptfree.tld [scriptfree]
sql-1_0-rt.tld [sql_rt]
sql-1_0.tld [sql]
sql.tld [sql]
x-1_0-rt.tld [x_rt]
x-1_0.tld [x]
x.tld [x]
Resource Bundles
Configuration

Figure 7.11. Selecting TLD File

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

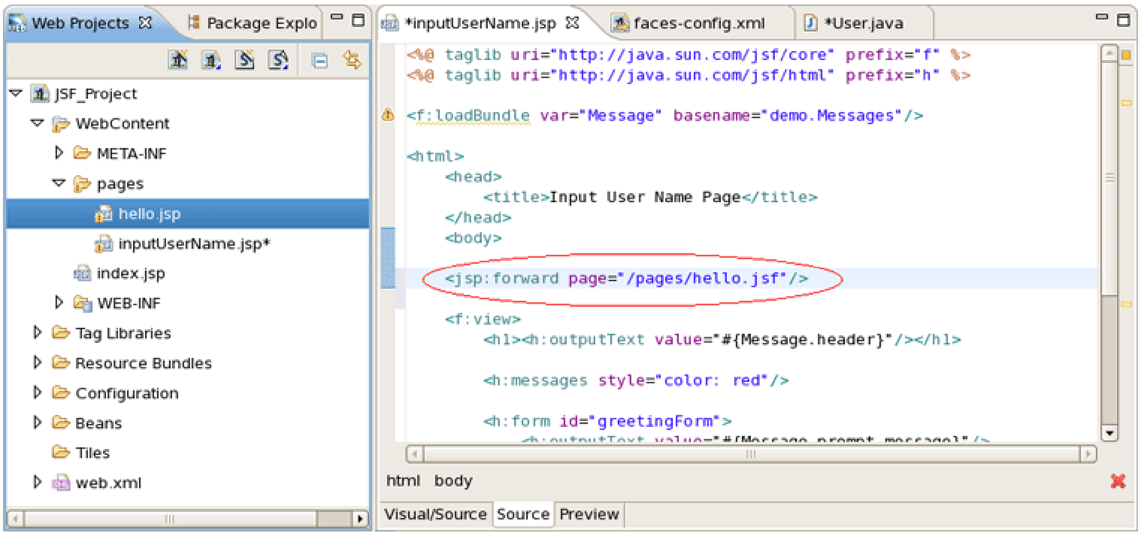

inputUserName.jsp

```
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>
<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>
```

Figure 7.12. Inserted TLD File

7.2.5. For JSP Pages

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


Web Projects
Package Explo
JSF_Project
WebContent
META-INF
pages
hello.jsp
inputUserName.jsp*
index.jsp
WEB-INF
Tag Libraries
Resource Bundles
Configuration
Beans
Tiles
web.xml
*inputUserName.jsp
faces-config.xml
*User.java
html
body
Visual/Source
Source
Preview

```
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>
<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>

<f:loadBundle var="Message" basename="demo.Messages"/>

<html>
    <head>
        <title>Input User Name Page</title>
    </head>
    <body>

    <jsp:forward page="/pages/hello.jsf"/>

    <f:view>
        <h1><h:outputText value="#{Message.header}"/></h1>

        <h:messages style="color: red"/>

        <h:form id="greetingForm">
```

Figure 7.13. Creating JSP Forward

7.3. 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.

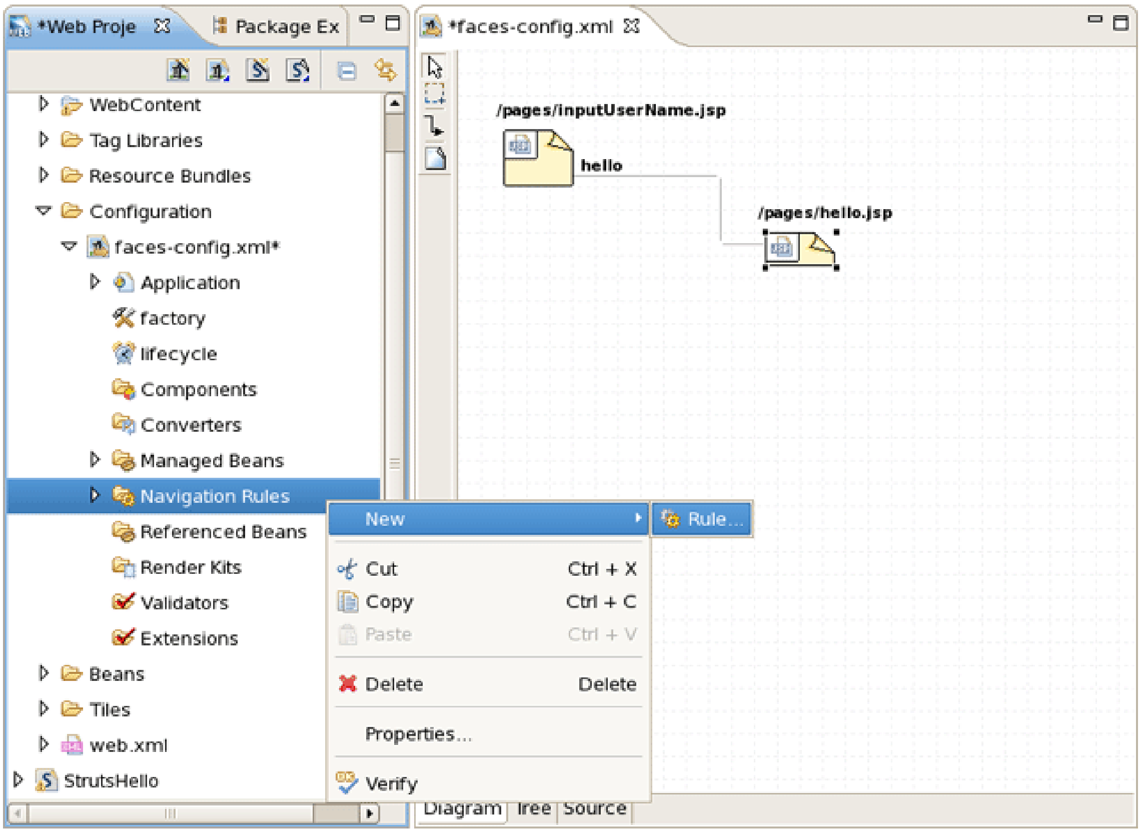

*Web Proje
Package Ex
WebContent
Tag Libraries
Resource Bundles
Configuration
faces-config.xml*
Application
factory
lifecycle
Components
Converters
Managed Beans
Navigation Rules
Referenced Beans
Render Kits
Validators
Extensions
Beans
Tiles
web.xml
StrutsHello
New
Rule...
Cut Ctrl + X
Copy Ctrl + C
Paste Ctrl + V
Delete Delete
Properties...
Verify
*faces-config.xml
/pages/inputUserName.jsp
hello
/pages/hello.jsp
Diagram
Tree
Source

Figure 7.14. Creating New Navigation Rule

7.4. 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*:

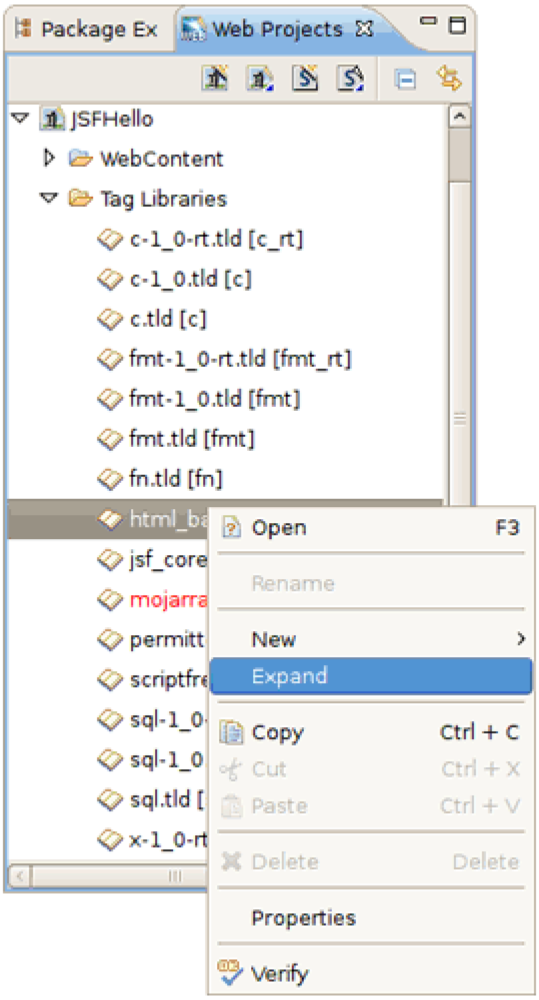

Package Ex
Web Projects
JSFHello
WebContent
Tag Libraries
c-1_0-rt.tld [c_rt]
c-1_0.tld [c]
c.tld [c]
fmt-1_0-rt.tld [fmt_rt]
fmt-1_0.tld [fmt]
fmt.tld [fmt]
fn.tld [fn]
html_ba
jsf_core
mojarra
permitt
scriptfre
sql-1_0
sql-1_0
sql.tld [
x-1_0-rt
Open F3
Rename
New
Expand
Copy Ctrl + C
Cut Ctrl + X
Paste Ctrl + V
Delete Delete
Properties
Verify

Figure 7.15. Expanding Tag Library File

The TLD file will now be expanded:

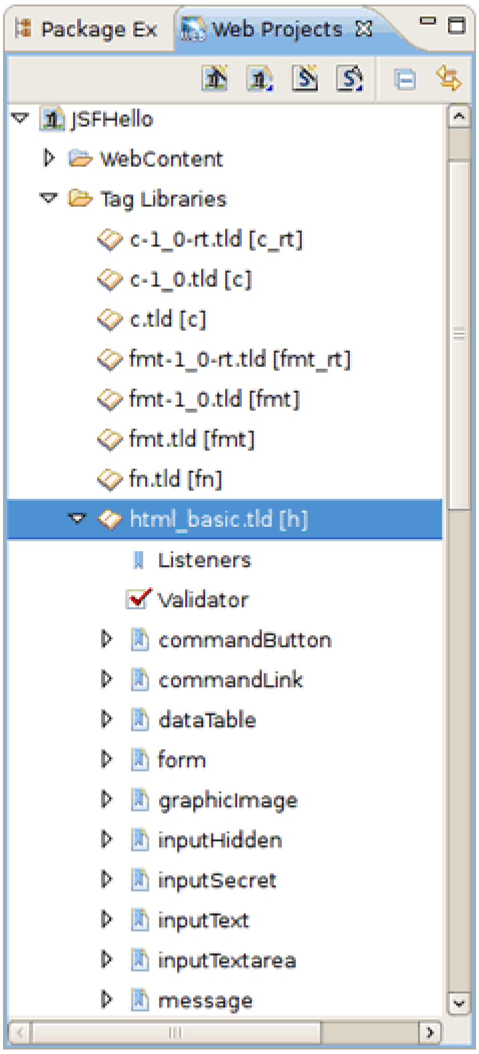

Package Ex
Web Projects
JSFHello
WebContent
Tag Libraries
c-1_0-rt.tld [c_rt]
c-1_0.tld [c]
c.tld [c]
fmt-1_0-rt.tld [fmt_rt]
fmt-1_0.tld [fmt]
fmt.tld [fmt]
fn.tld [fn]
html_basic.tld [h]
Listeners
Validator
commandButton
commandLink
dataTable
form
graphicImage
inputHidden
inputSecret
inputText
inputTextarea
message

Figure 7.16. Expanded File

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

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

Read *Adding Tag Libraries* to learn about this.

7.6. 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

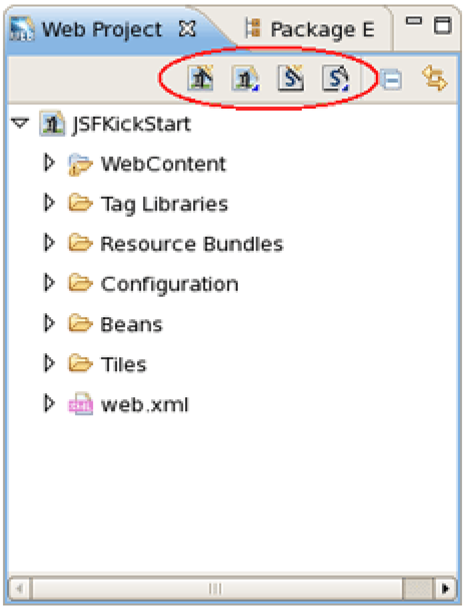

Web Project
Package E
JSFKickStart
WebContent
Tag Libraries
Resource Bundles
Configuration
Beans
Tiles
web.xml

Figure 7.17. 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.

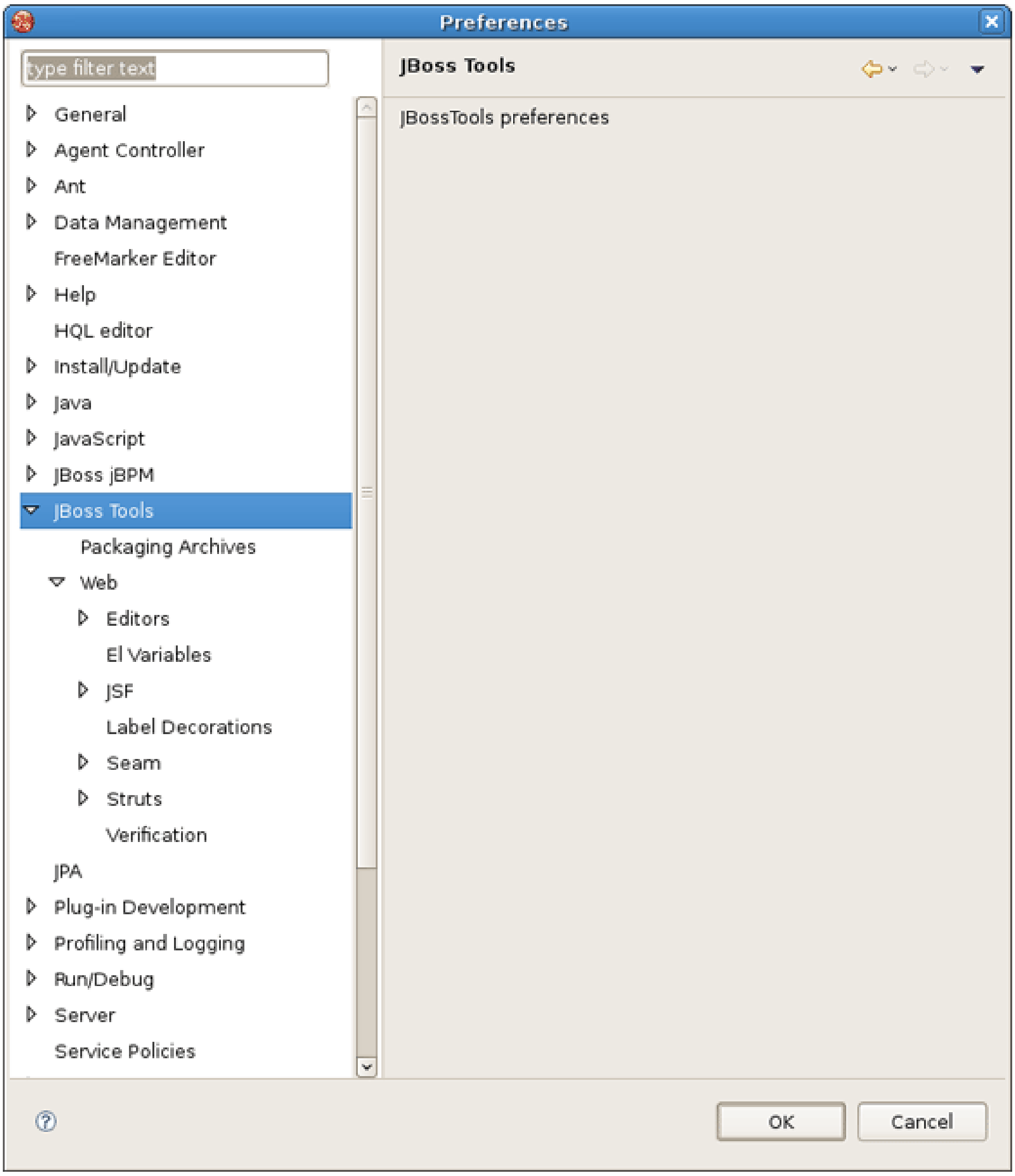

Preferences
type filter text
General
Agent Controller
Ant
Data Management
FreeMarker Editor
Help
HQL editor
Install/Update
Java
JavaScript
JBoss jBPM
JBoss Tools
Packaging Archives
Web
Editors
El Variables
JSF
Label Decorations
Seam
Struts
Verification
JPA
Plug-in Development
Profiling and Logging
Run/Debug
Server
Service Policies
JBoss Tools
JBossTools preferences
OK
Cancel

Figure 8.1. Preferences are included in this dialog.

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

- *Packaging Archives*

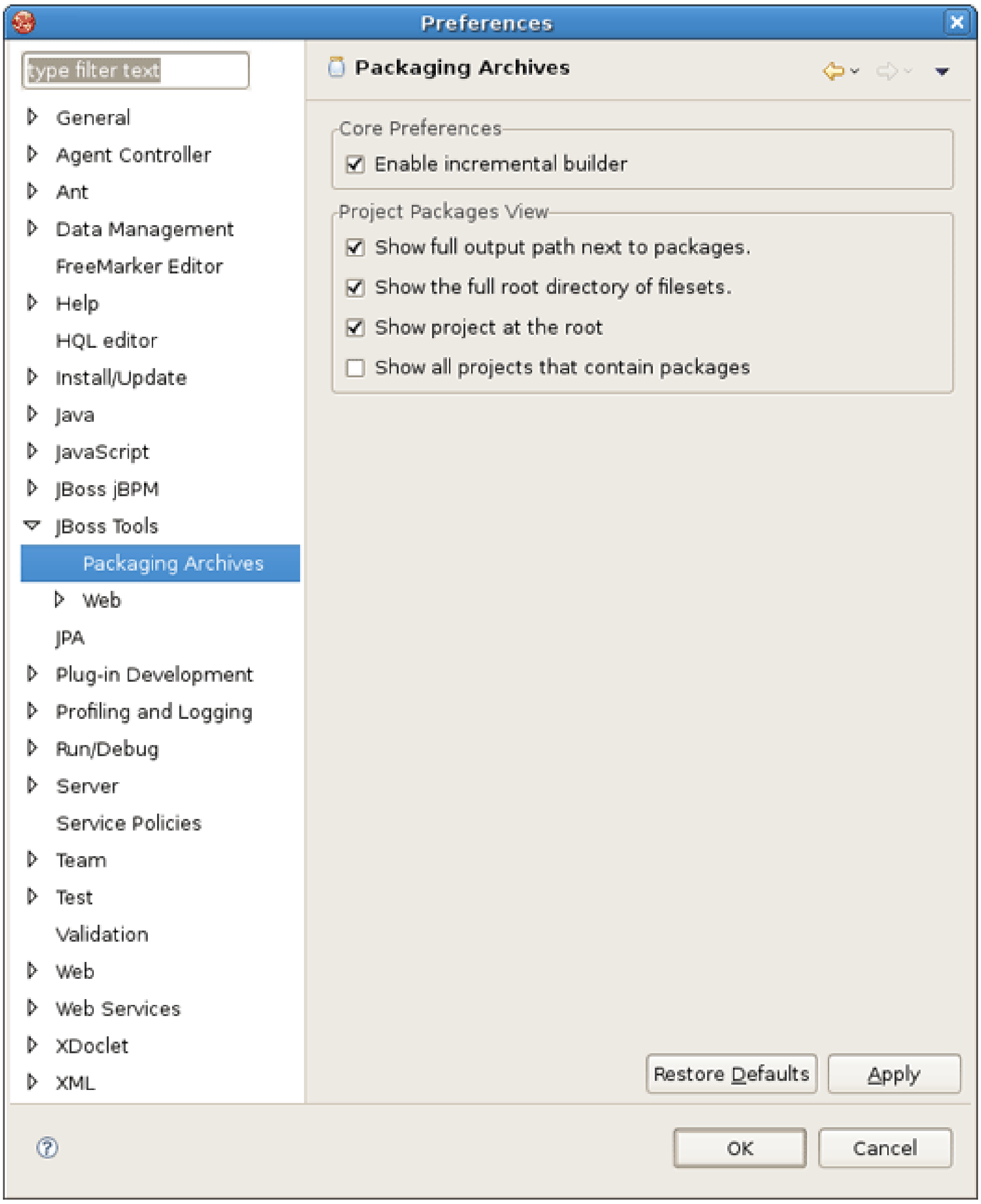
- *Editors*
- *Visual Page Editor*
- *El Variables*
- *JSF*
- *JSF Page*
- *JSF Project*
- *JSF Flow Diagram*
- *Seam*
- *Seam Validator*
- *Struts*
- *Struts Automatic*
- *Plug-in Insets*
- *Resource Insets*
- *Struts Customization*
- *Struts Project*
- *Struts Support*
- *Struts Pages*
- *Struts Flow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

The Preferences dialog *(Window > Preferences)* also allows to adjust settings for *JBoss Server* and *XDoclet* module.

8.1. 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.



Packaging Archives

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

Preferences
type filter text
General
Agent Controller
Ant
Data Management
FreeMarker Editor
Help
HQL editor
Install/Update
Java
JavaScript
JBoss jBPM
JBoss Tools
Packaging Archives
Web
JPA
Plug-in Development
Profiling and Logging
Run/Debug
Server
Service Policies
Team
Test
Validation
Web
Web Services
XDoclet
XML
Restore Defaults
Apply
OK
Cancel

Figure 8.2. Packaging Archives

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

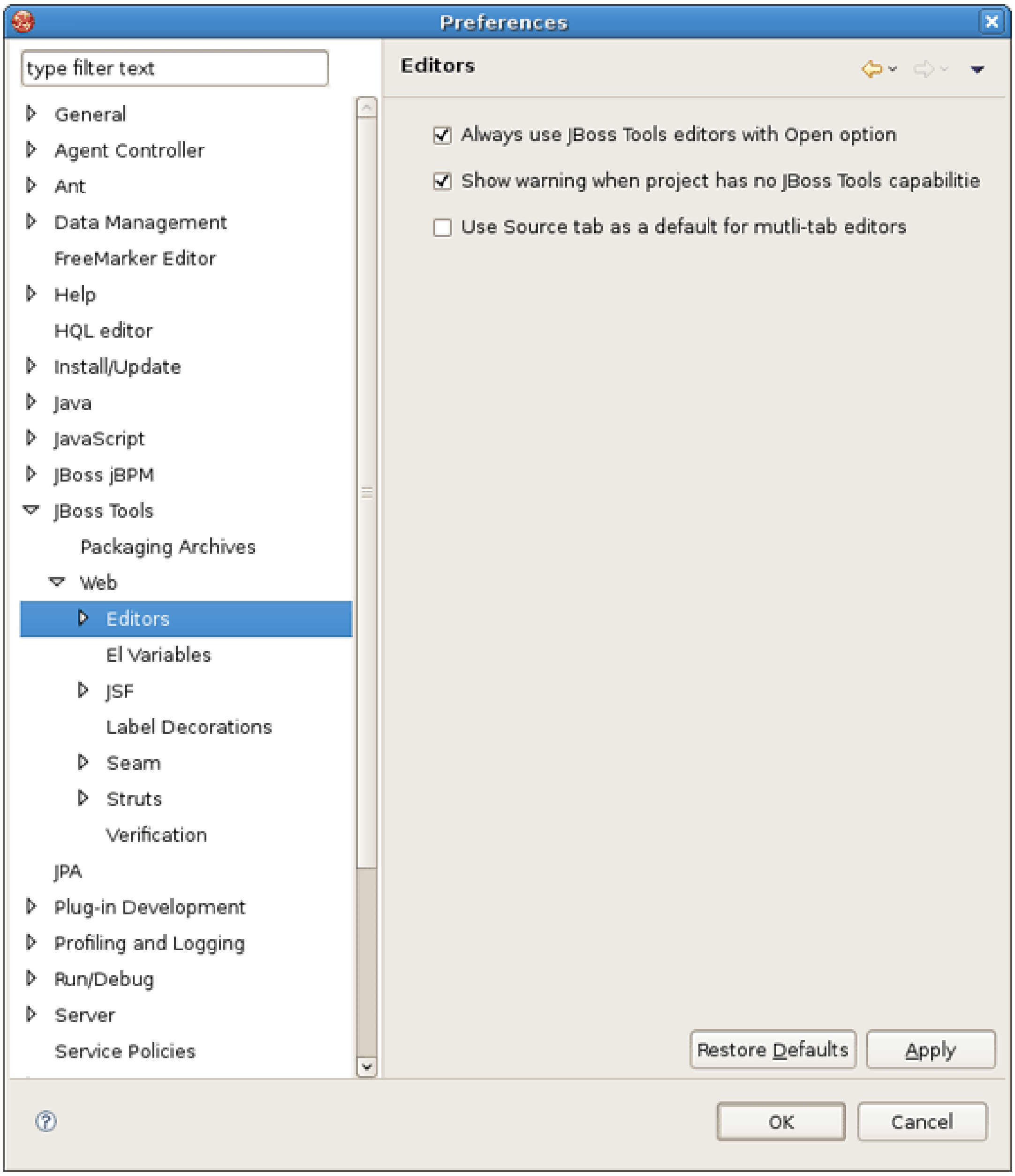
Table 8.1. Packaging Archives Preferences

