Struts Tools Reference Guide

Version: 3.1.1.GA

Introduction

JBoss Developer Studio provides a number of editors and wizards that allow Struts developers to quickly and efficiently build high quality web applications.

Note:

Note that JBoss Tools support the Struts 1.1, 1.2.x versions.

In this guide you will learn how to take advantage of Struts support that is provided by JBoss Developer Studio.

1.1. Key Features of Struts Tools

The table below lists the main features provided by Struts Tools:

Table 1.1. Key Functionality for Struts Tools

Feature	Benefit	Chapter
Struts Support	Step-by-step wizards for creating a new Struts project with a number of predefined templates, importing existing ones and adding struts capabilities to non-Struts web projects.	*Chapter 2, Projects*
Support for Struts Configuration File	Working on files using three modes: diagram, tree and source. Synchronization between the modes and full control over the code. Easily move around the diagram using the Diagram Navigator. Working with struts projects that have multiple modules. The option to use thee Struts configuration file debugger, which allows break points to be set on a Struts diagram and then launch the server in debug mode.	*Section 3.2, "Graphical Editor for Struts Configuration Files" Chapter 6, Struts Configuration File Debugger*
Support for Struts modules	A Struts module (`struts-config.xml`) is automatically created when creating a new project. There is also the option to add new or edit already existing modules in your existing project or while importing Struts project.	*Chapter 4, Modules*
Verification and Validation	All errors will be immediately reported by verification feature, no matter which view you are working in. Constant validation and errors checking allows developers to catch many of the errors during development process, which significantly reduces development time.	*Chapter 8, Struts Project Verification*

1.2. Other relevant resources on the topic

All JBoss Developer Studio/JBoss Tools documentation you can find on*JBoss Tools project page* [http://docs.jboss.org/tools/].

The latest documentation builds are available on*nightly docs page* [http://download.jboss.org/jbosstools/nightly-docs/].

Projects

JBoss Tools provides the following functionality when working with Struts:

- Create new Struts projects
- Import (open) existing Struts projects. You can import any project structure
- Add Struts capabilities to any existing Eclipse project
- Import and add Struts capabilities to any existing project created outside Eclipse.

Now, we'll focus on all these points more fully.

2.1. Creating a New Struts Project

JBoss Tools provides a **New Struts Project** Wizard that significantly simplifies the process of creating a new Struts project. You just need to follow these steps:

- Select **File** → **New** → **Project...** from the menu bar. Then, select **JBoss Tools Web** → **Struts** → **Struts Project** from the dialog box. Click the **Next** button:

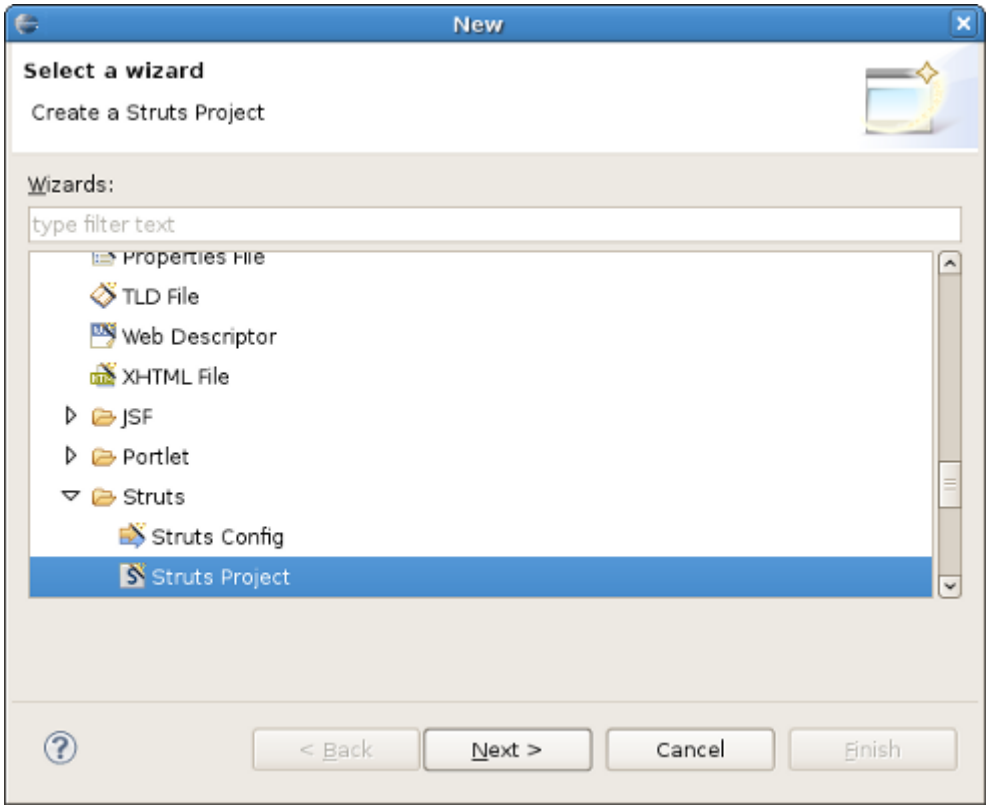

Figure 2.1. Selecting Struts Wizard

- On this form, provide the Project Name. You can also select where to create the project, or use the default path.

  The Struts Environment sets the Struts version that will be used.

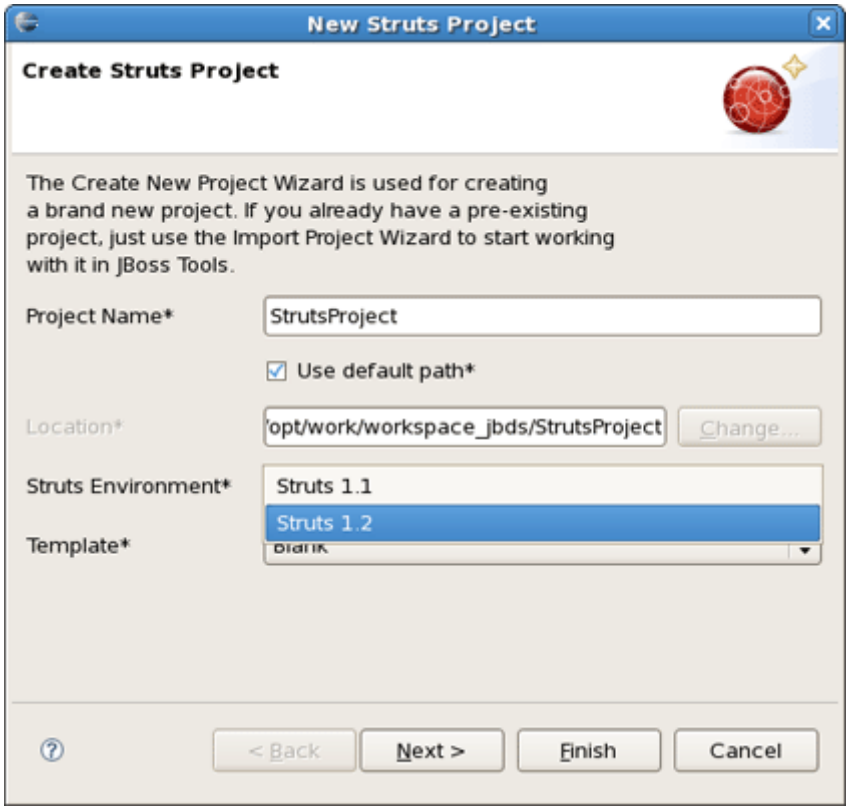

Figure 2.2. Creating Struts Project

Tip:

Don't put spaces in project names since some operating systems may experience problems processing and searching for these files.

You can select the KickStart template, which results in a project that includes a simple Hello World type application that is ready to run.

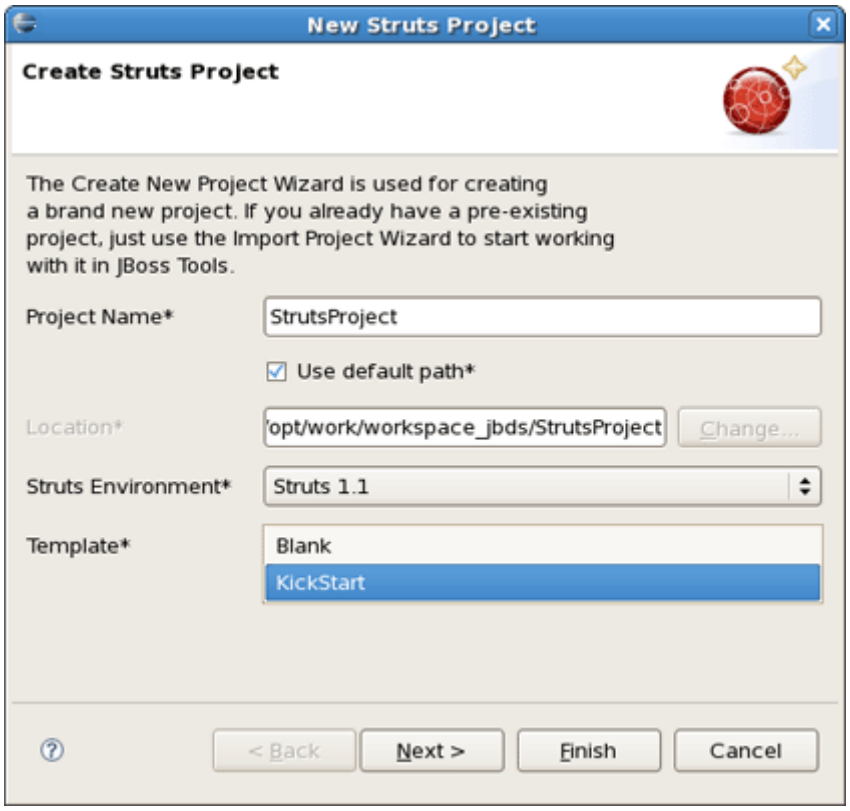

Figure 2.3. Choosing Struts Template

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

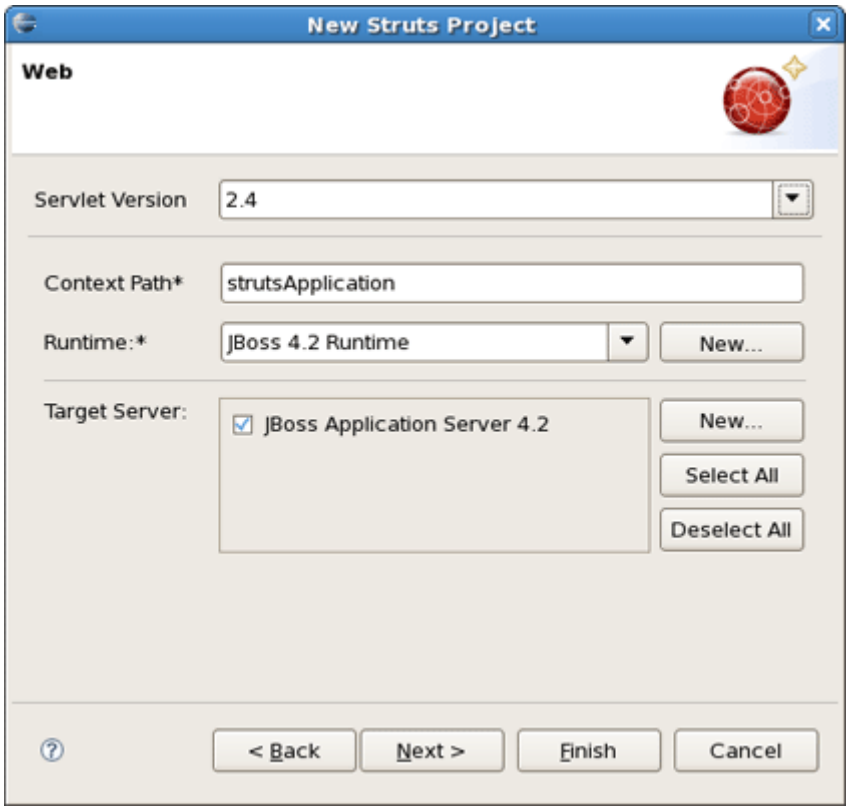

Figure 2.4. Registering The Project at Server

- On the next form, you can select the TLD files to include in this project:

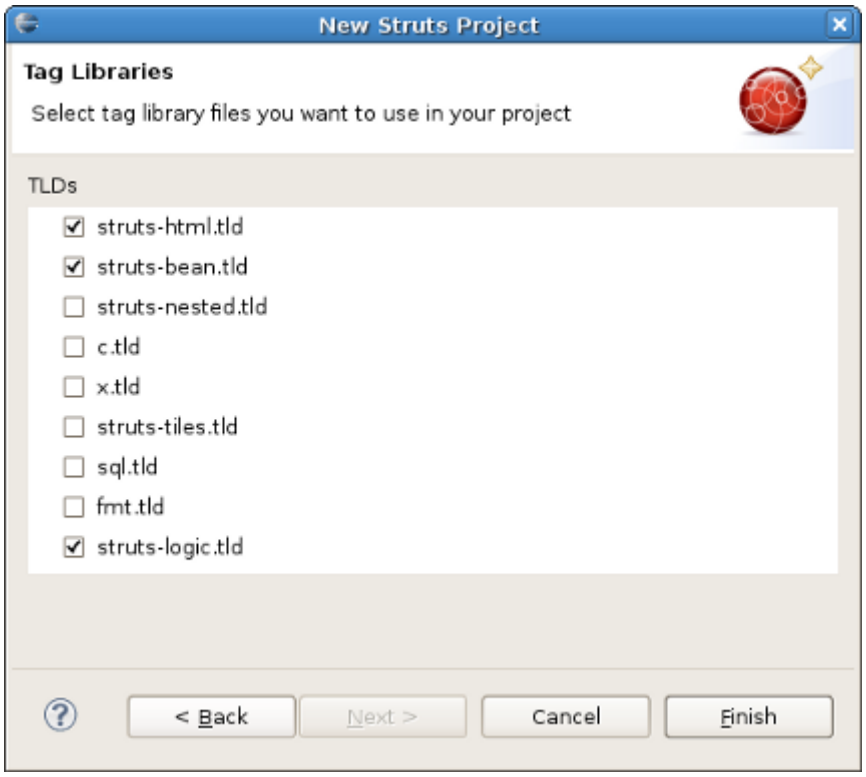

Figure 2.5. Selecting Tag Libraries

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

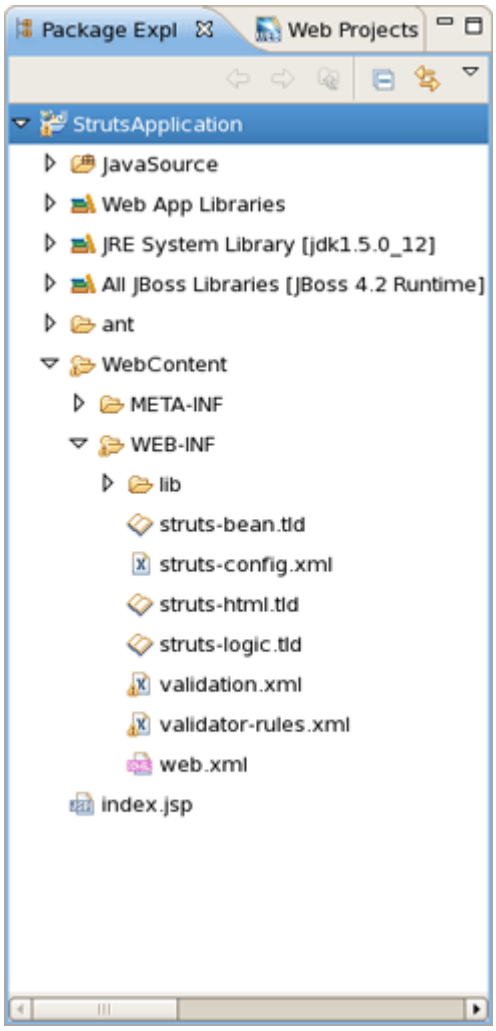

Figure 2.6. Project Structure

Tip:

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

2.2. Importing an Existing Struts Project with Any Structure

For detailed information on migration projects to JBoss Developer Studio see the Migration Guide.

2.3. Adding Struts Capability to an Existing Web Application

This section will describe how to add Struts functionality (Struts libraries, tag libraries and a Struts configuration file) to any existing Web application project in your Eclipse workspace.

By adding a Struts Nature to your project, you can now edit files using JBoss Tools editors, such as the *Struts configuration editor* and the JBoss Tools JSP editor. To take advantage of this just right-click the project and select **JBoss Tools** → **Add Struts Capabilities** from the context menu. This will start the process of adding all the necessary Struts libraries and files to your existing project.

