

# Seam Developer Tools Reference Guide

Version: 4.2.0.Final-SNAPSHOT

---

---

---

<b>1. Introduction</b>	1
1.1. What is Seam?	1
1.2. Key Features of JBoss Seam	1
<b>2. Creating a New Seam Project via the New Seam Project wizard</b>	3
2.1. Create standalone Seam Web Project	3
2.2. Selecting the Project Facets	12
2.2.1. Seam project with JPA facets	15
2.3. Java application building configuration	17
2.4. How to Configure Web Module Settings	19
2.5. Adding JSF Capabilities	21
2.6. Configure Seam Facet Settings	30
<b>3. Maven Integration for Seam Projects</b>	35
3.1. Creating Maven ready Seam project	35
3.2. Existing Maven Projects Import	40
<b>4. Directory Structure of the Generated Project</b>	43
4.1. WAR Deployment	43
4.2. EAR Deployment	44
4.3. Changing the Seam Version	44
4.4. Changing Seam Parent Project	46
4.5. Renaming the Projects and Folders	47
<b>5. Seam Menus and Actions</b>	49
5.1. File Menu Actions	49
5.2. Navigate Menu Actions	49
5.2.1. Find Seam References/Declarations	49
5.2.2. Open Seam Component	51
<b>6. Seam Wizards</b>	53
6.1. New Seam Action	54
6.2. New Seam Form	57
6.3. New Seam Conversation	60
6.4. New Seam Entity	63
<b>7. Seam Generate Entities</b>	69
<b>8. Seam Editors</b>	71
8.1. Visual Page Editor	71
8.2. Seam Pages Editor	71
8.2.1. Graphical Mode	71
8.2.2. Tree Mode	75
8.2.3. Outline Support for Seam Pages Editor	75
8.3. Seam Components Editor	77
8.4. Main Features of Seam Editors	79
8.4.1. Content Assist	79
8.4.2. Content Assist for Page Descriptors	83
8.4.3. OpenOn	84
8.4.4. Seam Validation	88
<b>9. Seam Views</b>	93

9.1. Seam Components View .....	93
9.2. Project Explorer integration .....	97
<b>10. Seam Preferences .....</b>	<b>99</b>
10.1. General Preferences .....	99
10.1.1. Validator Preferences .....	101
10.2. Project Preferences .....	105
10.3. Adding Custom Capabilities .....	108
<b>11. Seam Refactoring Tools .....</b>	<b>111</b>
11.1. Using the Refactorinig Wizard .....	112
11.2. Renaming Seam Components .....	114
11.2.1. Renaming Seam Components in the Java Editor .....	114
11.2.2. Renaming Seam Components in the Seam Components View .....	115
11.2.3. Renaming Seam Components in the Seam Components Editor .....	116
11.3. Renaming Seam Context Variables in EL .....	116
<b>12. Adding Seam support to EAR project .....</b>	<b>121</b>
<b>13. Generate a CRUD Database Application .....</b>	<b>123</b>
13.1. What is CRUD? .....	123
13.2. How to create the CRUD Database Application with Seam .....	123
<b>14. The CRUD Application Walkthrough .....</b>	<b>145</b>
14.1. Using CRUD Application .....	145
<b>15. What is TestNG? .....</b>	<b>151</b>
15.1. How to use the generated Seam-test project to run Seam tests? .....	151
<b>16. FAQ .....</b>	<b>159</b>
16.1. How to get Code Assist for Seam specific resources in an externally generated project? .....	159
16.2. How to import an example Seam project from jboss-eap directory? .....	159
16.3. How to change the deploy folders for Seam source code from src/hot and src/main? .....	159

# Introduction

## 1.1. What is Seam?

This introduction will help you to understand what is Seam.

Seam is a fully featured application framework on top of Java EE 5. It is also one of the most popular enterprise Java frameworks today. Seam deeply integrates many other standard-based or open source frameworks (e.g., JSF, EJB3, JMS, Web Services, jBPM, JBoss Rules, Ajax4jsf, RichFaces, Facelets, Spring, iText, Quartz, TestNG, etc.), and provides a single programming model for developers to "drive" those underlying frameworks via simple annotated POJOs (Plain Old Java Objects). It makes life easier for developers to tackle complex enterprise applications with many component frameworks.

## 1.2. Key Features of JBoss Seam

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

**Table 1.1. Key Functionality for JBoss Seam**

Feature	Benefit	Chapter
Integrated JSF with EJB 3.0	JBoss Seam combines the component models of EJB3 and JSF, eliminating glue code, and thus, letting the developer think about the business problem.	
Integrated AJAX	Developer can add AJAX capability to the user interface of his application without the need to write any JavaScript code as Seam supports JBoss RichFaces and ICEfaces - the best open source JSF-based AJAX solutions.	
Bijection	Bijection allows auto-assembly of stateful components by the container. It even allows a component to safely and easily manipulate the value of a context variable, just by assigning it to an attribute of the component.	
New Seam Project wizard	An easy step-by-step wizard to create a stand alone Seam Web Project with a form to create runtime and server instances in order to get started creating, running, and debugging J2EE (only) applications. You may also add JSF capabilities.	<a href="#">Chapter 2, Creating a New Seam Project via the New Seam Project wizard</a>
Seam component generations options	All the Seam component generations options known from Seam-gen are available as	<a href="#">Chapter 6, Seam Wizards</a>

Feature	Benefit	Chapter
	wizards (with sensible auto-defaulting) for creating various common Seam components.	
Seam Editors	Visual Page Editor, Seam Pages Editor and Seam Components Editor provide you with a range of great possibilities and will be helpful when working with JBoss Seam.	<a href="#">Chapter 8, Seam Editors</a>
Seam Validator	A configured Seam Validator provides validation of various possible problematic definitions.	<a href="#">Section 8.4.4, “Seam Validation”</a>
CRUD Database Application Generation	JBoss Seam allows adding CRUD support to a Seam Web application.	<a href="#">Chapter 13, Generate a CRUD Database Application</a>
TestNG Framework	TestNG ("Testing, the Next Generation") is a Java unit testing framework that aims to overcome many limitations of JUnit.	<a href="#">Chapter 15, What is TestNG?</a>

# Creating a New Seam Project via the New Seam Project wizard

In this chapter we provide you with the necessary steps to start working with Seam Framework.

It is suggested that you use the specific Seam perspective that combines a number of different views and editors needed for work with resources concerned. For that select **Window** → **Open Perspective** → **Other** → **Seam**.

The **Open Web Browser** action is directly available in the Seam perspective.



**Figure 2.1. Embedded Web Browser Button**

## 2.1. Create standalone Seam Web Project

The best way to get started with Seam is to create a simple Seam Project and experiment with it.


Thus, you should select **File** → **New** → **Other** → **Seam** → **Seam Web Project** to run the New Seam Project wizard. The wizard form allows you to create runtime and server instances in order to get started creating, running, and debugging J2EE (only) applications.

Seam Web Project wizard has an option for selecting the actual Server (not just WTP runtime) that will be used for the project. This allows the wizard to identify correctly where the required datasource and driver libraries need to go.

Let's get through the wizard step-by-step. First, you should enter a name and a location directory for your new project.

## Seam Web Project

Create standalone Seam Web Project



Project name:

Project location

☒ Use default location

Location:

Target runtime

Dynamic web module version

Target Server

Configuration

Configures a Dynamic Web application to use Seam v2.3 (Technical Preview)


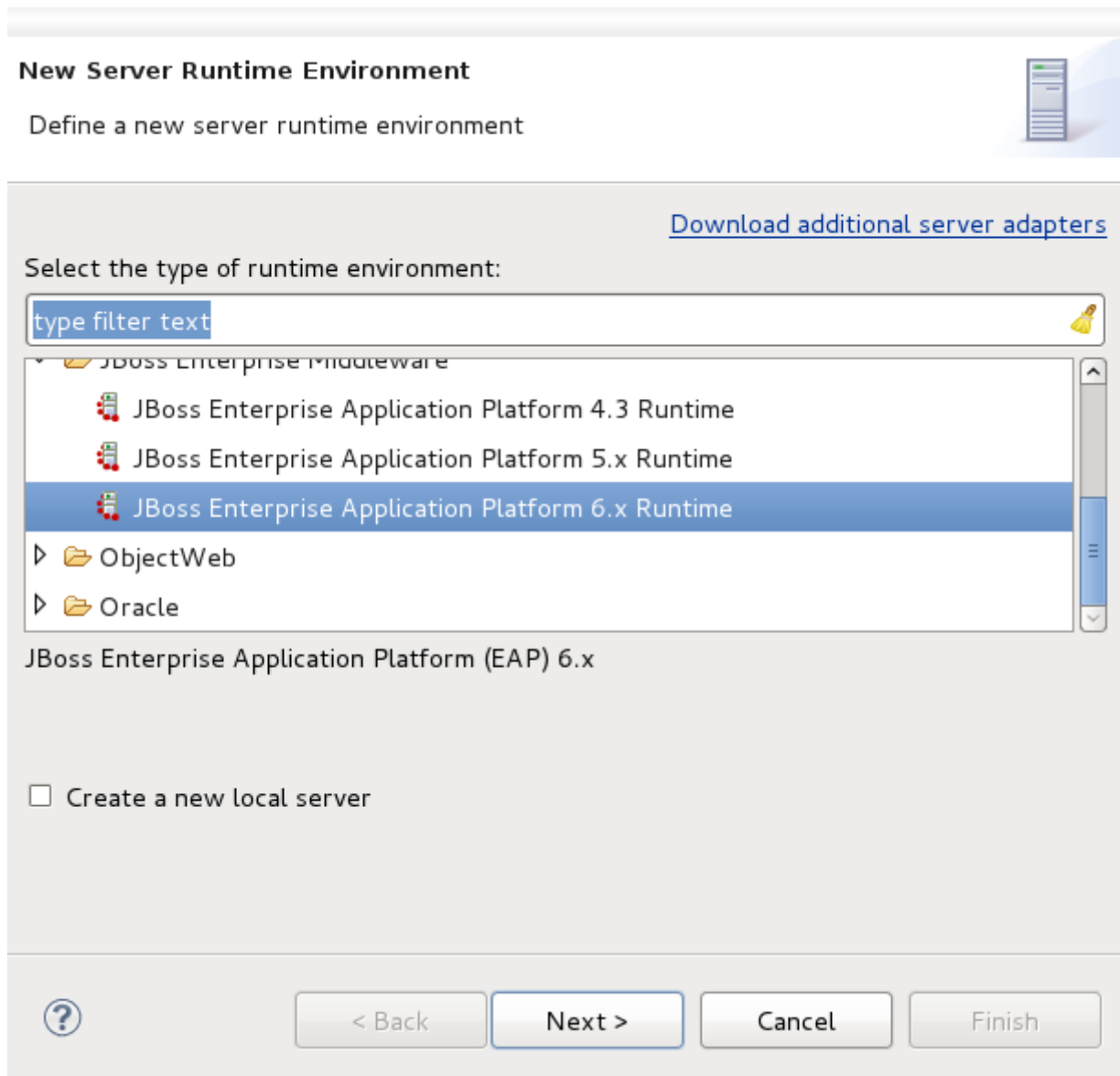


Figure 2.2. New Seam Project Wizard



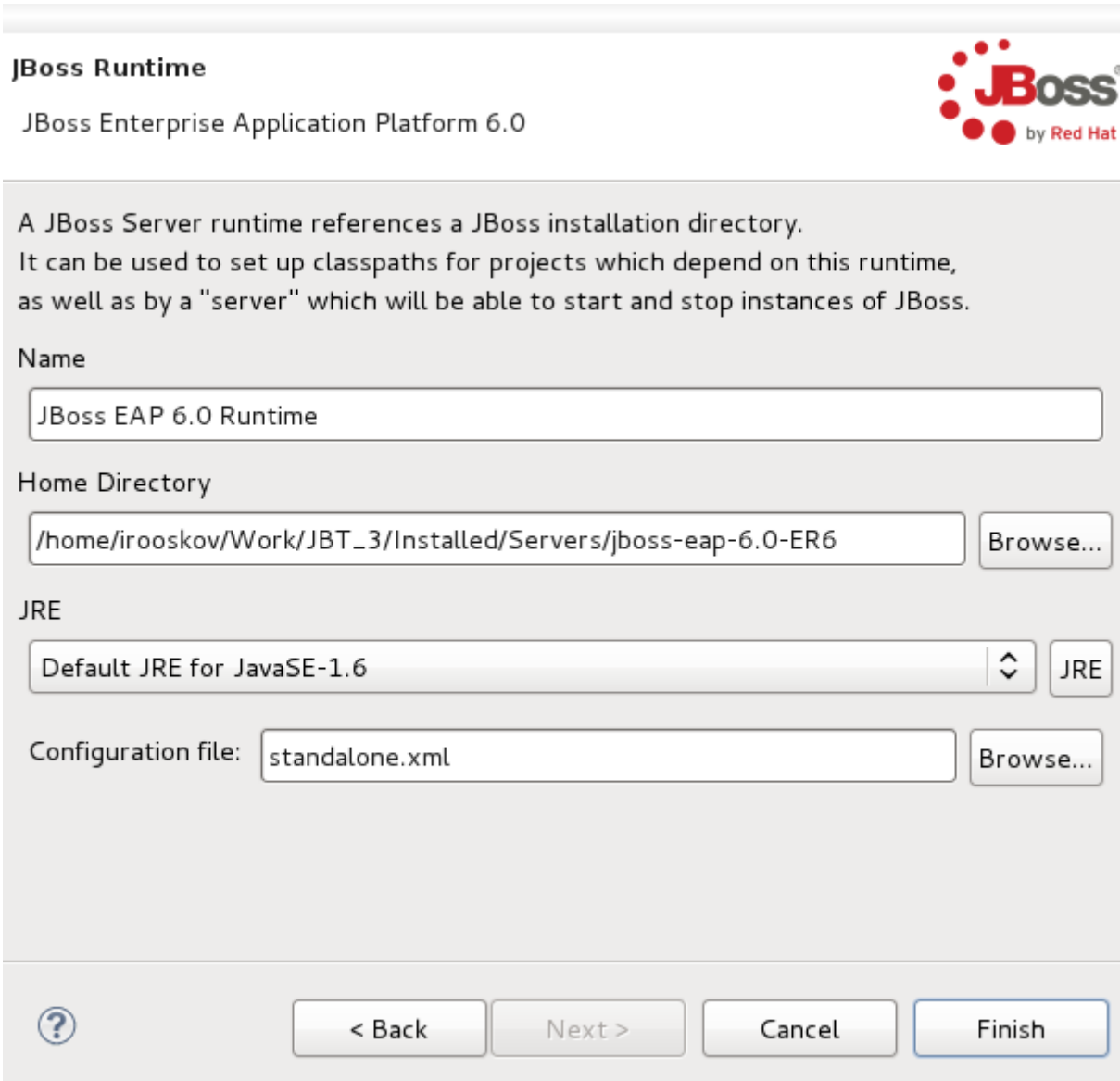
On the figure above you can see the runtime and the server already created.

If you need to create a new runtime, click on the **New** button in the Target Runtime section. It brings up the wizard where you can specify a new JBoss Server Runtime environment or the other type of runtime appropriate for your project configuration. Let's create one more JBoss Enterprise Application Platform 6.0 Runtime. Hence, after choosing it click on **Next** button.



**Figure 2.3. Specifying Target Runtime**

All what you need here is to name runtime, type the path to its install directory or locate it by using **Browse** button, select a Java Runtime Environment, and select which configuration you want.



**JBoss Runtime**

JBoss Enterprise Application Platform 6.0

A JBoss Server runtime references a JBoss installation directory.  
It can be used to set up classpaths for projects which depend on this runtime,  
as well as by a "server" which will be able to start and stop instances of JBoss.

Name

Home Directory

JRE

Configuration file:

**Figure 2.4. Specifying Target Runtime Configurations**

Clicking on **Finish** returns you to the [Figure 2.2, "New Seam Project Wizard"](#).

The next step is to define a Server by clicking on **New** button in the Target Server section. In appeared New Server dialog the last server which matches the runtime will be selected.

All declared runtimes are listed in the combobox under the servers view. Here, you can indicate a server runtime that you need. Click **Add** if you want to add a new Server Runtime.

**Define a New Server**

Choose the type of server to create

[Download additional server adapters](#)

Select the server type:

type filter text

- JBoss Enterprise Application Platform 5.x
- JBoss Enterprise Application Platform 6.x**

JBoss Enterprise Application Platform (EAP) 6.x

Server's host name: localhost

Server name: JBoss EAP 6.0 Runtime Server (1)

Server runtime environment: JBoss EAP 6.0 Runtime [Add...](#)

[Configure runtime environments...](#)


< Back Next > Cancel Finish

**Figure 2.5. Specifying Target Server**

Next page allows you to verify the information for a chosen server. Leave everything as it is and click on **Next**.

Create a new JBoss Server

JBoss Enterprise Application Platform 6.0



A JBoss Server manages starting and stopping instances of JBoss.  
It manages command line arguments and keeps track of which modules have been deployed.

Runtime Information

If the runtime information below is incorrect, please press back, Installed Runtimes..., and then Add to create a new runtime from a different location.

Home Directory	/home/irooskov/Work/JBT_3/Installed/Servers/jboss-eap-6.0-ER6
Execution Environment	Java Platform, Standard Edition 6.0
JRE	Default JRE for JavaSE-1.6


Server Behaviour

☐ Server is externally managed. Assume server is started.

☐ Listen on all interfaces to allow remote web connections

☐ Expose your management port as the server's hostname

Local



< Back

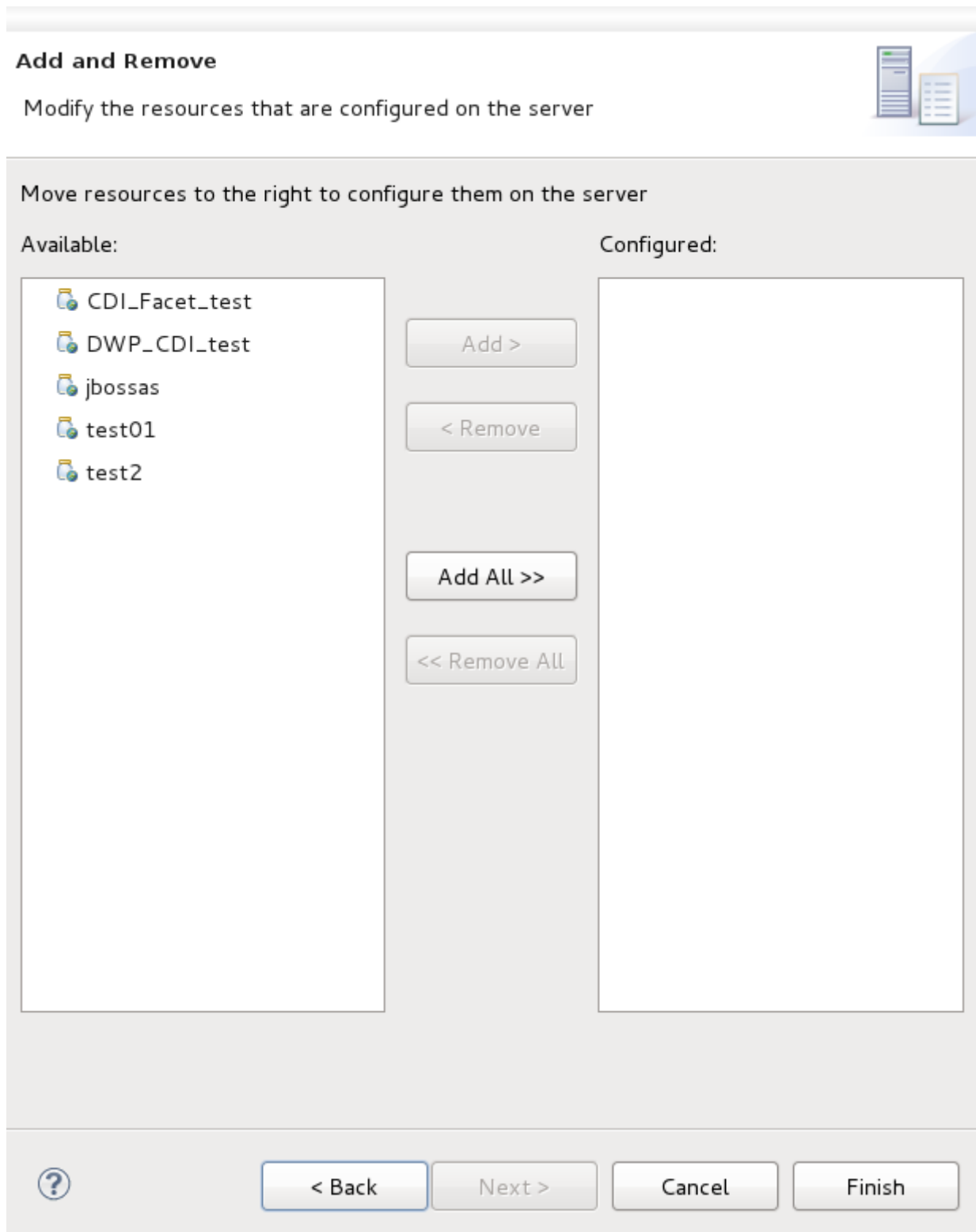
Next >

Cancel

Finish

**Figure 2.6. JBoss Runtime Summary**

On the last wizard step you can modify your projects to configure them on the Server.



**Figure 2.7. Project Modification for Configuring on the Server**

Once you have the Target Server defined click on **Finish** button to return to the first page of the New Seam Project wizard.




### Note

Refer to the Application Server Manager Guide to find out more about runtimes and servers.

## Seam Web Project

Create standalone Seam Web Project



Project name:

Project location

☒ Use default location

Location:

Target runtime

Dynamic web module version

<custom>

BIRT Charting Web Project

BIRT Reporting Web Project

Default Configuration for JBoss EAP 6.0 Runtime

Dynamic Web Project with CDI (Context and Dependency Injection)

Dynamic Web Project with Seam 2.3

JavaServer Faces v2.0 Project

Minimal Configuration




Figure 2.8. Completion of Runtime and Server Configuration

11

The last section on this wizard step is Configuration. Here, you can select one of the predefined project configurations either associated with Seam 1.2, Seam 2.0 or with Seam 2.1. Furthermore, you can create your own configuration by pressing the **Modify** button. It will open the dialog which allows to configure your own set of facets for adding extra functionality to your project.

Pass to the next section to find out more details on this dialog.

## 2.2. Selecting the Project Facets

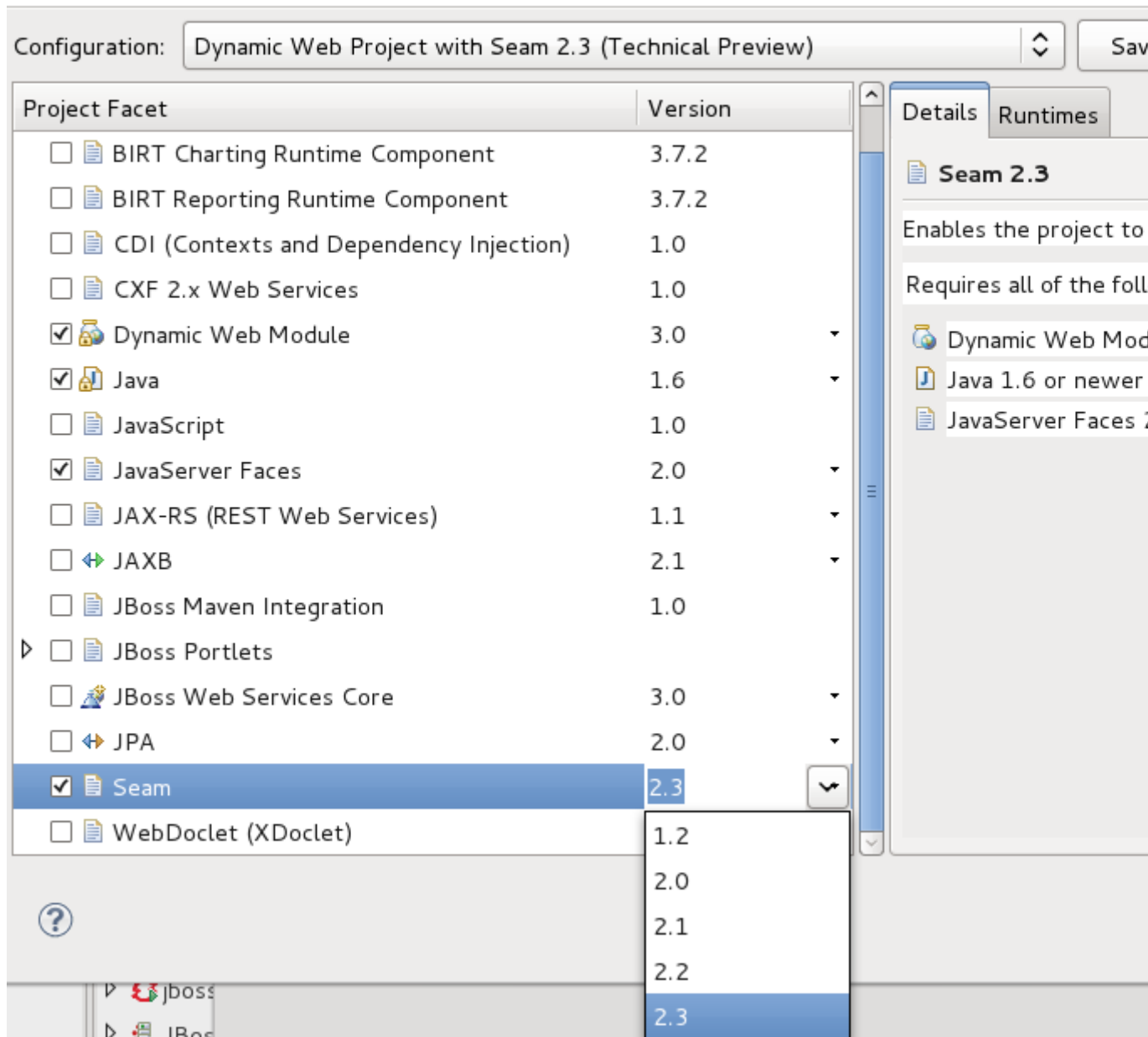
The Project Facets wizard allows you to enable or disable specific facets which define necessary features for the project. When you switch to this wizard form, all critical facets are already checked for the chosen Configuration.

Notice that this page of the wizard also allows you to set the necessary version for any facet.



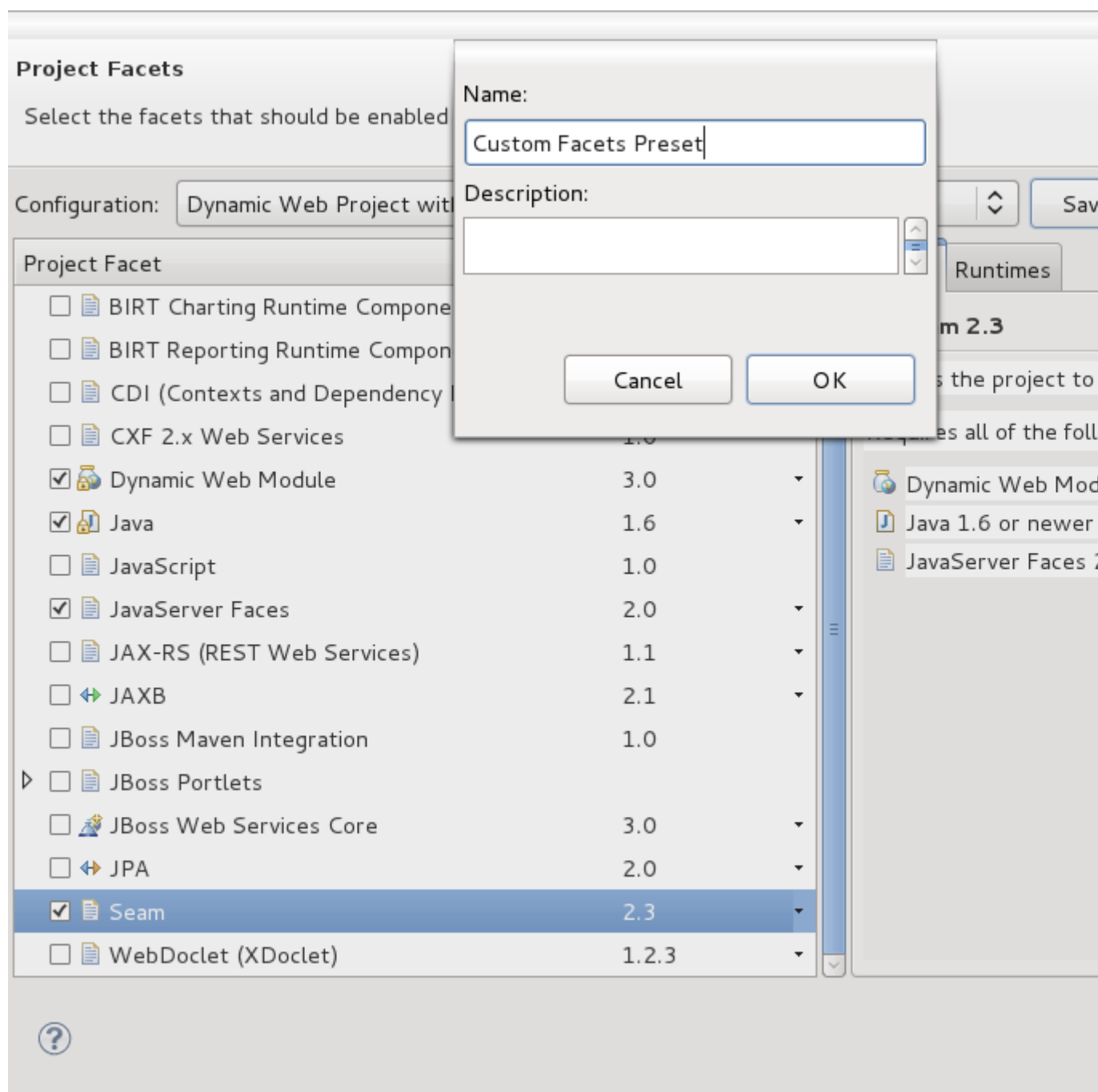
## Project Facets

Select the facets that should be enabled for this project.



**Figure 2.9. Project Facets Selection**

Moreover, here you can specify your own preset of selected facets by checking needed ones in project facets window and clicking on the **Save as** button.

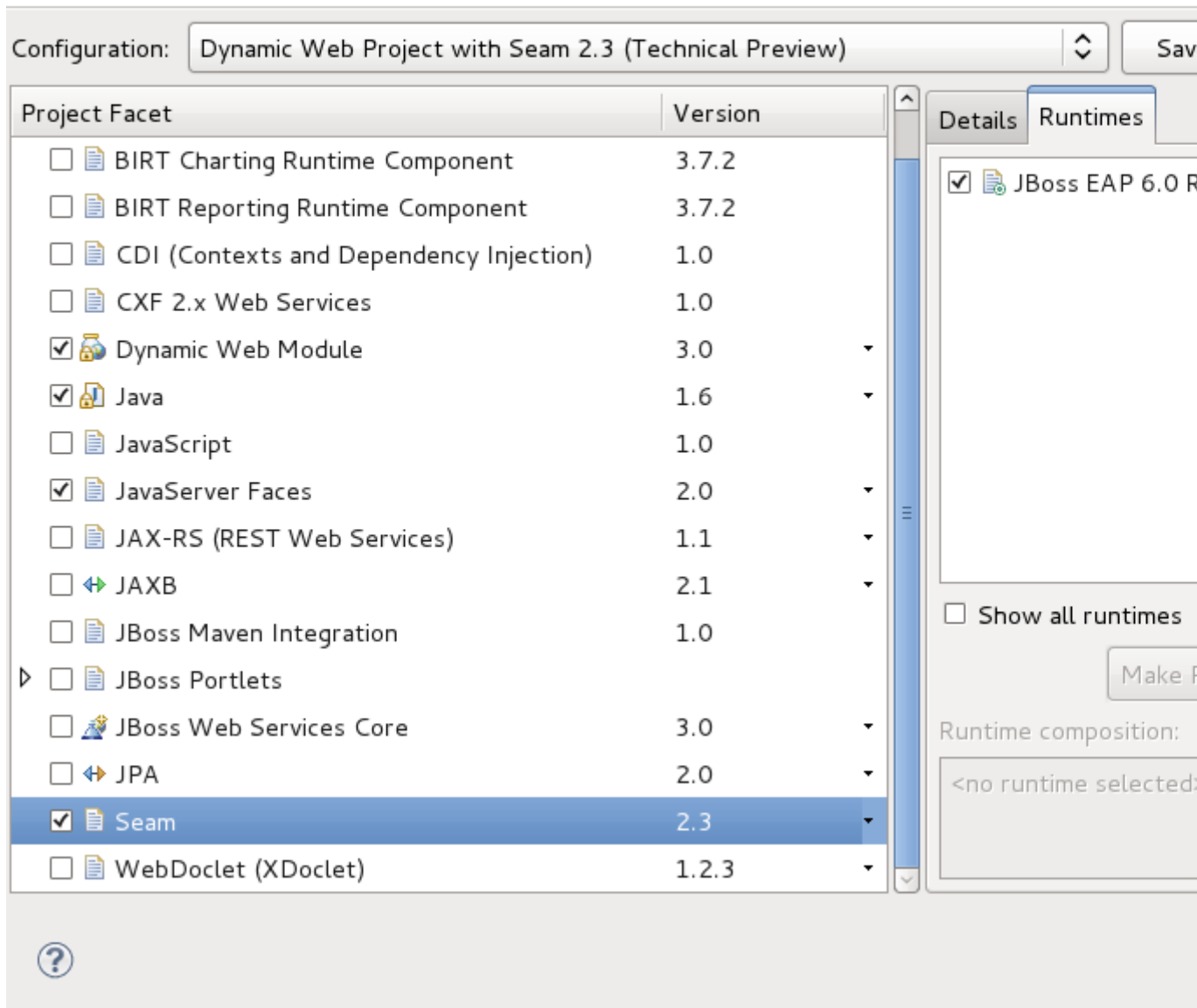


**Figure 2.10. Specifying Custom Facet Preset**

To see all available Server runtimes click on Runtimes tab on the left. You can create a new one using the **New** button. If more than one runtime is checked here, the **Make Primary** button will not be dimmed yet. Thus, you can make use of it to mark primary runtime.

## Project Facets

Select the facets that should be enabled for this project.



**Figure 2.11. Setting Runtime**

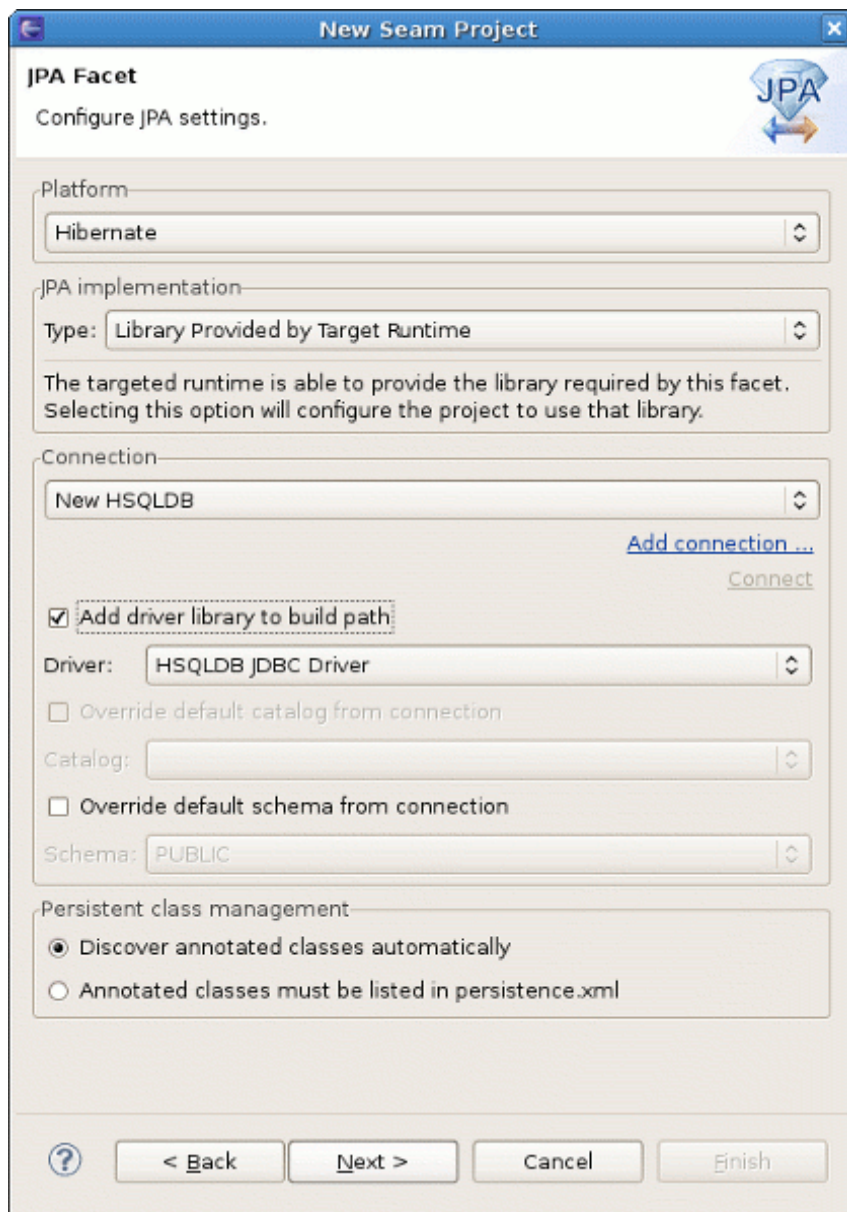
Clicking the **OK** button will bring you to the Web Module wizard form again.

### 2.2.1. Seam project with JPA facets

For creating a Seam project with JPA support you need to select Java Persistent facet on Project Facets page and click **OK**.

Then press Next and you can use the default value on the Java application building page

After that you will see JPA facet page.



**Figure 2.12. JPA Facet page**

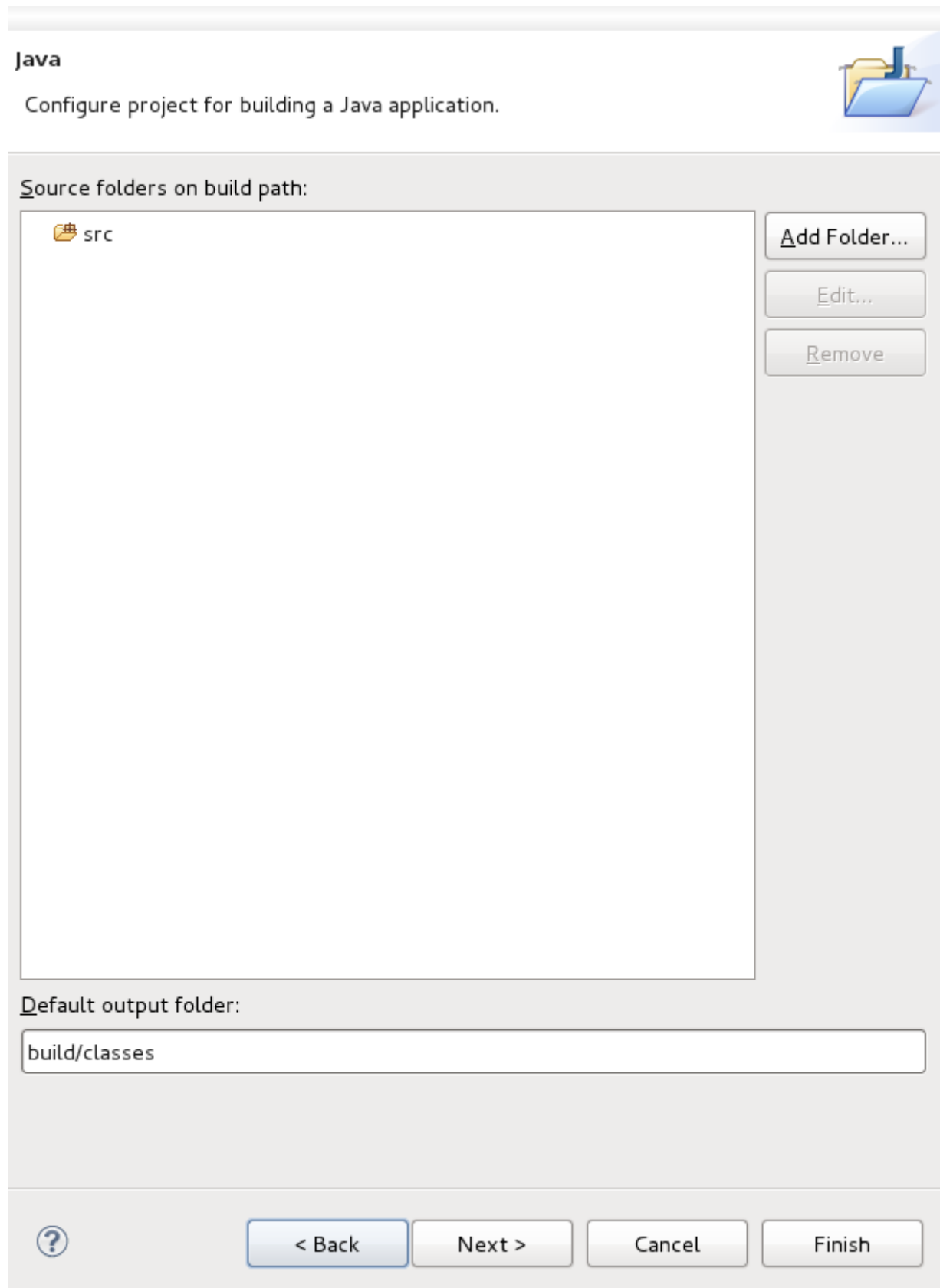
On this page you can choose Platform (vendor-specific JPA implementation) for your project, type of JPA Implementation, and configure the Connection to database in the Connection area.

The next page concerns Web Module Settings. Accept the default values and press the **Next** button.

On the next step you will see JSF Capabilities and Seam facet pages, which are mostly the same as in general Seam project.

## 2.3. Java application building configuration

With this wizard you can define Java source directories which will be generated on build path. Also you can specify a default output folder. If you are agree with default values, press **Next**



**Figure 2.13. Java application building Wizard**

## 2.4. How to Configure Web Module Settings

As we deal with a Dynamic Web Application, we should at first specify the top level directory of our application for deploying it to a server afterwards. This kind of application contains Web resources, so it is important to indicate the content directory. The wizard will put all those values itself, so you can leave everything as is.

**Web Module**

Configure web module settings.

Context root:

Content directory:

☐ Generate web.xml deployment descriptor



**Figure 2.14. Web Module Settings**



Choose **Next** to switch to the next wizard form.


## 2.5. Adding JSF Capabilities

This wizard helps you to add JSF capabilities to your project.

Choose *Library provided by Target Runtime* from Library Type list if you'd like to use a default JSF implementation given by the present runtime.

### JSF Capabilities

Add JSF capabilities to this Web Project



JSF Implementation Library

Type: Library Provided by Target Runtime

The targeted runtime is able to provide the library required by this facet. Selecting this option will configure the project to use that library.

☒ Configure JSF servlet in deployment descriptor

JSF Configuration File: /WEB-INF/faces-config.xml

JSF Servlet Name: Faces Servlet

JSF Servlet Class Name: javax.faces.webapp.FacesServlet

URL Mapping Patterns: \*.seam

Add...  
Remove

?

< Back

Next >


Cancel


Finish

Figure 2.15. Adding JSF Capabilities to Web Project

If you prefer to use your custom JSF implementation, choose User Library Item from Library Type list. In User Library list you can check required library.

### JSF Capabilities

 At least one user library must be selected.



JSF Implementation Library

Type: User Library

☒ Include libraries with this application

☒ Configure JSF servlet in deployment descriptor


JSF Configuration File: /WEB-INF/faces-config.xml

JSF Servlet Name: Faces Servlet

JSF Servlet Class Name: javax.faces.webapp.FacesServlet

URL Mapping Patterns: \*.seam

Add...Remove



< BackNext >CancelFinish

Figure 2.16. Using Custom JSF implementation Library

If a new library is required, click Manage libraries... (



) button. Then you should click New.

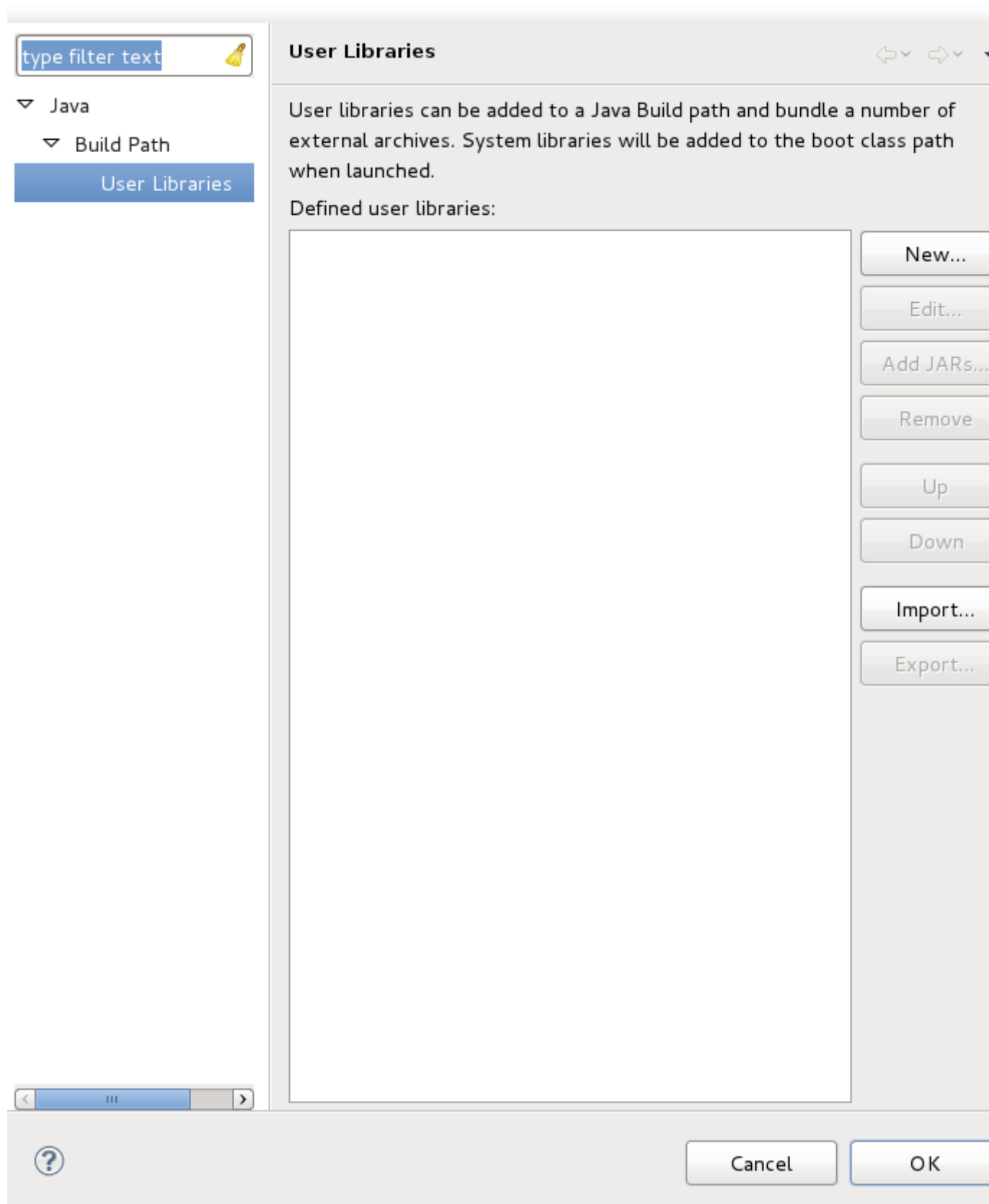
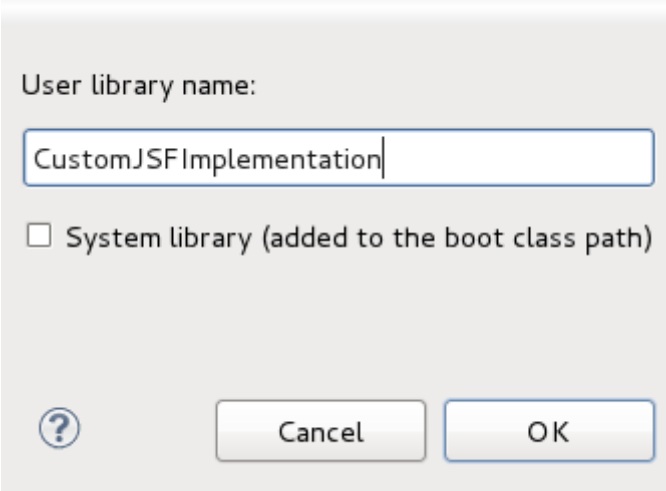


Figure 2.17. Using Custom JSF implementation Library

Here, it's necessary to type a Library Name.