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

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 .	On
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

8.2. Editors

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


Preferences
type filter text
General
Agent Controller
Ant
Data Management
FreeMarker Editor
Help
HQL editor
Install/Update
Java
JavaScript
JBoss jBPM
JBoss Tools
Packaging Archives
Web
Editors
El Variables
JSF
Label Decorations
Seam
Struts
Verification
JPA
Plug-in Development
Profiling and Logging
Run/Debug
Server
Service Policies
Editors
Restore Defaults
Apply
OK
Cancel

☑ 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

Figure 8.3. Editors

On the Editors page the following preferences are available:

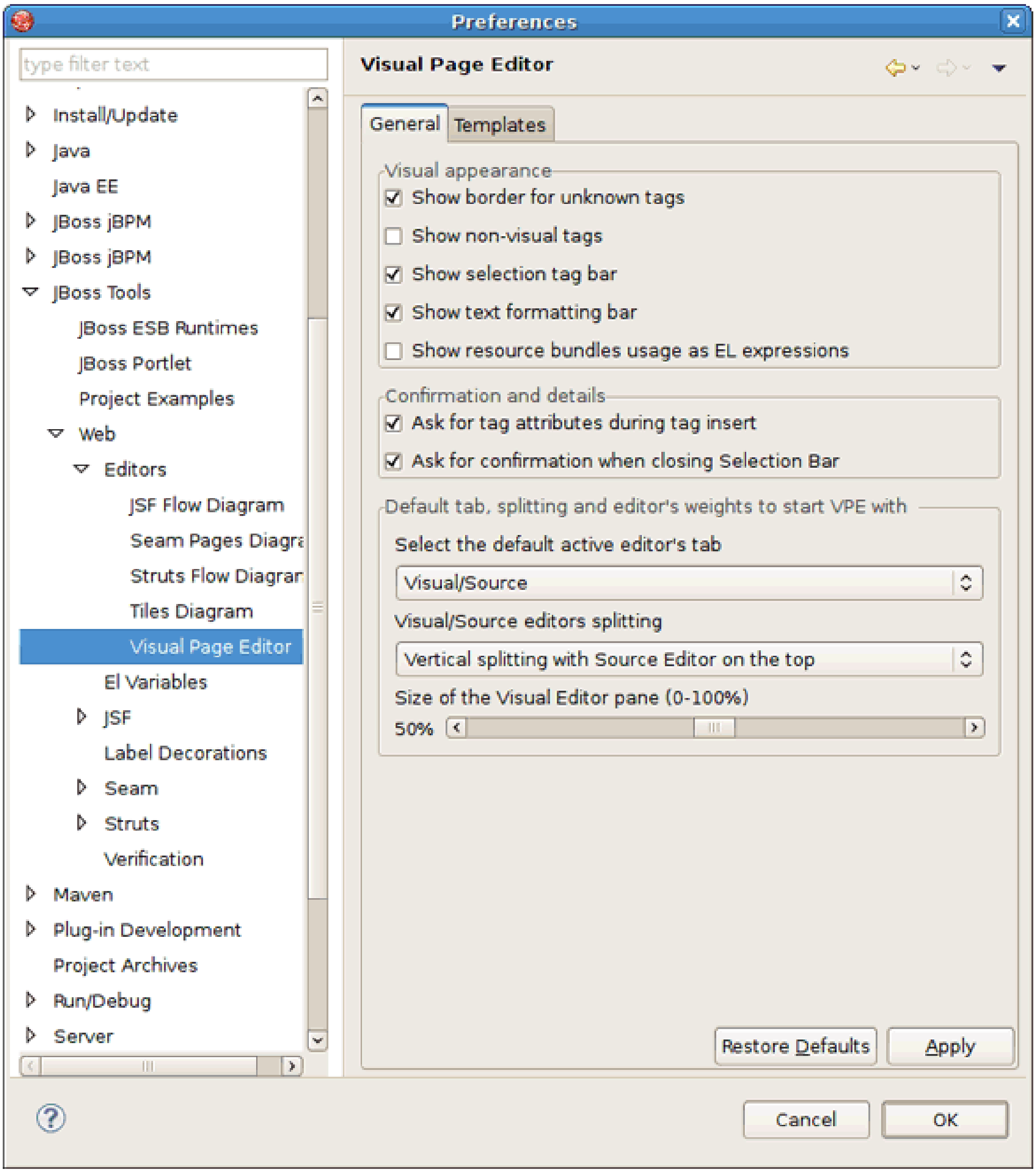
Table 8.2. Editors Preferences

Option	Description	Default
Always use JBoss 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.	On
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

8.3. 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.


Preferences
type filter text
Install/Update
Java
Java EE
JBoss jBPM
JBoss jBPM
JBoss Tools
JBoss ESB Runtimes
JBoss Portlet
Project Examples
Web
Editors
JSF Flow Diagram
Seam Pages Diagra
Struts Flow Diagrar
Tiles Diagram
Visual Page Editor
El Variables
JSF
Label Decorations
Seam
Struts
Verification
Maven
Plug-in Development
Project Archives
Run/Debug
Server
Visual Page Editor
General
Templates
Restore Defaults
Apply
Cancel
OK

Visual appearance

☑ Show border for unknown tags

☐ Show non-visual tags

☑ Show selection tag bar

☑ Show text formatting bar

☐ Show resource bundles usage as EL expressions

Confirmation and details

☑ Ask for tag attributes during tag insert

☑ Ask for confirmation when closing Selection Bar

Default tab, splitting and editor's weights to start VPE with

Select the default active editor's tab: Visual/Source

Visual/Source editors splitting: Vertical splitting with Source Editor on the top

Size of the Visual Editor pane (0-100%): 50%

Figure 8.4. Visual Page Editor

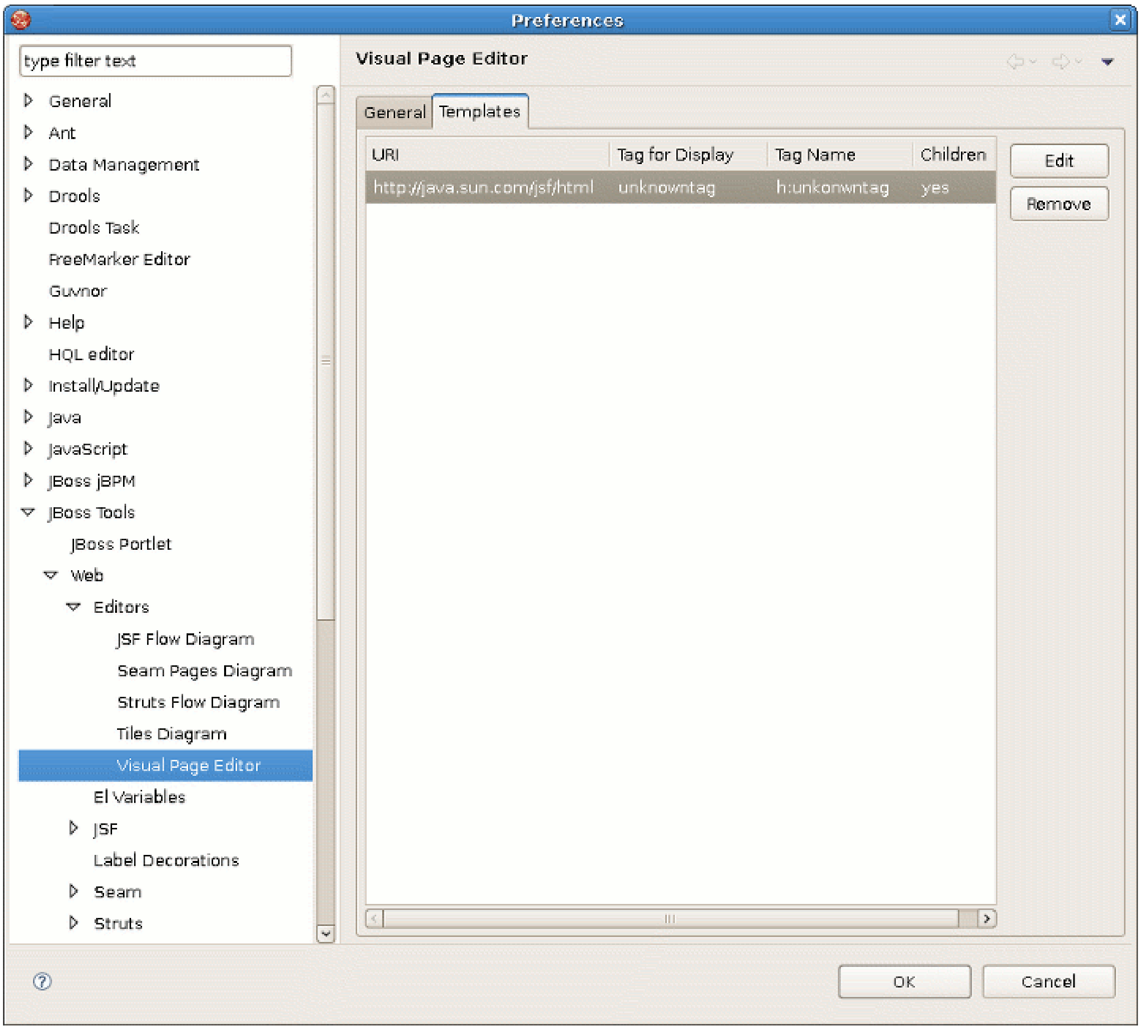
The next table lists the possible settings that you can adjust on the General tab of the VPE Preferences page.

Table 8.3. VPE Preferences

Option	Description	Default
Show border for unknown tags	The option allows to place the border around unknown tags or undo this	On

Option	Description	Default
Show non-visual tags	Check this box, if you want the editor shows non-visual elements on the page you're editing	Off
Show selection tag bar	The option allows to show/hide the Selection Bar	On
Show text formatting bar	Check this box in order to show/hide the Text Formatting bar	On
Show resource bundles usage as EL expressions	If the option is checked, the editor will show EL expressions instead of the resource values	Off
Ask 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
Ask for confirmation when closing the Selection Bar	Check this box if you don't want the confirmation window appears when closing the Selection Bar	On
Select the default active editor's tab	The option provides possibility to choose one of the following views - Visual/Source, Source or Preview, as default when opening the editor	Visual/Source
Visual/Source editors splitting	The option allows to choose one of the following Visual,Source layouts - Vertical Source on top, Vertical Visual on top,Horizontal Source to the left or Horizontal Visual to the left, as a default one when opening the Visual/Source view	Vertical splitting with Source Editor on the top
Size of the 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 *VPE templates*.


Preferences
type filter text
General
Ant
Data Management
Drools
Drools Task
FreeMarker Editor
Guvnor
Help
HQL editor
Install/Update
Java
JavaScript
JBoss jBPM
JBoss Tools
JBoss Portlet
Web
Editors
JSF Flow Diagram
Seam Pages Diagram
Struts Flow Diagram
Tiles Diagram
Visual Page Editor
El Variables
JSF
Label Decorations
Seam
Struts
Visual Page Editor
General
Templates
Edit
Remove
OK
Cancel

URI	Tag for Display	Tag Name	Children
http://java.sun.com/jsf/html	unknowntag	h:unkonwntag	yes

Figure 8.5. Visual Page Editor Templates

Select a template for editing from the available list and press *Edit* button. It will pick up the *Template dialog [52]* where you can adjust new settings.

8.4. 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*.

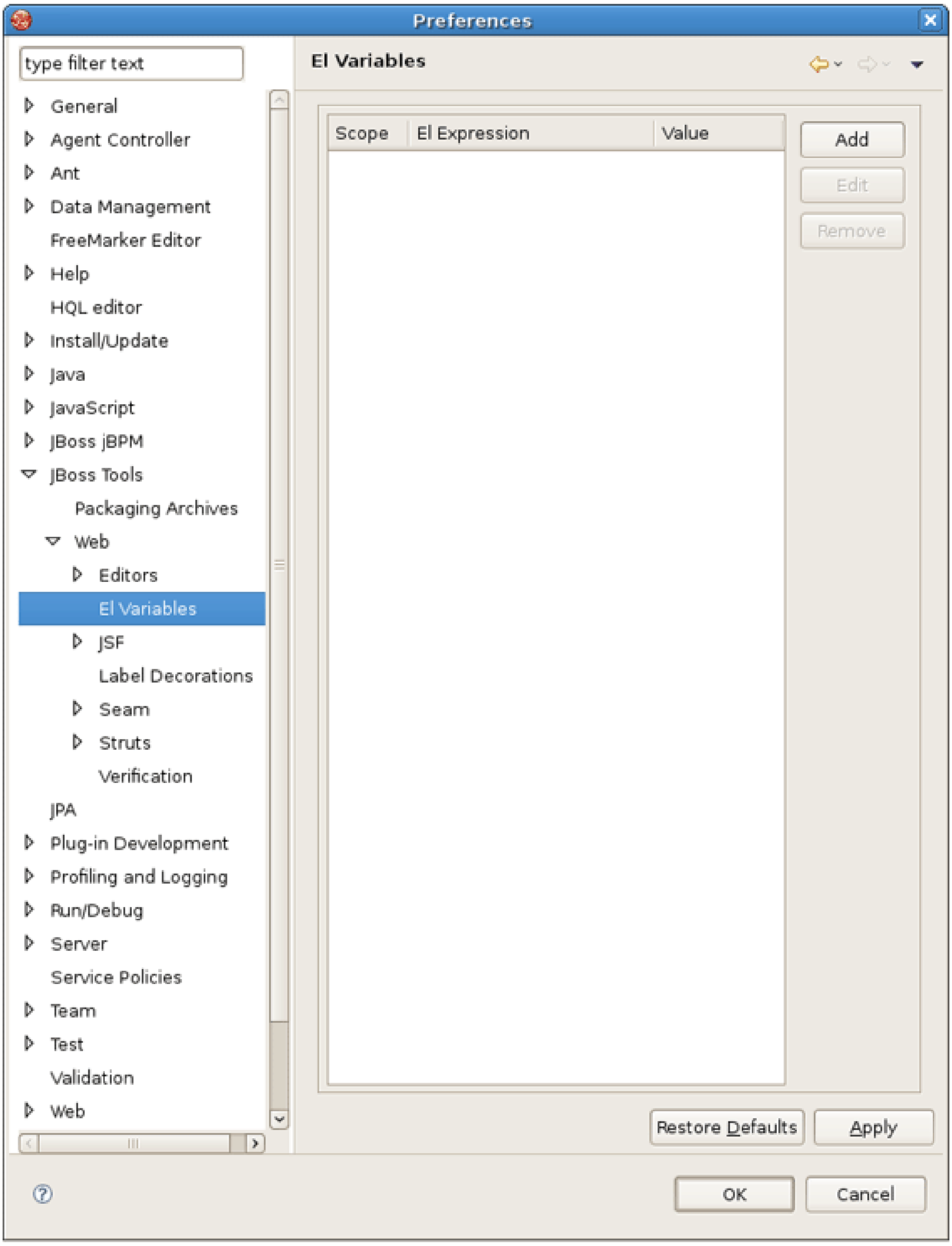
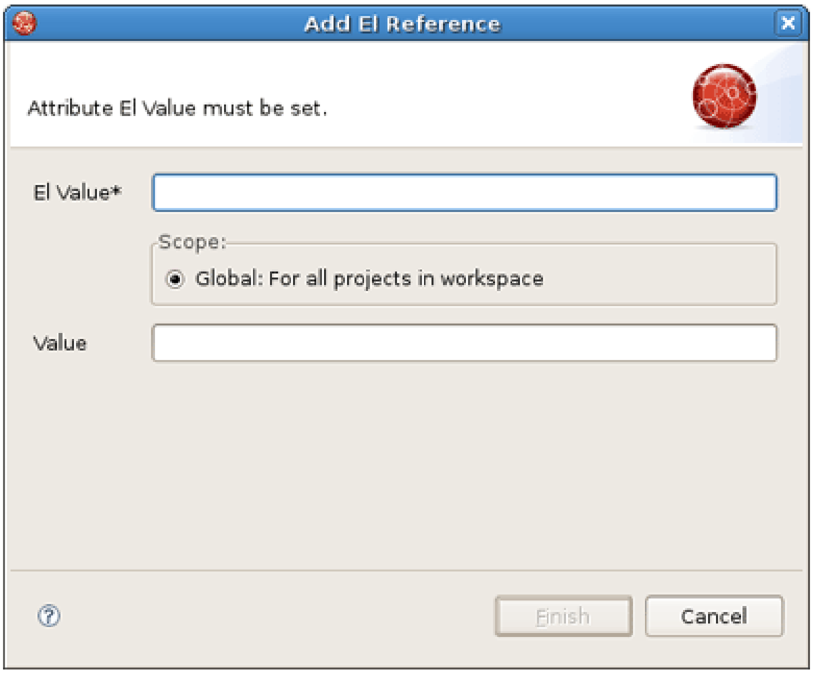

Preferences
type filter text
General
Agent Controller
Ant
Data Management
FreeMarker Editor
Help
HQL editor
Install/Update
Java
JavaScript
JBoss jBPM
JBoss Tools
Packaging Archives
Web
Editors
El Variables
JSF
Label Decorations
Seam
Struts
Verification
JPA
Plug-in Development
Profiling and Logging
Run/Debug
Server
Service Policies
Team
Test
Validation
Web
El Variables
Scope
El Expression
Value
Add
Edit
Remove
Restore Defaults
Apply
OK
Cancel

Figure 8.6. 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* .



Attribute El Value must be set.

El Value*

Scope:

● Global: For all projects in workspace

Value

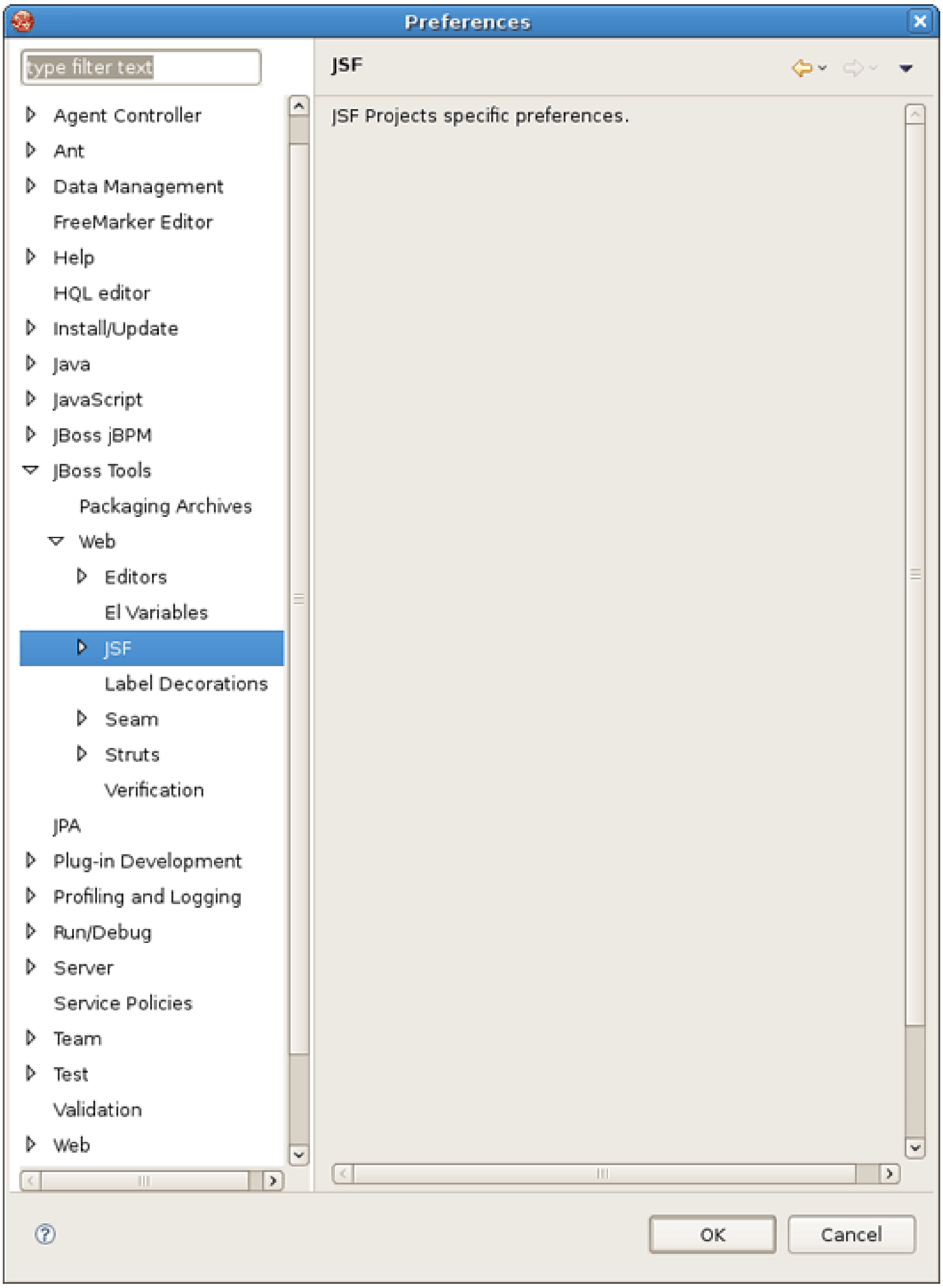
Figure 8.7. Adding a Global El Variable

Tip:

If you specify an equal variable in *Substitute EL expressions dialog* *[59]* and in Preference EL dialog, variable from preference dialog will have priority.

8.5. JSF

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



Preferences
type filter text
Agent Controller
Ant
Data Management
FreeMarker Editor
Help
HQL editor
Install/Update
Java
JavaScript
JBoss jBPM
JBoss Tools
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
JSF
OK
Cancel

JSF Projects specific preferences.

