Seam Dev Tools Reference Guide

Version: 3.0.0.GA

1. Introduction 1
1.1. What is Seam? 1
1.2. Installation into Eclipse 1
1.3. Other relevant resources on the topic 1
2. Creating a New Seam Project via the New Seam Project wizard 3
2.1. Create standalone Seam Web Project 3
2.2. Select the Project Facets 10
2.3. How to Configure Web Module Settings 13
2.4. Adding JSF Capabilities 14
2.5. Configure Seam Facet Settings 17
3. Directory Structure of the Generated Project 25
3.1. WAR Deployment 25
3.2. EAR Deployment 25
3.3. Changing the Seam Version 26
3.4. Changing Seam Parent Project 29
3.5. Renaming the Projects and Folders 30
4. Seam Menus and Actions 31
4.1. File Menu Actions 31
4.2. Navigate Menu Actions 31
4.2.1. Find Seam References/Declarations 31
4.2.2. Open Seam Component 33
5. Seam Wizards 35
5.1. New Seam Action 36
5.2. New Seam Form 40
5.3. New Seam Conversation 43
5.4. New Seam Entity 46
6. Seam Generate Entities 51
7. Seam Editors 53
7.1. Visual Page Editor 53
7.2. Seam Pages Editor 53
7.2.1. Graphical Mode 53
7.2.2. Tree Mode 56
7.2.3. Outline Support for Seam Pages Editor 57
7.3. Seam Components Editor 58
7.4. Main Features of Seam Editors 60
7.4.1. Content Assist 60
7.4.2. OpenOn 63
7.4.3. Seam Validation 64
8. Seam Views 67
8.1. Seam Components View 67
8.2. Project Explorer integration 70
9. Seam Preferences 73
9.1. General Preferences 73
9.1.1. Validator Preferences 75

9.2. Project Preferences 79
10. Adding Seam support to EAR project 85
11. Generate a CRUD Database Application 93
11.1. What is CRUD? 93
11.2. How to create the CRUD Database Application with Seam 93
12. The CRUD Application Walkthrough 115
12.1. Using CRUD Application 115
13. Using TestNG project 121
13.1. What is TestNG? 121
13.2. Other relevant resources on the topic 121
13.3. How to use the generated Seam-test project to run Seam tests? 122

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 framework 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. Installation into Eclipse

Here, we are going to explain how to install Seam plugin into Eclipse.

Seam is a one module of JBoss Tools project. Since Seam has a dependence on other JBoss Tools modules we recommend you to install a bundle of all *JBoss Tools plug-ins* [http://labs.jboss.com/tools/download/index.html]. Installation instruction you can find on JBoss Wiki in *InstallingJBossTools* [http://labs.jboss.com/wiki/InstallingJBossTools] section.

1.3. Other relevant resources on the topic

The *Seam Framework* [http://seamframework.org/] - Next generation enterprise Java development.

All latest release versions of JBoss Developer Studio/JBoss Tools documentation you can find *here* [http://docs.jboss.org/tools/2.1.0.GA/].

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

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.

At first, we suggest setting 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* or you can also access it through the button in the right top corner.

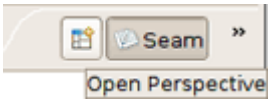


Figure 2.1. Seam Perspective Icon

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

Figure 2.2. Embedded Web Browser Button

2.1. Create standalone Seam Web Project

The best way to get started with Seam is to organise a simple Seam Project and experiment with it by creating variations.

Thus, you should select *File > New > 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.

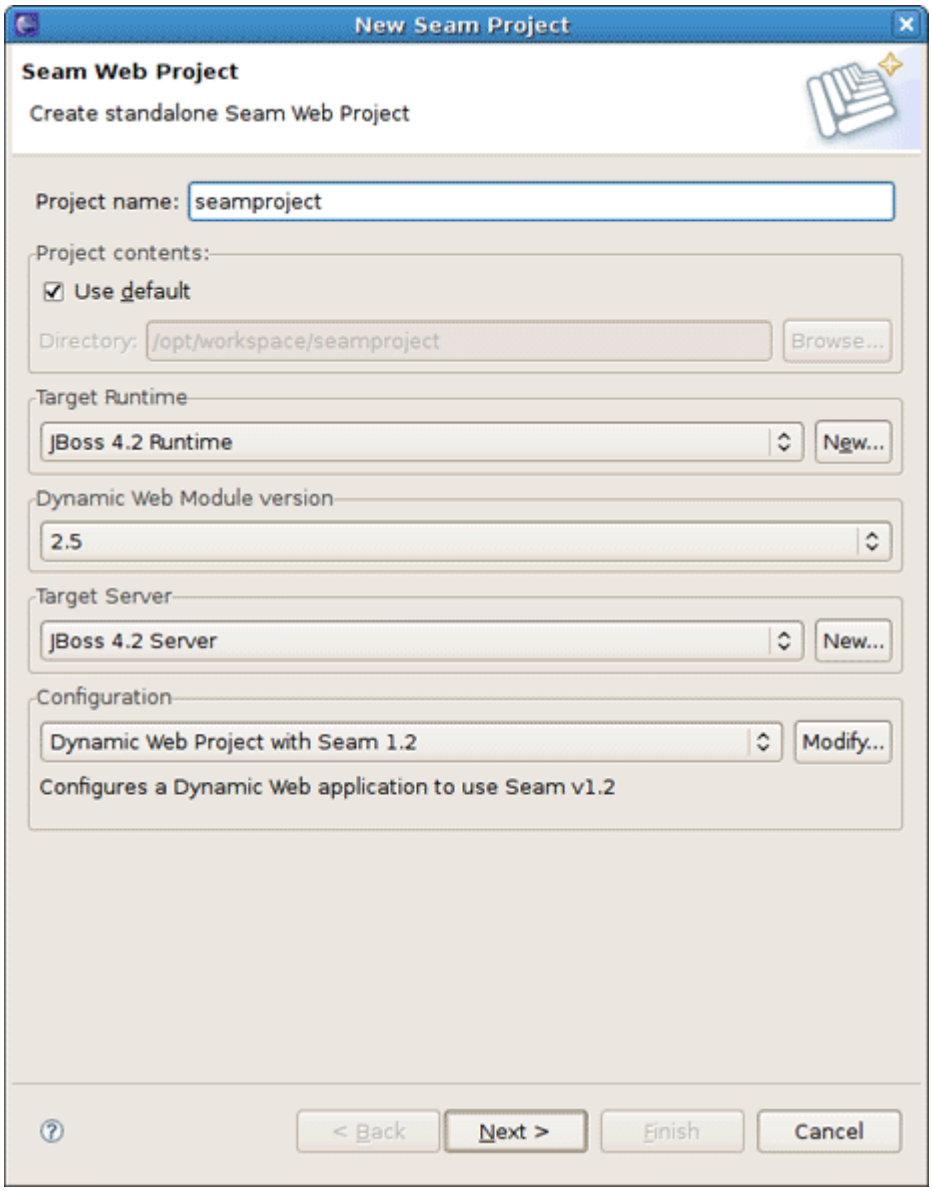


Figure 2.3. New Seam Project Wizard

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 configuring your project. Let's create one more JBoss 4.2 Runtime. Hence, after choosing it click on *Next* button.

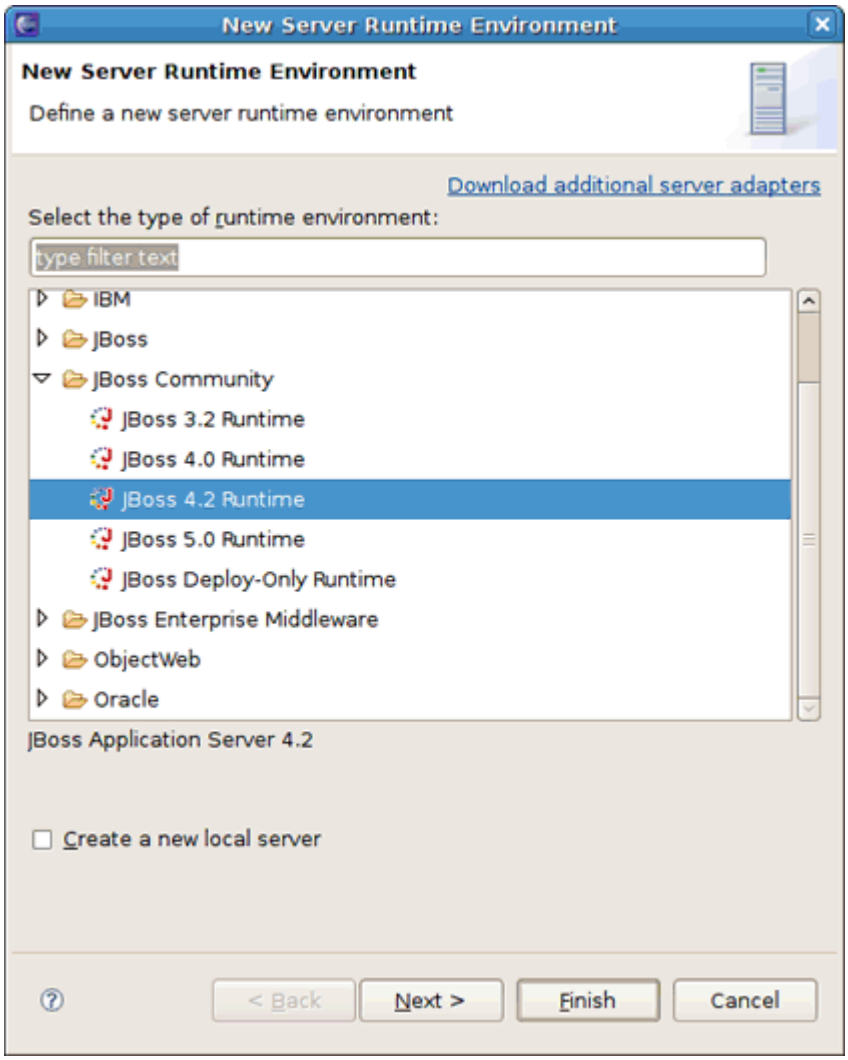


Figure 2.4. Specifying Target Runtime

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.

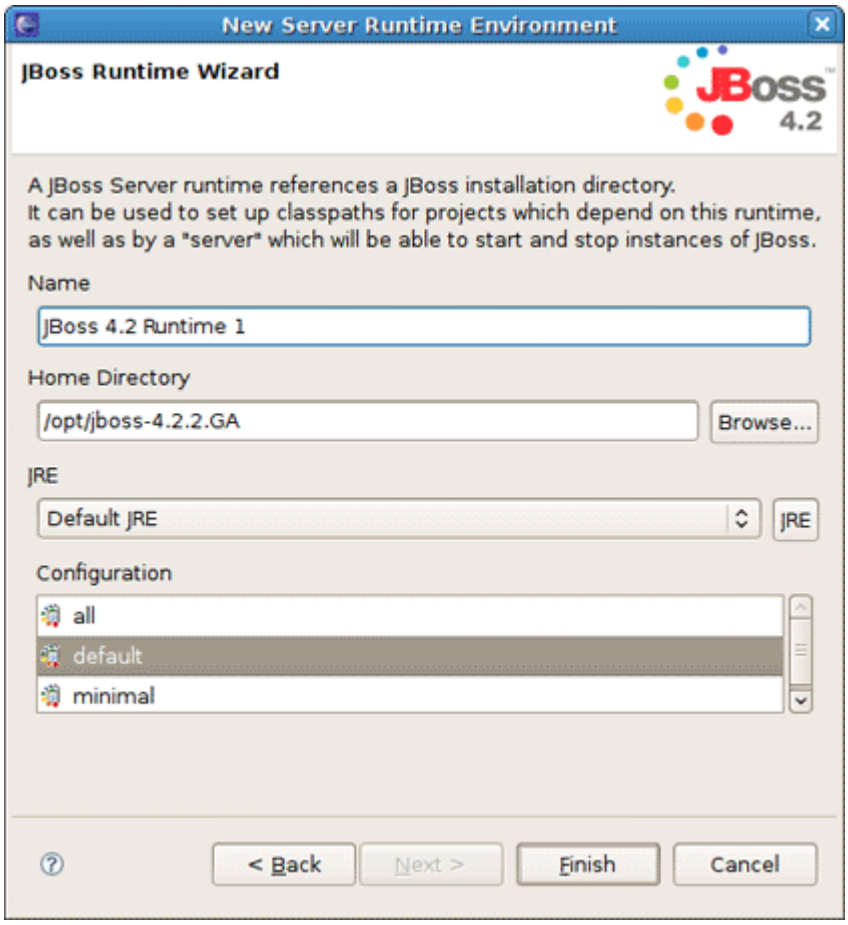


Figure 2.5. Specifying Target Runtime Configurations

Clicking on *Finish* returns you to the *New Seam Project wizard page* .

The next step is to define a Server that you can do 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 combo box 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.