User library name:

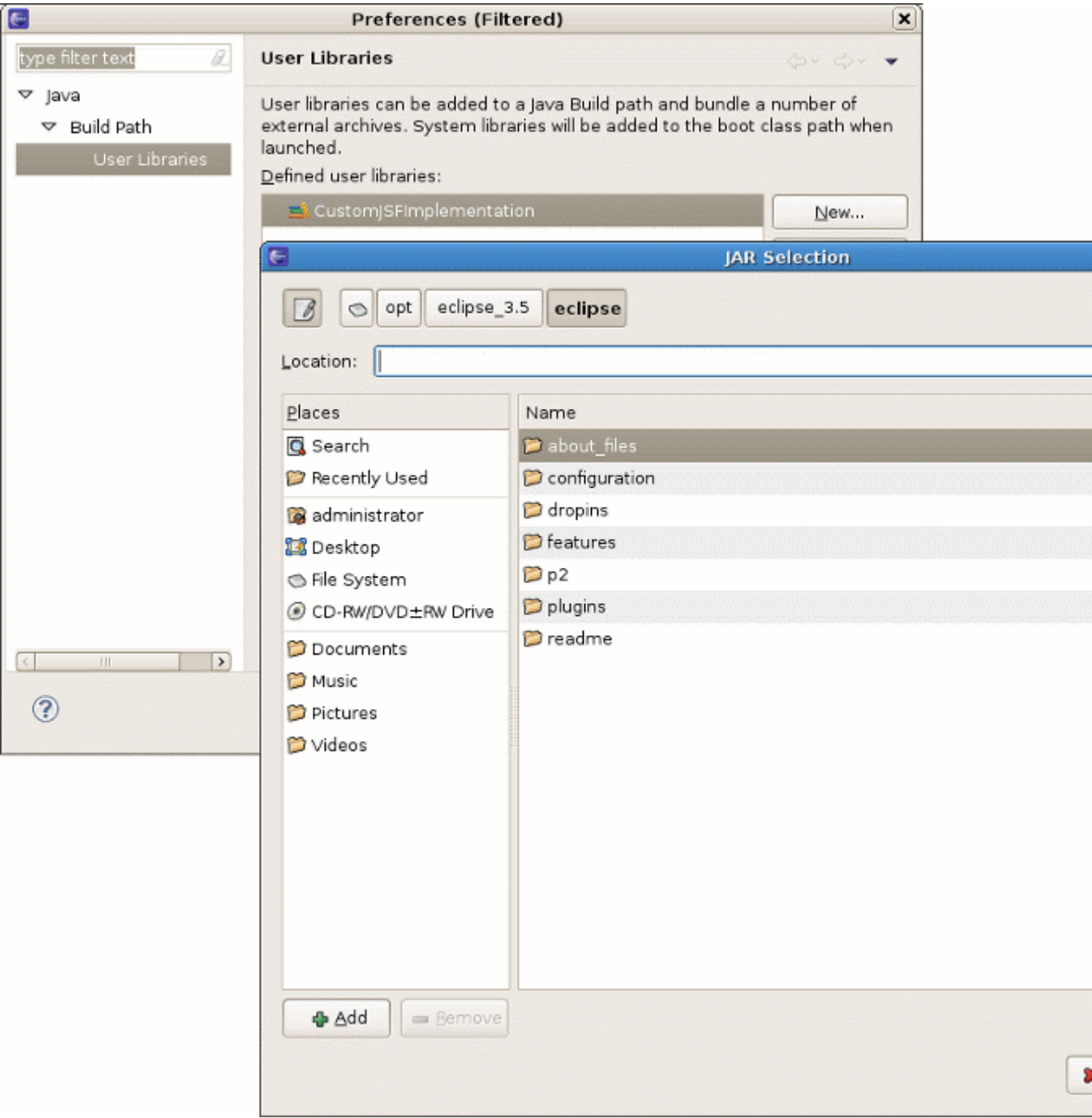
CustomJSFImplementation

☐ System library (added to the boot class path)


? Cancel OK

**Figure 2.18. Entering New User Library Name**


To add proper Library JARs, click Add JARs... button and select the JARs on your hard drive.



**Figure 2.19. Selection Library JARs**

You can download necessary JSF implementation libraries provided by Sun and Apache Foundation as well. Click Download... (  ) button and choose one of proposed libraries.




**Download Library**

Select a library to download from the specified provider.

Library	Download Provider
JSF 2.0 (Apache Myfaces JSF Core-2.0 API : The Apache Software Foundation	
JSF 2.0 (Mojarra 2.0.3-FCS)	Oracle Corporation

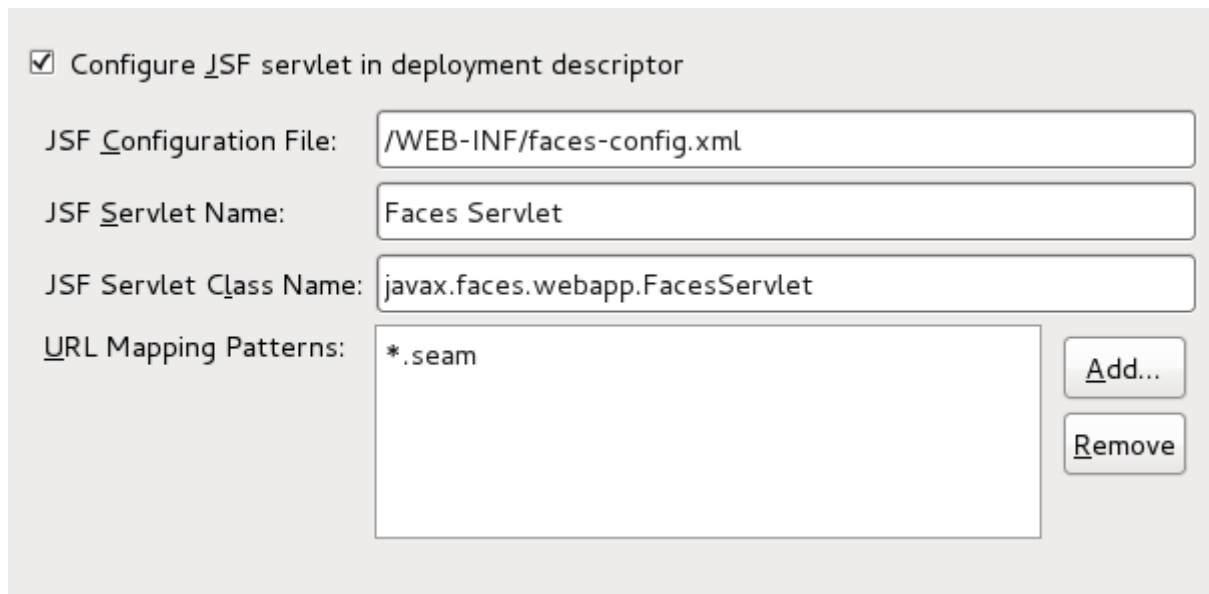
Library name:

Download destination:



**Figure 2.20. Download JSF Implementation Library**

The last wizard options allows to edit a path for JSF Configuration File, a name for JSF Servlet, JSF Servlet Classname and change URL Mapping Patterns.



The image shows a 'JSF Capabilities Wizard' dialog box. At the top, there is a checked checkbox labeled 'Configure JSF servlet in deployment descriptor'. Below this, there are four input fields: 'JSF Configuration File' with the value '/WEB-INF/faces-config.xml', 'JSF Servlet Name' with the value 'Faces Servlet', 'JSF Servlet Class Name' with the value 'javax.faces.webapp.FacesServlet', and 'URL Mapping Patterns' with the value '\*.seam'. To the right of the 'URL Mapping Patterns' field are two buttons: 'Add...' and 'Remove'.

**Figure 2.21. JSF Capabilities Wizard**


Since we are arranging the Seam Web project, the last step we should do is to adjust project configurations associated with the Seam.

## 2.6. Configure Seam Facet Settings


The last wizard step is related to Seam facet and allows you to do the following:

## Seam Facet

Configure Seam Facet Settings




### General

Seam Runtime:  


Deploy as: ☒ WAR ☐ EAR


EJB project name:

EAR project name:

Libraries:  

### Database

Database Type:  

Connection profile:  

Database Schema Name:

Database Catalog Name:

DB Tables already exists in database: ☐

Recreate database tables and data on deploy: ☐

### Code Generation

Session Bean Package Name:

Entity Bean Package Name:

Create Test Project: ☐


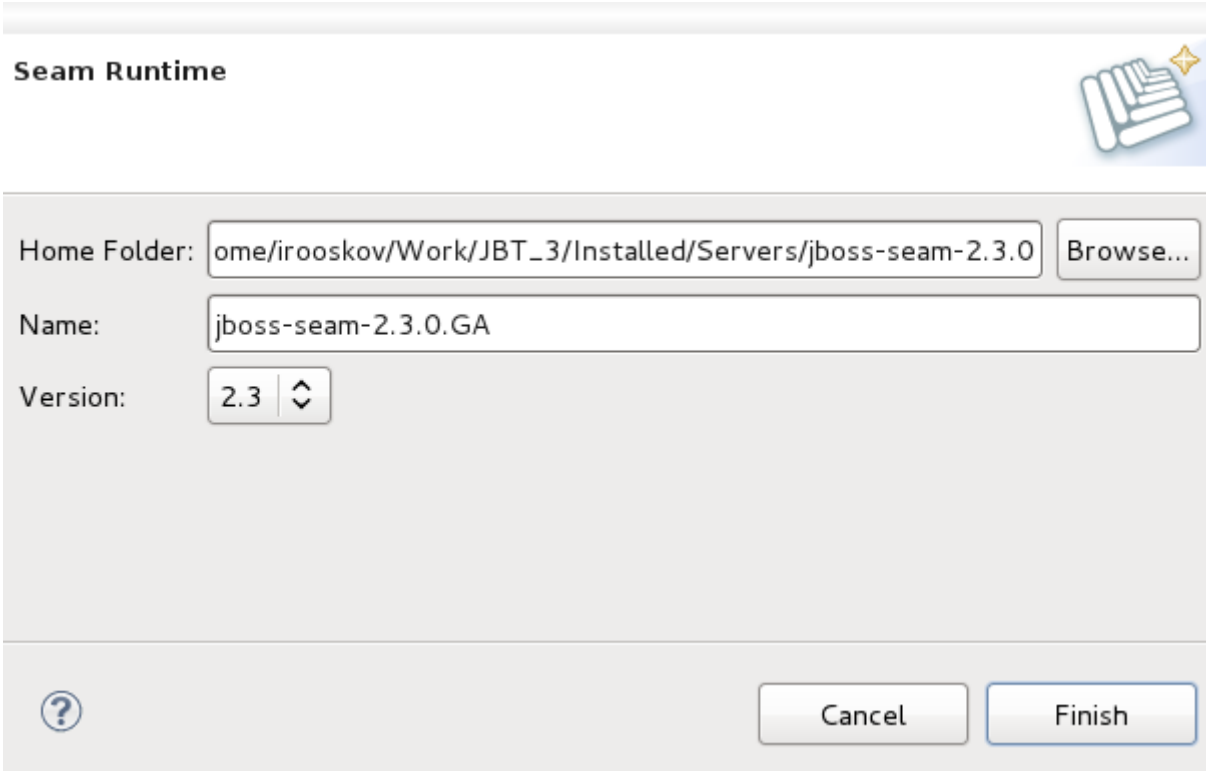


Figure 2.22. Seam Facet Settings

- Create Seam runtime and define Seam home folder.

For that click on **Add** button in the General section. Note, in the wizard shown below you can create a Seam runtime only for the version that was selected in the [Section 2.2, "Selecting the Project Facets"](#) wizard.

The image shows a 'Seam Runtime' configuration window. At the top, the title 'Seam Runtime' is on the left, and a stack of papers icon with a star is on the right. Below the title bar, there are three input fields: 'Home Folder:' with the text 'ome/irooskov/Work/JBT\_3/Installed/Servers/jboss-seam-2.3.0' and a 'Browse...' button; 'Name:' with the text 'jboss-seam-2.3.0.GA'; and 'Version:' with a dropdown menu showing '2.3' and up/down arrows. At the bottom left is a help icon (a question mark in a circle). At the bottom right are 'Cancel' and 'Finish' buttons.

Seam Runtime

Home Folder: ome/irooskov/Work/JBT\_3/Installed/Servers/jboss-seam-2.3.0 Browse...

Name: jboss-seam-2.3.0.GA

Version: 2.3

?

Cancel Finish

**Figure 2.23. Seam Runtime Creation**

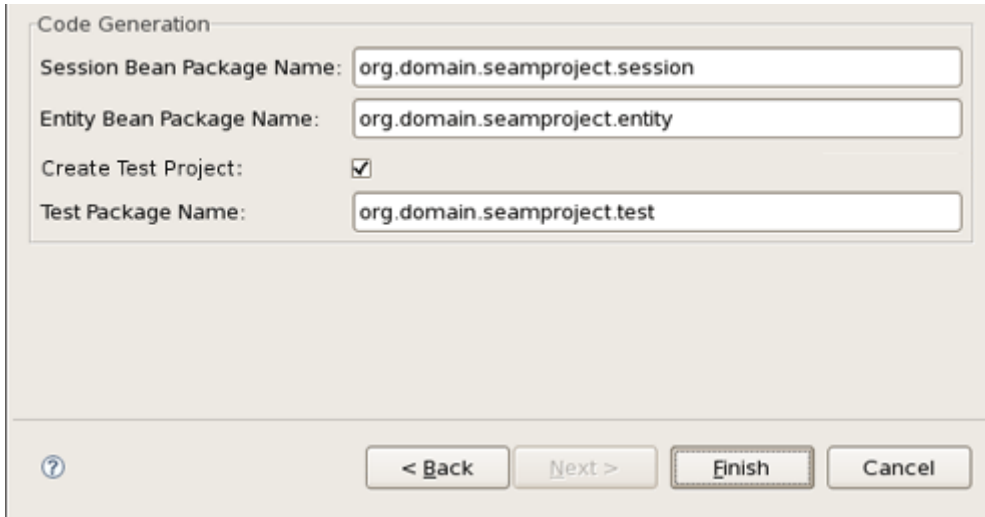
- Select EAR or WAR deployment by checking a necessary radio button.
- Enter EJB project name and EAR project name in case of EAR radio button is checked.



### Note

If you want to name your web project "MyProject-war" the EJB project should not be "MyProject-war-ejb", it should be "MyProject-ejb". The same for EAR and Test projects' names. They should be respectively "MyProject-ear" and "MyProject-test".

In the Code Generation section the wizard has already provided the names for your Session Bean, Entity Bean and Test packages generation (optional). Of course, you can change them on others of your choice.



The image shows a 'Code Generation' dialog box with a light beige background. It contains four input fields: 'Session Bean Package Name' with the value 'org.domain.seamproject.session', 'Entity Bean Package Name' with the value 'org.domain.seamproject.entity', 'Create Test Project' with a checked checkbox, and 'Test Package Name' with the value 'org.domain.seamproject.test'. At the bottom, there is a help icon (a question mark in a circle) on the left, and four buttons: '< Back', 'Next >', 'Finish' (which is highlighted with a dashed border), and 'Cancel'.


Code Generation

Session Bean Package Name:

Entity Bean Package Name:

Create Test Project: ☒

Test Package Name:



**Figure 2.24. Code Generation Section**

Click on **Finish** to generate a project.



# Maven Integration for Seam Projects

Maven integration is included by default, allowing you to create mavenized Seam projects and use Seam tools with imported Maven projects.

## 3.1. Creating Maven ready Seam project

Maven Integration makes the Seam Wizard capable of creating Maven ready projects to let Maven get the libraries instead of using the Seam runtime.

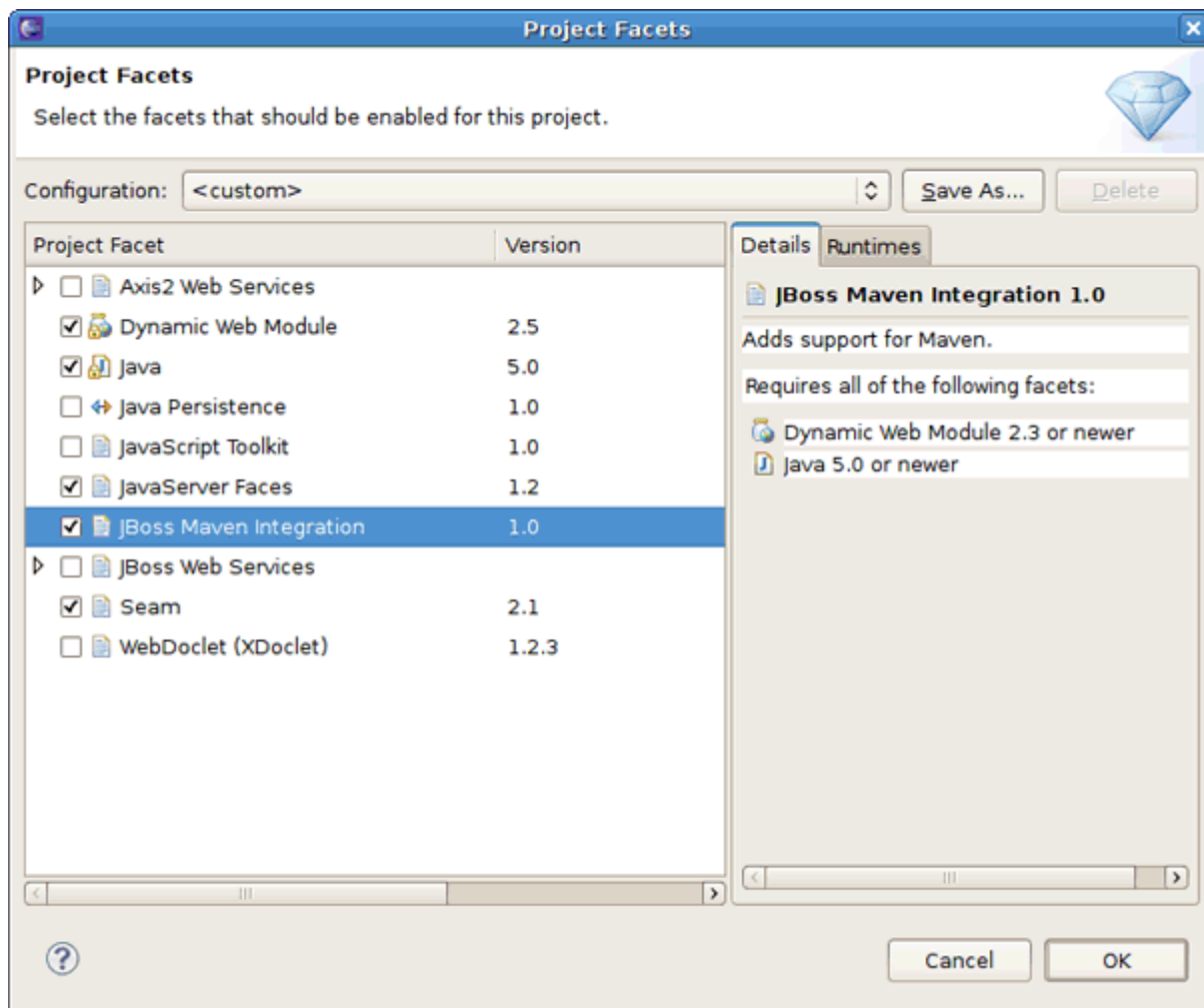
To create a mavenized Seam project you should perform the following steps:

- Navigate to **File** → **New** → **Other** → **Seam Web Project**. Give the project a meaningful name, specify a target runtime and server and select the proper configuration for your Seam project:



Figure 3.1. Starting the Mavenized Seam Project

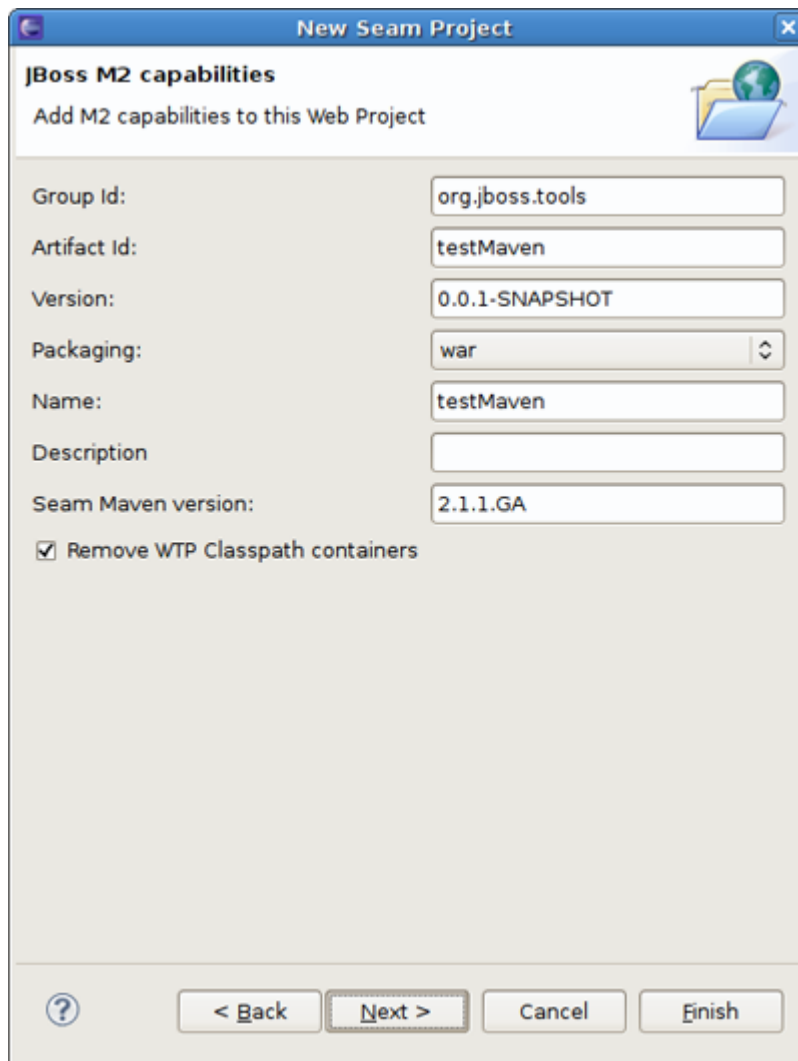
- Click the *Modify* button to enable the *Jboss Maven Integration* facet:



**Figure 3.2. Enabling the Jboss Maven Integration Facet**

- On the JBoss M2 capabilities page you'll be prompted to add the following maven properties: groupId, artifactId, version, packaging, name and description. Here it's possible to set a maven version of Seam and also decide whether to remove WTP classpath containers (where the path starts with org.eclipse.jst).





**New Seam Project**

**JBoss M2 capabilities**  
Add M2 capabilities to this Web Project

Group Id:

Artifact Id:

Version:


Packaging:

Name:

Description:

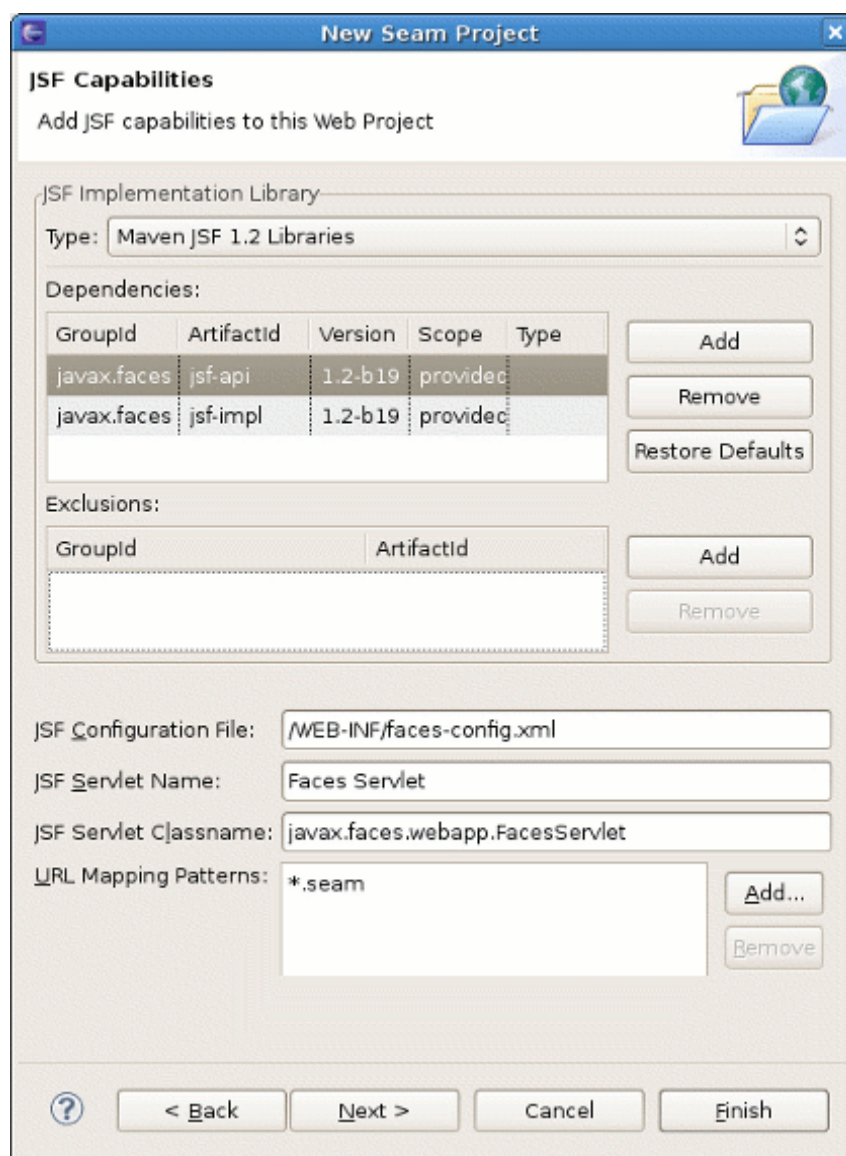
Seam Maven version:

☒ Remove WTP Classpath containers



**Figure 3.3. Adjusting JBoss M2 Capabilities**

- On the JSF Capabilities page you can choose Maven as libraries provider. This is possible when you use Seam version compatible Maven JSF Libraries as a type of JSF Implementing Library. Also it gives you capability to edit Dependencies and Exclusions for the mavenized Seam project directly from the wizard.



**Figure 3.4. Using JSF Implementation libraries provided by Maven**

- On the Seam Facet page configure the Seam Facet settings and click *Finish*:

**New Seam Project**

**Seam Facet**  
Configure Seam Facet Settings

**General**

Seam Runtime: jboss-seam-2.1.1.GA Add...

Deploy as: ☐ WAR ☒ EAR

**Database**

Database Type: HSQL

Connection profile: New HSQLDB Edit... New...

Database Schema Name:

Database Catalog Name:

DB Tables already exists in database: ☐

Recreate database tables and data on deploy: ☐

**Code Generation**

Session Bean Package Name: org.domain.testmaven.session

Entity Bean Package Name: org.domain.testmaven.entity

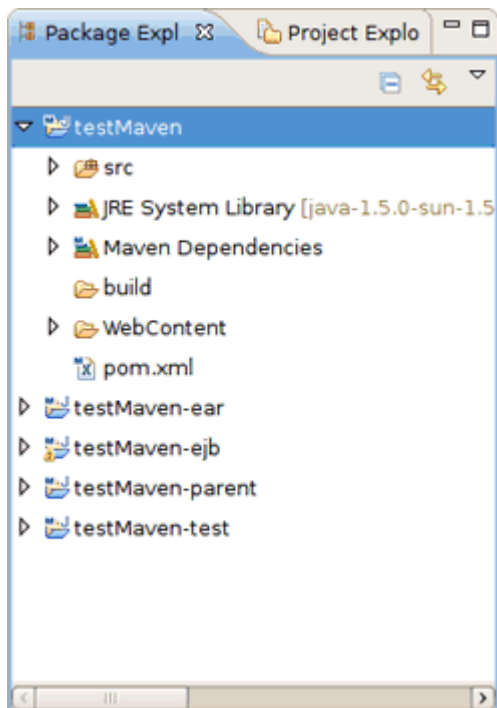
Create Test Project: ☒

Test Package Name: org.domain.testmaven.test

? < Back Next > Cancel Finish

**Figure 3.5. Seam Facet Settings**

The organized Seam project contains five projects with the Maven nature, builder, Maven classpath and the *pom.xml* files added: projectname, projectname-ear, projectname-ejb, projectname-parent and projectname-test.



**Figure 3.6. Mavenized Seam Project**

## 3.2. Existing Maven Projects Import

Maven Integration includes the Seam Maven configurator which is called when importing a project, changing *pom.xml* and/or calling **Maven** → **Update Project Configuration**. When importing a Maven Seam project, the Seam Maven configurator picks up the settings set up in *pom.xml* and adds to the project the following:

- Seam nature
- Seam facet and dependent facets to the EJB, EAR and WAR projects
- JBoss Maven Integration facet if the Maven project contains a dependency with the org.jboss.seam groupId and an artifactId starting with 'jboss-seam'
- other required facets (Web Module, Java, JSF, etc.)

If the Seam version from *pom.xml* matches a version of a Seam runtime configured in the workspace, Seam configurator will set that runtime, otherwise no Seam runtime will be set and you'll have to set it manually under [Section 10.2, "Project Preferences"](#).

Seam configurator recognizes the Seam artifacts in the following way: the view folder is a web content folder from the WAR project, the source folder is the first Eclipse source folder. If there is a folder containing "hot" in the name, it will be chosen as the action source folder. Package is the first package that has children or doesn't have any subpackage.

Configuring the Seam when importing a Maven project as well as a Seam Runtime, Seam artifacts and some facets could be suspended in the JBoss Maven Integration preferences (**Window** → **Preferences** → **JBoss Tools** → **JBoss Maven Integration**).

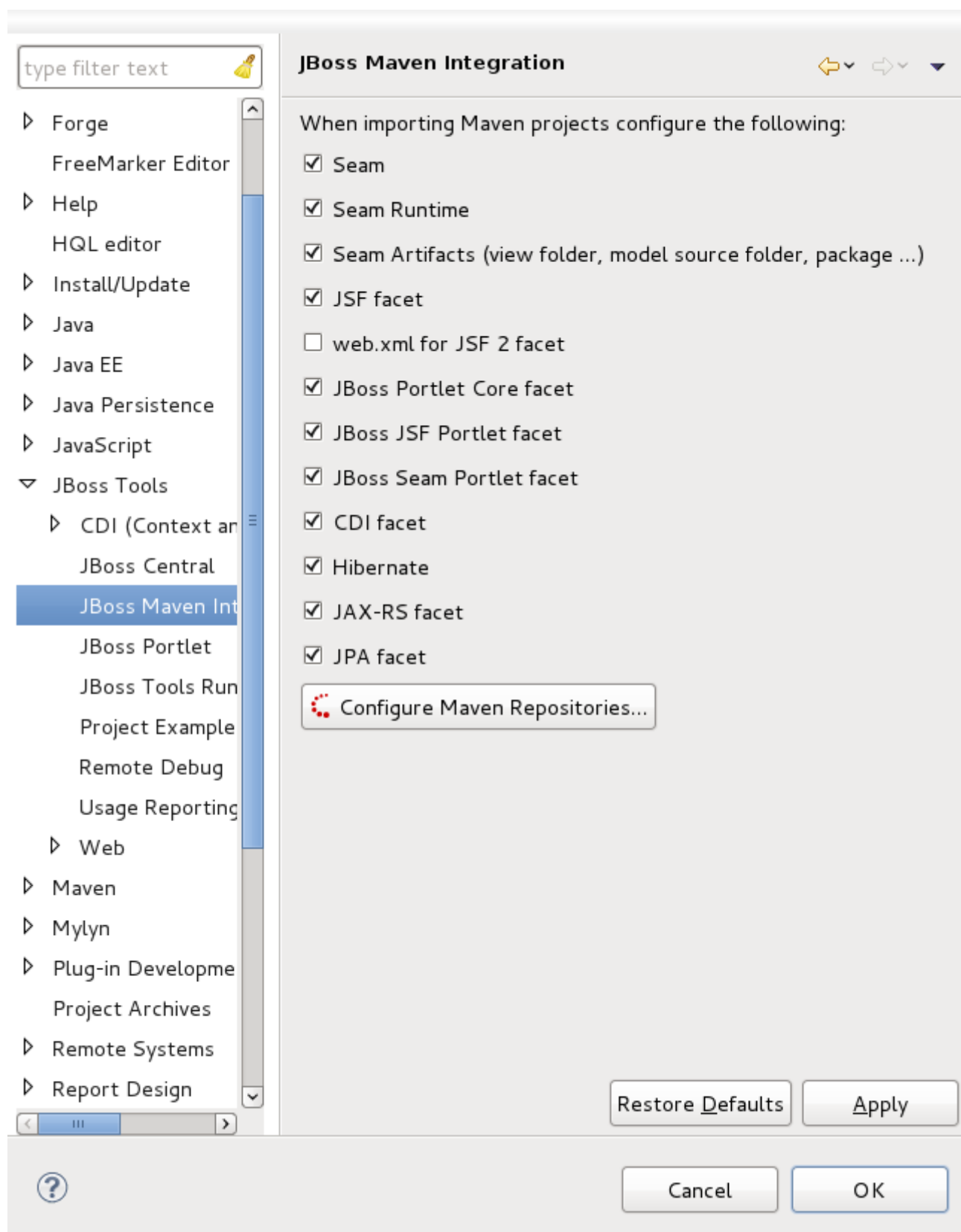


Figure 3.7. JBoss Maven Integration Preferences

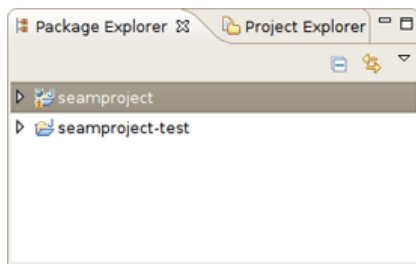
# Directory Structure of the Generated Project

In this chapter we describe where the Seam wizard puts the generated files for both EAR and WAR deployments.

The Seam Project wizard generates projects like Eclipse WTP style in order to utilize Eclipse WTP features and to have a correct classpath. To be more precise it generates one project per artifact.

## 4.1. WAR Deployment

The project layout for WAR projects is:



**Figure 4.1. Project Layout for WAR projects**

A WAR project can only use Seam JavaBean and JPA Entity bean components; it cannot use EJB3 Session beans etc.

WAR projects are generated to enable Seam war hotdeploy feature. Classes put into *src/action* will be deployed to *WEB-INF/dev* from which Seam automatically will perform hotdeploy of new components.



### Note:

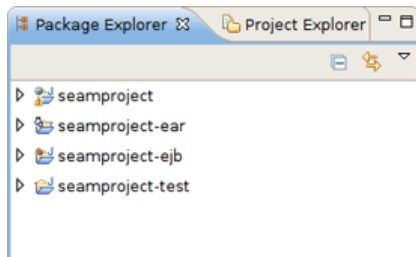
Because of Eclipse WTP limits the hot deployed classes also existed in *WEB-INF/classes*, but since Seam gives *WEB-INF/dev* precedence it will work.

Furthermore the Seam Project wizard generates a test project that is setup to run directly against the proper libraries and server runtime libraries. You can run tests via **Run As** → **TestNG Test**: [Chapter 15, What is TestNG?](#).

In order to deploy WAR project on server, right-click on the project and select **Run As** → **Run on Server**. Studio will deploy WAR project into one web application on server to deploy folder.

## 4.2. EAR Deployment

The project layout for EAR projects is:



**Figure 4.2. Project Layout for EAR projects**

An EAR project can use the whole range of Seam components, including EJB3 Session beans.

In order to deploy EAR project on server, right-click on the project with -ear postfix and select **Run As** → **Run on Server**. Studio will take care about all modules and deploy EAR project into one enterprise application on server to deploy folder. EAR application keeps ejb and war modules of the EAR project.



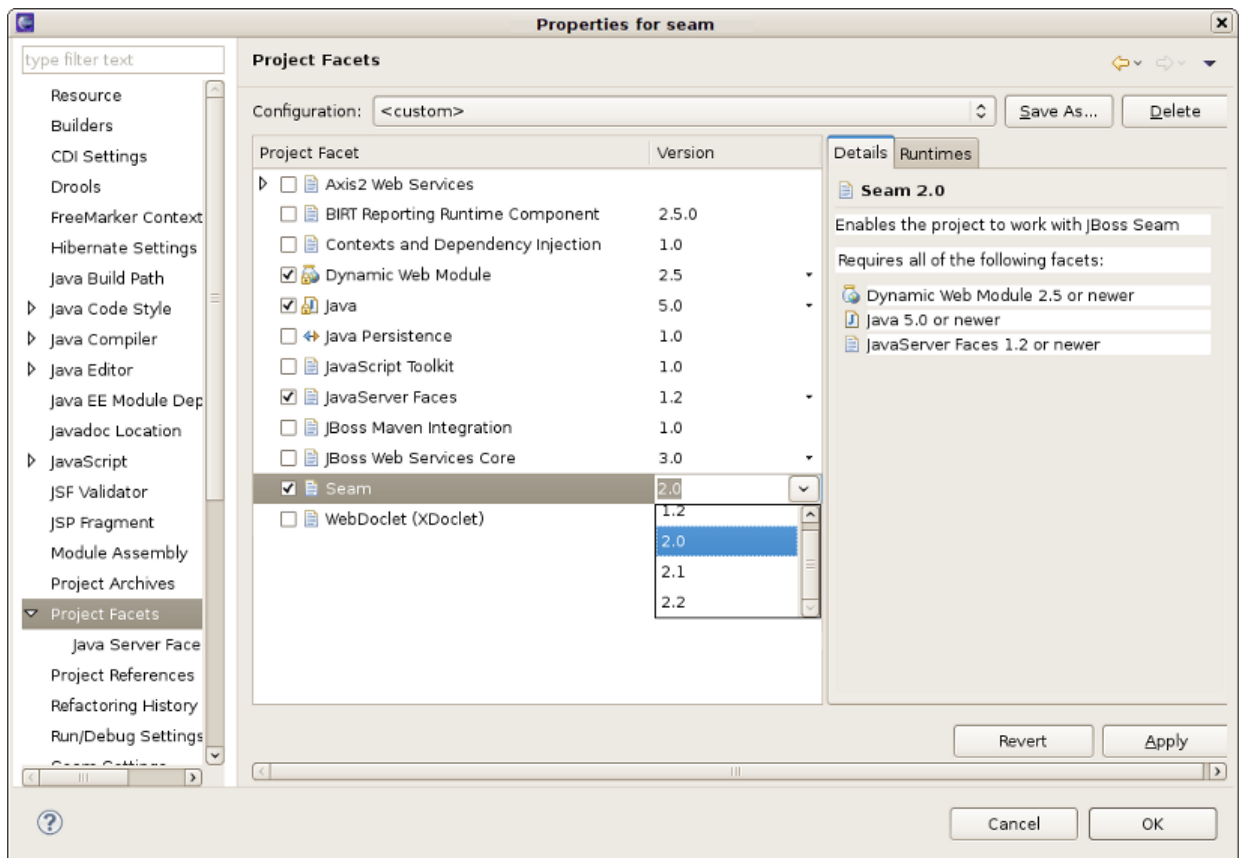
### Note:

The WAR generated for EARs is now configured to utilize Seam hot-deployment (*WEB-INF/dev*) similar to standalone WAR projects.

## 4.3. Changing the Seam Version

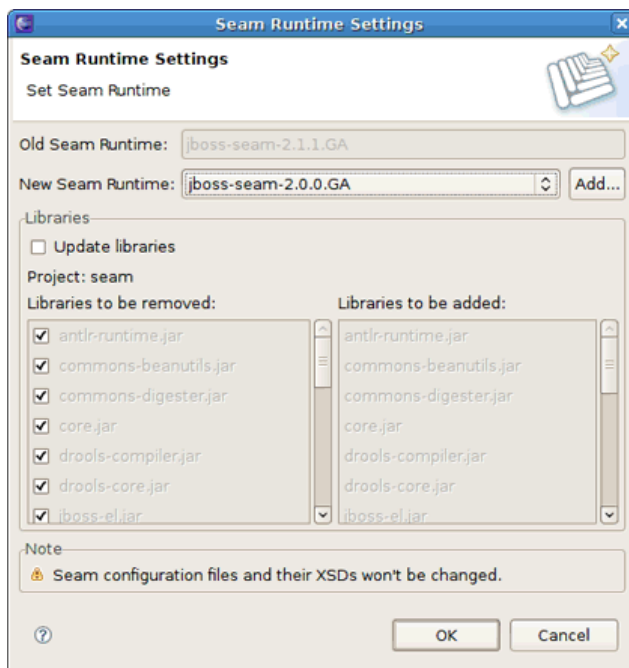
To upgrade or downgrade your projects Seam version use the facet preferences. You should right-click your project, open its Properties and choose *Project Facets* category. Next select *Seam* and change its version to needed one.





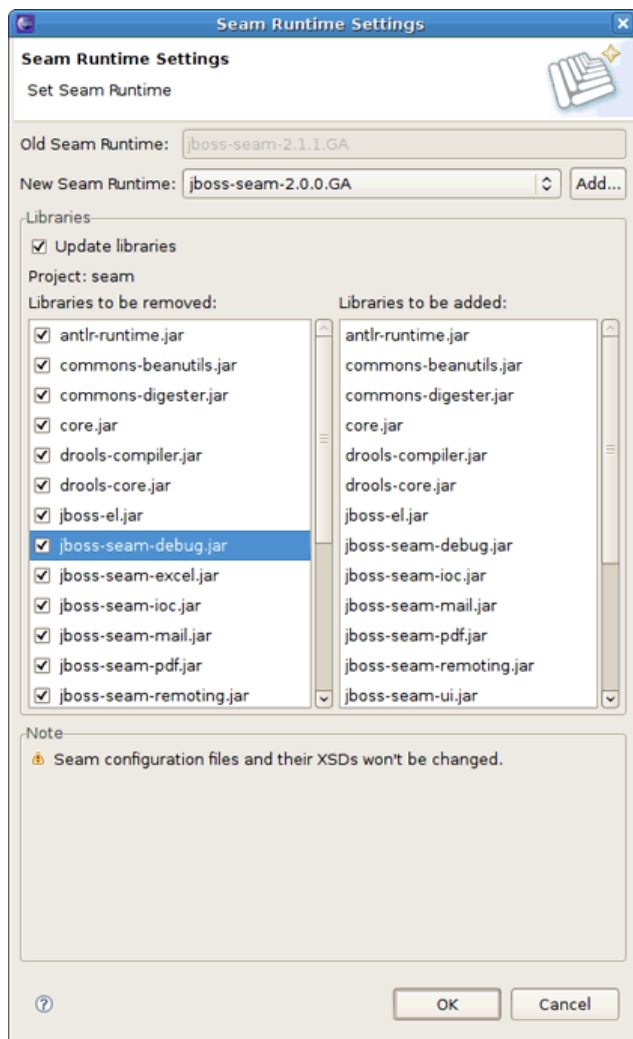
**Figure 4.3. Changing the Seam Facet Version**

After pressing *Apply* the wizard for adjusting new Seam runtime settings appears.



**Figure 4.4. Changing the Seam Facet Version**

If you need to update the libraries for you project, check the *Update libraries* option. All libraries you checked will be removed and the libraries from the new Seam distribution will be added after clicking *Ok*.



**Figure 4.5. Changing the Seam Facet Version**

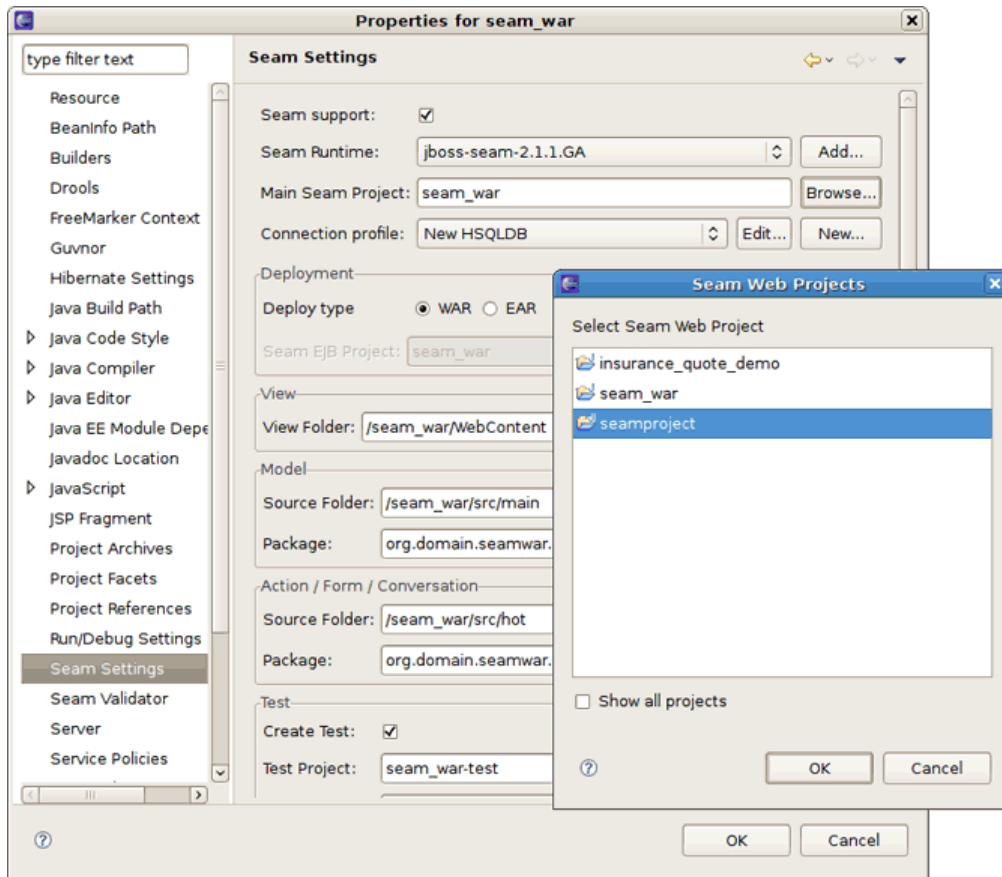


### Note:

The note "Seam configuration files and their XSDs won't be changed" is meant that the libraries, Seam facet and runtime version will be changed, but configuration files that refer to the old version will have to be manually updated.

## 4.4. Changing Seam Parent Project

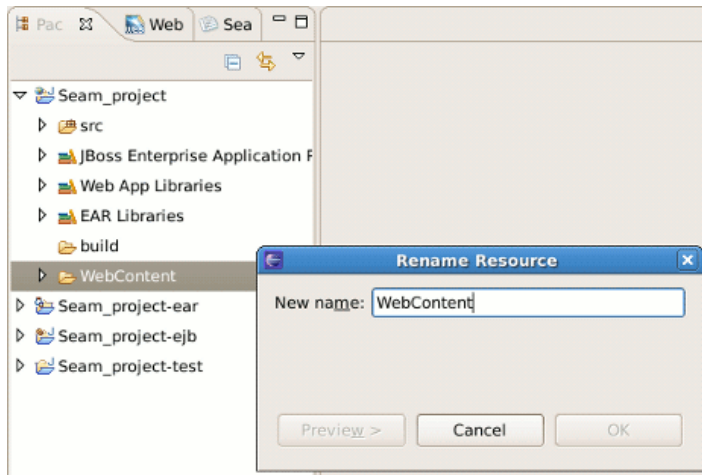
To change the Seam parent project, navigate to your project preferences and select *Seam Settings* category on the left. Press the *Browse* button next to the *Main Seam Project* section to select the other Seam parent project.



**Figure 4.6. Changing Seam Parent Project**

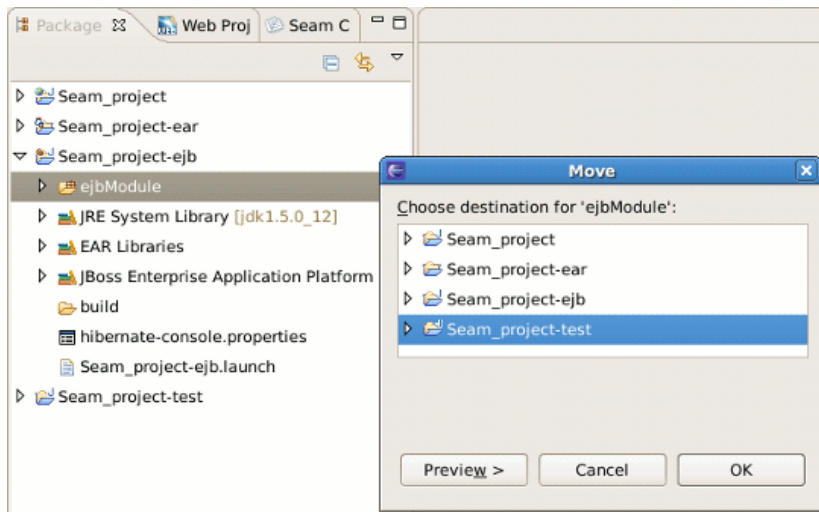
## 4.5. Renaming the Projects and Folders

If you need to rename one of the Seam Project artifacts (`<project_name>`, `<project_name>-ear`, `<project_name>-ejb` or `<project_name>-test`) or any entire folder like `<project_name>/WebContent`, `<project_name>/ejbModule`, `<project_name>-test/test-src`, or project name in packages `org.domain.<project_name>.session`, `org.domain.<project_name>.entity`, you can do this by accessing the context menu and navigating to *Refactor > Rename...* or just pressing *Shift + Alt + R* under the chosen resource.



**Figure 4.7. Renaming the folder**

Use *Refactor > Move...* (or *Shift + Alt + V*), if you need to move `<project_name>/WebContent` folder, `<project_name>/ejbModule` folder or `<project_name>/test-src` folder in the other place within the Project structure.



**Figure 4.8. Moving the folder**

# Seam Menus and Actions

In this chapter we provide a description of Seam actions that are available from

- Menu bar
- Toolbar
- Context menus in views

## 5.1. File Menu Actions

The following actions are in the Seam Perspective by default: **File** → **New** submenu

**Table 5.1. Seam Actions in the New Submenu**

Name	Function
Seam Web Project	Runs <a href="#">Section 2.1, “Create standalone Seam Web Project”</a> wizard for creating a new Seam project
Seam Action	Runs <a href="#">Section 6.1, “New Seam Action”</a> wizard for creating a new Seam action
Seam Form	Runs <a href="#">Section 6.2, “New Seam Form”</a> wizard for creating a new Seam form
Seam Entity	Runs <a href="#">Section 6.4, “New Seam Entity”</a> wizard for creating a new Seam entity
Seam Conversation	Runs <a href="#">Section 6.3, “New Seam Conversation”</a> wizard for creating a new Seam conversation
Seam Generate Entities	Runs <a href="#">Chapter 7, Seam Generate Entities</a> wizard

## 5.2. Navigate Menu Actions

In the next sections we are going to describe Seam actions which can help you to easily navigate through the source code.

### 5.2.1. Find Seam References/Declarations

To find EL expressions both in `.java` and `.xhtml` files use Find Seam References and Declarations actions. For that, in the main menu bar click on **Search** → **Find Seam References** or **Search** → **Find Seam Declarations**.