Figure 8.8. JSF

8.6. JSF Pages

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

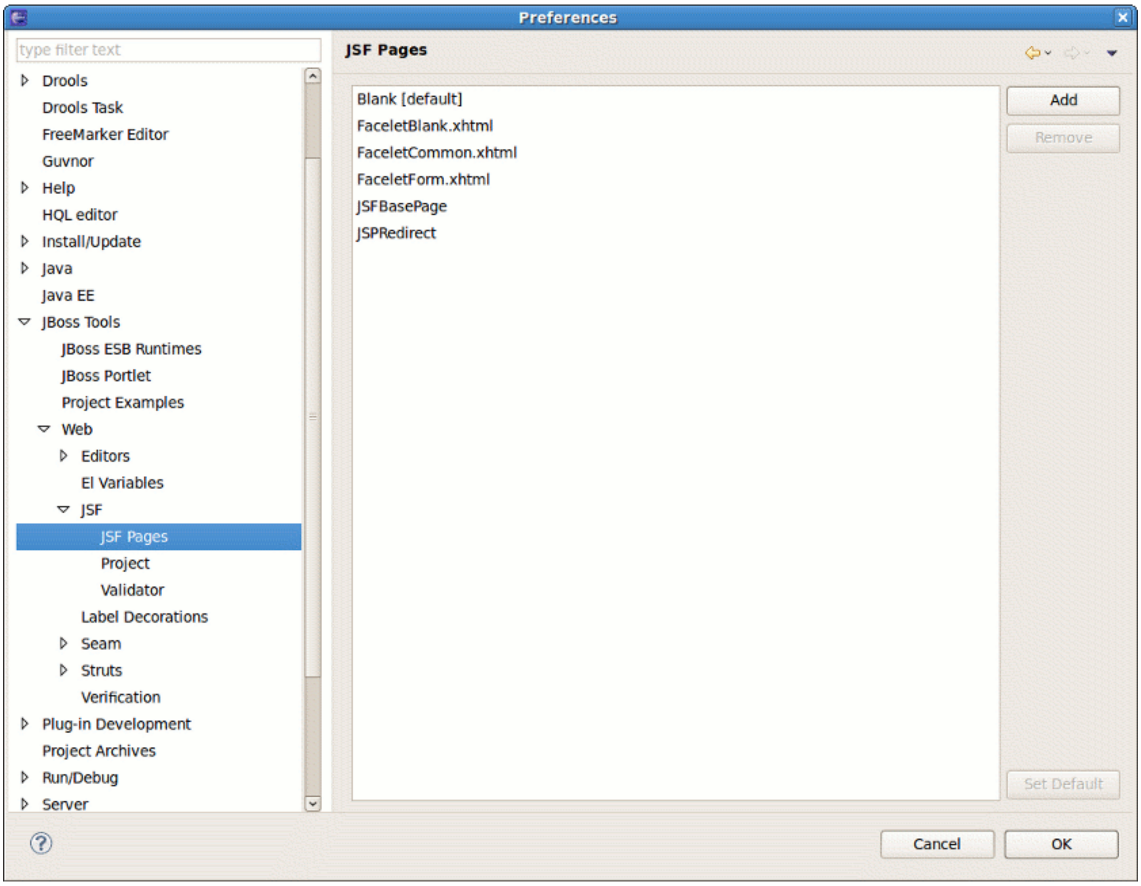

Preferences
type filter text
Drools
Drools Task
FreeMarker Editor
Guvnor
Help
HQL editor
Install/Update
Java
Java EE
JBoss Tools
JBoss ESB Runtimes
JBoss Portlet
Project Examples
Web
Editors
El Variables
JSF
JSF Pages
Project
Validator
Label Decorations
Seam
Struts
Verification
Plug-in Development
Project Archives
Run/Debug
Server
JSF Pages
Blank [default]
FaceletBlank.xhtml
FaceletCommon.xhtml
FaceletForm.xhtml
JSFBasePage
JSPRedirect
Add
Remove
Set Default
Cancel
OK

Figure 8.9. JSF Page

8.7. JSF Project

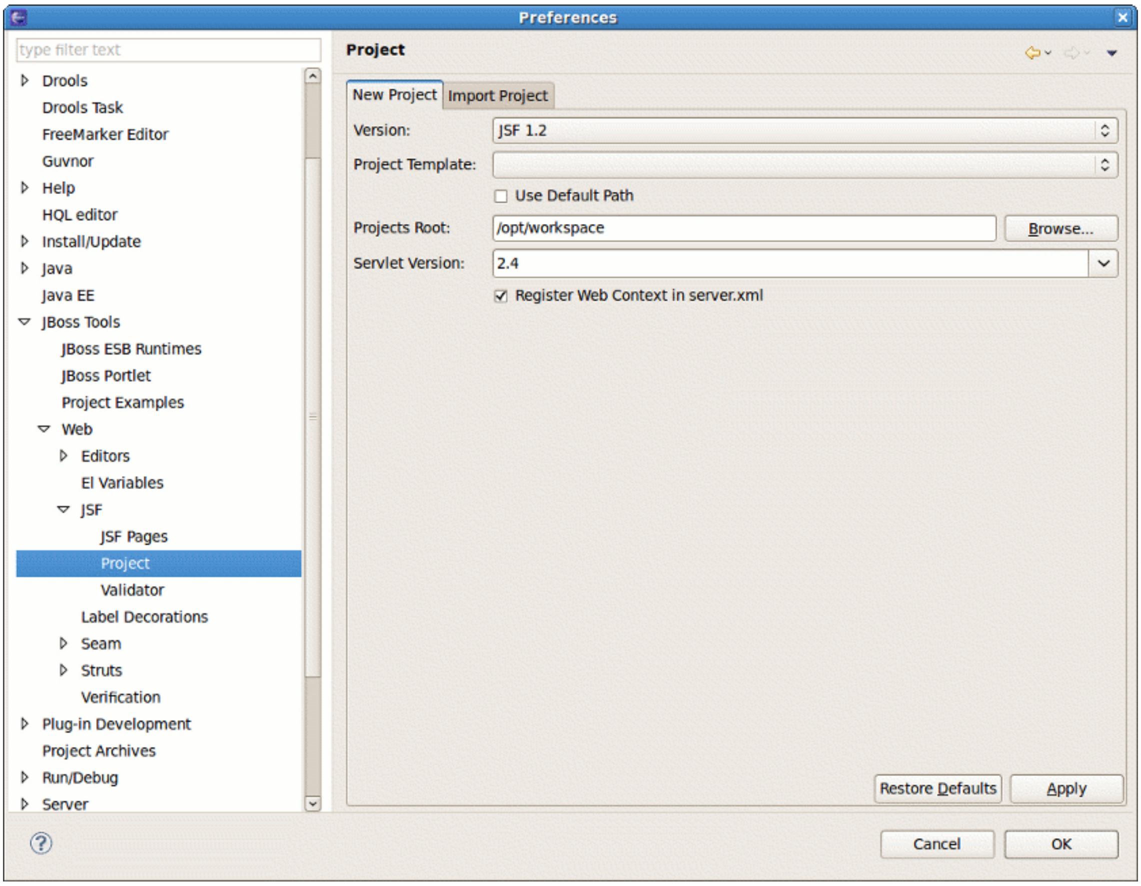
Select *JBoss Tools > Web > JSF > Project* to see JSF Project preferences page.

On the *New Project* tab you can set default values for *New JSF Project* 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
type filter text
Drools
Drools Task
FreeMarker Editor
Guvnor
Help
HQL editor
Install/Update
Java
Java EE
JBoss Tools
JBoss ESB Runtimes
JBoss Portlet
Project Examples
Web
Editors
El Variables
JSF
JSF Pages
Project
Validator
Label Decorations
Seam
Struts
Verification
Plug-in Development
Project Archives
Run/Debug
Server
Project
New Project
Import Project
Browse...
Restore Defaults
Apply
Cancel
OK

Version: JSF 1.2

Project Template:

☐ Use Default Path

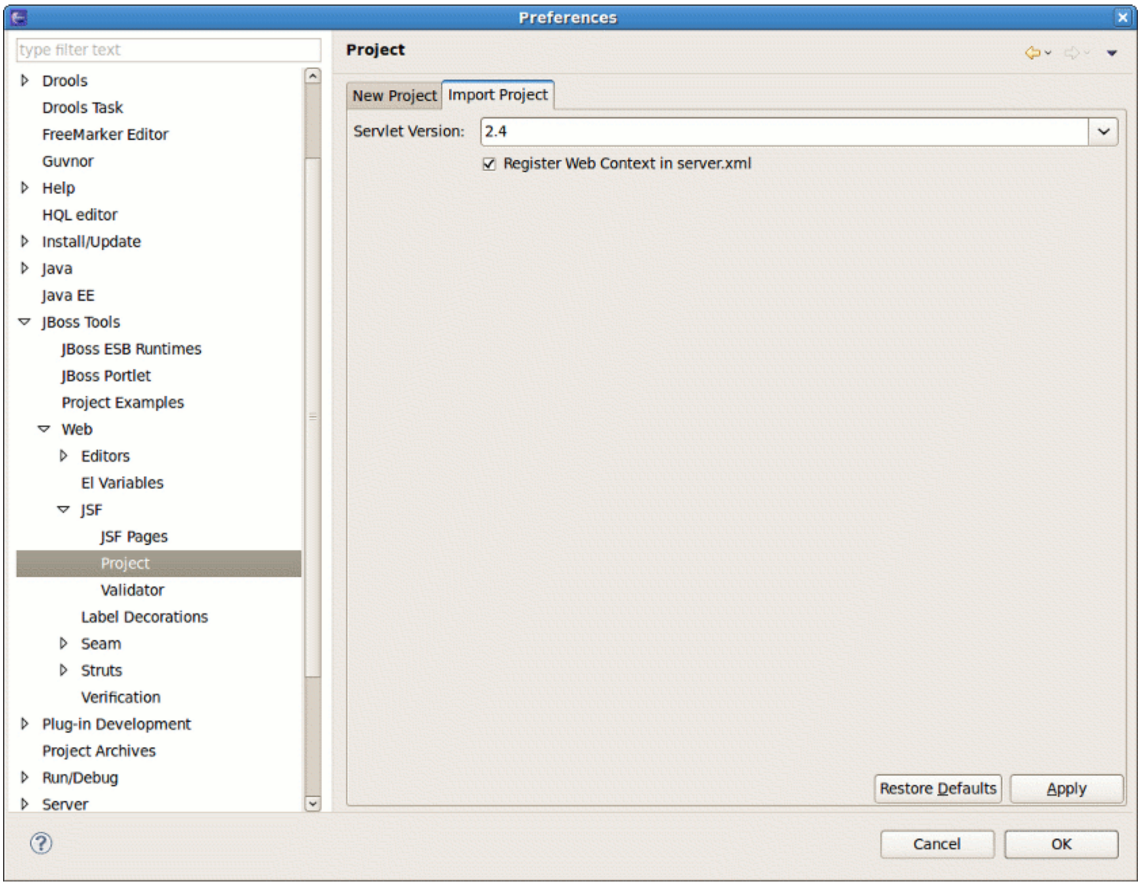
Projects Root: /opt/workspace

Servlet Version: 2.4

☑ Register Web Context in server.xml

Figure 8.10. New JSF Project 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.


Preferences
type filter text
Drools
Drools Task
FreeMarker Editor
Guvnor
Help
HQL editor
Install/Update
Java
Java EE
JBoss Tools
JBoss ESB Runtimes
JBoss Portlet
Project Examples
Web
Editors
El Variables
JSF
JSF Pages
Project
Validator
Label Decorations
Seam
Struts
Verification
Plug-in Development
Project Archives
Run/Debug
Server
Project
New Project
Import Project
Restore Defaults
Apply
Cancel
OK

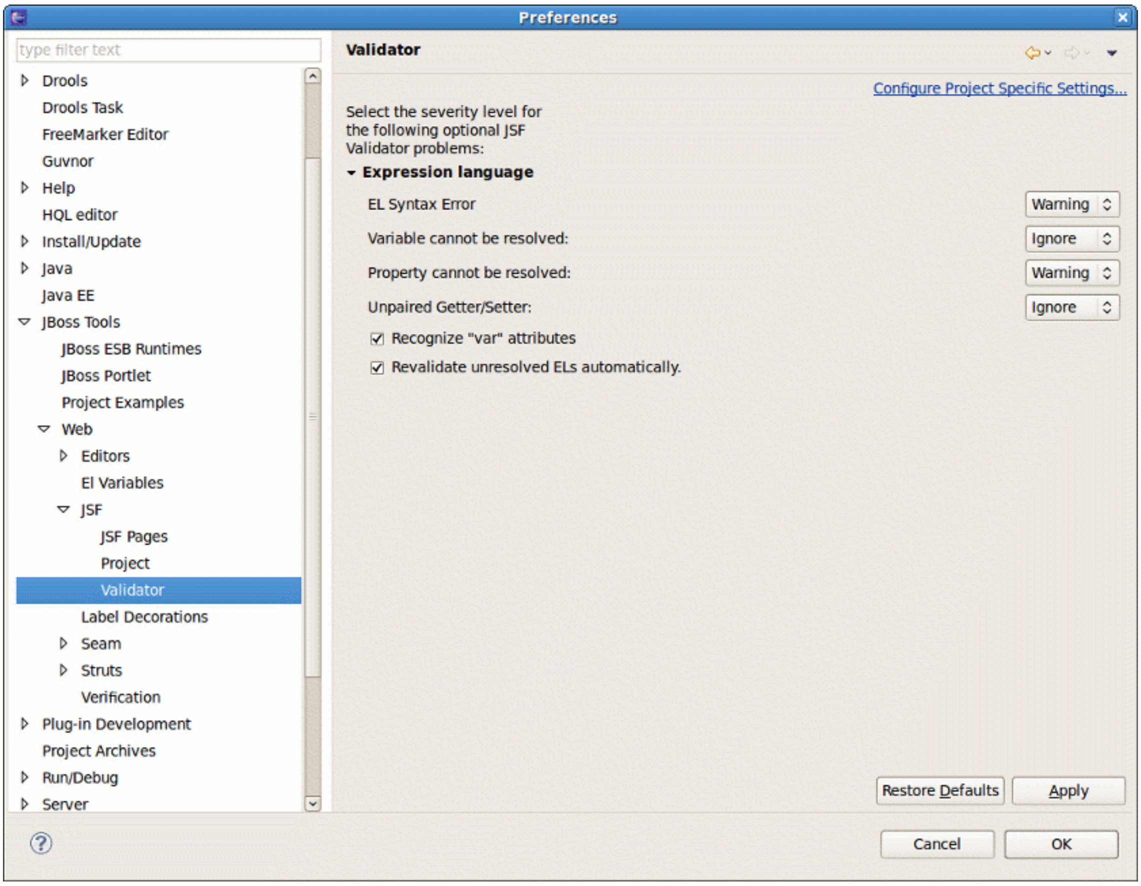
Servlet Version: 2.4

☑ Register Web Context in server.xml

Figure 8.11. Import JSF Project Preferences

8.8. JSF Validator

Select JBoss Tools > Web > JSF > Validator page to configure the validator behavior. On this page you can change the severity level for different jsf problems which are controlled by validator.


Preferences
type filter text
Drools
Drools Task
FreeMarker Editor
Guvnor
Help
HQL editor
Install/Update
Java
Java EE
JBoss Tools
JBoss ESB Runtimes
JBoss Portlet
Project Examples
Web
Editors
El Variables
JSF
JSF Pages
Project
Validator
Label Decorations
Seam
Struts
Verification
Plug-in Development
Project Archives
Run/Debug
Server
Configure Project Specific Settings...
Restore Defaults
Apply
Cancel
OK

Validator

Select the severity level for the following optional JSF Validator problems:

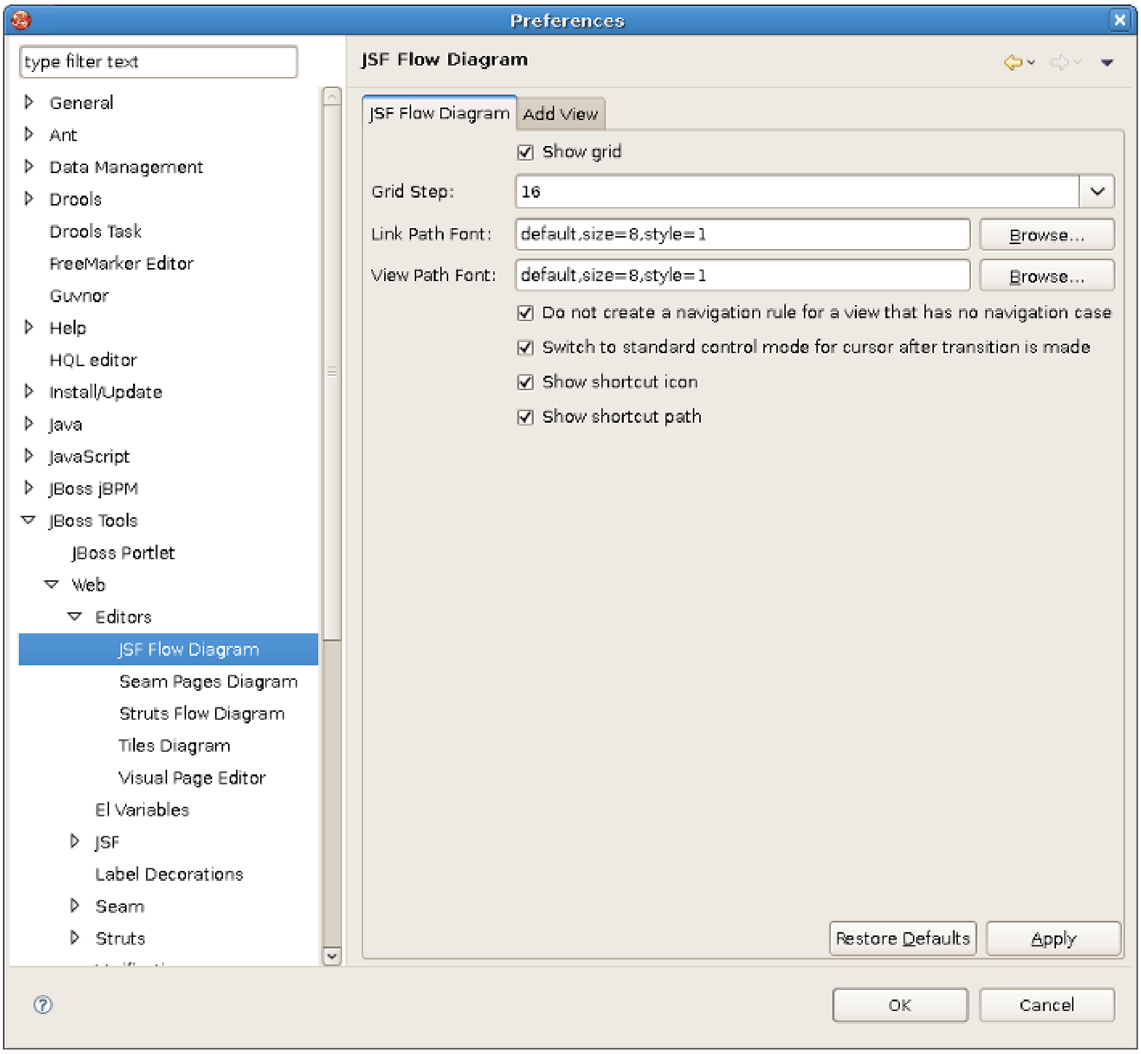
Expression language

EL Syntax Error: Warning
Variable cannot be resolved: Ignore
Property cannot be resolved: Warning
Unpaired Getter/Setter: Ignore
☑ Recognize "var" attributes
☑ Revalidate unresolved ELs automatically.

Figure 8.12. JSF Validator

8.9. 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.


Preferences
type filter text
General
Ant
Data Management
Drools
Drools Task
FreeMarker Editor
Guvnor
Help
HQL editor
Install/Update
Java
JavaScript
JBoss jBPM
JBoss Tools
JBoss Portlet
Web
Editors
JSF Flow Diagram
Seam Pages Diagram
Struts Flow Diagram
Tiles Diagram
Visual Page Editor
El Variables
JSF
Label Decorations
Seam
Struts
JSF Flow Diagram
JSF Flow Diagram
Add View
Browse...
Browse...
Restore Defaults
Apply
OK
Cancel

☑ Show grid

Grid Step: 16

Link Path Font: default,size=8,style=1

View Path Font: default,size=8,style=1

☑ Do not create a navigation rule for a view that has no navigation case

☑ Switch to standard control mode for cursor after transition is made

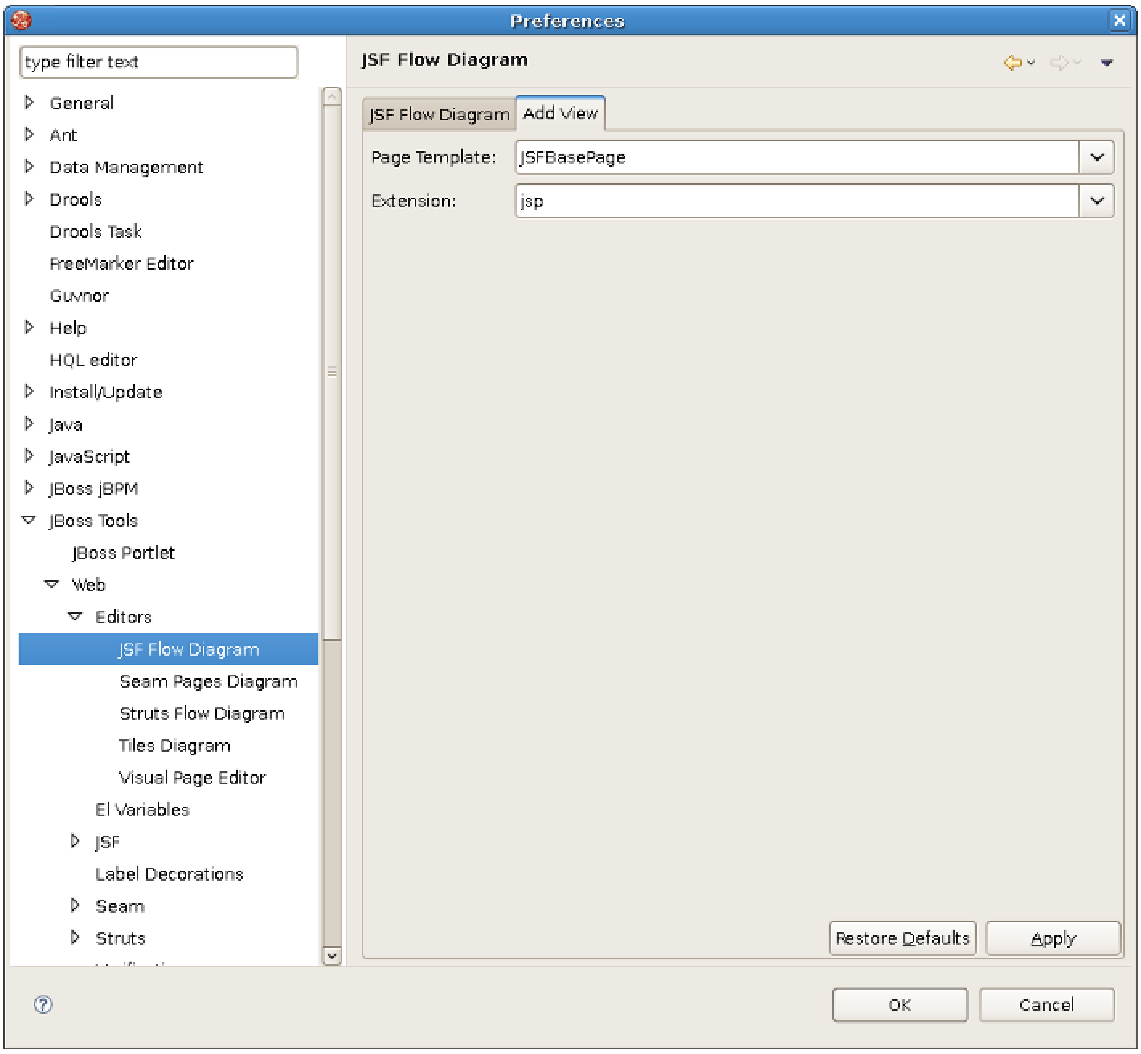
☑ Show shortcut icon

☑ Show shortcut path

Figure 8.13. 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.


Preferences
type filter text
General
Ant
Data Management
Drools
Drools Task
FreeMarker Editor
Guvnor
Help
HQL editor
Install/Update
Java
JavaScript
JBoss jBPM
JBoss Tools
JBoss Portlet
Web
Editors
JSF Flow Diagram
Seam Pages Diagram
Struts Flow Diagram
Tiles Diagram
Visual Page Editor
El Variables
JSF
Label Decorations
Seam
Struts
JSF Flow Diagram
JSF Flow Diagram
Add View
Restore Defaults
Apply
OK
Cancel

Page Template: JSFBasePage

Extension: jsp

Figure 8.14. 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.

8.10. Label Decorations

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

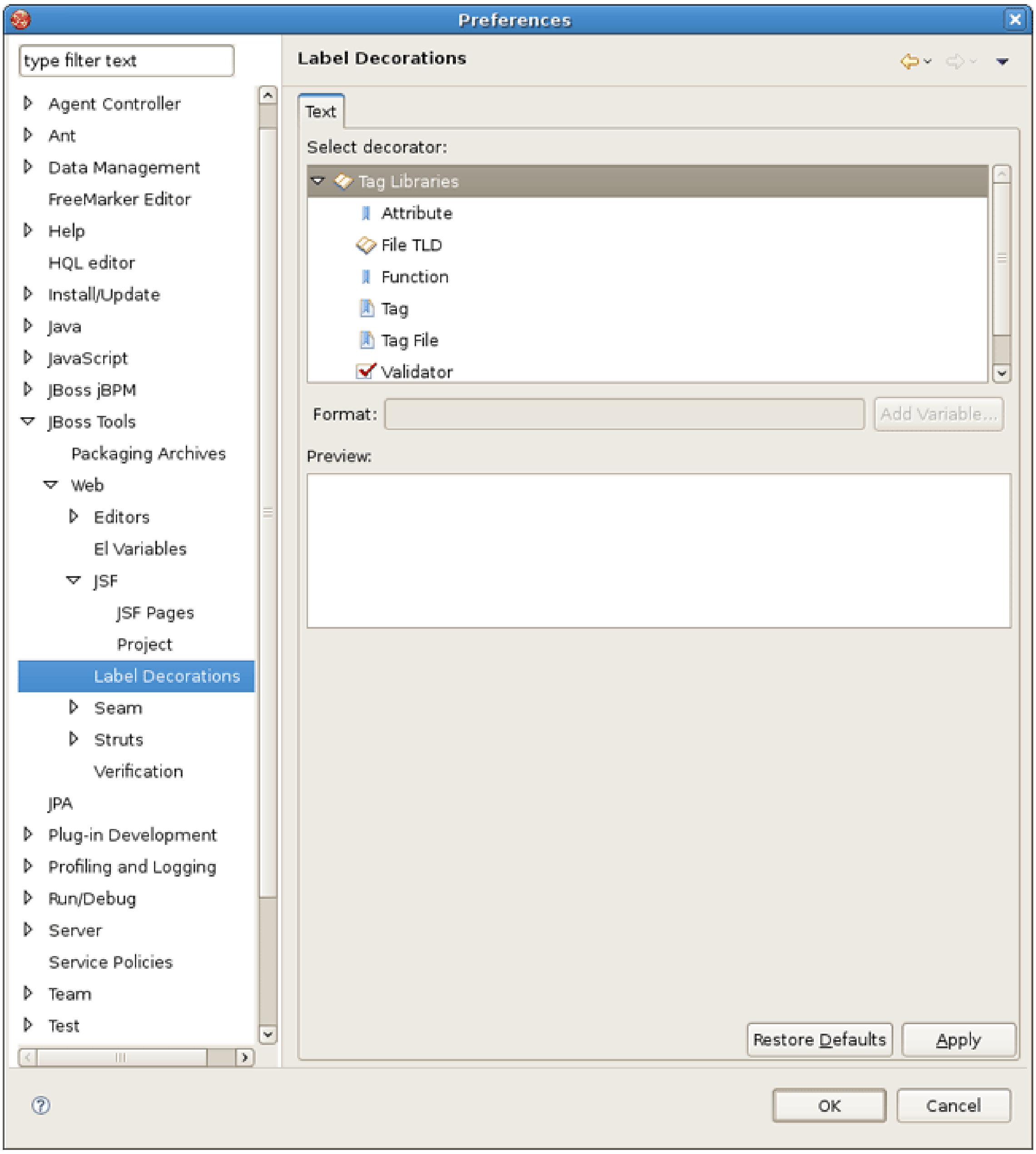

Preferences
type filter text
Agent Controller
Ant
Data Management
FreeMarker Editor
Help
HQL editor
Install/Update
Java
JavaScript
JBoss jBPM
JBoss Tools
Packaging Archives
Web
Editors
El Variables
JSF
JSF Pages
Project
Label Decorations
Seam
Struts
Verification
JPA
Plug-in Development
Profiling and Logging
Run/Debug
Server
Service Policies
Team
Test
Label Decorations
Text
Select decorator:
Tag Libraries
Attribute
File TLD
Function
Tag
Tag File
Validator
Format:
Add Variable...
Preview:
Restore Defaults
Apply
OK
Cancel

Figure 8.15. 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.