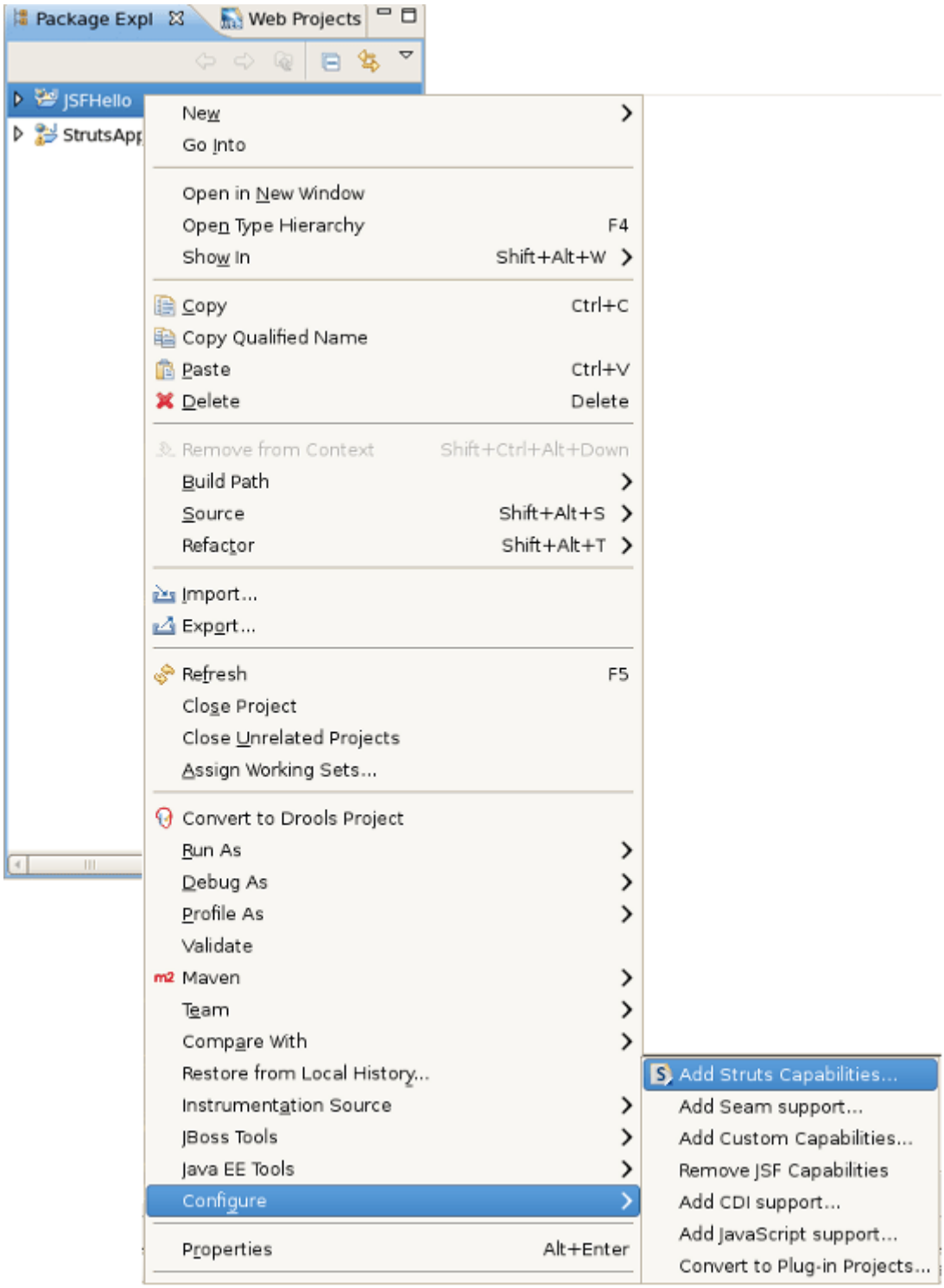

Figure 2.7. Adding Struts Capabilities

In the wizard you should point to the location of your deployment descriptor file `web.xml` and name of the project.

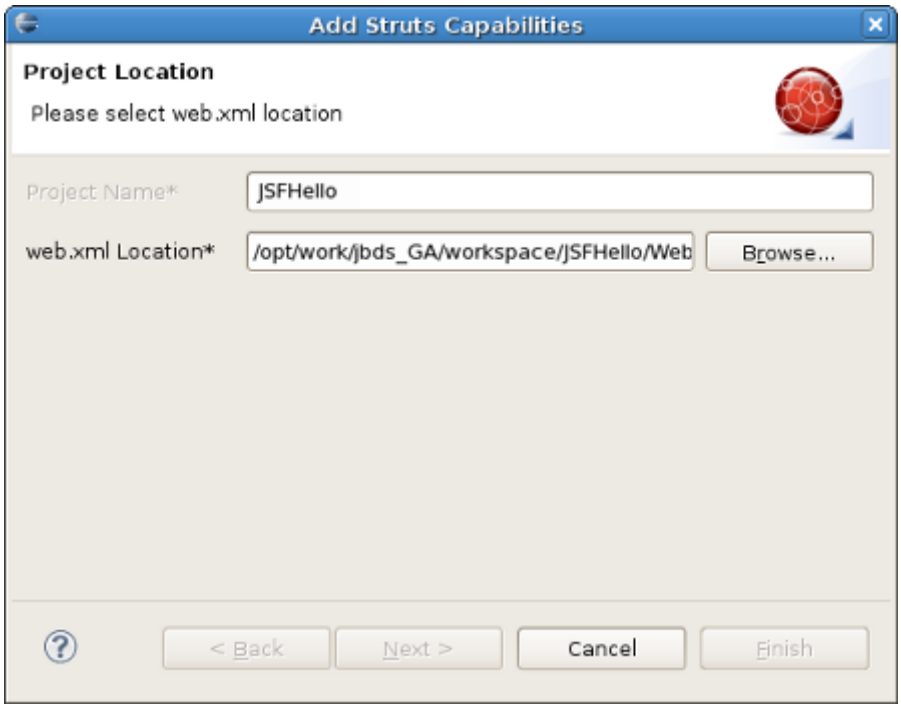

Figure 2.8. Choosing Project Location

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

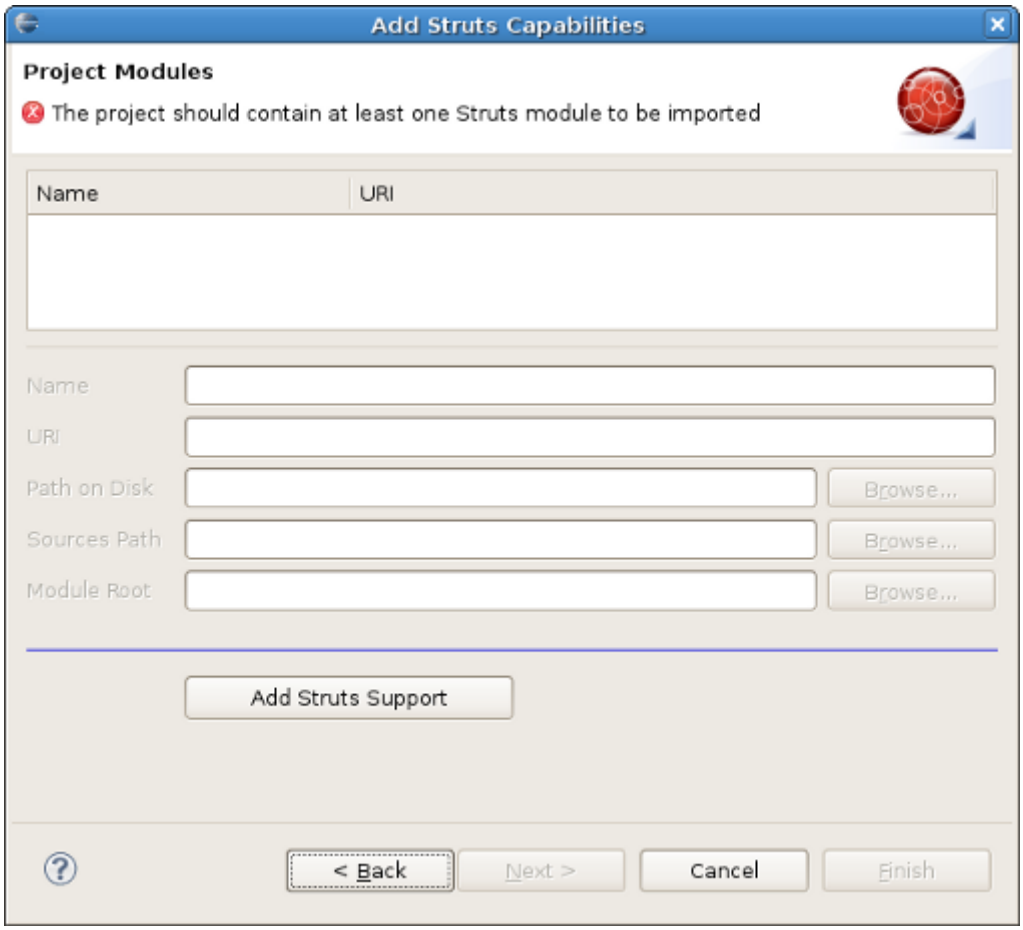

Figure 2.9. Project Modules

Here you can select which Struts Version, Servlet Class, URL Pattern and TLDs to add to this project.

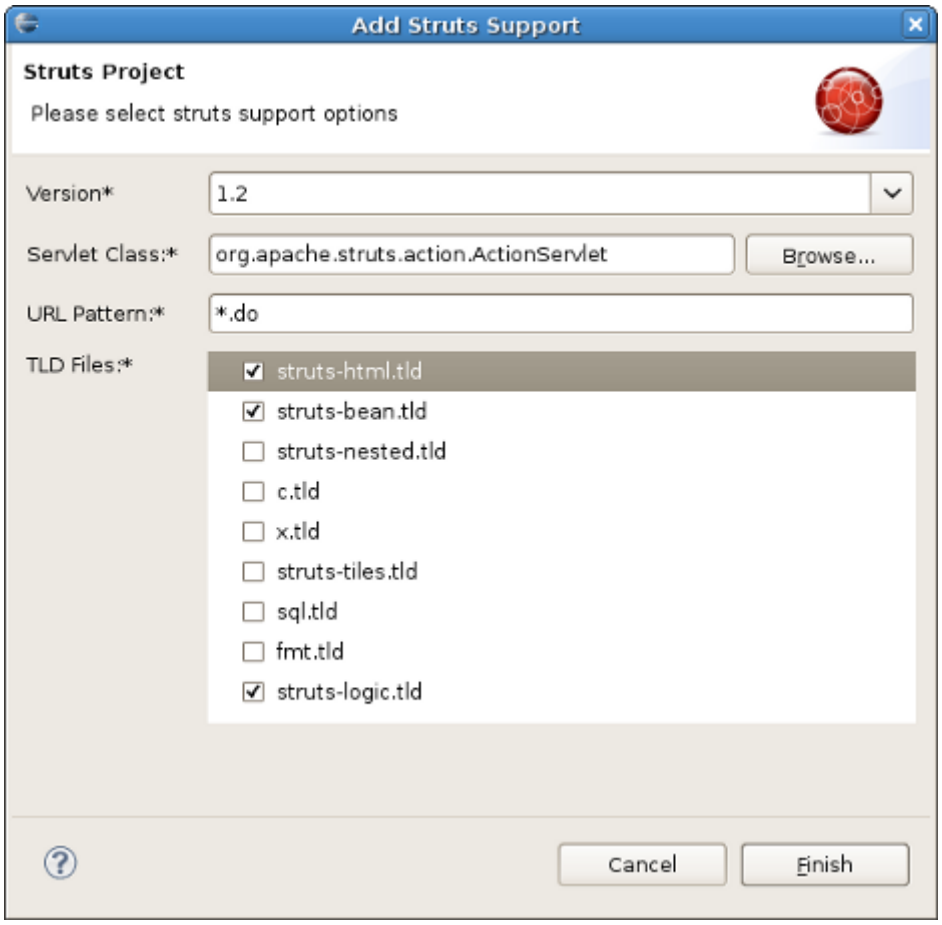

Figure 2.10. Selecting Struts Support Options

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

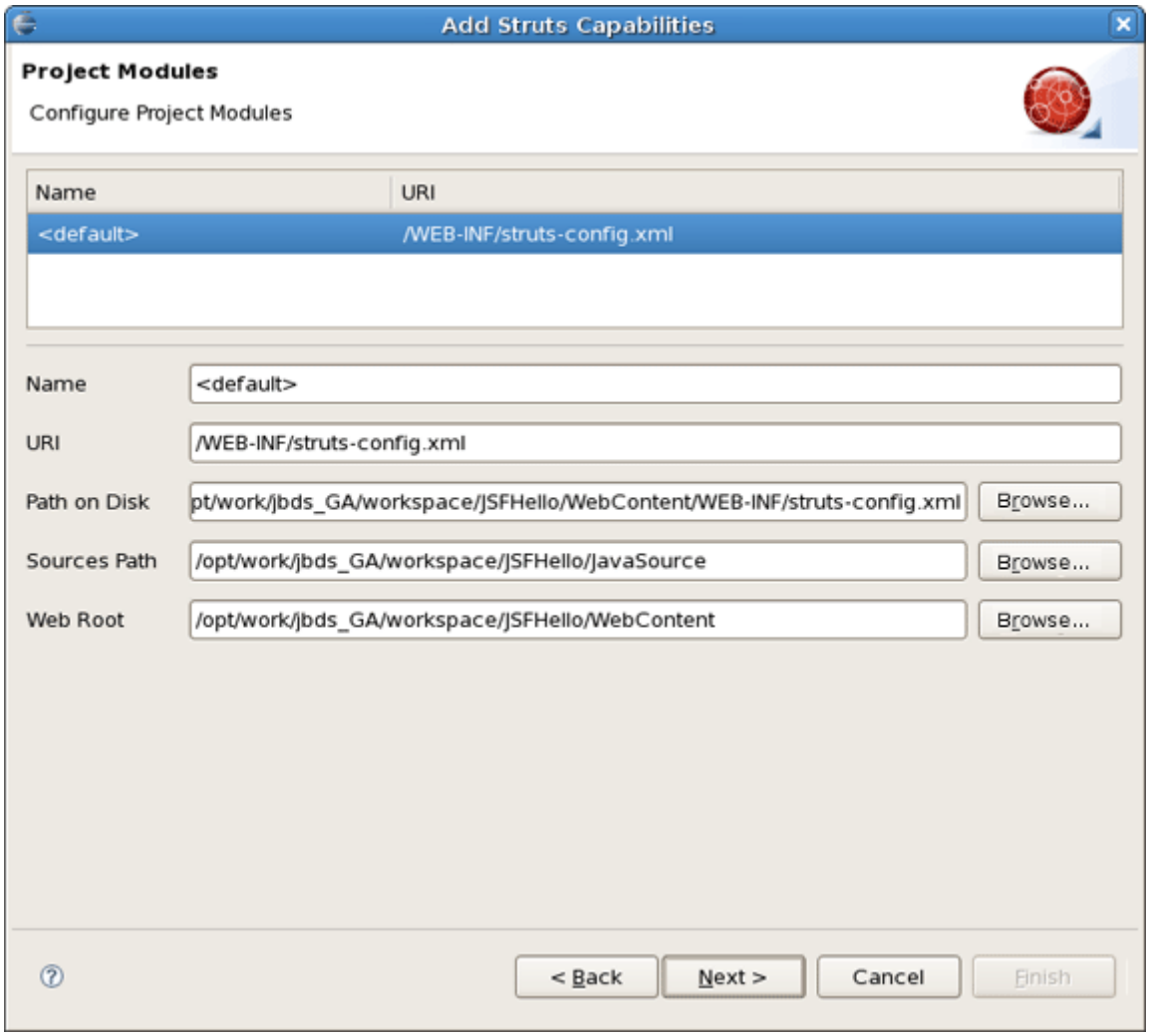

Figure 2.11. Project Configuration Information

On the last screen you can set the different folders for your project as well as register this application with a servlet container. If you want the libraries (.jar files) to be automatically added to your project, tick the **Add Libraries** checkbox.

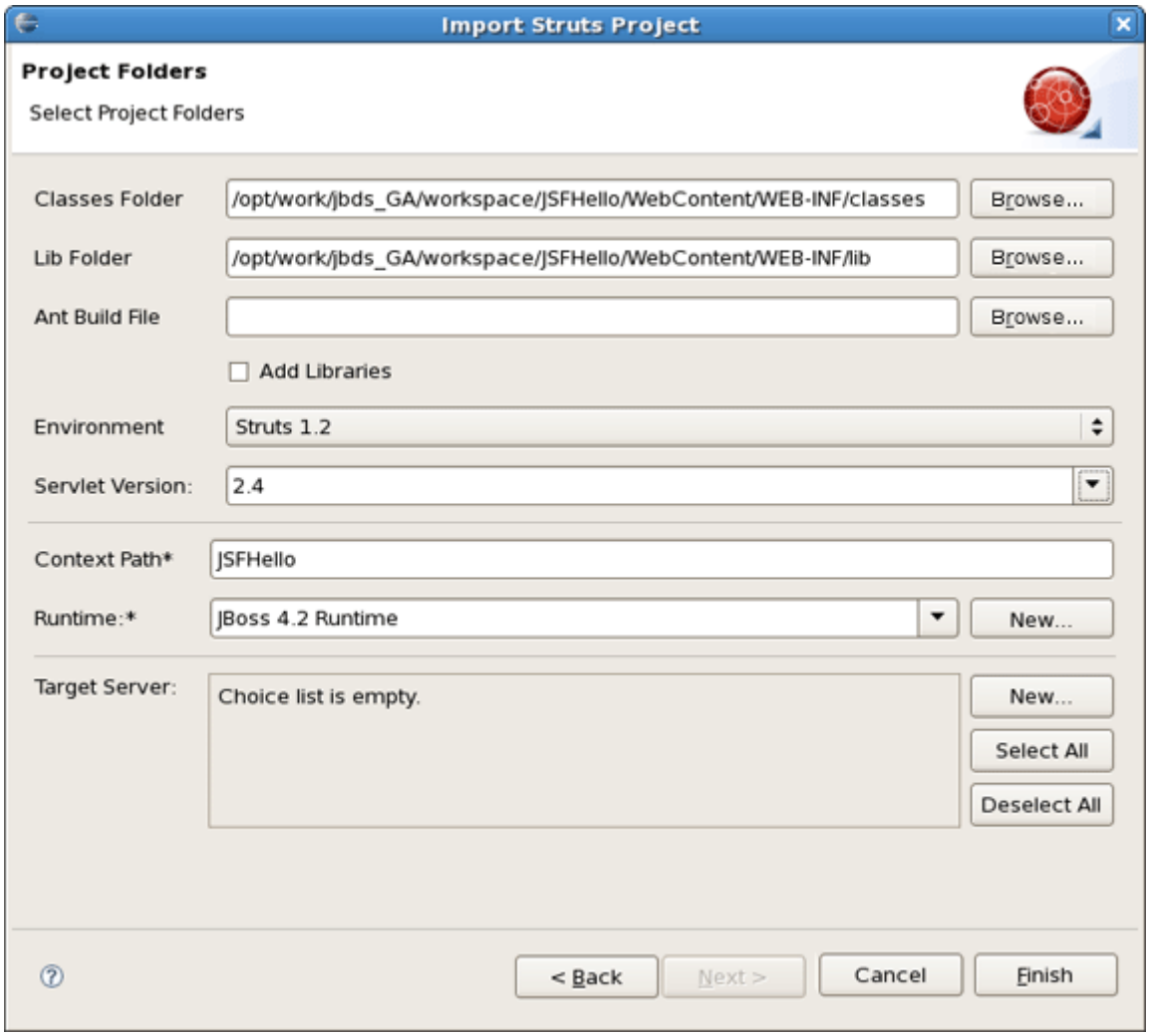

Figure 2.12. Registering the Project at Server

When done, you can open and edit the `struts-config.xml` file using useful Struts configuration file editor provided by JBDS. (The Struts configuration is shown below in the Tree viewer).