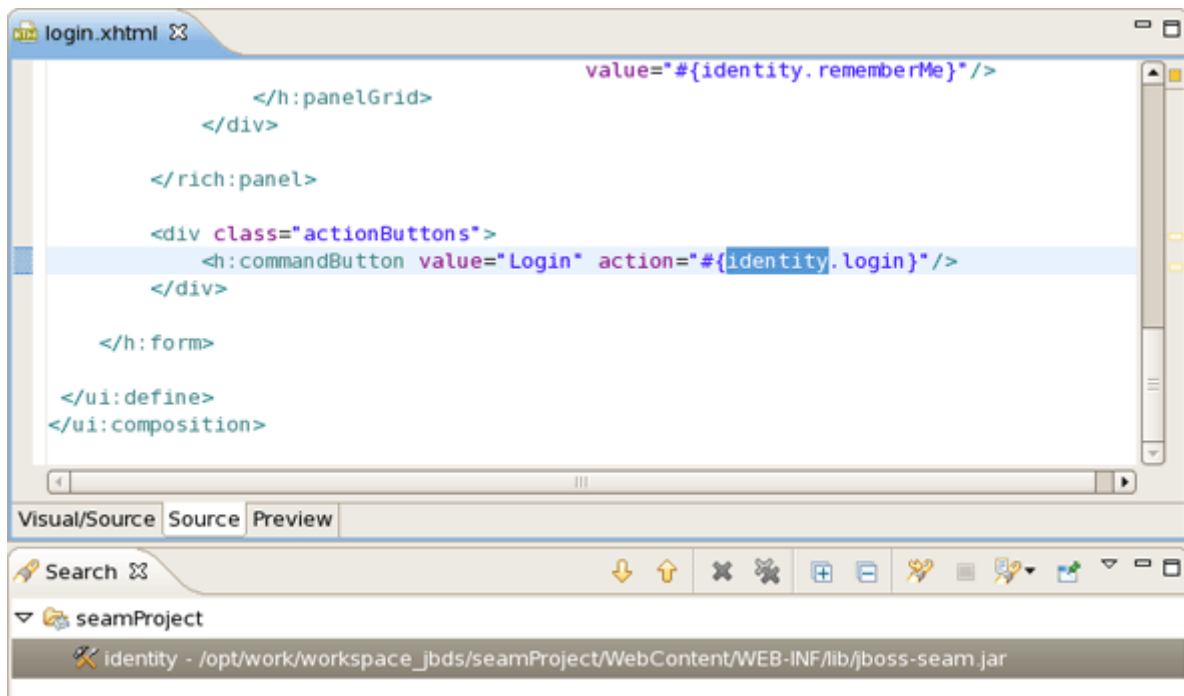
Look at the description of the actions in the table below.

**Table 5.2. Find Seam References/Declarations actions**

Name	Function	Keyboard Shortcut
Find Seam References	Find all references and declarations to the selected element. It's available for EL expressions in both <code>.java</code>	Ctrl+G

Name	Function	Keyboard Shortcut
Find Seam Declarations	and .xhtml files. Differs from normal Eclipse Find References/Declarations by showing an EL or Seam references in the Search View.	Ctrl+Shift+G also Ctrl + 1 for .java files

On the screenshot example below you could see that the search results are listed in the Search view.



**Figure 5.1. Find Seam Declaration for "Identity"**

You can also use Ctrl + 1 in .java files to activate the actions:

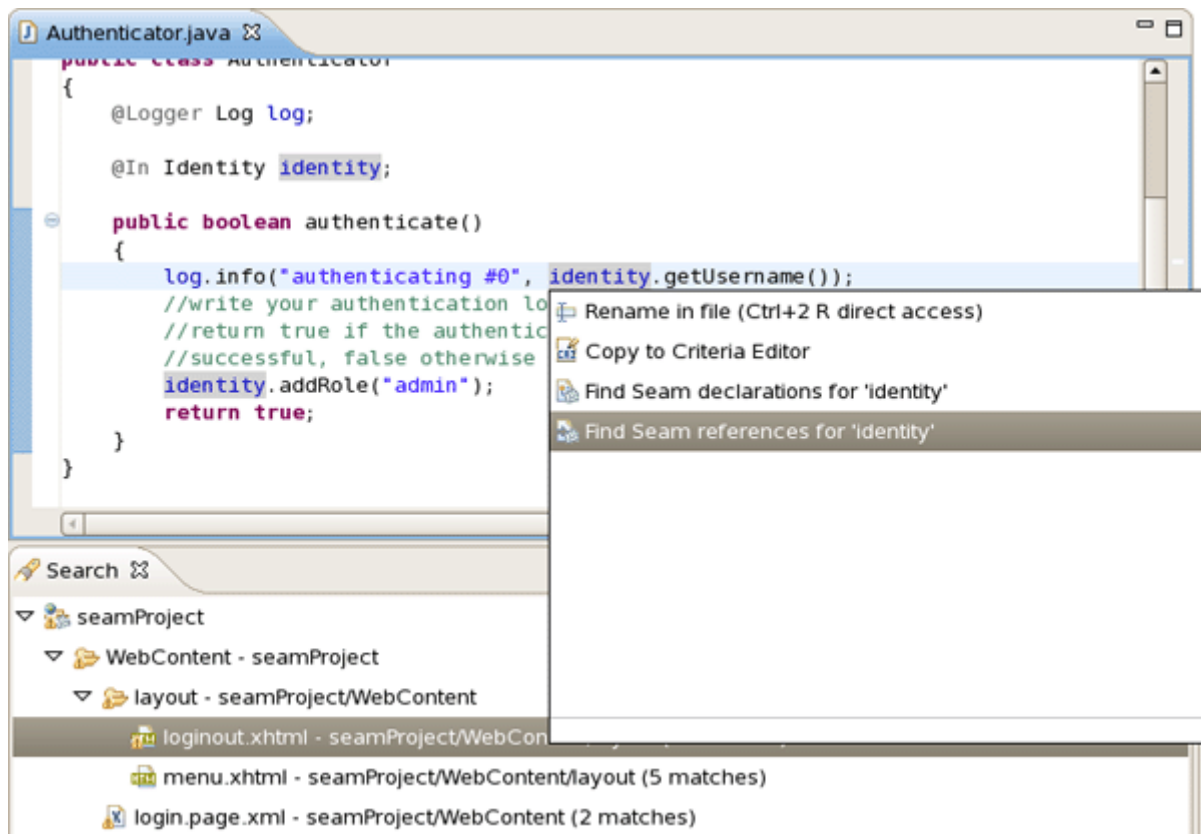


Figure 5.2. Find Seam References for "Identity"

## 5.2.2. Open Seam Component

To open Seam Components click on **Navigate** → **Open Seam Component** in the main menu bar. This dialog is also available from toolbar icon or with hot keys combination "Ctrl+Shift+Z".



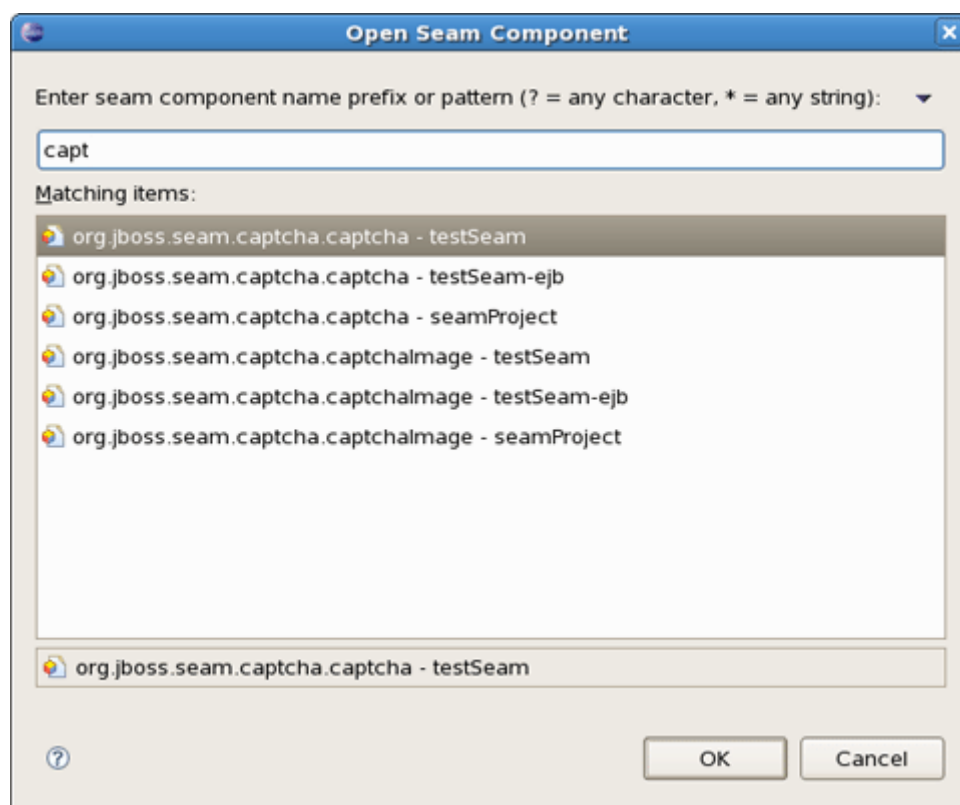
Figure 5.3. Open Seam Components icon

In the table below read a description about the dialog.

Table 5.3. Open Seam Components Dialog

Name	Function	Keyboard Shortcut
Open Seam Components	Brings up the Open Seam Component dialog to open a component in the editor. The Open Seam Component selection dialog shows all Seam components existing in the workspace. You can search the components via their short, imported or full name.	Ctrl+Shift+Z

Enter a name in the text field and see the results as it shown on screenshot:



**Figure 5.4. Open Seam Components Dialog**

You can also open Seam Component from the other component where the required one is declared in `@In` by using [Section 8.4.3, “OpenOn”](#) (Ctrl + left click and select Open Declaration of Seam Component for `seam_component` in `Seam_class.java` ).



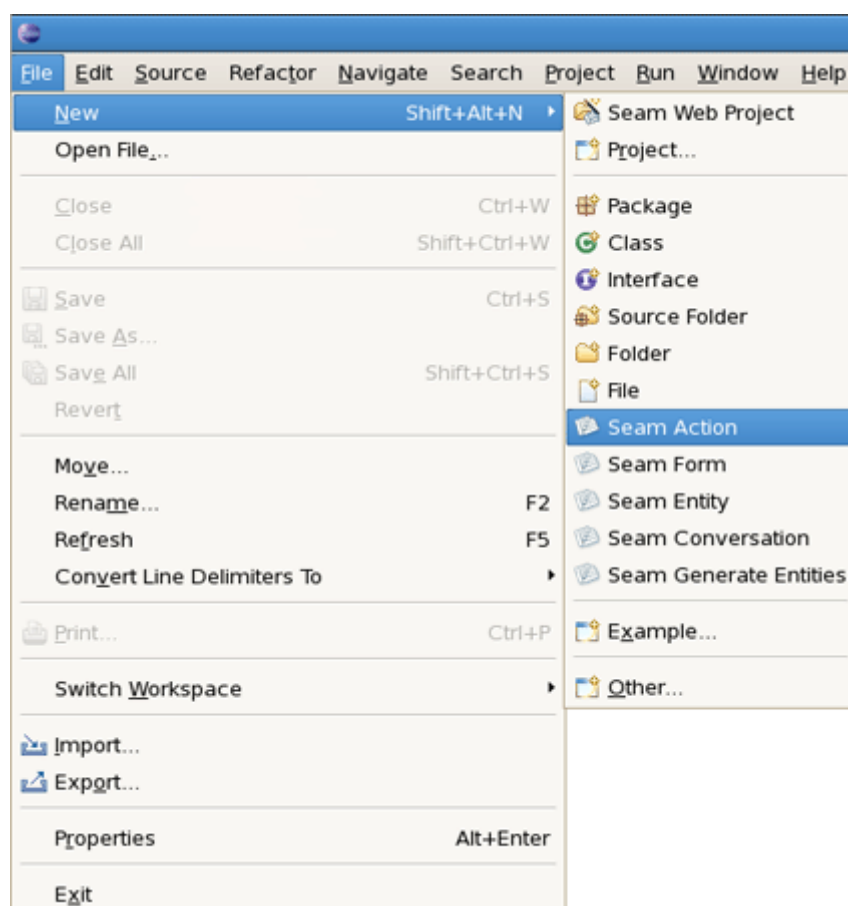
# Seam Wizards

This chapter introduces you with Seam Components.

All the Seam component generations options known from Seam-gen are available as wizards (with sensible auto-defaulting) for creating various common Seam components:

- Seam Action
- Seam Form
- Seam Entity
- Seam Conversation

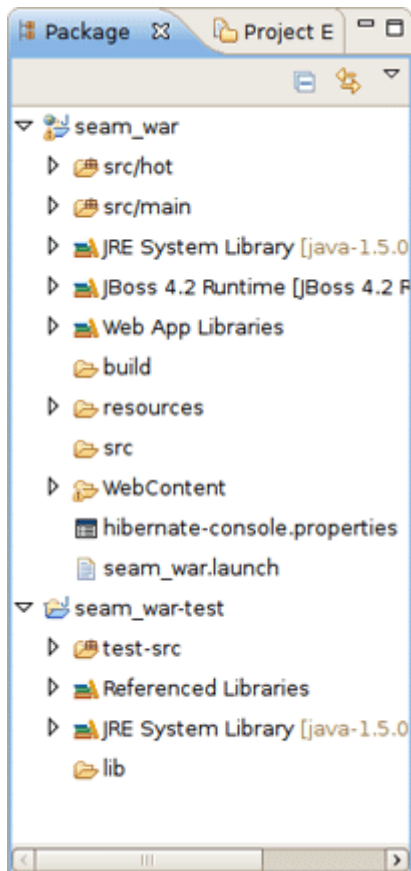
Go to **File** → **New** and select the component wizard.



**Figure 6.1. Seam Component Wizards**

The wizards create multiple resources and place it in the appropriate folders depending on your project structure (WAR or EAR).

Let's create a WAR project using the New Seam Project wizard.



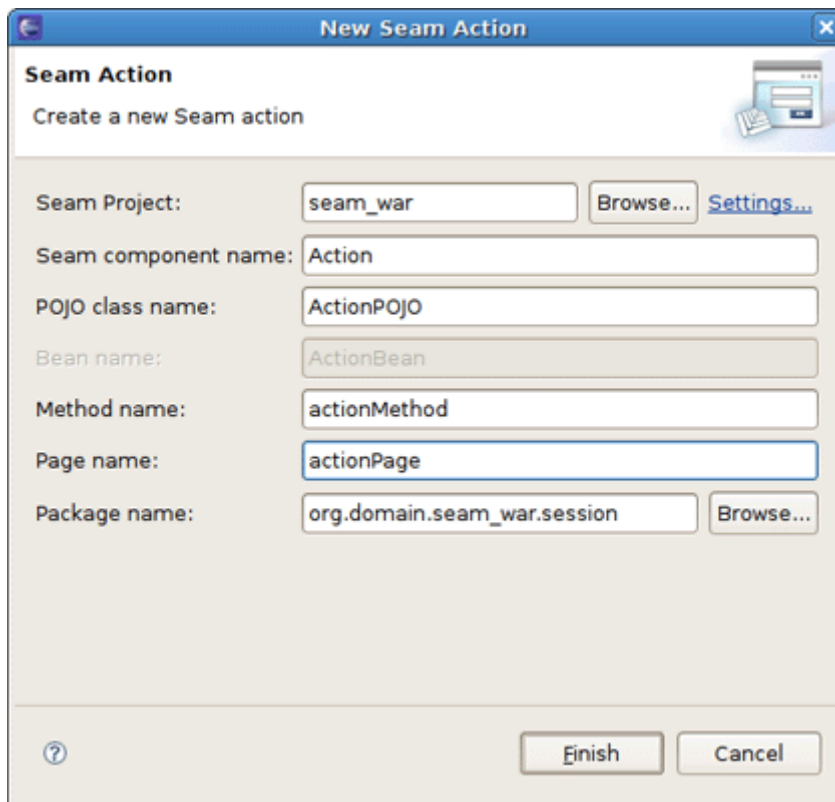
**Figure 6.2. Seam Project WAR Deployment**

After the project is created you need deploy it on server.

### 6.1. New Seam Action

In this and following sections you can see example creating Seam Components.

To create a New Seam Action you should select a necessary project, type a name for Seam component, POJO class, Method, Page and select a Package using the *Browse* button.



**Figure 6.3. New Seam Action Wizard**

If you specify a class which already exists, the wizard will warn you about it.

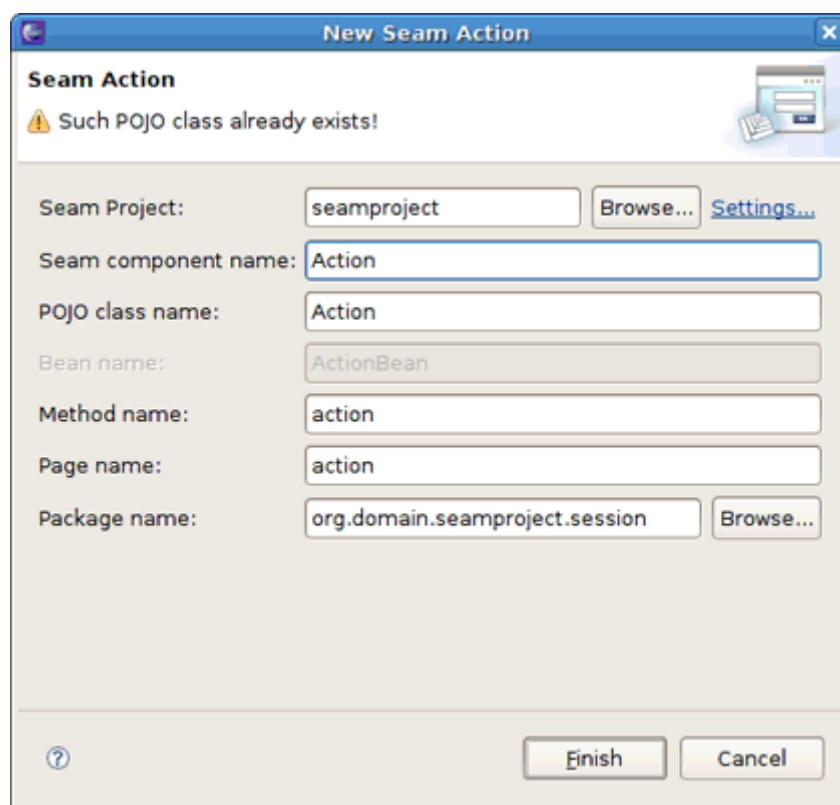


Figure 6.4. Wizard Warning against Existing File

You can see the action page in WebContent folder. Click on it to open in JBoss Tools HTML Editor.

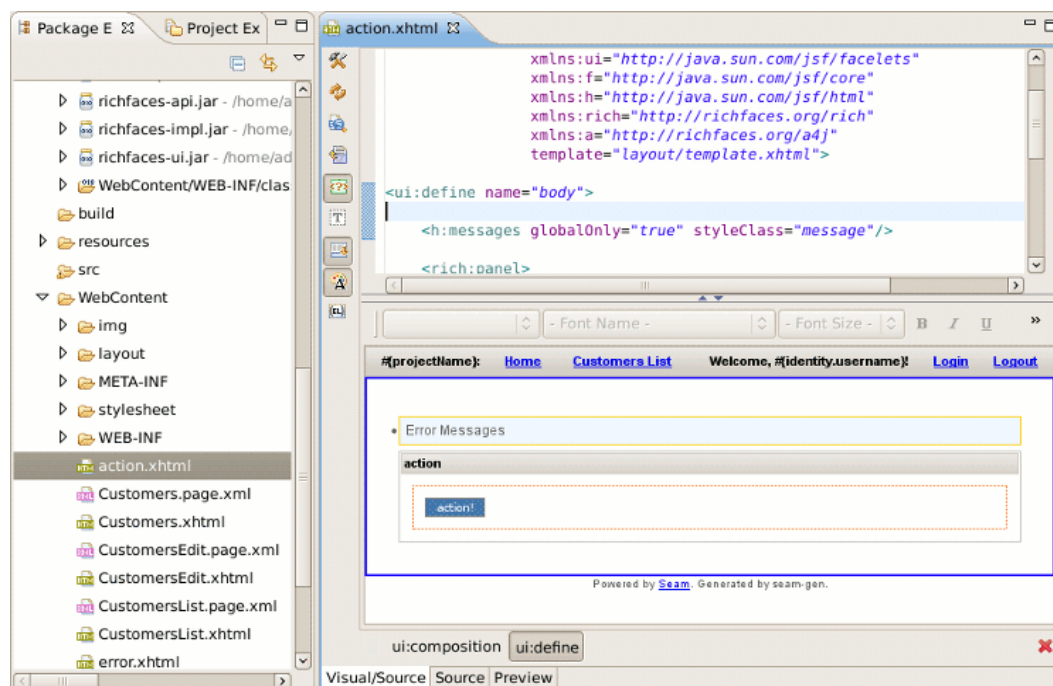
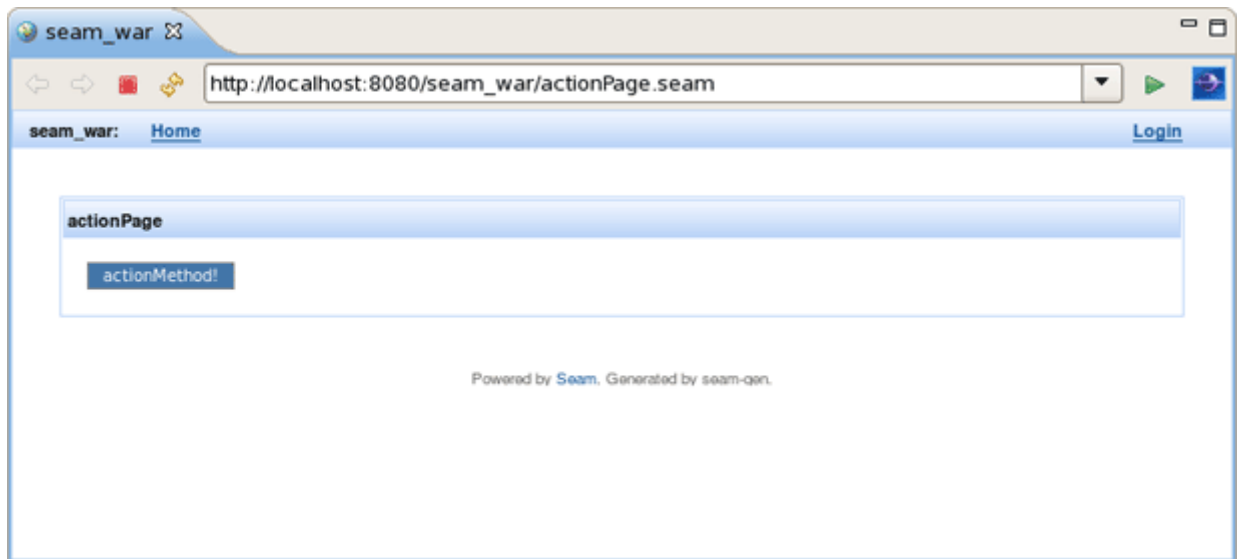


Figure 6.5. Action Page in JBoss Tools HTML Editor.

**Note:**

You don't need to restart the server to see how the action component works. Just use context menu **Run As** → **Run On Server**.

Action component was hot-deployed. Forms and Conversations will work the same way.

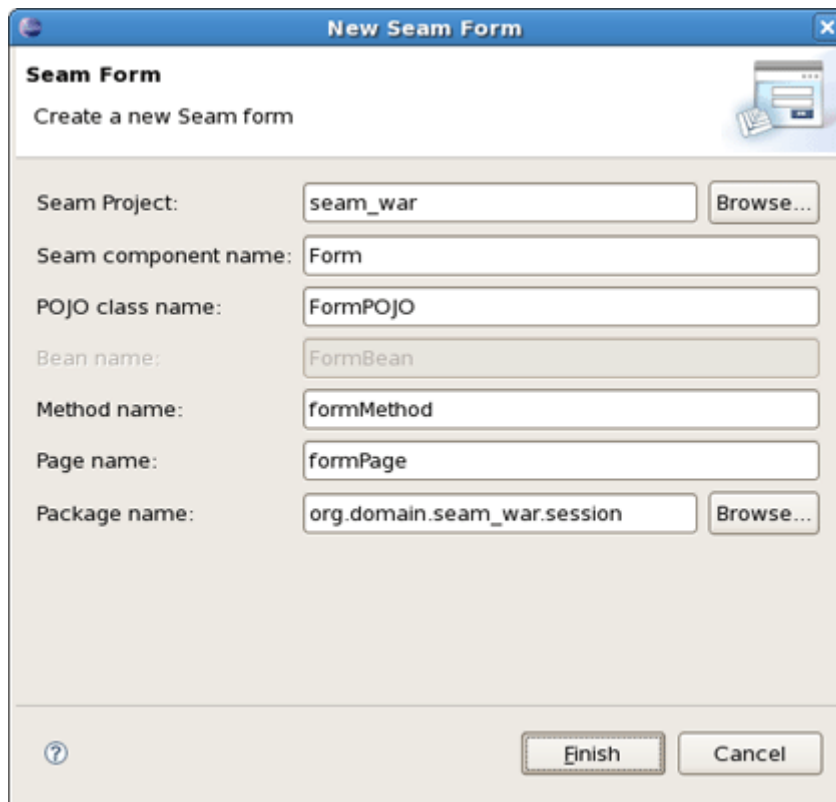


**Figure 6.6. Action Component**

Click on **actionMethod** in the internal browser and add a form in your project using the **New Seam Form** wizard **File** → **New** → **Seam Form**.

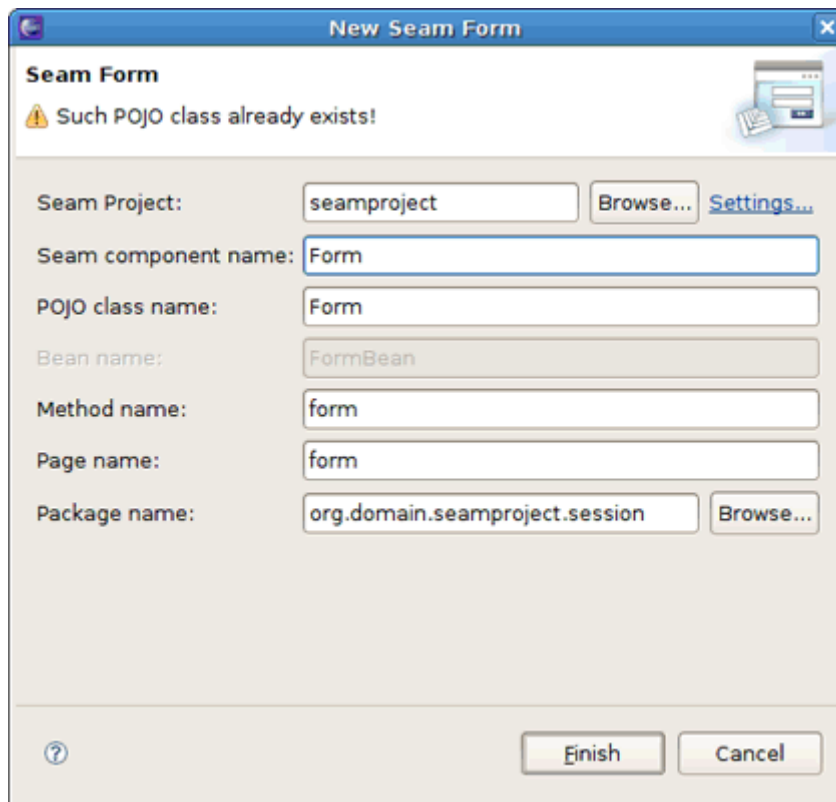
## 6.2. New Seam Form

Select a necessary project, type a name for Seam component, POJO class, Method, Page and select a Package using *Browse* button.



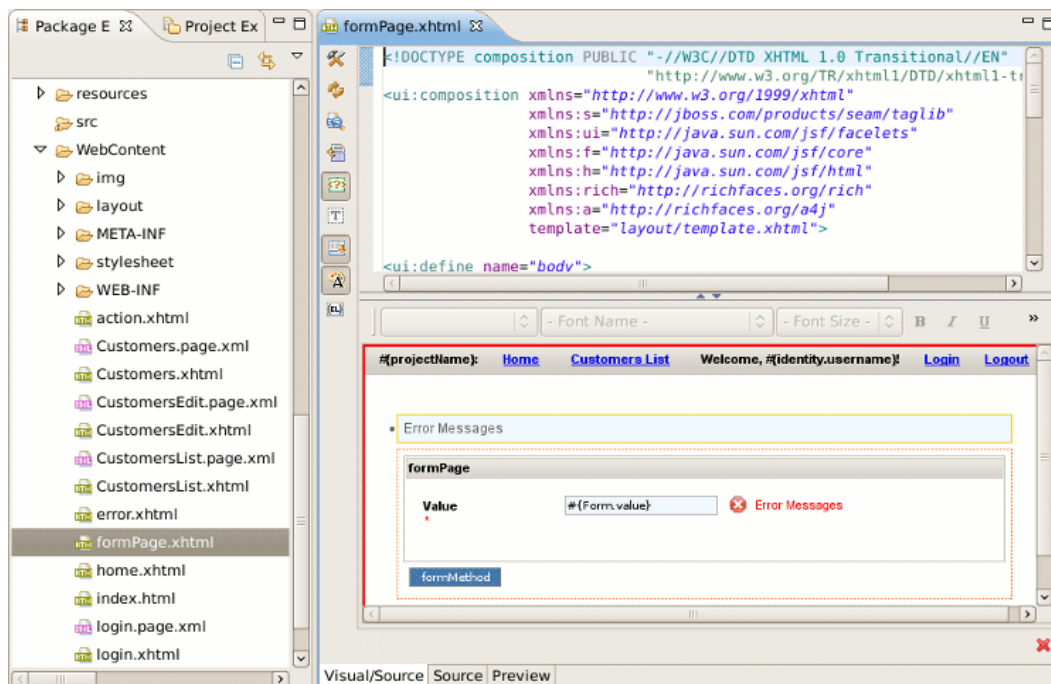
**Figure 6.7. New Seam Form Wizard**

If you specify a class which already exists, the wizard will warn you about it.



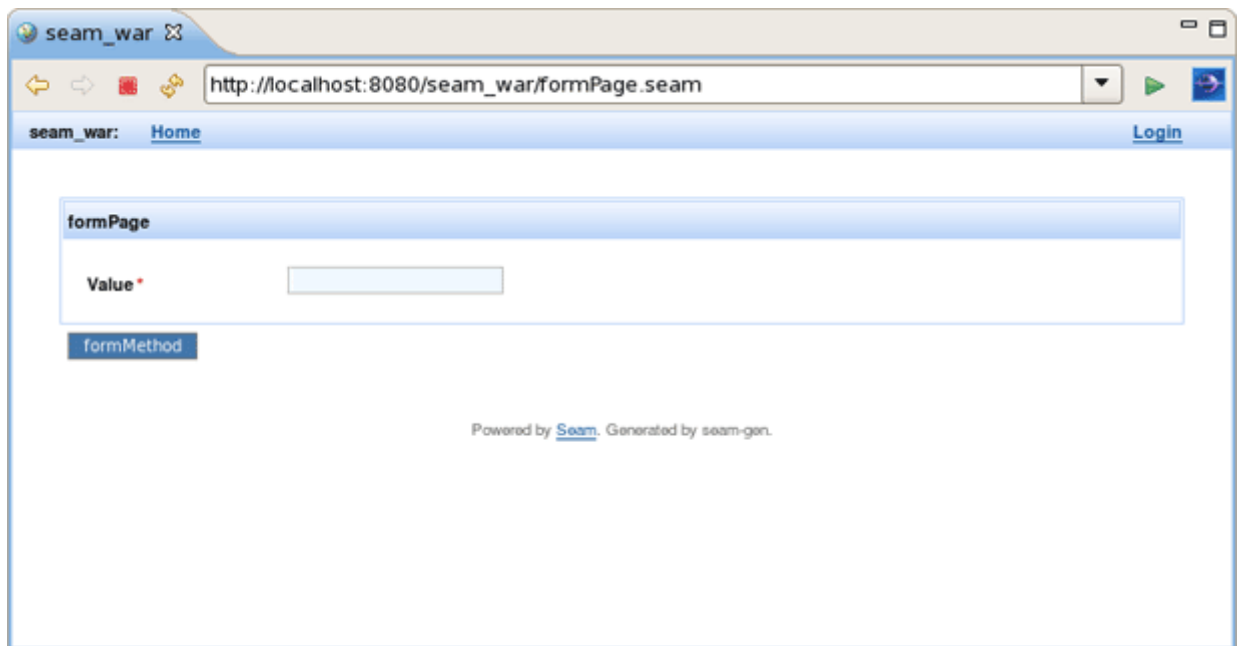
**Figure 6.8. Wizard Warning against Existing File**

The Form Page was created in WebContent folder.



**Figure 6.9. Form Page in JBoss Tools HTML Editor.**

Deploy the form on server. Right click on Form Page, select **Run As** → **Run On Server**.



**Figure 6.10. Form Component**

Form component was hot-deployed.

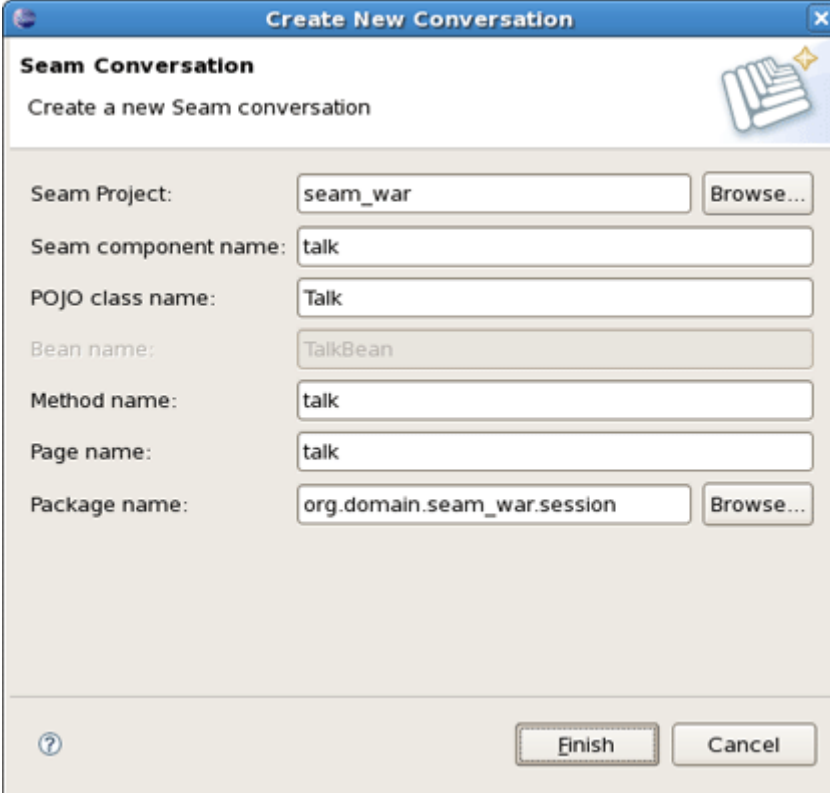
Enter some value in the text field (for example, value1) and click on **formMethod**.

### 6.3. New Seam Conversation

Add a conversation using the New Seam Conversation wizard **File** → **New** → **Seam Form**.

You should select a necessary project, type a name for Seam component, POJO class, Method, Page and select a Package using *Browse* button.





**Create New Conversation**

**Seam Conversation**  
Create a new Seam conversation

Seam Project:  

Seam component name:

POJO class name:

Bean name:

Method name:

Page name:

Package name:

**Figure 6.11. New Seam Conversation Wizard**

If you specify a class which already exists, the wizard will warn you about it.

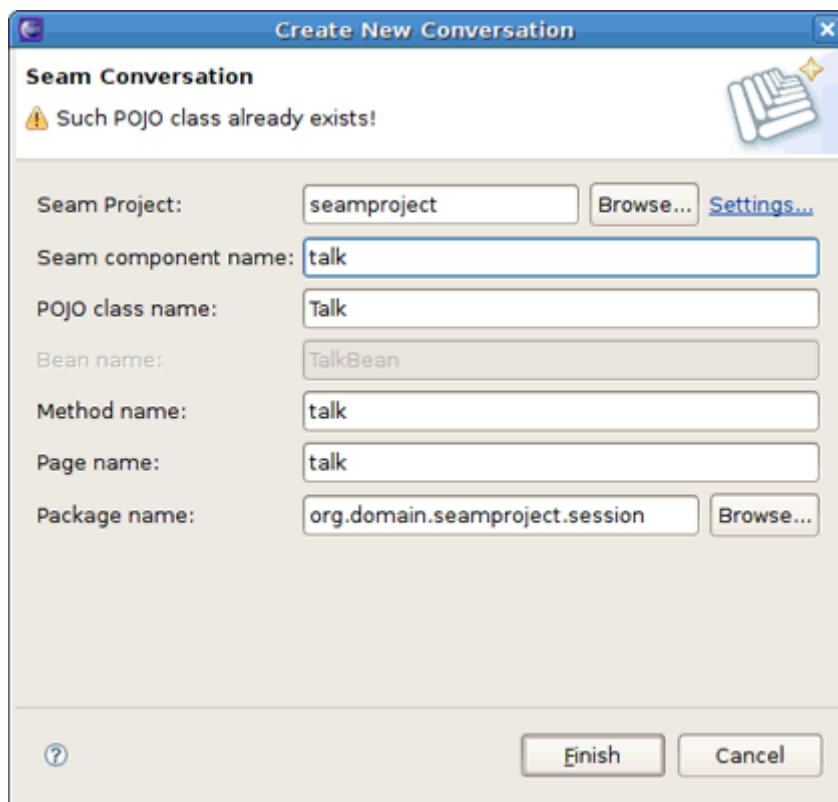


Figure 6.12. Wizard Warning against Existing File

Conversation page was created in WebContent folder.

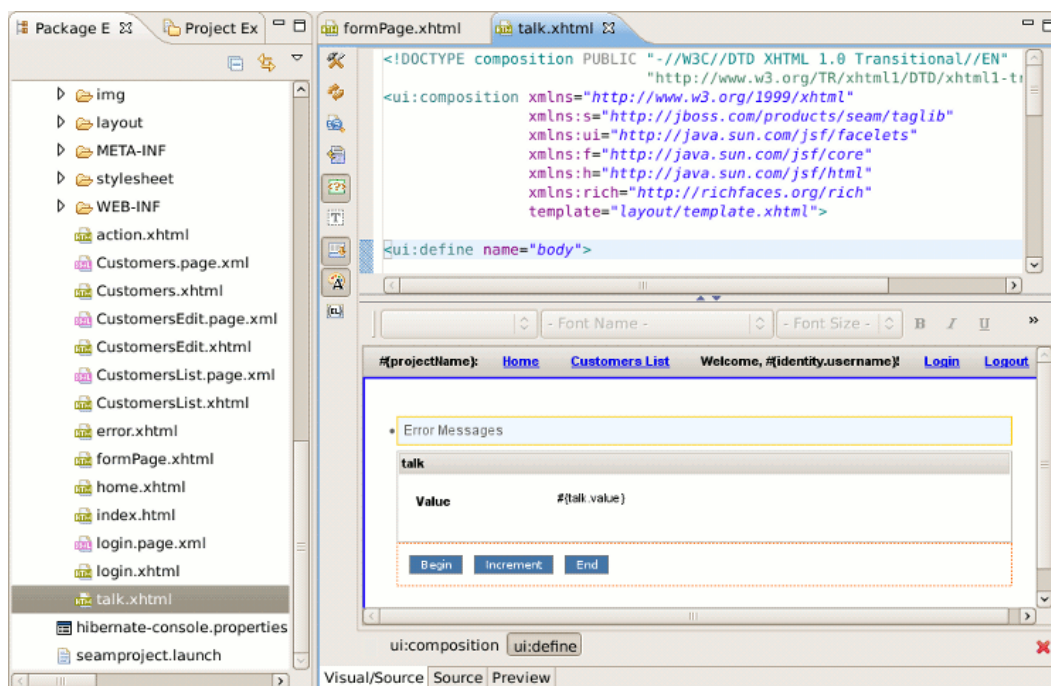
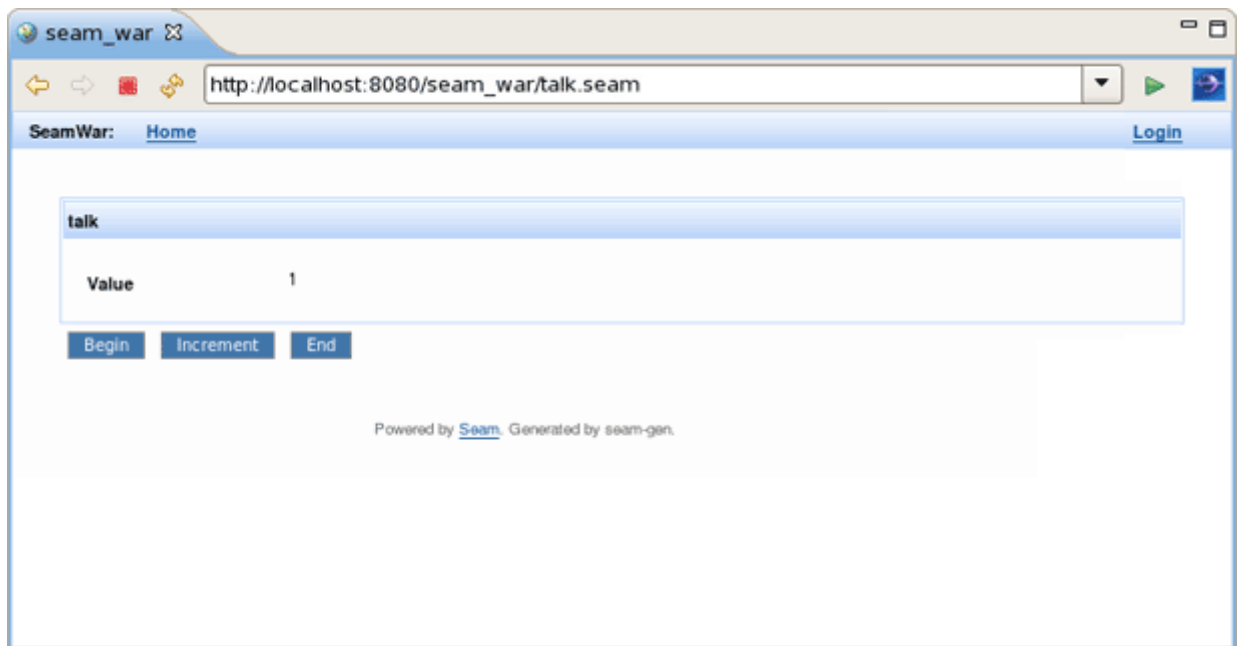


Figure 6.13. Conversation Page in JBoss Tools HTML Editor.

Right click on Conversation page, select **Run As** → **Run On Server**.



**Figure 6.14. Conversation Component**

Conversation component was hot-deployed.

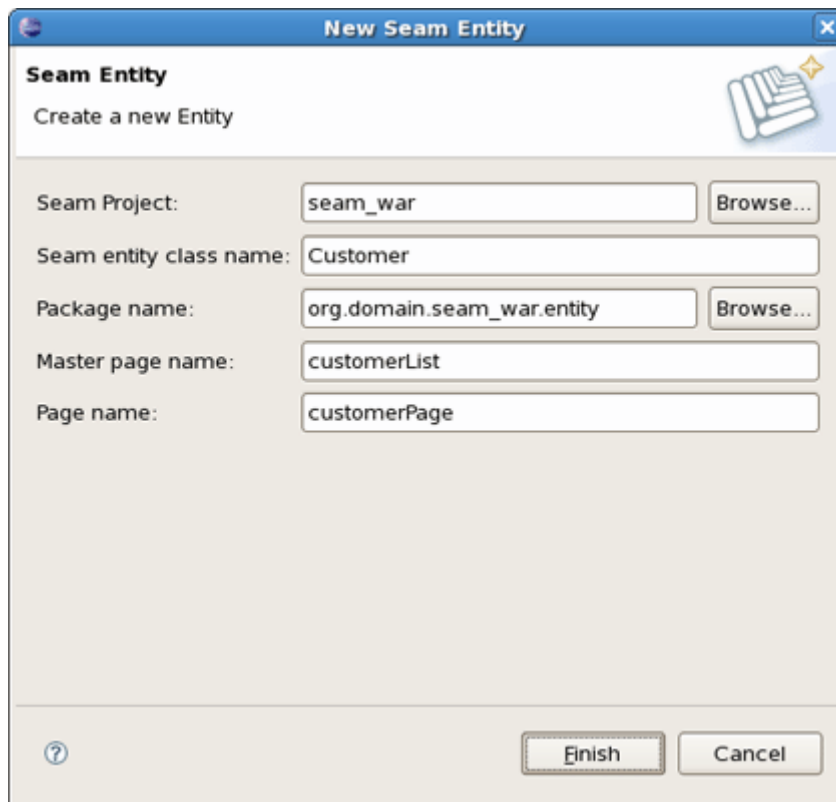
Click on Begin and Increment buttons to check the conversation functionality.

## 6.4. New Seam Entity

Entities cannot be hot-deployed, so we need to stop the server.

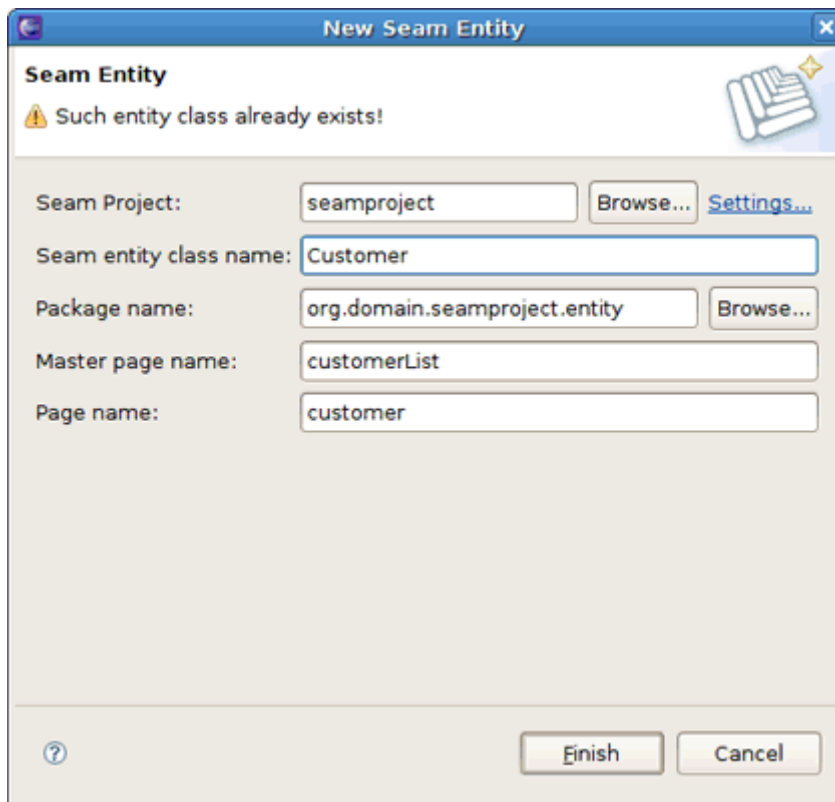
Create an Entity using the New Entity wizard **File** → **New** → **Seam Entity**.

You should select a necessary project, type a name for Entity class, select a Package using *Browse* button, type a name for Master Page and Page.



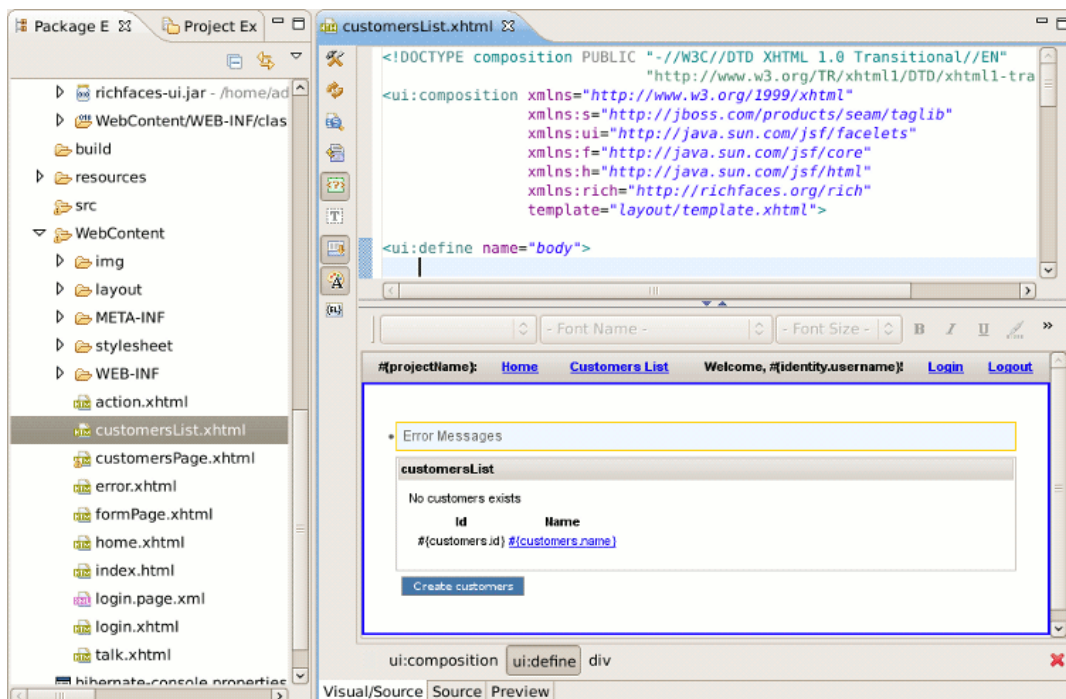
**Figure 6.15. New Seam Entity Wizard**

If you specify a class which already exists, the wizard will warn you about it.