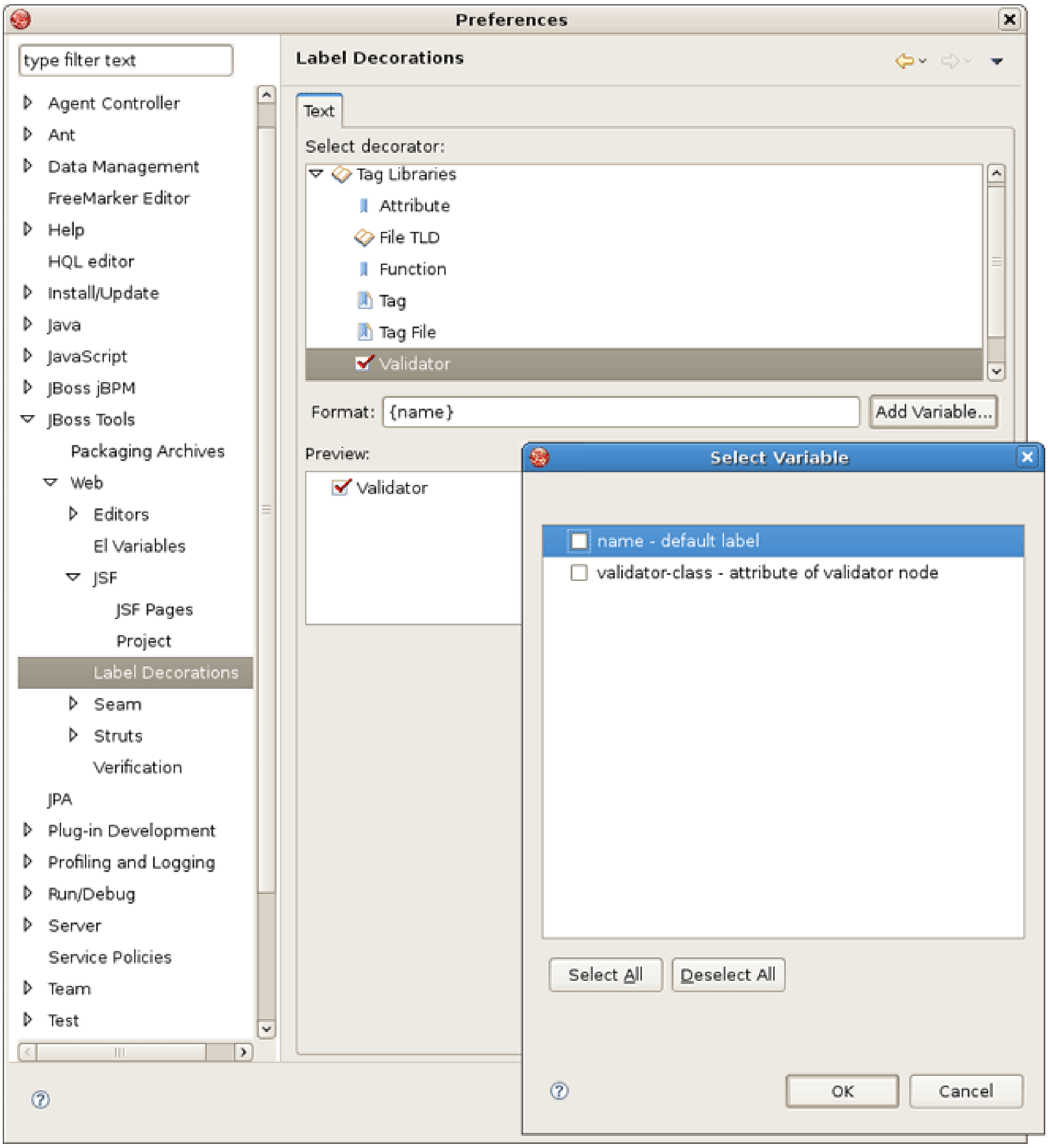

Preferences
type filter text
Agent Controller
Ant
Data Management
FreeMarker Editor
Help
HQL editor
Install/Update
Java
JavaScript
JBoss jBPM
JBoss Tools
Packaging Archives
Web
Editors
El Variables
JSF
JSF Pages
Project
Label Decorations
Seam
Struts
Verification
JPA
Plug-in Development
Profiling and Logging
Run/Debug
Server
Service Policies
Team
Test
Label Decorations
Text
Select decorator:
Tag Libraries
Attribute
File TLD
Function
Tag
Tag File
Validator
Format:
{name}
Add Variable...
Preview:
Validator
Select Variable
name - default label
validator-class - attribute of validator node
Select All
Deselect All
OK
Cancel

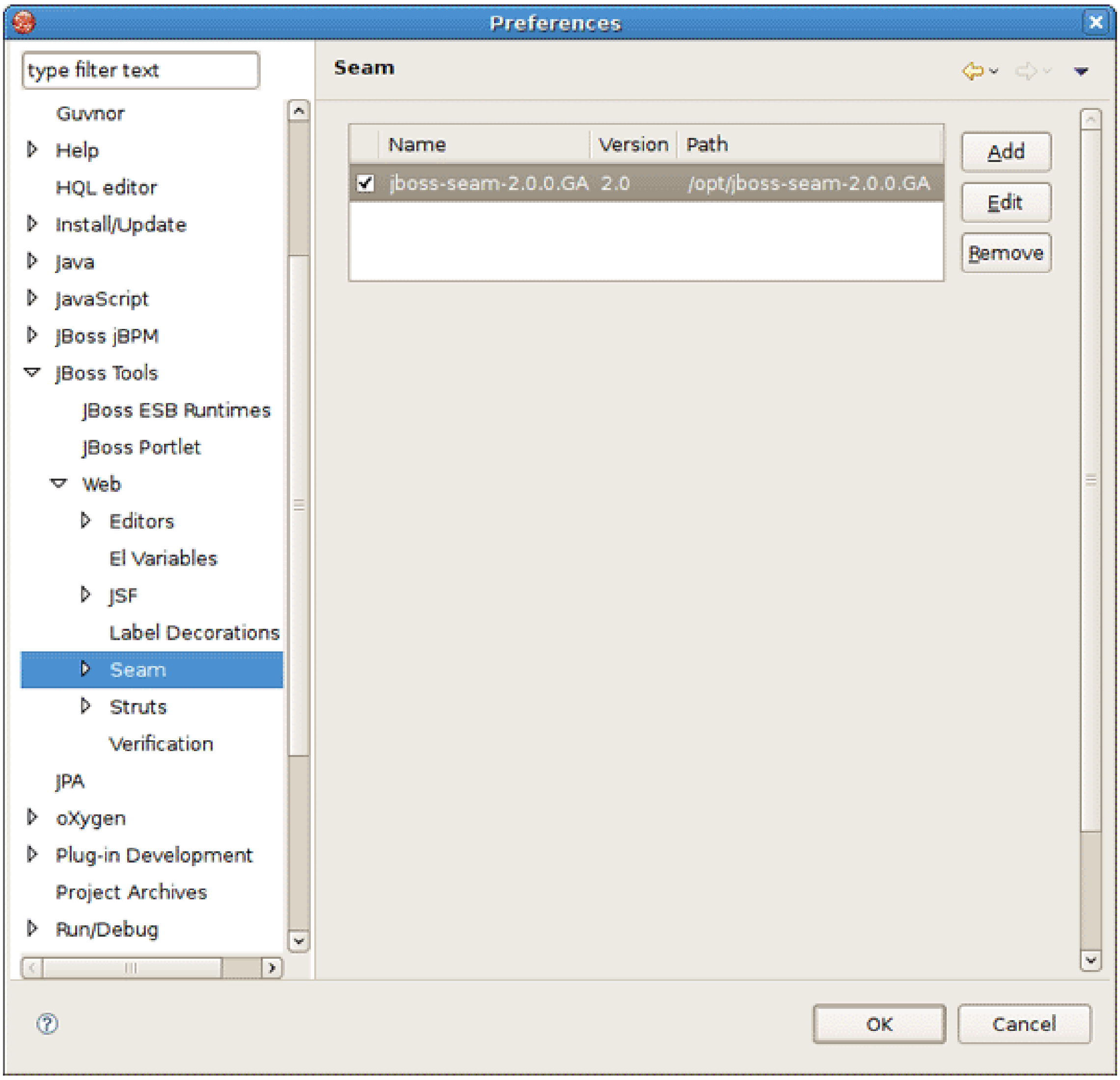
Figure 8.16. Label Decoration for Validator

8.11. Seam

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

On *Seam* screen you can add and remove Seam runtimes.

Here is what Seam preference page looks like:


Preferences
type filter text
Guvnor
Help
HQL editor
Install/Update
Java
JavaScript
JBoss jBPM
JBoss Tools
JBoss ESB Runtimes
JBoss Portlet
Web
Editors
El Variables
JSF
Label Decorations
Seam
Struts
Verification
JPA
oXygen
Plug-in Development
Project Archives
Run/Debug
Seam
Add
Edit
Remove
OK
Cancel

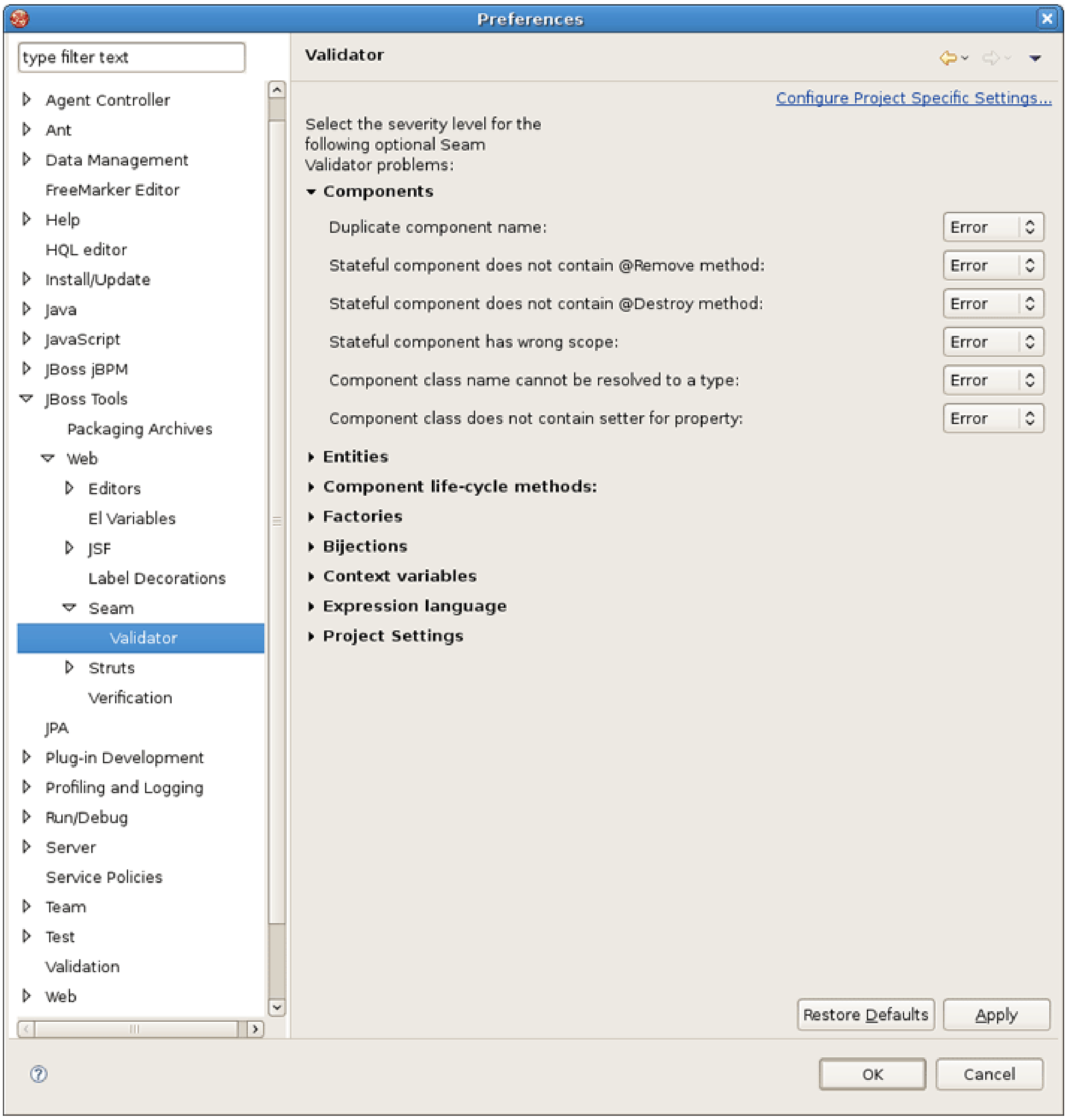
	Name	Version	Path
☑	jboss-seam-2.0.0.GA	2.0	/opt/jboss-seam-2.0.0.GA

Figure 8.17. Seam preference page

8.12. Seam Validator

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

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


Preferences
type filter text
Agent Controller
Ant
Data Management
FreeMarker Editor
Help
HQL editor
Install/Update
Java
JavaScript
JBoss jBPM
JBoss Tools
Packaging Archives
Web
Editors
El Variables
JSF
Label Decorations
Seam
Validator
Struts
Verification
JPA
Plug-in Development
Profiling and Logging
Run/Debug
Server
Service Policies
Team
Test
Validation
Web
Configure Project Specific Settings...
Restore Defaults
Apply
OK
Cancel

Validator

Select the severity level for the following optional Seam Validator problems:

Components

Duplicate component name: Error

Stateful component does not contain @Remove method: Error

Stateful component does not contain @Destroy method: Error

Stateful component has wrong scope: Error

Component class name cannot be resolved to a type: Error

Component class does not contain setter for property: Error

Entities

Component life-cycle methods:

Factories

Bijections

Context variables

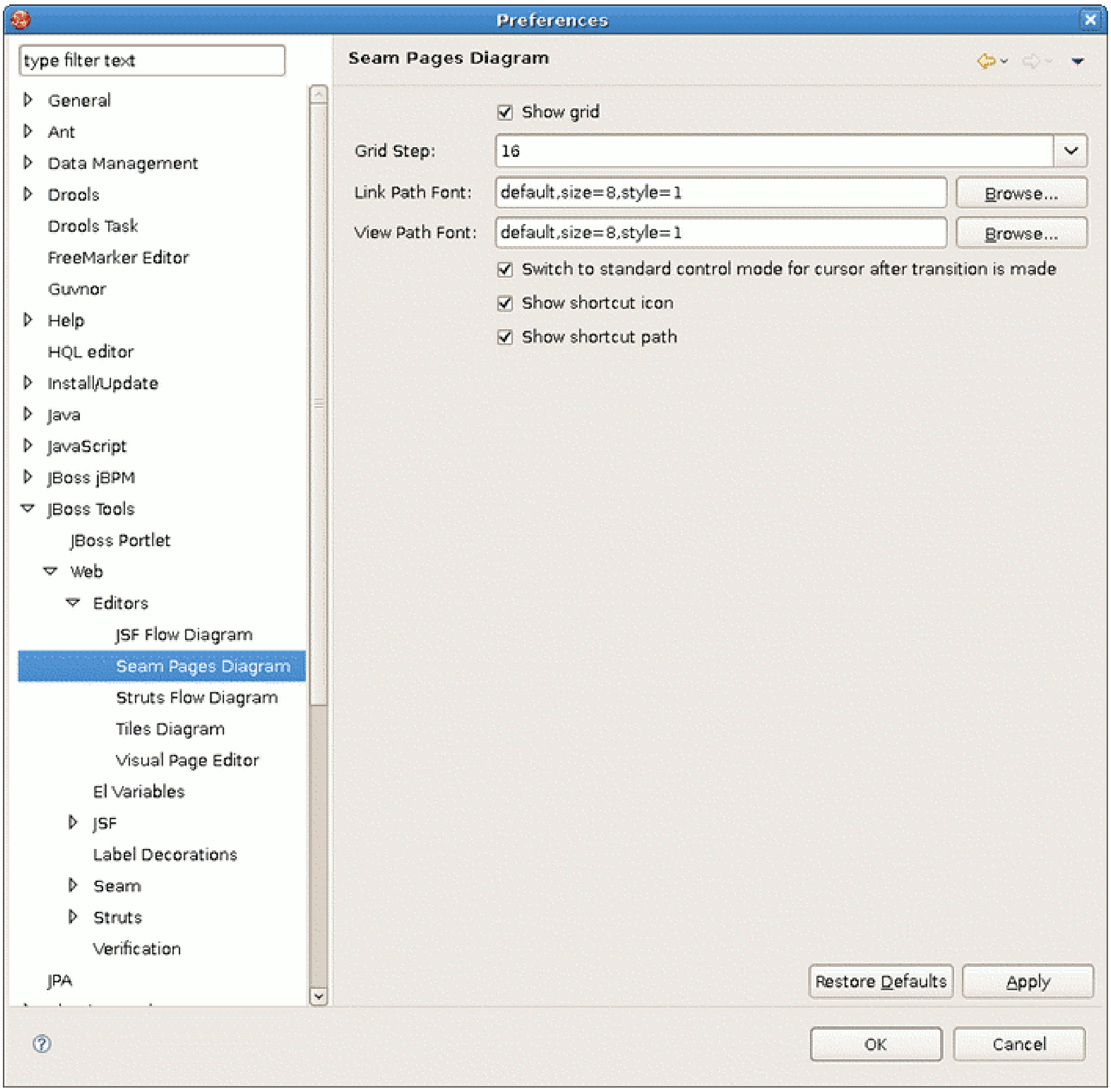
Expression language

Project Settings

Figure 8.18. Seam Validator

8.13. Seam Pages Diagram

In order to customize the layout of the Diagram used for editing and composing page.xml file in Graphical mode of Seam Pages Editor you can go to Window > Preferences > JBoss Tools > Web > Seam > Editors > Seam Pages Diagram.


Preferences
type filter text
General
Ant
Data Management
Drools
Drools Task
FreeMarker Editor
Guvnor
Help
HQL editor
Install/Update
Java
JavaScript
JBoss jBPM
JBoss Tools
JBoss Portlet
Web
Editors
JSF Flow Diagram
Seam Pages Diagram
Struts Flow Diagram
Tiles Diagram
Visual Page Editor
El Variables
JSF
Label Decorations
Seam
Struts
Verification
JPA
Browse...
Browse...
Restore Defaults
Apply
OK
Cancel

Seam Pages Diagram

☑ Show grid

Grid Step: 16

Link Path Font: default,size=8,style=1

View Path Font: default,size=8,style=1

☑ Switch to standard control mode for cursor after transition is made

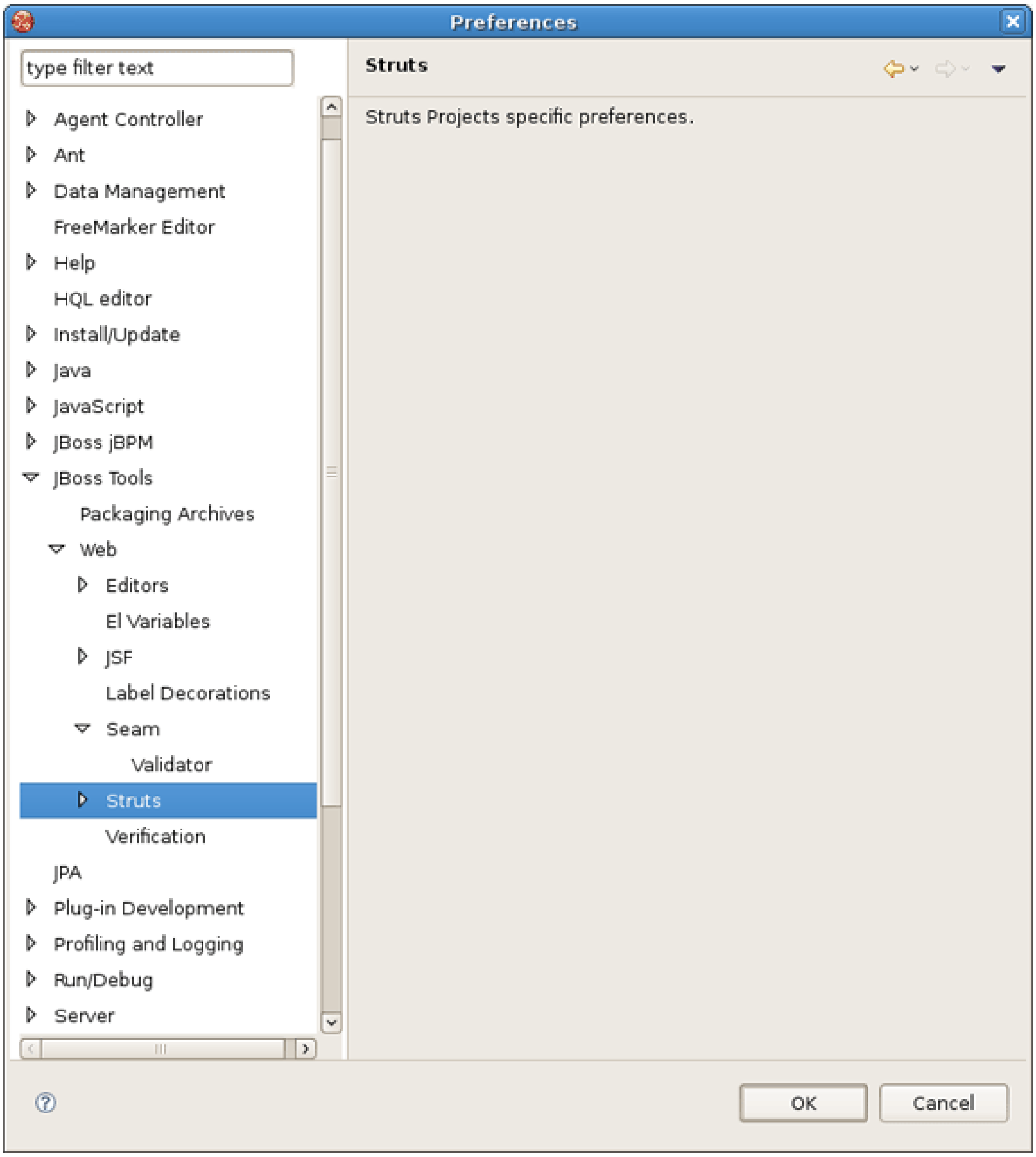
☑ Show shortcut icon

☑ Show shortcut path

Figure 8.19. Preferences of Seam Pages Diagram

8.14. Struts

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

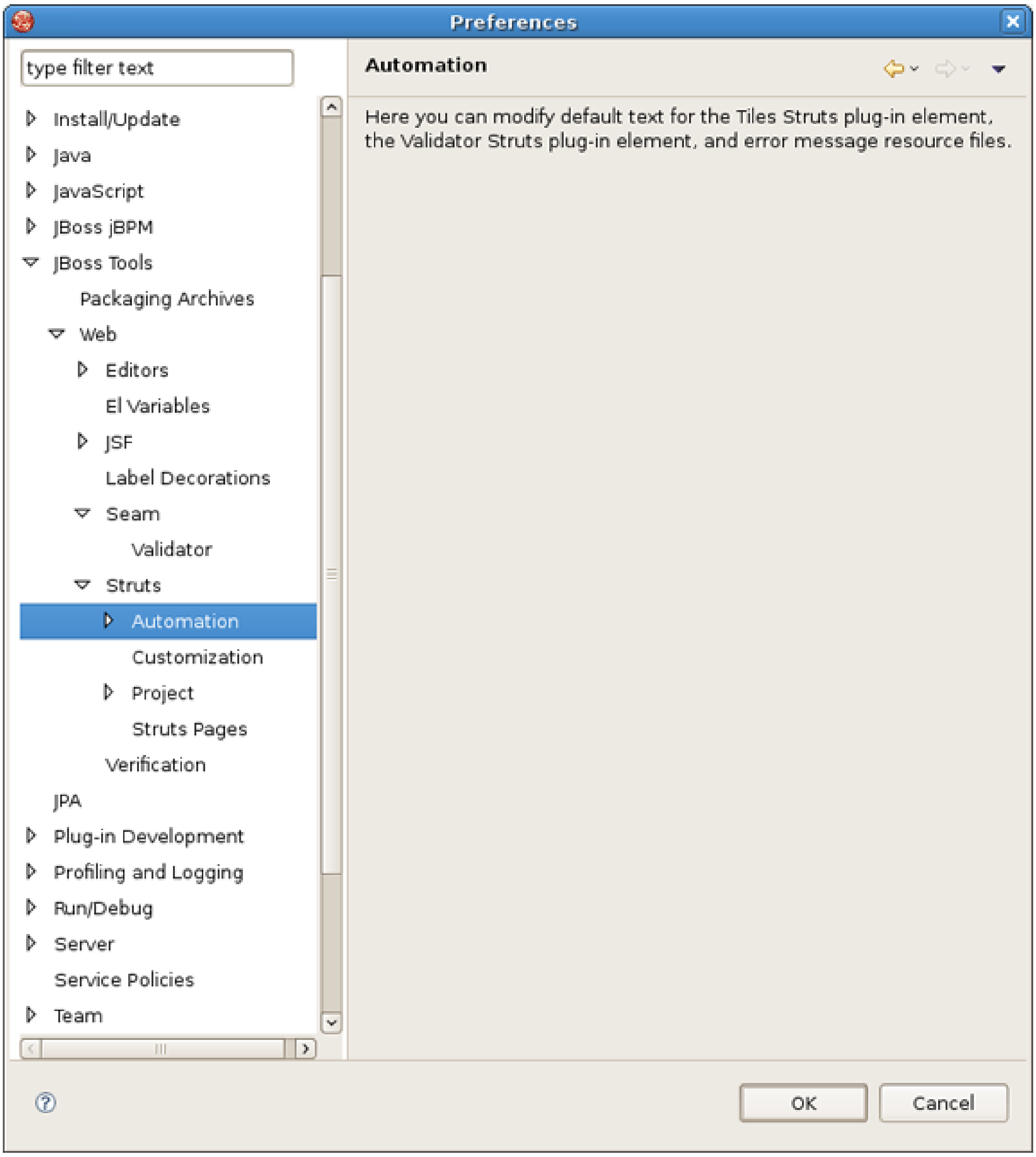

Preferences
type filter text
Agent Controller
Ant
Data Management
FreeMarker Editor
Help
HQL editor
Install/Update
Java
JavaScript
JBoss jBPM
JBoss Tools
Packaging Archives
Web
Editors
El Variables
JSF
Label Decorations
Seam
Validator
Struts
Verification
JPA
Plug-in Development
Profiling and Logging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

Struts Projects specific preferences.