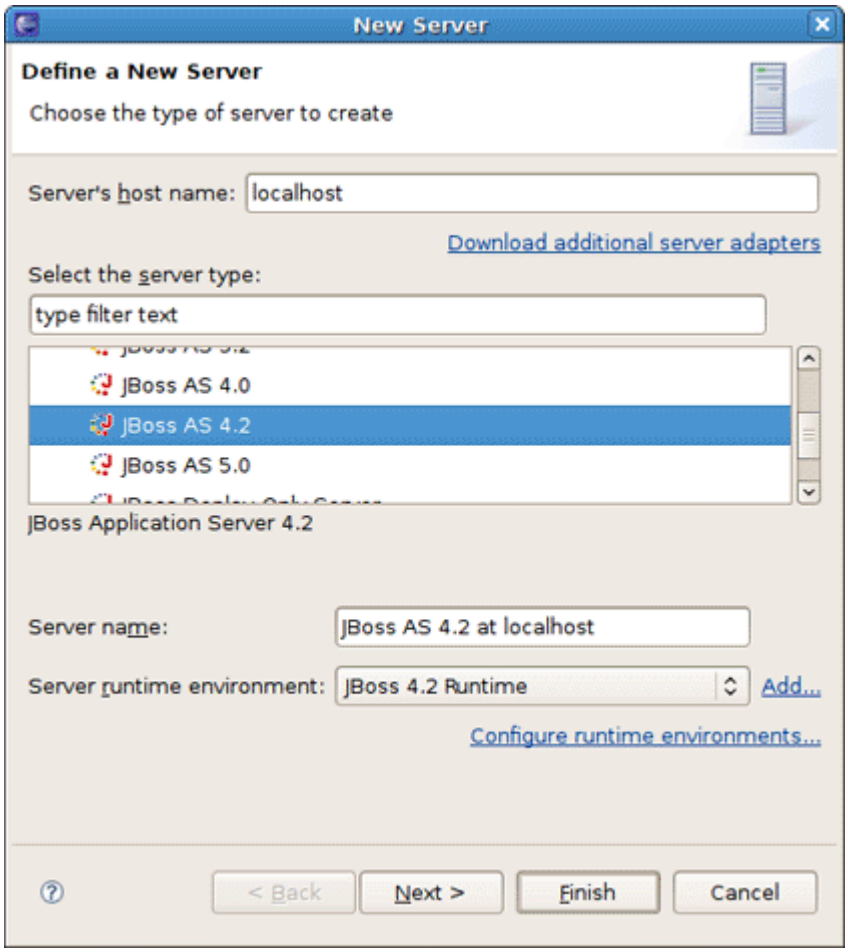


Figure 2.6. Specifying Target Server

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

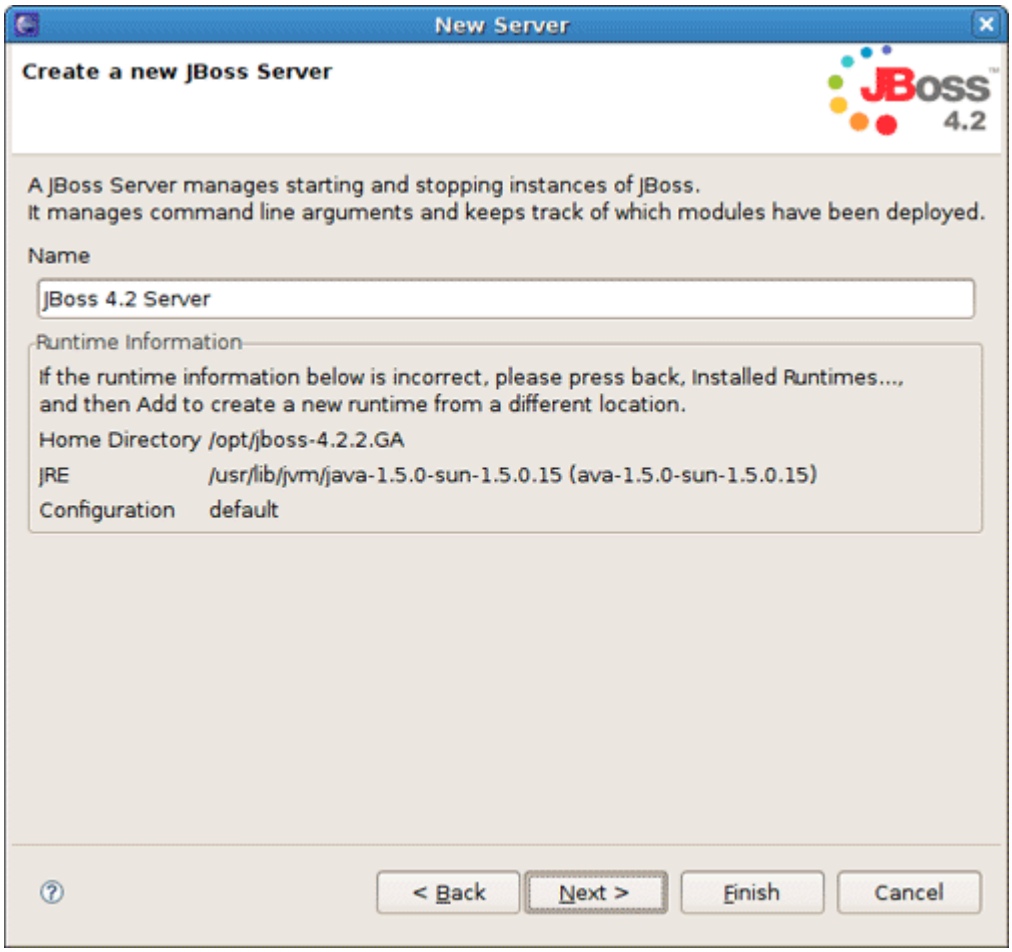


Figure 2.7. Specifying Server Configurations

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

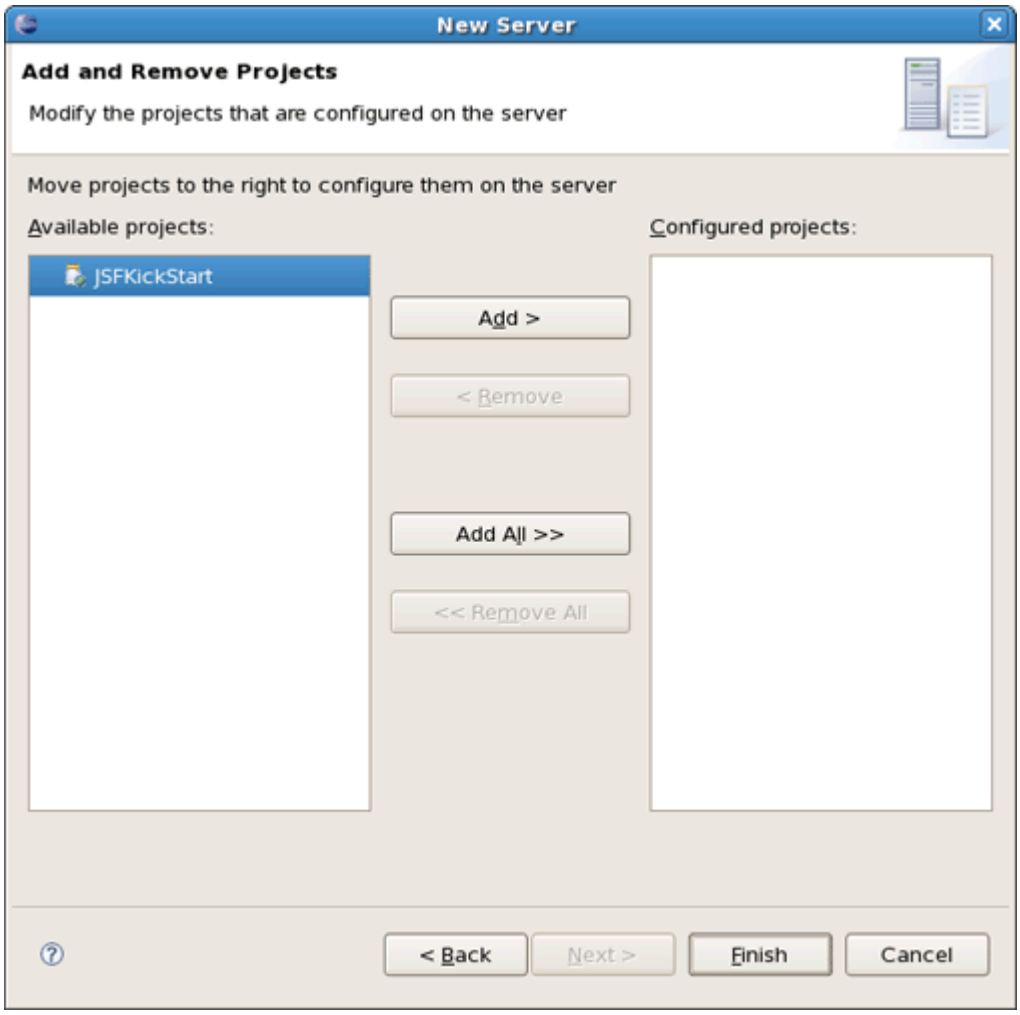


Figure 2.8. Project Modification for Configuring on the Server

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

Tip:

We suggest that you look through our *AS manager guide* [../../as/html_single/index.html] to find out more about runtimes and servers.

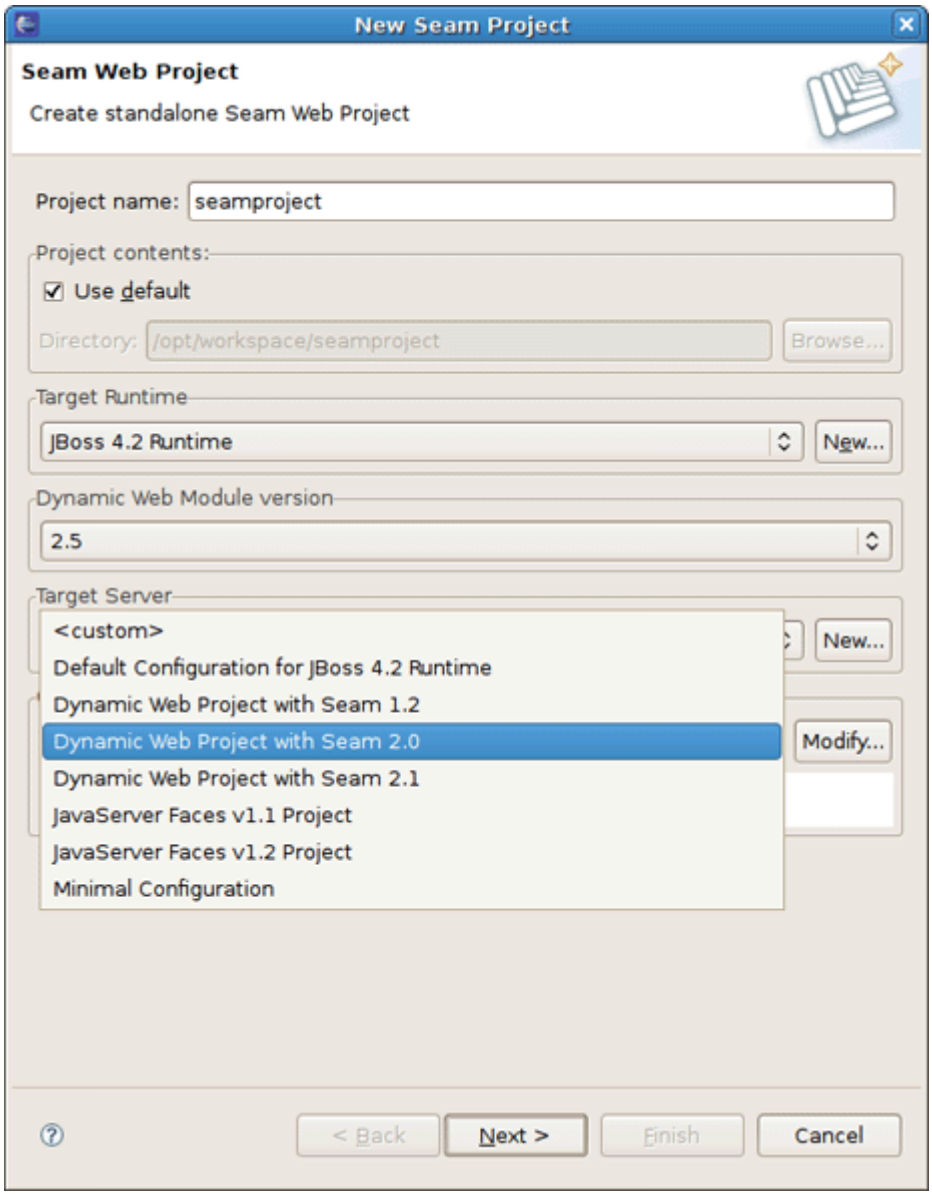


Figure 2.9. Completion of Runtime and Server Configuration

The last section on this wizard step is Configuration. Here, you can select one of the pre-defined 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. Select the Project Facets

The Project Facets wizard allows you to enable or disable specific facets which define necessary characteristics for the project. In time 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.

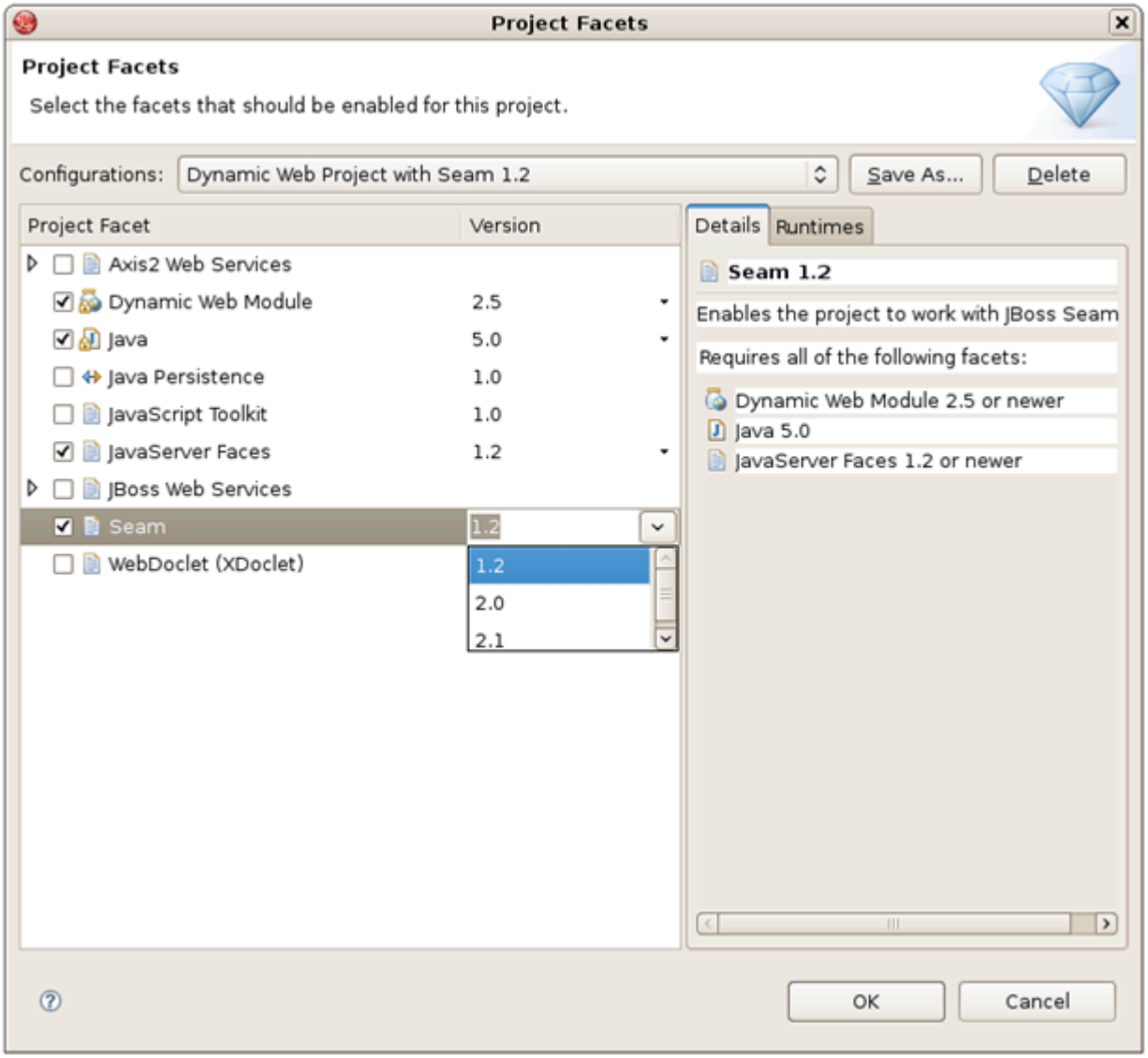


Figure 2.10. Project Facets Selection

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

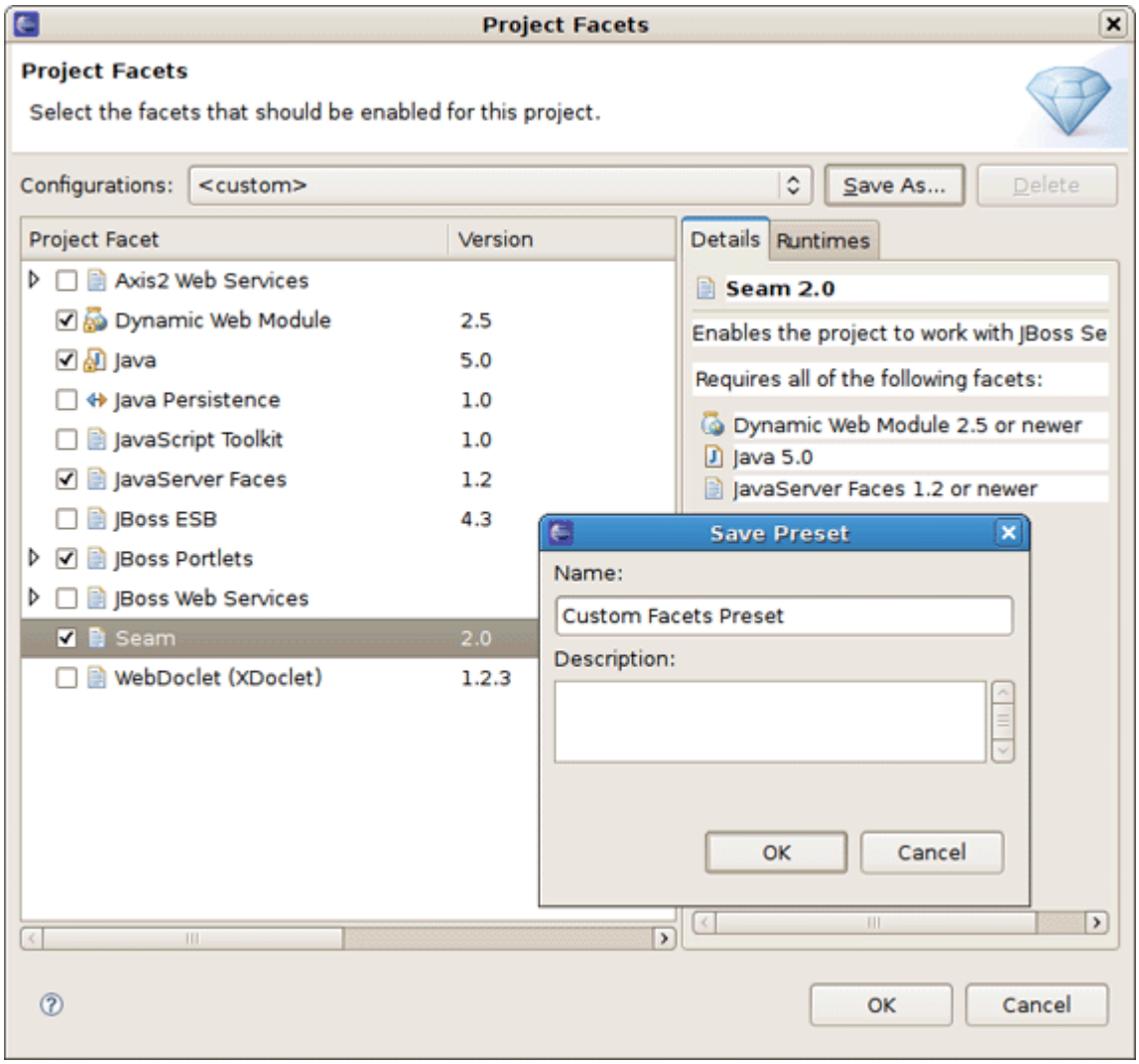


Figure 2.11. Specifying Custom Facet Preset

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 runtimes are checked here, the *Make Primary* button won't be dimmed yet. So you can make use of it to mark primary runtime.