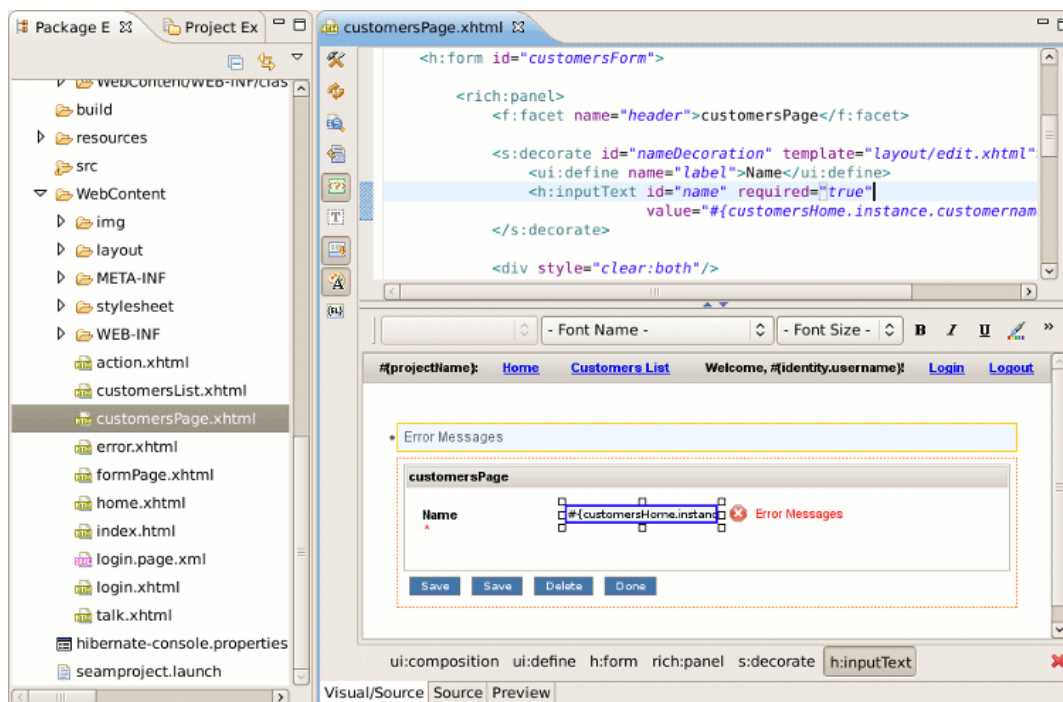
**Figure 6.16. Wizard Warning against Existing File**

The Master Page and the Entity were created in WebContent folder.



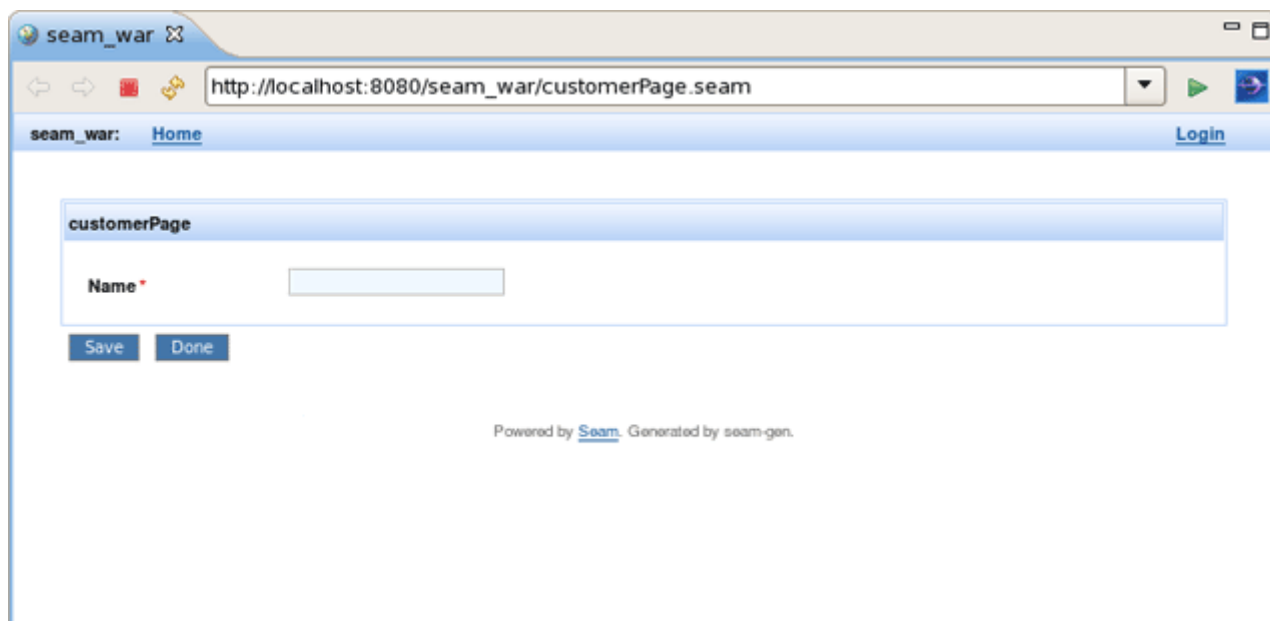
**Figure 6.17. Master Page in JBoss Tools HTML Editor.**

The Entity page is:



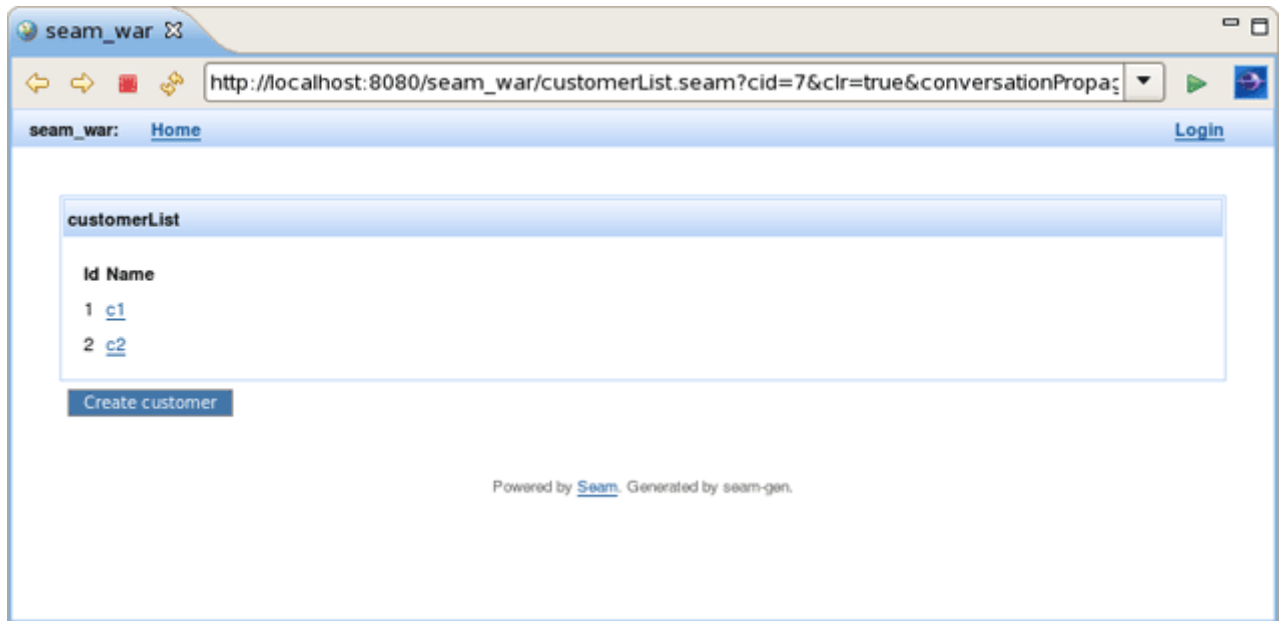
**Figure 6.18. Entity Page in JBoss Tools HTML Editor.**

Run the Entity page on server. This is what you get:



**Figure 6.19. Customer Page**

Let's create two customers c1 and c2. Enter the name in the text field and press the Save button. Customer should be successfully created. Press Done. Do the same for c2 customer. The result should be:



**Figure 6.20. Two Customers Are Created**

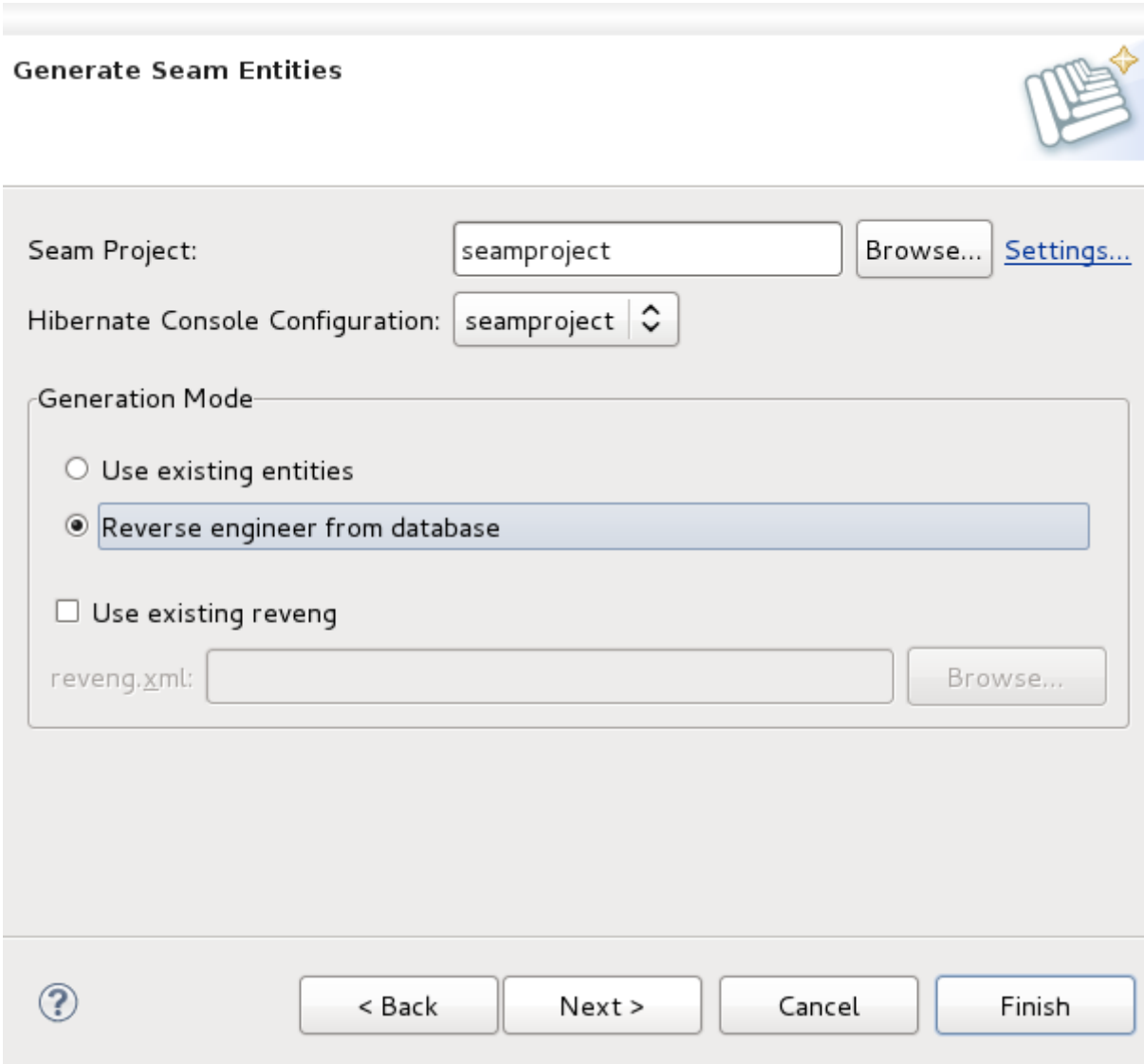




## Seam Generate Entities

Generate Entities is available by navigating to **File** → **New** → **Other** → **Seam** → **Seam Generate Entities**

The wizard generates a set of CRUD Seam components and web pages based on existing tables in a database or on existing entities in your application.



**Generate Seam Entities**

Seam Project:   [Settings...](#)

Hibernate Console Configuration:

**Generation Mode**

☐ Use existing entities

☒ Reverse engineer from database

☐ Use existing reveng

reveng.xml:

**Figure 7.1. Generate Seam Entities Wizard**

By default, the generation mode is set to *Reverse engineer from database*. The reverse engineering from a database could be described in four steps:

1. The wizard gets in database, extracts the tables and their connections

2. On basis of this metainformation the Entity classes are generated into `org.domain.project.entity` package
3. For the entities from step 2 the classes `EntityList` and `EntityHome` are generated into `org.domain.project.session` package
4. The xhtml pages are generated.



### Note:

The entities are generated from a database based on the console configuration that is created automatically during a Seam project creation. (Read more about the console configuration in the Hibernate guides). The created console configuration is based on the connection profile set on the [Section 2.6, “Configure Seam Facet Settings”](#) while creating a project. Configured connection profile settings are hold in the generated `hibernate-console.properties`. To change the connection profile, double-click the console configuration. It could be found in the Hibernate Configurations view (**Window** → **Show view** → **Other** → **Hibernate** → **Hibernate Configurations** or just switch to the Hibernate perspective).

Checking the *Use existing entities* mode the wizard executes only 3 and 4 steps. It generates missing classes and xhtml pages.

Read the [Chapter 13, Generate a CRUD Database Application](#) chapter in order to see how the Generate Seam Entities wizard can be used.

# Seam Editors

This chapter tells about Seam Editors and their features.

## 8.1. Visual Page Editor

Visual Page Editor is a great tool to use for the authoring of Seam pages. The major features of VPE are listed in [Section 8.4, “Main Features of Seam Editors”](#)

You can also read more about Visual Page Editor in the Visual Web Tools Reference Guide.

## 8.2. Seam Pages Editor

Seam Pages Editor provides a handy way to edit the pages.xml file.

You can edit the pages.xml file in three modes: Graphical, Tree and Source.

Seam Pages Editor supports synchronization with Package Explorer. This means that while renaming files in the Package Explorer all the changes immediately affect the pages.xml file.

### 8.2.1. Graphical Mode

Graphical mode provides you with a set of visual tools to organize your project pageflow, exception handling etc.

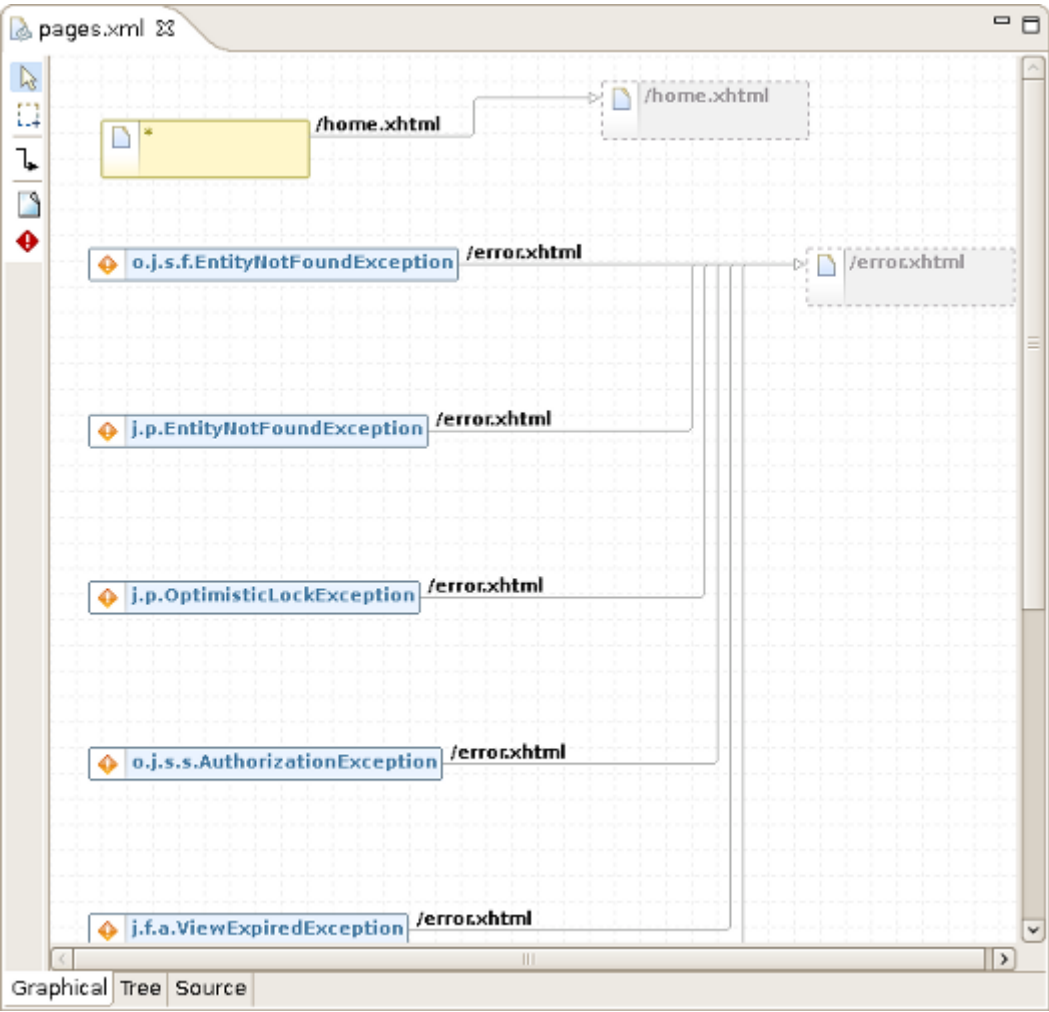

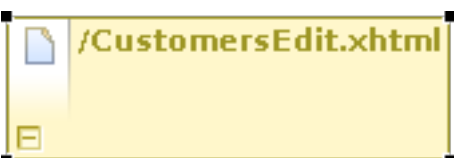

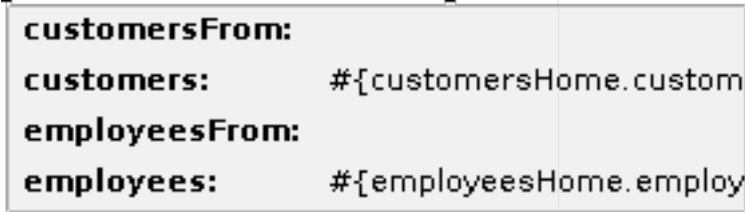





Figure 8.1. Seam Pages Editor: Graphical View

The Graphical part of the editor has some visual elements. The table below shows graphical representation of the elements and explains their meanings.

Table 8.1. Pages Editor: Graphical View. Visual elements





Element	Description
 <b>PageName.xhtml</b>	A yellow box with a solid border represents a <code>&lt;page&gt;</code> element.

Element	Description
	Pressing on the plus icon (  ) on the <code>&lt;page&gt;</code> element reveals a box that lists the parameters for the page.
	
	A red cross in the upper left corner of the page box indicates that the view-id is not found in the project.
	A gray box with a dashed border represents a page that has navigation (navigation rule) to but the page is not defined in the page.xml file.
	An <code>&lt;exception&gt;</code> is represented by a blue box with an orange diamond.

Pageflow relations are shown with gray arrows, when you select a relationship the arrow is changed to orange.

On the left hand side of the Graphical view of Seam Page Editor you can find a toolbar with a set of icons for the most frequently used commands.

**Table 8.2. Pages Editor: Graphical View. Commands Icons**

Icon Image	Command
	Select a page element
	Marquee a page item
	Add a page relationship
	Add an exception rule

Context menus are also available when you click either on an item or a blank space.

A context menu called on a blank space of the diagram provides the following options:

- **Page** creates a new page element on the diagram
- **Exception** creates a new exception element on the diagram
- **Auto Layout** formats the layout of the diagram automatically

- **Select Element** calls the Select Element Wizard with a filter to perform a quick search through the pages.xml file

Right clicking on a selected `<page>` element calls a context menu where you can choose the following options if you click on the **New** menu item:

- **Rule** opens the Add Navigation Rule dialog in which you can browse your project for a view page to set the view-id in the navigation rule and specify **render** or **redirect** navigation options
- **Param** calls a dialog box where you can define a parameter (using the `<param>` tag) for the `<page>` element

### 8.2.1.1. Seam Pages Diagram Preferences

In order to customize the layout of Diagram you can go to **Window** → **Preferences** → **JBoss Tools** → **Web** → **Editors** → **Seam Pages Diagram**.

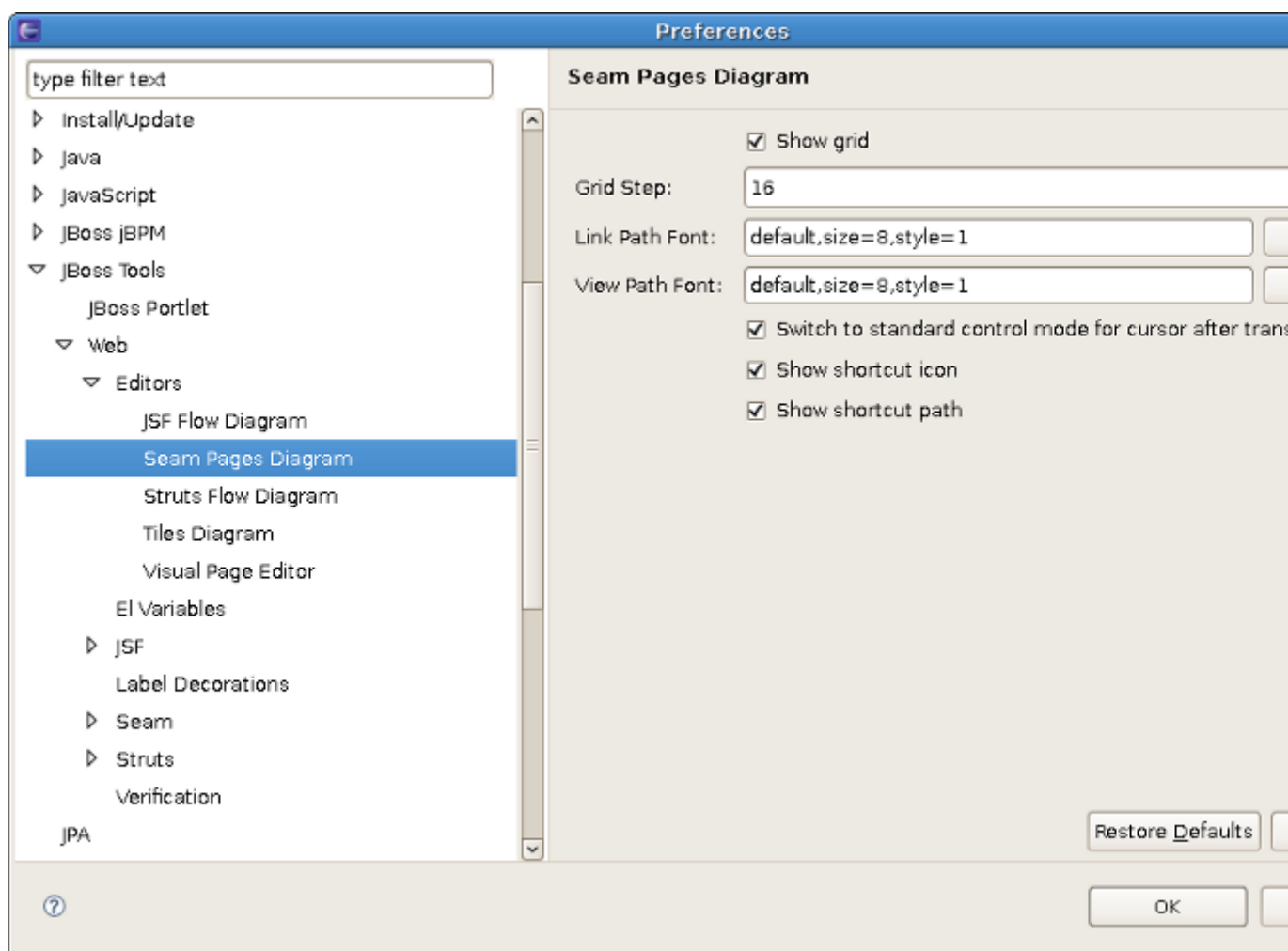
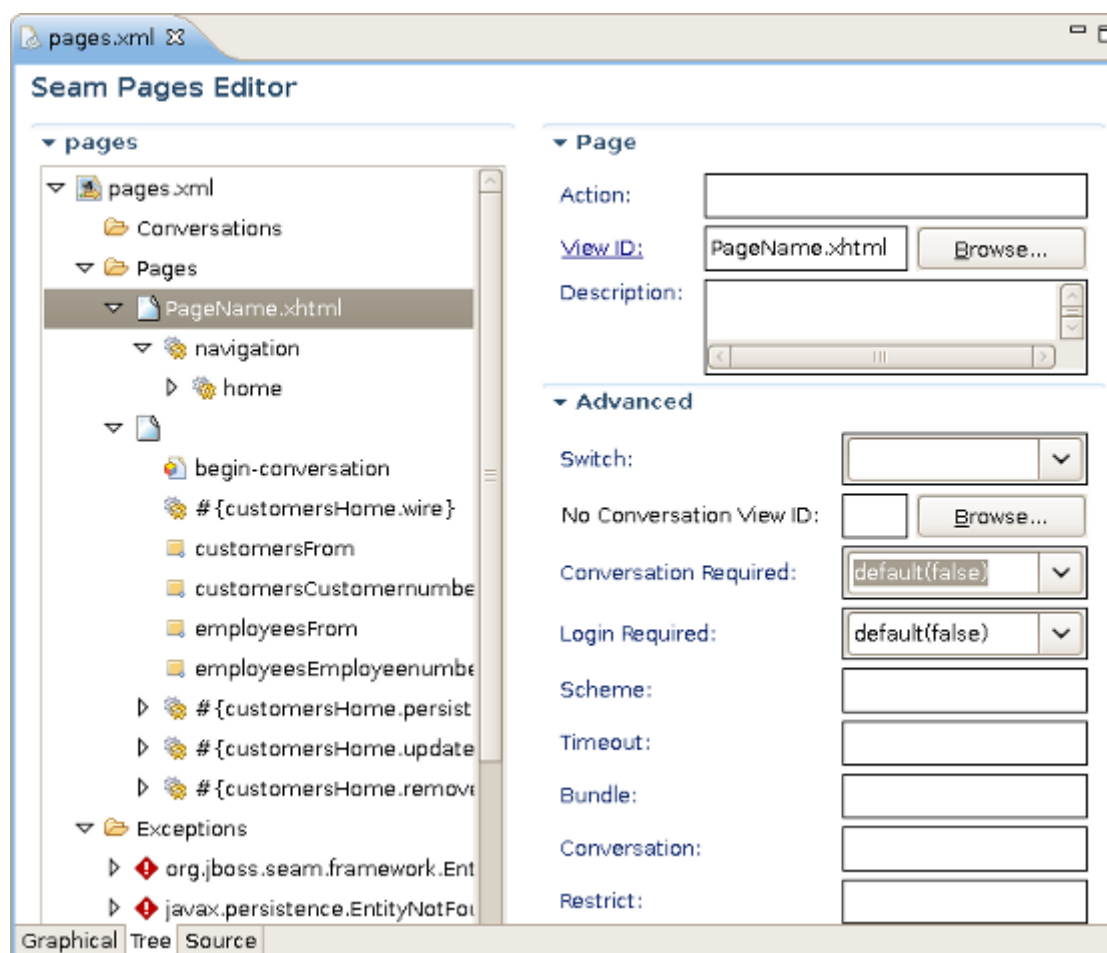


Figure 8.2. Preferences of Seam Pages Diagram

## 8.2.2. Tree Mode



Tree Mode provides a wider range of options and way to edit and modify the pages.xml file.



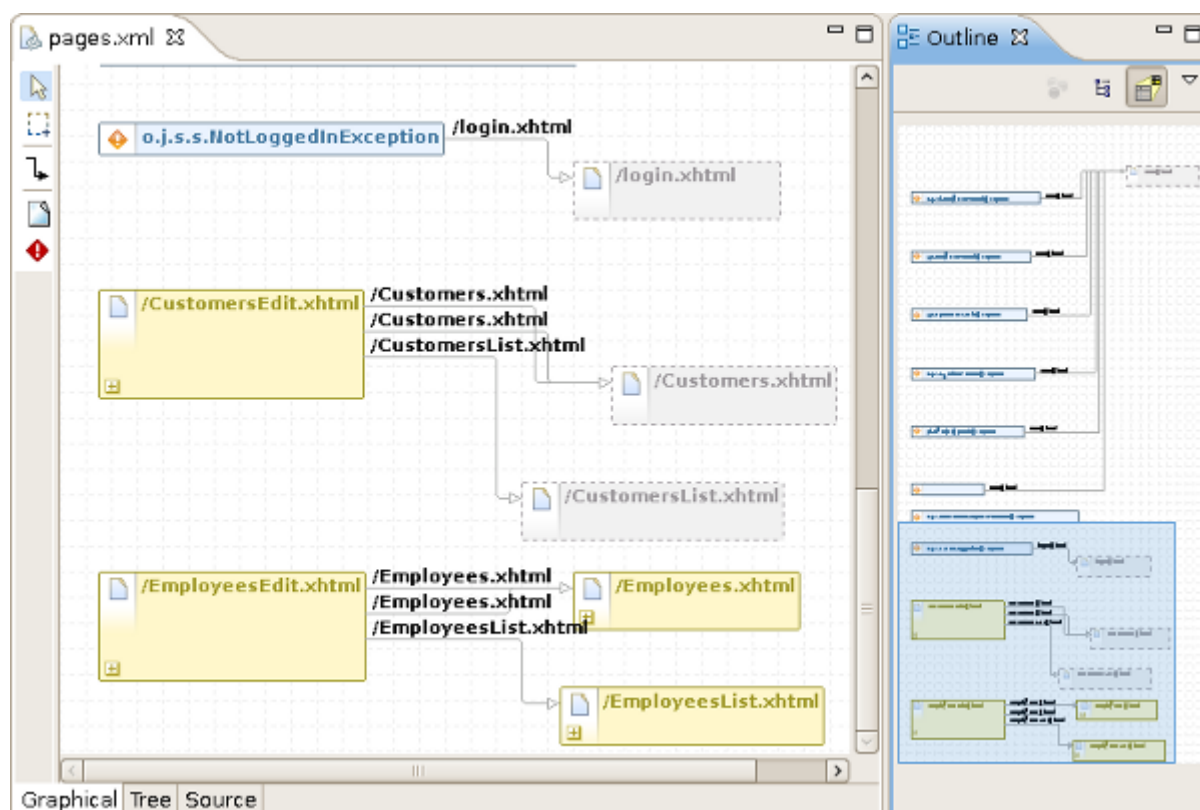
**Figure 8.3. Seam Pages Editor: Tree View**

In this mode all elements and attributes of the page.xml file can be viewed and edited in a visual, user friendly way.

## 8.2.3. Outline Support for Seam Pages Editor

Seam Pages Editor provides Outline view support. You can explore the pages.xml using Outline view in two modes: Tree (  ) and Diagram Navigation (  ).

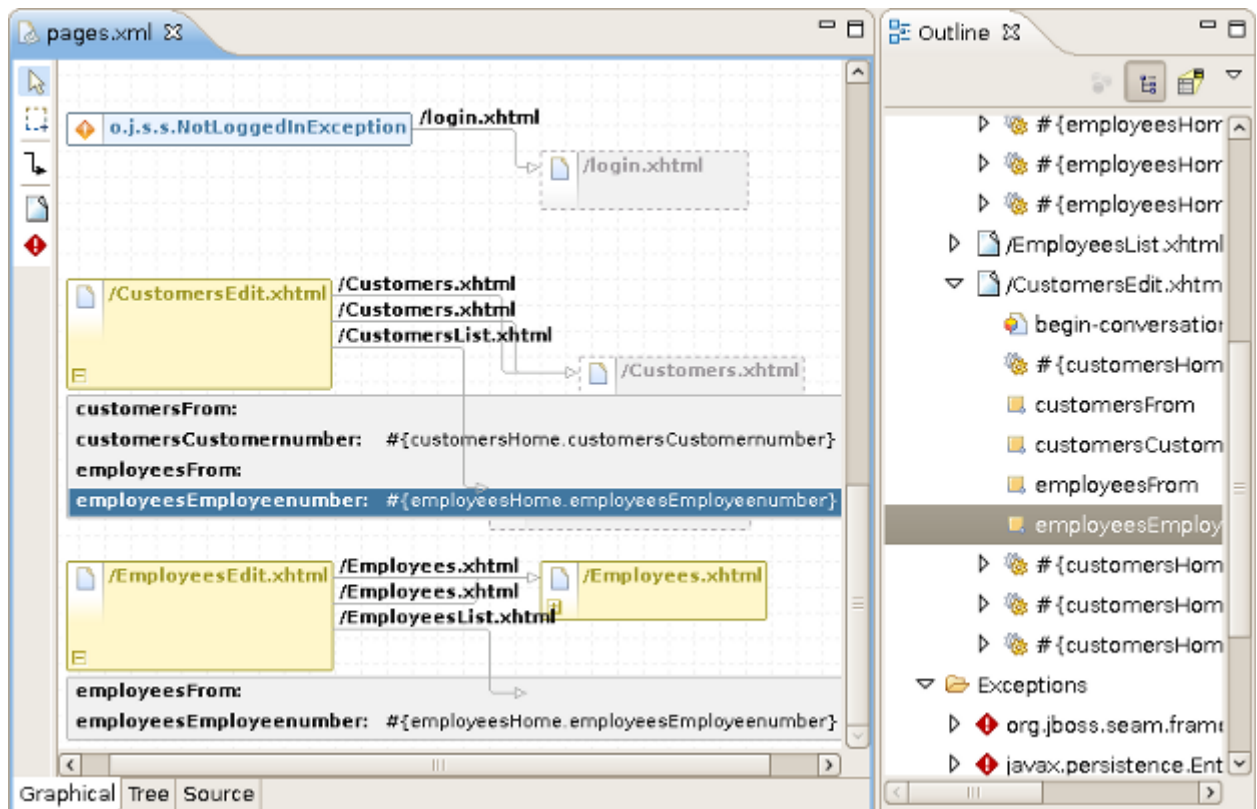
The Diagram Navigation mode of Outline view gives a birds-view of the layout to allow quick navigation for large-scale applications.



**Figure 8.4. Outline view: Diagram Navigation mode**

The Tree mode shows a tree for every element in the pages.xml file. You can get a quick overview of the whole file. When selecting an element in the Outline view the related element is highlighted in the Graphical, Tree or Source page of Seam Pages Editor.

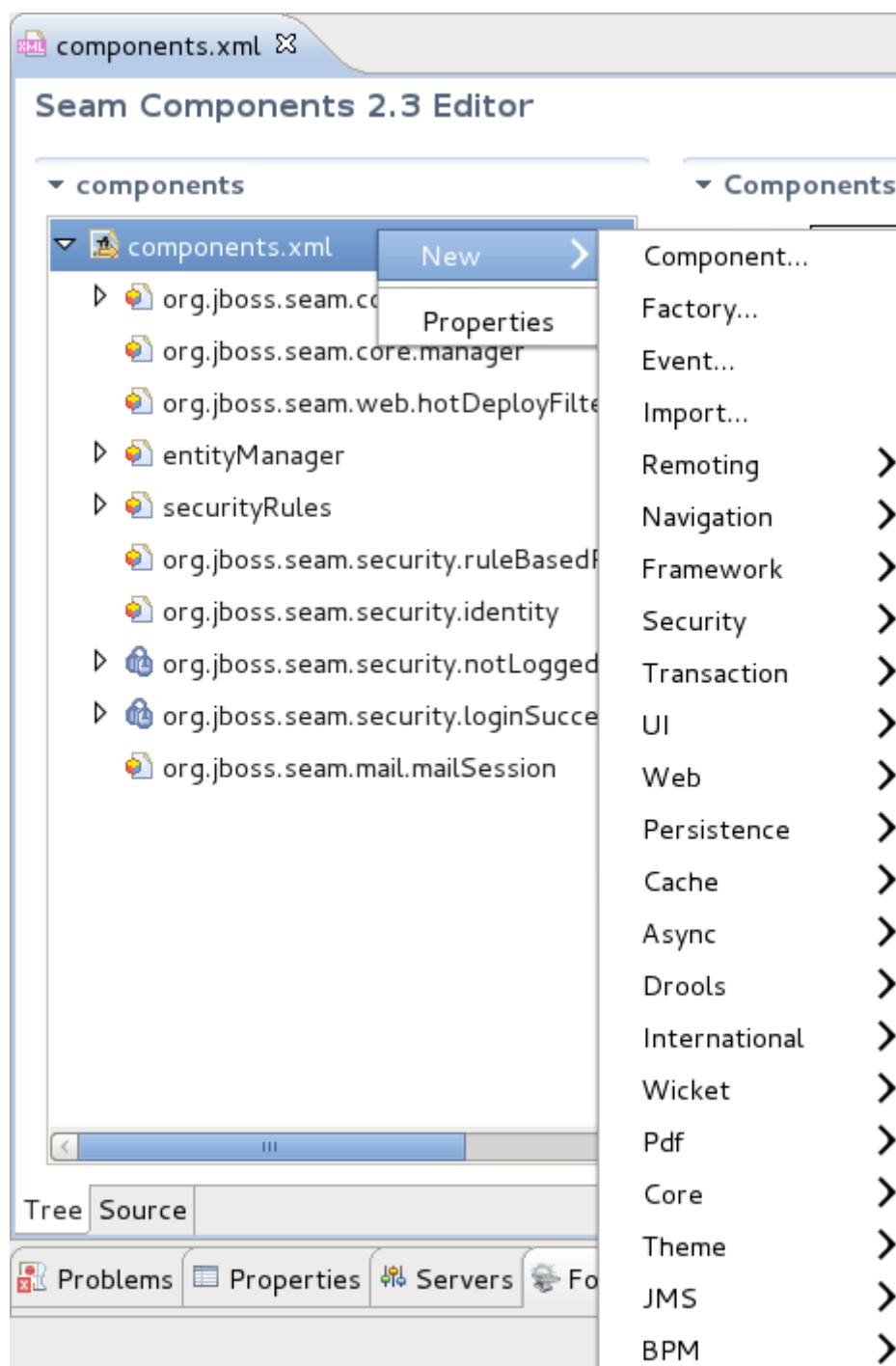




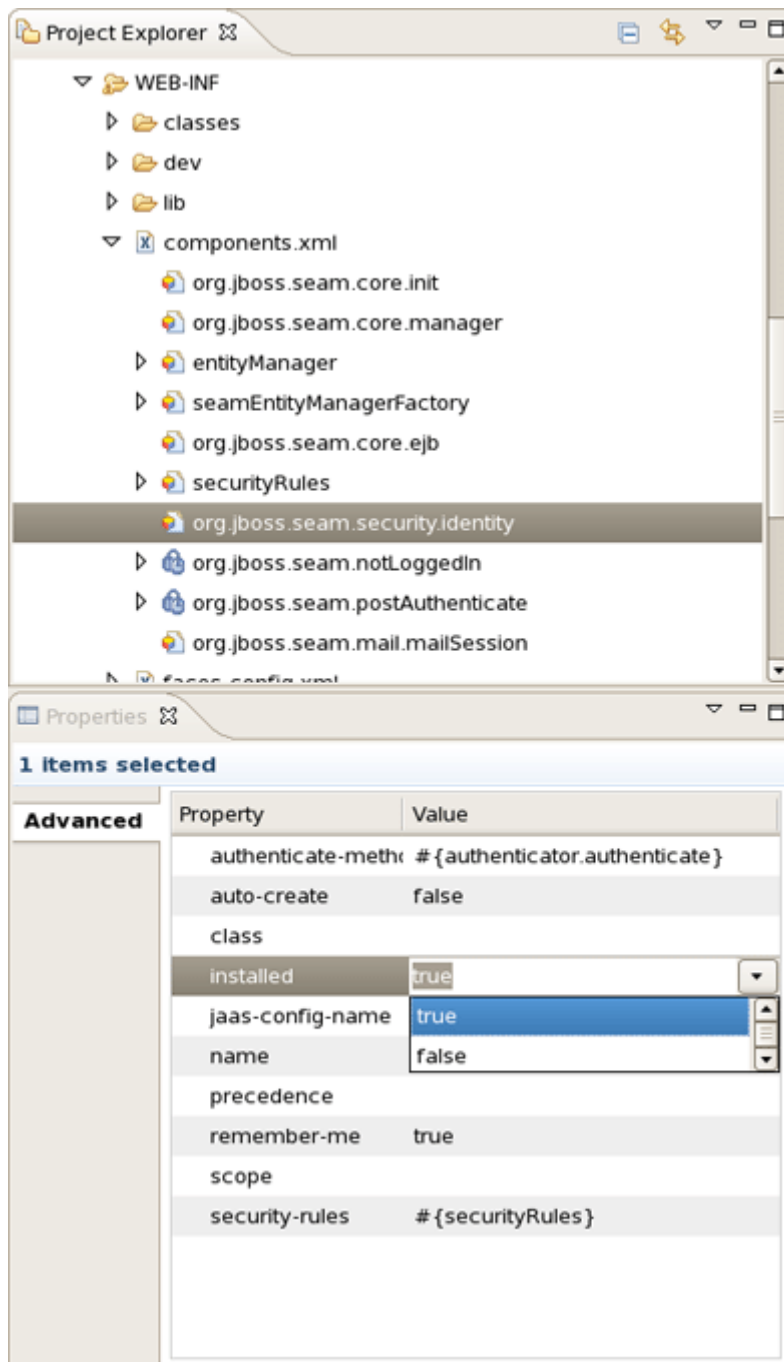
### Figure 8.5. Outline view: Tree mode

### 8.3. Seam Components Editor

When editing `components.xml` a structured tree editor is available in addition to pure source editing. It has a graphical view (Tree tab) and source (Source tab).

**Figure 8.6. component.xml Editor****Note:**

You can view and edit components.xml and other xml files directly in the Project Explorer and Properties sheet without opening the components.xml editor.



**Figure 8.7. component.xml Editor**

## 8.4. Main Features of Seam Editors

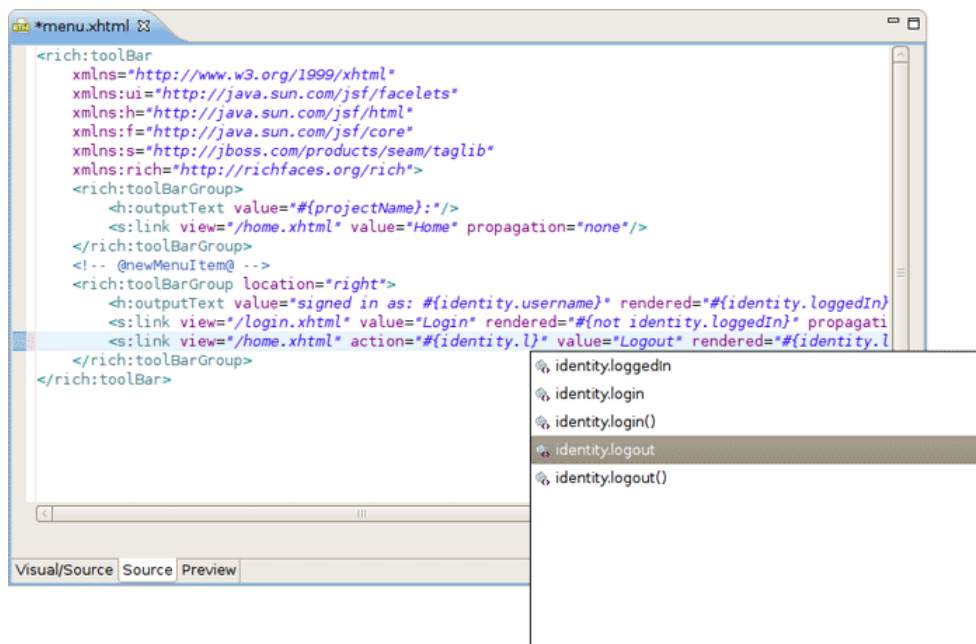
In this chapter you will know what Seam Editors features are and how to work with them.

### 8.4.1. Content Assist

Content Assist (Ctrl + Space) is available when using EL expressions in the following file formats:

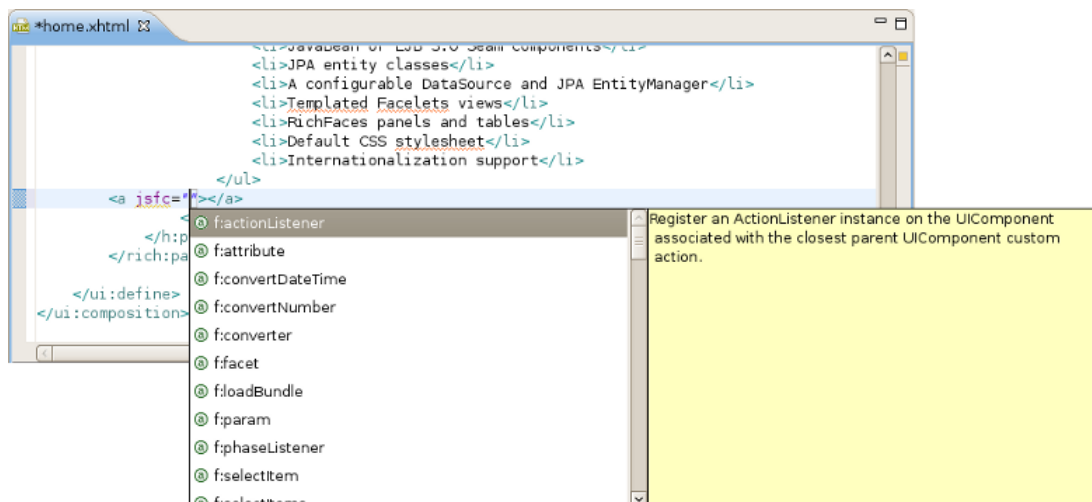
- JSP
- XHTML
- XML
- JAVA

Notice, the code completion for the Seam components shows the proposals marked with Seam icon.



**Figure 8.8. Content Assist**

Content Assist is also available for the "jsfc" attribute, it lets you select from a list of possible attribute values. Whereas, JSF components appear above all other components in the list.



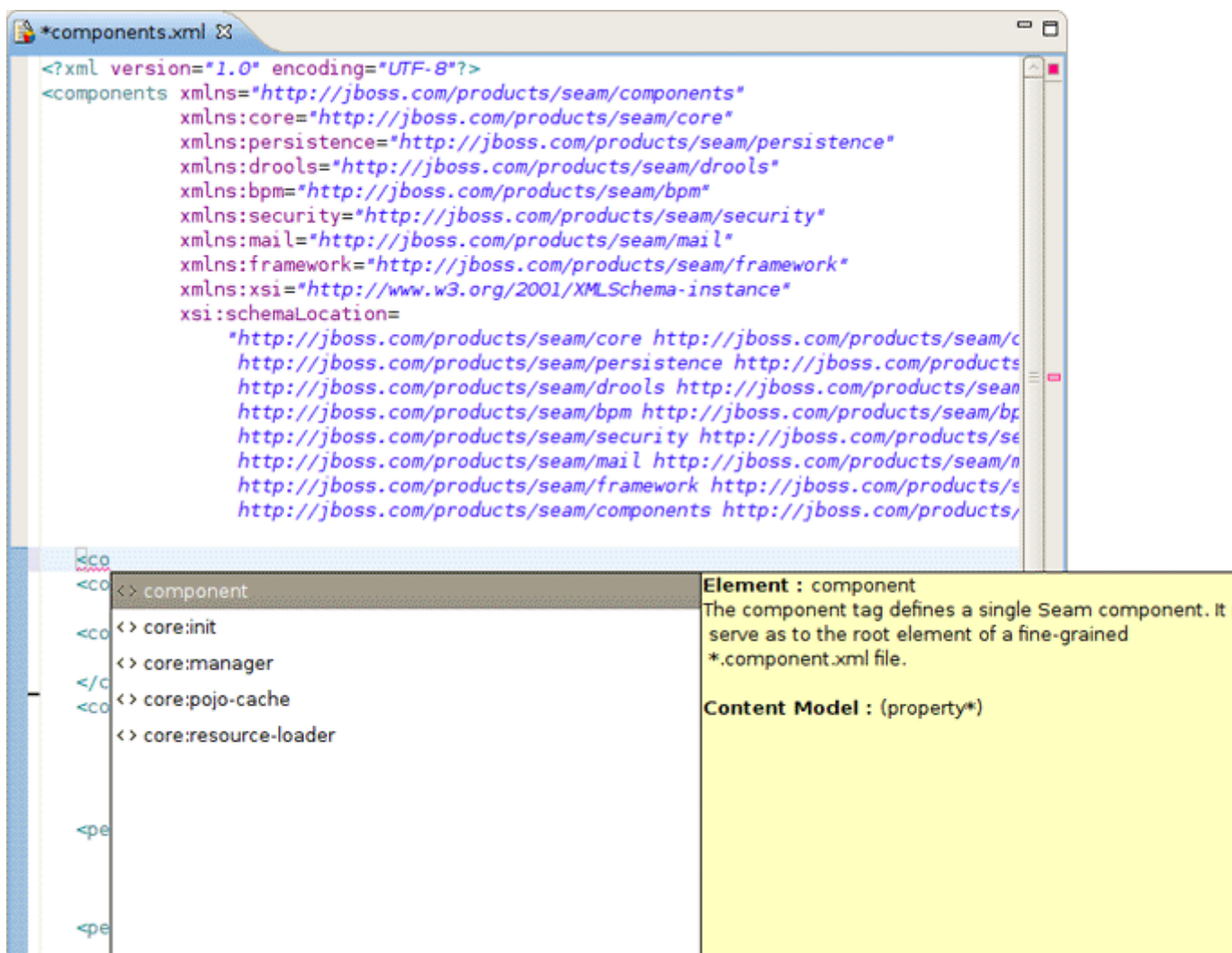
**Figure 8.9. Content Assist for the "jsfc" attribute**

**Note:**

To get Code Assist available for an externally generated and imported project, don't forget to enable Seam features and configure Seam Settings in [Section 10.2](#), “Project Preferences”.

### 8.4.1.1. Content Assist for components.xml

While you are editing a `components.xml` file you can make use of the Content Assist for inserting needed elements.



**Figure 8.10. Content Assist in the component.xml Editor**

The editor also makes code completion aware of methods/attributes for EntityQuery component.