Figure 8.20. Struts projects preferences Page.

8.15. Struts Automation

On *Automation* panel you can modify default text for the Title Struts plug-in element, the Validator Struts plug-in element, and error message resource files.



Automation

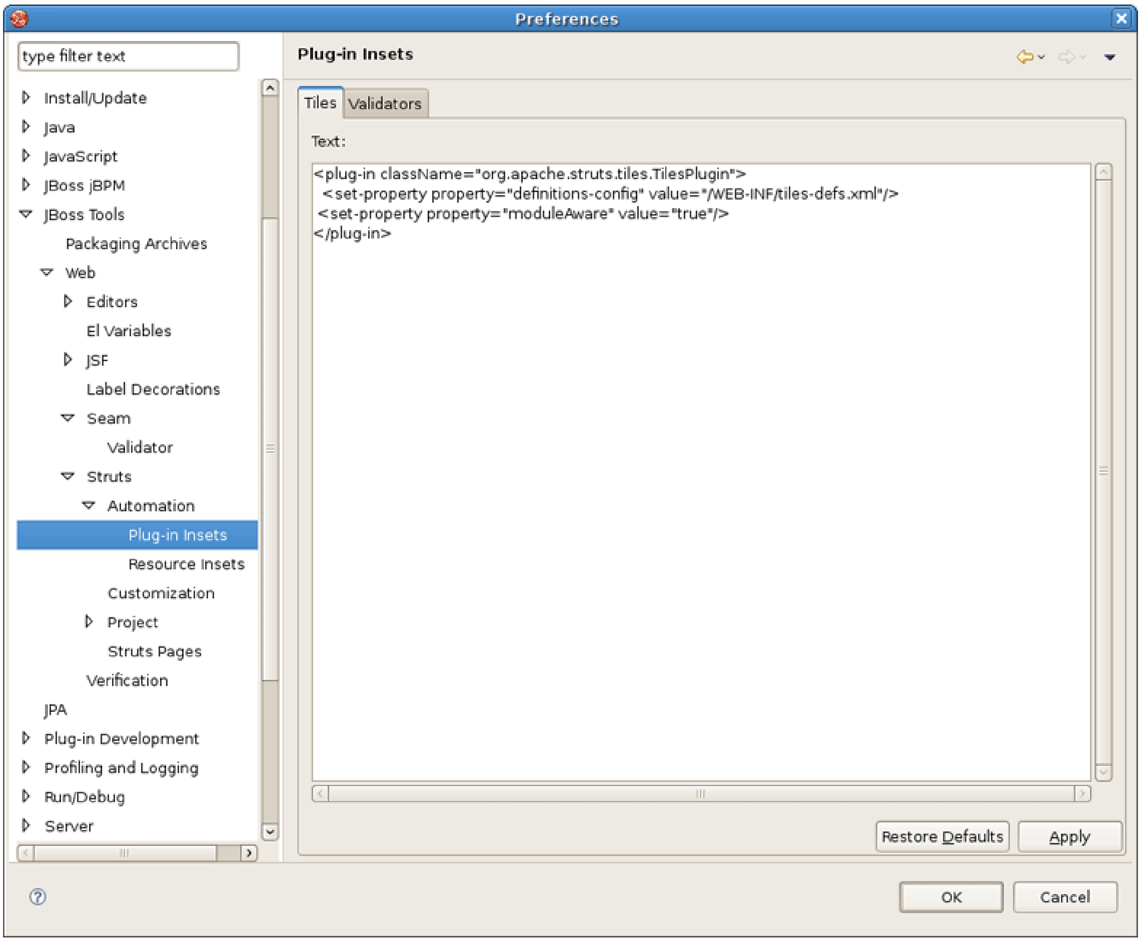
Here you can modify default text for the Tiles Struts plug-in element, the Validator Struts plug-in element, and error message resource files.

Preferences
type filter text
Install/Update
Java
JavaScript
JBoss jBPM
JBoss Tools
Packaging Archives
Web
Editors
El Variables
JSF
Label Decorations
Seam
Validator
Struts
Automation
Customization
Project
Struts Pages
Verification
JPA
Plug-in Development
Profiling and Logging
Run/Debug
Server
Service Policies
Team
OK
Cancel

Figure 8.21. Struts Automatic

8.16. Plug-in Insets

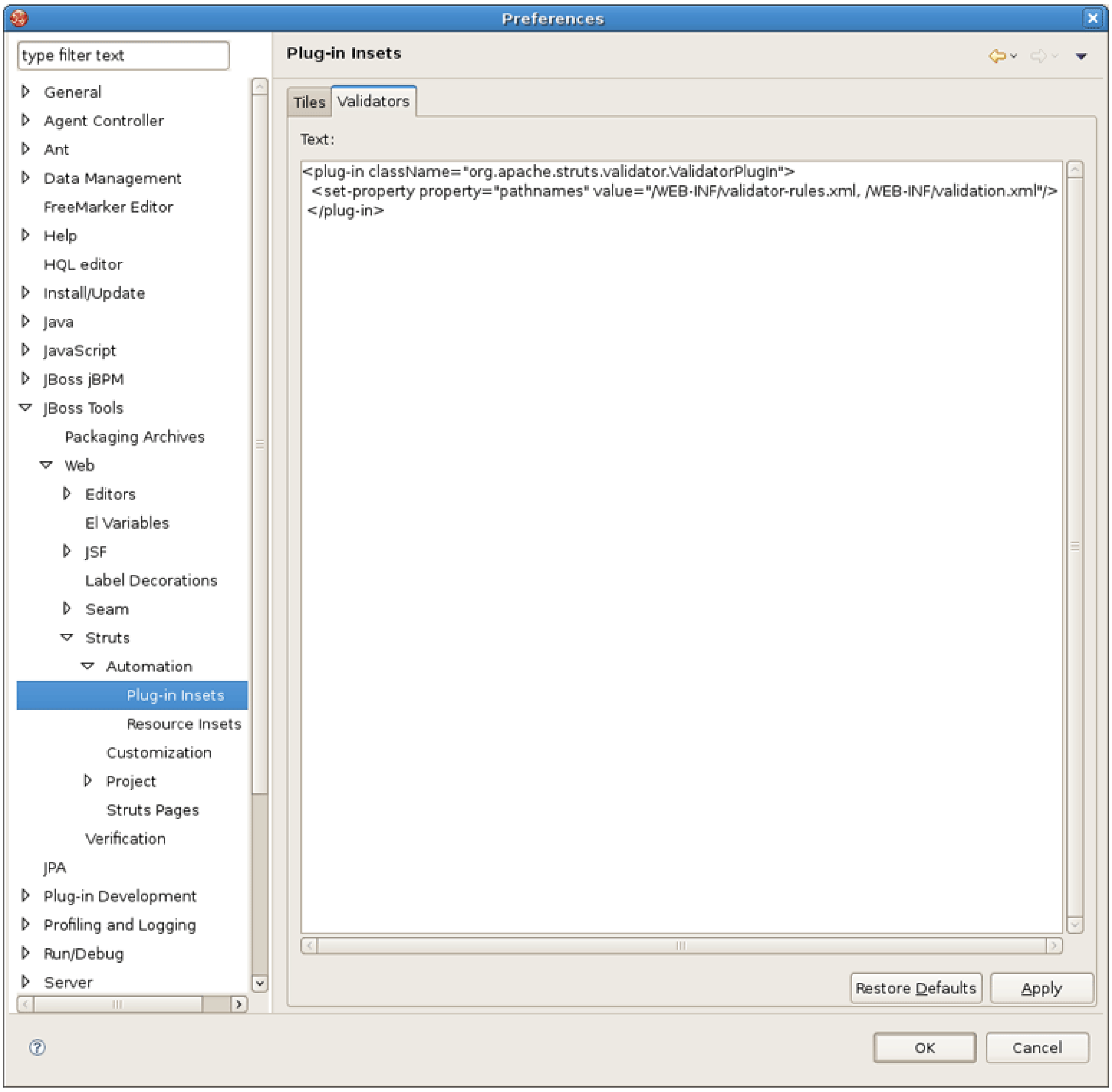
By selecting *Web > Struts > Automation > Plug-in Insets* on tab Tiles you can define a default text for tiles plugin.


Preferences
type filter text
Install/Update
Java
JavaScript
JBoss jBPM
JBoss Tools
Packaging Archives
Web
Editors
El Variables
JSF
Label Decorations
Seam
Validator
Struts
Automation
Plug-in Insets
Resource Insets
Customization
Project
Struts Pages
Verification
JPA
Plug-in Development
Profiling and Logging
Run/Debug
Server
Plug-in Insets
Tiles
Validators
Text:
Restore Defaults
Apply
OK
Cancel

```
<plug-in className="org.apache.struts.tiles.TilesPlugin">
  <set-property property="definitions-config" value="/WEB-INF/tiles-defs.xml"/>
 <set-property property="moduleAware" value="true"/>
</plug-in>
```

Figure 8.22. Plug-in Insets

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


Preferences
type filter text
General
Agent Controller
Ant
Data Management
FreeMarker Editor
Help
HQL editor
Install/Update
Java
JavaScript
JBoss jBPM
JBoss Tools
Packaging Archives
Web
Editors
El Variables
JSF
Label Decorations
Seam
Struts
Automation
Plug-in Insets
Resource Insets
Customization
Project
Struts Pages
Verification
JPA
Plug-in Development
Profiling and Logging
Run/Debug
Server
Plug-in Insets
Tiles
Validators
Text:
Restore Defaults
Apply
OK
Cancel

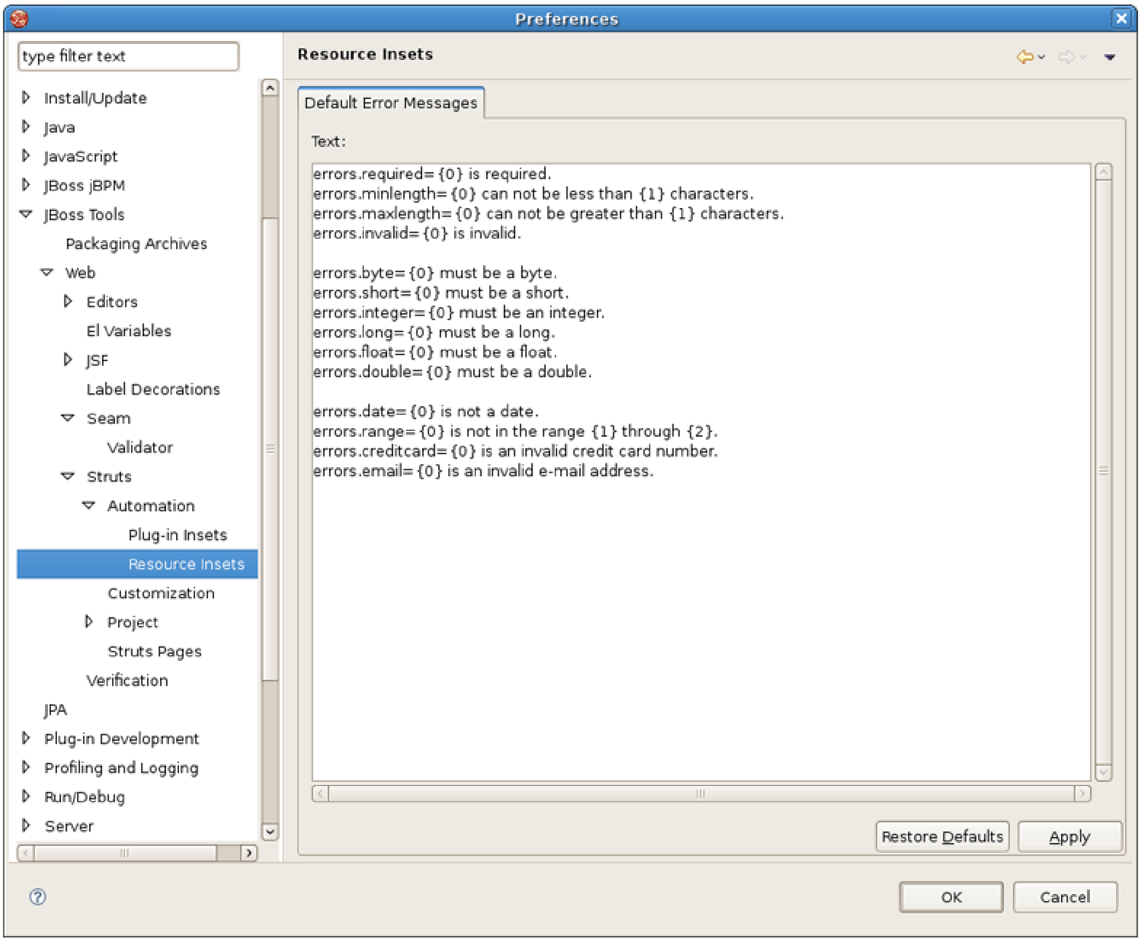
```
<plug-in className="org.apache.struts.validator.ValidatorPlugIn">
 <set-property property="pathnames" value="/WEB-INF/validator-rules.xml, /WEB-INF/validation.xml"/>
</plug-in>
```

Figure 8.23. Plug-in Insets of Validators

8.17. 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.


Preferences
type filter text
Install/Update
Java
JavaScript
JBoss jBPM
JBoss Tools
Packaging Archives
Web
Editors
El Variables
JSF
Label Decorations
Seam
Validator
Struts
Automation
Plug-in Insets
Resource Insets
Customization
Project
Struts Pages
Verification
JPA
Plug-in Development
Profiling and Logging
Run/Debug
Server
Resource Insets
Default Error Messages
Text:
Restore Defaults
Apply
OK
Cancel

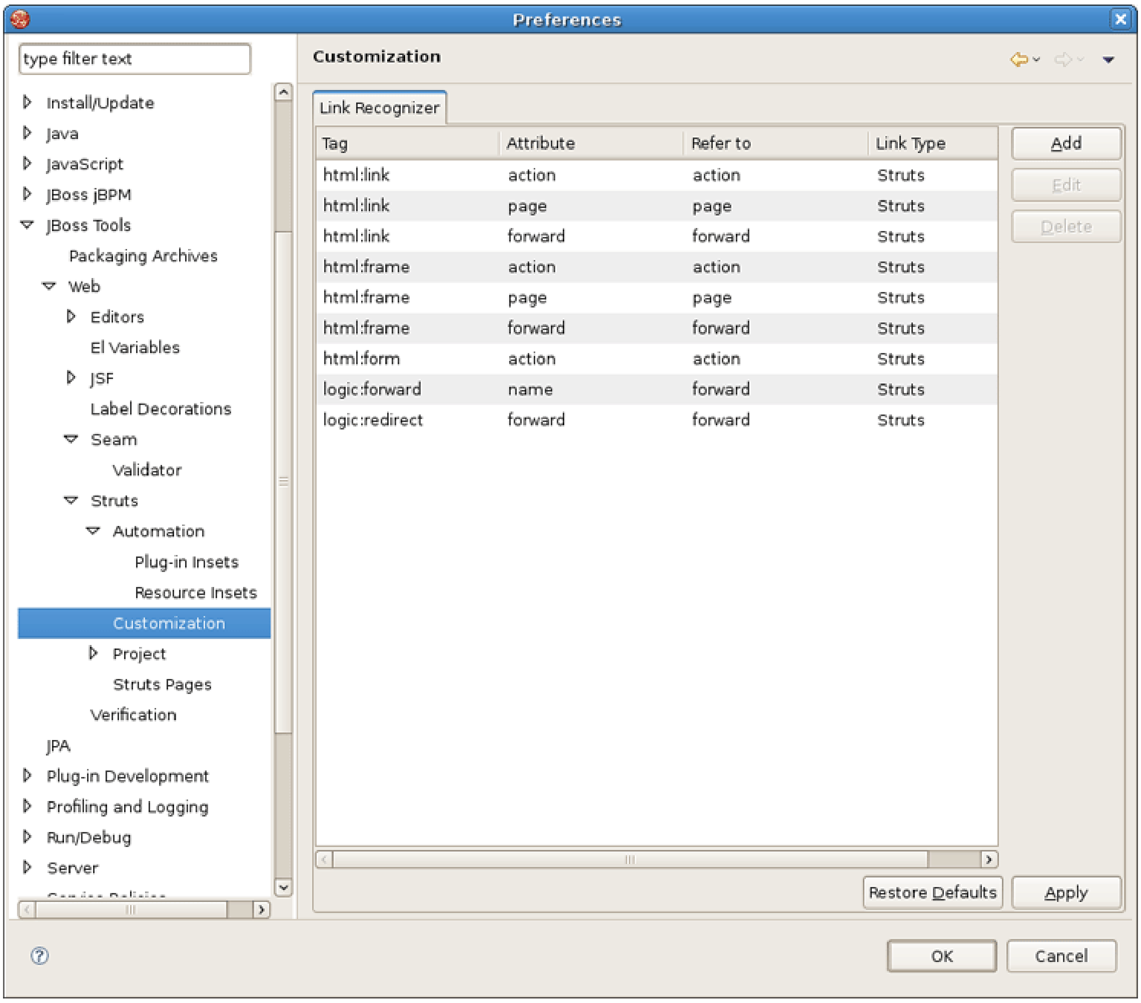
```
errors.required={0} is required.
errors.minlength={0} can not be less than {1} characters.
errors.maxlength={0} can not be greater than {1} characters.
errors.invalid={0} is invalid.

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}.
errors.creditcard={0} is an invalid credit card number.
errors.email={0} is an invalid e-mail address.
```

Figure 8.24. Resource Insets

8.18. 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.


Preferences
type filter text
Install/Update
Java
JavaScript
JBoss jBPM
JBoss Tools
Packaging Archives
Web
Editors
El Variables
JSF
Label Decorations
Seam
Validator
Struts
Automation
Plug-in Insets
Resource Insets
Customization
Project
Struts Pages
Verification
JPA
Plug-in Development
Profiling and Logging
Run/Debug
Server
Customization
Link Recognizer
Add
Edit
Delete
Restore Defaults
Apply
OK
Cancel

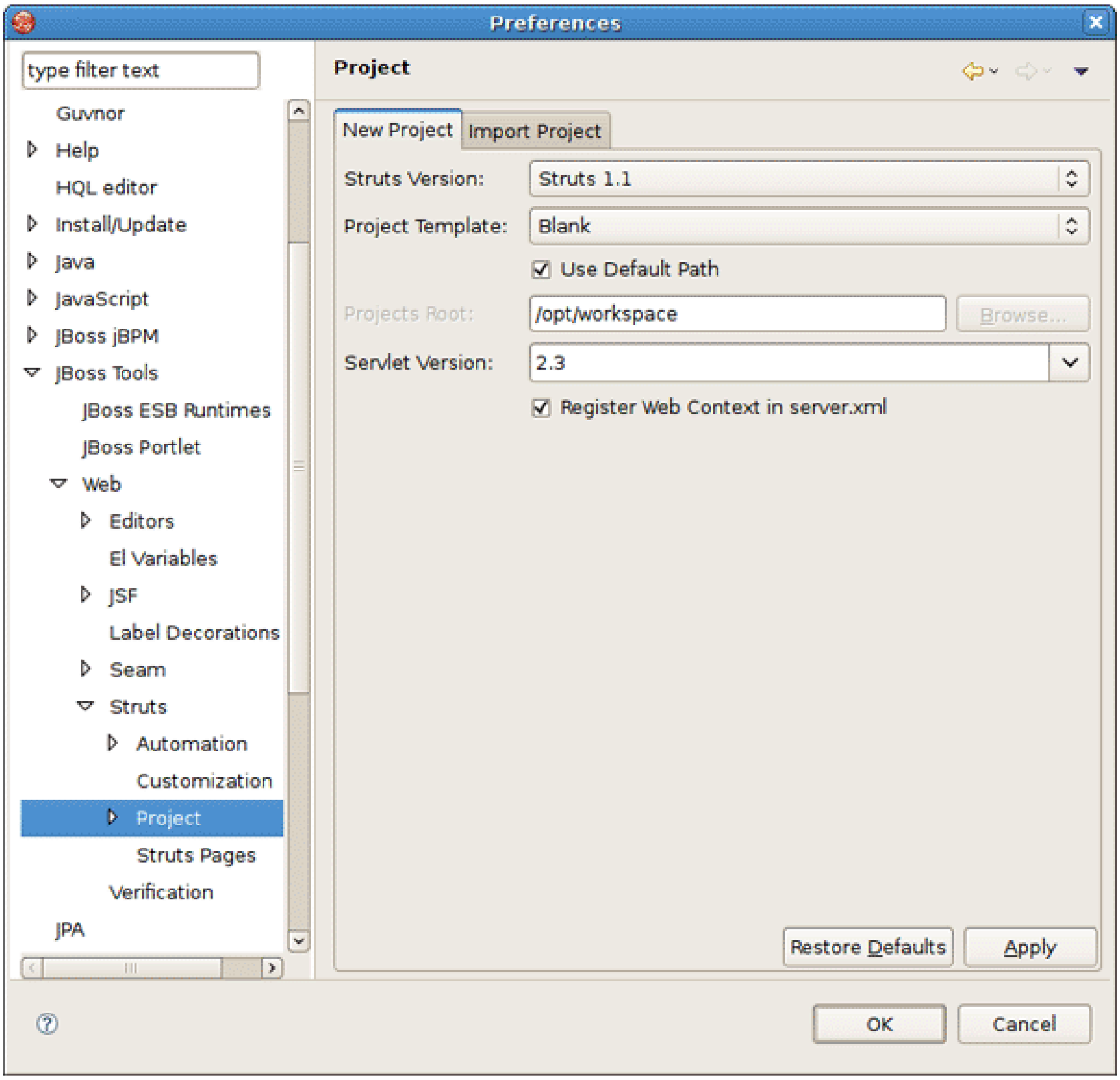
Tag	Attribute	Refer to	Link Type
html:link	action	action	Struts
html:link	page	page	Struts
html:link	forward	forward	Struts
html:frame	action	action	Struts
html:frame	page	page	Struts
html:frame	forward	forward	Struts
html:form	action	action	Struts
logic:forward	name	forward	Struts
logic:redirect	forward	forward	Struts

Figure 8.25. Struts Customization

8.19. 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.