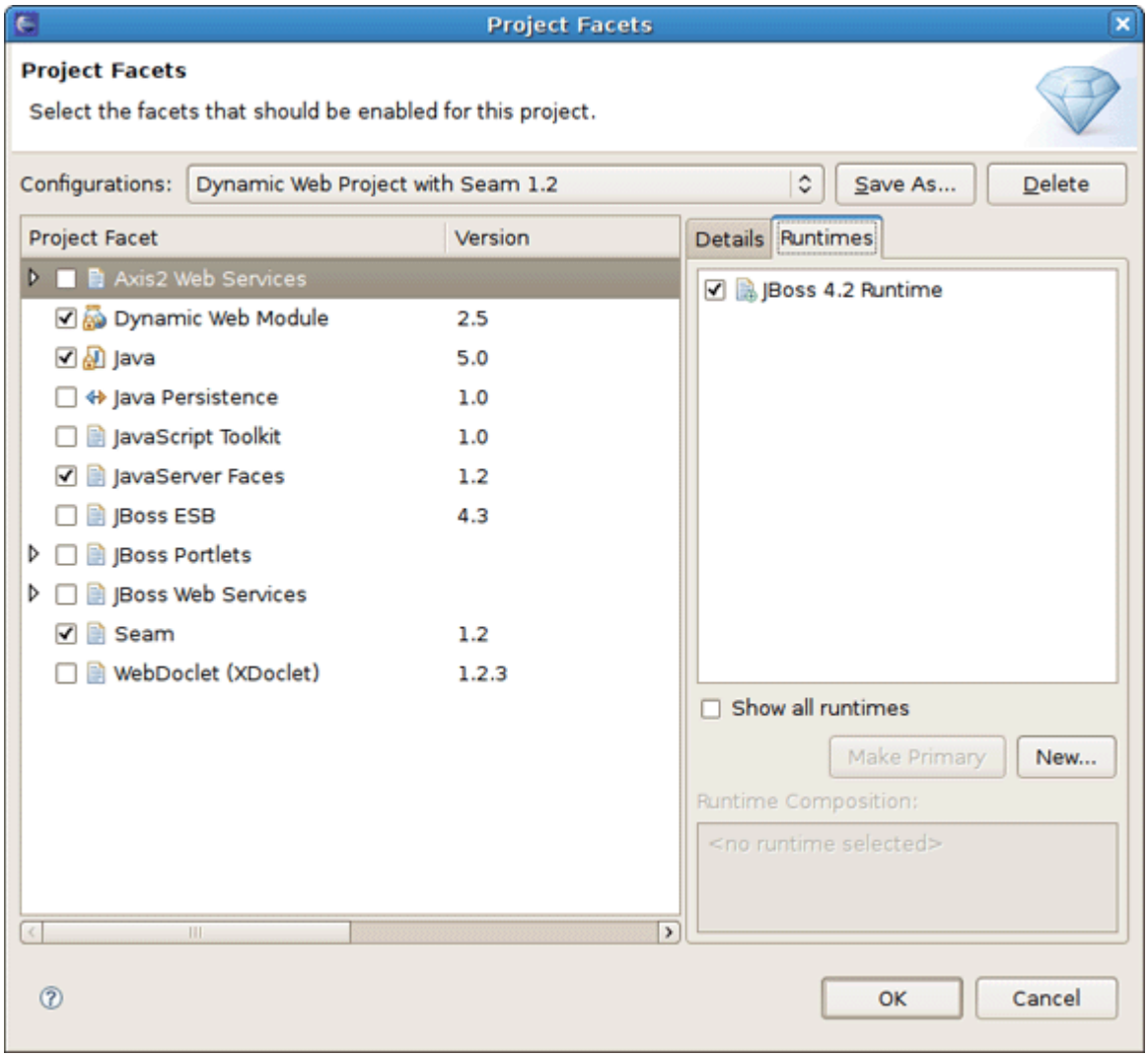


Figure 2.12. Runtime Manipulations

Ok button will bring you to the Web Module wizard form again.

2.3. How to Configure Web Module Settings

As we deal with a Dynamic Web Application we should first specify the top level directory of our application for deploying it to a server afterwards. You know, this kind of application contains both Web and Java code resources. Thus, it's also important to indicate the content directory as well as Java source directory. The wizard will put all those values itself. So you can leave everything as it is.

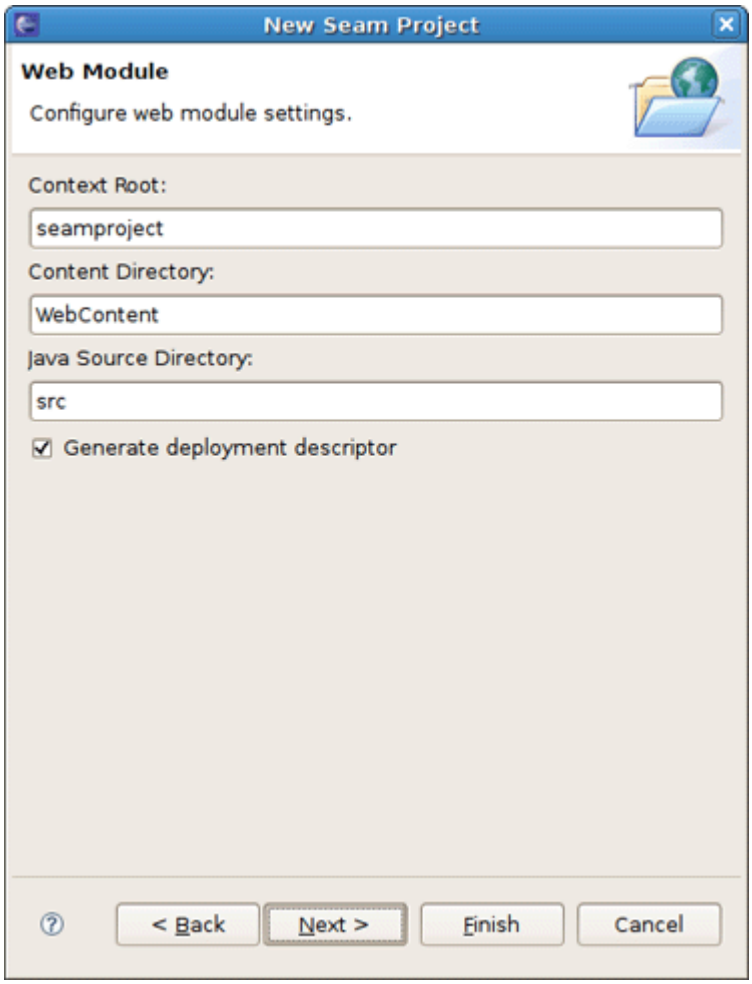


Figure 2.13. Web Module Settings

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

2.4. Adding JSF Capabilities

This wizard helps you to add JSF capabilities into your project for representing appropriate behaviours associated with JSF.

Checking *Server Supplied JSF Implementation* means that you will have a default JSF implementation given by server.

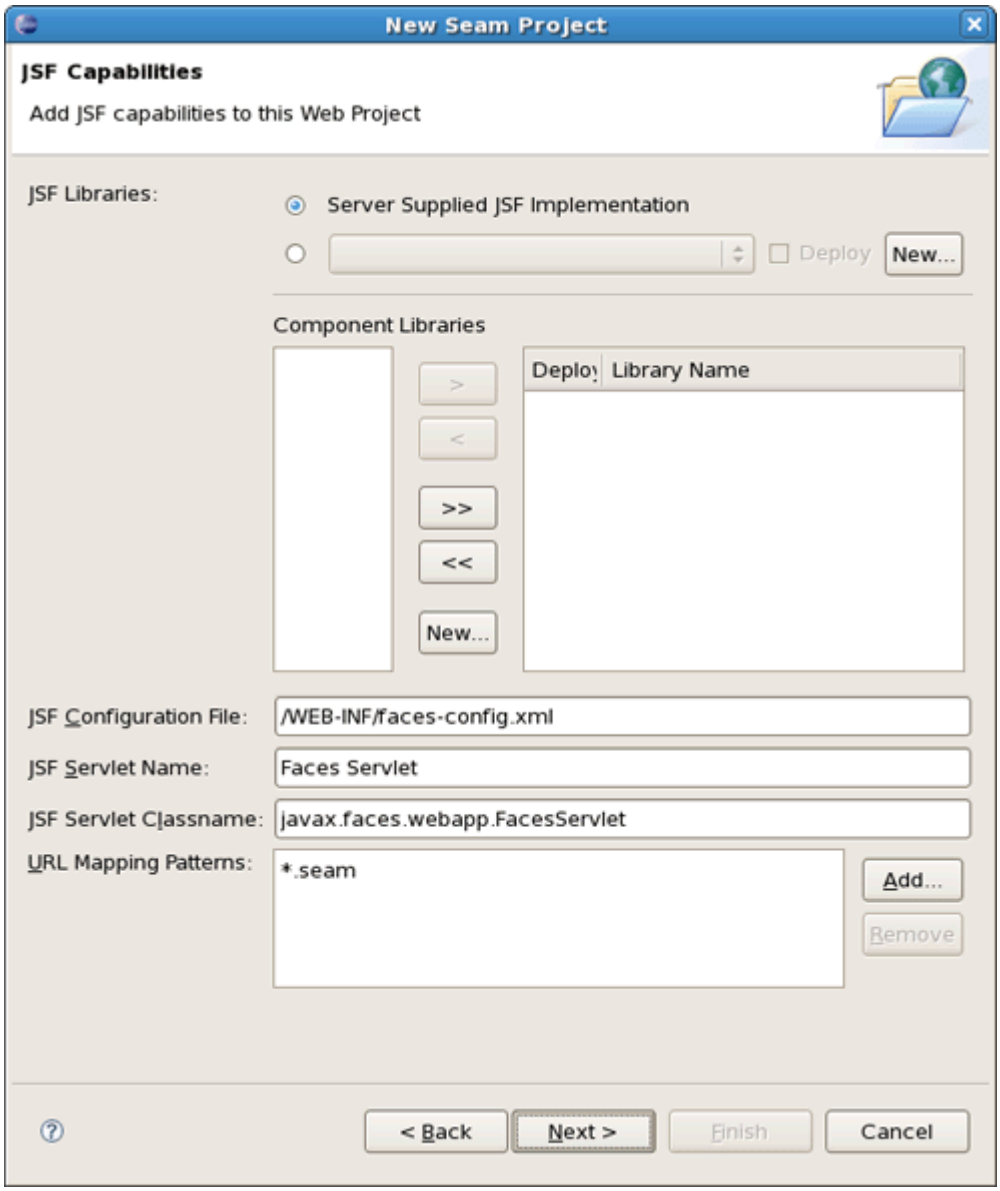


Figure 2.14. Adding JSF Capabilities to Web Project

In case when you want to use your custom JSF implementation check a lower radio button. You are able to create a library of jars by clicking on *New* button.

Here, it's necessary to type a Library Name, select a Version Supported and add proper Library jars. Then click on *Finish* to complete the choice.

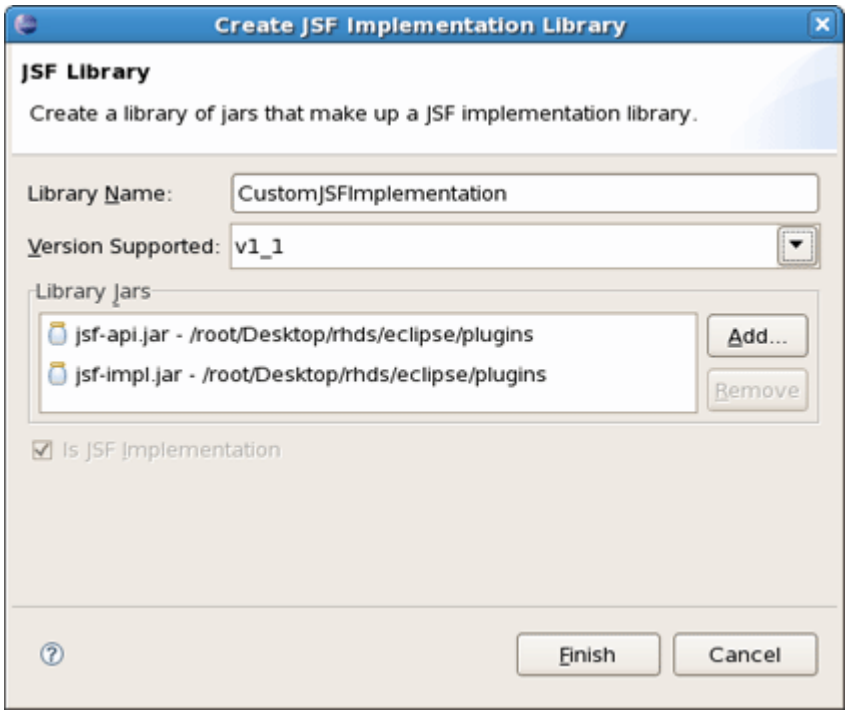


Figure 2.15. Create JSF Implementation Library

In the Component Libraries section of the wizard you can also add Component Libraries (e.g. *Richfaces* [http://www.jboss.org/jbossrichfaces]). Just click on *New* button. Appeared dialog will ask you to type the Library name, supported version and add necessary jar's. Press *Finish* to complete the choice.