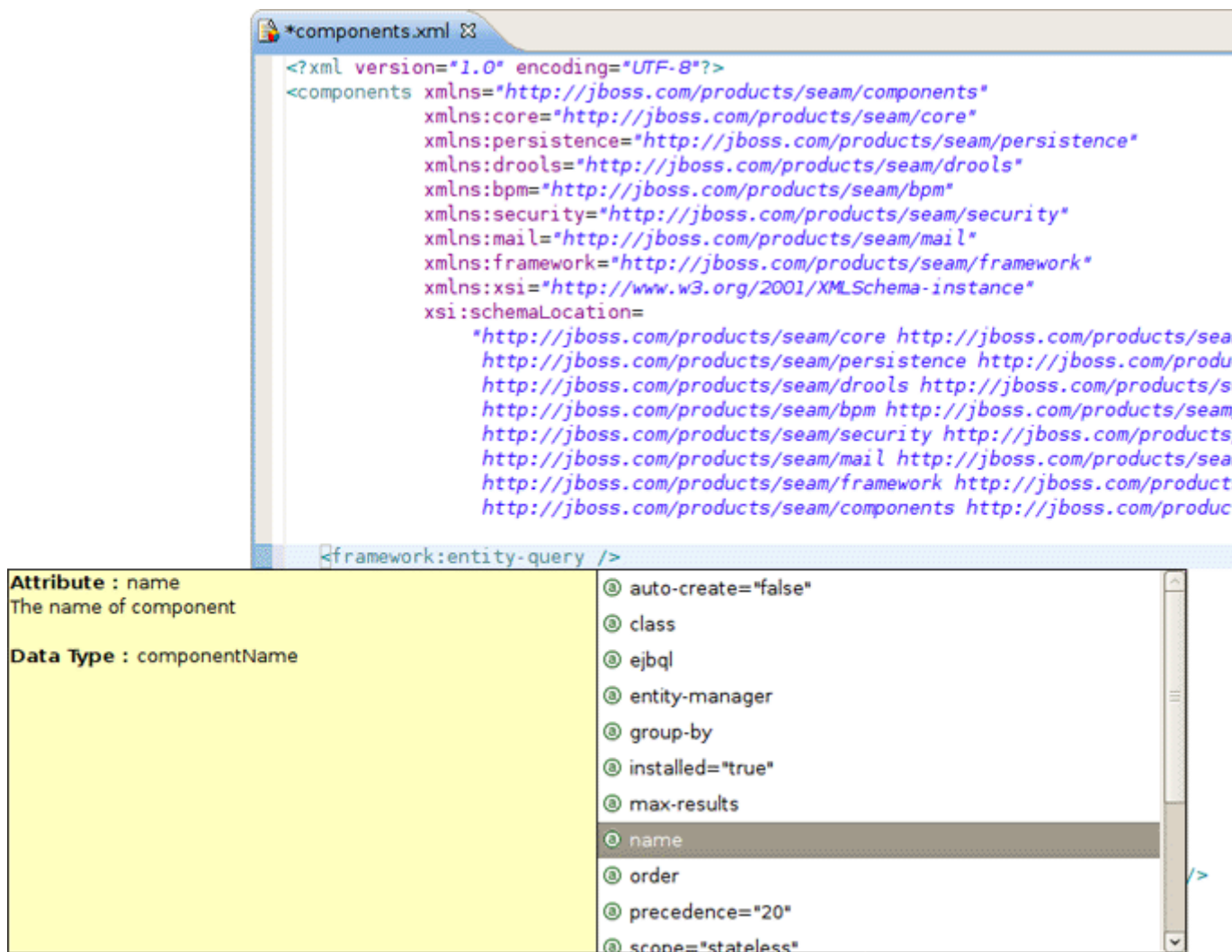
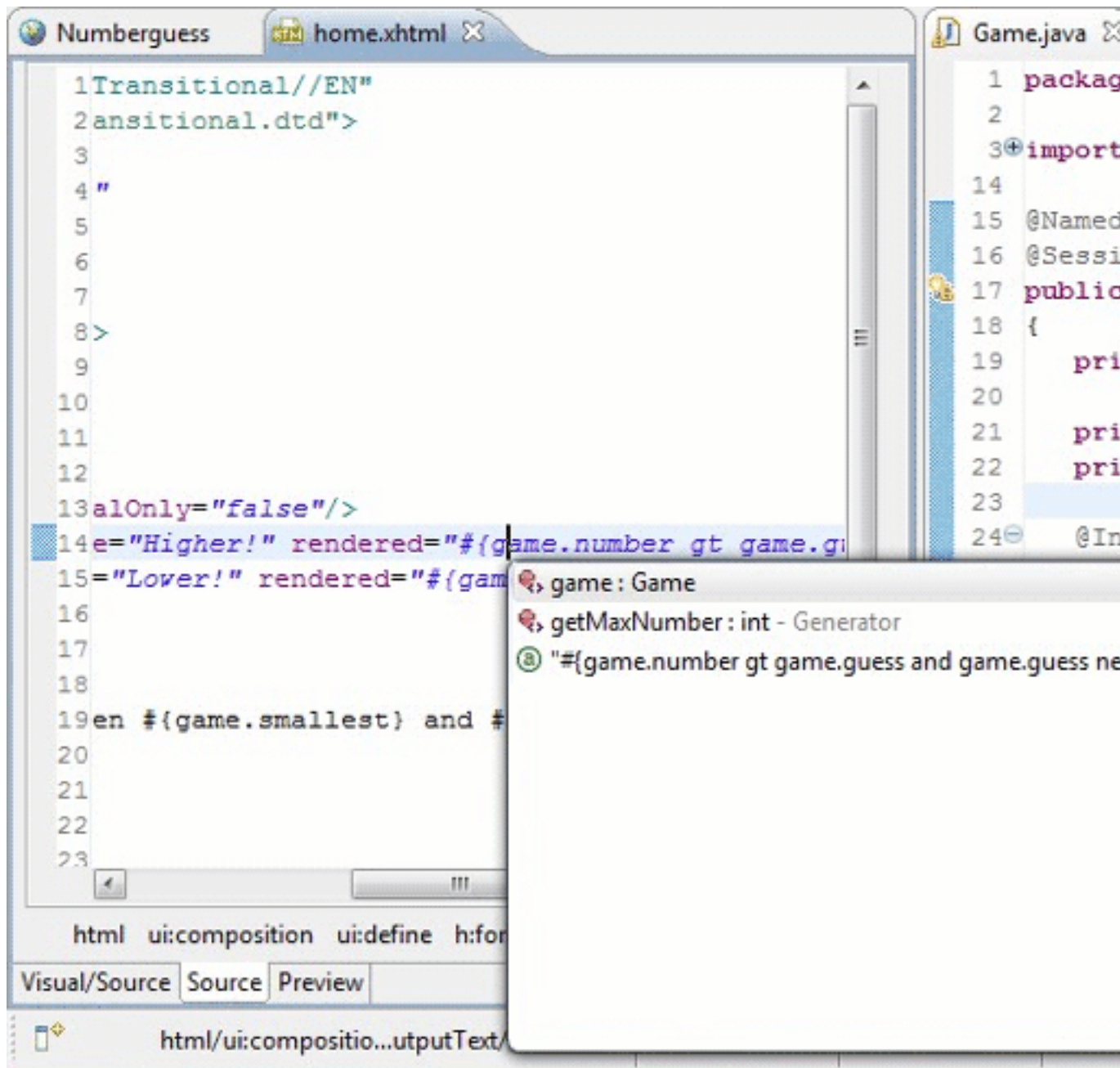


Figure 8.11. Content Assist for EntityQuery Component

#### 8.4.1.2. Content assist for @Named beans

To the nice code completion you can also use Content assist for all occurrences where @Named beans are used.





**Figure 8.12. Content assist for @Named beans**

### 8.4.2. Content Assist for Page Descriptors

Content Assist ( CA ) is available for Page Descriptors, `.page.xml` and `page.xml` files. Content Assist lets you easily compose a Page Descriptor file by suggesting elements and attributes. The suggestions are context dependent, which means that CA provides only the elements, attributes and in some cases values for the attributes, suitable in a particular place of your Page Descriptor.

Seam Pages Editor also supports code completion in EL-expressions suggesting you possible bean properties and methods.

### 8.4.3. OpenOn

OpenOn lets you easily navigate through your project without using the Package Explorer or Project Explorer. After pressing *Ctrl + left click* (or just F3) you will see a corresponding method or class.



**Figure 8.13. OpenOn**

OpenOn is available for the following files:

- [OpenOn in XHTML files](#)
- [OpenOn in XML files](#)
- [OpenOn in Java files](#)



#### Note

For more information on OpenOn, see the Visual Web Tools Reference Guide.

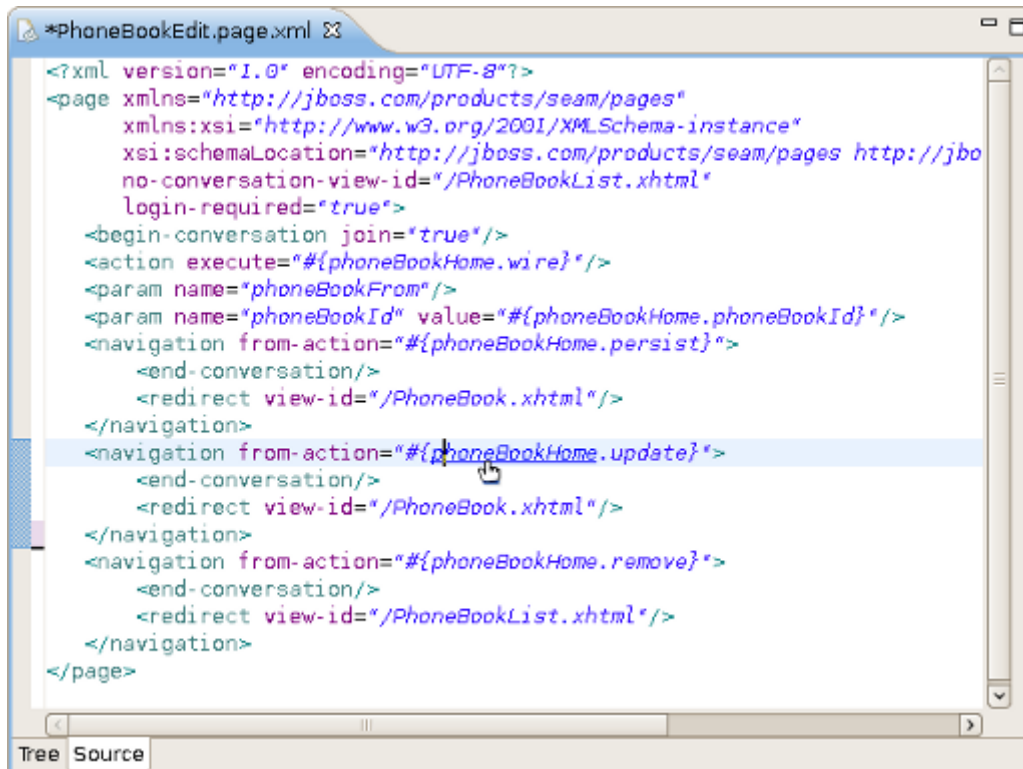
**OpenOn in XHTML files.** Using OpenOn in .xhtml files you can open:

- components or properties resolved using an EL expression;
- all actions described in pages.xml;
- other .xhtml files that paths are specified in a current file;



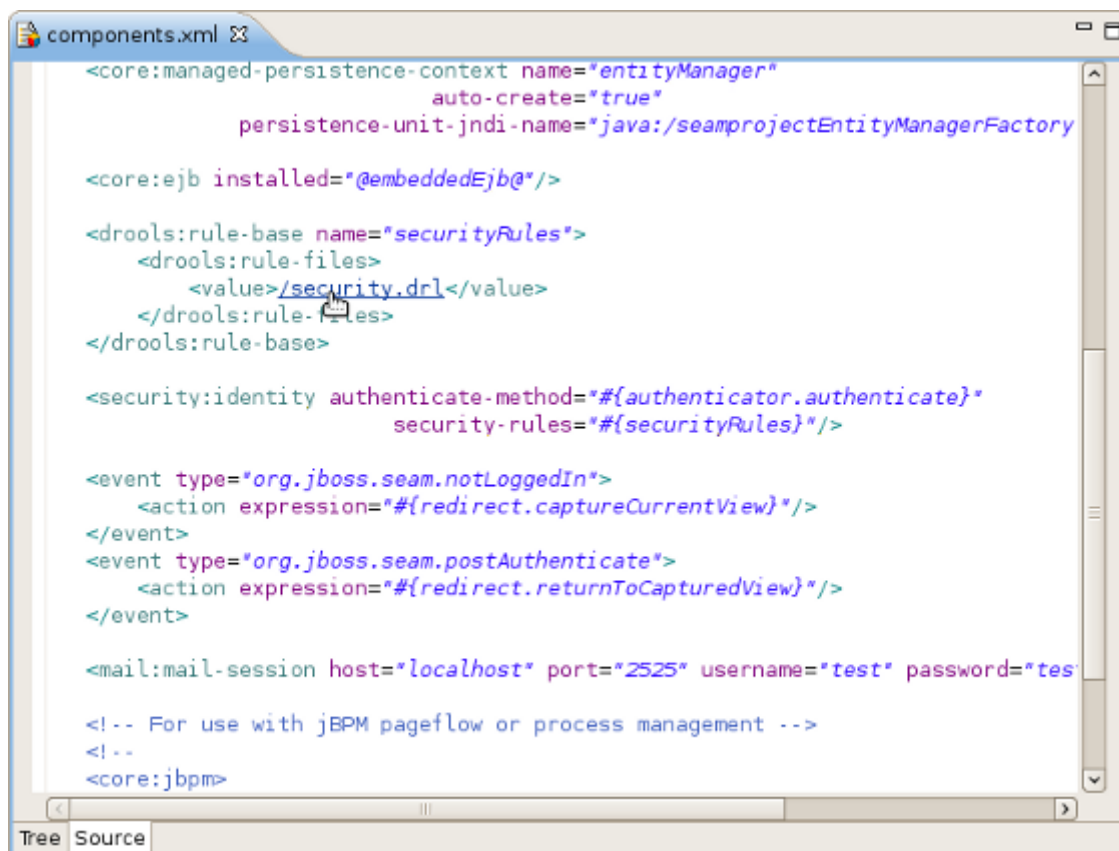
- image files that paths are set in the `value` property. The files will be opened in the default system graphical editor.

**OpenOn in XML files.** OpenOn is also supported in Page Descriptors (`.page.xml` and `pages.xml`). OpenOn allows you to navigate to Java beans as well as to view pages. You can Ctrl + left click on a bean or on view page file (for example, a XHTML file) to navigate to the file you clicked on.



**Figure 8.14. OpenOn in .page.xml file**

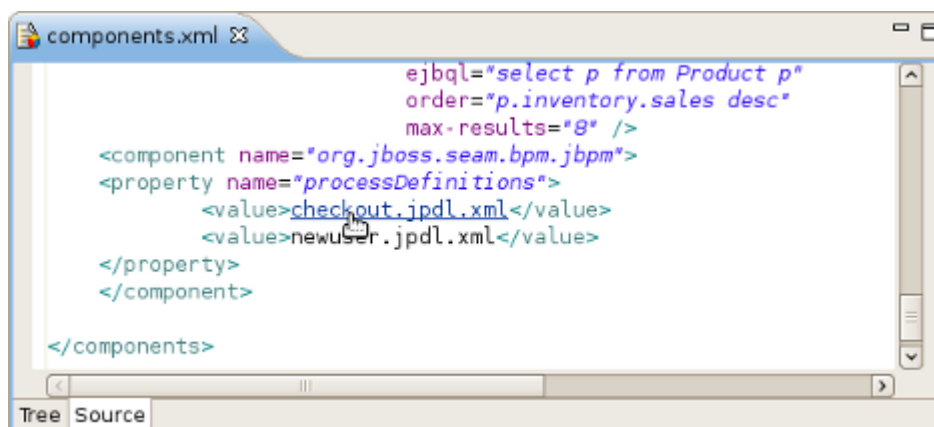
If you use Rule-based authorization with Drools in your Seam application you can apply OpenOn in `components.xml` to open the files containing rule definitions in the `"rule-files"` tag.



**Figure 8.15. OpenOn in components.xml file**

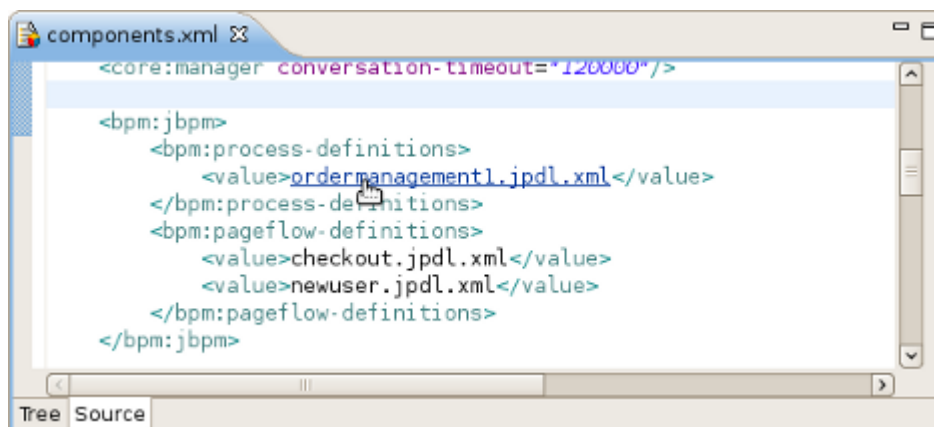
You can also use OpenOn to open jBPM components that are described in Seam component descriptor. Two places where it works are available:

- inside `<component>` definition;



**Figure 8.16. OpenOn in components.xml file**

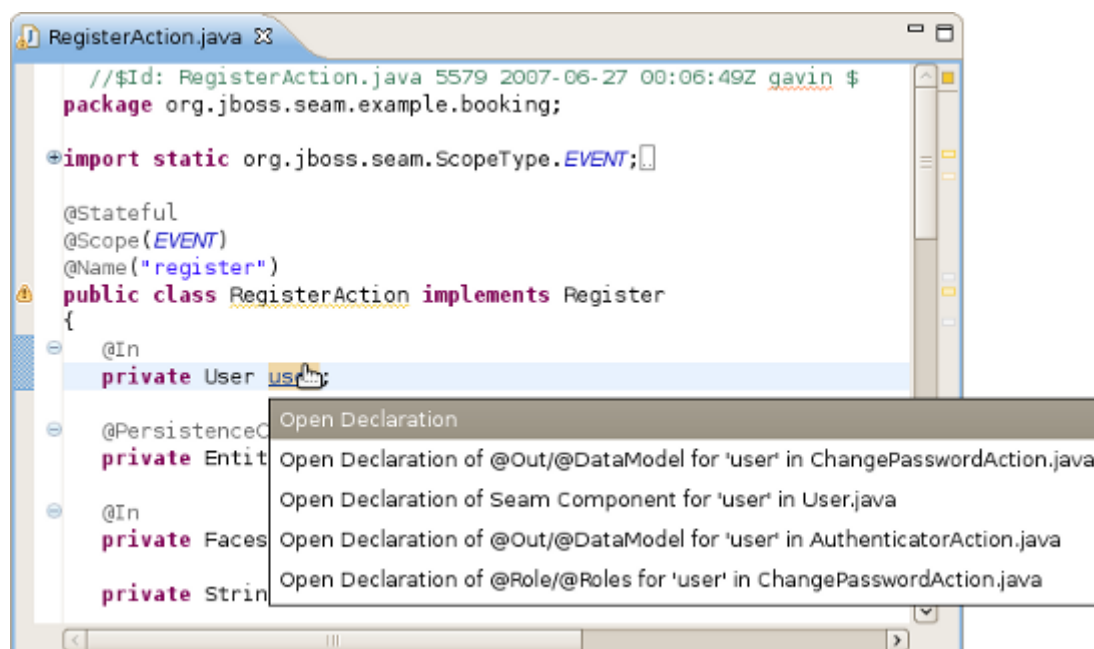
- inside `jbpm` tags;



**Figure 8.17. OpenOn in components.xml file**

**OpenOn in Java files.** OpenOn is also supported in seam components where In annotation is presented. After pressing Ctrl + left click on the seam component specified in @In you will get the possibility to open the file where the component is declared as well as all of the seam components where it is used in the next declarations:

- @Out
- @DataModel
- @Role
- @Roles

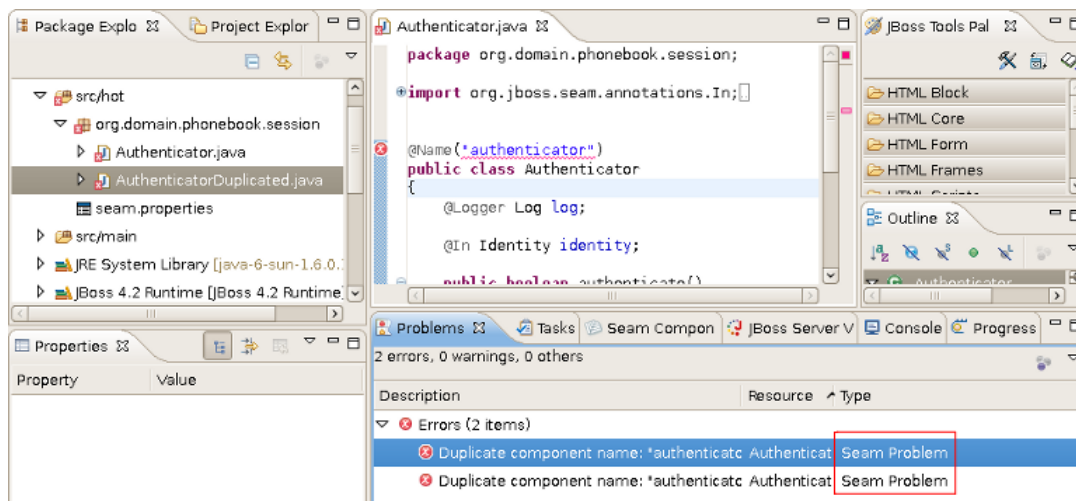


**Figure 8.18. OpenOn in Seam Component**

### 8.4.4. Seam Validation

Validation of various possible problematic definitions is implemented for Seam applications.

If an issue is found it will be shown in the standard Problems View.



**Figure 8.19. Seam Validation**



#### Note:

Please note, only files within source path and web content are validated, meaning temporary files or test files are not included.

In the preferences page for Seam Validator you can see and modify the validator behavior. Go to **Window → Preferences → JBoss Tools → Web → Seam → Validator** and select the severity level for the optional Seam Validator problem.

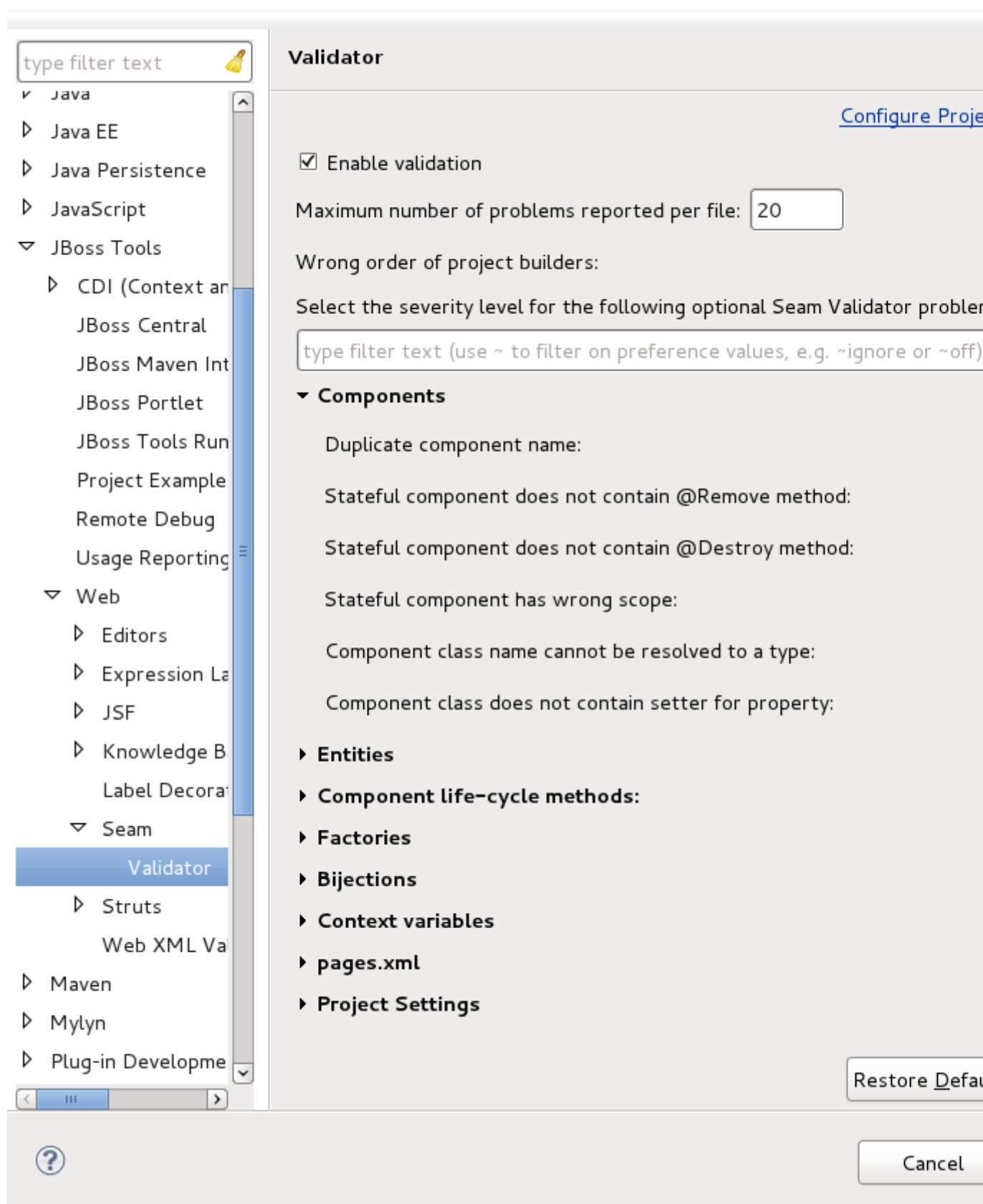
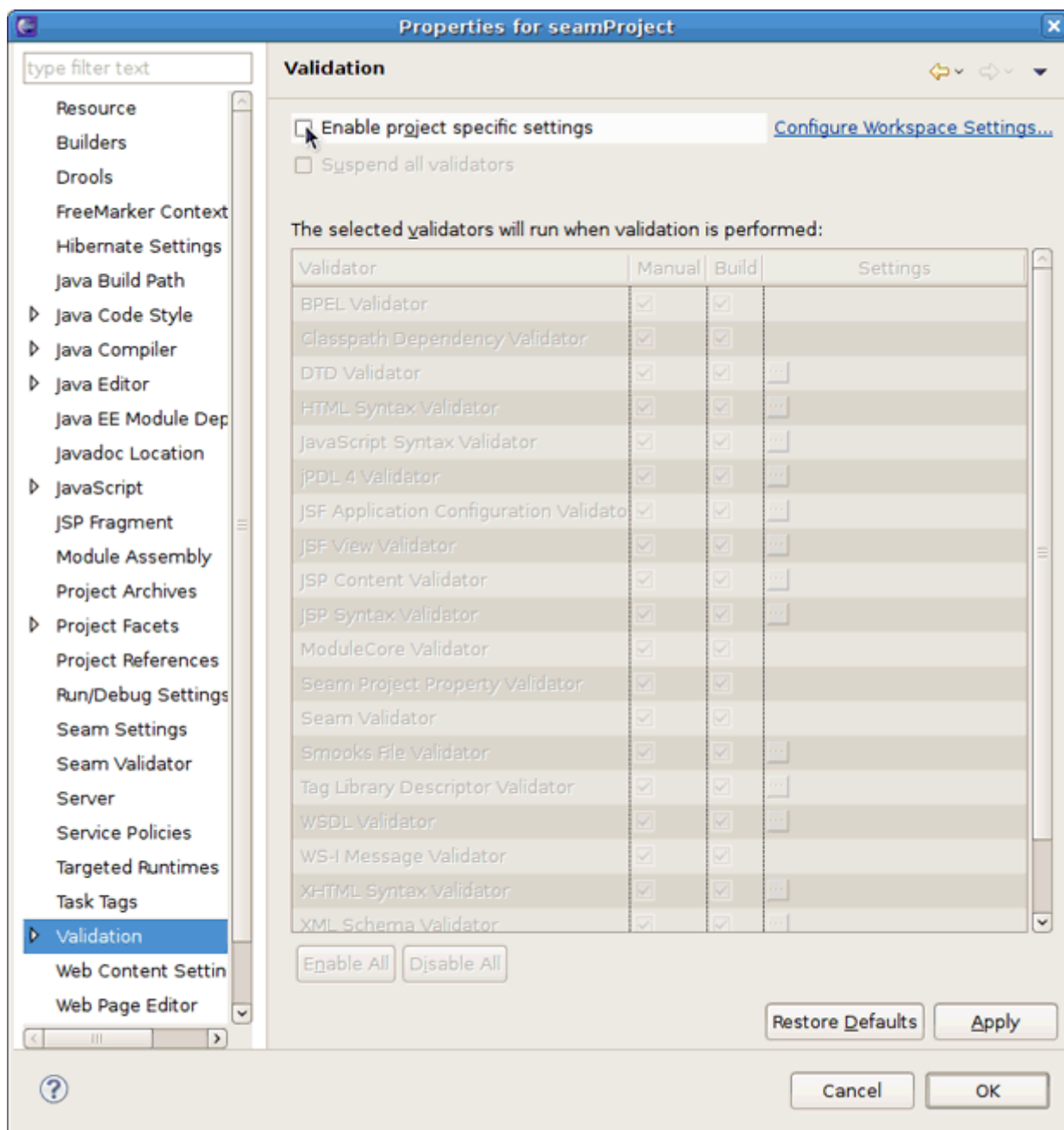


Figure 8.20. Seam Validator Preferences

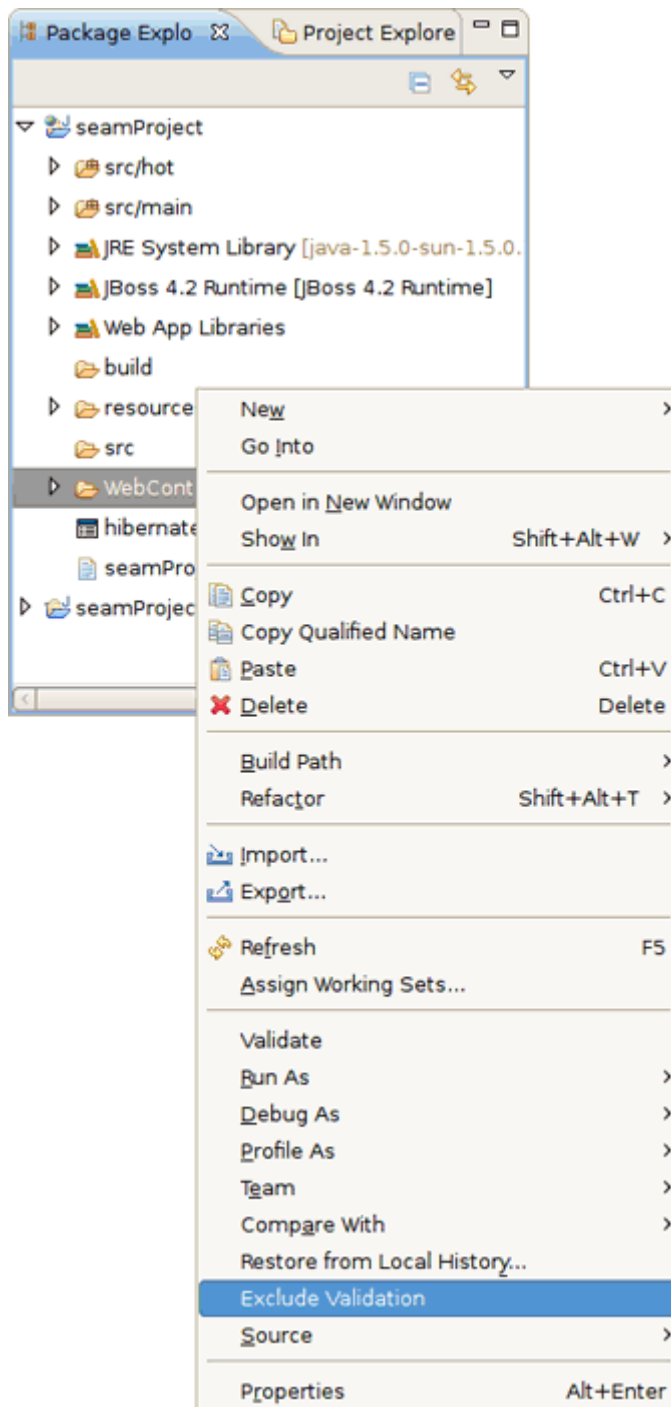
On WTP projects validation are enabled by default and thus executed automatically, but on normal Java projects you will have to go and add the Validation builder of your project . It is available in the properties of your project under *Validation*.



**Figure 8.21. Enabling the Validation Builder**

The validations can be run manually by clicking *Validate* via the context menu on your project (folder or file inside your project) which will execute all the active WTP validations. Following to the project's preferences under *Validation* it's possible to enable/disable any validator.

It's also possible to turn off the validation for some resource (a file, folder) inside your project via the *Exclude Validation* context menu option.



**Figure 8.22. Excluding Validation for the WebContent Folder**

As a result, if you've turned off the validation for a folder (or file), it is put to the excluded group of the Validation filters and be ignored during the validation.

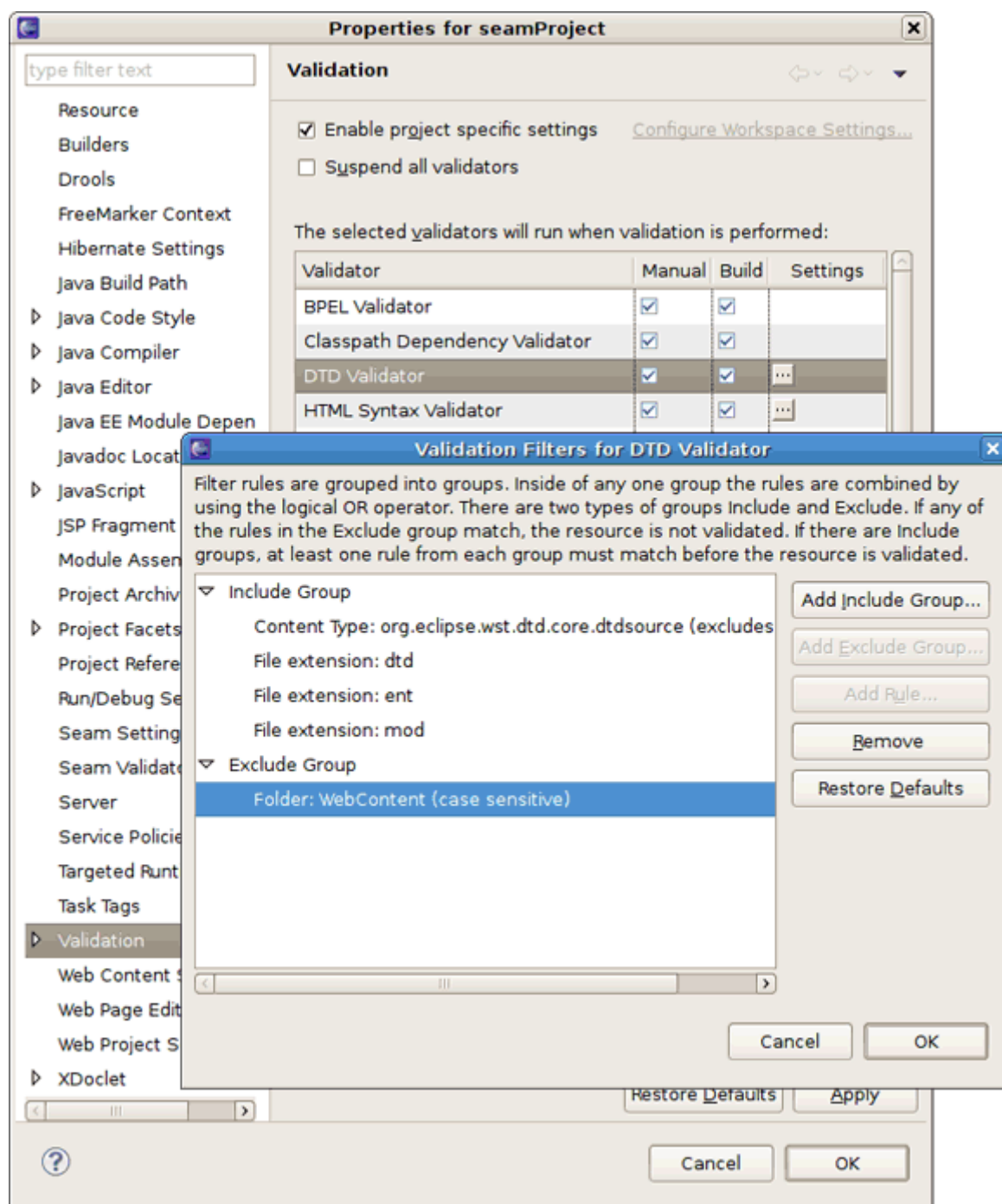


Figure 8.23. Validation Filters

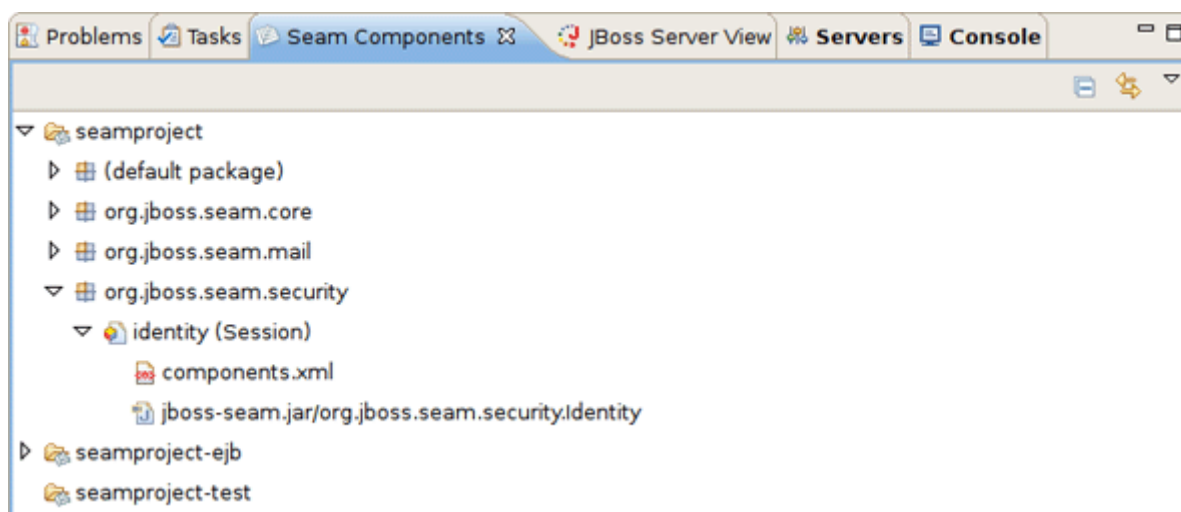


# Seam Views

## 9.1. Seam Components View

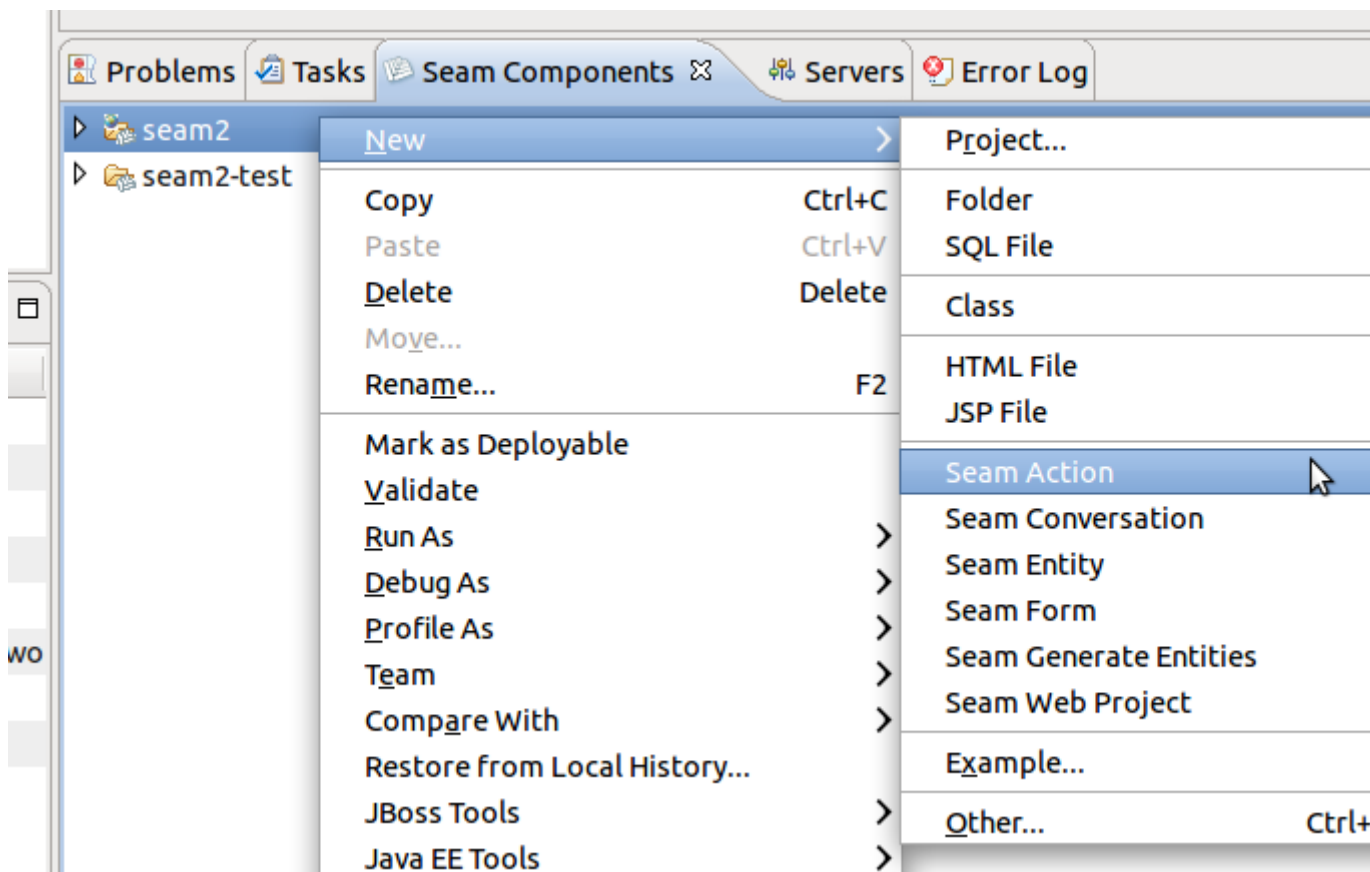
This chapter introduces you with **Seam Components** View.

The **Seam Components** view is available from Seam perspective. It provides a list of seam components found in a project.



**Figure 9.1. Seam Components View**

By right-clicking on a project and bringing up the context menu, you can create a new Seam project, action, conversation, form or entity. You can also generate entities from this menu and it is all able to be done from the **Seam Components** view.



**Figure 9.2. Seam Components View**

The **Seam Components** View can show a components default scope in two ways:

- as labels on each component (click on the triangular symbol at the top of the Seam Components View page and select **Scope Presentation** → **Label**)



**Figure 9.3. Label Scope Presentation of Seam Components**

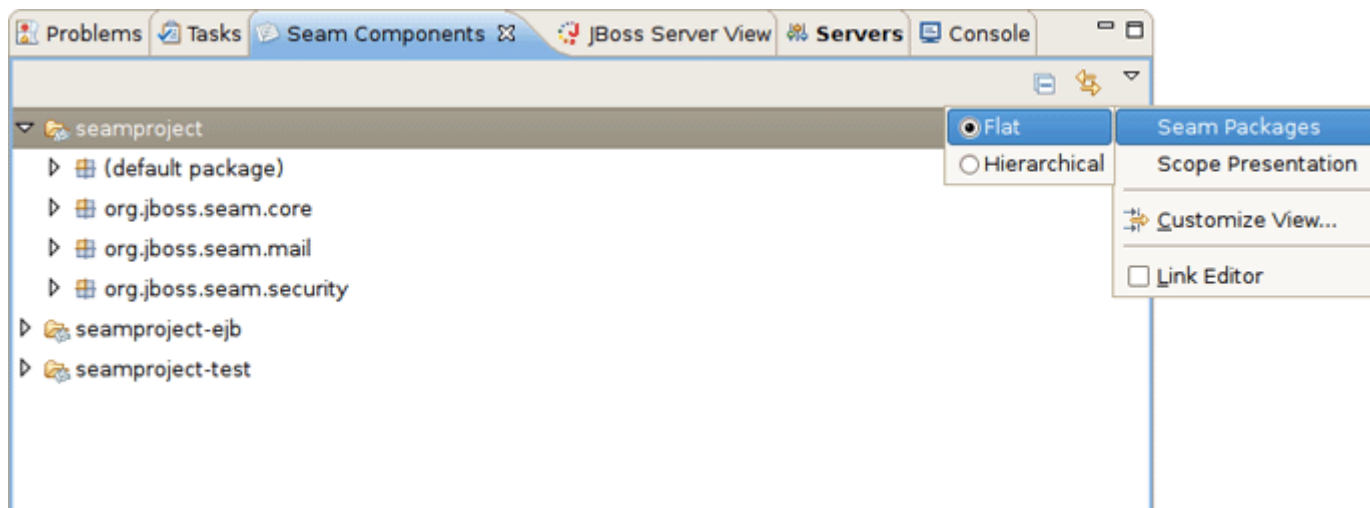
- as a node per scope where the components are grouped under a node representing its default scope.



**Figure 9.4. Node Scope Presentation of Seam Components**

The Seam Packages can be presented in two ways:

- Flat



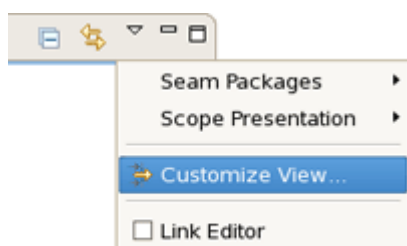
**Figure 9.5. Flat Presentation of Seam Packages**

- Hierarchical



**Figure 9.6. Hierarchical Presentation of Seam Packages**

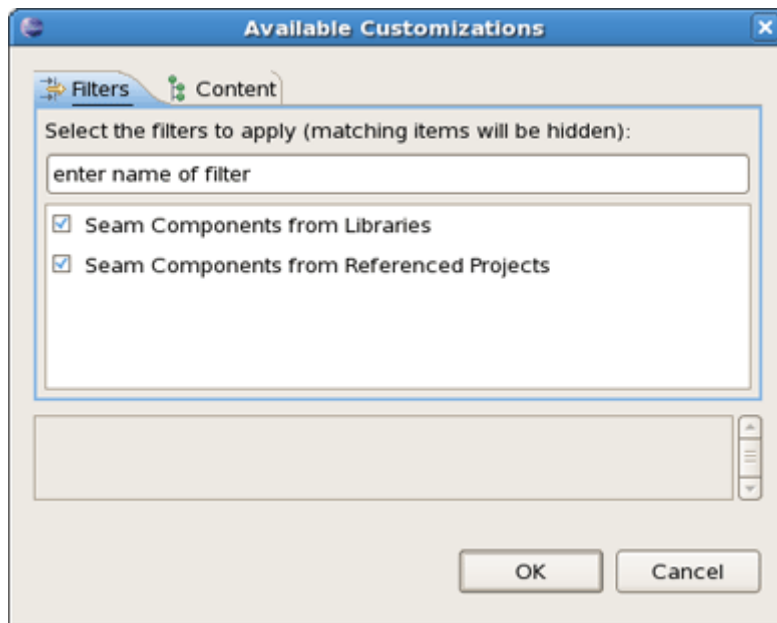
The **Seam Components** view can be filtered by choosing **Customize View**.



**Figure 9.7. Customize View**

Select the *Seam Components from Libraries* under the **Filters** tab. This will make the view ignore components defined in jars. This will hide the many built-in Seam components and leave only those that are actually defined in the project or have been actively configured via `components.xml`. Therefore, deselecting the filter will show you all available components.

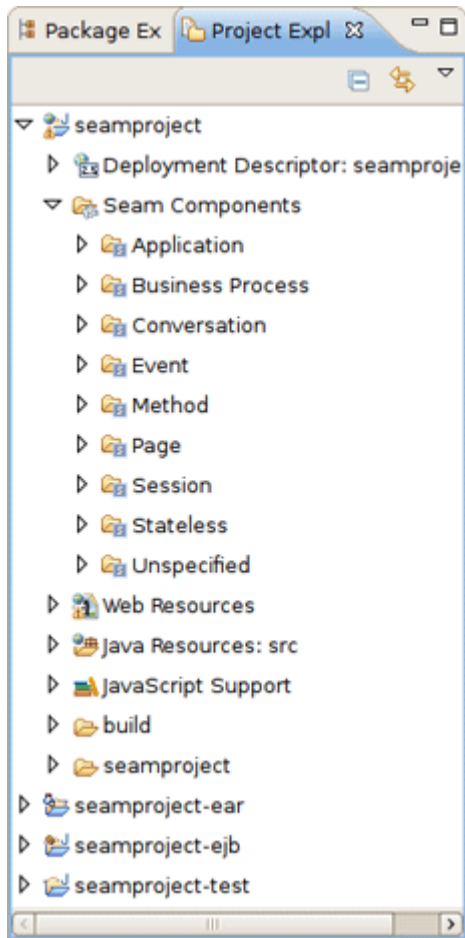
Selecting the *Seam Components from Referenced Projects* will hide the components that depend on other projects.



**Figure 9.8. Available Seam Component View Customizations**

## 9.2. Project Explorer integration

If you don't like to have a view for every piece of information in Eclipse, the content of the Seam Components view is also available as a node in the built-in Project Explorer (not Package Explorer!) view in Eclipse.