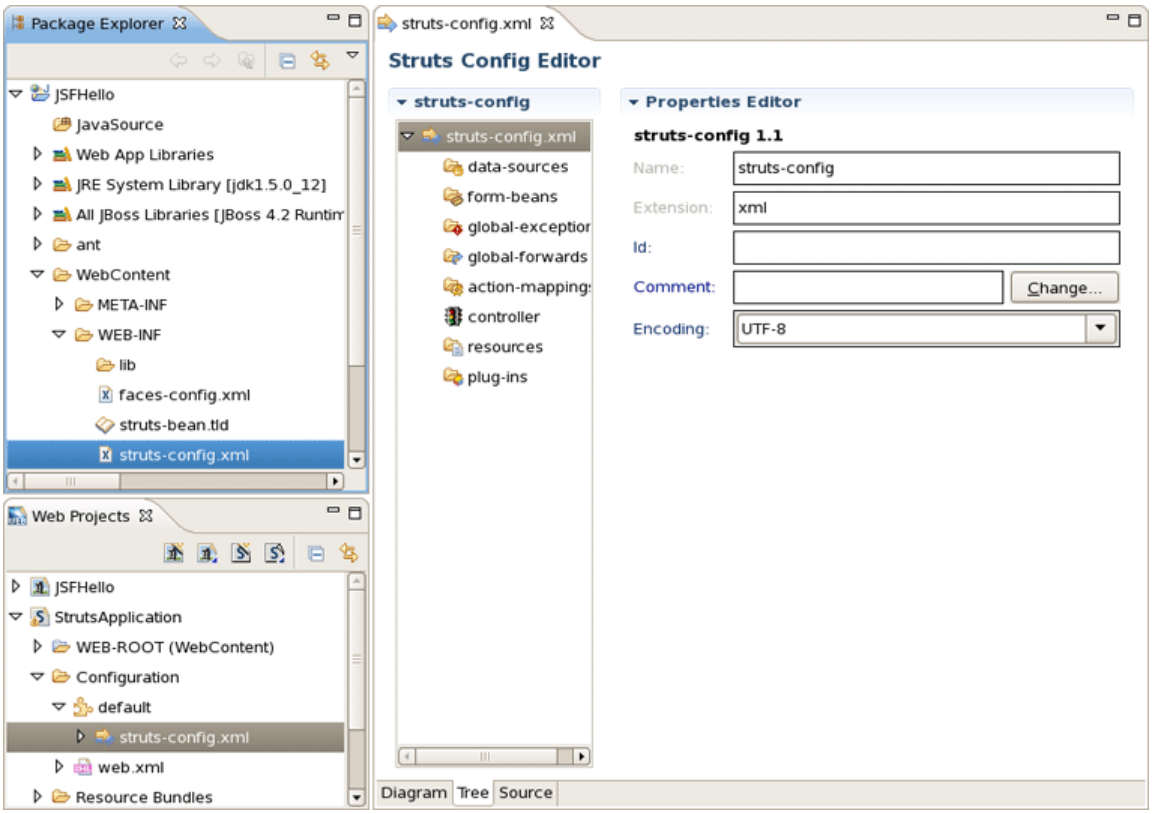

Figure 2.13. Struts-config.xml File

2.4. Relevant Resources Links

You can find more in-depth explanation on how to work with special wizards, editors and views that can be used in various scenarios while developing Struts applications in our Visual Web Tools Guide.

Editors

This chapter will cover the graphical editors provided for editing specific Struts files such as Struts Configuration files, Tiles files, Struts Validation files and web.xml.

3.1. Web.xml Editor

The `web.xml` file inside the `WEB-INF` folder is a deployment descriptor file for a Web Application. It describes the servlets, other components and deployment properties that make up your application.

JBoss Tools adds the `web.xml` file to a Struts project automatically and provides a special editor through which the file can be edited. See the Visual Web Tools guide that gives a descriptive information on the `web.xml` editor.

3.2. Graphical Editor for Struts Configuration Files

First, let's dwell on the Struts Configuration file editor.

This editor provides three views of the `struts-config.xml` file: Diagram, Tree and Source. The views can be selected using the tabs at the bottom of the editor. Any changes made in one view are immediately visible when you switch to any other view.

Now, we'll consider each of these three views in more detail.

3.2.1. Diagram View

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

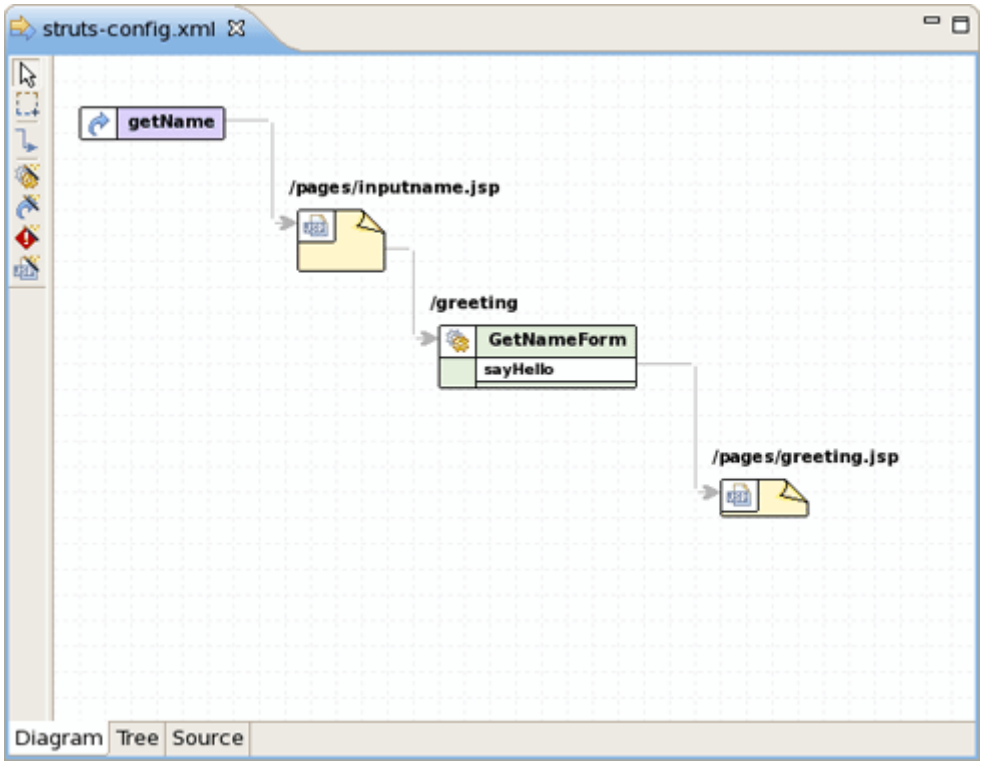

Figure 3.1. Diagram View

The Diagram view provides a way to edit the navigation in your Struts application. By right-clicking anywhere on the diagram, you can use a context menu to create the building blocks of a Struts application:

- Actions
- Global forwards
- Global exceptions
- JSP Pages

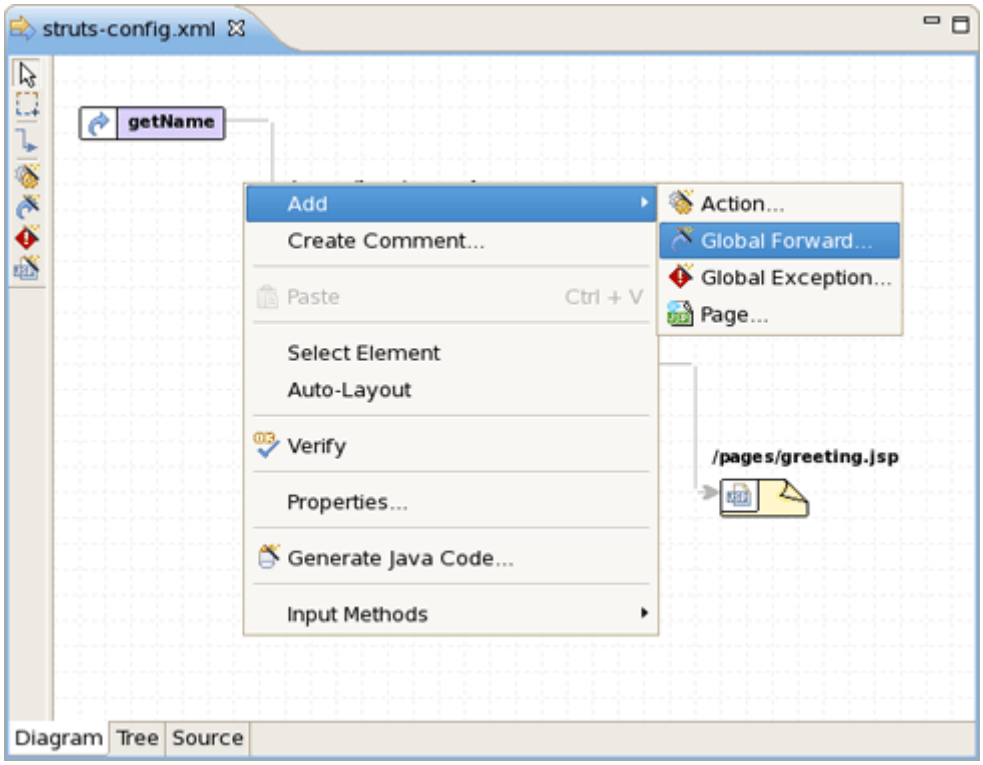

Figure 3.2. Diagram Context Menu

Along the upper-left side of the editor is a stack of seven icons for changing the behavior of the cursor in the diagram.

Figure 3.3. Editor Icons

The first icon switches to the default regular selection cursor, the second to the marquee selection cursor and the third to the new connection cursor. The last four icons switch the cursor to an insert cursor for each type of Struts build block listed above (and in the order listed).

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

3.2.2. Tree View

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

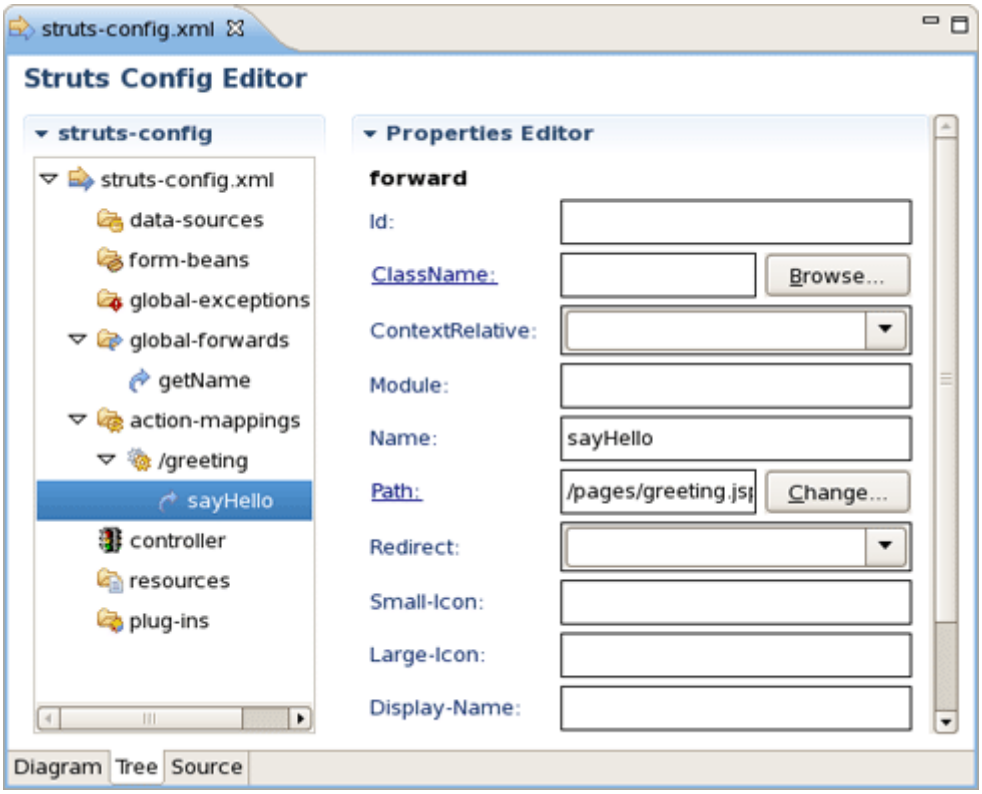

Figure 3.4. Tree View

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

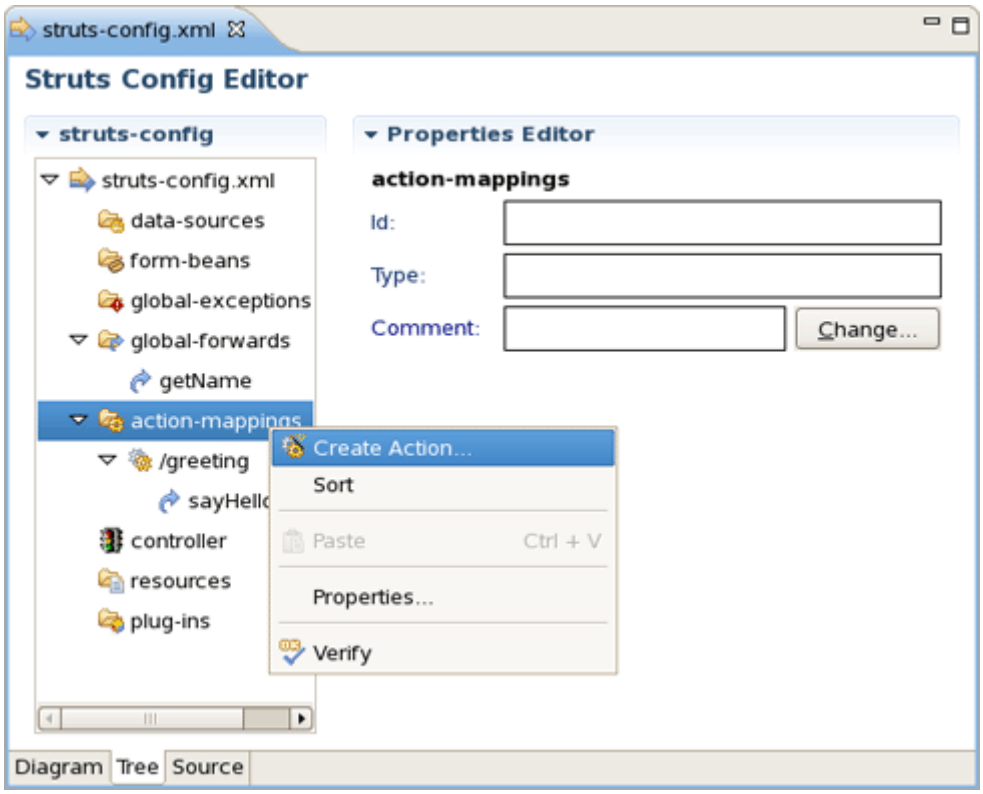

Figure 3.5. Tree Context Menu

Let's consider the tree on the left more closely.

- Under the *data-sources* node you can create a Data Source object that will be configured and made available as a servlet context attribute.
- The *form-beans* node is meant for creating a set of form bean descriptors for this module. Every element under this node is a JavaBean that implements the `org.apache.struts.action.ActionForm` class. Use the Properties editor on the right to adjust the properties specific to each form-bean.
- The *global-exceptions* node is intended for registering handlers for the exceptions that might be thrown by an Action object.
- Use the *global-forwards* node to add, edit or delete a global forwards that represent ActionForward objects available to all Action objects as a return value.
- The *controller* node allows you to configure the controller properties.
- Under the *resources* node you can add, delete, or edit message resources.
- Under the *plug-ins* node you can define a Struts plug-in. Right-click the node, select **Create Plug-in** and specify the plug-in *Id* and *ClassName* by pointing to the Java class which implements the `org.apache.struts.action.PlugIn` interface.

3.2.3. Source View

In the Source view, you have complete editing control of the underlying XML coding.

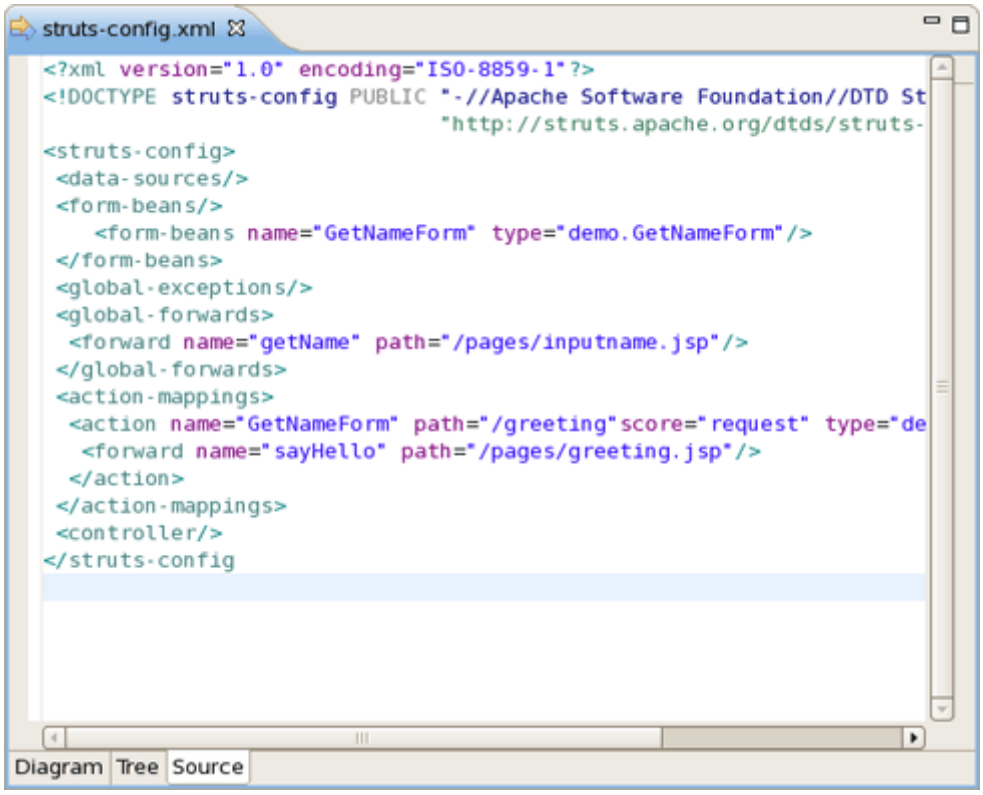

Figure 3.6. Source View

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

- Content Assist
- Open On Selection
- File Folding

You can take advantage of code assist.