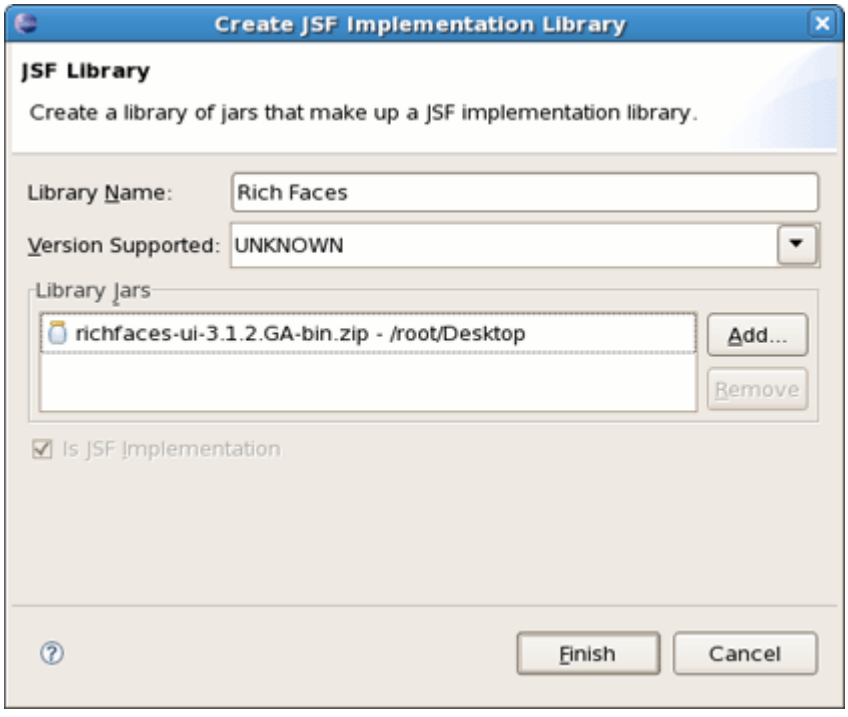


Figure 2.16. Create JSF Implementation Component Library

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.

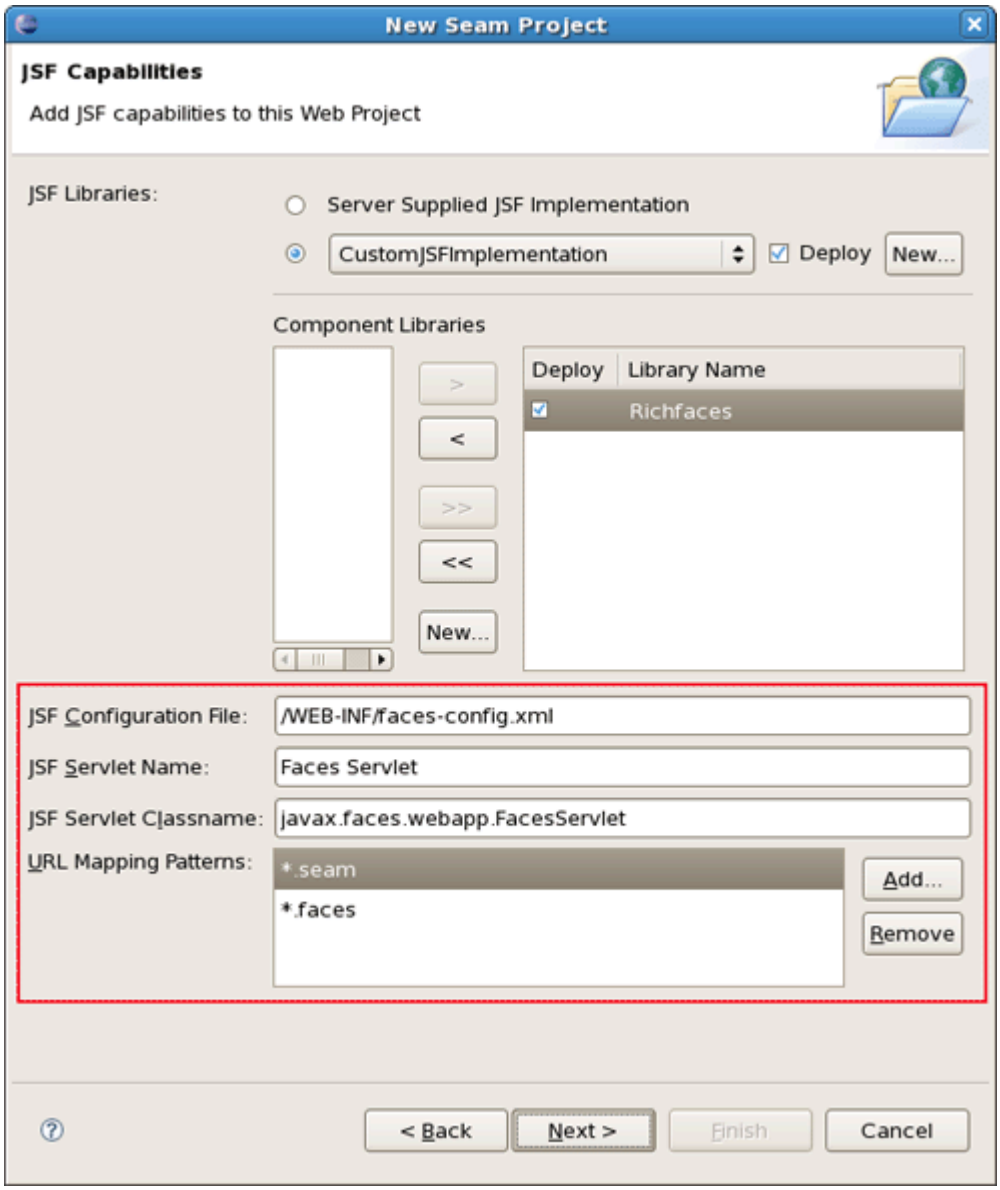


Figure 2.17. JSF Capabilities Wizard

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

2.5. Configure Seam Facet Settings

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

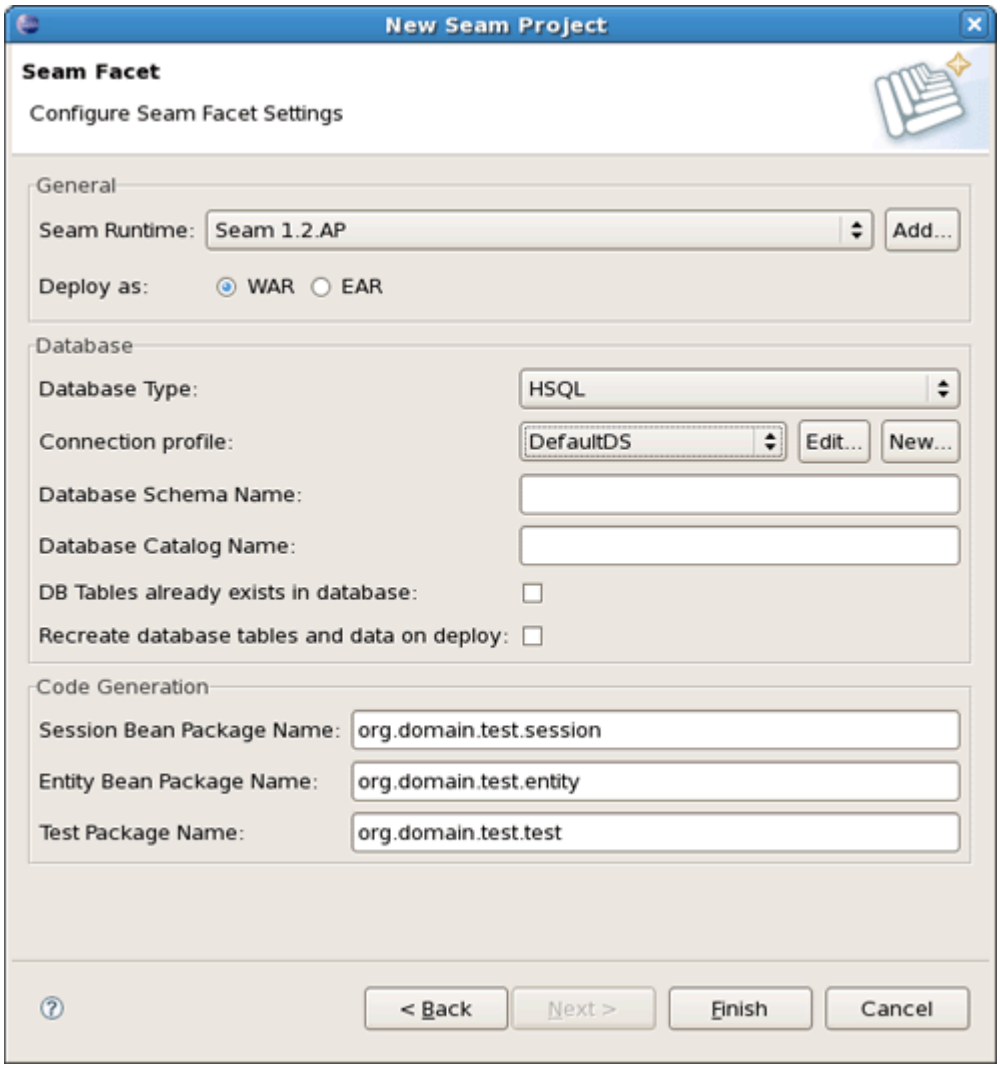


Figure 2.18. Seam Facet Settings

- Create Seam runtime and define Seam home folder.

For that click on *Add* button in the General section. Notice that in this wizard presented below you can create a Seam runtime only for that version which was selected in the *Project Facets* wizard (version 1.2 in our case).

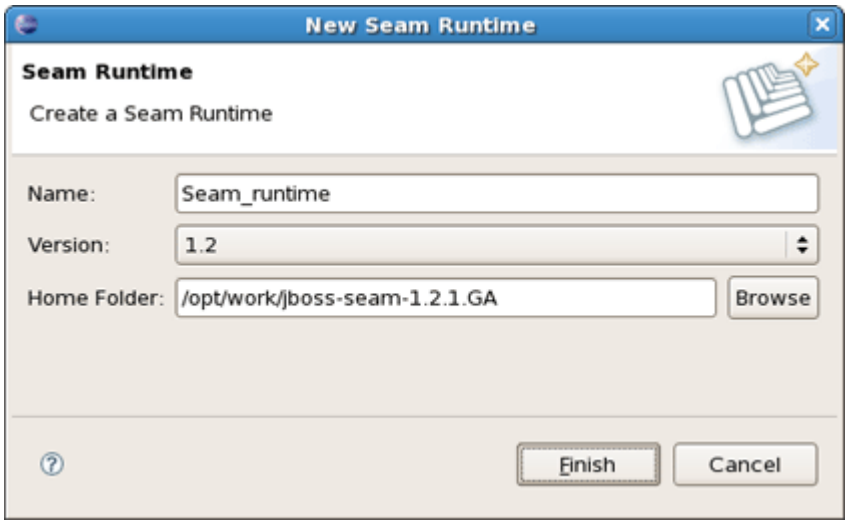


Figure 2.19. Seam Runtime Creation

- Select EAR or WAR deployment by checking a necessary radio button.
- Select a *Database Type*

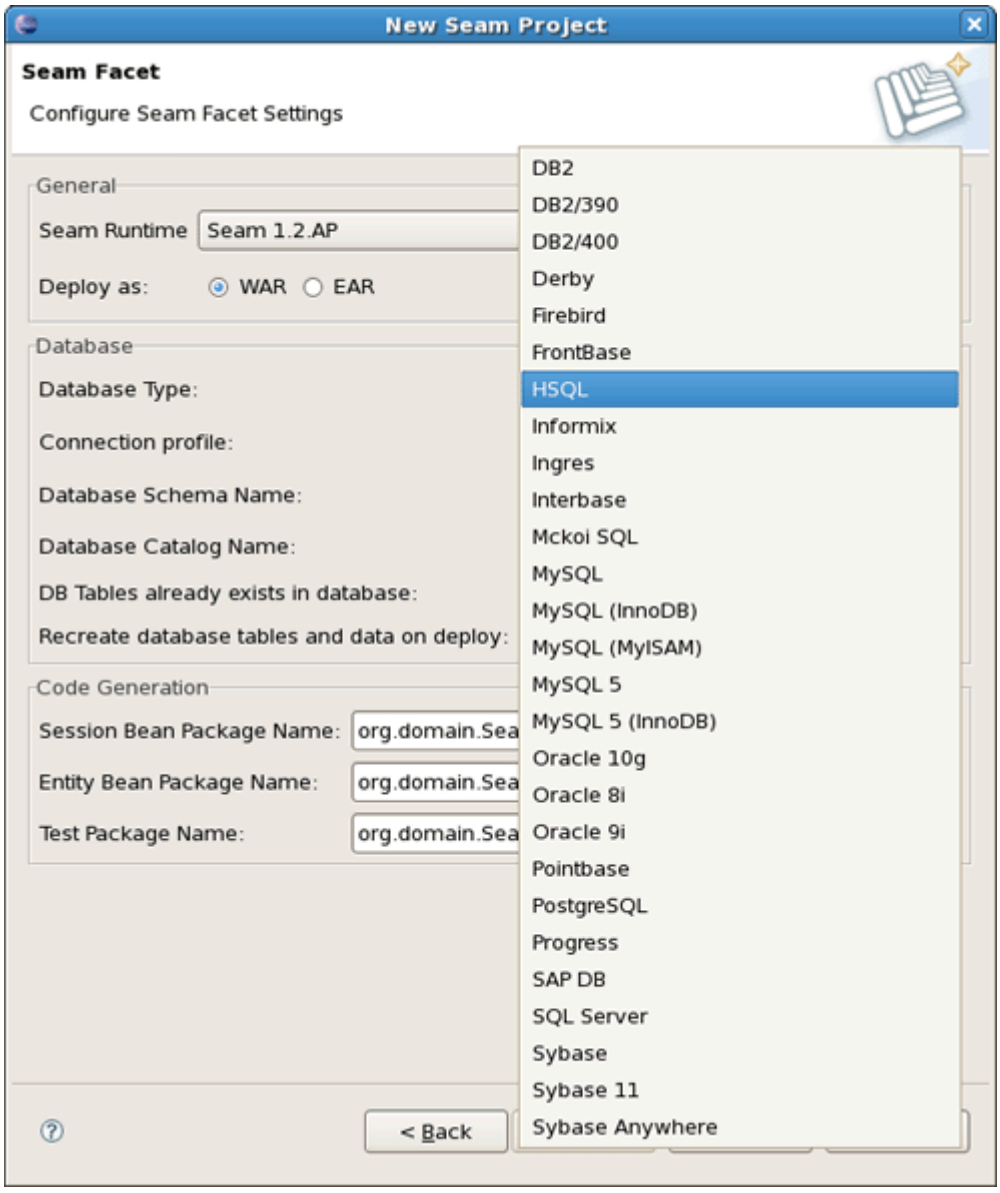


Figure 2.20. Seam Runtime Creation

- and then specify a *Connection profile* appropriate for your database.