**Figure 9.9. Seam Components in Project Explorer**

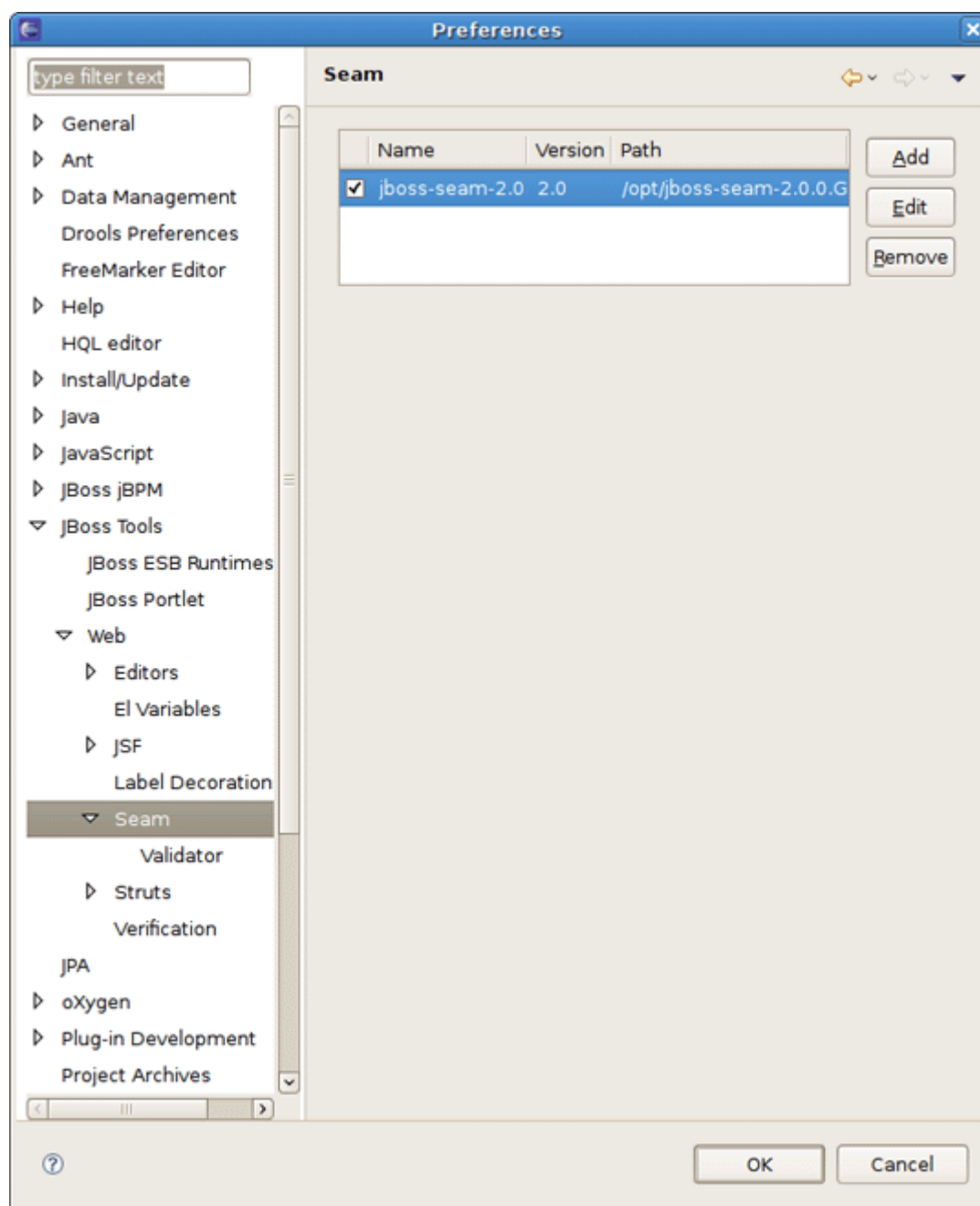
# Seam Preferences

In this chapter you get to know how Seam preferences can be modified during the development process.

## 10.1. General Preferences

Seam preferences can be set using the Seam preference page. Click on **Window** → **Preferences** → **JBoss Tools** → **Web** → **Seam**.

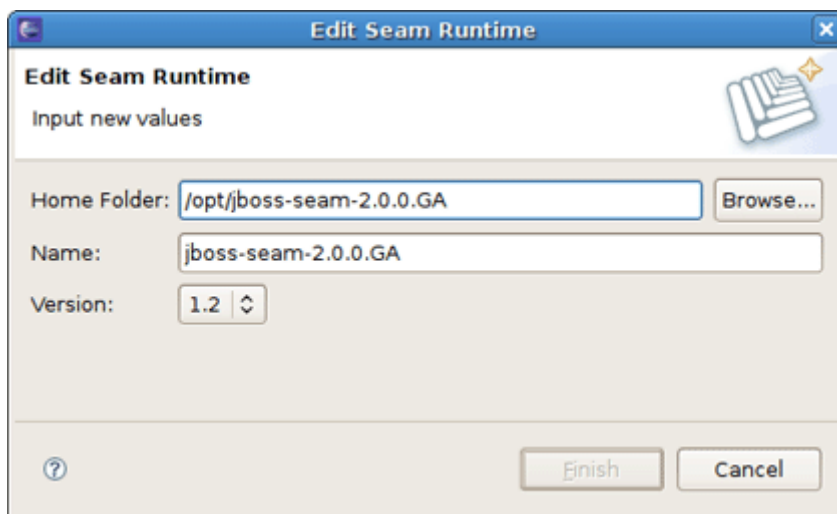
On this page you can manage the Seam Runtime. Use the appropriate buttons to Add more runtimes or to Remove those that are not needed.



**Figure 10.1. Seam Preferences Page**

Clicking on Edit button you get the form where you can change the path of Seam runtime home folder, modify name and version. Press Finish to apply the changes.





**Figure 10.2. Edit Seam Runtime**



**Tip:**

If you try to delete the Seam Runtime that is already in use by some project in the current workspace, then a warning message will appear. To confirm the removal press OK

Besides, when you right-click on your Seam project with deleted runtime in Project Explorer and select **Properties** → **Seam Settings** the error message "Runtime <runtime\_name> does not exist" will appear.

### 10.1.1. Validator Preferences

Seam preference page includes a subsection Validator. See **Window** → **Preferences** → **JBoss Tools** → **Web** → **Seam** → **Validator**.

On this page you can choose a severity level for the various Seam validator problems. For example, if you want to ignore the case when component name is duplicated expand the Components node and select Ignore next to Duplicate component name. After that you won't see the error.

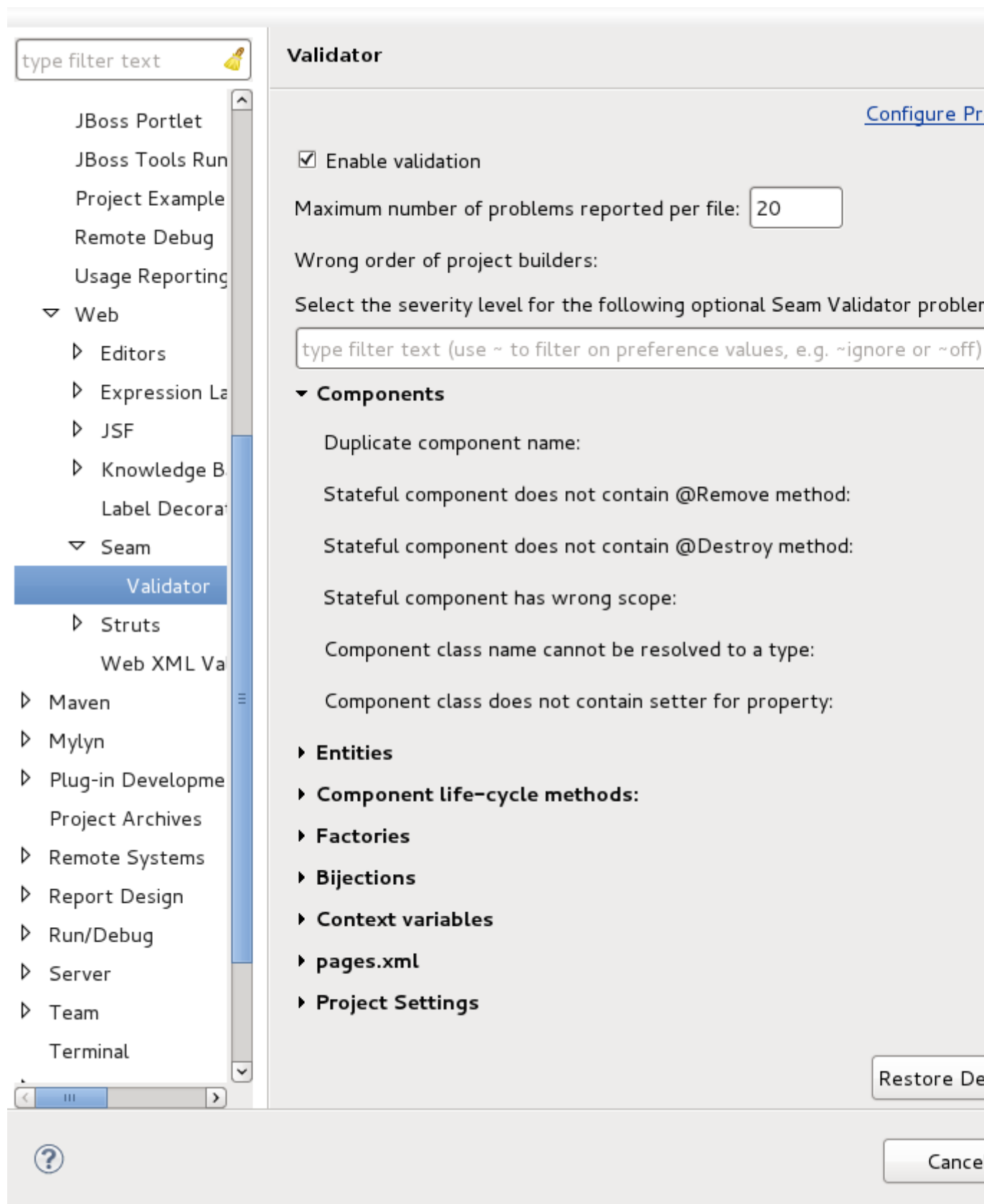
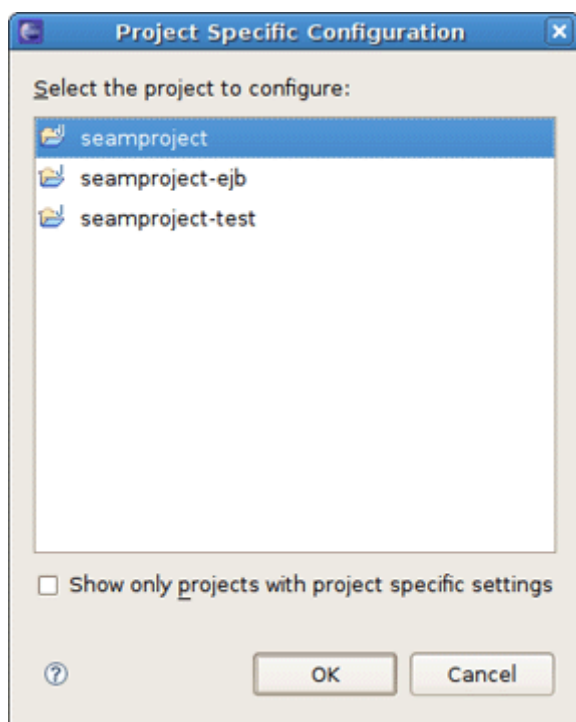


Figure 10.3. Seam Validator Preference Page

In the upper right corner of the Seam Validator preferences page there is a *Configure Project Specific Settings* link. Clicking on it you get the form where you can choose a project for specific setting. Project specific configuration allows you to have different validator settings for each project. Check the *Show only projects with project specific settings* if you want to see the projects that have been already set. Click on Ok.



**Figure 10.4. Project Specific Configuration**

You get the validator properties page for chosen project. Check the Enable project specific settings to be able to change the settings.



**Note:**

You can open the same page by right clicking on the needed project in Package Explorer, then **Properties** → **Seam Validator**.

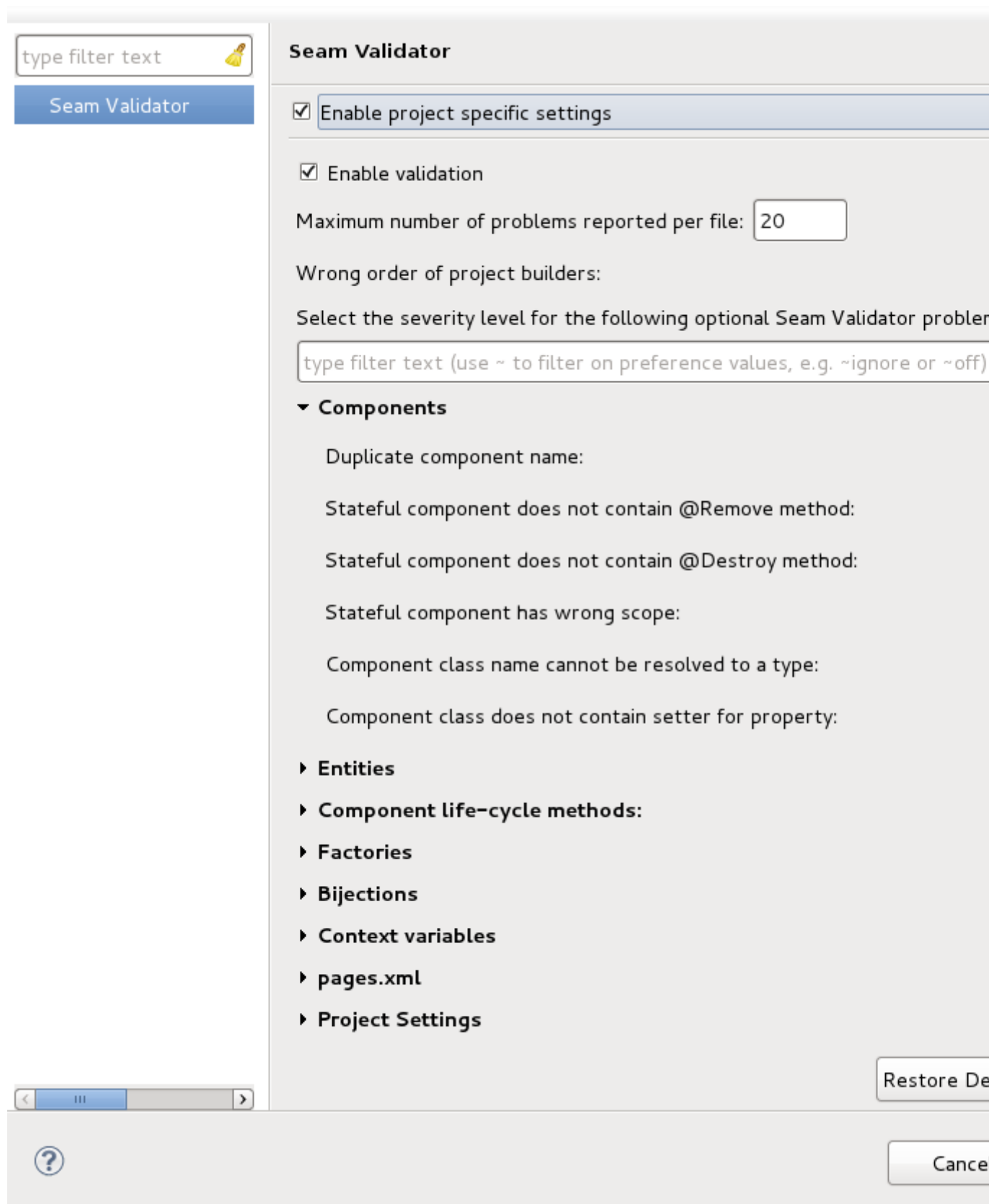


Figure 10.5. Validator Properties Page for Chosen Project

## 10.2. Project Preferences

Once Seam project is created you can modify its settings. Right click on Seam project in Project Explorer and select **Properties** → **Seam Settings**.

This project properties page allows you to have a flexible project layout. It means that you are not restricted with a specific project structure. You can use the Seam wizards (New Action, Form, Entity, etc.) on Maven, command line seam-gen or your own project structure.

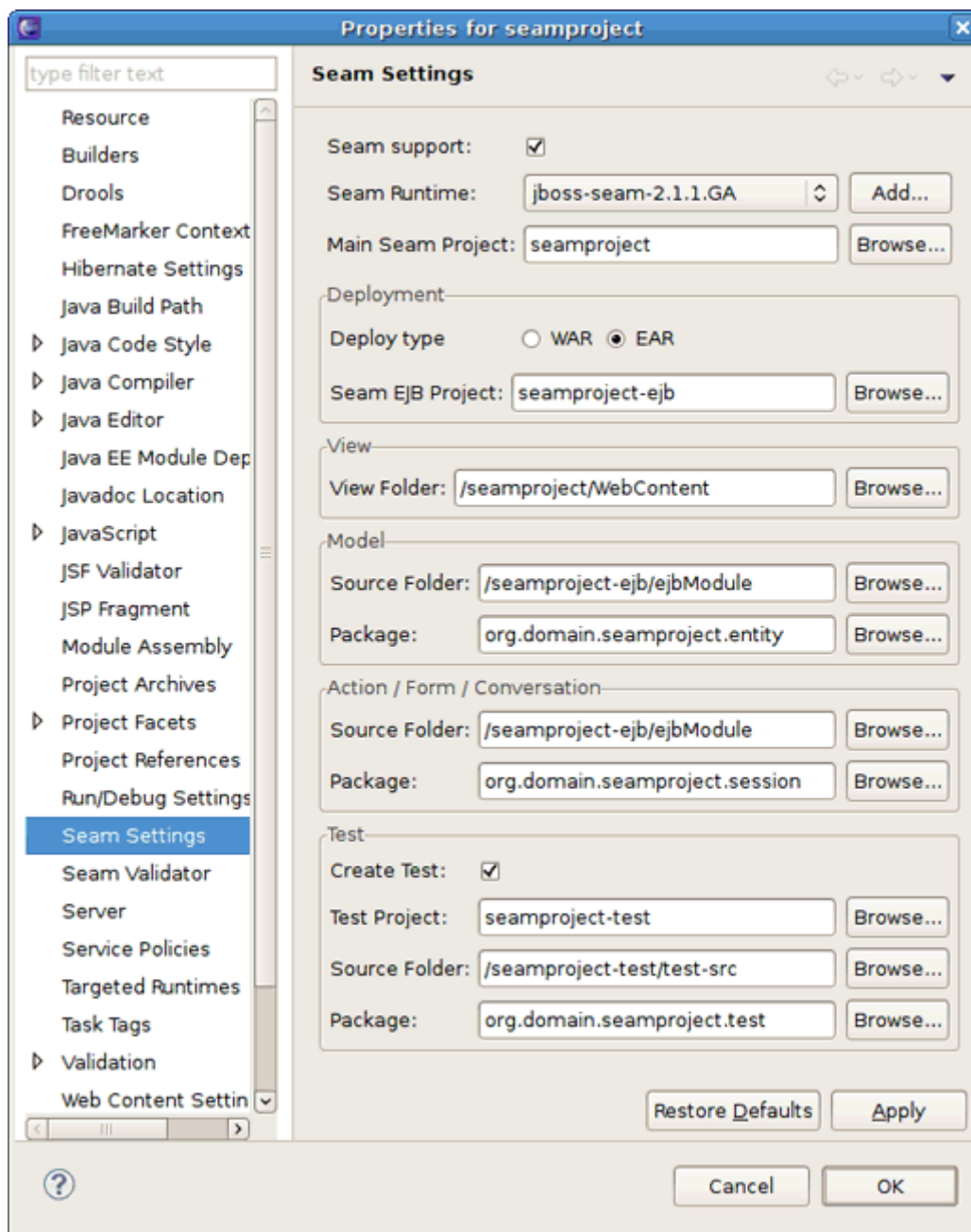


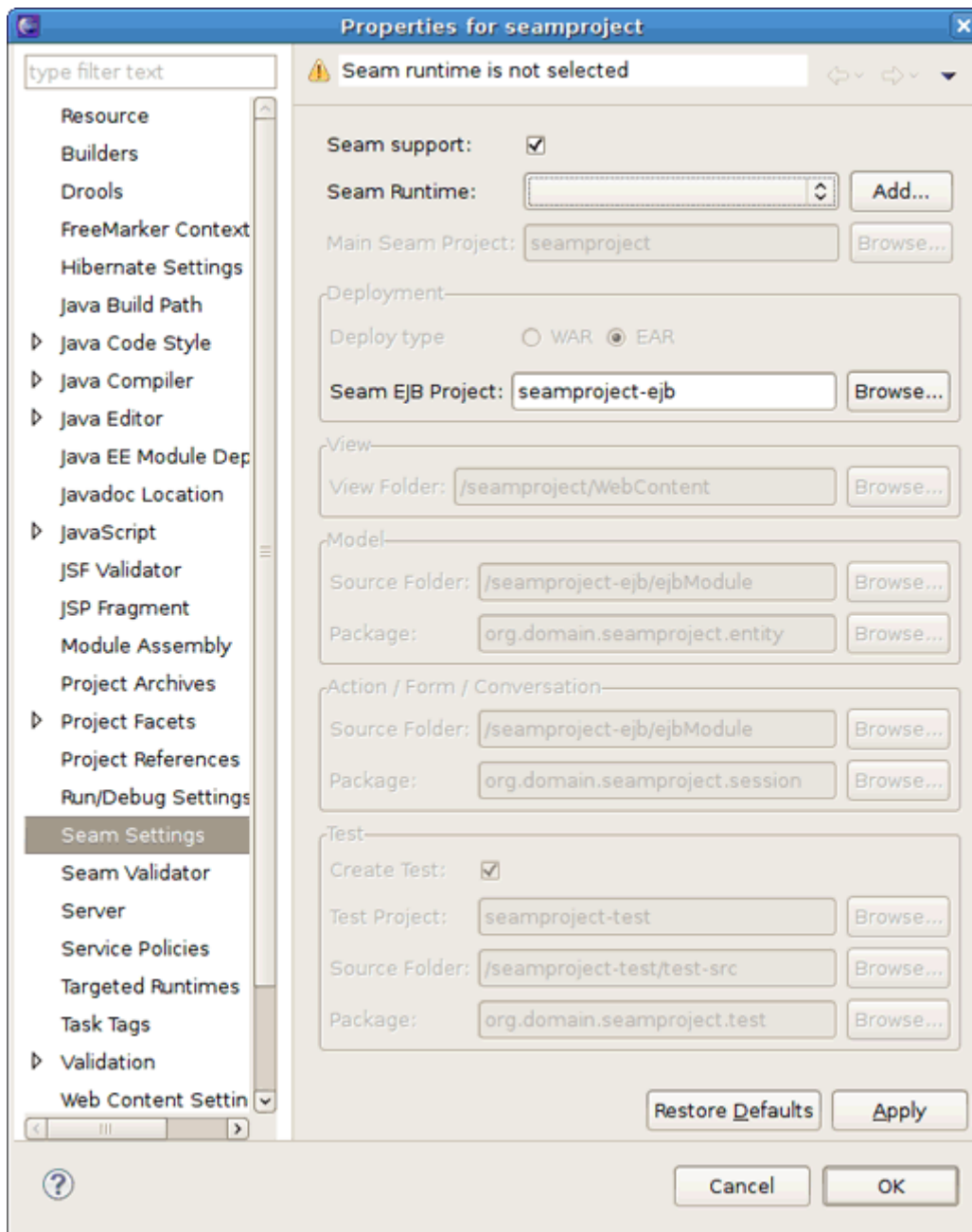
Figure 10.6. Properties for Seam Project



### Tip:

The wizard doesn't allow the renaming of the artifacts listed in the Seam Settings. It's possible to do in the Package Explorer. See [Section 4.5, “Renaming the Projects and Folders”](#).

Also notice, you are not required to use the New Seam Project wizard to benefit from Seam artifact wizards. You can just enable Seam on your existing project by checking Seam Support and targeting to Seam Runtime, and then configure the folders as you want.



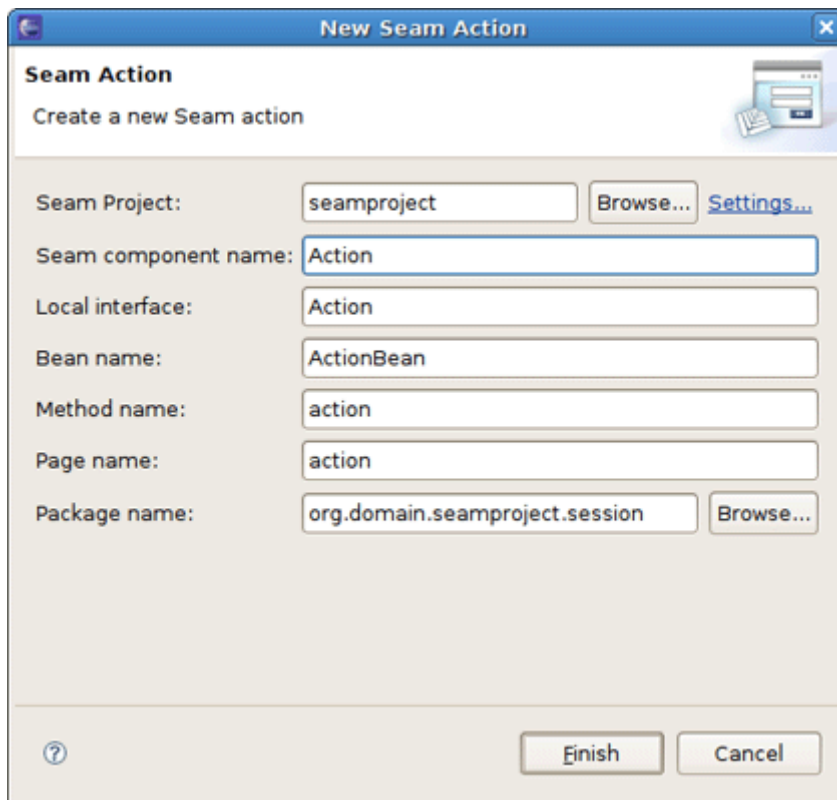
**Figure 10.7. Properties for Seam Project when Seam runtime is not selected**



**Tip:**

On the figure above fields for configuring Seam artifacts are unavailable because Seam Runtime is not selected.

In Seam Wizards (New Action, Form, Entity, Conversation, Generate Entities) you can get a quick access to project settings using the Settings link in the upper right corner of each wizard.



**Figure 10.8. Quick Setting Link In Seam Wizards**

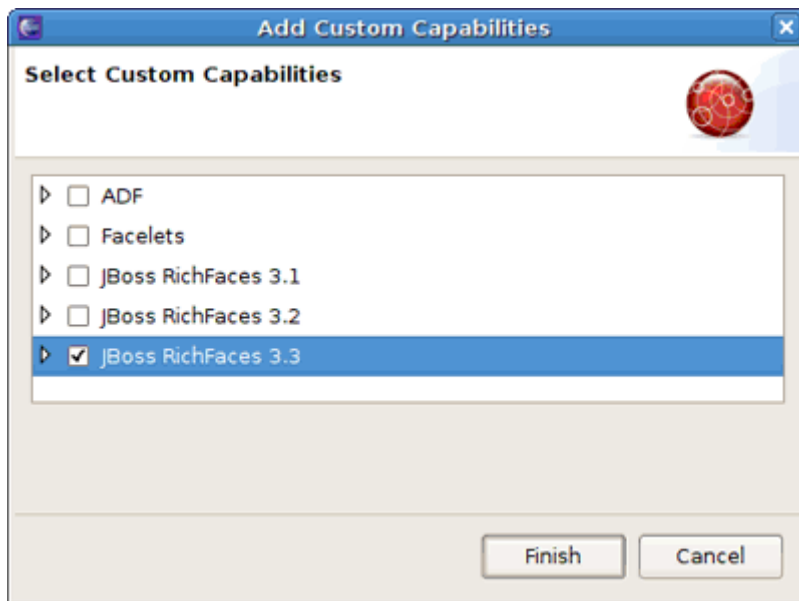
### 10.3. Adding Custom Capabilities

JBoss Tools provides a possibility to add custom capabilities to any Seam project, 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)

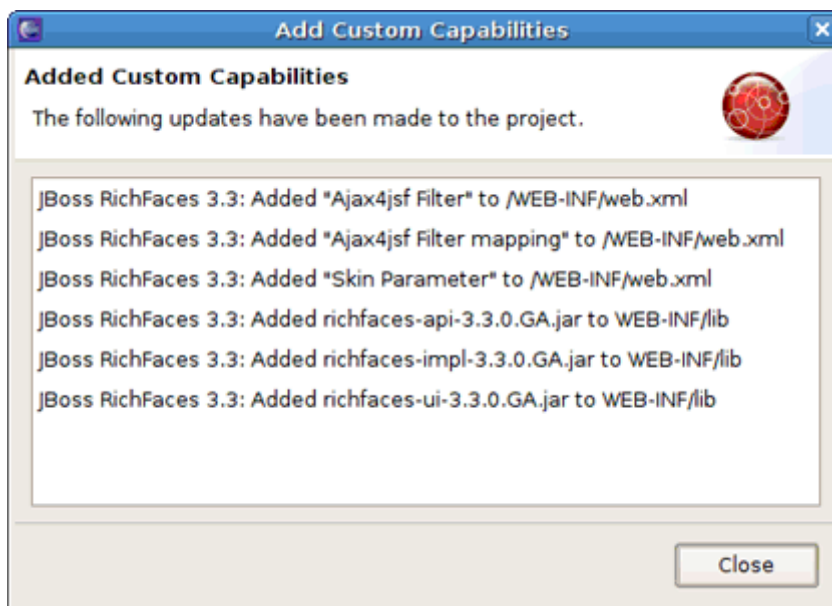
To enable it, you should call the context menu for a Seam project and select **Configure** → **Add Custom Capabilities**. Then check the needed modules and press *Finish*.





**Figure 10.9. Custom Capabilities be added to Seam Project**

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



**Figure 10.10. Adding Custom Capabilities to Seam Project**



# Seam Refactoring Tools

The goal of Seam components refactoring is to make system-wide code changes without affecting the behavior of the program. Seam Tools provides assistance with code refactoring.

Seam refactoring tools includes the following refactoring operations:

- [Section 11.2, “Renaming Seam Components”](#)
- [Section 11.3, “Renaming Seam Context Variables in EL”](#)

Refactoring commands are available from the context menus of several views and editors; for details, see [Section 11.1, “Using the Refactorinig Wizard”](#).



## Note:

Seam refactoring could be only performed in the file that is not marked as *Read only* in its properties.

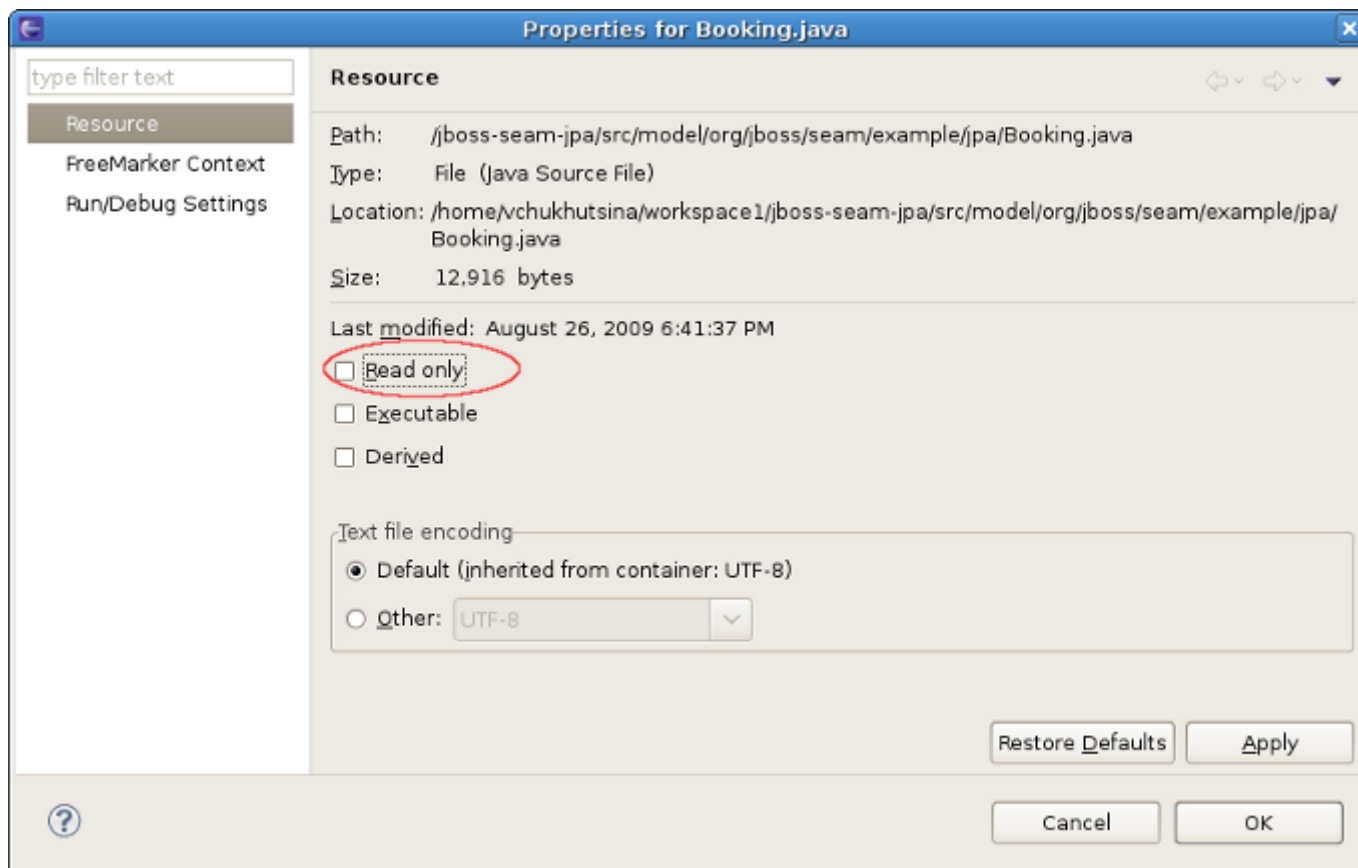
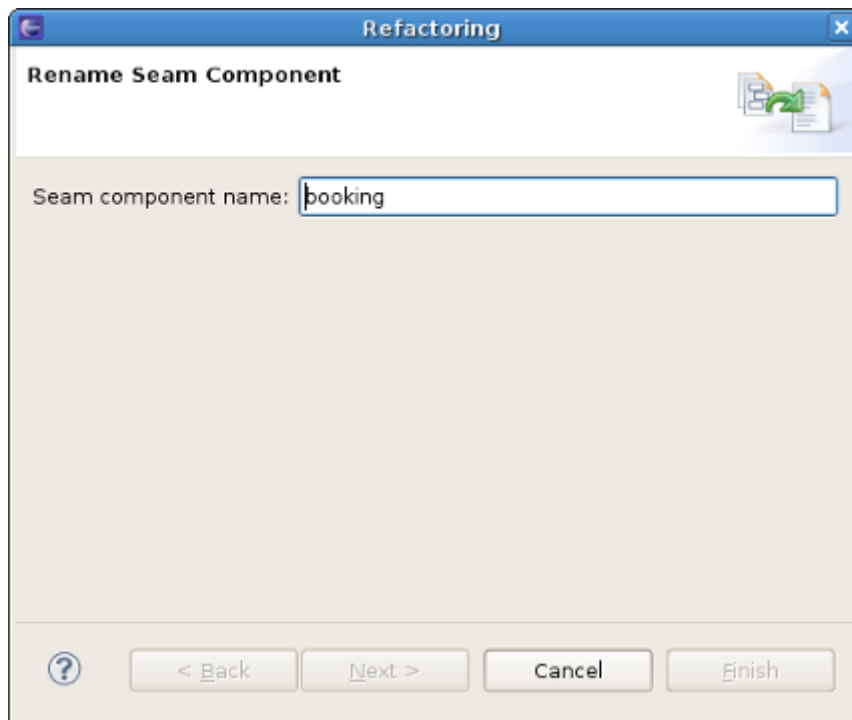


Figure 11.1. Properties View for Java File

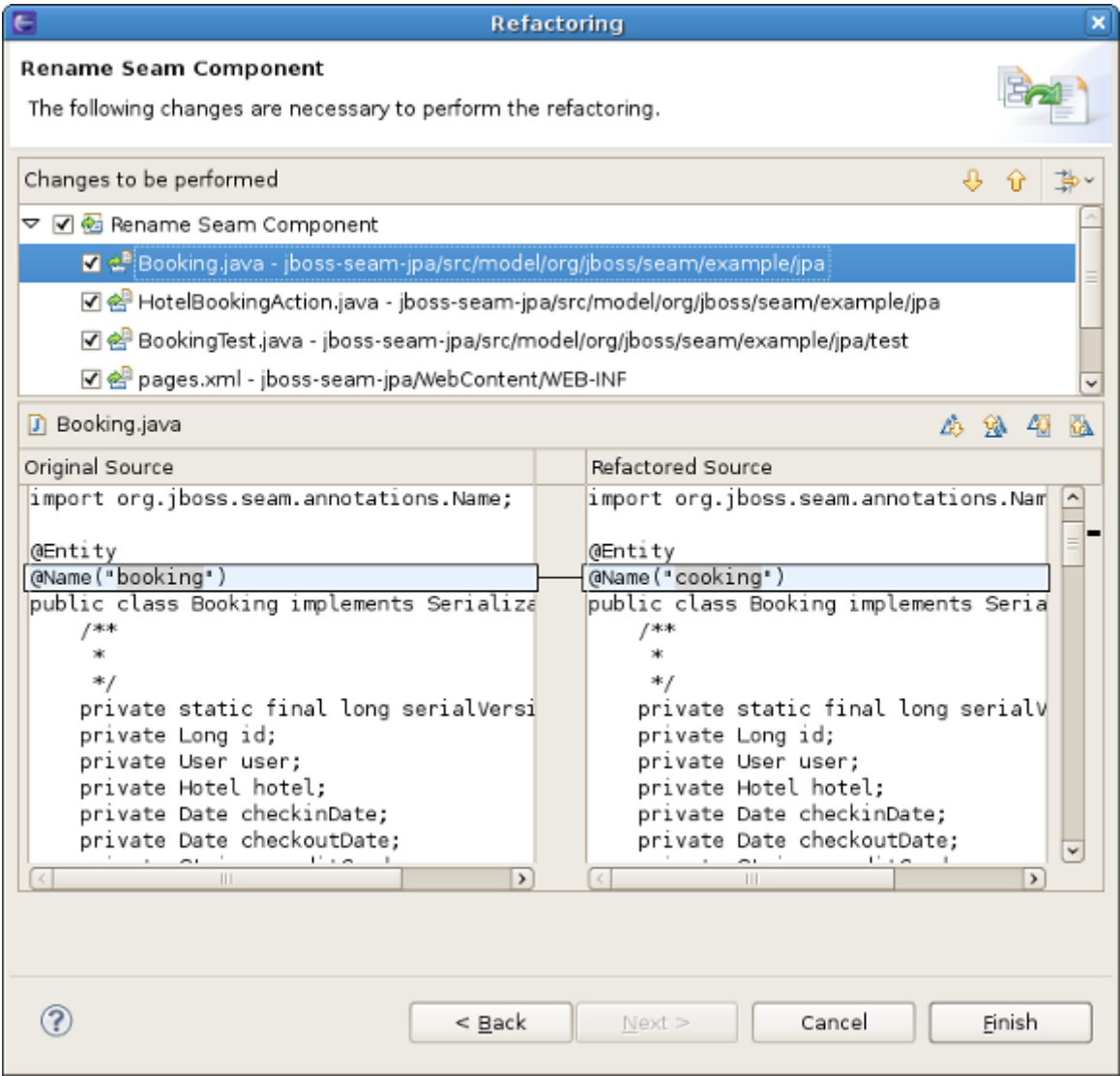
## 11.1. Using the Refactorinig Wizard

This wizard is opened always when the user clicks the *Rename Seam Component*, *Rename Seam Context Variable* or *Rename* command in any of the views/editors mentioned later in this chapter.



**Figure 11.2. Refactoring Wizard**

In the Seam Components name field you should enter a new name of the component, then click Next and follow the next wizard steps:



**Figure 11.3. Navigating Between the Differences**

As you see, when performing a refactoring operation, you can preview all of the changes resulting from a refactoring action before you choose to carry them out.

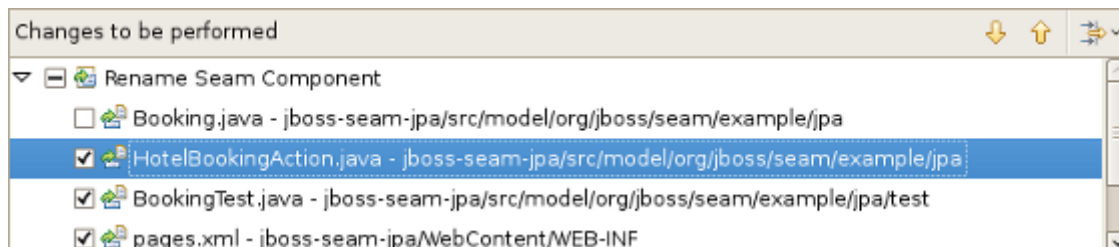
The view represents two windows: one with the original source code and the other with refactored one.

With the help of





buttons you can quickly navigate between the differences in the code. If you don't agree with some changes you can't undo them but you can remove the class from the list of classes that need refactoring.



**Figure 11.4.** List of classes that need refactoring

To apply the changes click *Finish*.

Renaming the selected seam component corrects all references to the component (also in other files in the project). Renaming a type doesn't allow to rename similarly named variables and methods.

## 11.2. Renaming Seam Components

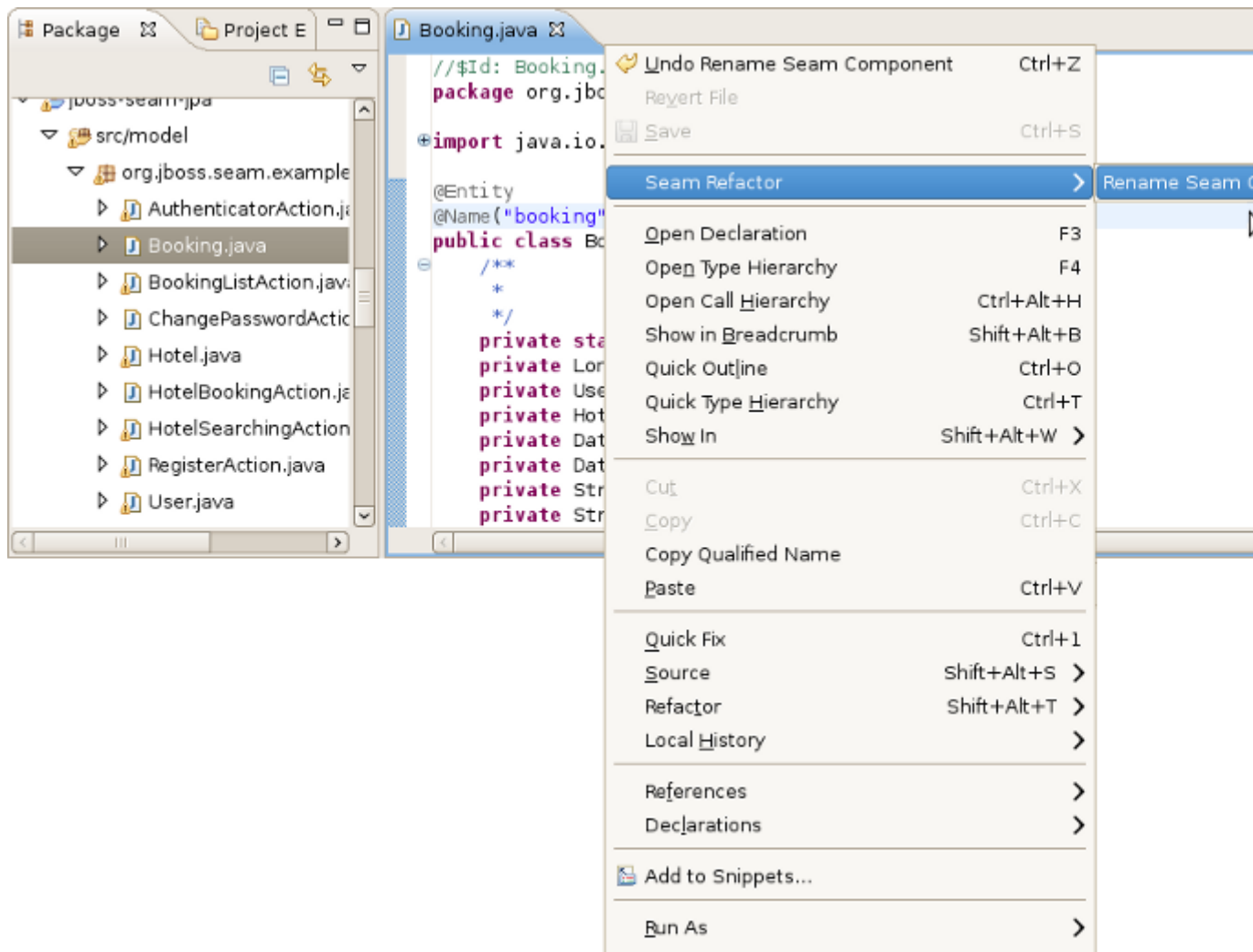
Renaming a Seam component could be performed in the:

- [Section 11.2.1, “Renaming Seam Components in the Java Editor”](#)
- [Section 11.2.2, “Renaming Seam Components in the Seam Components View”](#)
- [Section 11.2.3, “Renaming Seam Components in the Seam Components Editor”](#)

### 11.2.1. Renaming Seam Components in the Java Editor

If you want to rename certain seam component using Java editor, first of all it's necessary to open it. If you don't know, where the seam component is declared, use [Section 5.2.2, “Open Seam Component”](#) icon.

To rename a Seam component in the Java editor it's necessary to select the component, left click the file and then select **Seam Refactor** → **Rename Seam Component**.



**Figure 11.5. Opening Refactoring Wizard in Java Editor**

### 11.2.2. Renaming Seam Components in the Seam Components View

To open the Refactoring wizard in the Seam Components view you should left click the component you want to open and choose Rename Seam Component option.

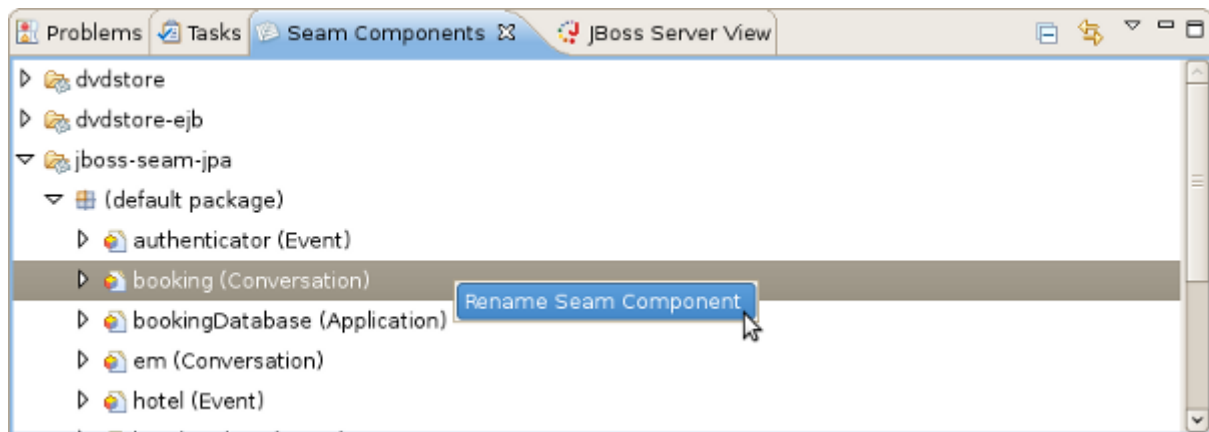


Figure 11.6. Opening Refactoring Wizard in Seam Components View

### 11.2.3. Renaming Seam Components in the Seam Components Editor

When you open `components.xml` file using JBoss Tools XML Editor, open the Refactoring wizard in the Tree tab by left-clicking the component in components area and choosing Rename option.

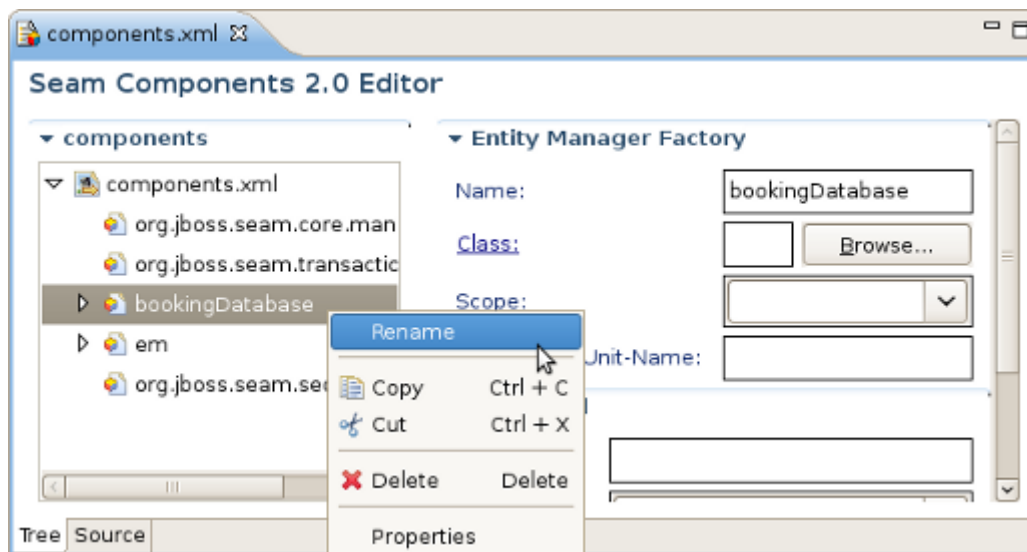


Figure 11.7. Opening Refactoring Wizard in *components.xml* File

## 11.3. Renaming Seam Context Variables in EL

Renaming a Seam context variables in EL could be executed in `.java`, `.xml`, `.jsp`, `.xhtml` and `.properties` files in Seam projects using the context menu command **Seam Refactor** → **Rename Seam Context Variable**.