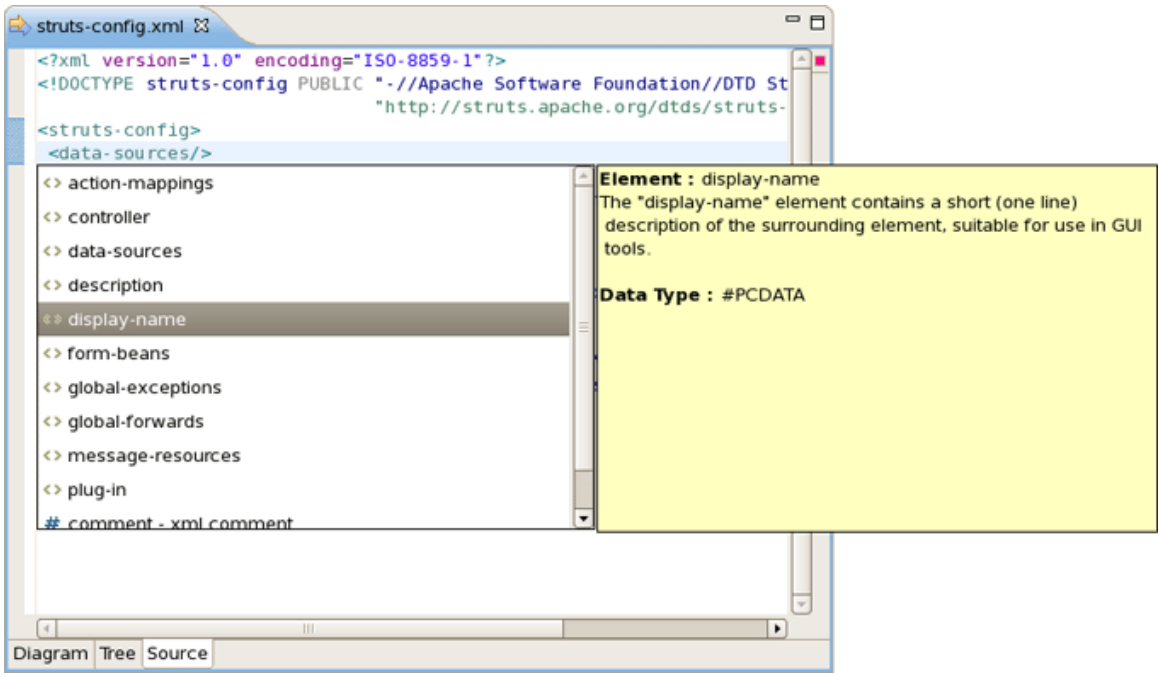

Figure 3.7. Code Assist

The editor will also immediately flag any errors.

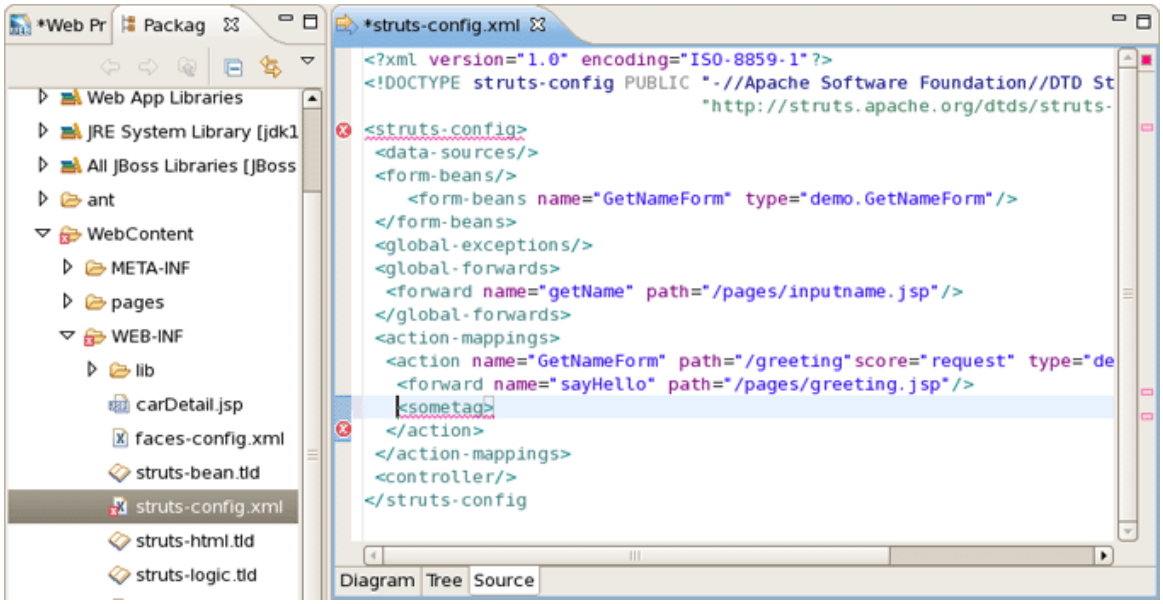

Figure 3.8. Errors in Source View

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

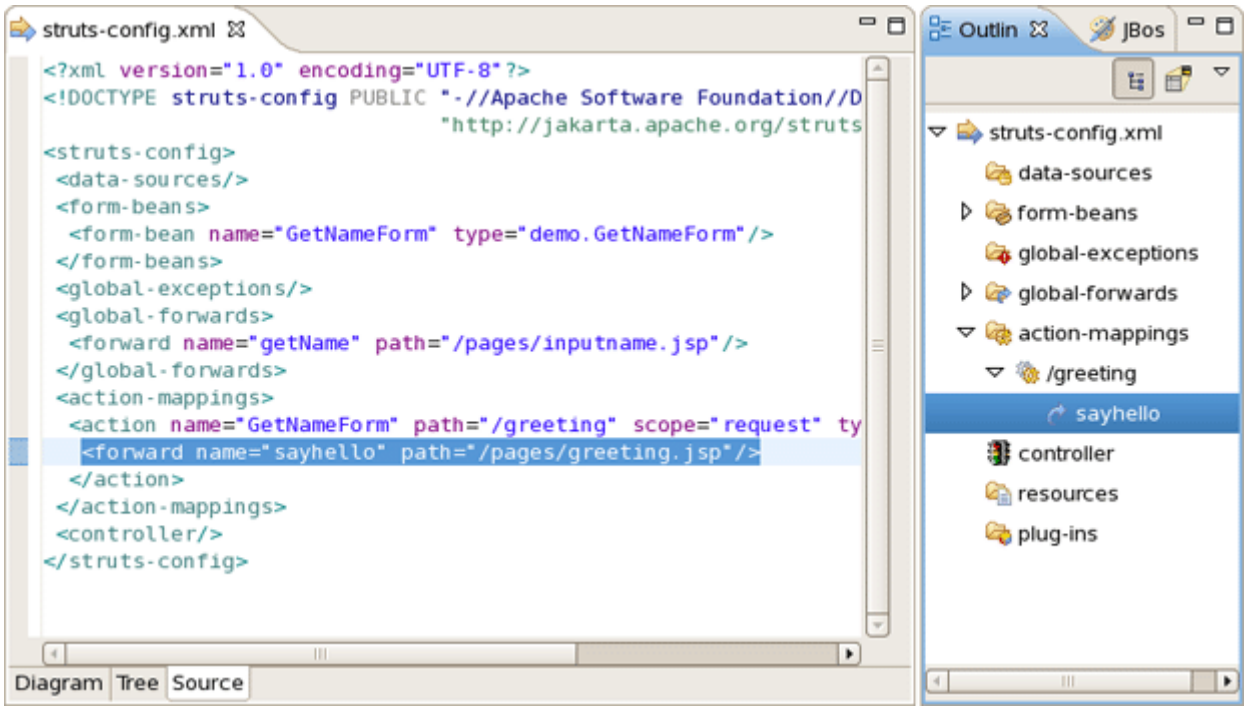

Figure 3.9. Outline View

Find more information about editor features in the editor features chapter.

3.3. Graphical Editor for Tiles Files

Here, you'll know how to make use of the special graphical editor for Tiles configuration files.

The editor has three main views: Tree, Diagram and Source. The views can be selected via the tabs at the bottom of the editor. Any changes made in one view are immediately visible when you switch to any other view.

Before we consider each view of the editor, let's look at the way of creating new Tiles files.

3.3.1. Create New Tiles File

To create new Tiles files, right click any folder and select **New** → **Tiles File**.

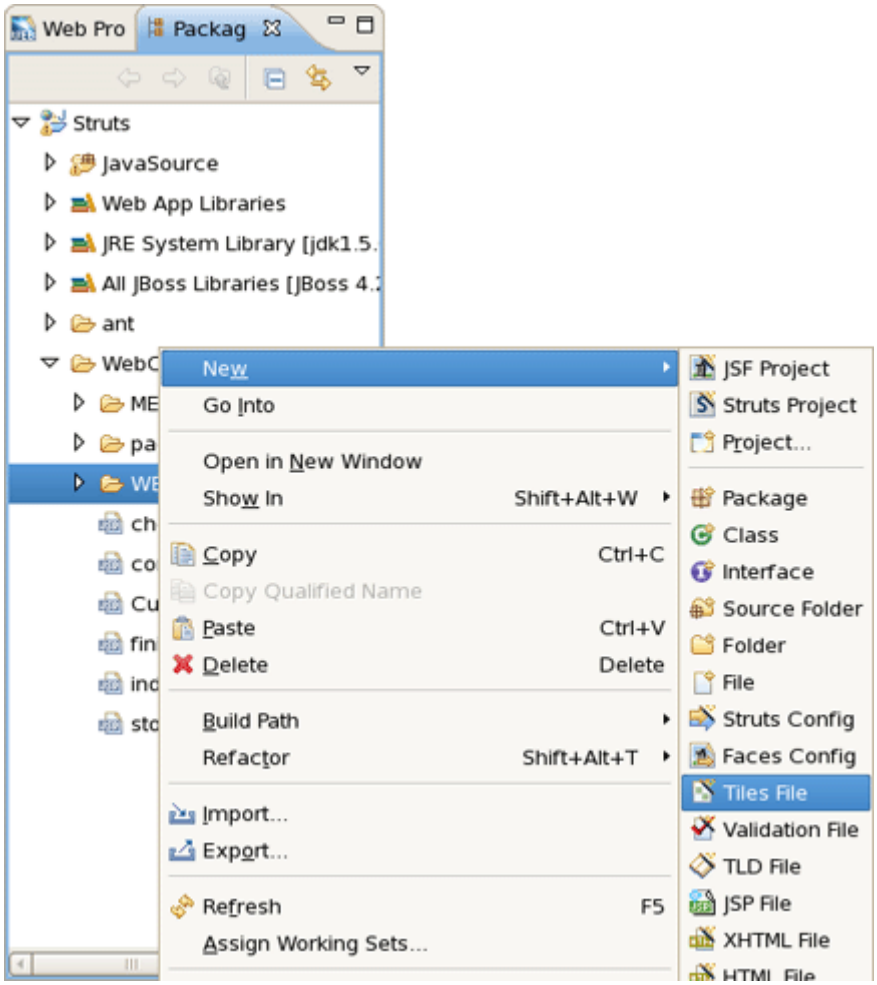

Figure 3.10. Creating a New Tiles File

3.3.2. Tree View

The Tree view represents the different elements of the Tiles file organized into functional categories on the left-hand side and a form for editing the properties of currently selected items on the right-hand side.

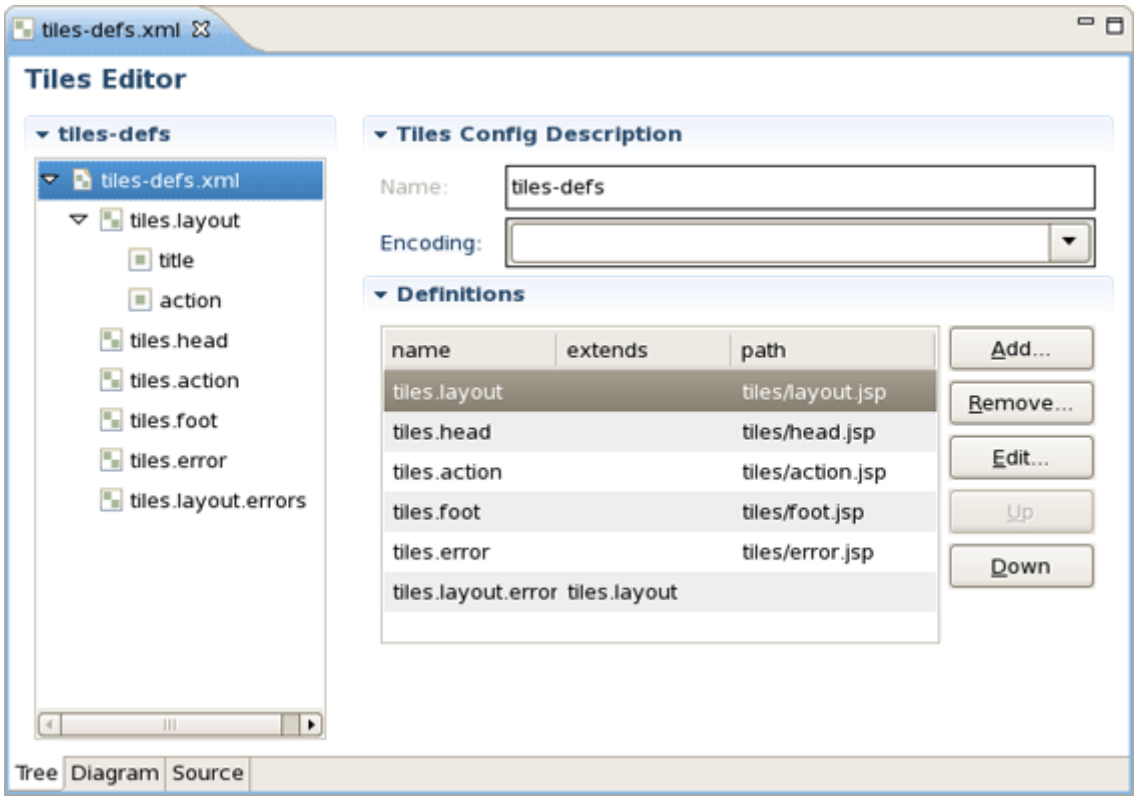

Figure 3.11. Tree View

To edit the file, simply right click any node and select from the available actions.

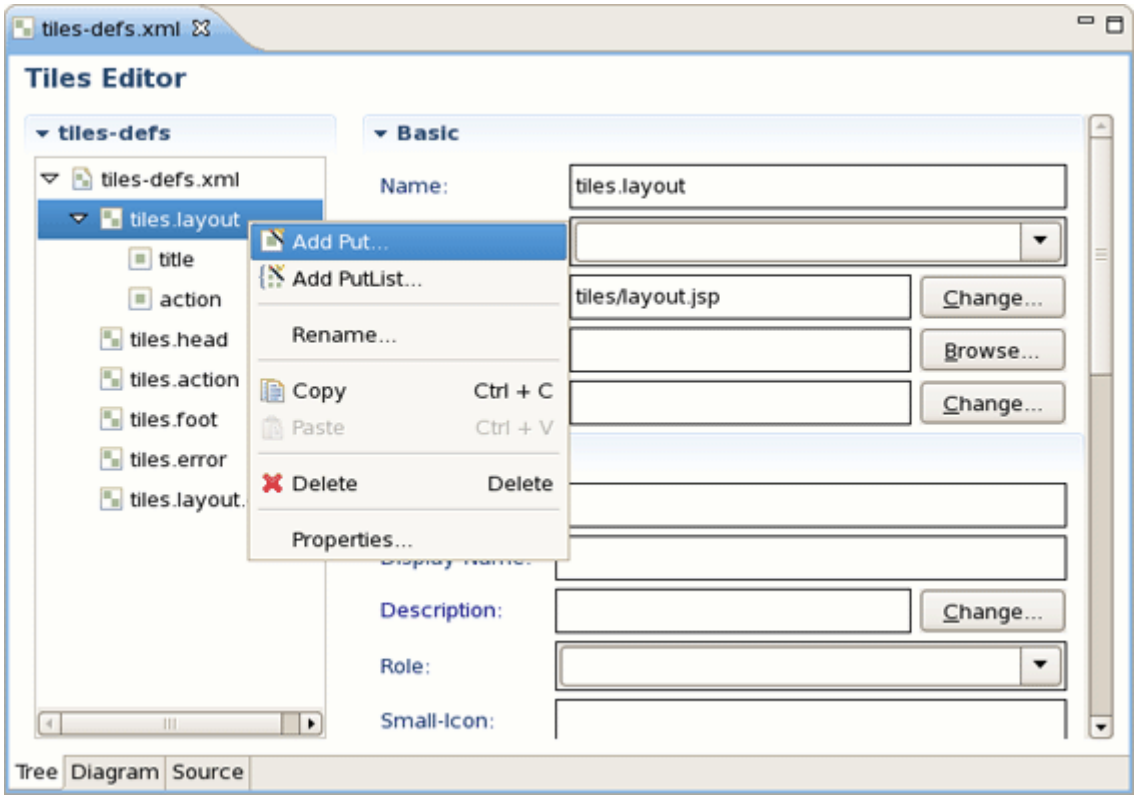

Figure 3.12. Editing in Tiles Editor

3.3.3. Diagram View

The Diagram view allows you to create complex Tiles files in the form of a diagram.