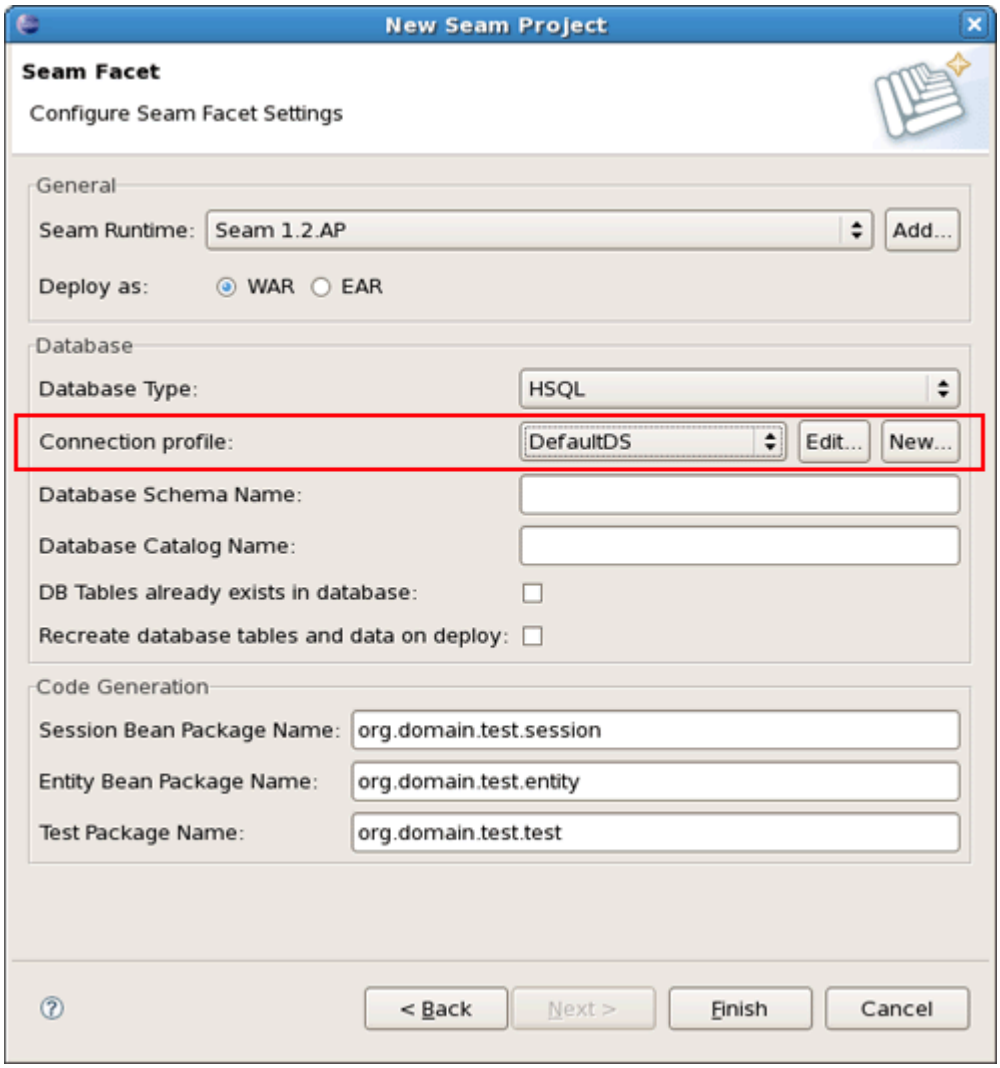


Figure 2.21. Connection Profile Options

You can edit chosen profile by using *Edit* button or organise a new one by clicking on *New* button and selecting necessary for you type of connection profile.

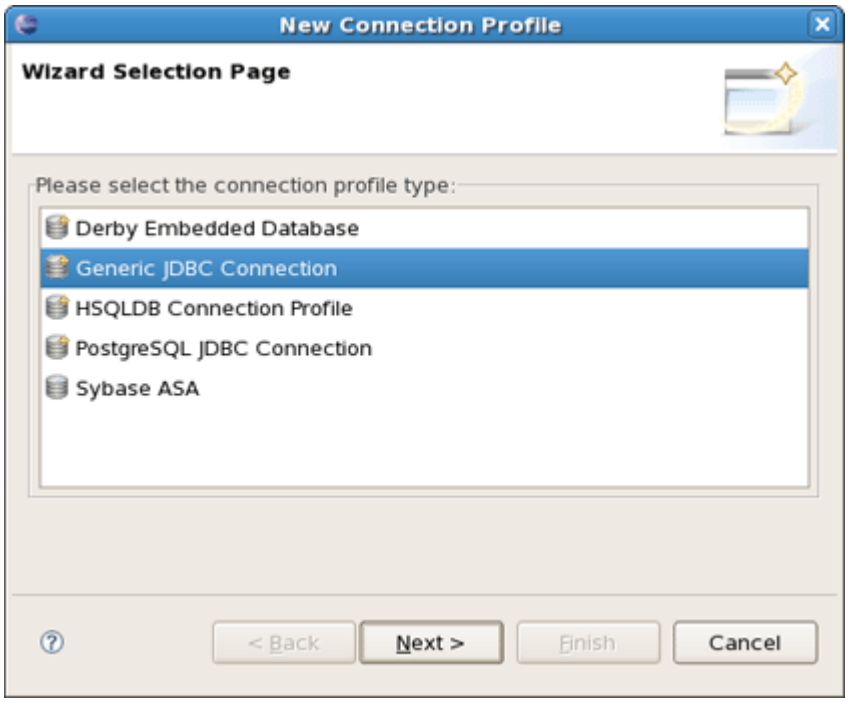


Figure 2.22. Connection Profile Selecting

On the other dialog you'll be asked to enter its name and description. And then you should select a proper driver and adjust connection details. Press *Next* to preview all the adjusted settings and complete the creation of the new profile.

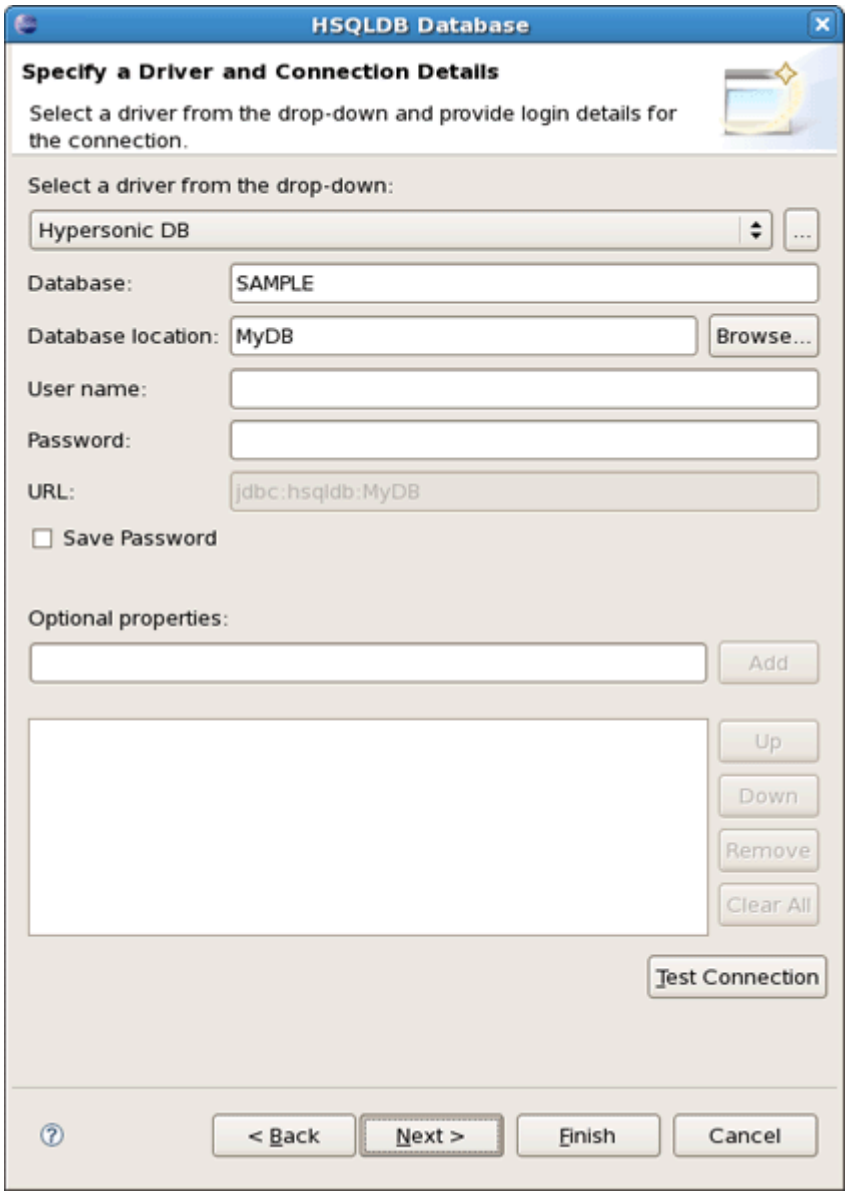


Figure 2.23. Connection Details

The next block of settings in the Seam Facet wizard are describing a Database and a connection to it.

In the Code Generation section the wizard have already put the names for your Session Bean, Entity Bean and Test packages. Of course, you can change them into the others which you like.

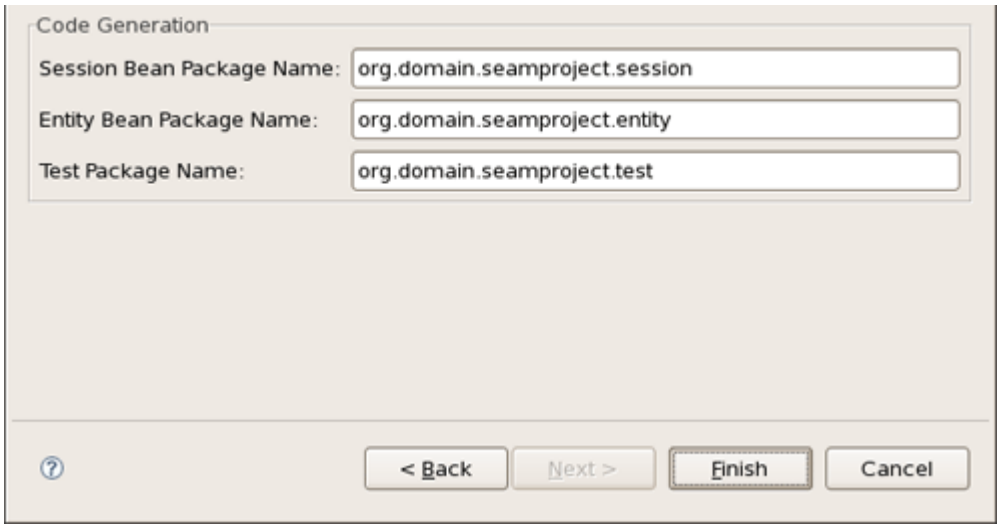


Figure 2.24. Code Generation Section

Click on *Finish* to generate a project.

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.

3.1. WAR Deployment

The project layout for WAR projects is:

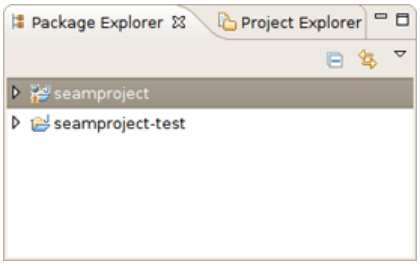


Figure 3.1. Project Layout for WAR projects

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 1.2.1 war hotdeploy feature. Classes put into *src/action* will be deployed to *WEB-INF/dev* from which Seam 1.2.1 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 *TestNG* directly against the proper libraries and server runtime libraries. When the TestNG plugin is installed you can just run your tests via *Run As > TestNG Test*.

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.

3.2. EAR Deployment

The project layout for EAR projects is:

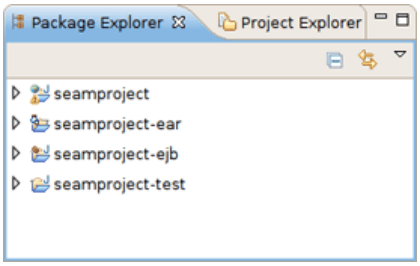


Figure 3.2. Project Layout for EAR projects

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.

3.3. Changing the Seam Version

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

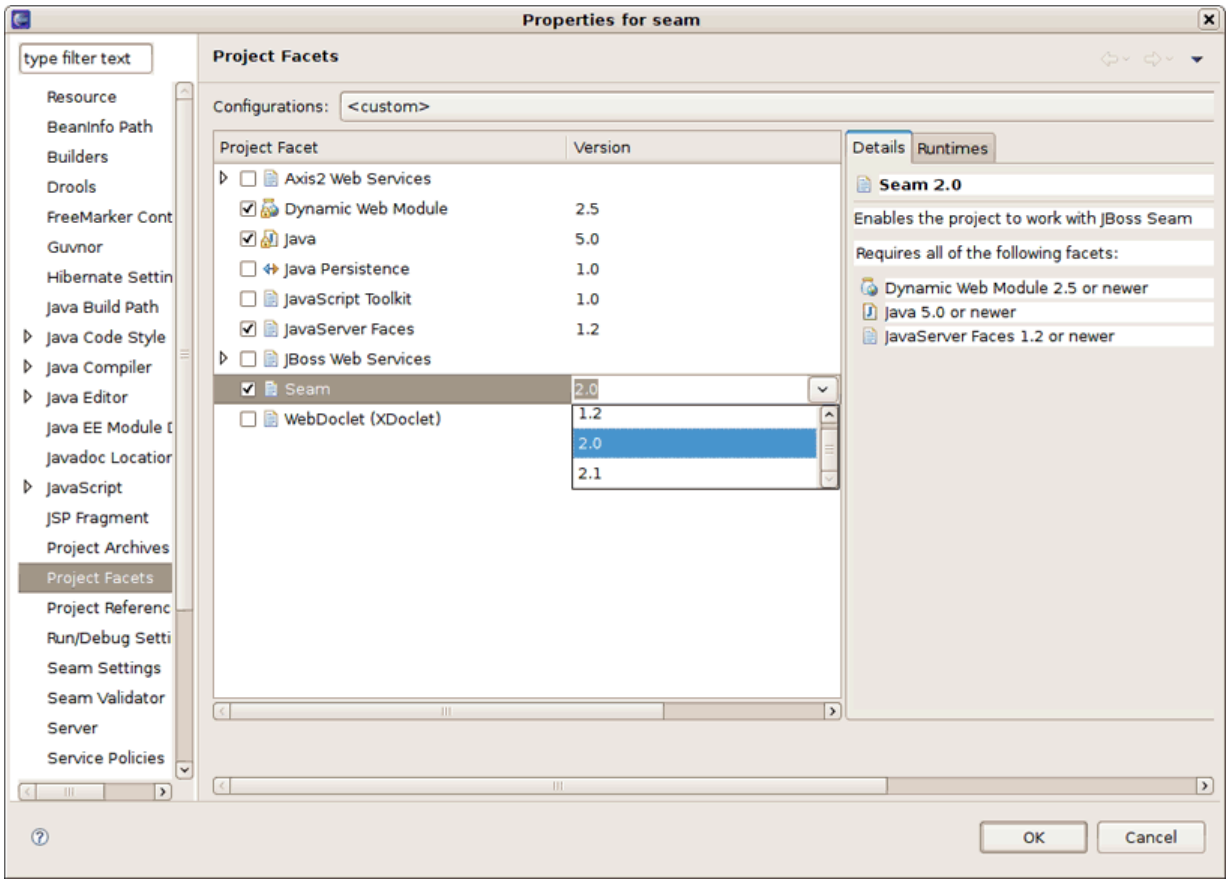


Figure 3.3. Changing the Seam Facet Version

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

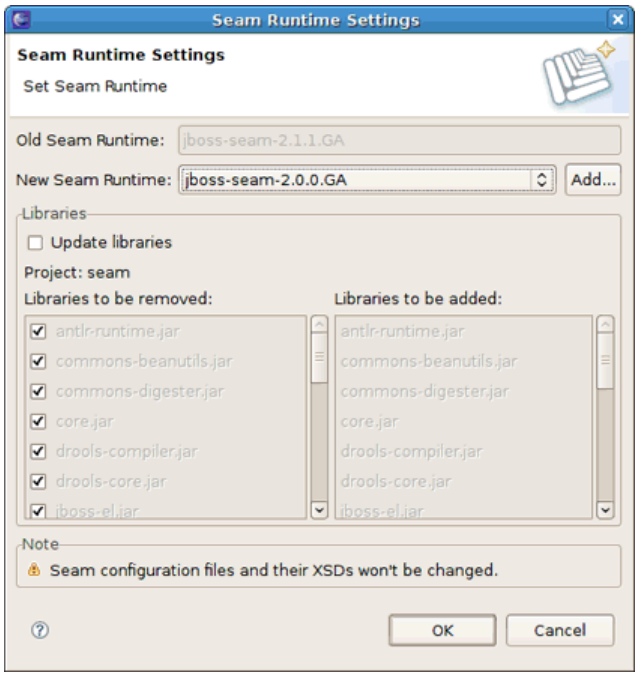


Figure 3.4. Changing the Seam Facet Version

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*.