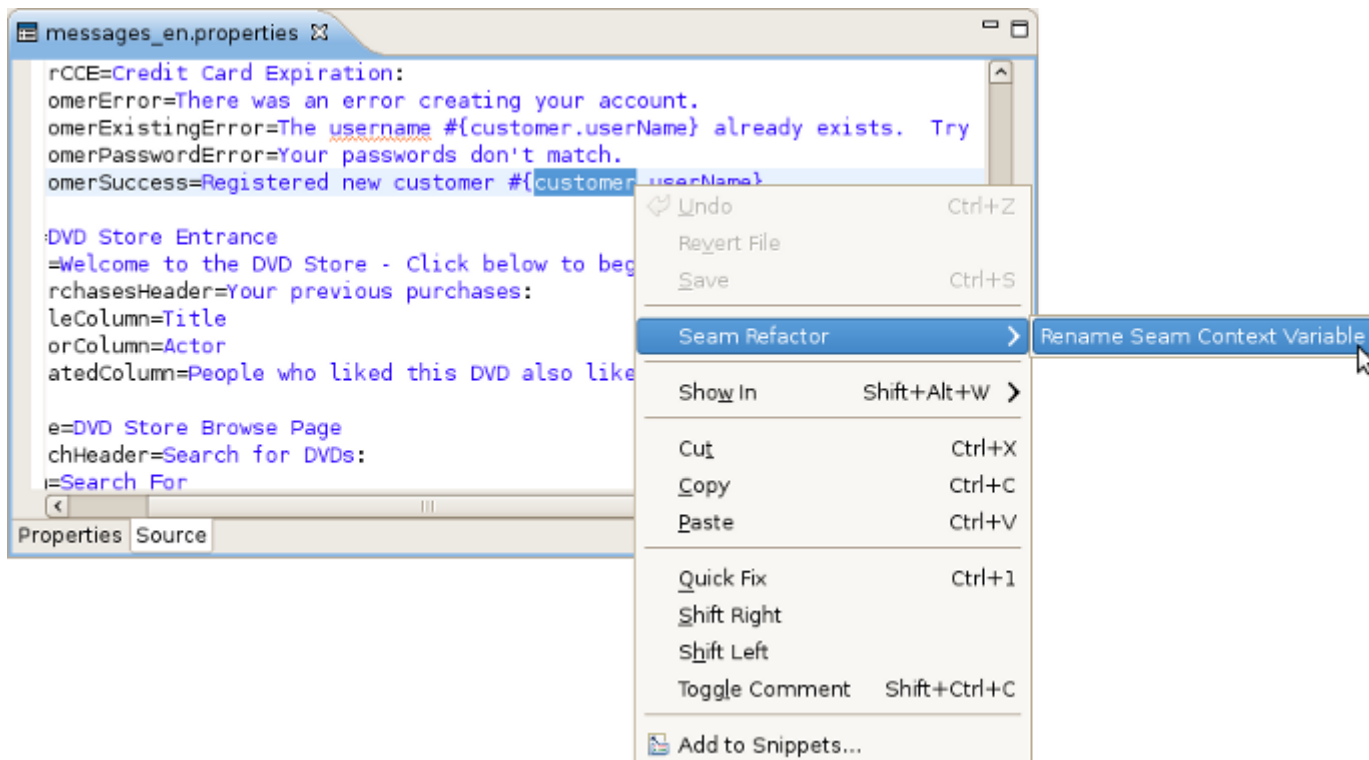


Figure 11.8. Refactoring Seam Context Variable in .properties File

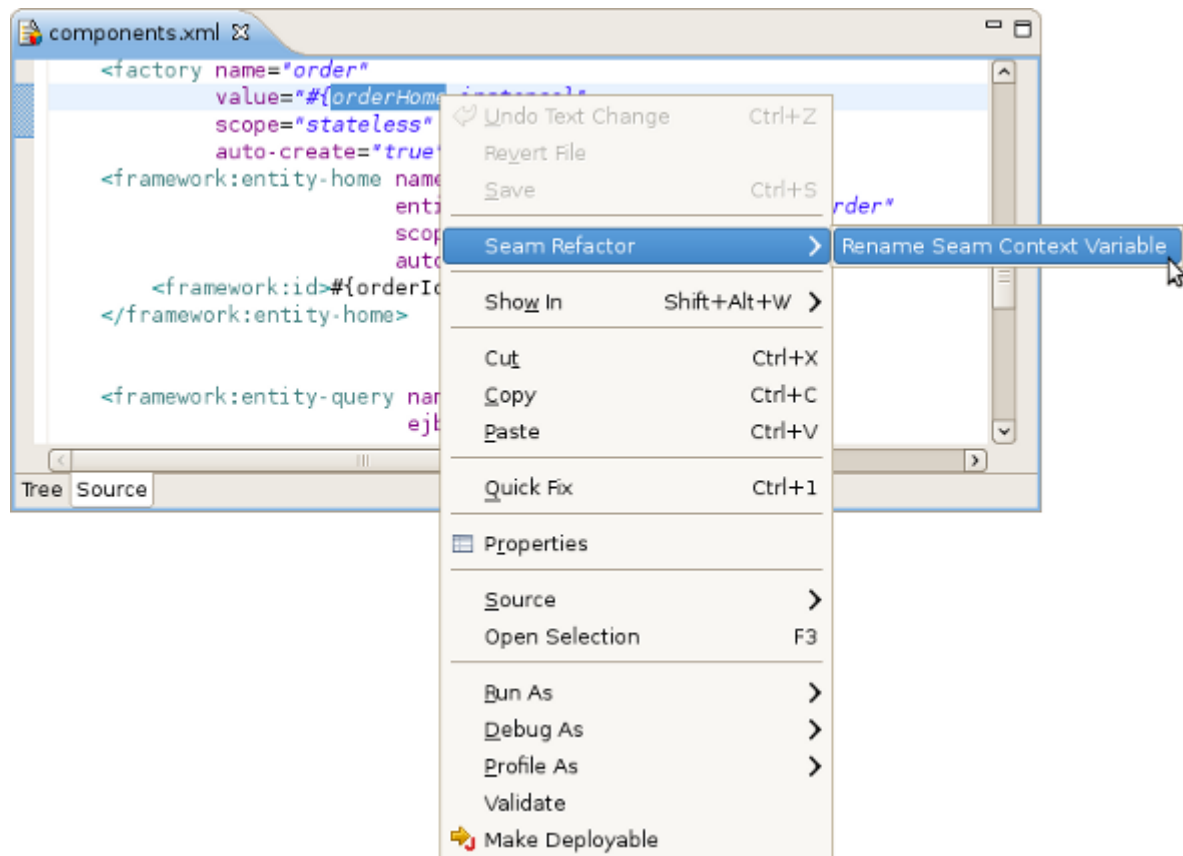


Figure 11.9. Opening Refactoring wizard in components.xml file

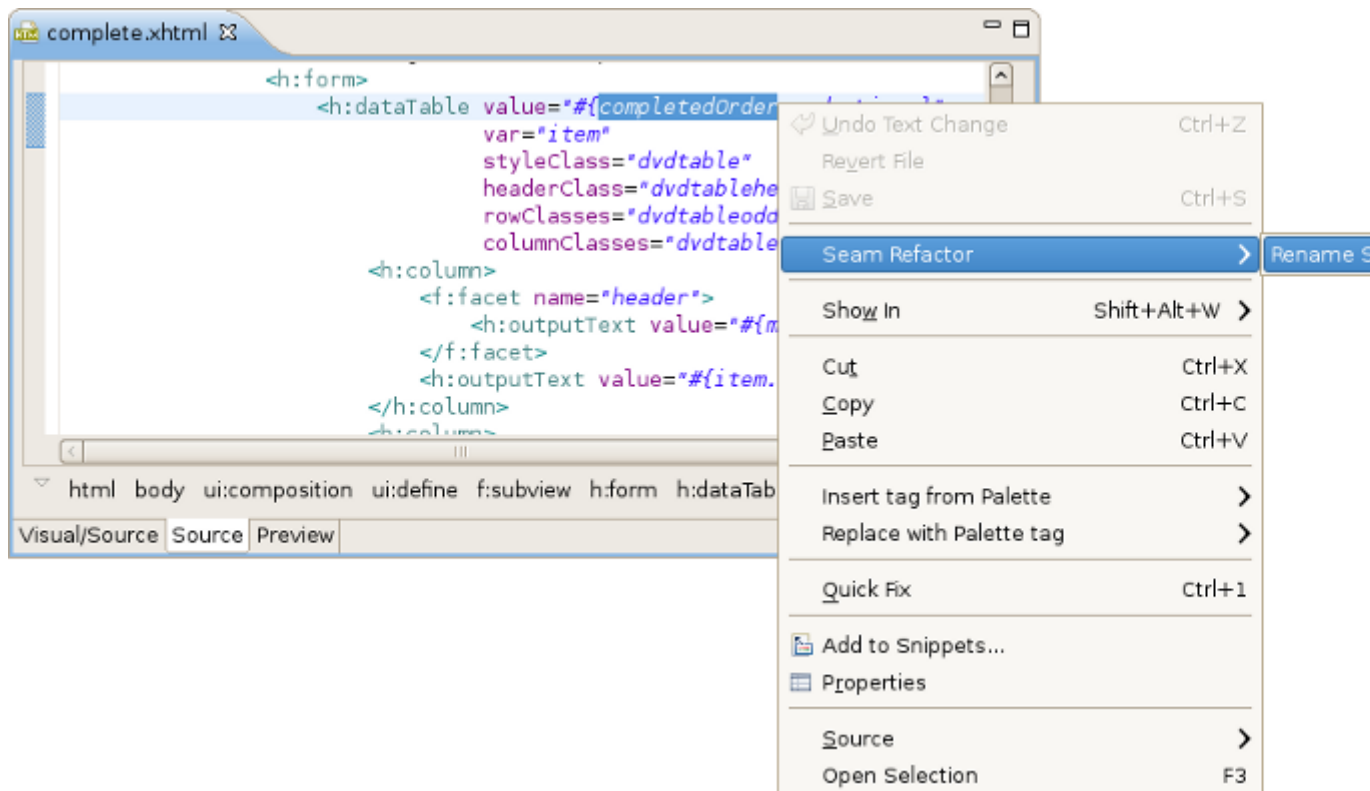


Figure 11.10. Opening Refactoring wizard in .html file



## Adding Seam support to EAR project

From this chapter you will find out how to add Seam support to EAR project

For example you have several WTP projects:

- seamproject-ear (You can create WTP EAR project using **New** → **Project** → **Java EE** → **Enterprise Application Project wizard**)
- seamproject-ejb (You can create WTP EJB project using **New** → **Project** → **EJB** → **EJB Project wizard**)
- seamproject-war (You can create WTP WEB project using **New** → **Project** → **Web** → **Dynamic Web Project wizard**)



### Tip

You can add as many EJBs modules as you want just doing the same for each EJB project.

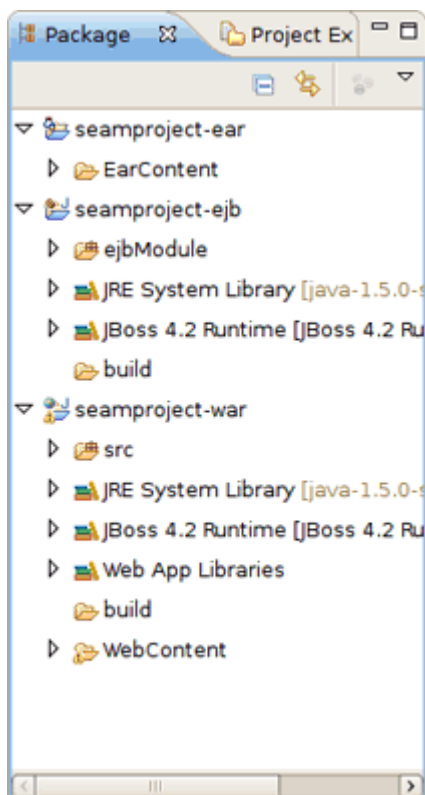


Figure 12.1. WTP Projects

To enable Seam support for an existing EAR, WAR or EJB project, right-click on the project in the **Project Explorer** to bring up the context menu and navigate to **Configure** → **Enable Seam Support**.

Copy required JAR files manually into the project. For EAR projects specifically, copy library folders into the projects `/lib` folder and they will be automatically detected.

# Generate a CRUD Database Application

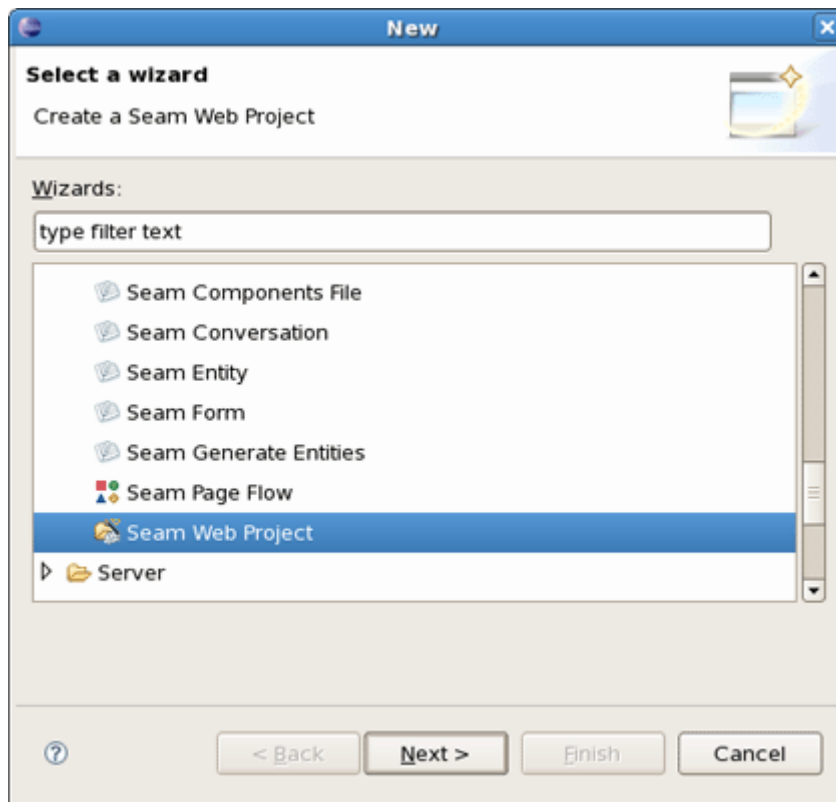
## 13.1. What is CRUD?

CRUD is an acronym for the four basic types of SQL commands: *Create* , *Read* , *Update* , *Delete* . Most applications have some kind of CRUD functionality, and we can assume that every programmer had to deal with CRUD at some point. A CRUD application is one that uses forms to get data into and out of a database.

In the next section we will create a Seam Web application, connect it to the [HSQL](http://www.hsqldb.org/) database and add CRUD support.

## 13.2. How to create the CRUD Database Application with Seam

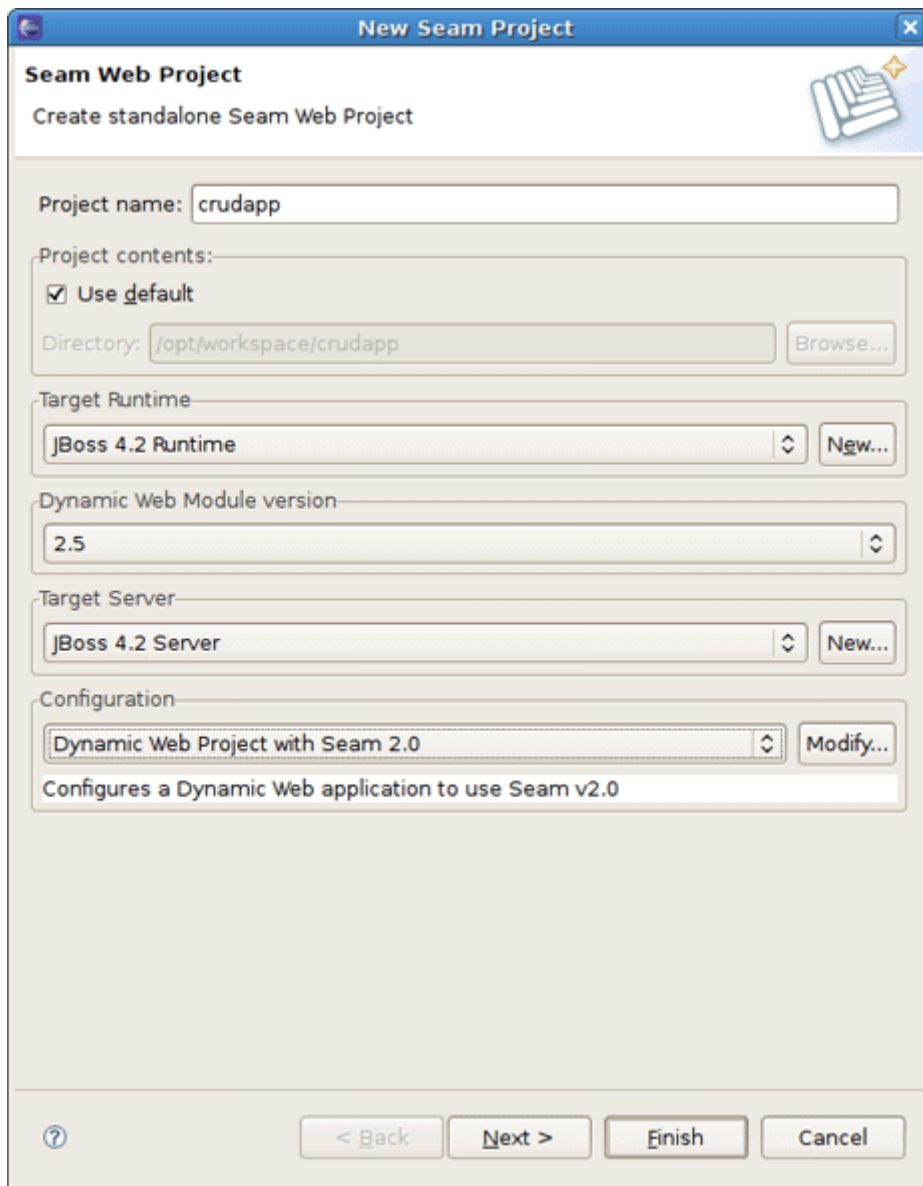
- First, you should [download](http://docs.jboss.org/tools/resources/) the `employee-hsqldb` database, extract the `employee-hsqldb.zip` and start it by running `./runDBServer.sh` or `runDBServer.bat` from the database directory.
- Create a new Seam Web Project using New Seam Project wizard or main menu **File** → **New** → **Seam Web Project**.



**Figure 13.1. Seam Web Project Creation**

- Name your project as *crudapp*, specify Target Runtime, Server and Seam Runtime configuration. Then press *Next* and follow the next wizard steps keeping default settings.






**Figure 13.2. New Seam Project Wizard**



**Tip:**

Please have a look at [Chapter 2, Creating a New Seam Project via the New Seam Project wizard](#) how to create Target Runtime and Seam Runtime in order to get started creating, running, and debugging J2EE applications.

- On *Seam Facet* page click *New...* next to the Connection profile section to create a new Connection profile.

**Seam Facet**

Configure Seam Facet Settings

**General**

Seam Runtime:

jboss-seam-2.3.0

⬆⬇⬆

Add...

Deploy as:

☒ WAR ☐ EAR

EJB project name:

crudapp-ejb

EAR project name:

crudapp-ear

Libraries:

Copy Libraries From Seam Runtime

⬆⬇⬆

**Database**

Database Type:

HSQL

⬆⬇⬆

Connection profile:

BIRT Classic Model

⬆⬇⬆

Edit...

New...

Database Schema Name:

Database Catalog Name:

DB Tables already exists in database:

☐

Recreate database tables and data on deploy:

☐

**Code Generation**

Session Bean Package Name:


org.domain.crudapp.session

Entity Bean Package Name:

org.domain.crudapp.entity

Create Test Project:

☐



< Back

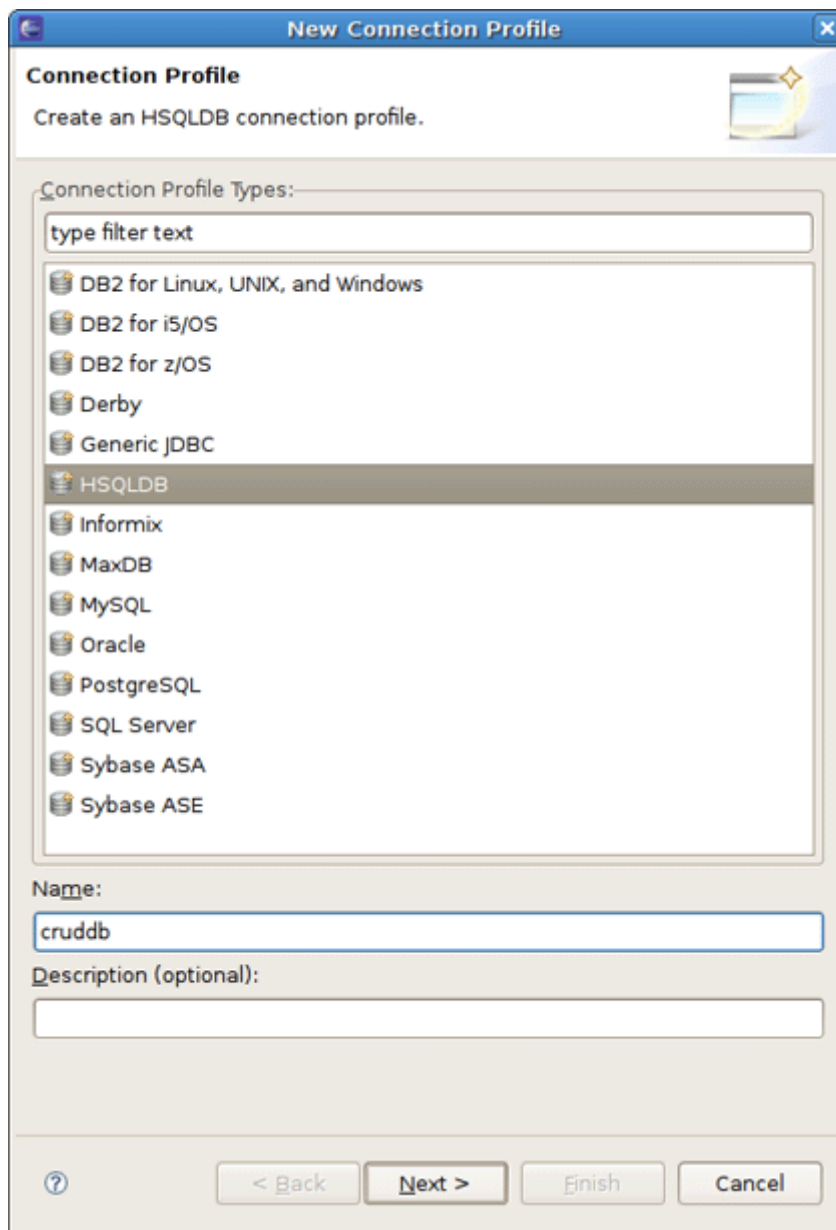
Next >

Cancel

Finish

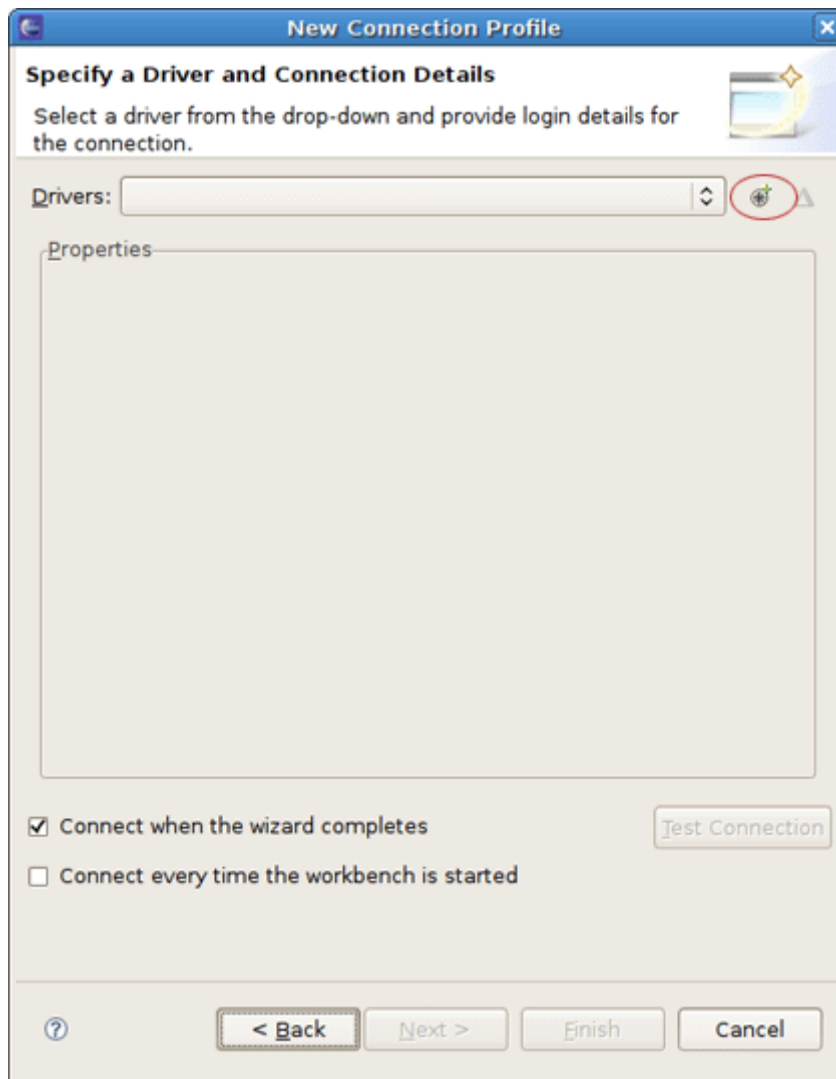
Figure 13.3. Seam Facet page

- On New Connection Profile dialog select the *HSQLDB* connection profile type, name it *cruidb* and click *Next*.



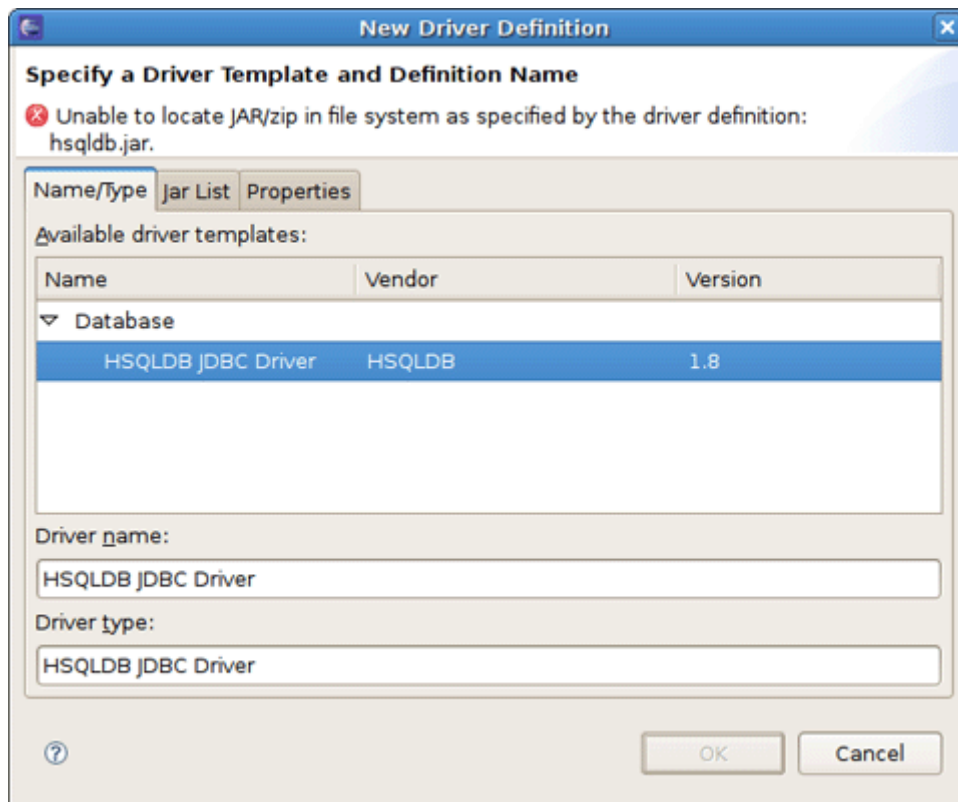
**Figure 13.4. New JDBC Connection Profile**

- On the next page click the round icon next to the *Drivers* field to select a database driver.



**Figure 13.5. New JDBC Connection Profile Database Driver**

- Now you should see the New Driver Definition dialog. On the first tab select the *HSQldb JDBC Driver*. Underneath in the *Driver name* field you can change its name if you need.



**Figure 13.6. Database Driver Type**

- You may notice the note on the previous figure. It prompts that you should specify the driver of the type you pointed. Set the location of the driver by selecting `hsqldb.jar` from the list and clicking the **Edit JAR/ZIP** button.

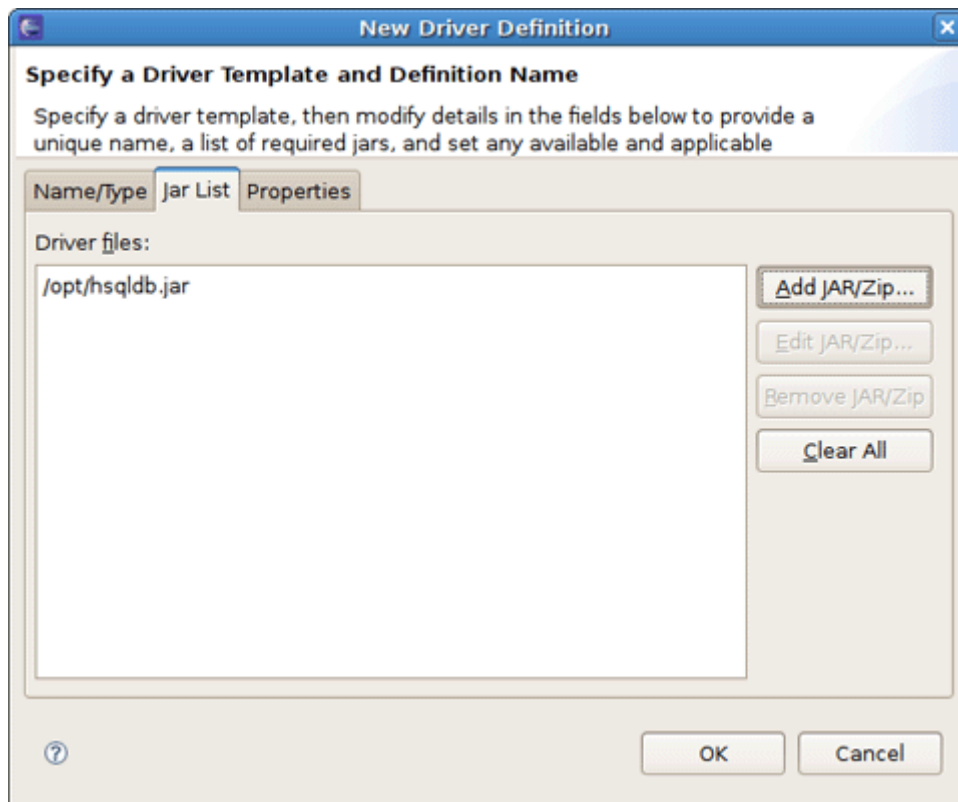
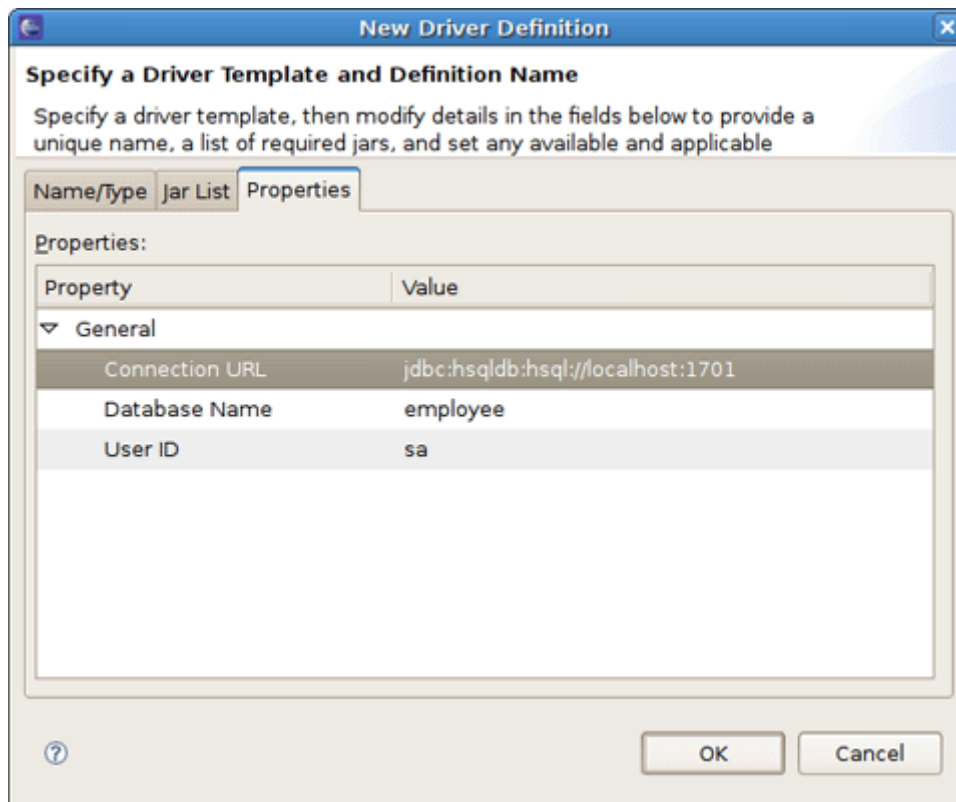


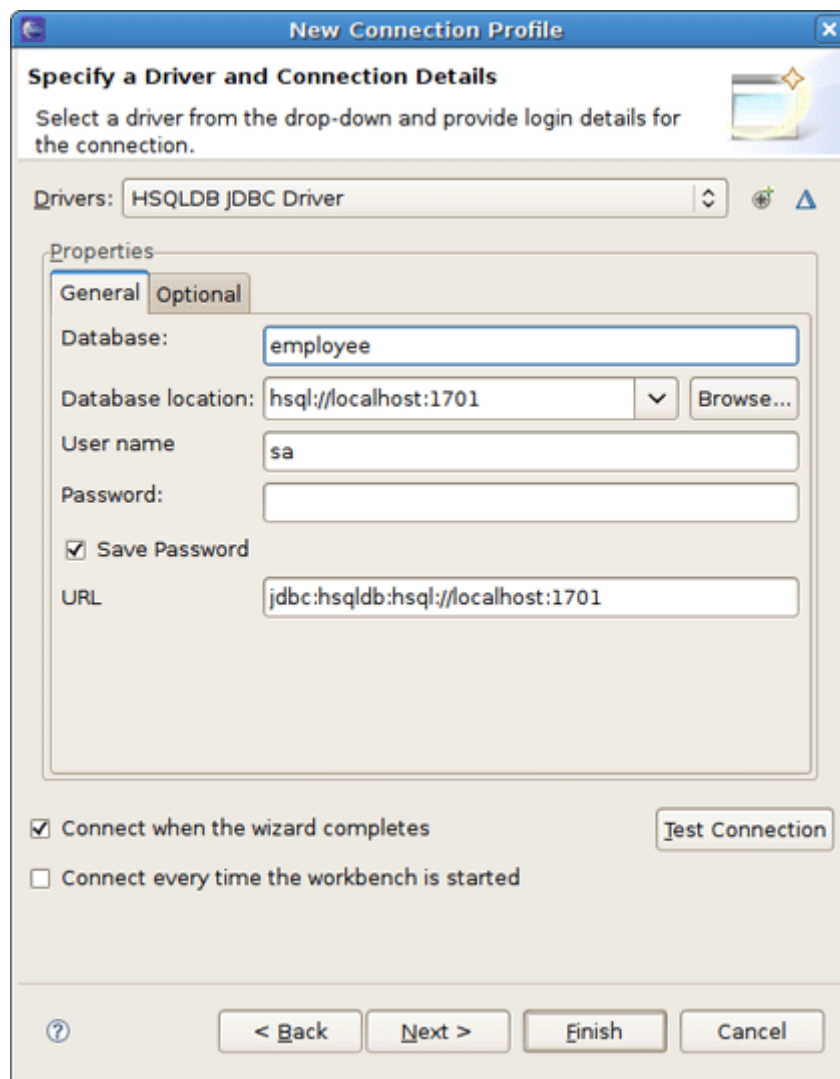
Figure 13.7. Driver Definition

- On the **Properties** tab set the **Connection URL** to *jdbc:hsqldb:hsqldb://localhost:1701*, **Database Name** to *employee* and **User ID** to *sa* and click **OK**.



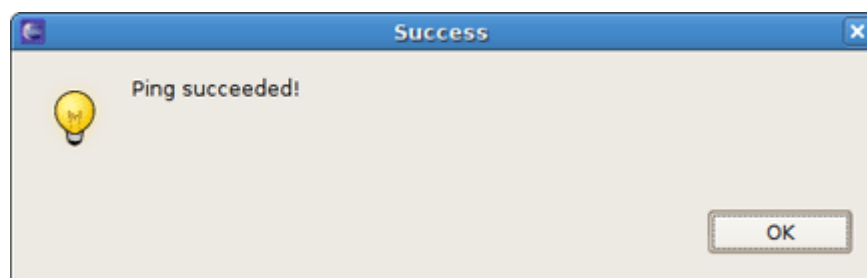
**Figure 13.8. Driver Definitions Properties**

- After clicking **OK** to submit the newly created driver you can observe and if you need edit all specified connection details.



**Figure 13.9. Driver and Connection Details**

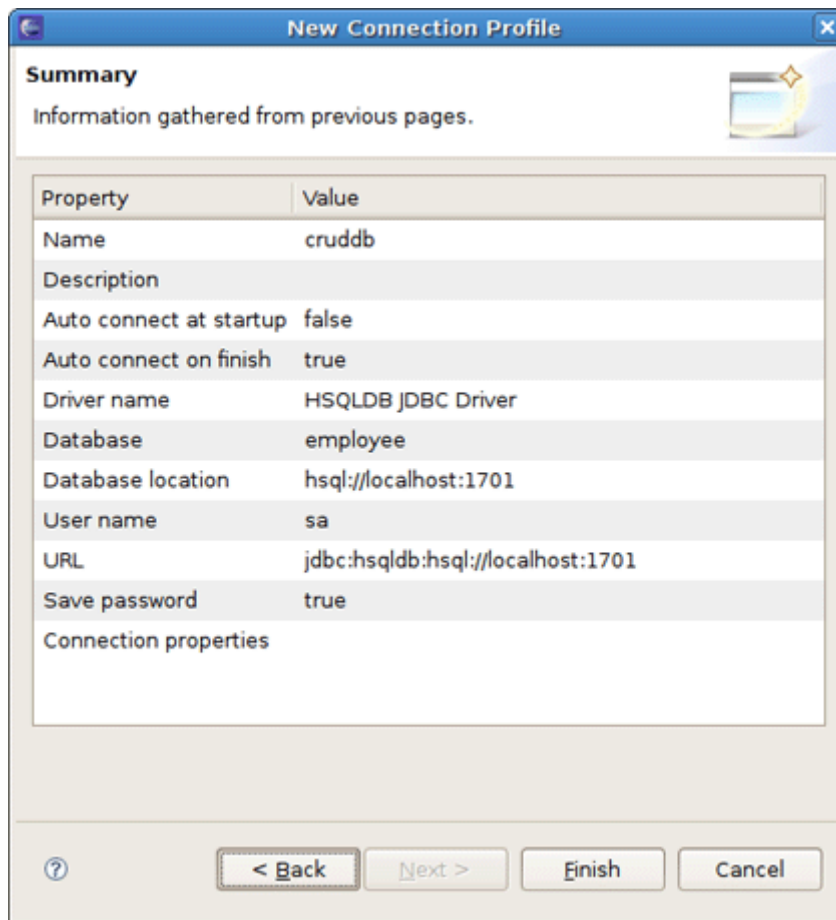
- Now click *Test Connection* to be sure that connection can be established.



**Figure 13.10. JDBC Connection is OK**

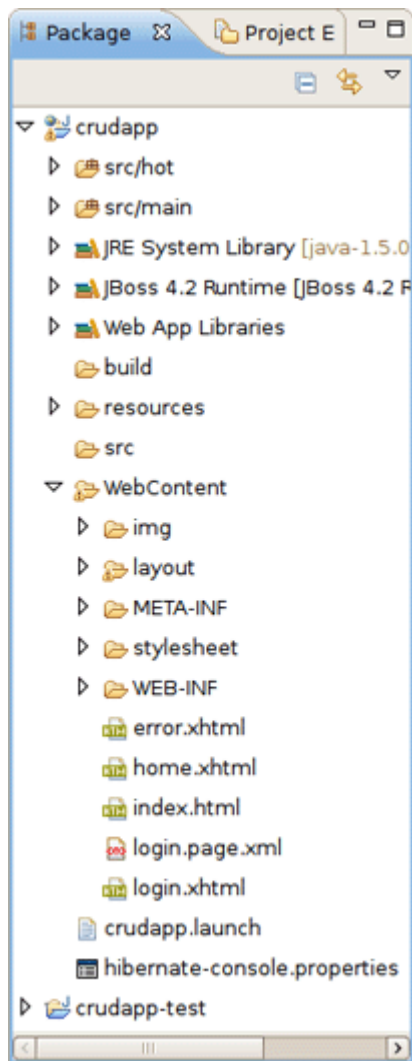
- Validate *JDBC Connection profile* settings and press *Finish* or *Back* if something is wrong.





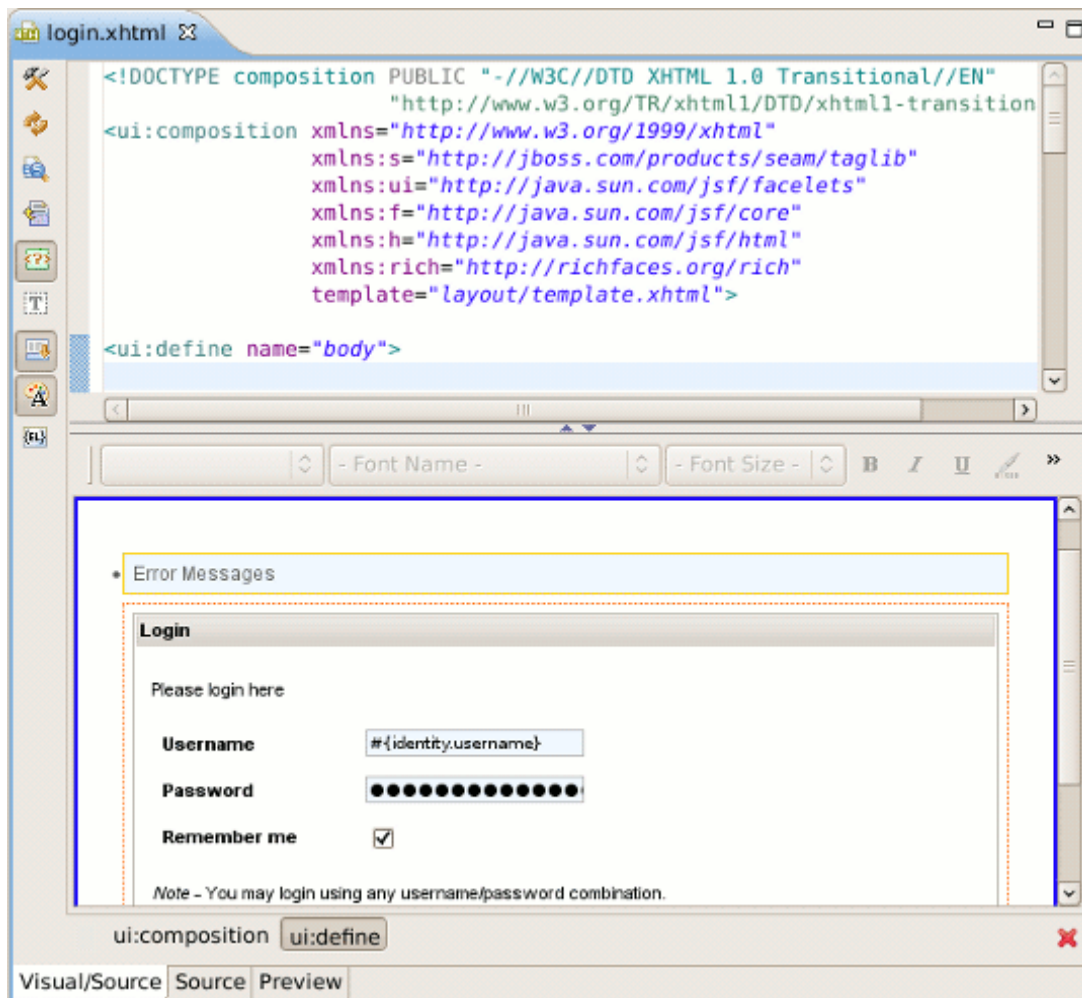
**Figure 13.11. Validate JDBC Connection settings**

- After clicking *Finish* two projects *crudapp* and *crudapp-test* will be created.



**Figure 13.12. CRUDAPP Seam Project**

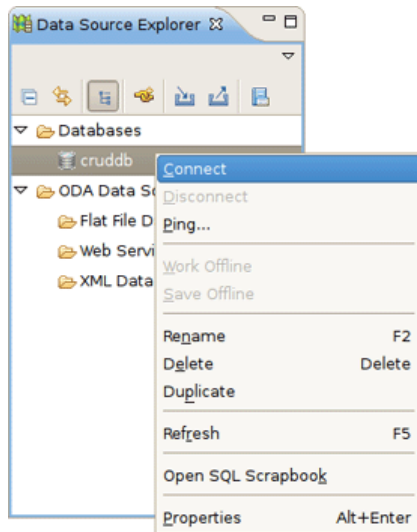
- You can expand the WebContent folder of the crudapp project in the **Project Explorer** and open home.xhtml or login.xhtml with JBoss Visual Editor.



**Figure 13.13. login.xhtml in VPE**

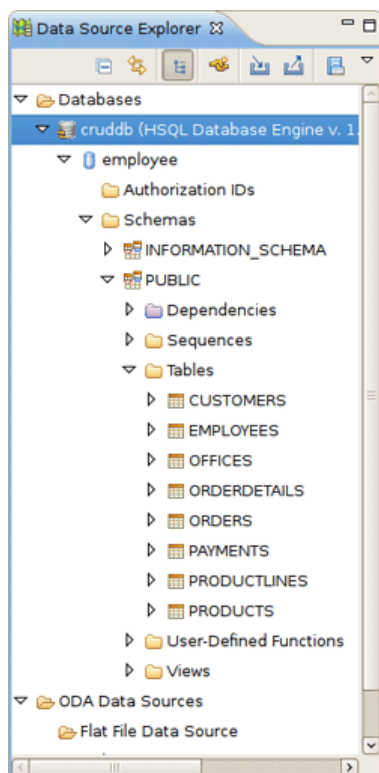
- Switch to **Database Development** perspective with **Window** → **Open Perspective** → **Other**.

Connect to the *cruidb* database if not already connected to it.



**Figure 13.14. Connecting to the CRUDB database**

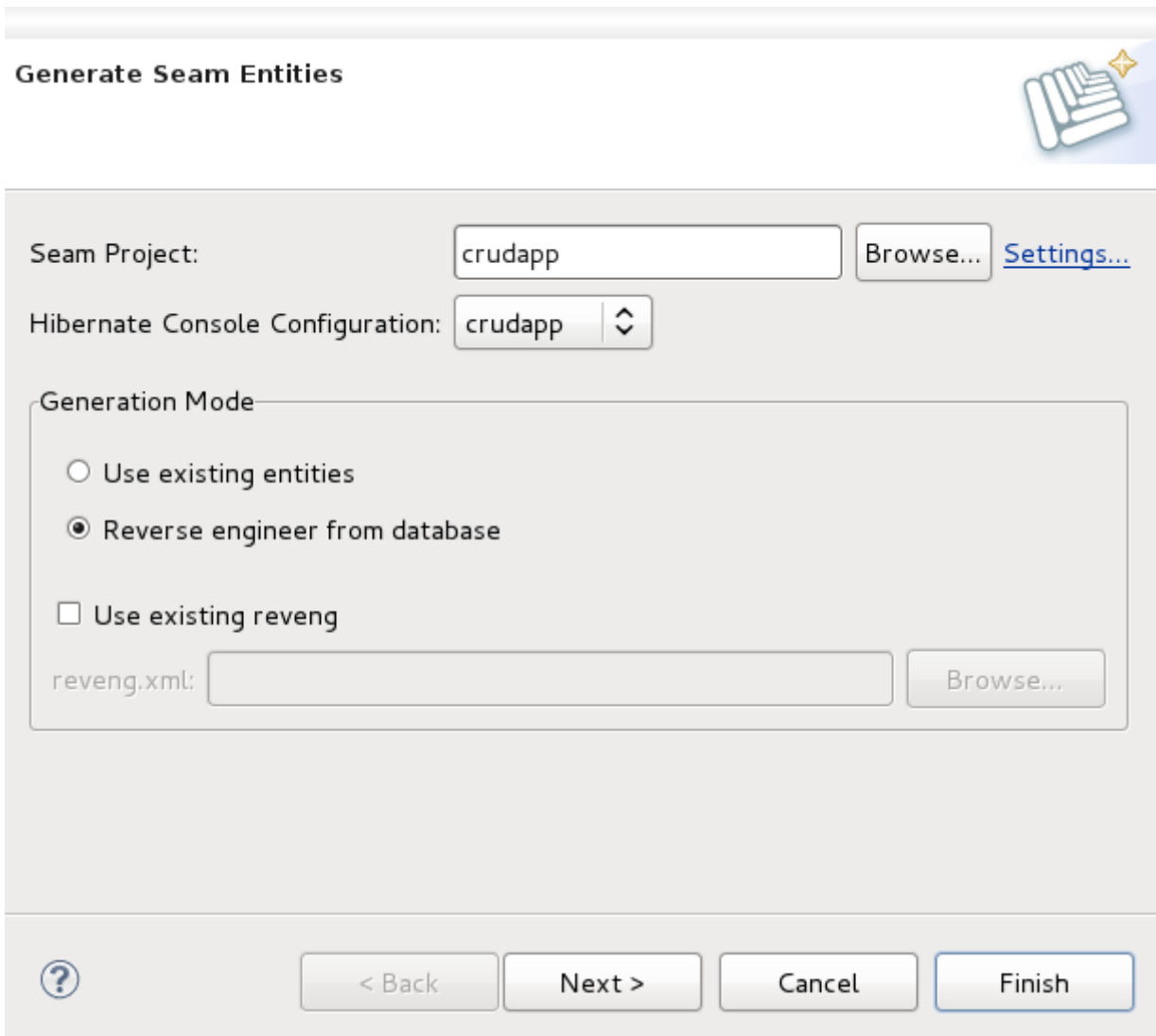
- Expand *cruddb* nodes to view its *Schemas*, *Tables*, *Columns* etc.