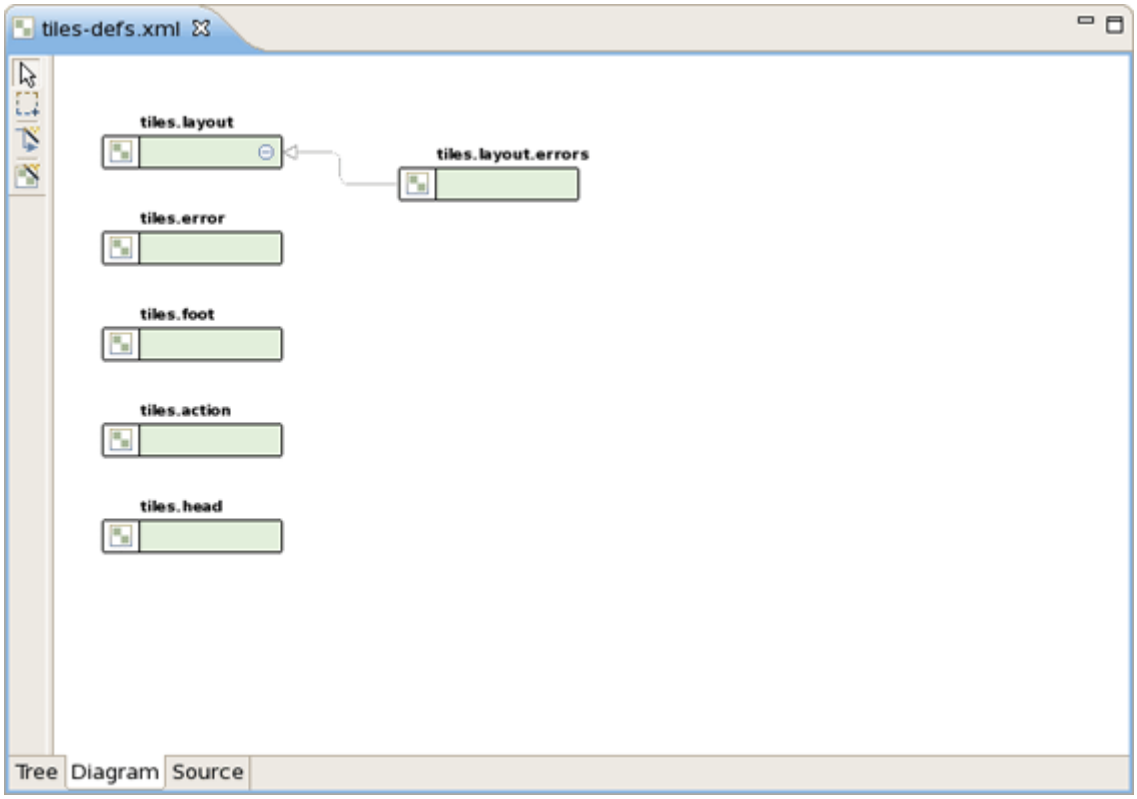

Figure 3.13. Diagram Mode

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

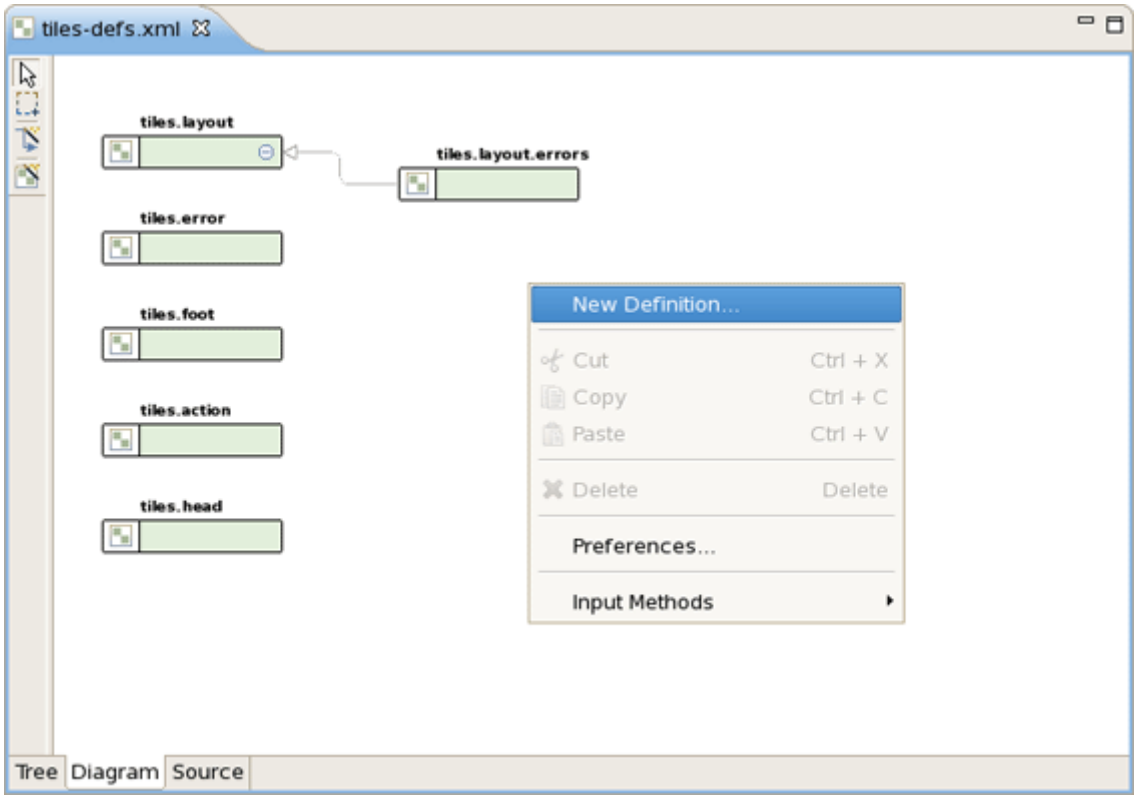

Figure 3.14. Creating New Definition

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

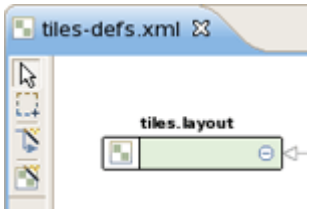

Figure 3.15. Diagram Toolbar

It contains four icons for changing the cursor state. The first one is the default cursor state for selecting existing nodes. The second icon is marquee selector. The third is used for creating new connections and the last one is for adding definition template to the content.

3.3.4. Source

The other view of the Tiles editor is the Source view that gives you full control over the source. Any changes here will immediately appear in other modes when you switch to them.

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

- Content Assist

- Open On Selection

Figure 3.16. Source View

Code assist is available in the Source mode.

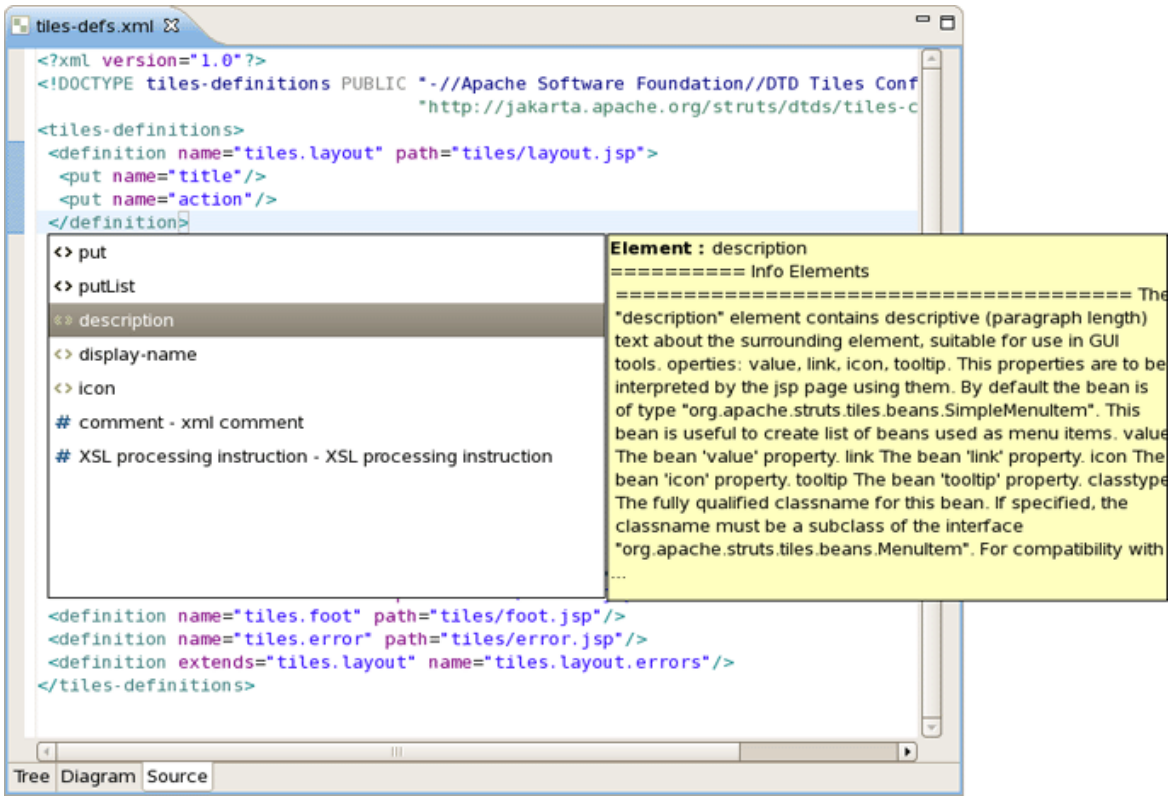

Figure 3.17. Code Assist

Any errors are immediately reported as shown below:

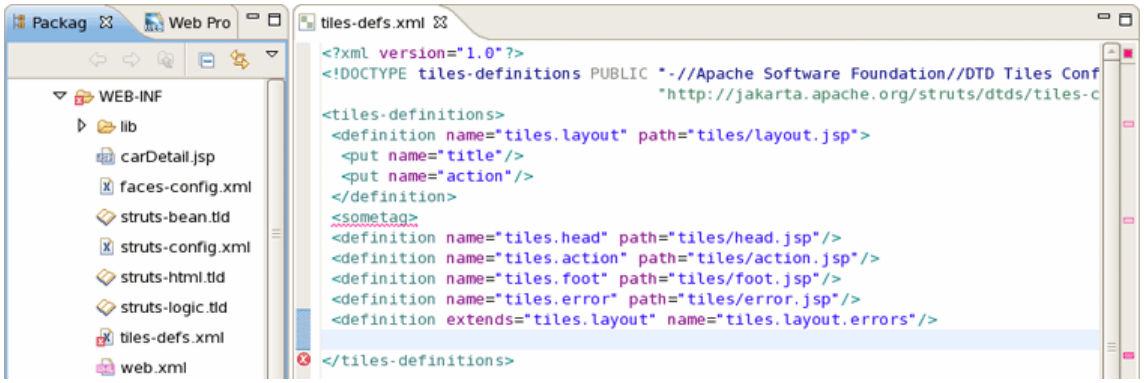

Figure 3.18. Errors Reporting

You can also use the Outline view together with the editor's Source mode. It provides a way to more easily navigate through the file.

Figure 3.19. Outline View

3.4. Graphical Editor for Struts Validation Files

JBoss Tools comes with a visual validation editor that provides full support for the development of Struts applications. To open the editor double-click on the validation file, or create a new one if it doesn't exist.

To create a new validation file, right click any folder in Project Explorer and select **File** → **New** → **Other...** from the context menu and then select **JBoss Tools Web** → **Struts** → **Validation File**.

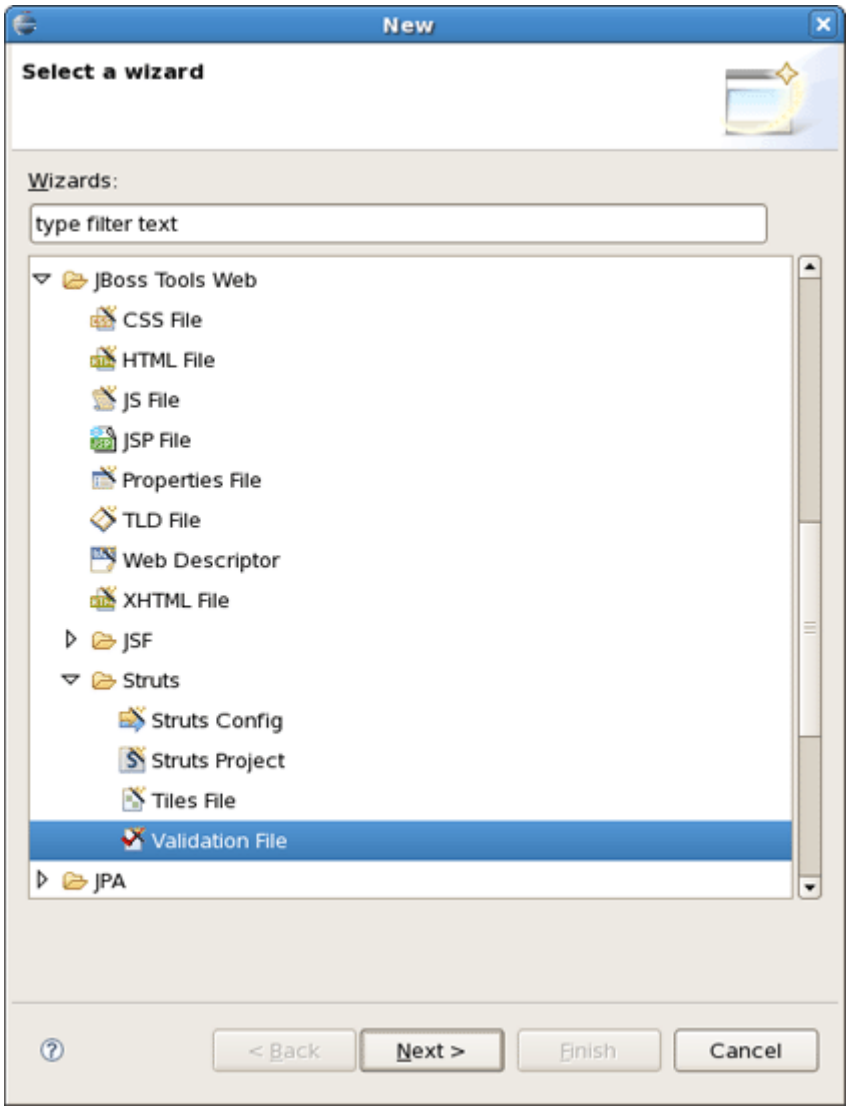

Figure 3.20. Creating New Validation File

The validation editor works with five modes: Formsets, Validators, Constants and standard Tree and Source that you can easily switch between using tabs at the bottom of the editor.

The Formsets view shows forms and their elements on the left side and the dialogue for defining their validation rules on the right side.

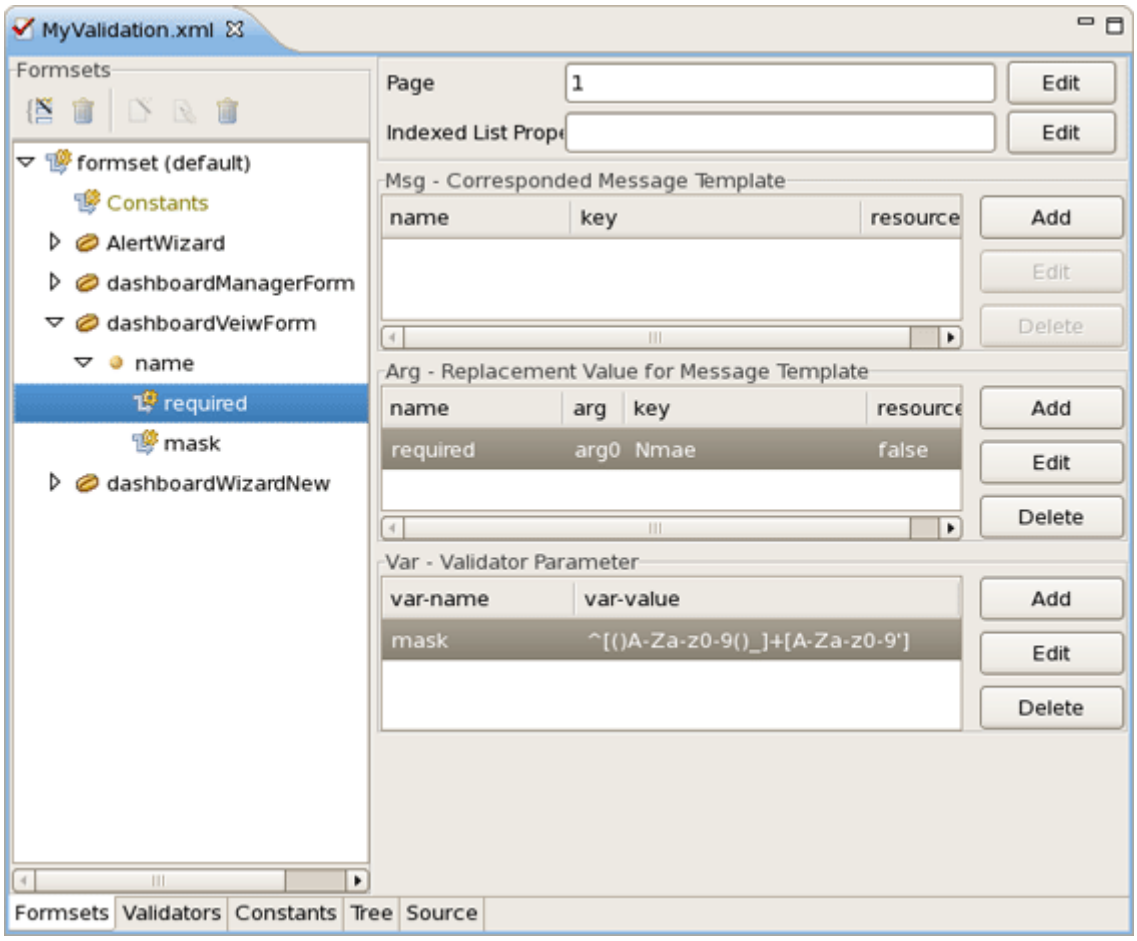

Figure 3.21. Formsets View

The Constants view let you set constant values for your validation rules.