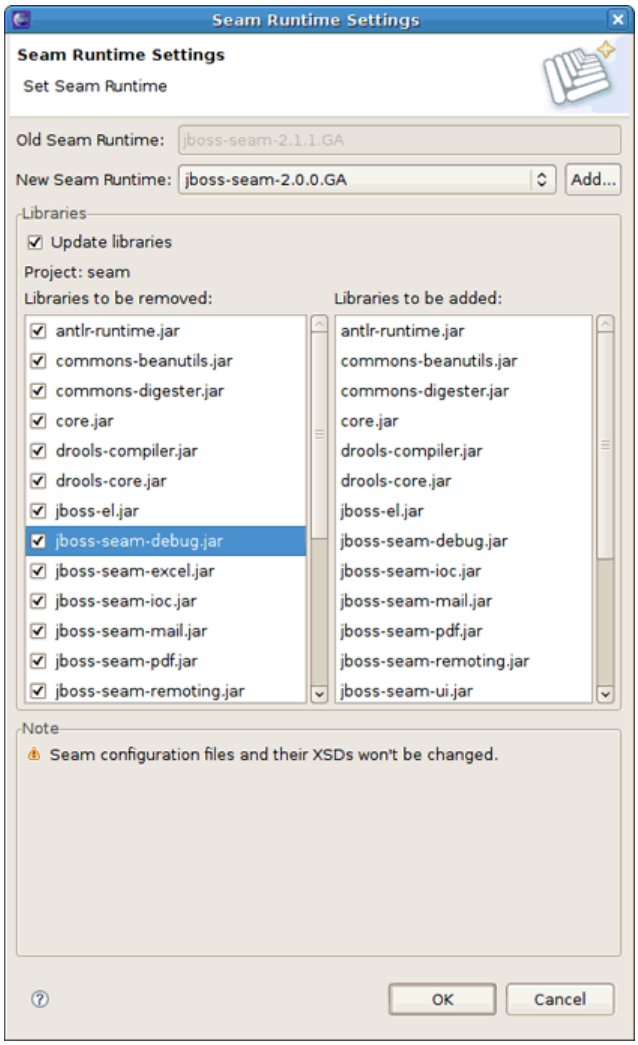


Figure 3.5. Changing the Seam Facet Version

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.

3.4. Changing Seam Parent Project

Starting from the 2.0.0.CR2 version of JBoss Tools it is possible to change the Seam parent project. In earlier versions this was only controllable at a project creation time, now it can be changed for existing projects too.

Go 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.

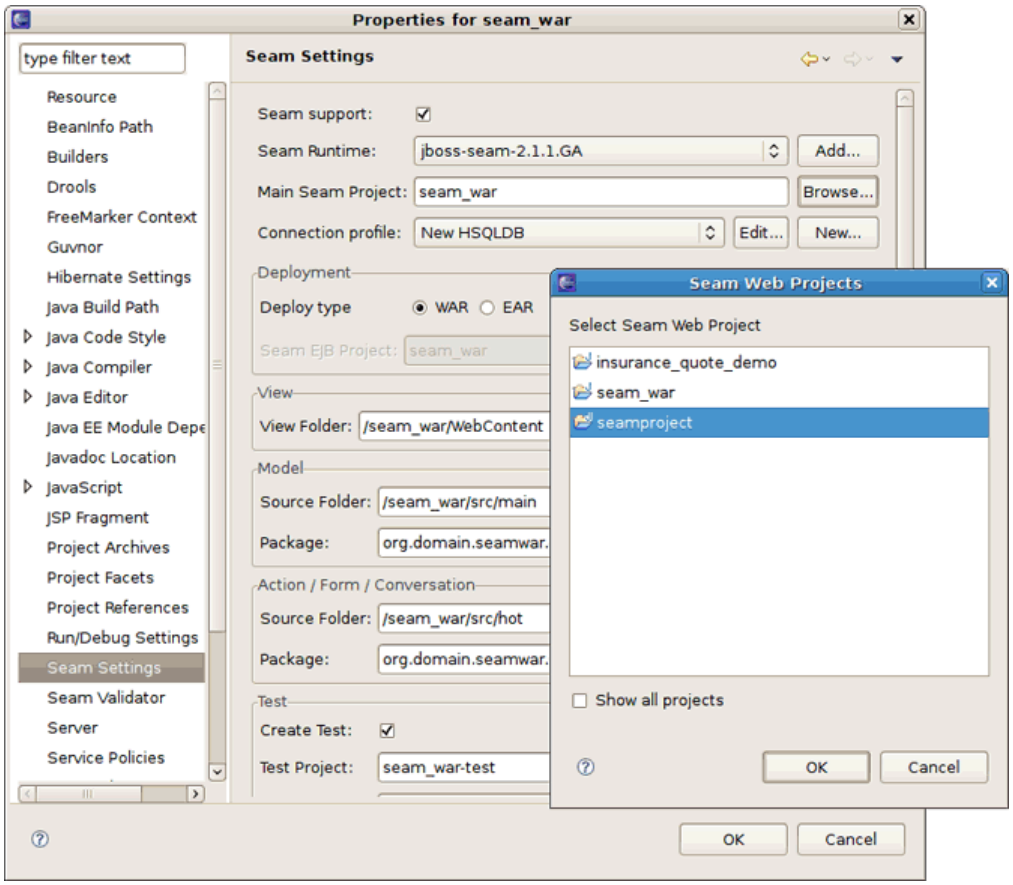


Figure 3.6. Changing Seam Parent Project

3.5. Renaming the Projects and Folders

If you need to rename one of the Seam Project artifacts (*<project_name>*, *<project_name>-test*, *<project_name>-test* or *<project_name>-ejb*) 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 brining the context menu and navigating *Refactor > Rename...* or just pressing *Shift + Alt + R* under the chosen resource.

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.

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

4.1. File Menu Actions

In a Seam perspective, by default there are the following actions in *File > New* submenu

Table 4.1. Seam Actions in the New Submenu

Name	Function
Seam Web Project	Runs *New Seam Project* wizard for creating a new Seam project
Seam Action	Runs *New Seam Action* wizard for creating a new Seam action
Seam Form	Runs *New Seam Form* wizard for creating a new Seam form
Seam Entity	Runs *New Seam Entity* wizard for creating a new Seam entity
Seam Conversation	Runs *New Seam Conversation* wizard for creating a new Seam conversation
Seam Generate Entities	Runs *Generate Seam Entities* wizard

4.2. 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.

4.2.1. Find Seam References/Declarations

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

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

Table 4.2. Find Seam References/Declarations actions

Name	Function	Keyboard Shortcut
		Ctrl+G

Name	Function	Keyboard Shortcut
Find Seam References Find Seam Declarations	Find all references and declarations to the selected element. It's available for EL expressions in both .java 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 4.1. Find Seam Declaration for "Identity"

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

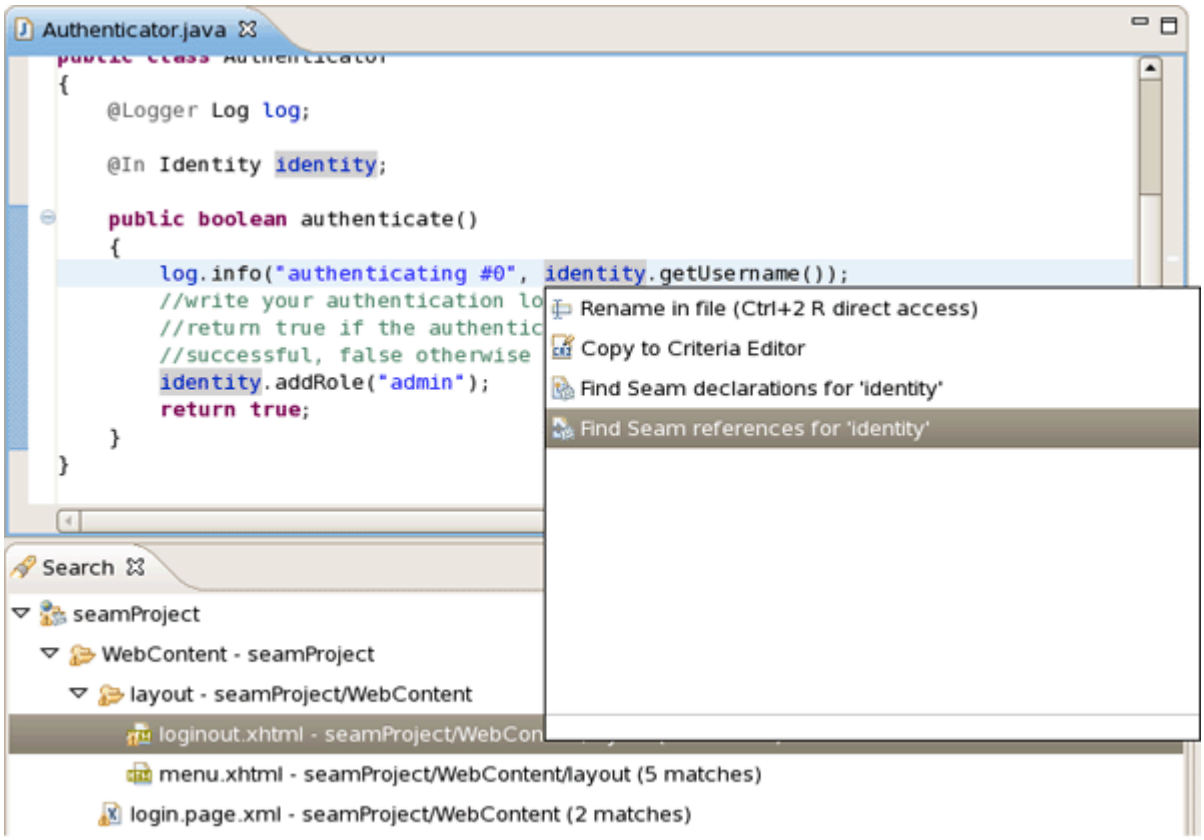


Figure 4.2. Find Seam References for "Identity"

4.2.2. 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 4.3. Open Seam Components icon

In the table below read a description about the dialog.

Table 4.3. Open Seam Components Dialog

Name	Function	Keyboard Shortcut
Open Seam Component	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:

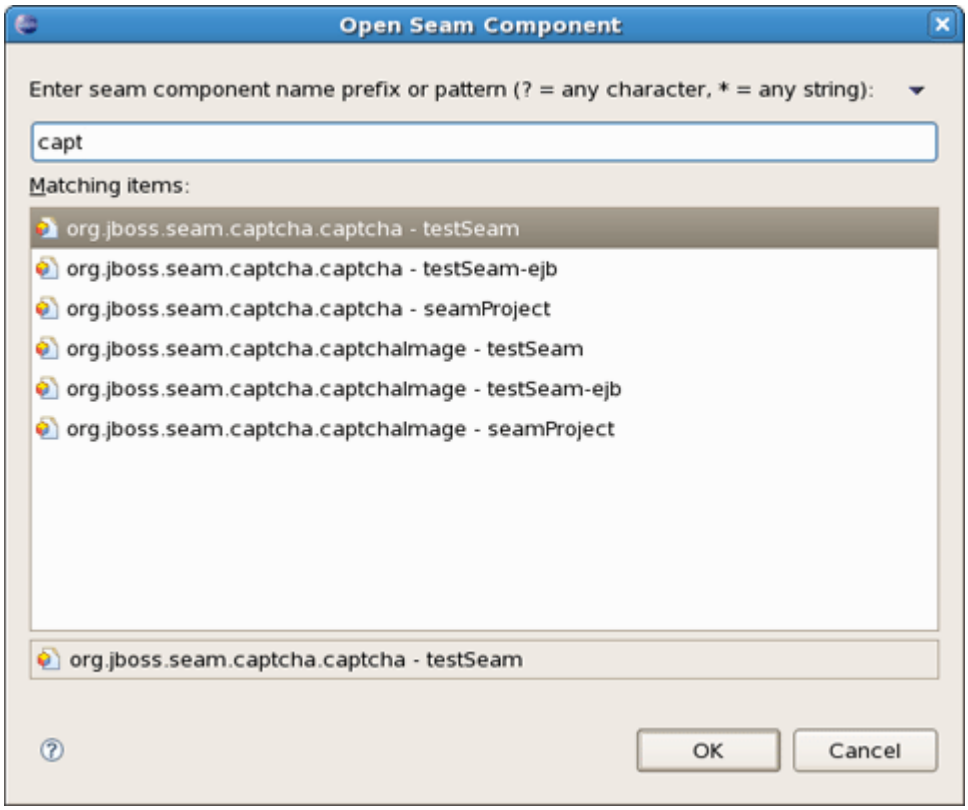


Figure 4.4. Open Seam Components Dialog

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.

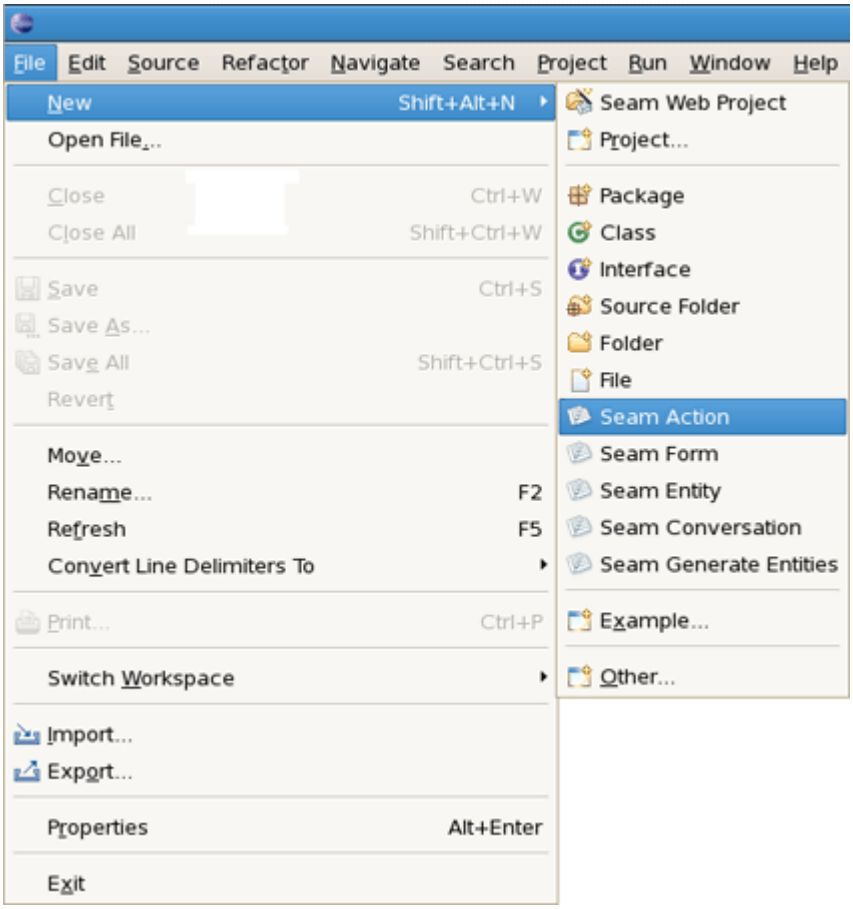


Figure 5.1. Seam Component Wizards

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.