**Figure 13.15. CRUDB Database**


- Switch back to the Seam perspective. From the toolbar select **New** → **Seam Generate Entities** to create a set of CRUD Seam components and web pages based on existing tables in the

database. On the first page of *Generate Seam Entities* wizard, select **Reverse engineer from database** click press **Next**.



**Generate Seam Entities**

Seam Project:  [Browse...](#) [Settings...](#)

Hibernate Console Configuration:  


Generation Mode

☐ Use existing entities

☒ Reverse engineer from database

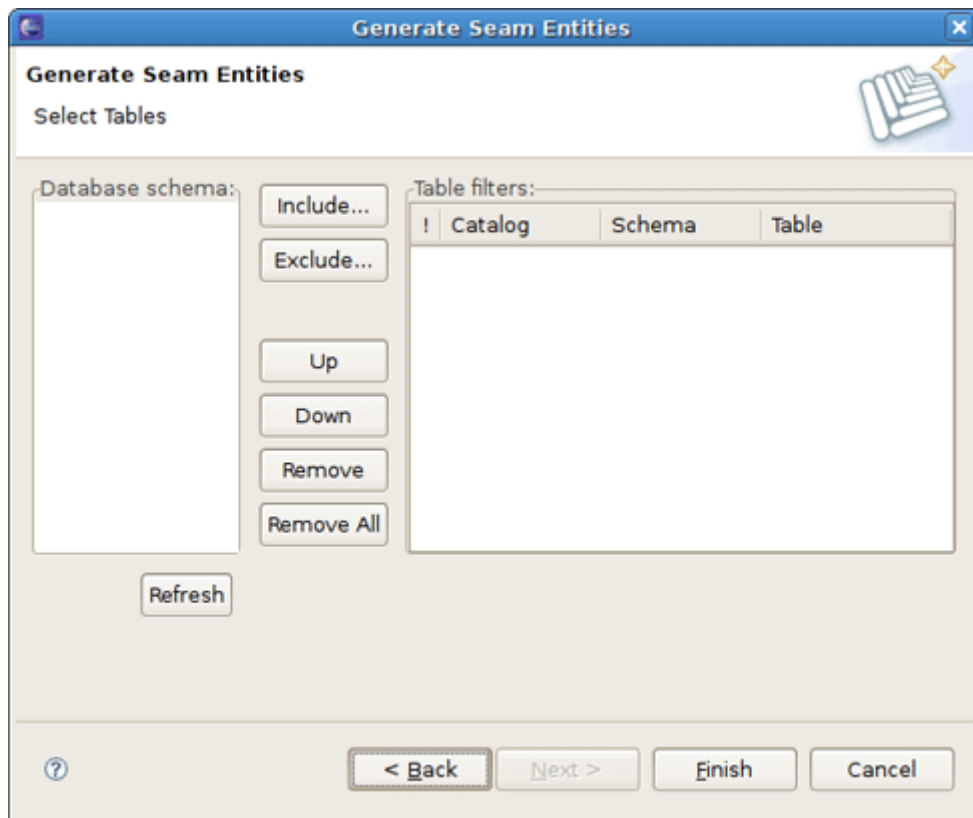
☐ Use existing reveng

reveng.xml:  [Browse...](#)

 [< Back](#) [Next >](#) [Cancel](#) [Finish](#)

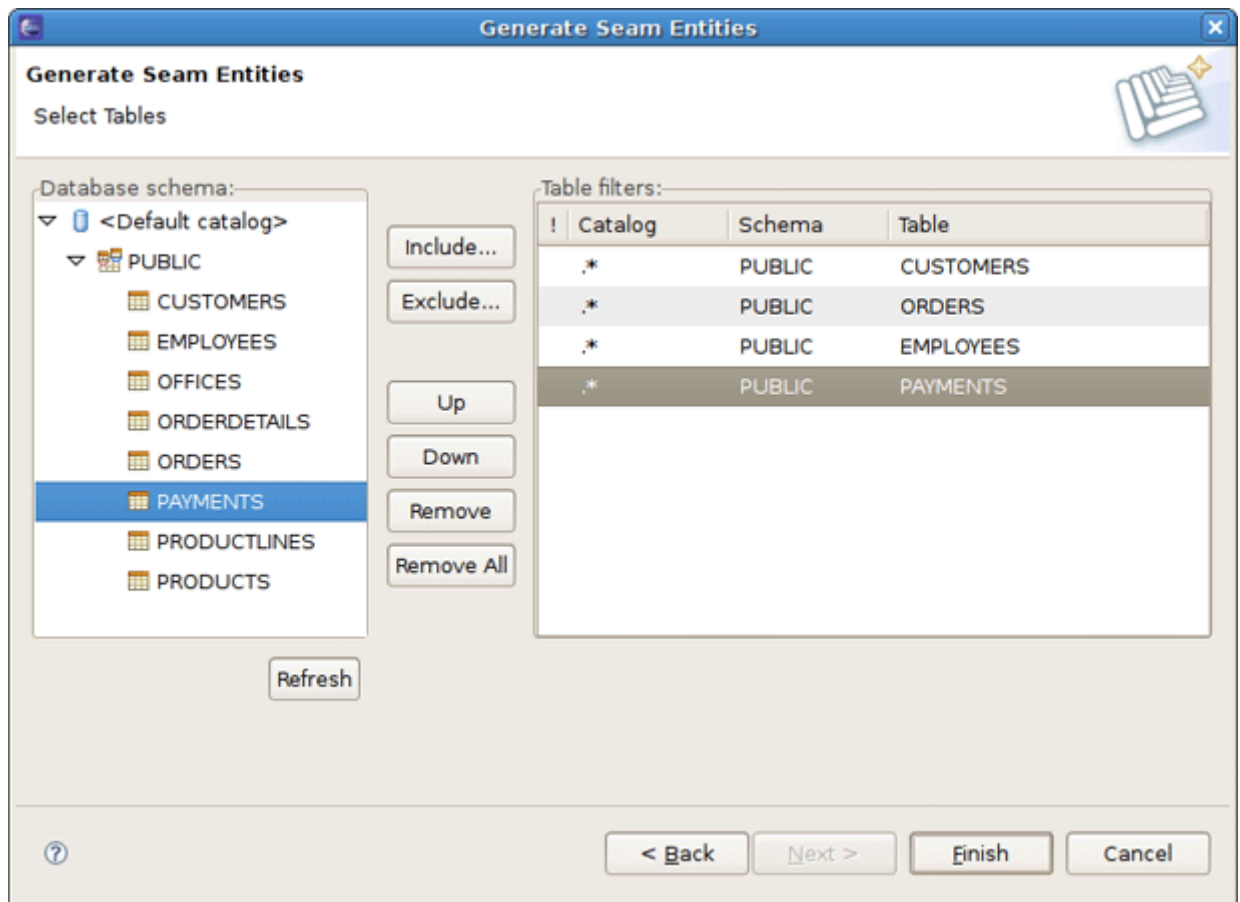
**Figure 13.16. Generate Seam Entities**

- Click *Refresh* to display all the tables from the database.



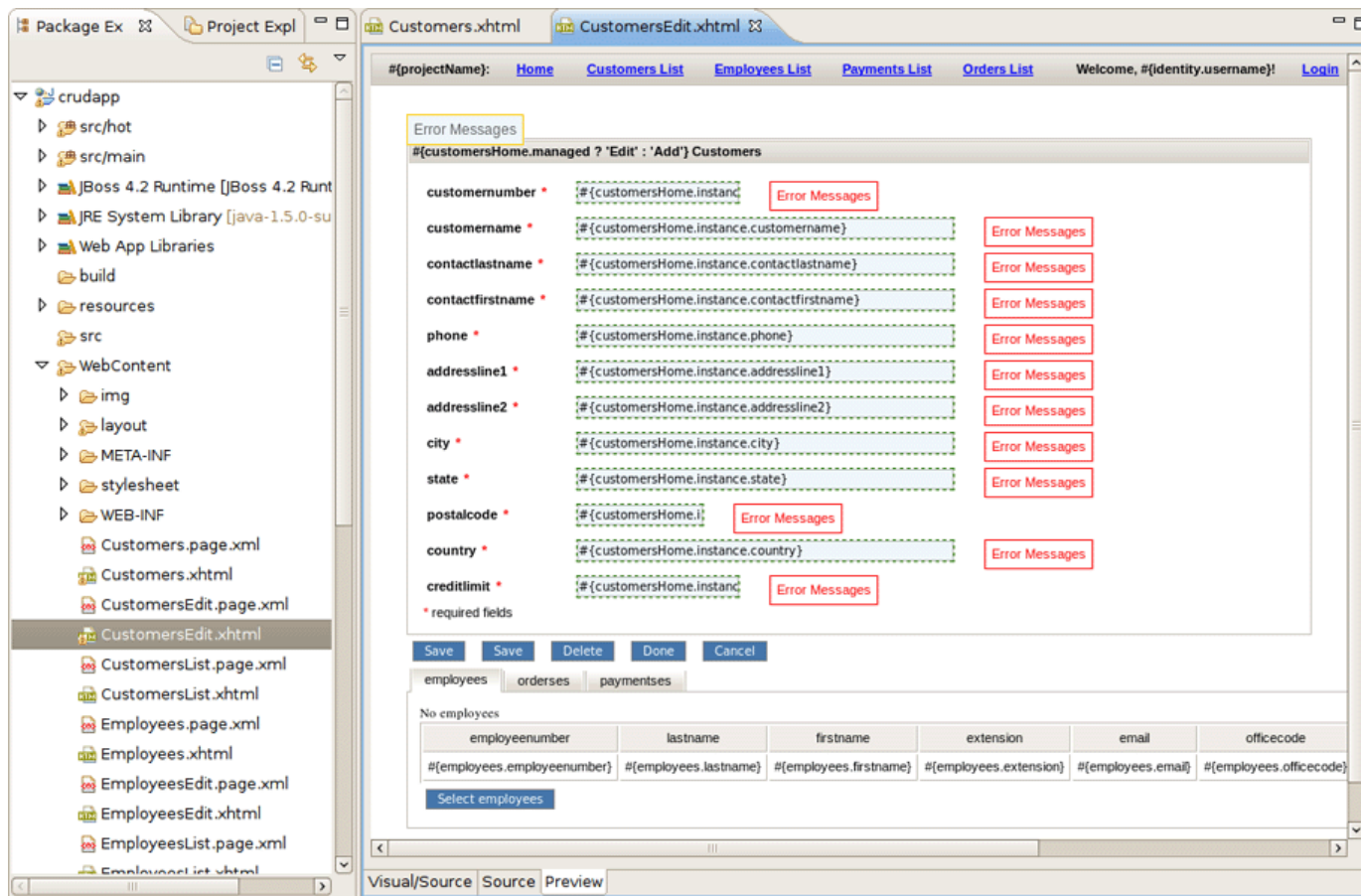
**Figure 13.17. Dialog for Selecting Tables**

- Now you can filter the tables. Use the **Include** button to include those that are necessary and click **Finish**.



**Figure 13.18. Selecting Tables**

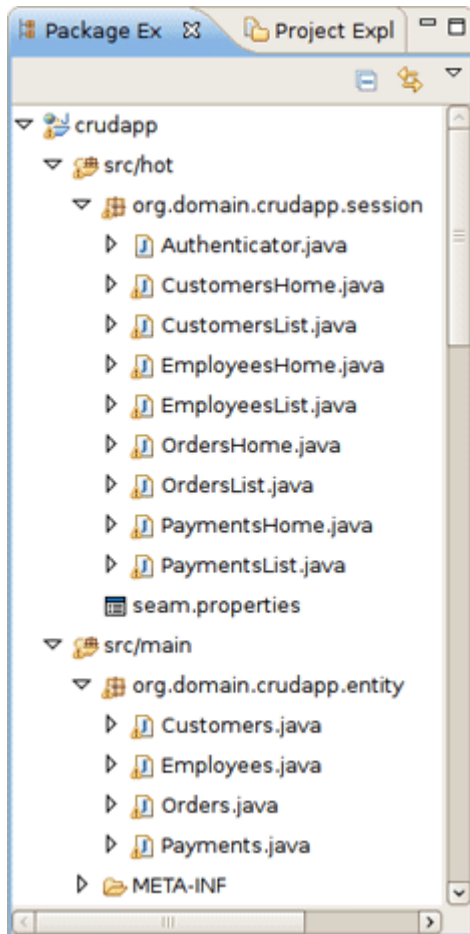
- Under *WebContent* folder you can find all generated xhtml files:



**Figure 13.19. Entities Web Pages**

- And under `src` folder java classes are created.





**Figure 13.20. Entities Java Classes**

- Switch to Hibernate perspective with **Window** → **Open Perspective** → **Other....** On Hibernate Configurations view expand the *crudapp* configuration. Right click on Customers and select *Open Mapping Diagram* from the popup menu.

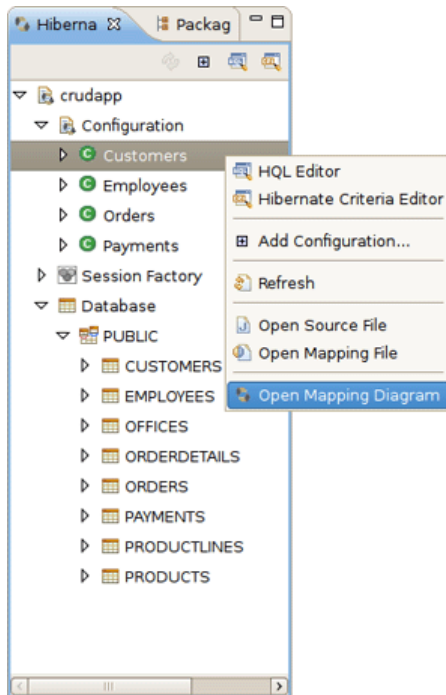


Figure 13.21. Hibernate Configurations View

- Observe the Mapping Diagram opened in the editor. Here you can see the relations between models and database tables. For better navigating on the diagram use the Outline view.

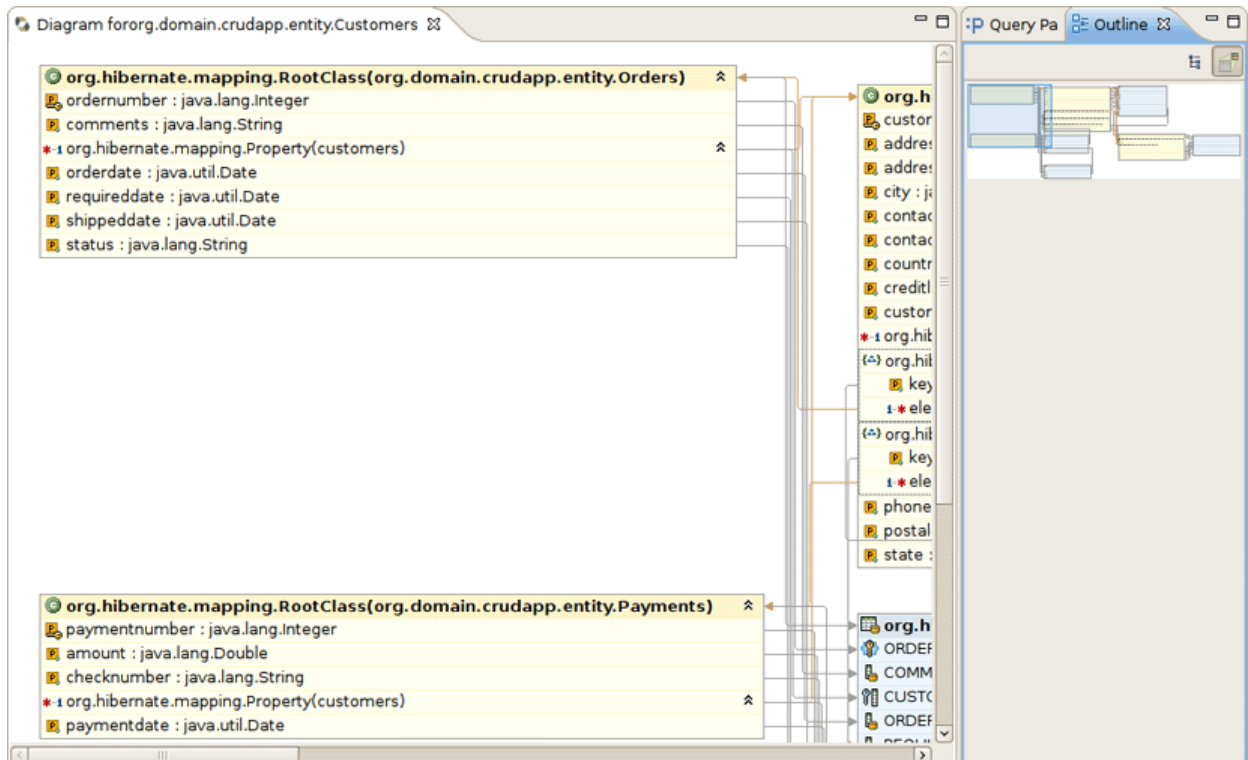
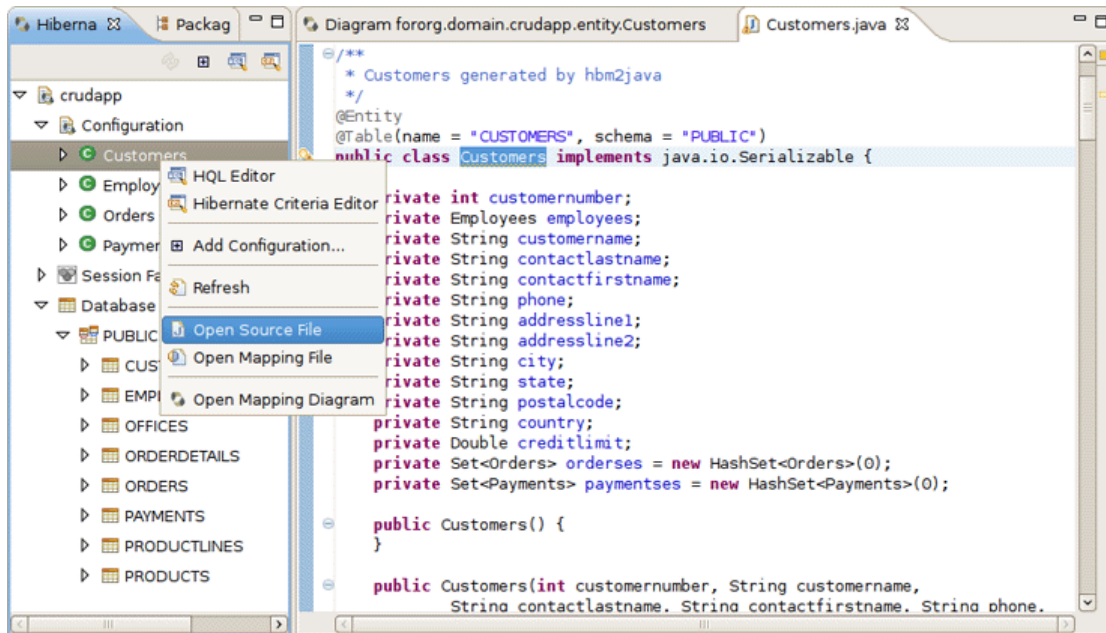


Figure 13.22. Mapping Diagram

- For example, select Customers entity, right click and select *Open Source File*. This will open the Customers.java file in the java editor. You can also do this by right-clicking on the same entity on the Mapping Diagram.



**Figure 13.23. Entity class from Mapping Diagram**

So far, you are ready to deploy your application to JBoss Application Server. This is described in the next chapter.



# The CRUD Application Walkthrough

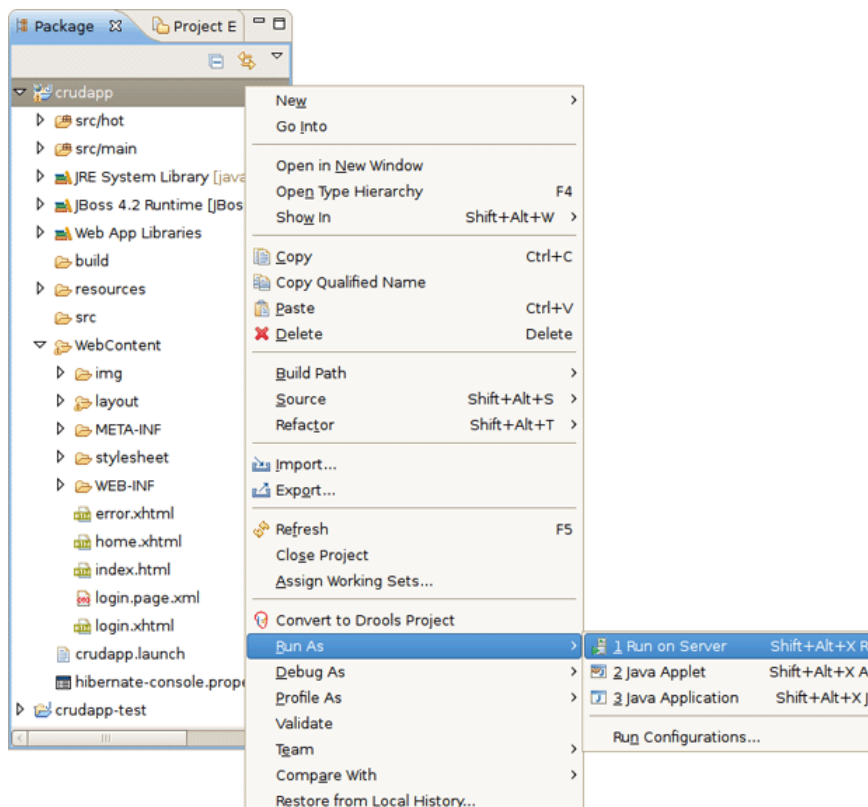
After you familiarized oneself with example of creating the CRUD Database Application with Seam, you can read this charter.

## 14.1. Using CRUD Application

To run your CRUD Application you should do the following steps:

- Run a project on the Server.

On *Package Explorer View* right click on the *crudapp* project, select **Run As** → **Run on Server**.



**Figure 14.1. Run Project on Server**

Select a Server and click Finish

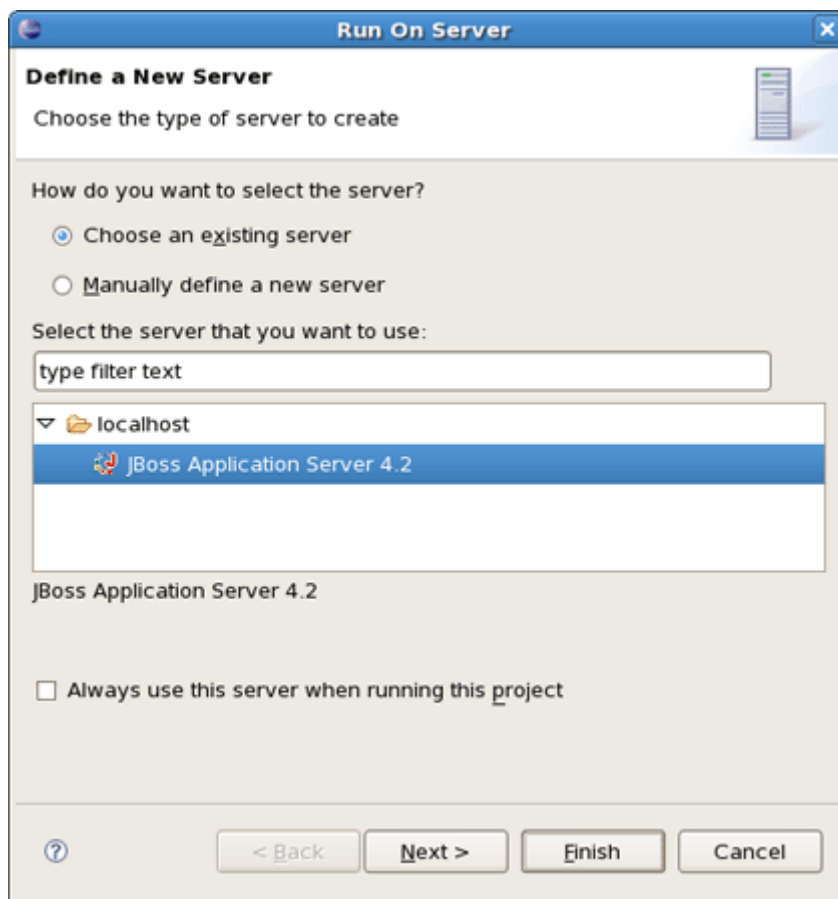


Figure 14.2. Select a Server

Home page of the *crudapp* project should appear in *Web Browser*.

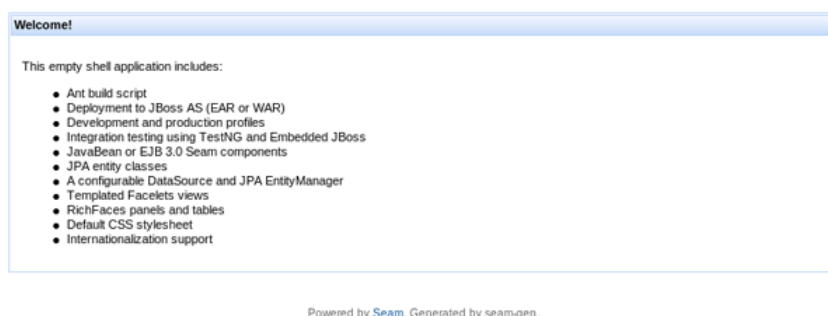


Figure 14.3. Home page

After that you can use the CRUD application with the *employee* database.

You can use internal JBDS Web Browser or your external Web Browser with the same link (<http://localhost:8080/crudapp/home.seam>).

Hover the mouse over **Browse data** in the very top menu and select **Employees List** from the expanded list. Observe that data from the *employee* database is displayed.

Employees search parameters

lastname

firstname

extension

email

officecode

jobtitle

Search

Employees search results

employeenumber	lastname	firstname	extension	email	officecode	reportsto	jobtitle	action
1002	Murphy	Diane	x5800	dmurphy@classicmodelcars.com	1		President	<a href="#">Select</a>
1056	Patterson	Mary	x4611	mpatterso@classicmodelcars.com	1	1002	VP Sales	<a href="#">Select</a>
1076	Firrelli	Jeff	x9273	jfirrelli@classicmodelcars.com	1	1002	VP Marketing	<a href="#">Select</a>
1088	Patterson	William	x4871	wpatterson@classicmodelcars.com	6	1056	Sales Manager (APAC)	<a href="#">Select</a>
1102	Bondur	Gerard	x5408	gbondur@classicmodelcars.com	4	1056	Sale Manager (EMEA)	<a href="#">Select</a>
1143	Bow	Anthony	x5428	abow@classicmodelcars.com	1	1056	Sales Manager (NA)	<a href="#">Select</a>
1165	Jennings	Leslie	x3291	ljennings@classicmodelcars.com	1	1143	Sales Rep	<a href="#">Select</a>
1166	Thompson	Leslie	x4065	lthompson@classicmodelcars.com	1	1143	Sales Rep	<a href="#">Select</a>
1188	Firrelli	Julia	x2173	jfirrelli@classicmodelcars.com	2	1143	Sales Rep	<a href="#">Select</a>

**Figure 14.4. Displayed employee data**

Use Employees search parameters fields to filter the selected list.

Click the **View** button in the **Action** column of one of the employees.

**Employees**

<b>employeenumber</b>	1002
<b>lastname</b>	Murphy
<b>firstname</b>	Diane
<b>extension</b>	x5800
<b>email</b>	dmurphy@classicmodelcars.com
<b>officecode</b>	1
<b>reportsto</b>	
<b>jobtitle</b>	President

[Edit](#) [Done](#)

**customerses**

There are no customerses associated with this employees.

[Add customers](#)

Powered by [Seam](#). Generated by seam-gen.

**Figure 14.5. Employee details Page**

Press Edit to edit employee fields.

Enter *Login* and *Password* to login (for example, admin as the login and leaving the password field blank).

• Please log in first

**Login**

Please login here

**Username**

**Password**

**Remember me** ☒

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

[Login](#)

Powered by [Seam](#). Generated by seam-gen.

**Figure 14.6. Login page**



**Edit Employees**

**employee number \*** 1002

**lastname \*** Murphy

**firstname \*** Mary

**extension \*** x5800

**email \*** dmurphy@classicmodelcars.com

**officecode \*** 1

**reportsto**

**jobtitle \*** President

\* required fields

Save Delete Done

customerses

There are no customerses associated with this employees.

Add customers

**Figure 14.7. Edit Employee**

Fill in *firstname* and press Save.

Database will be updated.



# What is TestNG?

TestNG ("Testing, the Next Generation") is a Java unit testing framework that aims to overcome many limitations of JUnit. TestNG introduces some new functionalities that make it more powerful and easier to use, such as:

- JDK 5 Annotations (JDK 1.4 is also supported with JavaDoc annotations)
- Flexible test configuration
- Support for data-driven testing (with `@DataProvider`)
- Support for parameters
- Allows distribution of tests on slave machines
- Powerful execution model (no more TestSuite)
- Supported by a variety of tools and plug-ins (Eclipse, IDEA, Maven, etc...)
- Embeds BeanShell for further flexibility
- Default JDK functions for runtime and logging (no dependencies)
- Dependent methods for application server testing

## 15.1. How to use the generated Seam-test project to run Seam tests?

- Create a new Seam Web Project with EAR deployment using the New Seam Project wizard.
- After a project is created you will have the generated Seam-test project that is setup to run TestNG directly against the proper libraries and server runtime libraries.

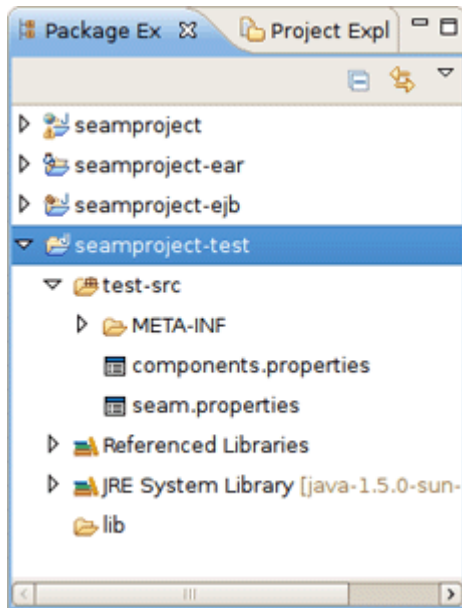


Figure 15.1. Seam-test Project

- Add Seam Action to your project via **File** → **New** → **Seam Action**.

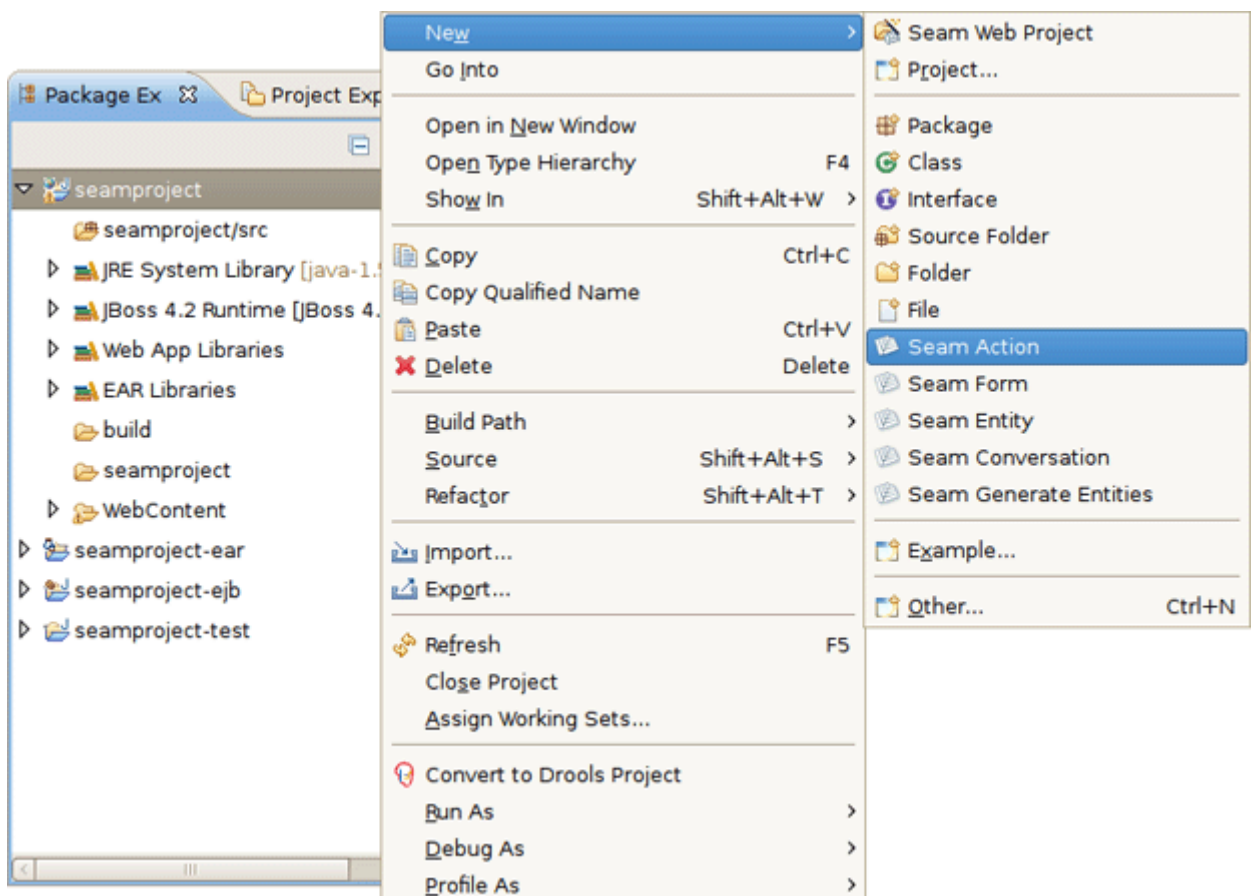
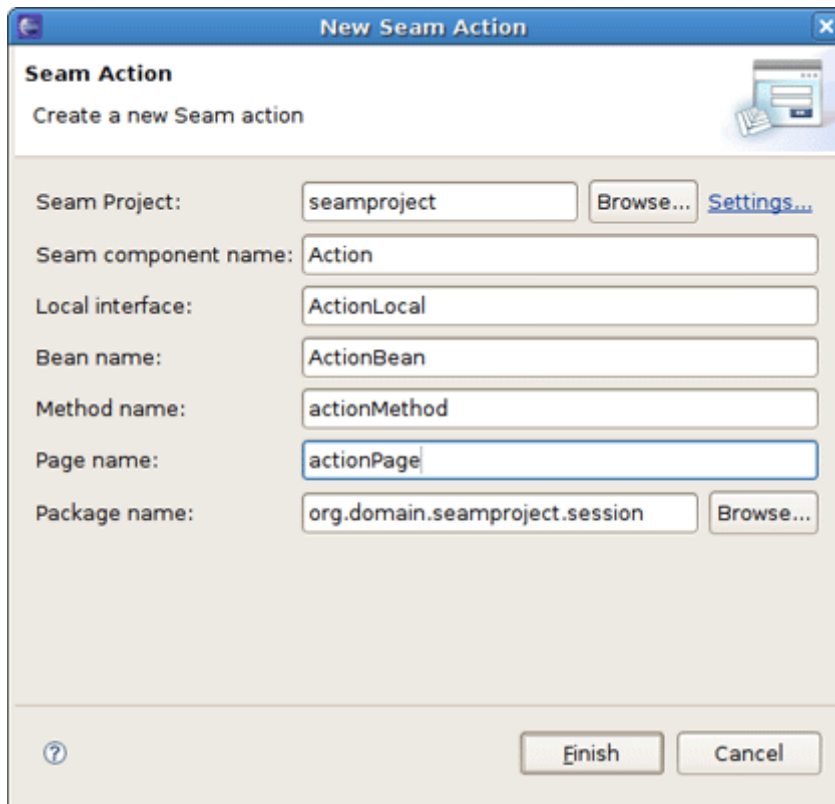


Figure 15.2. Seam Action Creation

- Fill out the wizard fields. New Seam Action wizard will create resources and place them in the appropriate folders dependent on EAR project structure.



**Figure 15.3. New Seam Action Wizard**

- When Action is created you will see actionPage.xhtml in Package Explorer view. ActionBean.java will be automatically opened in Java Editor.



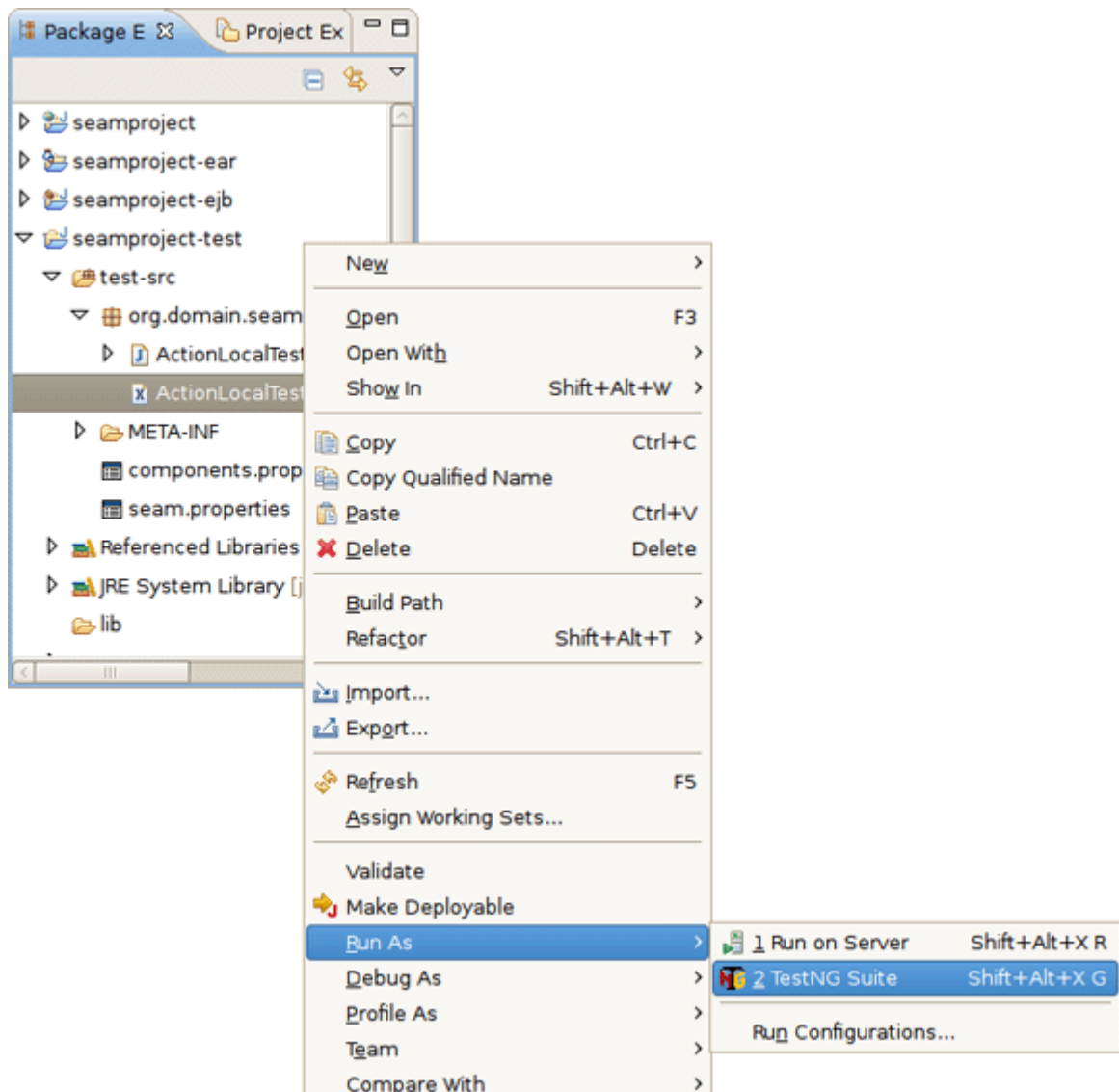
Figure 15.4. Created Action

- Select ActionLocalTest.xml in Seam-test project and run the test with right click **Run As** → **TestNG Suite**.



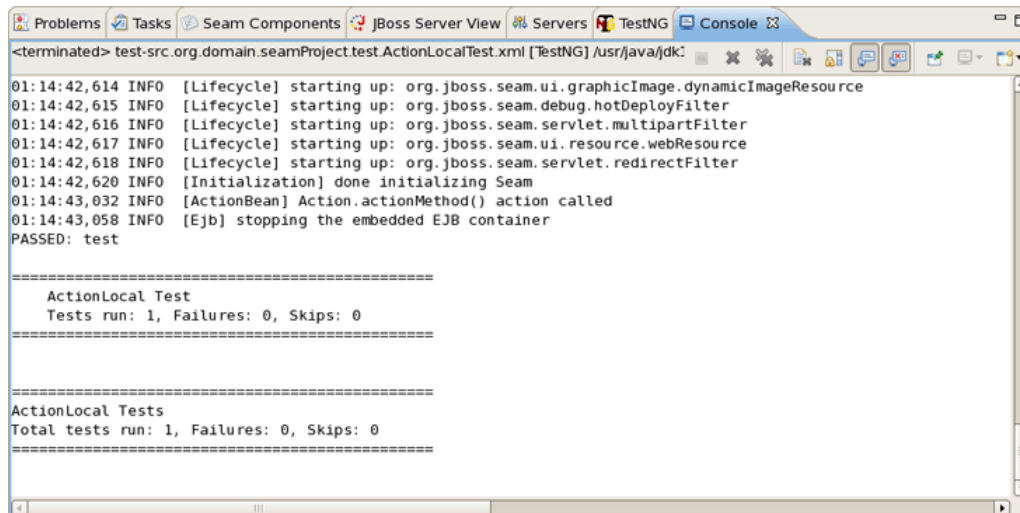
### Note

OpenOn is available in testNG XML files opened in JBoss XML Editor



**Figure 15.5. Running TestNG**

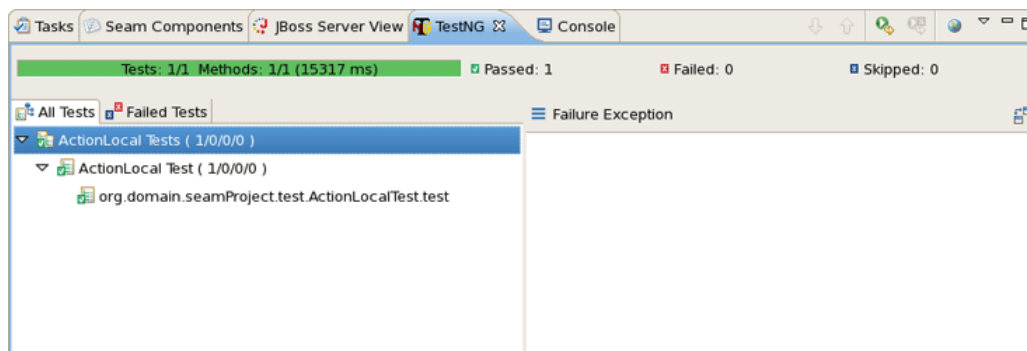
The test process will start and its output will be written in Console View.



**Figure 15.6. Test is Finished**

- After running TestNG you will have the test results in *test-output* folder in Seam-test project (press F5 to refresh the Package Explorer view). Open index.html file with Web Browser or simply use the TestNG view.

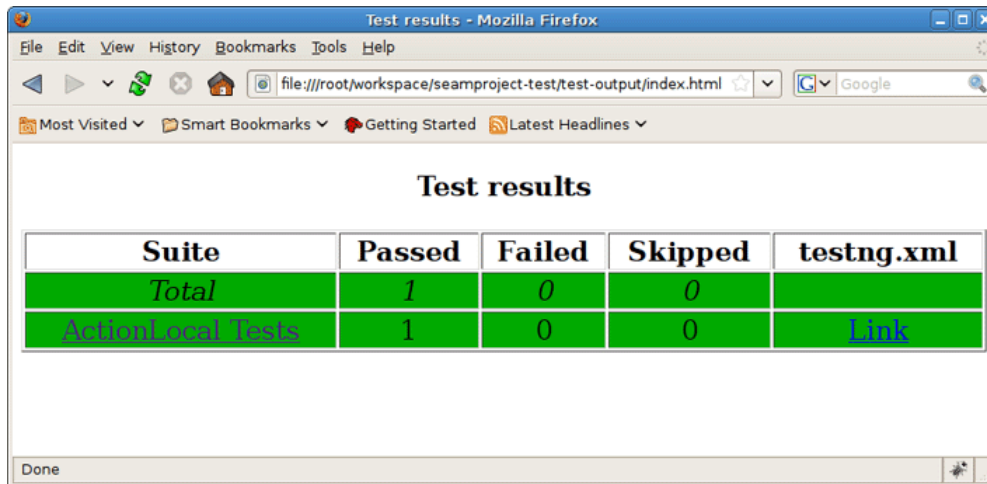
The below view shows a successful run of the test.



**Figure 15.7. Viewing the Test Results**

You can see the test results in Web Browser.





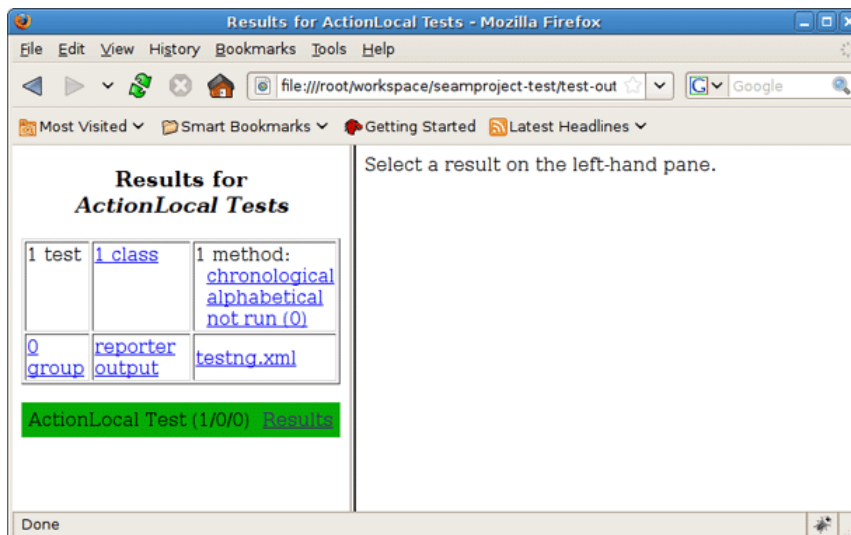
The screenshot shows a Mozilla Firefox browser window titled 'Test results - Mozilla Firefox'. The address bar shows the file path 'file:///root/workspace/seamproject-test/test-output/index.html'. The page content is titled 'Test results' and contains a table with the following data:

Suite	Passed	Failed	Skipped	testng.xml
Total	1	0	0	
ActionLocal Tests	1	0	0	<a href="#">Link</a>

The status bar at the bottom shows 'Done'.

**Figure 15.8. Test Results in Browser**

After clicking on *ActionLocal Tests* link you will see the Results for ActionLocal Tests.



The screenshot shows a Mozilla Firefox browser window titled 'Results for ActionLocal Tests - Mozilla Firefox'. The address bar shows the file path 'file:///root/workspace/seamproject-test/test-out'. The page content is titled 'Results for ActionLocal Tests' and is divided into two panes. The left-hand pane contains the following information:

1 test	<a href="#">1 class</a>	1 method: <a href="#">chronological</a> <a href="#">alphabetical</a> <a href="#">not run (0)</a>
0 group	<a href="#">reporter output</a>	<a href="#">testng.xml</a>

Below the table is a green bar with the text 'ActionLocal Test (1/0/0) Results'. The right-hand pane contains the text 'Select a result on the left-hand pane.' The status bar at the bottom shows 'Done'.

**Figure 15.9. Test Information**

Select a result on the left-hand pane and its details will be displayed on the right-hand one.

**Results for ActionLocal Tests**

1 test	<a href="#">1 class</a>	1 method: <a href="#">chronological</a> <a href="#">alphabetical</a> <a href="#">not run (0)</a>
<a href="#">0 group</a>	<a href="#">reporter output</a>	<a href="#">testng.xml</a>

ActionLocal Test (1/0/0) [Results](#)

**ActionLocal Test**

Tests passed/Failed/Skipped:	1/0/0
Started on:	Tue Dec 02 15:40:31 EE
Total time:	18 seconds (18077 ms)
Included groups:	
Excluded groups:	

(Hover the method name to see the test class name)

PASSED TESTS		
Test method	Time (seconds)	Exception
test_actionMethod	0	

file:///root/workspace/seamproject-test/test-output/ActionLocal Tests/ActionLocal Test.html

Figure 15.10. ActionLocal Test Details

Thus with Seam tooling you can easily take advantage of TestNG framework. As you can see, it generates its own TestNG project as a separate module within which you can easily monitor the tests execution and their output.

## FAQ

### 16.1. How to get Code Assist for Seam specific resources in an externally generated project?

To get Code Assist for Seam specific resources in an externally generated project, you should enable Seam features in Project Preferences. Right click an imported project and navigate **Properties** → **Seam Settings**. Check *Seam support* box to enable all available [Section 10.2, “Project Preferences”](#).

### 16.2. How to import an example Seam project from jboss-eap directory?

To import an example Seam project from *jboss-eap* into your working directory, you should perform the following steps:

- Go to **New** → **Other** → **Java Project from Existing Buildfile**
- Point to the *build.xml* of any chosen project by pressing *Browse* button
- Hit *Finish* to open the project

As these seam examples are non WTP projects, next you should enable Seam support for them. To do that, right click the project and go to **Properties** → **Seam Settings**.



#### Important

Projects are not directly deployable unless they are in the correct state, such as an archive created through **Project Archives**. As an alternative, working examples are provided for Seam by navigating to **Help** → **Project Examples** → **Seam**

### 16.3. How to change the deploy folders for Seam source code from src/hot and src/main?

You should right-click the project, select **Properties** → **Java Build Path**, then switch to the *Source* tab, select the output folder for *src/hot* ( *src/main*) and click *Edit* to change their output location.