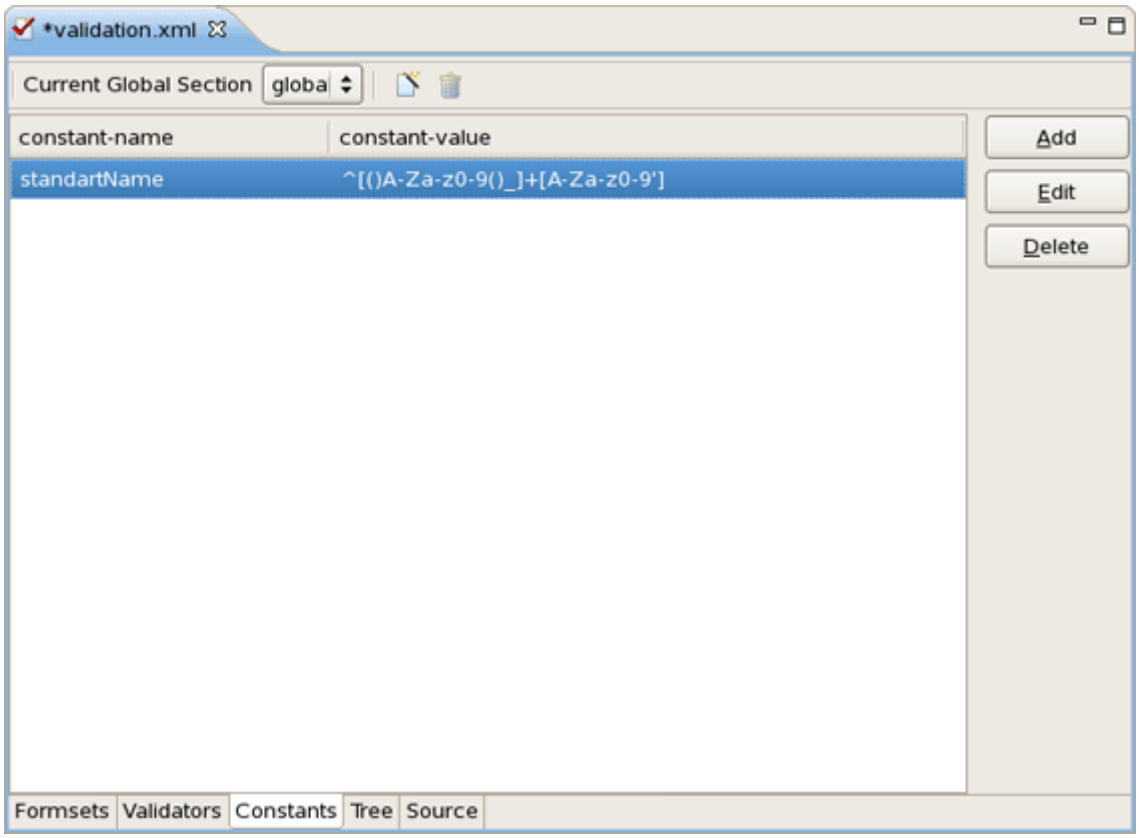

Figure 3.22. Constansts View

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

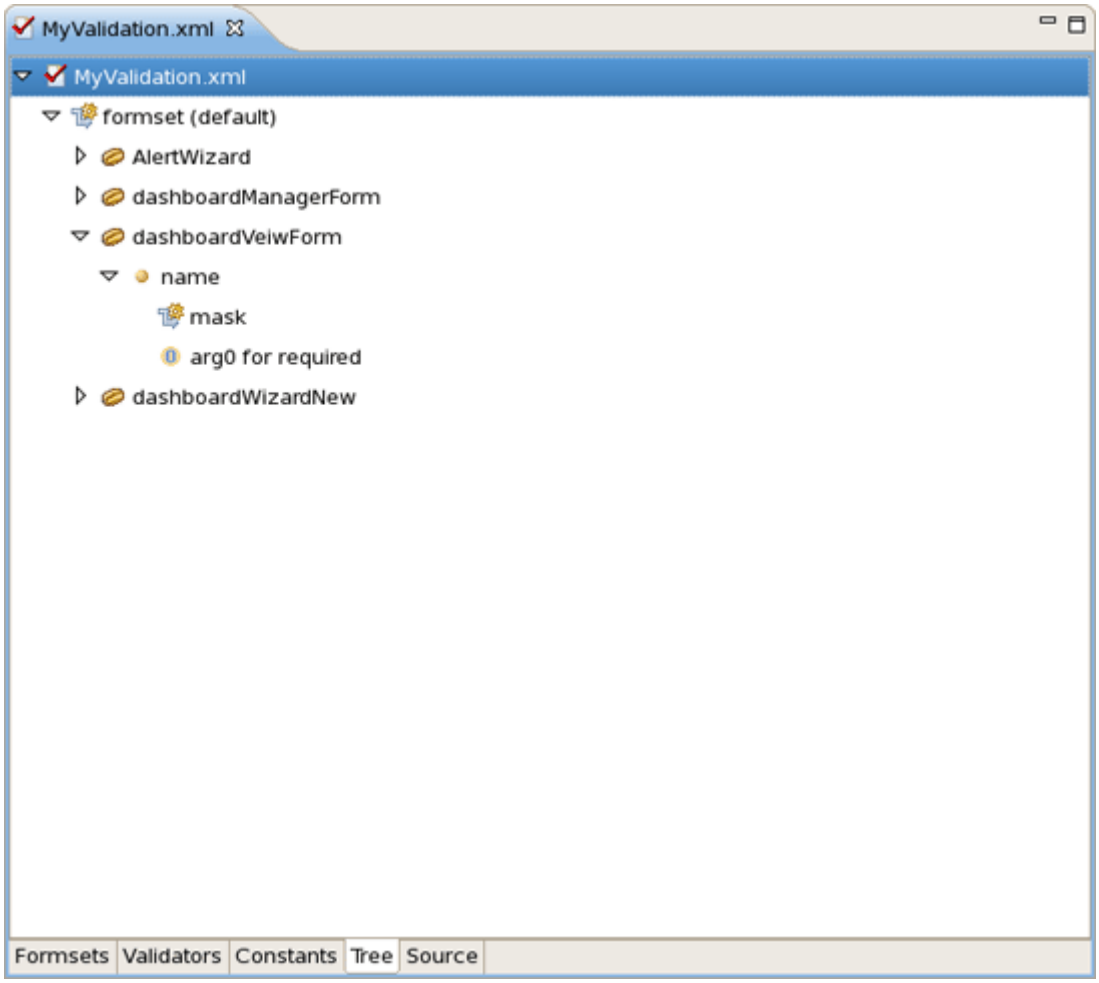

Figure 3.23. Tree View

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

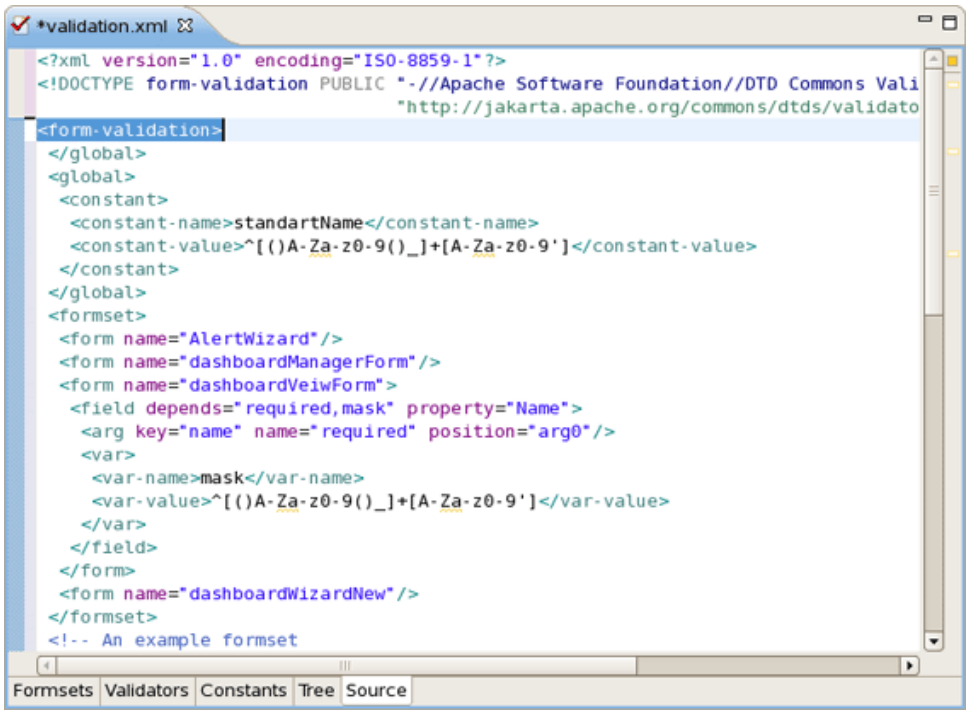

Figure 3.24. Source View

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

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

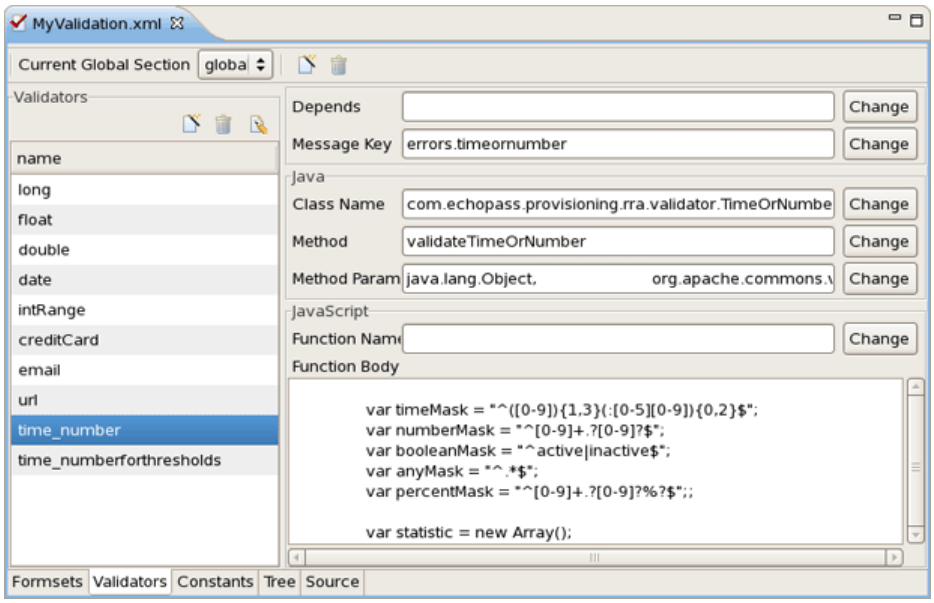

Figure 3.25. Validators View

Here are the validation rules shown in the Source mode.

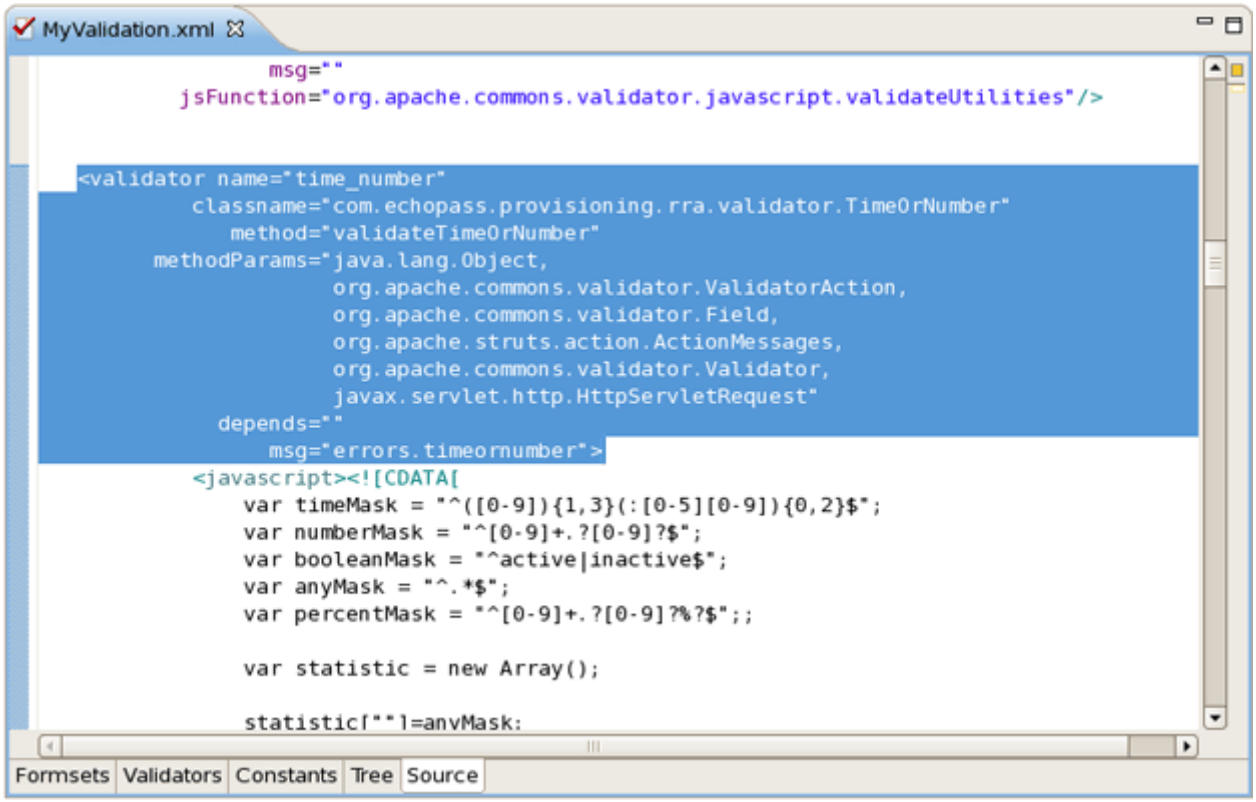

Figure 3.26. Validation Rules

Modules

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

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

Now, let's discuss this functionality in more detail.

4.1. When Importing a Struts Project

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

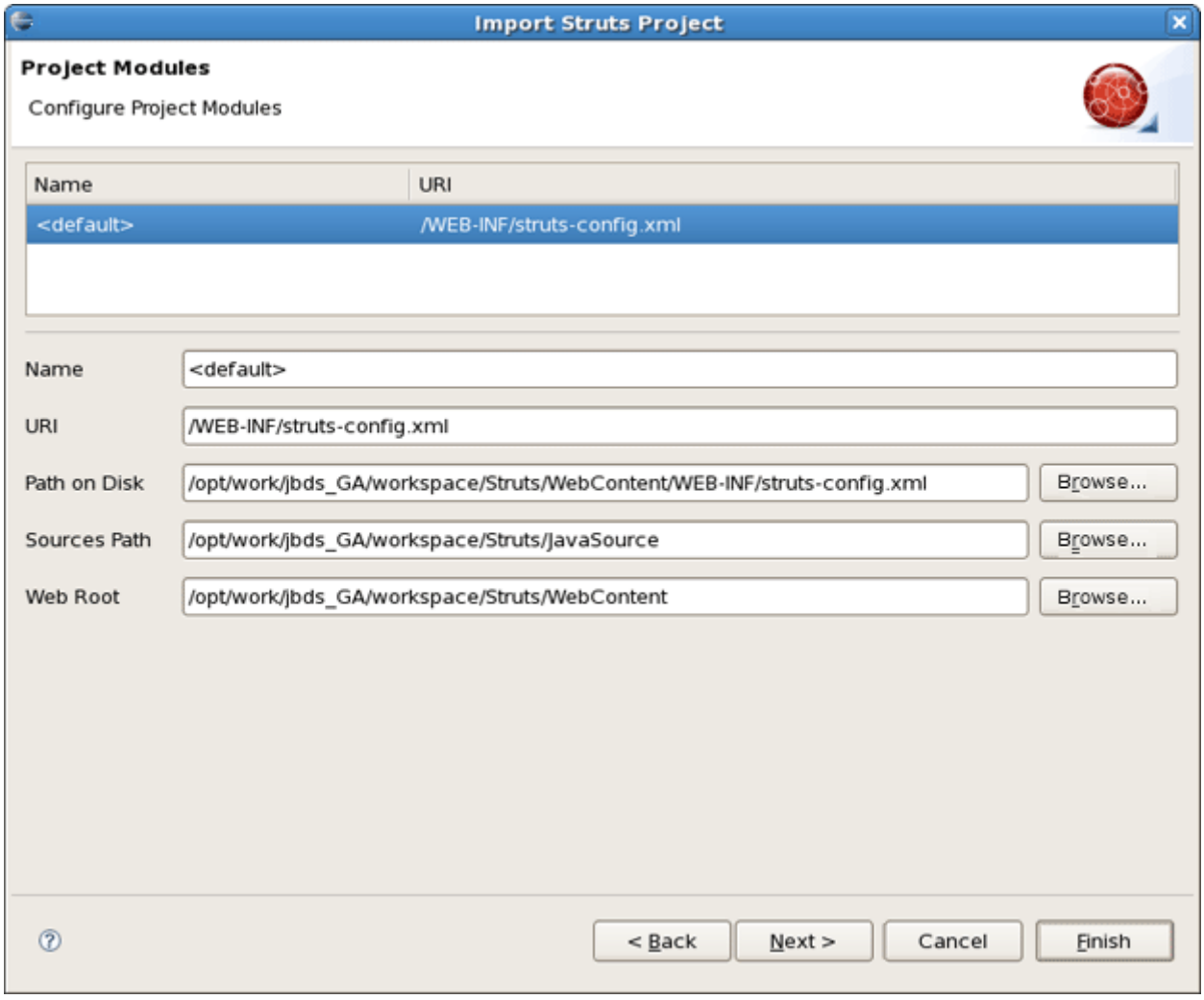

Figure 4.1. Configuring Project Modules

4.2. Editing Modules in an Existing Project

To edit modules in an existing project, right click the project in Web Project view and select **JBoss Tools Struts** → **Modules Configuration** (or right click the project in Package Explorer view and select **Configure** → **Modules Configuration**).