Preferences
type filter text
Guvnor
Help
HQL editor
Install/Update
Java
JavaScript
JBoss jBPM
JBoss Tools
JBoss ESB Runtimes
JBoss Portlet
Web
Editors
El Variables
JSF
Label Decorations
Seam
Struts
Automation
Customization
Project
Struts Pages
Verification
JPA
Project
New Project
Import Project
Browse...
Restore Defaults
Apply
OK
Cancel

Struts Version: Struts 1.1

Project Template: Blank

☑ Use Default Path

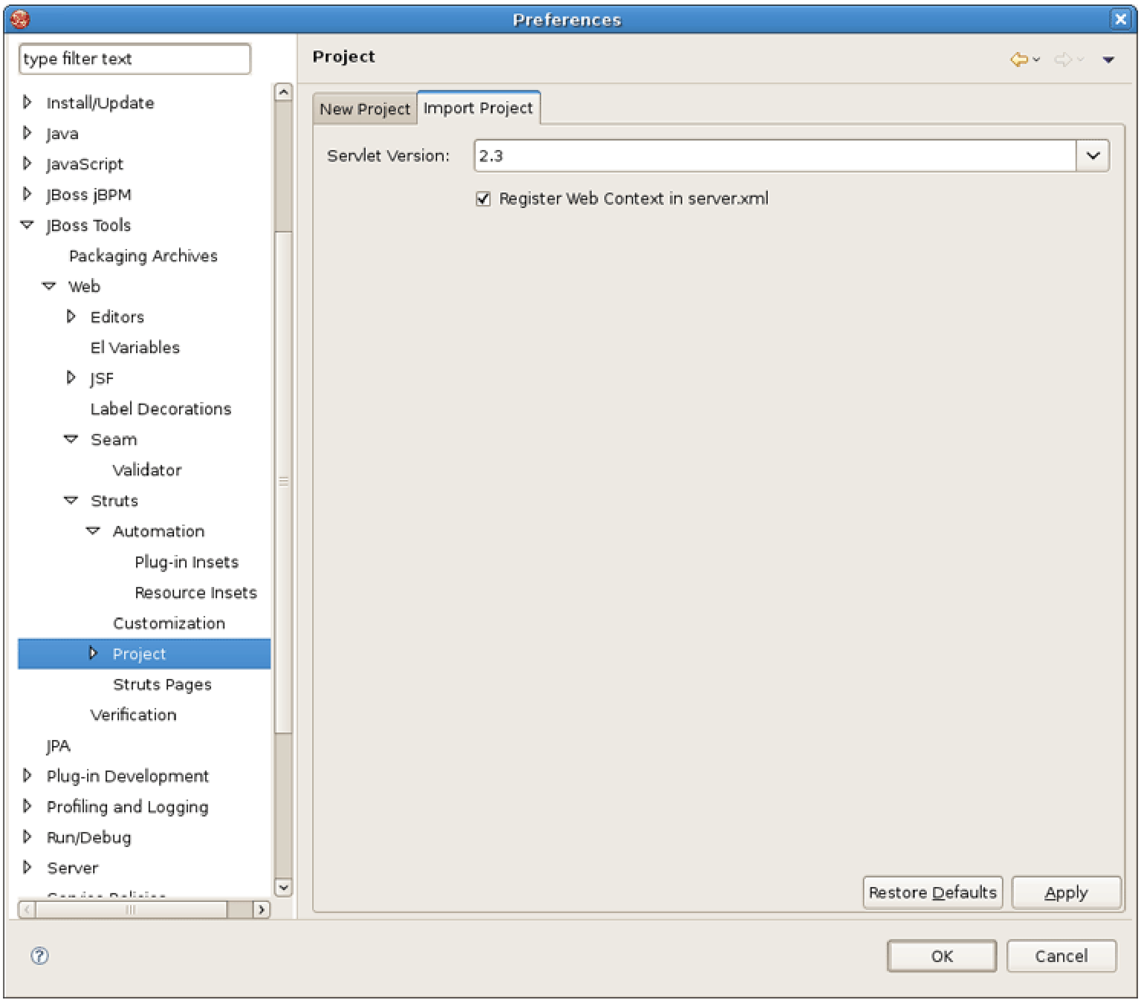
Projects Root: /opt/workspace

Servlet Version: 2.3

☑ Register Web Context in server.xml

Figure 8.26. 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.


Preferences
type filter text
Install/Update
Java
JavaScript
JBoss jBPM
JBoss Tools
Packaging Archives
Web
Editors
El Variables
JSF
Label Decorations
Seam
Validator
Struts
Automation
Plug-in Insets
Resource Insets
Customization
Project
Struts Pages
Verification
JPA
Plug-in Development
Profiling and Logging
Run/Debug
Server
Project
New Project
Import Project
Restore Defaults
Apply
OK
Cancel

Servlet Version: 2.3

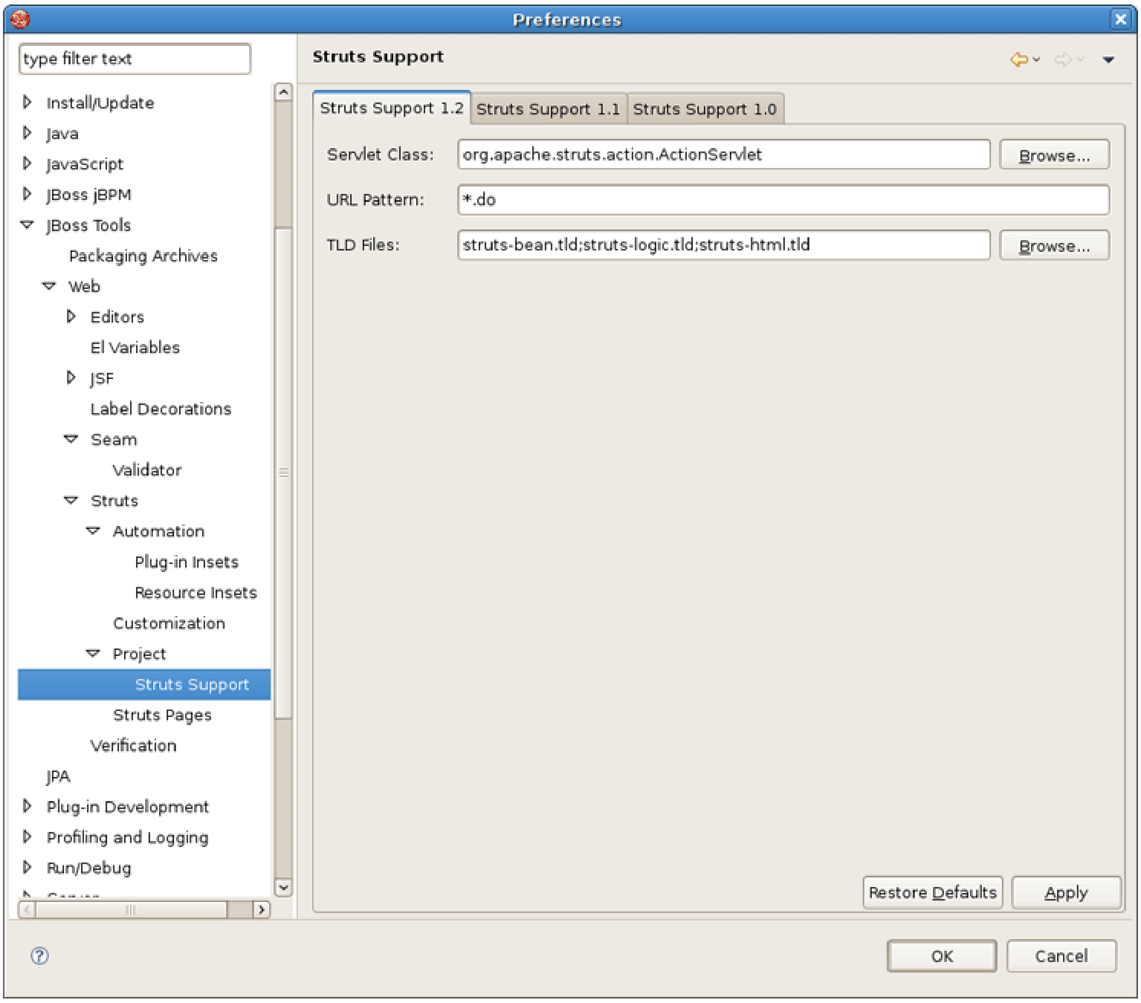
☑ Register Web Context in server.xml

Figure 8.27. Import Struts Pages

8.20. 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.


Preferences
type filter text
Install/Update
Java
JavaScript
JBoss jBPM
JBoss Tools
Packaging Archives
Web
Editors
El Variables
JSF
Label Decorations
Seam
Validator
Struts
Automation
Plug-in Insets
Resource Insets
Customization
Project
Struts Support
Struts Pages
Verification
JPA
Plug-in Development
Profiling and Logging
Run/Debug
Struts Support
Struts Support 1.2
Struts Support 1.1
Struts Support 1.0
Browse...
Browse...
Restore Defaults
Apply
OK
Cancel

Servlet Class: org.apache.struts.action.ActionServlet

URL Pattern: *.do

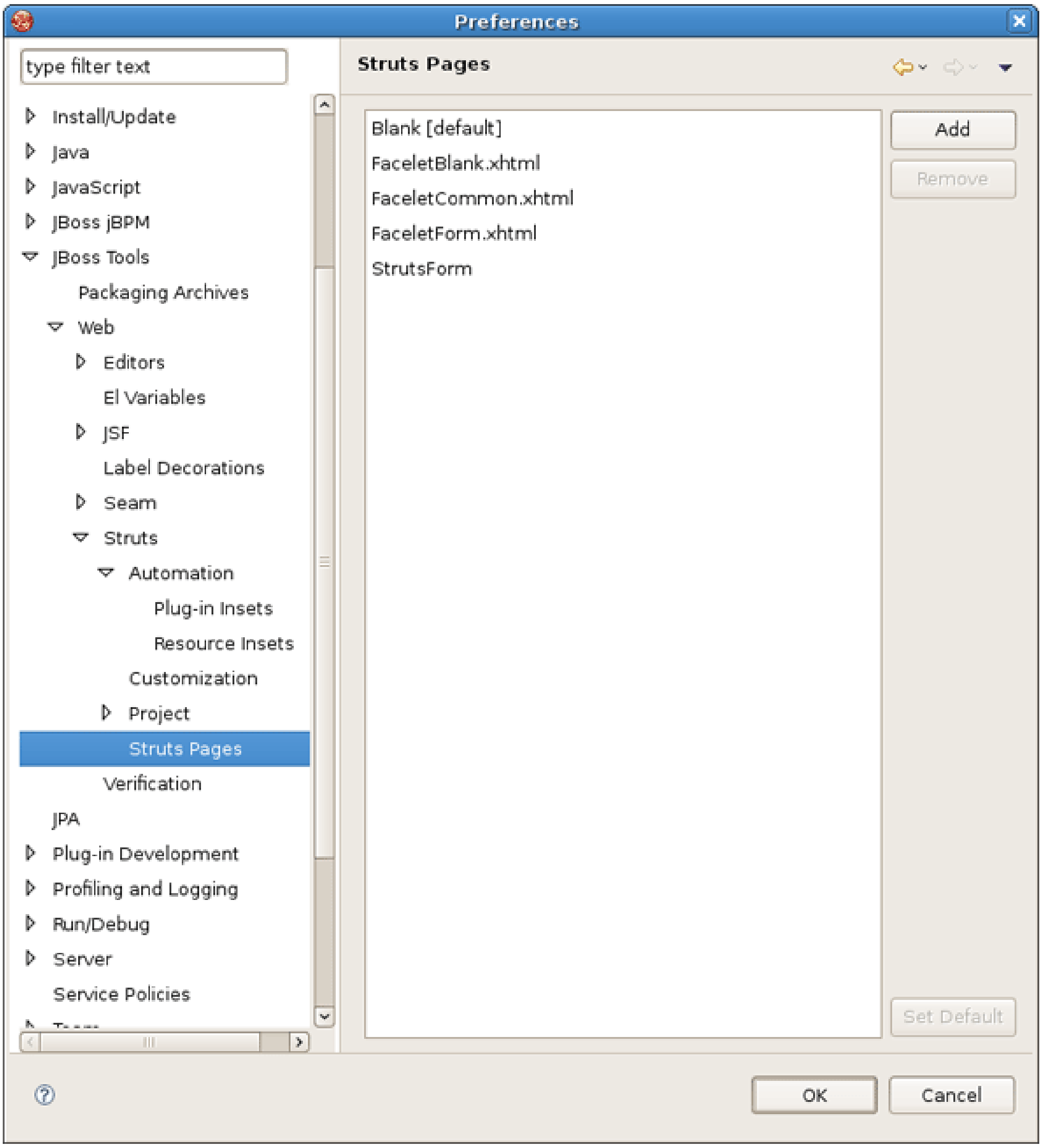
TLD Files: struts-bean.tld;struts-logic.tld;struts-html.tld

Figure 8.28. Struts Support

8.21. 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.

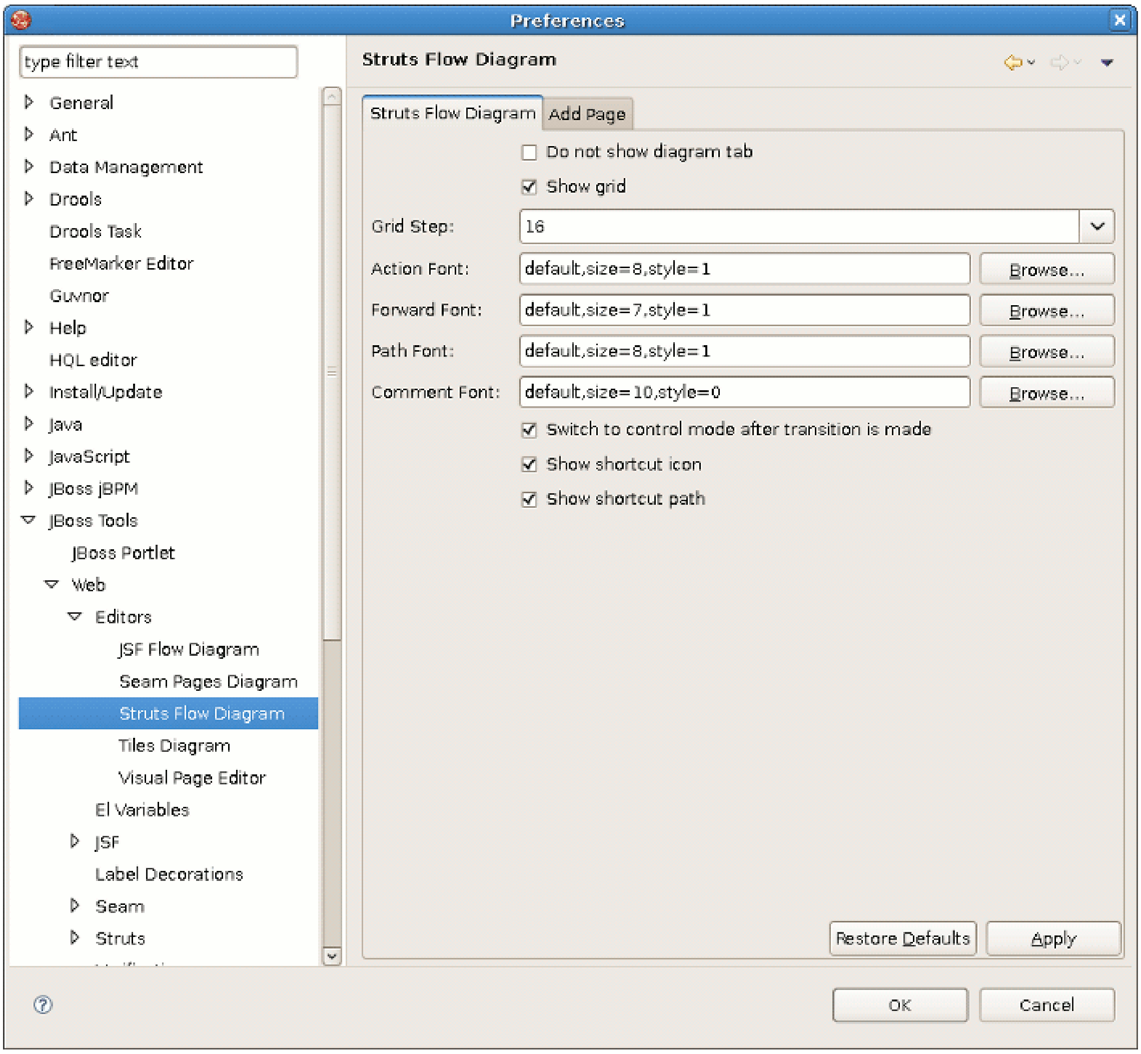

Preferences
type filter text
Install/Update
Java
JavaScript
JBoss jBPM
JBoss Tools
Packaging Archives
Web
Editors
El Variables
JSF
Label Decorations
Seam
Struts
Automation
Plug-in Insets
Resource Insets
Customization
Project
Struts Pages
Verification
JPA
Plug-in Development
Profiling and Logging
Run/Debug
Server
Service Policies
Struts Pages
Add
Remove
Set Default
OK
Cancel

Blank [default]
FaceletBlank.xhtml
FaceletCommon.xhtml
FaceletForm.xhtml
StrutsForm

Figure 8.29. Struts Pages

8.22. 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.


Preferences
type filter text
General
Ant
Data Management
Drools
Drools Task
FreeMarker Editor
Guvnor
Help
HQL editor
Install/Update
Java
JavaScript
JBoss jBPM
JBoss Tools
JBoss Portlet
Web
Editors
JSF Flow Diagram
Seam Pages Diagram
Struts Flow Diagram
Tiles Diagram
Visual Page Editor
El Variables
JSF
Label Decorations
Seam
Struts
Struts Flow Diagram
Struts Flow Diagram
Add Page
Browse...
Browse...
Browse...
Browse...
Restore Defaults
Apply
OK
Cancel

☐ Do not show diagram tab

☑ Show grid

Grid Step: 16

Action Font: default,size=8,style=1

Forward Font: default,size=7,style=1

Path Font: default,size=8,style=1

Comment Font: default,size=10,style=0

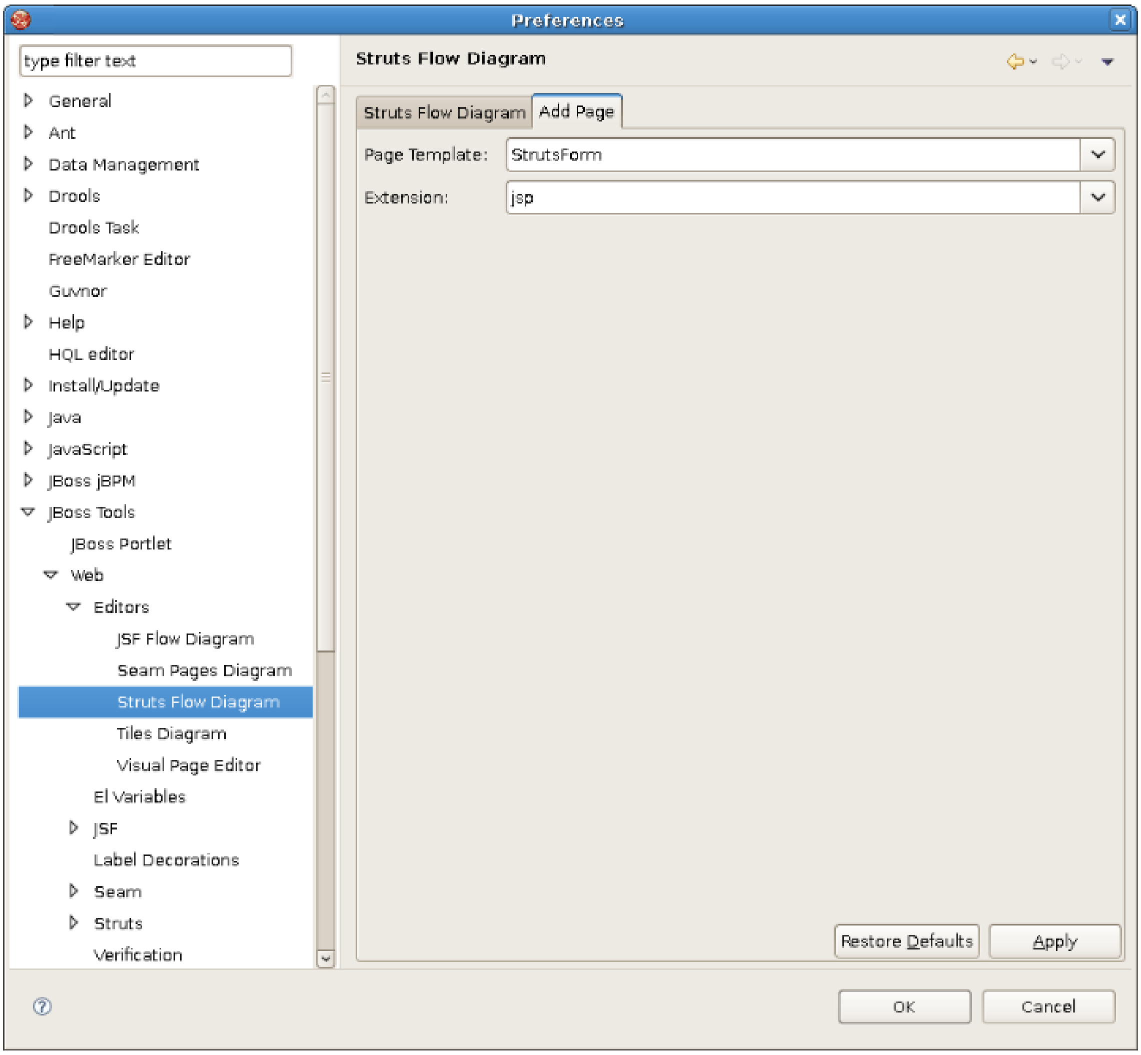
☑ Switch to control mode after transition is made

☑ Show shortcut icon

☑ Show shortcut path

Figure 8.30. 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.