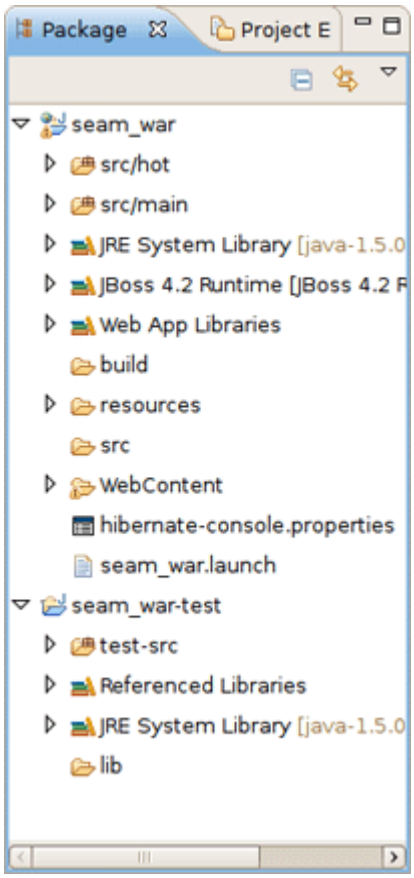


Figure 5.2. Seam Project WAR Deployment

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

5.1. 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.

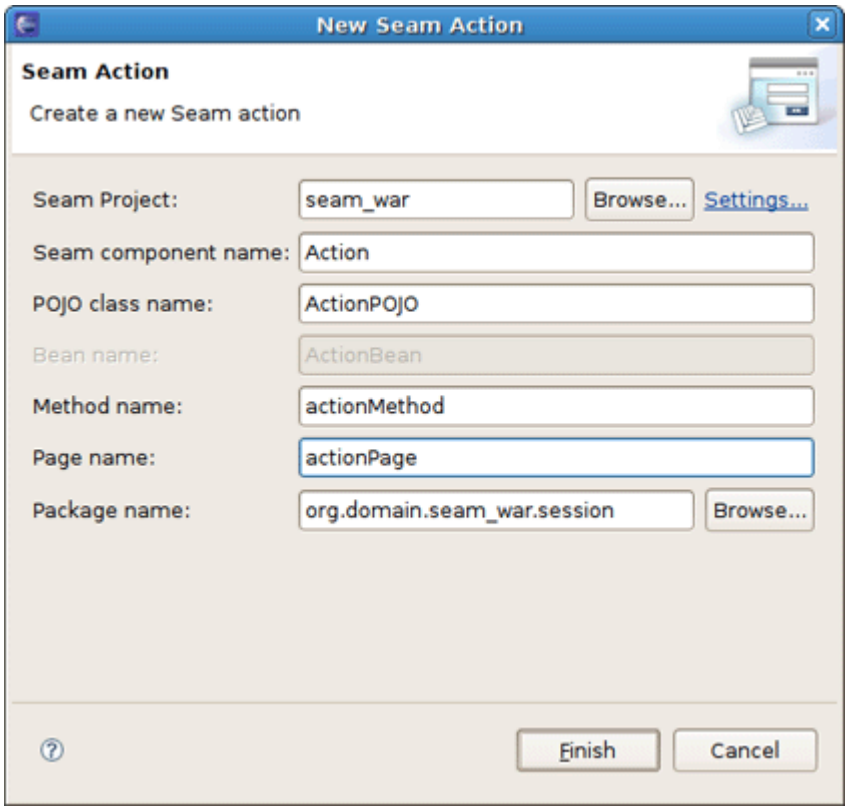


Figure 5.3. New Seam Action Wizard

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

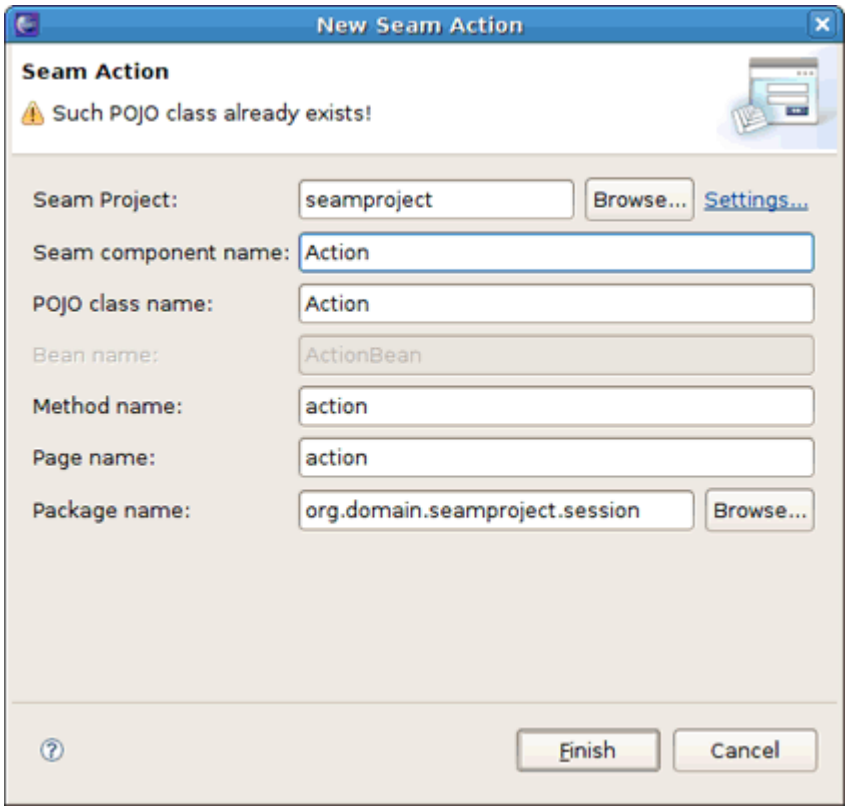


Figure 5.4. Wizard Warning against Existing File

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

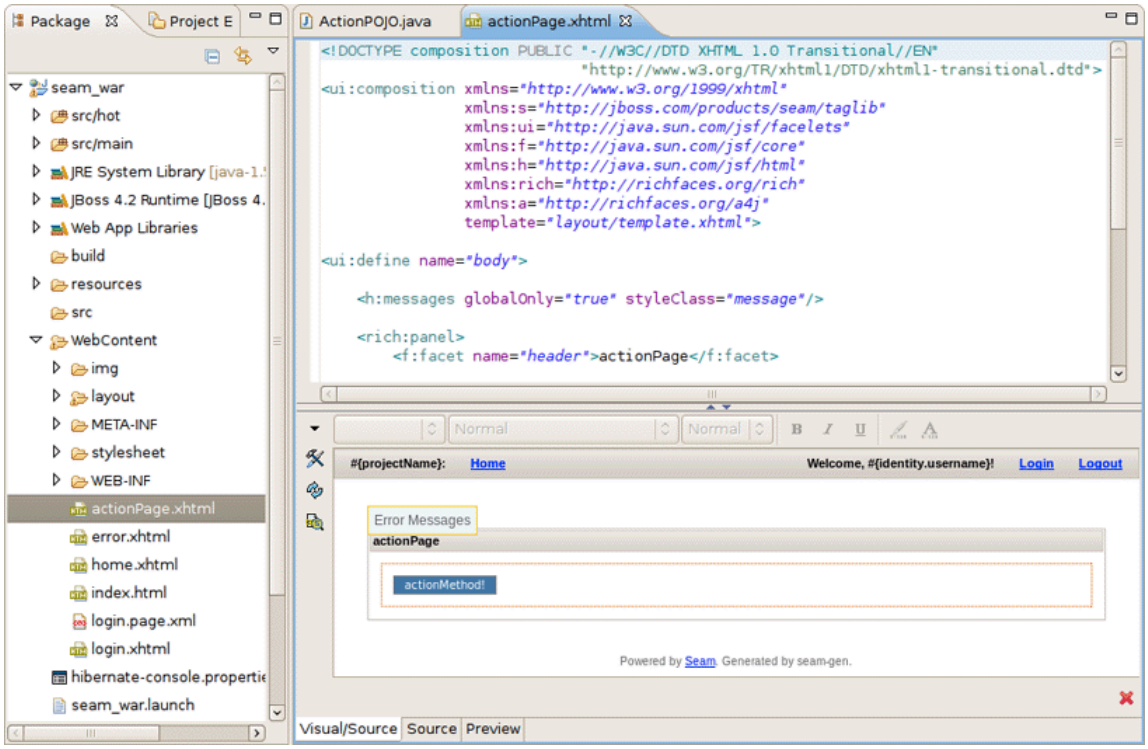


Figure 5.5. Action Page in JBoss Tools HTML Editor.

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.

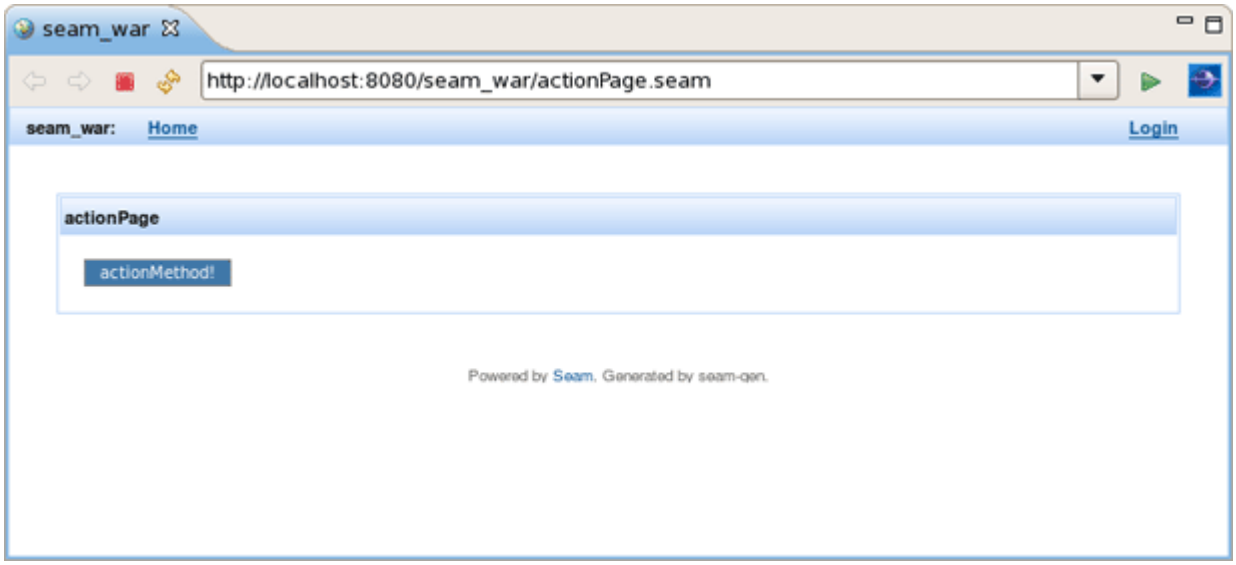


Figure 5.6. Action Component

5.2. New Seam Form

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

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

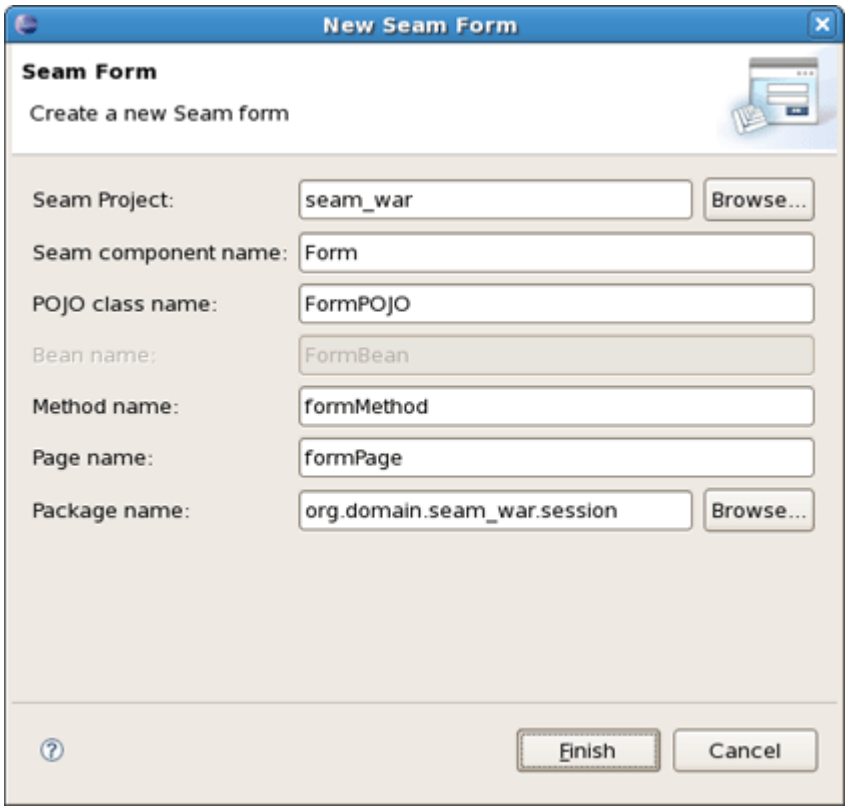


Figure 5.7. New Seam Form Wizard

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

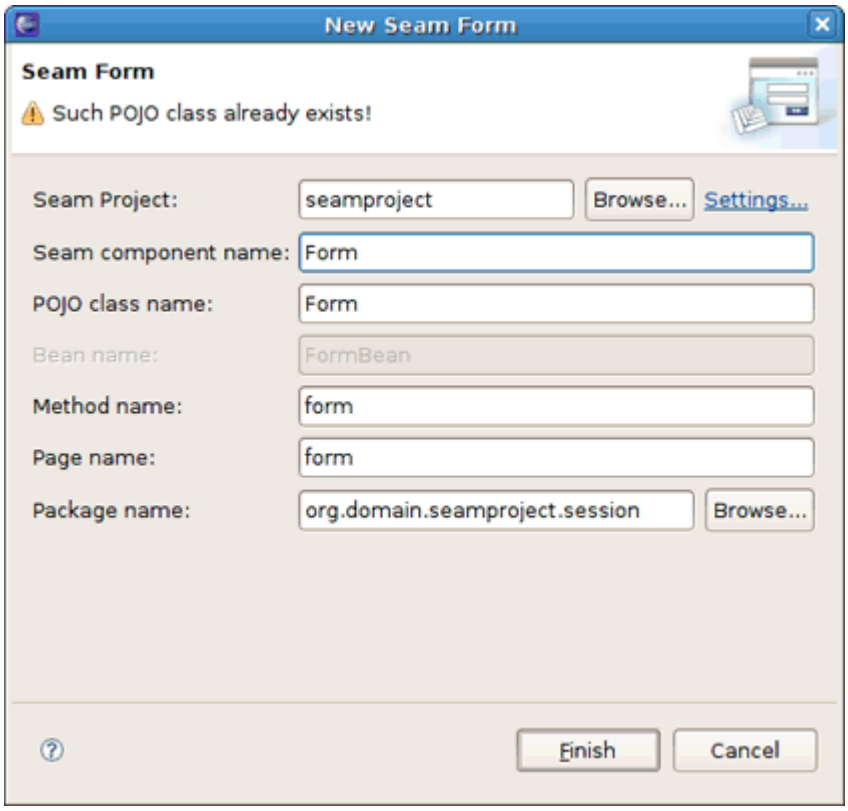


Figure 5.8. Wizard Warning against Existing File

The Form Page was created in WebContent folder.