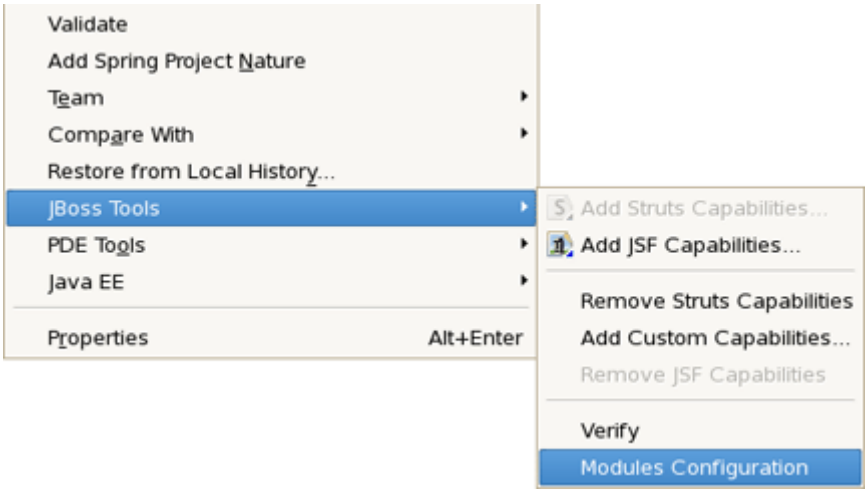

Figure 4.2. Choosing Modules Configuration

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

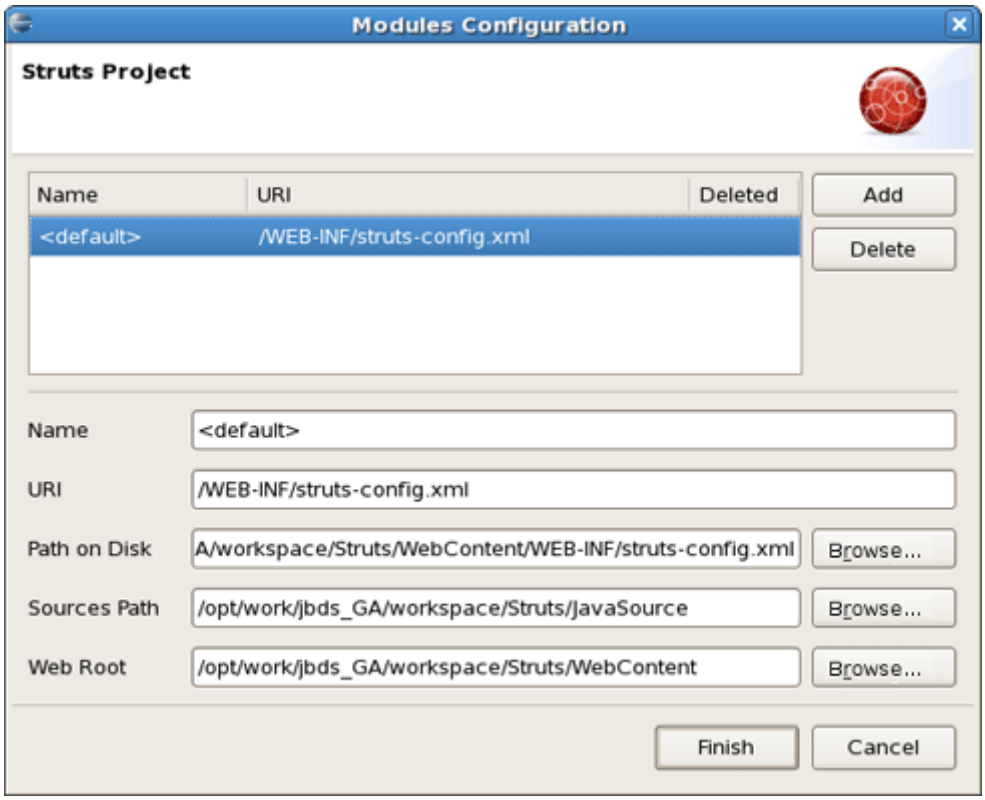

Figure 4.3. Modules Configuration

4.3. Adding New Modules

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

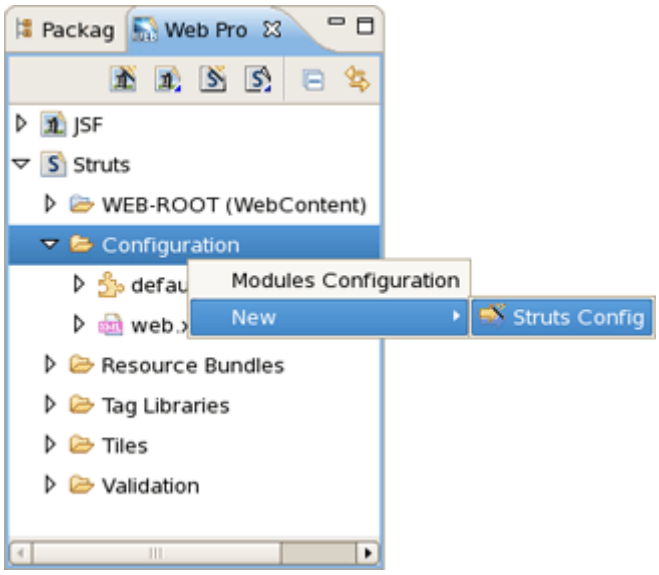

Figure 4.4. Adding New Modules

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

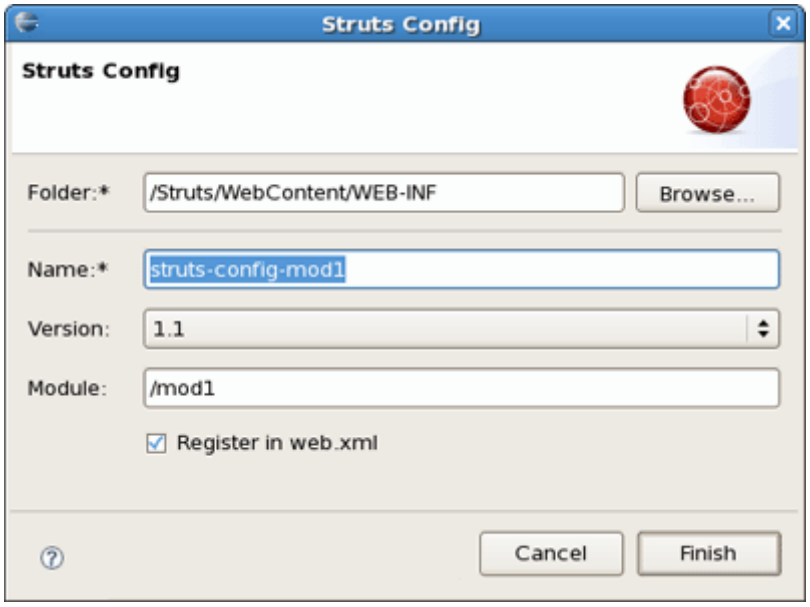

Figure 4.5. Adding New Modules

Code Generation

JBoss Tools comes with a code generation feature which can be used to generate stub code for Struts Actions, FormBeans, Forwards and Exceptions.

The code generation that JBoss tooling provides is based on Velocity templates which can be modified for your use. The templates are located at **{JBossStudioHome}** → **templates** → **codegeneration**.

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

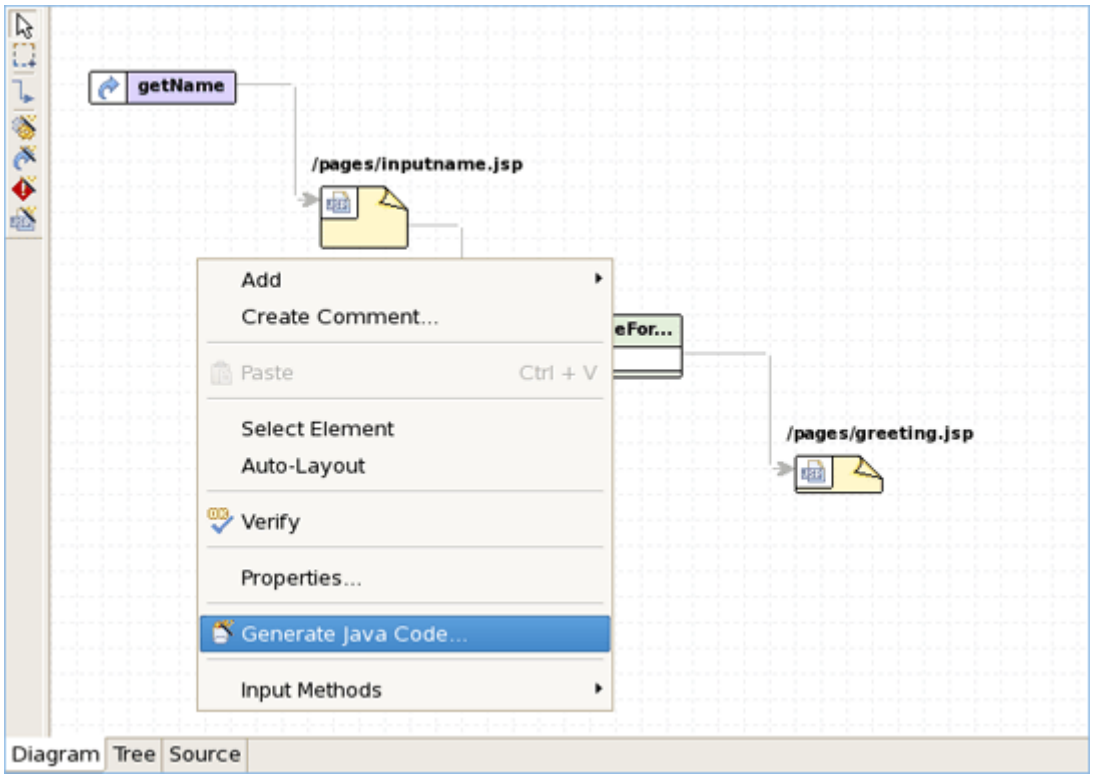

Figure 5.1. Selecting Generate Java Code

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

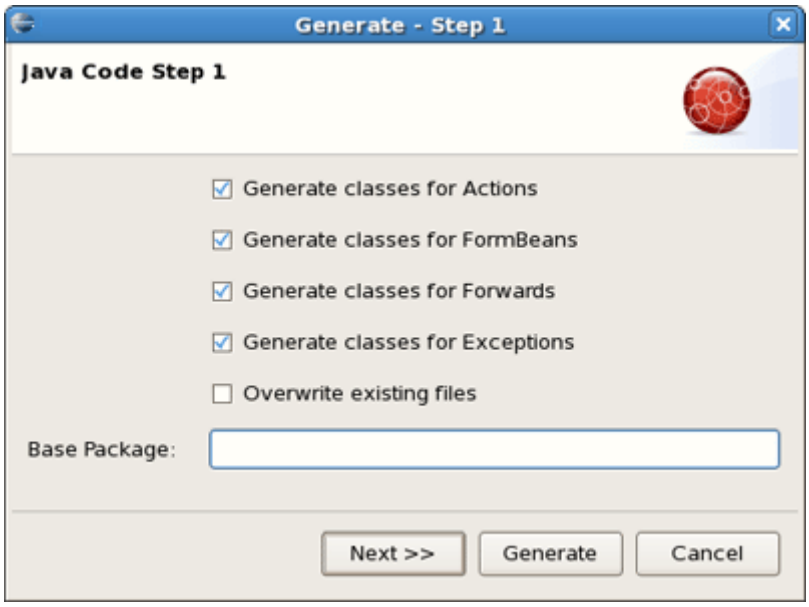

Figure 5.2. Generate - Step 1

Tip:

Please be careful not to override your existing files.