Preferences
type filter text
General
Ant
Data Management
Drools
Drools Task
FreeMarker Editor
Guvnor
Help
HQL editor
Install/Update
Java
JavaScript
JBoss jBPM
JBoss Tools
JBoss Portlet
Web
Editors
JSF Flow Diagram
Seam Pages Diagram
Struts Flow Diagram
Tiles Diagram
Visual Page Editor
El Variables
JSF
Label Decorations
Seam
Struts
Verification
Struts Flow Diagram
Struts Flow Diagram
Add Page
Restore Defaults
Apply
OK
Cancel

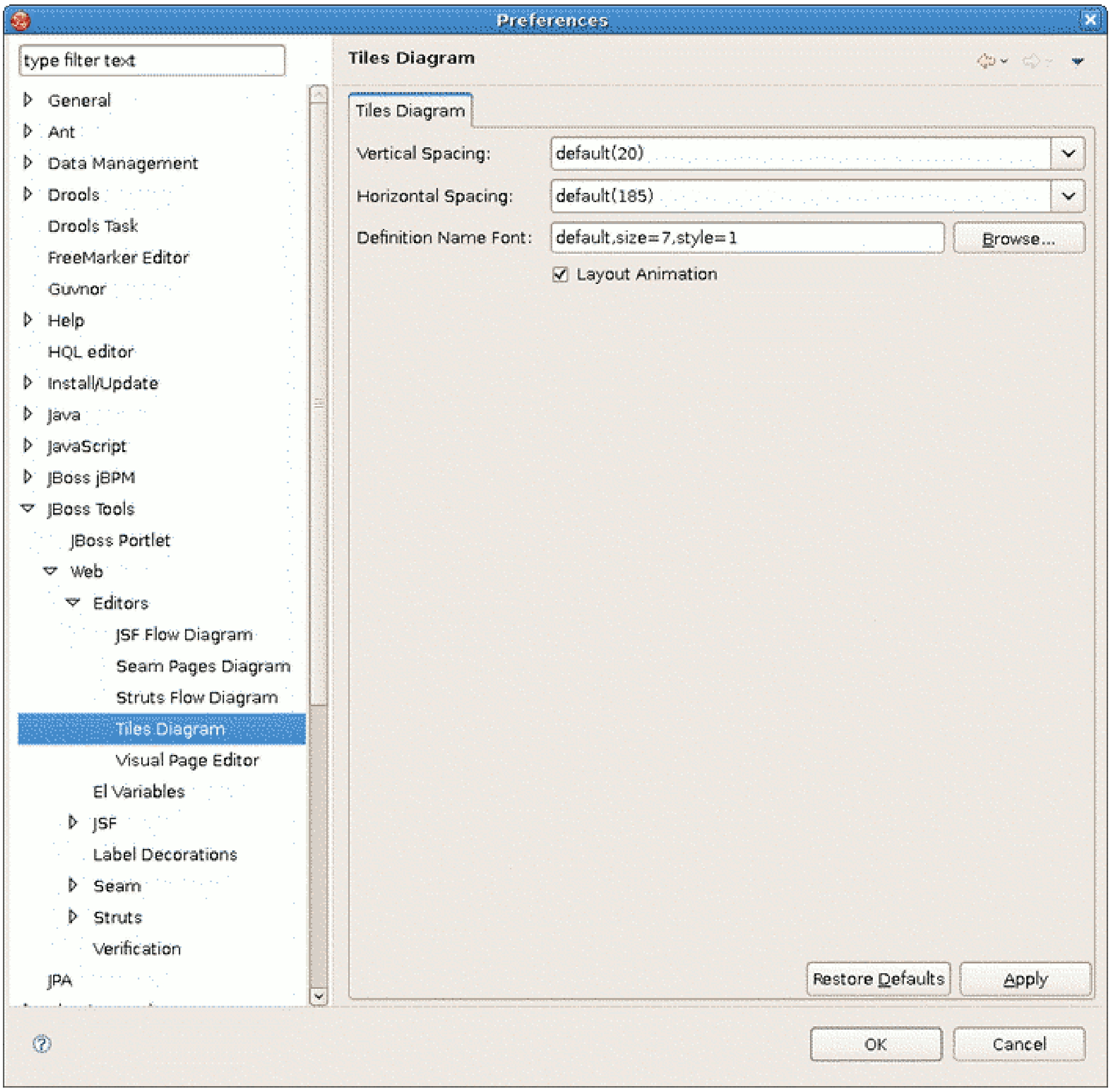
Page Template: StrutsForm

Extension: jsp

Figure 8.31. Adding Page

8.23. 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
General
Ant
Data Management
Drools
Drools Task
FreeMarker Editor
Guvnor
Help
HQL editor
Install/Update
Java
JavaScript
JBoss jBPM
JBoss Tools
JBoss Portlet
Web
Editors
JSF Flow Diagram
Seam Pages Diagram
Struts Flow Diagram
Tiles Diagram
Visual Page Editor
El Variables
JSF
Label Decorations
Seam
Struts
Verification
JPA
Tiles Diagram
Tiles Diagram
Browse...
Restore Defaults
Apply
OK
Cancel

Vertical Spacing: default(20)

Horizontal Spacing: default(185)

Definition Name Font: default,size=7,style=1

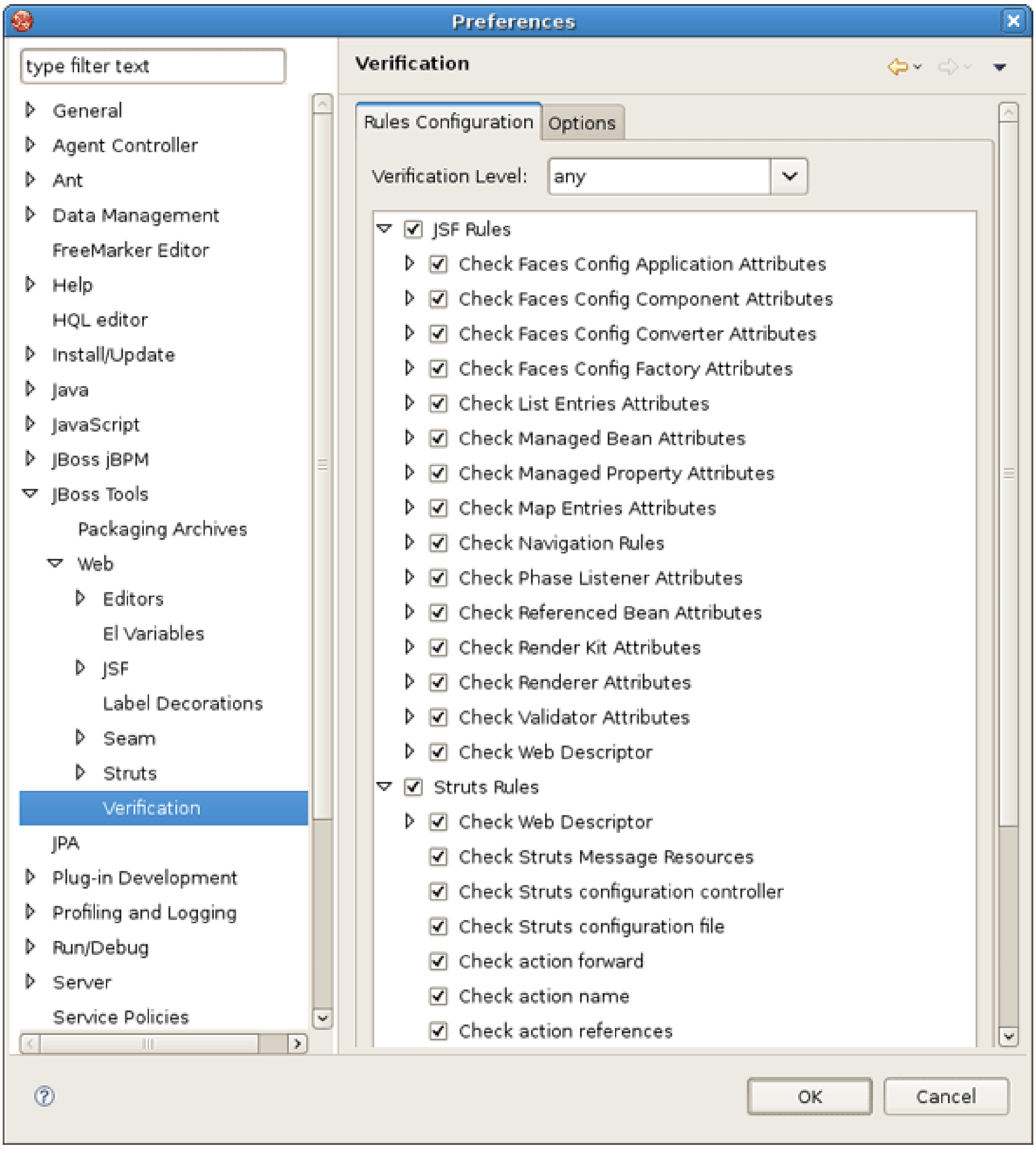
☑ Layout Animation

Figure 8.32. Title Diagram

8.24. Verification

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

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


Preferences
type filter text
General
Agent Controller
Ant
Data Management
FreeMarker Editor
Help
HQL editor
Install/Update
Java
JavaScript
JBoss jBPM
JBoss Tools
Packaging Archives
Web
Editors
El Variables
JSF
Label Decorations
Seam
Struts
Verification
JPA
Plug-in Development
Profiling and Logging
Run/Debug
Server
Service Policies
Verification
Rules Configuration
Options
OK
Cancel

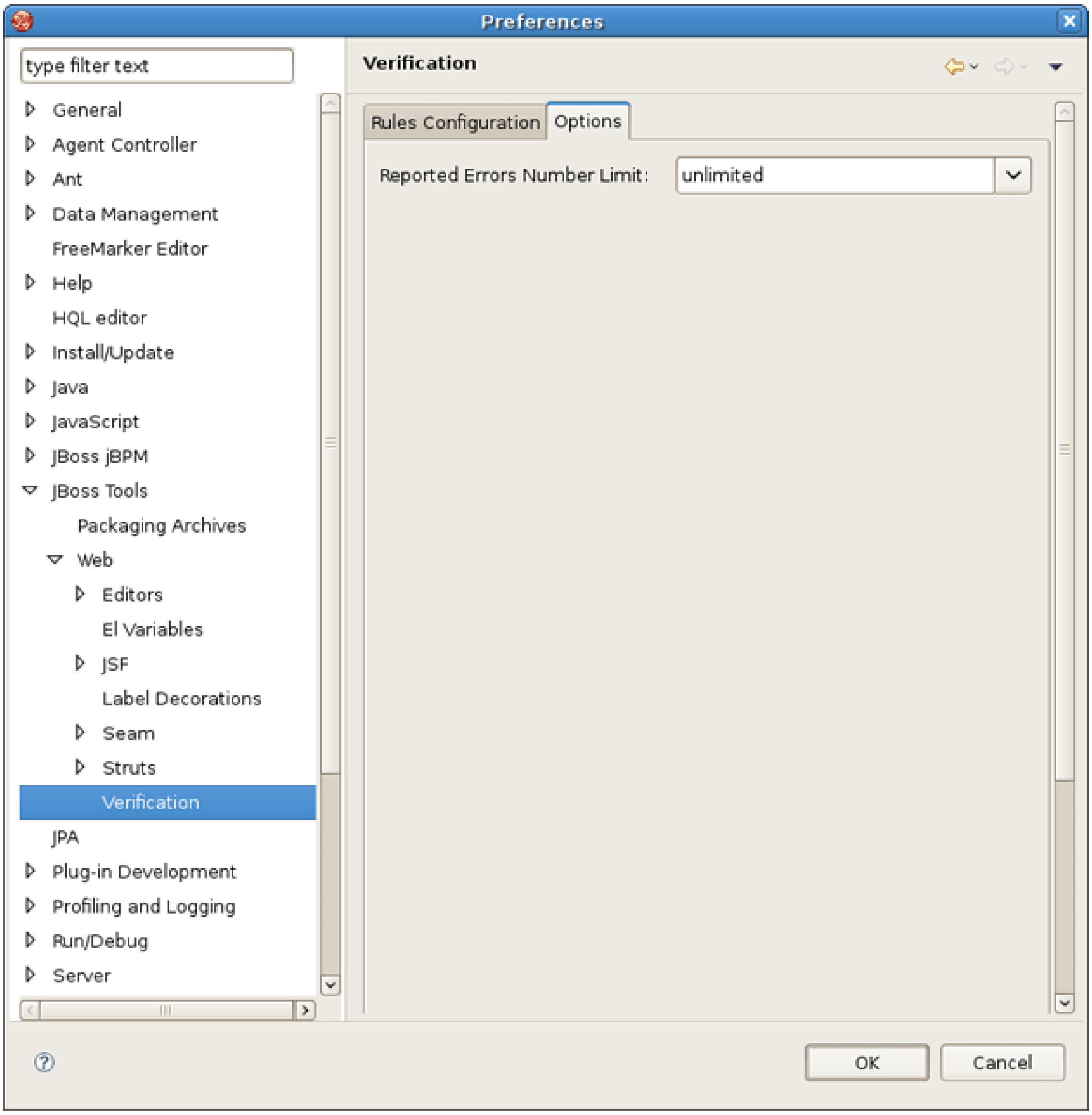
Verification Level: any

☑ JSF Rules
- ☑ Check Faces Config Application Attributes
- ☑ Check Faces Config Component Attributes
- ☑ Check Faces Config Converter Attributes
- ☑ Check Faces Config Factory Attributes
- ☑ Check List Entries Attributes
- ☑ Check Managed Bean Attributes
- ☑ Check Managed Property Attributes
- ☑ Check Map Entries Attributes
- ☑ Check Navigation Rules
- ☑ Check Phase Listener Attributes
- ☑ Check Referenced Bean Attributes
- ☑ Check Render Kit Attributes
- ☑ Check Renderer Attributes
- ☑ Check Validator Attributes
- ☑ Check Web Descriptor

☑ Struts Rules
- ☑ Check Web Descriptor
- ☑ Check Struts Message Resources
- ☑ Check Struts configuration controller
- ☑ Check Struts configuration file
- ☑ Check action forward
- ☑ Check action name
- ☑ Check action references

Figure 8.33. Verification

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



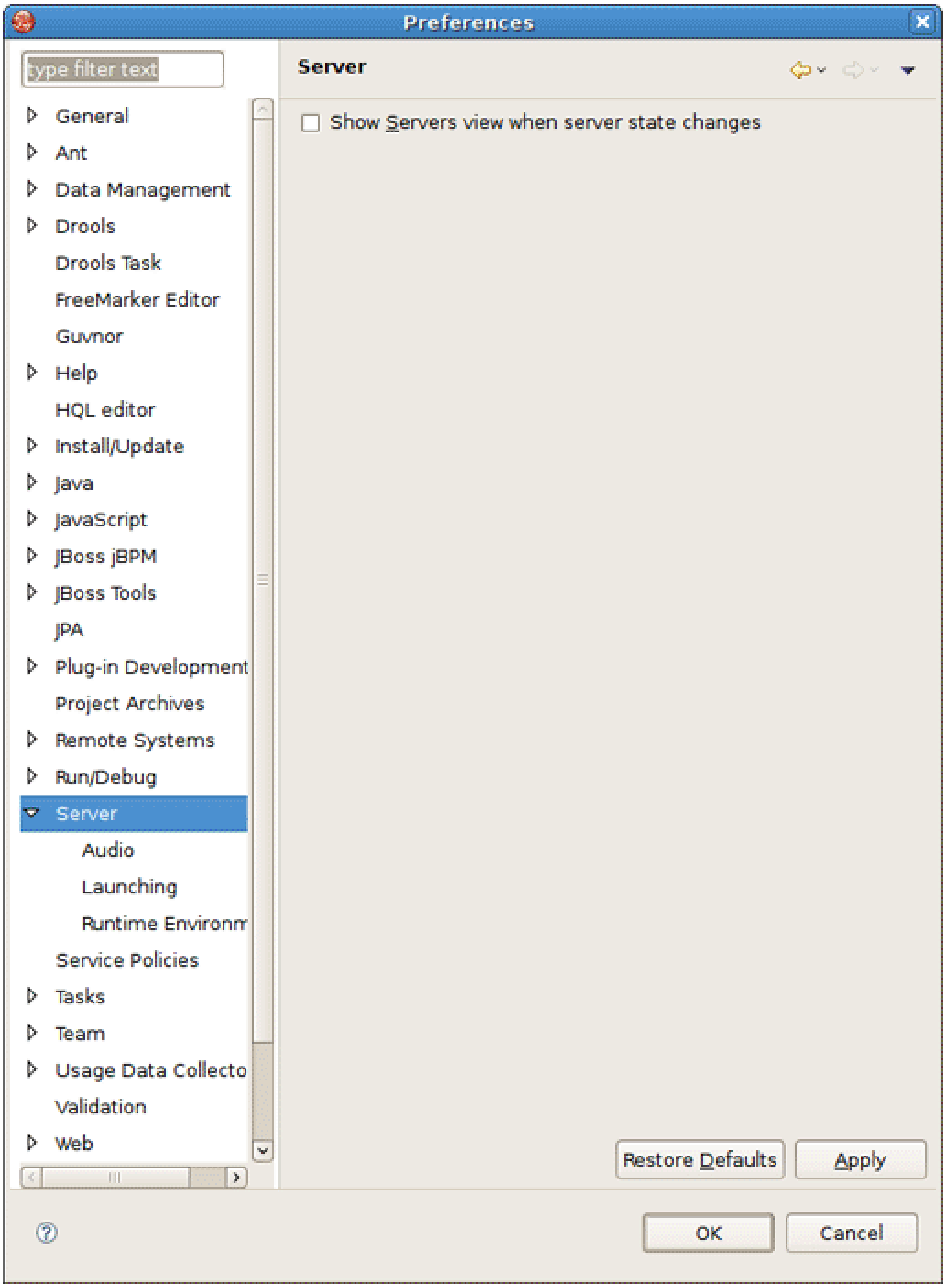
Preferences
type filter text
General
Agent Controller
Ant
Data Management
FreeMarker Editor
Help
HQL editor
Install/Update
Java
JavaScript
JBoss jBPM
JBoss Tools
Packaging Archives
Web
Editors
El Variables
JSF
Label Decorations
Seam
Struts
Verification
JPA
Plug-in Development
Profiling and Logging
Run/Debug
Server
Verification
Rules Configuration
Options
OK
Cancel

Reported Errors Number Limit: unlimited

Figure 8.34. Options of Verification

8.25. Server Preferences

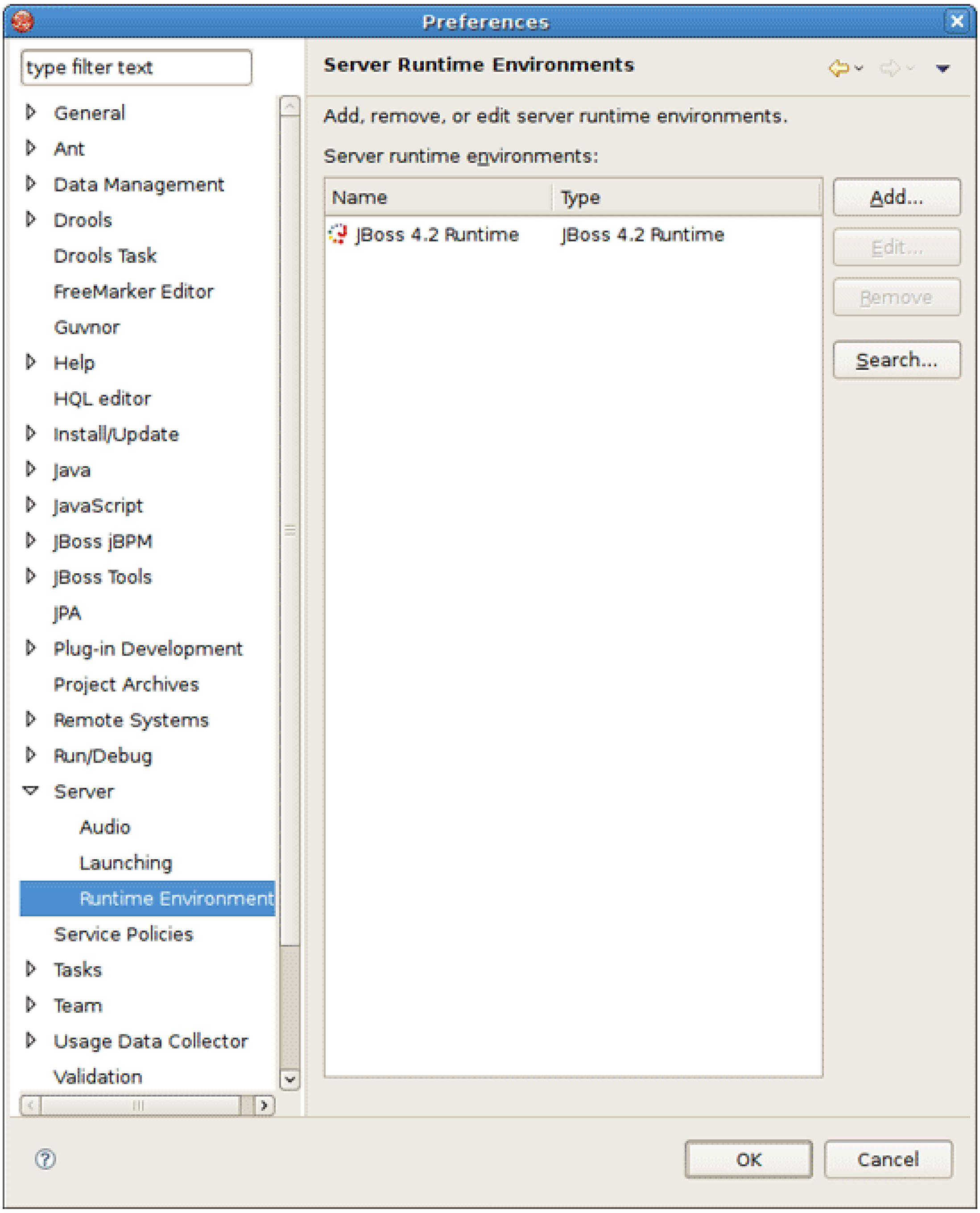
Preferences for JBoss Server and other servers can be changed on the *Server* page.


Preferences
type filter text
General
Ant
Data Management
Drools
Drools Task
FreeMarker Editor
Guvnor
Help
HQL editor
Install/Update
Java
JavaScript
JBoss jBPM
JBoss Tools
JPA
Plug-in Development
Project Archives
Remote Systems
Run/Debug
Server
Audio
Launching
Runtime Environm
Service Policies
Tasks
Team
Usage Data Collecto
Validation
Web
Server
Restore Defaults
Apply
OK
Cancel

☐ Show Servers view when server state changes

Figure 8.35. Server Preferences

On the *Server > Runtime Environments* page you can add new or modify already defined Server Runtime.


Preferences
type filter text
General
Ant
Data Management
Drools
Drools Task
FreeMarker Editor
Guvnor
Help
HQL editor
Install/Update
Java
JavaScript
JBoss jBPM
JBoss Tools
JPA
Plug-in Development
Project Archives
Remote Systems
Run/Debug
Server
Audio
Launching
Runtime Environment
Service Policies
Tasks
Team
Usage Data Collector
Validation
Server Runtime Environments
Add...
Edit...
Remove
Search...
OK
Cancel

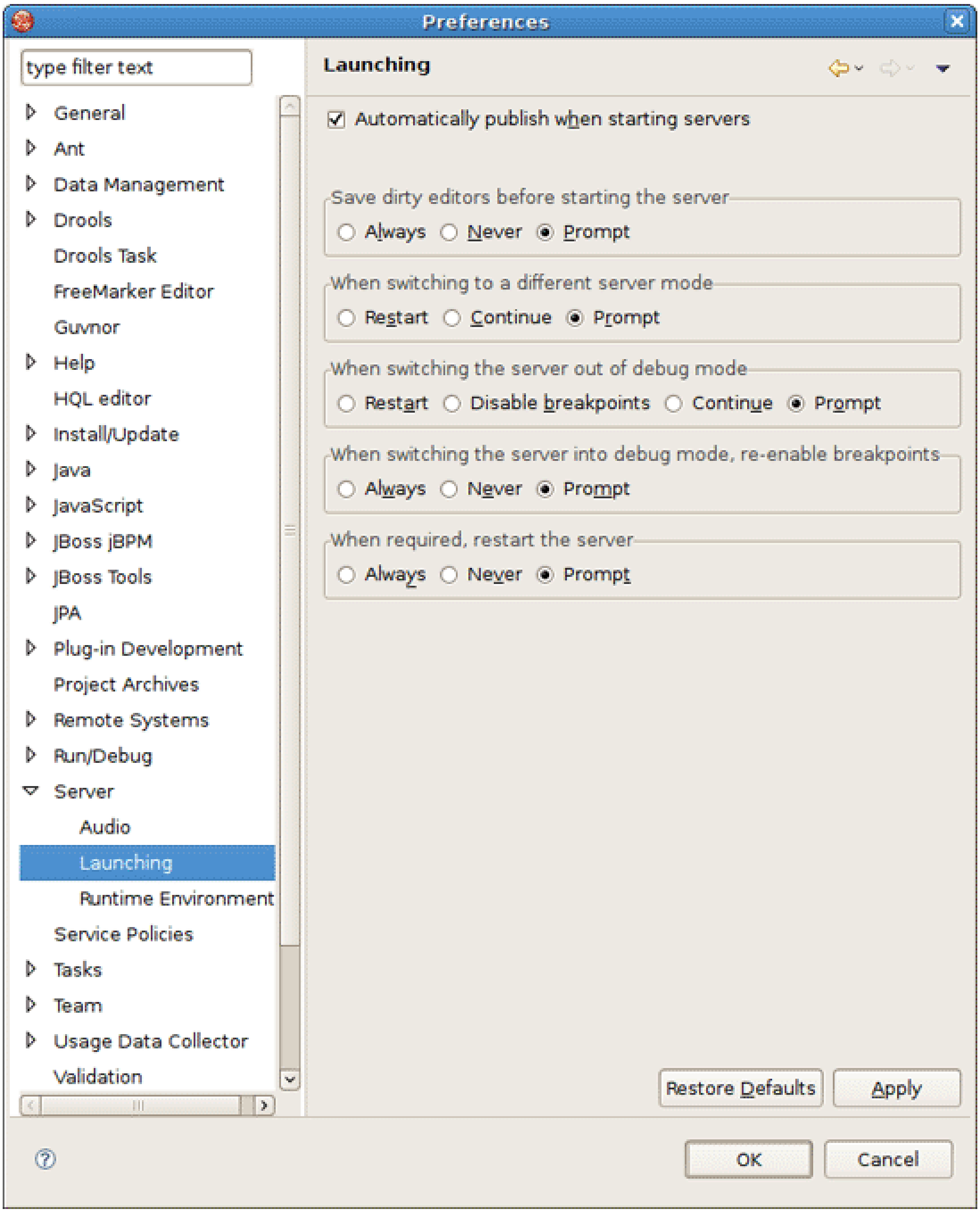
Add, remove, or edit server runtime environments.

Server runtime environments:

Name	Type
JBoss 4.2 Runtime	JBoss 4.2 Runtime

Figure 8.36. Runtime Environments

Server Launching preferences can be configured on the *Server > Launching* page.



Launching

☑ Automatically publish when starting servers

Save dirty editors before starting the server
○ Always ○ Never ● Prompt

When switching to a different server mode
○ Restart ○ Continue ● Prompt

When switching the server out of debug mode
○ Restart ○ Disable breakpoints ○ Continue ● Prompt

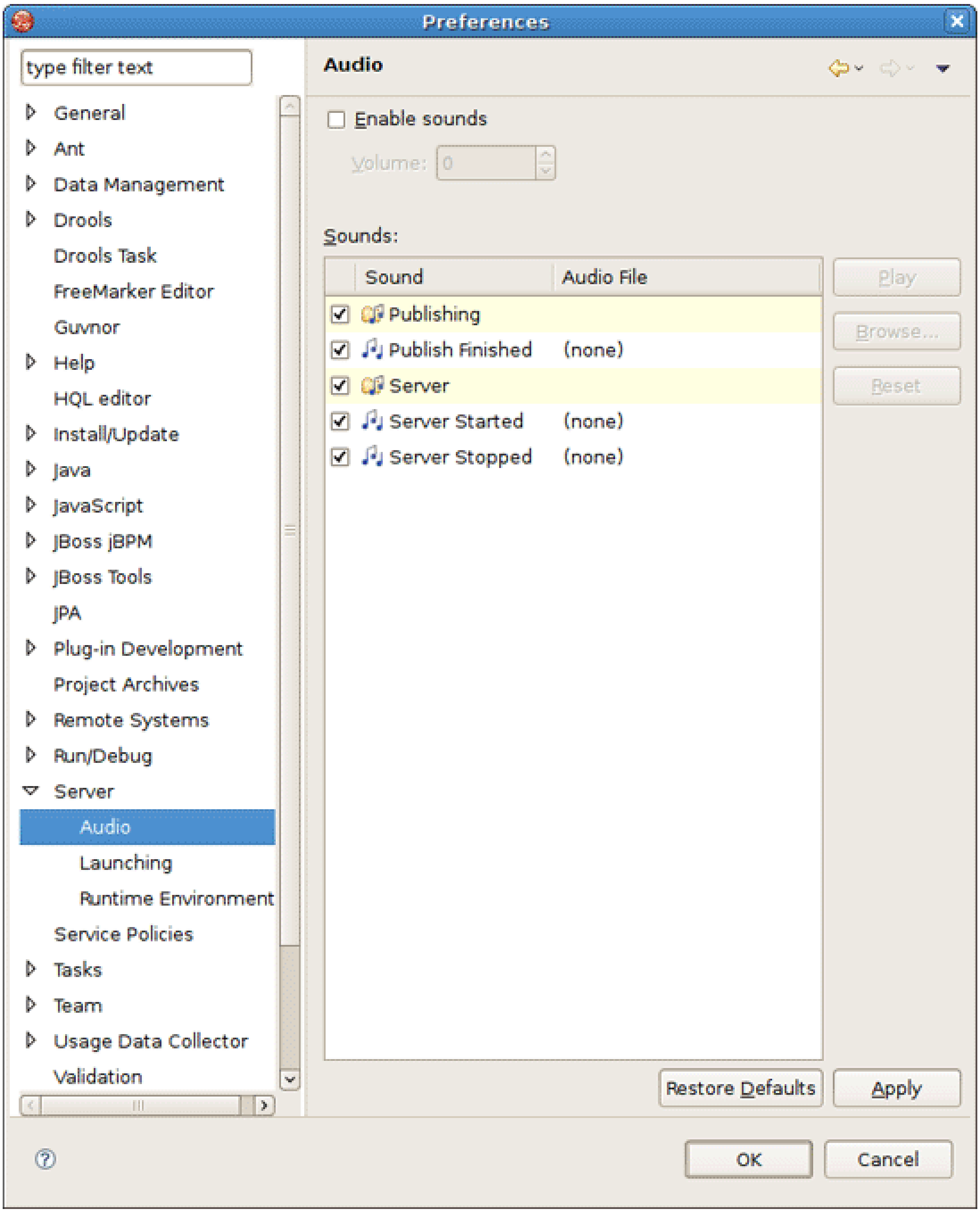
When switching the server into debug mode, re-enable breakpoints
○ Always ○ Never ● Prompt

When required, restart the server
○ Always ○ Never ● Prompt

Preferences
type filter text
General
Ant
Data Management
Drools
Drools Task
FreeMarker Editor
Guvnor
Help
HQL editor
Install/Update
Java
JavaScript
JBoss jBPM
JBoss Tools
JPA
Plug-in Development
Project Archives
Remote Systems
Run/Debug
Server
Audio
Launching
Runtime Environment
Service Policies
Tasks
Team
Usage Data Collector
Validation
Restore Defaults
Apply
OK
Cancel

Figure 8.37. Server Launching Preferences

Going to *Server > Audio* you can enable/disable the sound notification for different Server states and actions and set the sound volume as well.


Preferences
type filter text
General
Ant
Data Management
Drools
Drools Task
FreeMarker Editor
Guvnor
Help
HQL editor
Install/Update
Java
JavaScript
JBoss jBPM
JBoss Tools
JPA
Plug-in Development
Project Archives
Remote Systems
Run/Debug
Server
Audio
Launching
Runtime Environment
Service Policies
Tasks
Team
Usage Data Collector
Validation
Audio
Enable sounds
Volume:
Sounds:
Play
Browse...
Reset
Restore Defaults
Apply
OK
Cancel

☐ Enable sounds

Volume: 0

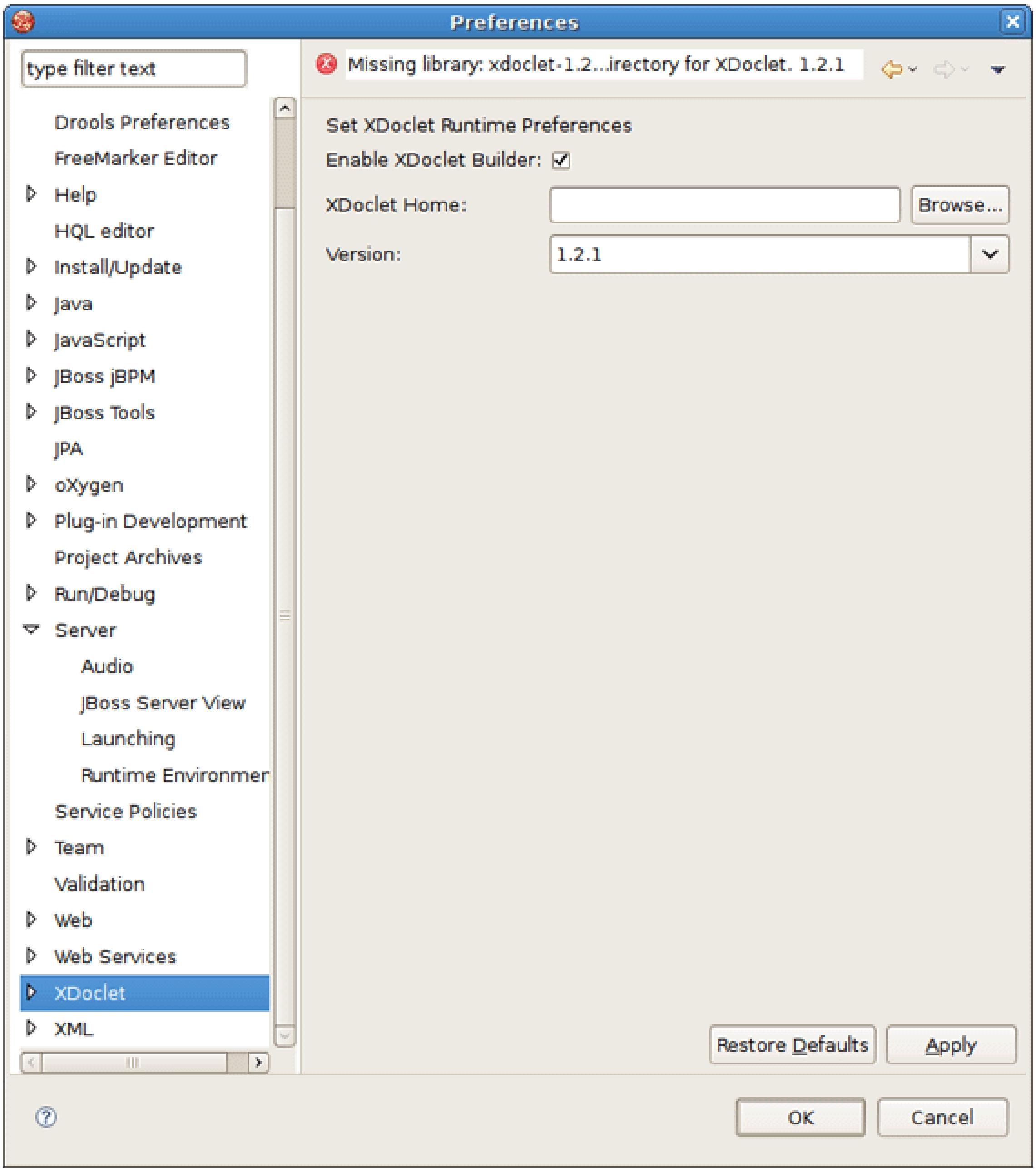
Sound	Audio File
☑ Publishing	
☑ Publish Finished	(none)
☑ Server	
☑ Server Started	(none)
☑ Server Stopped	(none)

Figure 8.38. Sound Notification Adjustment

8.26. XDoclet

The preferences for XDoclet can be changed if you click *XDoclet* on the left navigation bar.

On the *XDoclet* screen it's possible to enable/disable XDoclet builder by checking proper box, specify XDoclet home and determine XDoclet module version as well.


Preferences
type filter text
Drools Preferences
FreeMarker Editor
Help
HQL editor
Install/Update
Java
JavaScript
JBoss jBPM
JBoss Tools
JPA
oXygen
Plug-in Development
Project Archives
Run/Debug
Server
Audio
JBoss Server View
Launching
Runtime Environmer
Service Policies
Team
Validation
Web
Web Services
XDoclet
XML
Browse...
Restore Defaults
Apply
OK
Cancel

Missing library: xdoclet-1.2...irectory for XDoclet. 1.2.1

Set XDoclet Runtime Preferences

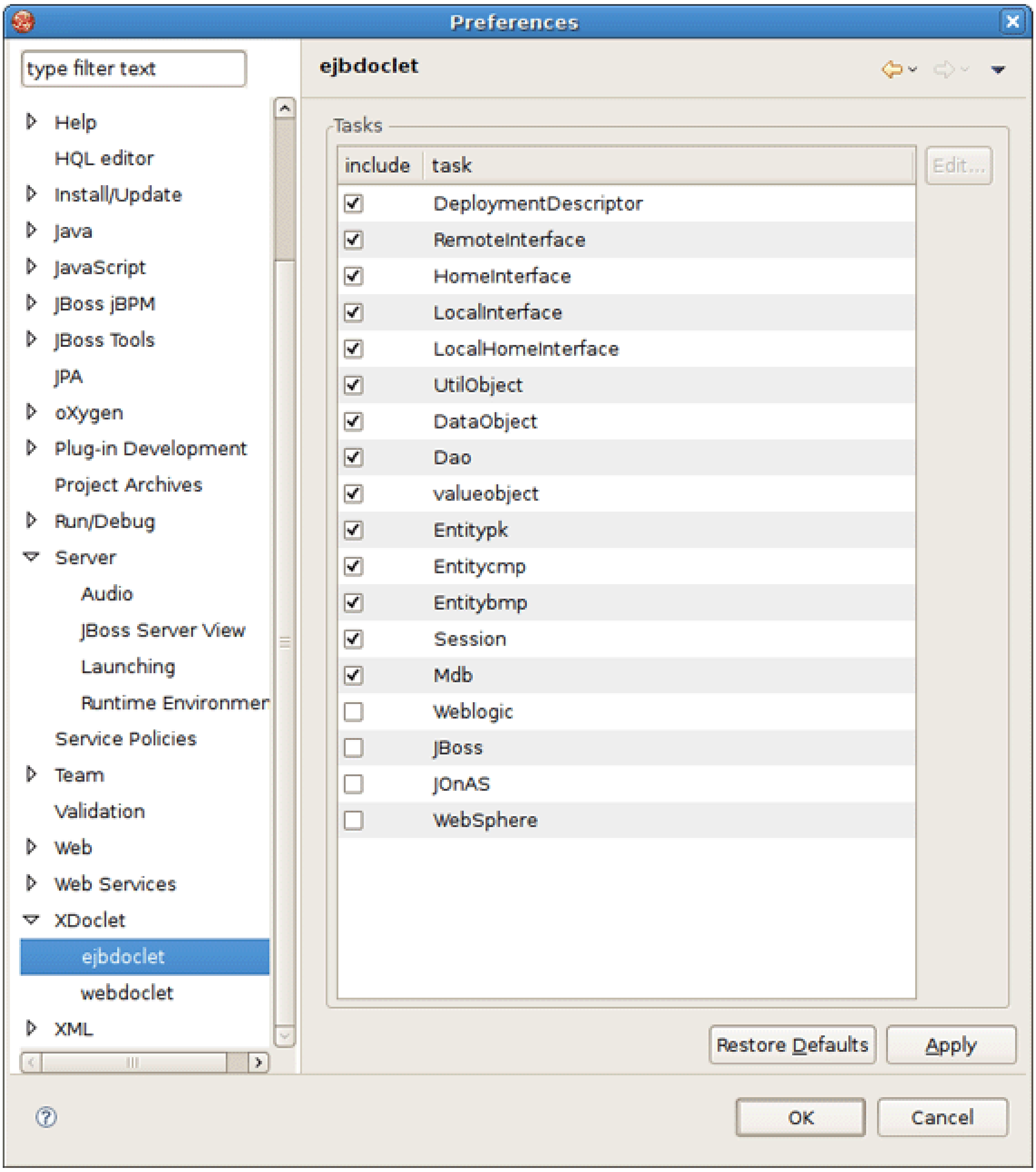
Enable XDoclet Builder: ☑

XDoclet Home:

Version: 1.2.1

Figure 8.39. XDoclet Runtime Preferences Page

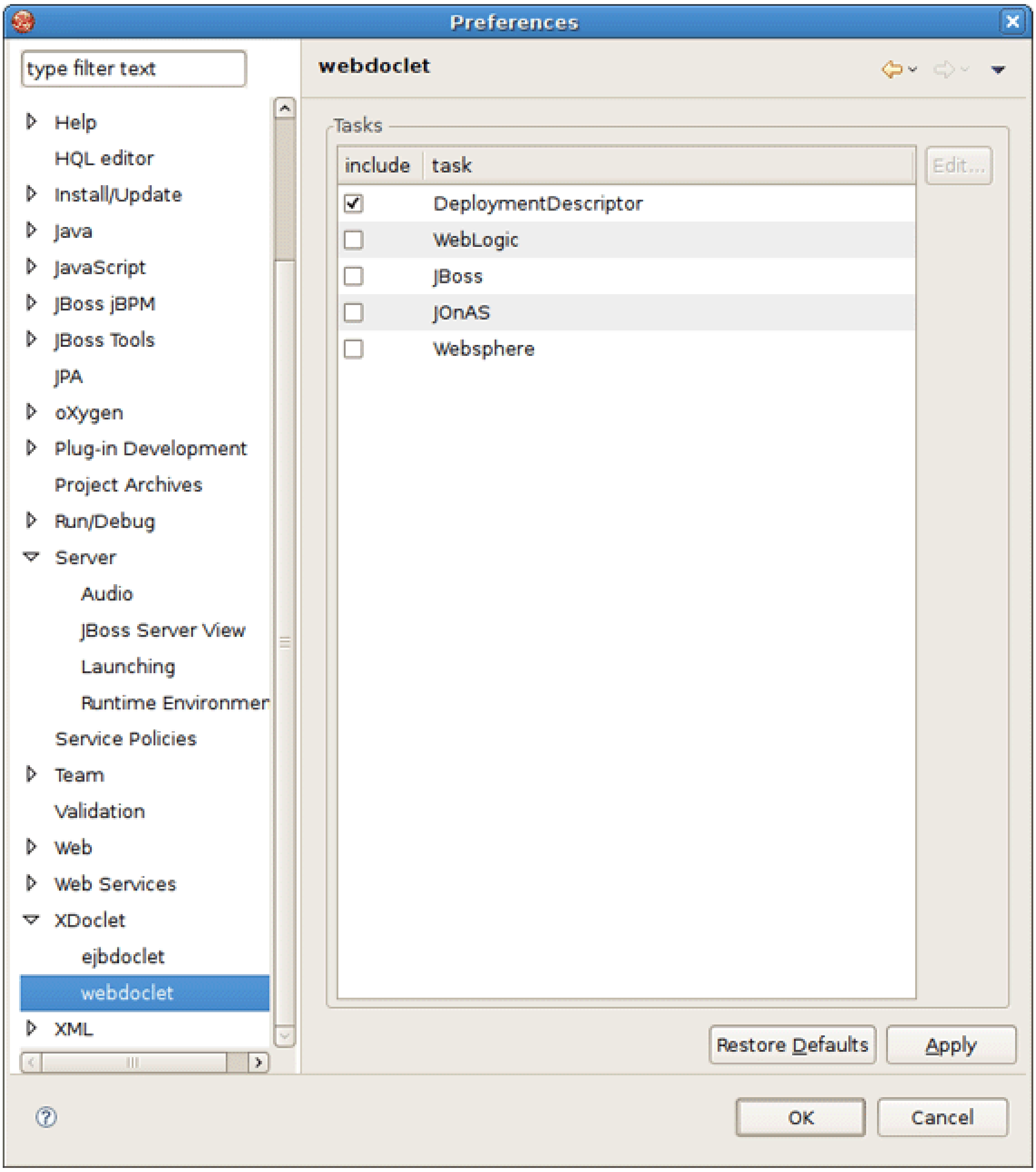
Switch to *XDoclet > ejbdoclet* page in order to adjust settings for EJB-specific sub-tasks.


Preferences
type filter text
Help
HQL editor
Install/Update
Java
JavaScript
JBoss jBPM
JBoss Tools
JPA
oXygen
Plug-in Development
Project Archives
Run/Debug
Server
Audio
JBoss Server View
Launching
Runtime Environmer
Service Policies
Team
Validation
Web
Web Services
XDoclet
ejbdoclet
webdoclet
XML
ejbdoclet
Tasks
Edit...
Restore Defaults
Apply
OK
Cancel

include	task
☑	DeploymentDescriptor
☑	RemoteInterface
☑	HomeInterface
☑	LocalInterface
☑	LocalHomeInterface
☑	UtilObject
☑	DataObject
☑	Dao
☑	valueobject
☑	Entitypk
☑	Entitycmp
☑	Entitybmp
☑	Session
☑	Mdb
☐	Weblogic
☐	JBoss
☐	JOnAS
☐	WebSphere

Figure 8.40. ejbdoclet

To configure settings for various web-specific XDoclet sub-tasks, follow to *XDoclet > webdoclet* page.


Preferences
type filter text
Help
HQL editor
Install/Update
Java
JavaScript
JBoss jBPM
JBoss Tools
JPA
oXygen
Plug-in Development
Project Archives
Run/Debug
Server
Audio
JBoss Server View
Launching
Runtime Environmer
Service Policies
Team
Validation
Web
Web Services
XDoclet
ejbdoclet
webdoclet
XML
webdoclet
Tasks
Edit...
Restore Defaults
Apply
OK
Cancel

include	task
☑	DeploymentDescriptor
☐	WebLogic
☐	JBoss
☐	JOnAS
☐	Websphere

Figure 8.41. webdoclet

Context Menu Preferences and Options

To adjust the project specific preferences, you should bring the context menu for your project and select the *Preferences* option. More details on what adjustments you can perform in the Preferences screen, see in the *"JBoss Tools Preferences"* chapter.

Under the *JBoss Tools* option in the context menu there are also several actions provided by JBDS:

- Add/Remove Struts Capabilities
- Add/Remove JSF Capabilities
- Add Custom Capabilities

9.1. Add/Remove Struts Capabilities

Please, for details refer to the *Struts Tools Reference Guide*.

9.2. Add/Remove JSF Capabilities

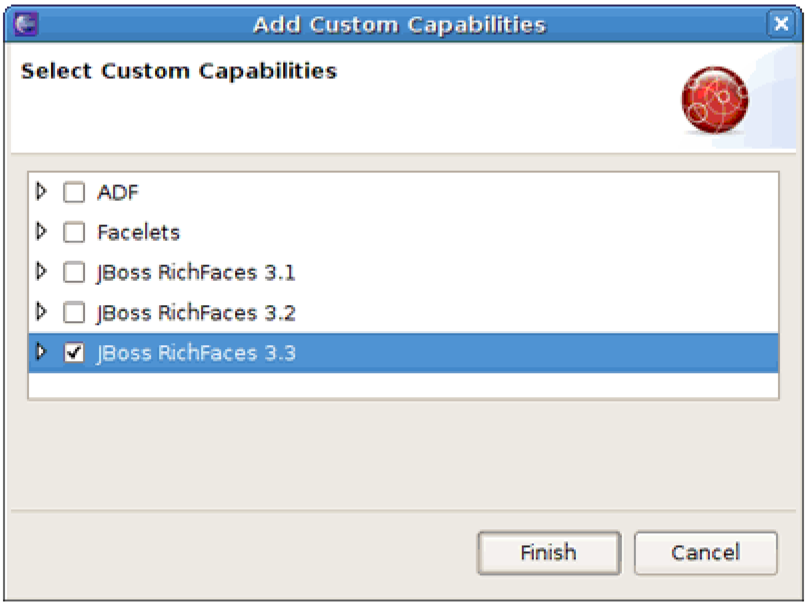
Please, for details refer to the *JSF Tools Reference Guide*.

9.3. Add Custom Capabilities

You can add custom capabilities to any JSF, Struts or Seam project made within JBDS, i.e. add a support of additional frameworks built on top of JSF, such as

- ADF
- Facelets
- JBoss Rich Faces (versions 3.1, 3.2, 3.3)

When the option is selected, the Add Custom Capabilities dialog appears. You should check the needed modules and press *Finish*.

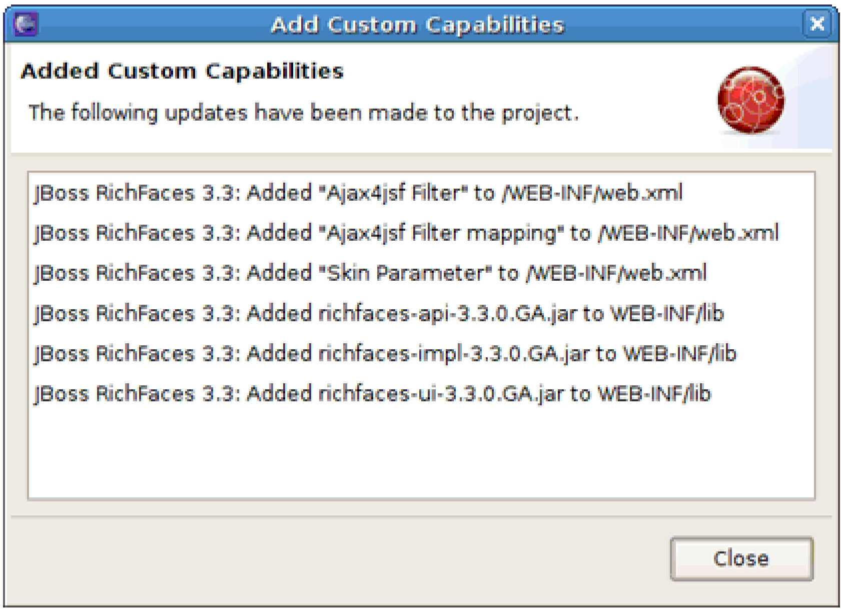


Select Custom Capabilities

- ☐ ADF
- ☐ Facelets
- ☐ JBoss RichFaces 3.1
- ☐ JBoss RichFaces 3.2
- ☑ JBoss RichFaces 3.3

Figure 9.1. Adding Custom Capabilities

The next page displays all the updates that have been made to the project.



Added Custom Capabilities

The following updates have been made to the project.

```
JBoss RichFaces 3.3: Added "Ajax4jsf Filter" to /WEB-INF/web.xml
JBoss RichFaces 3.3: Added "Ajax4jsf Filter mapping" to /WEB-INF/web.xml
JBoss RichFaces 3.3: Added "Skin Parameter" to /WEB-INF/web.xml
JBoss RichFaces 3.3: Added richfaces-api-3.3.0.GA.jar to WEB-INF/lib
JBoss RichFaces 3.3: Added richfaces-impl-3.3.0.GA.jar to WEB-INF/lib
JBoss RichFaces 3.3: Added richfaces-ui-3.3.0.GA.jar to WEB-INF/lib
```

Figure 9.2. Updates Displayed

FAQ

10.1. What should I do if Visual Page Editor does not start under Linux?

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

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

```
yum install compat-libstdc++-33.i386
```

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

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

- On Debian based distributives run the following command:

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

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

If it doesn't help and you use Fedora Core Linux and Eclipse Version: 3.4.1,the issue can be produced because libswt-xulrunner-gtk-3449.so file doesn't present in eclipse-swt-3.4.1-5.fc10.x86_64.rpm/eclipse/plugins/org.eclipse.swt.gtk.linux.x86_64_3.4.1.v3449c.jar.To add this file to eclipse you should:

- Decompress eclipse/plugins/org.eclipse.swt.gtk.linux.x86_3.4.1.v3449c.jar form eclipse-SDK-3.4.1-linux-gtk-x86_64.tar.gz
- Copy libswt-xulrunner-gtk-3449.so file to your Fedora Eclipse location.
- Open the file eclipse.ini,which can be found in your Fedora Eclipse location and add the following line:

```
-Dswt.library.path=/usr/lib/eclipse
```

,where `/usr/lib/eclipse` is the path to your eclipse folder.

10.2. How do I change the auto-formating preferences for the Visual Page Editor?

JBoss HTML/JSP editor uses basic eclipse HTML formatter to format files. So if you want to change preferences of formatter for the Visual Page Editor, you should change it for eclipse html editor (open *Window > Preferences*, then choose *Web > HTML Files > Editor*).

Conclusion

On the whole, this document should guide you to those parts of JBoss Tools which you specifically need to develop Web Applications. It covers 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 *JBoss Developer Studio Users Forum* or *JBoss Tools Users Forum* to ask questions. There we are also looking for your suggestions and comments.