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

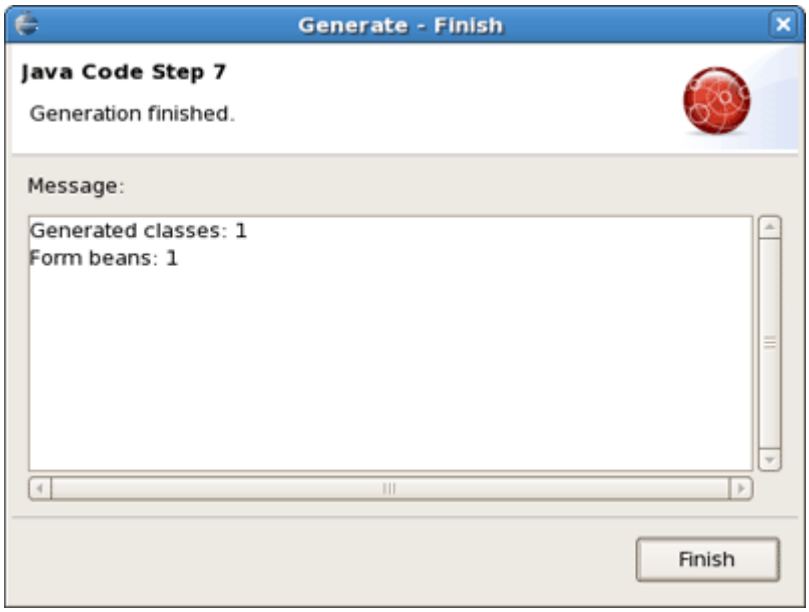

Figure 5.3. Generation Finished

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

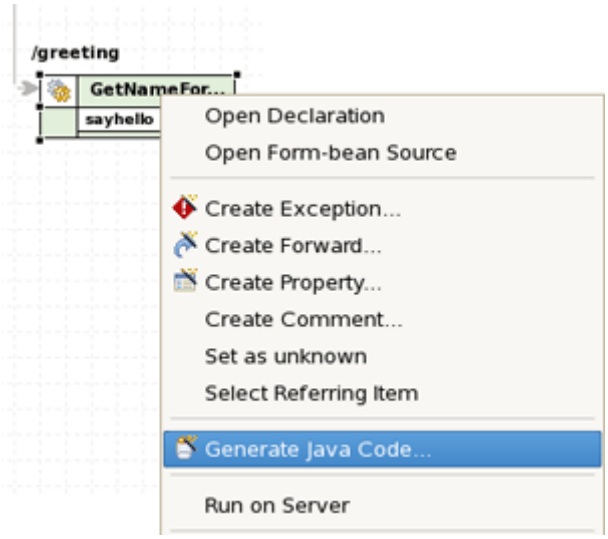

Figure 5.4. Generation For Individual Struts Artifact

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

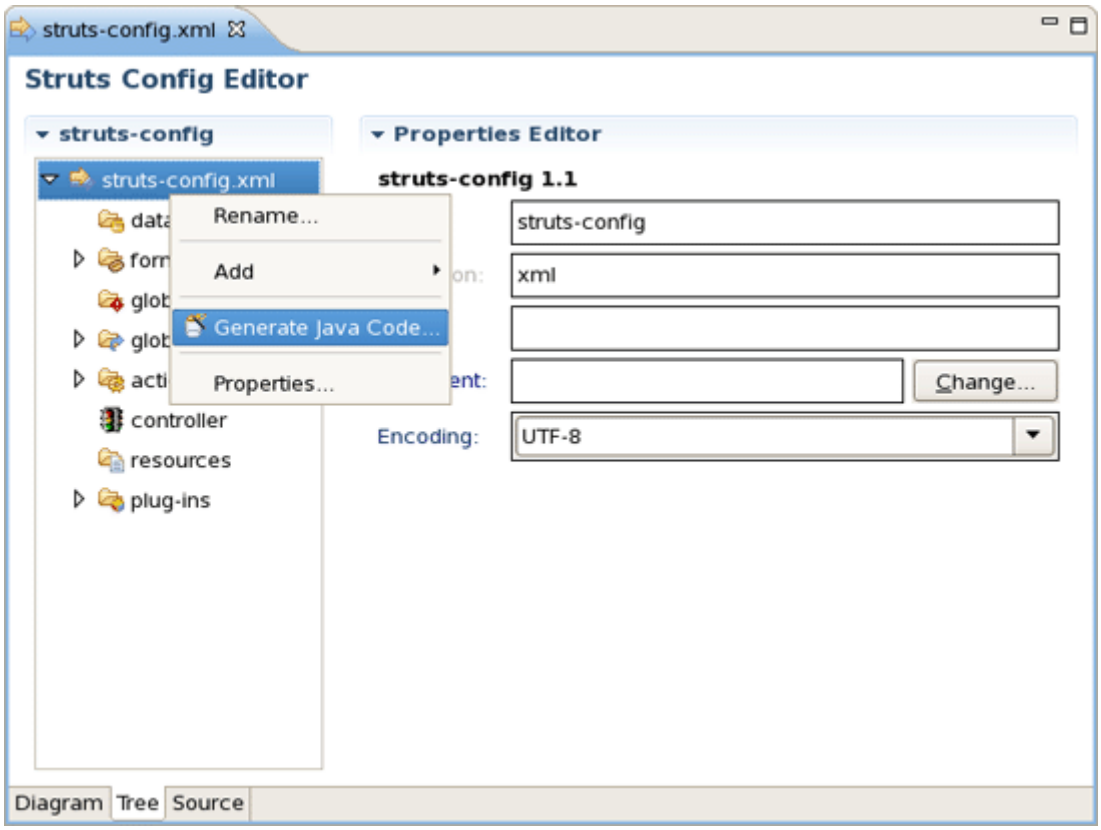

Figure 5.5. Generation in Struts Config Editor

Struts Configuration File Debugger

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

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

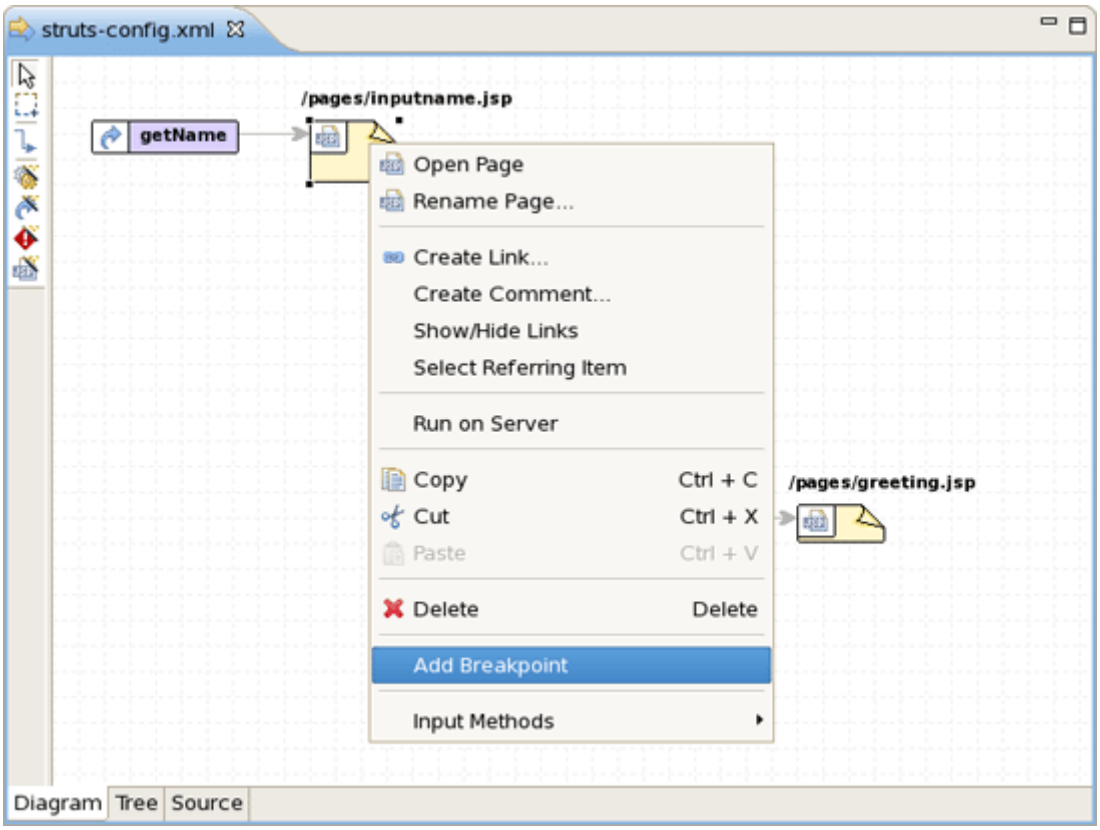

Figure 6.1. Adding Breakpoint

Customizable Page Links Recognizer

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

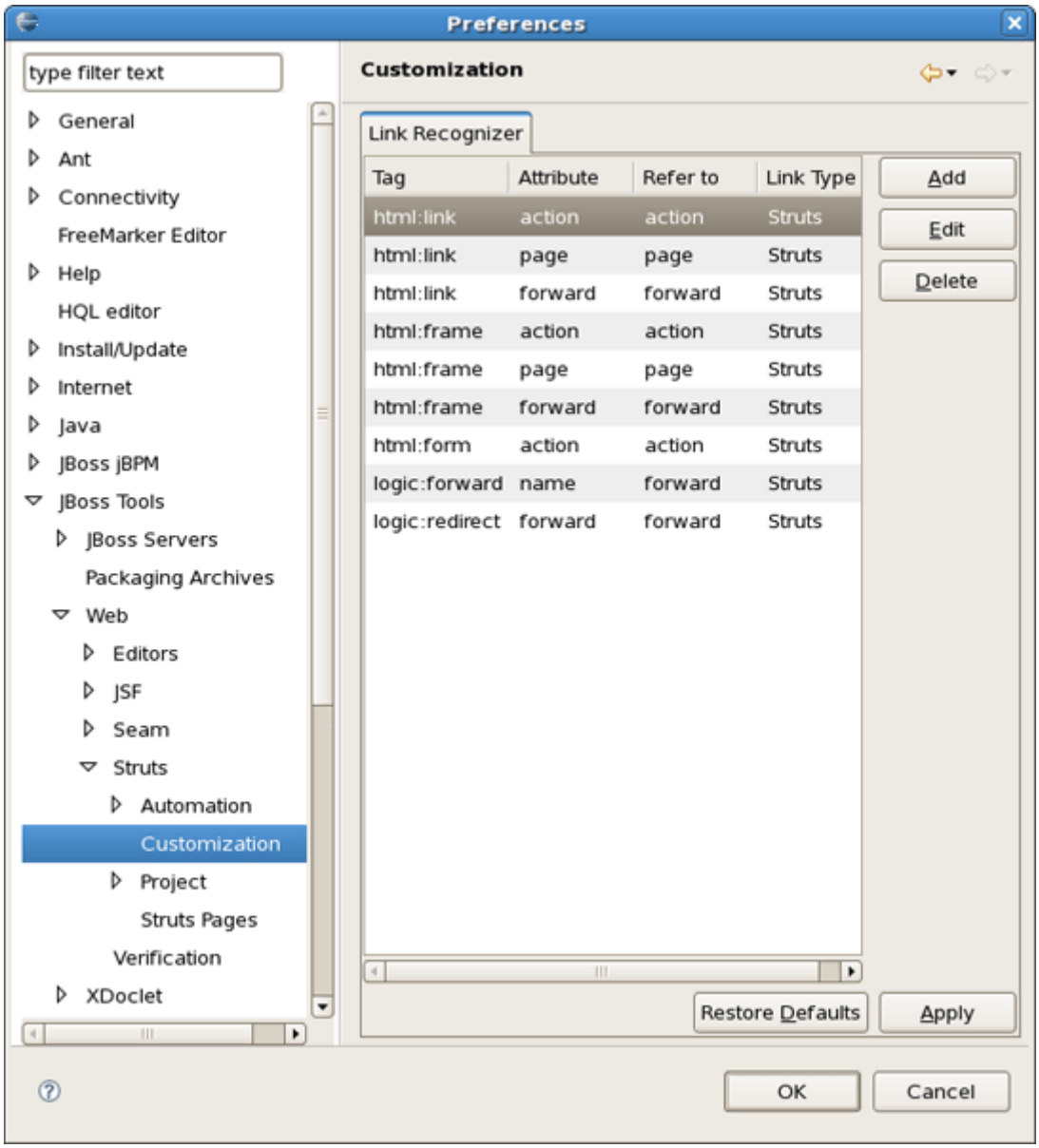

Figure 7.1. Customization Panel

Struts Project Verification

This section covers the Struts project verification functionality provided by JBoss Tools.

To configure Struts project verification select **Window** → **Preferences** from the menu bar, select **JBoss Tools** → **Web** → **Verification** from the Preferences dialog box, and then expand the Struts Rules node.

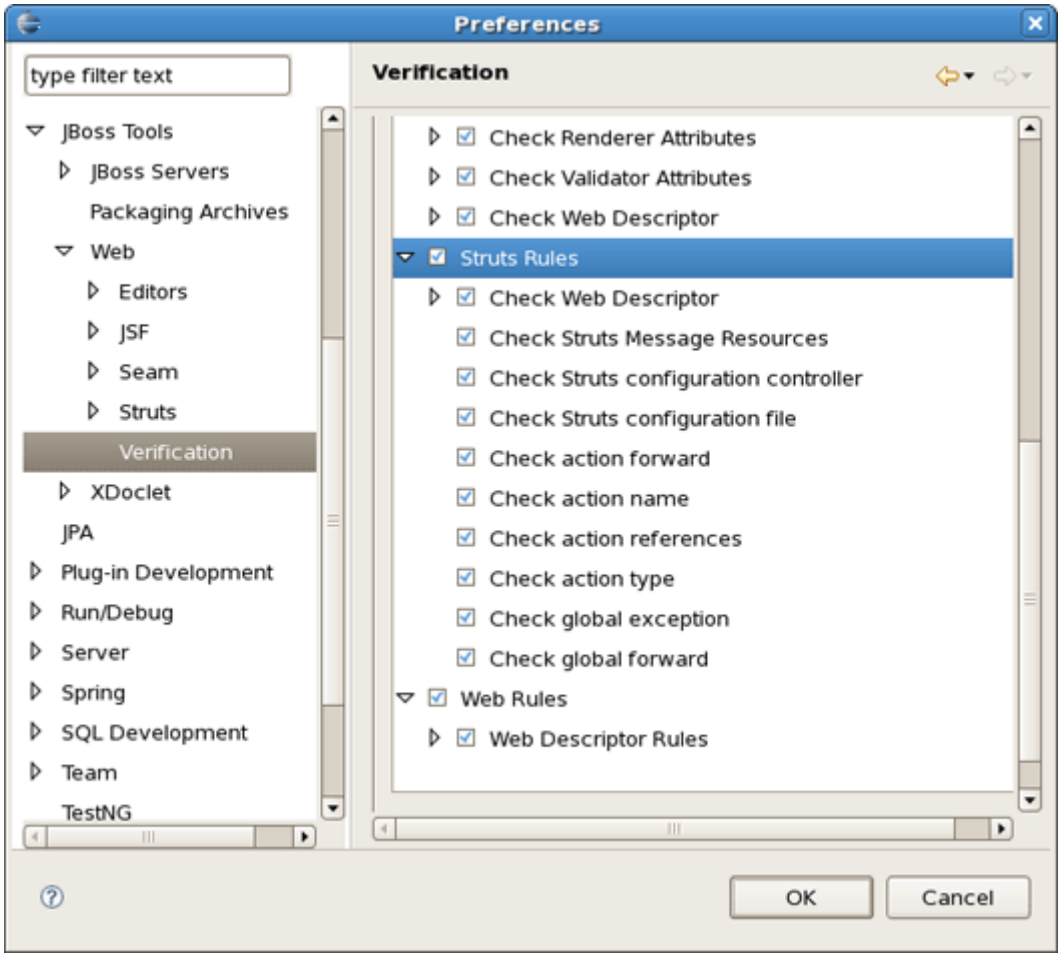

Figure 8.1. Struts Rules

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

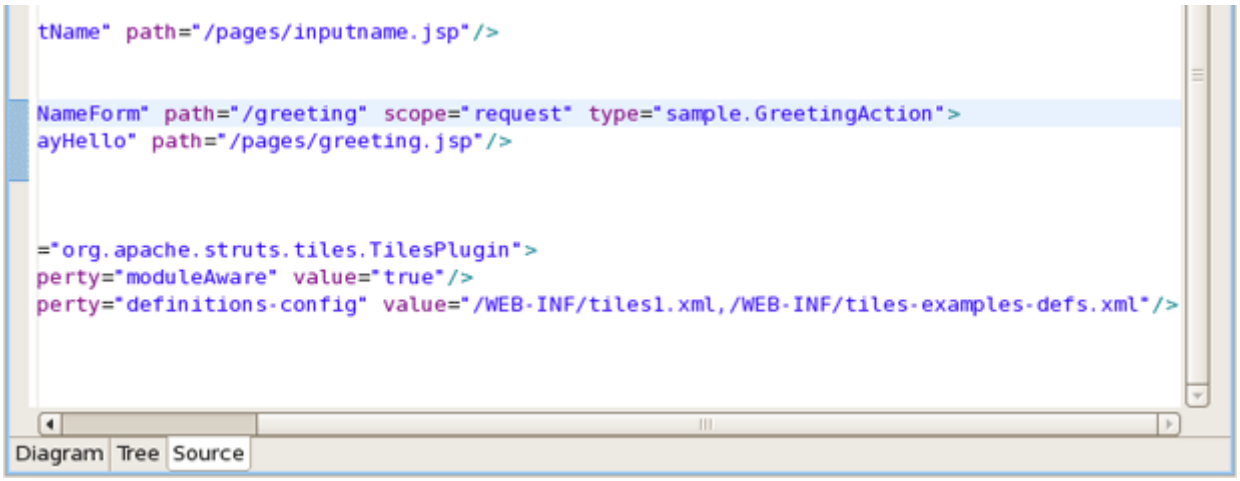

Figure 8.2. Struts Configuration File

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

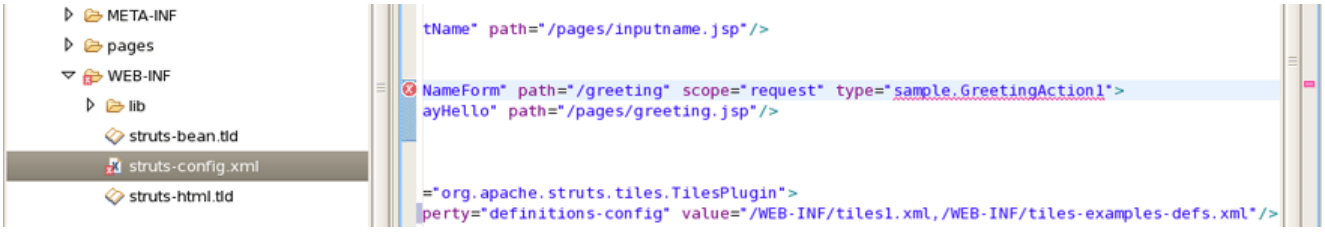

Figure 8.3. Error Reporting

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

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

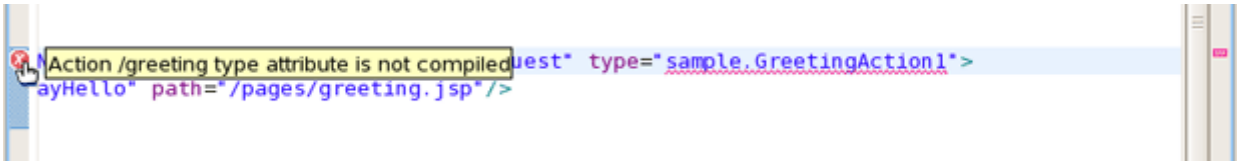

Figure 8.4. Error Message

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

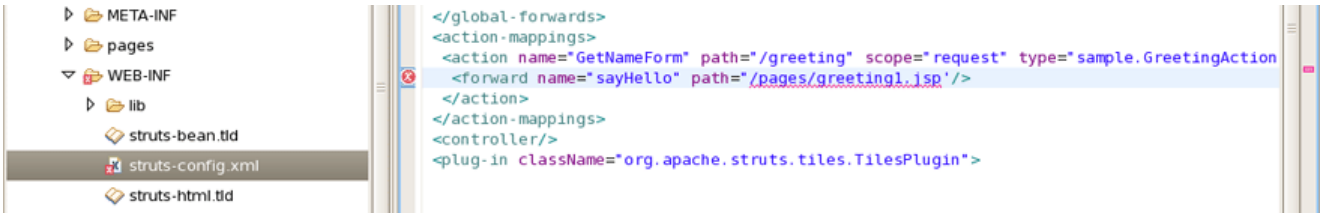

Figure 8.5. JSP Page Verification

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

Figure 8.6. Error Message

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

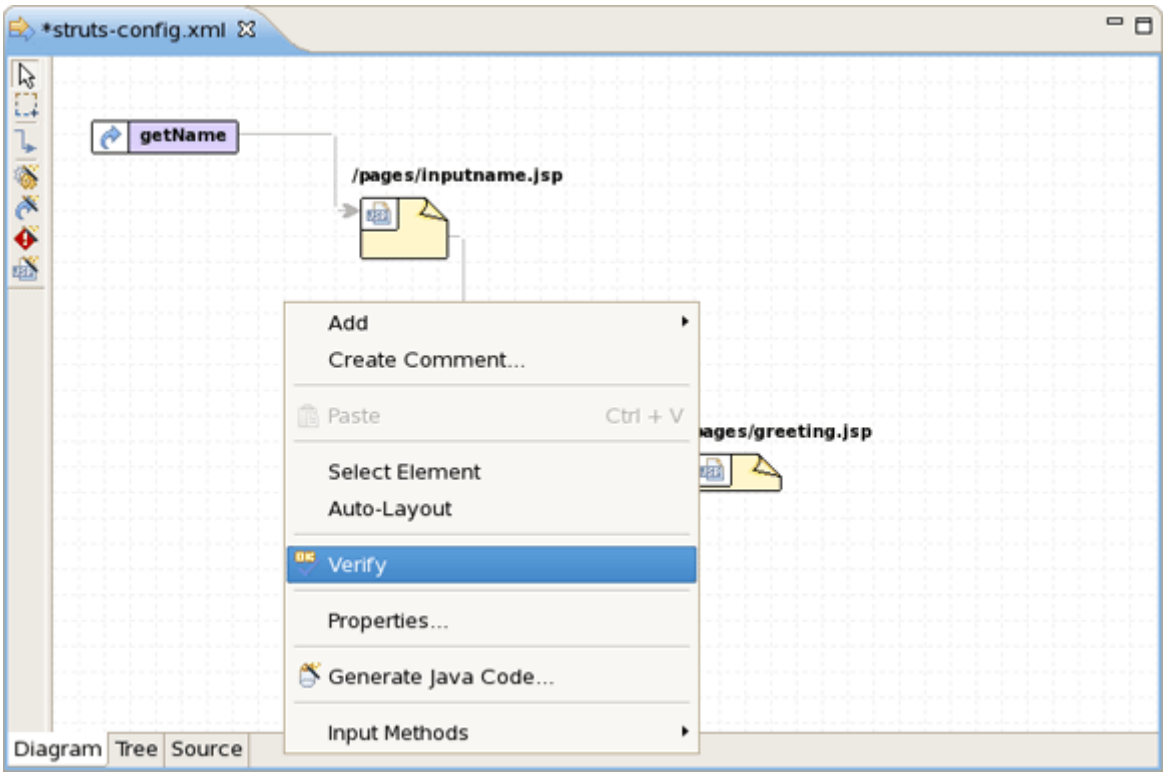

Figure 8.7. Verify Command

Relevant Resources Links

More information on Struts can be found on the *Struts website* [http://struts.apache.org/].

This reference has been desogned to help you become familiar with those parts of JBoss Tools which assist in developing using Struts technology. It covers how to create and import Struts projects, enable Struts capabilities for an existing web project, as well as organize and edit all necessary the necessary files in your Struts application using the views and editors provided by JBoss Tools.

Feedback is always appreciated. You can leave your questions and suggestions on our *Forum* [http://www.jboss.com/index.html?module=bb&op=viewforum&f=201].