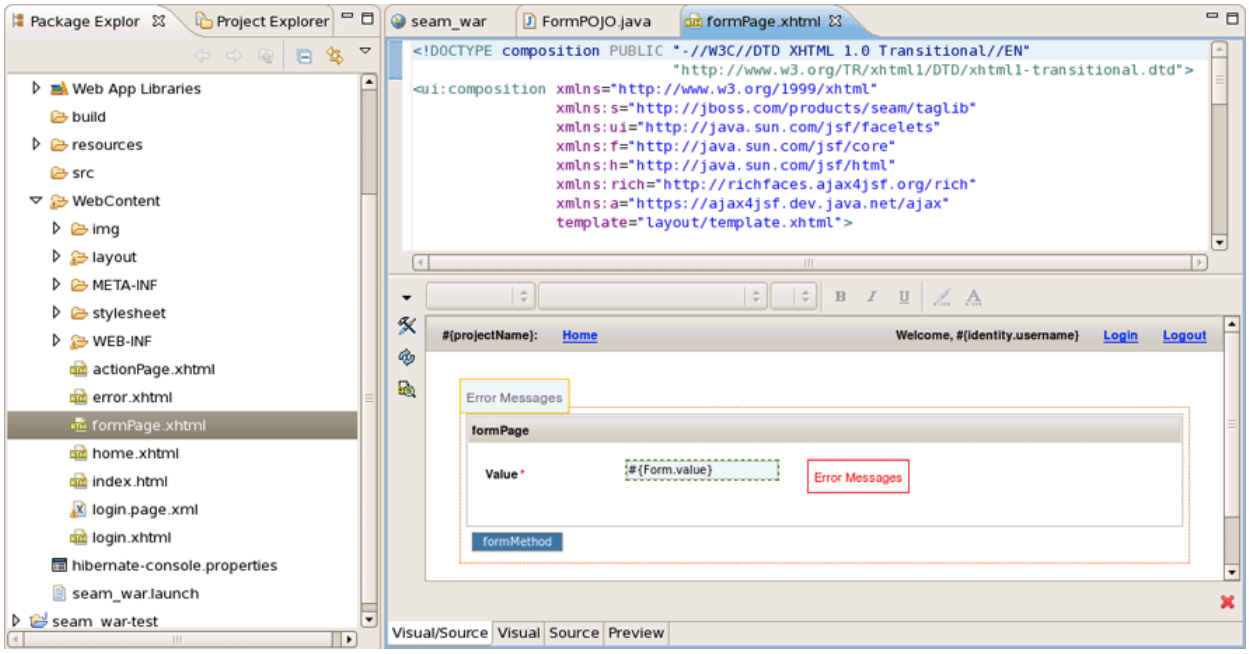


Figure 5.9. Form Page in JBoss Tools HTML Editor.

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

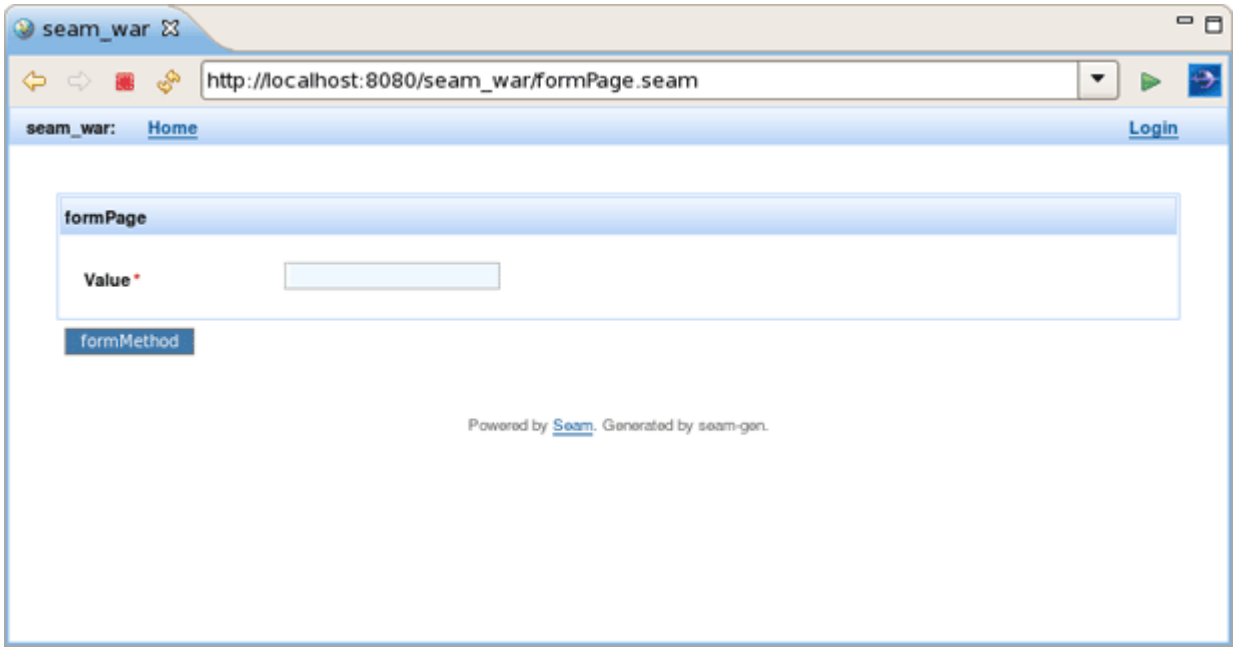


Figure 5.10. Form Component

Form component was hot-deployed.

5.3. New Seam Conversation

Enter some value in the text field (e.g. value1) and click on formMethod.

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.

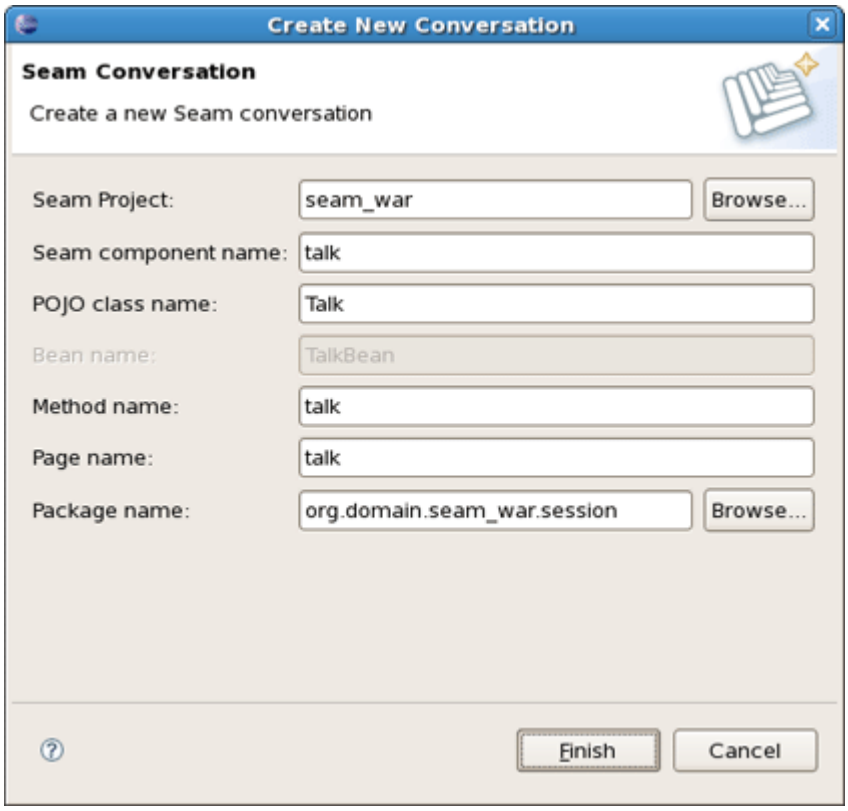


Figure 5.11. New Seam Conversation Wizard

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

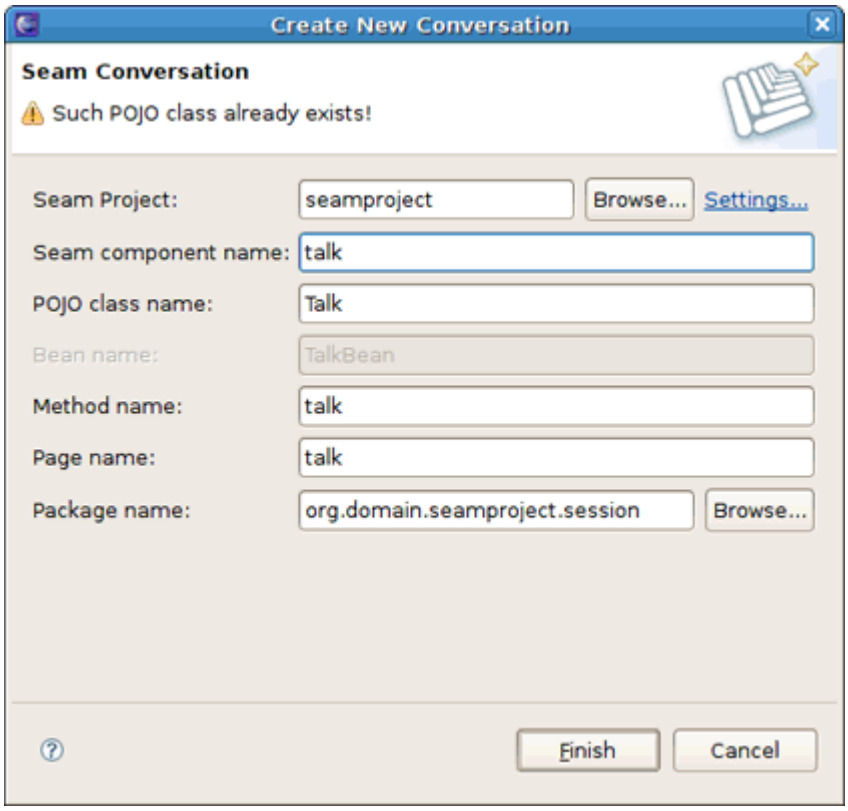


Figure 5.12. Wizard Warning against Existing File

Conversation page was created in WebContent folder.

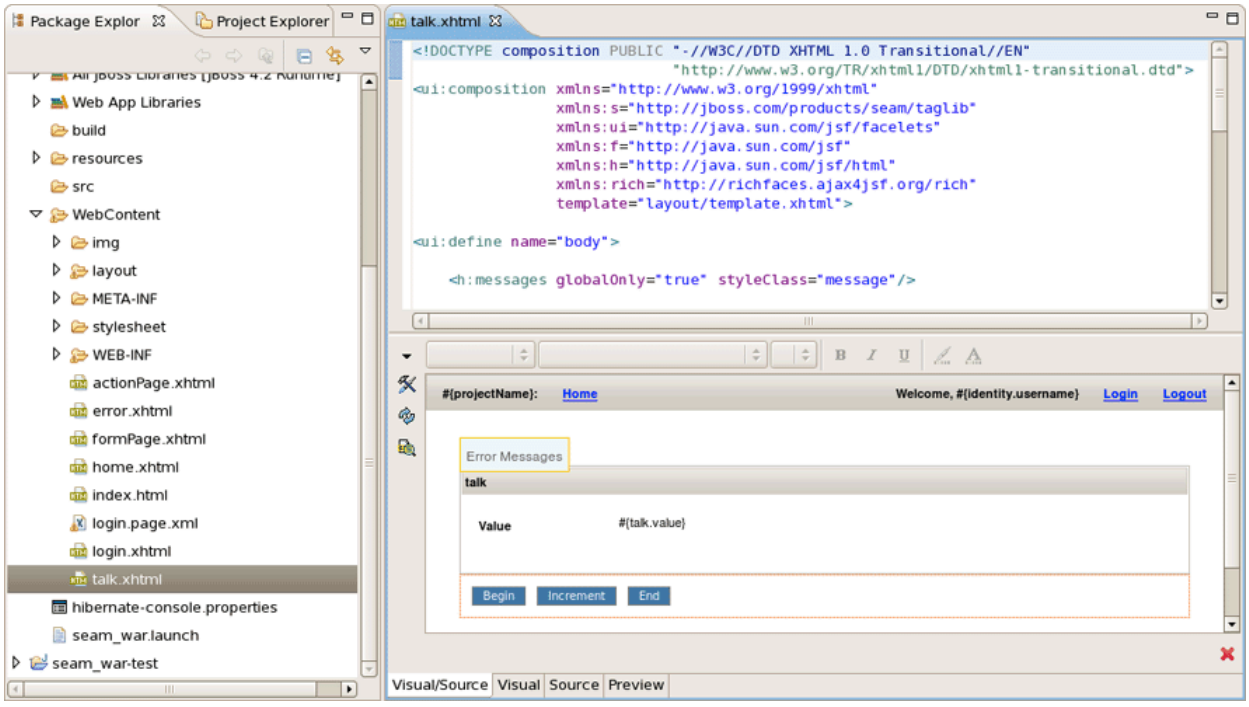


Figure 5.13. Conversation Page in JBoss Tools HTML Editor.

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

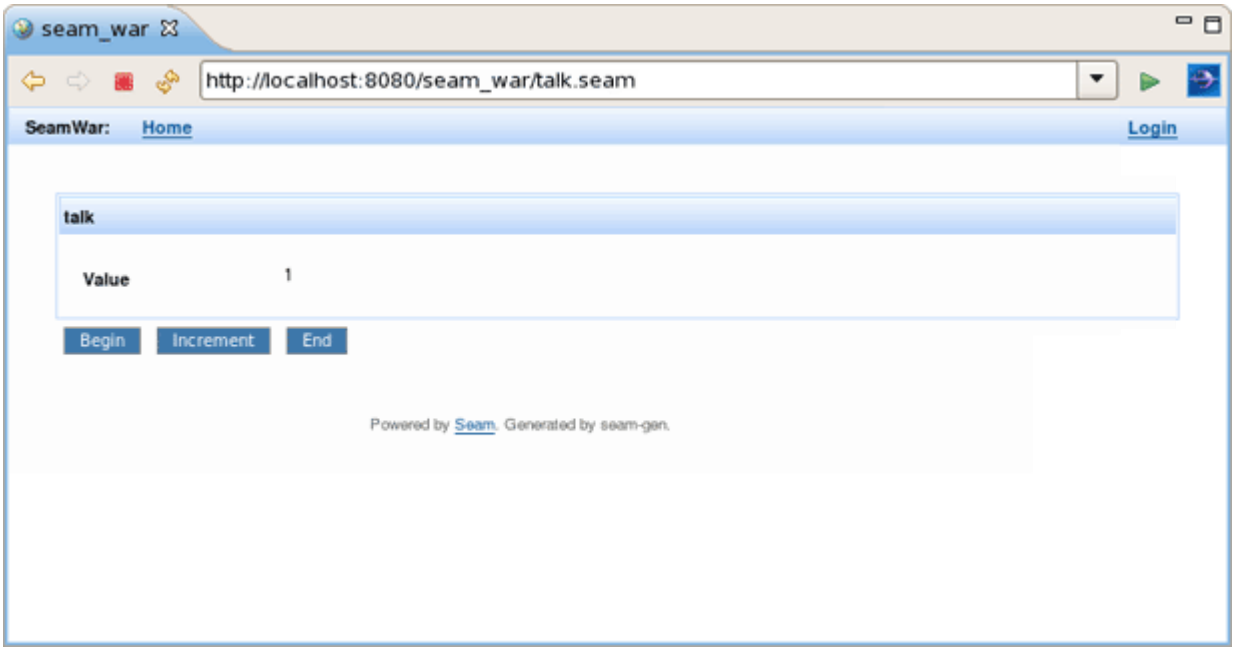


Figure 5.14. Conversation Component

Conversation component was hot-deployed.

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

5.4. 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.

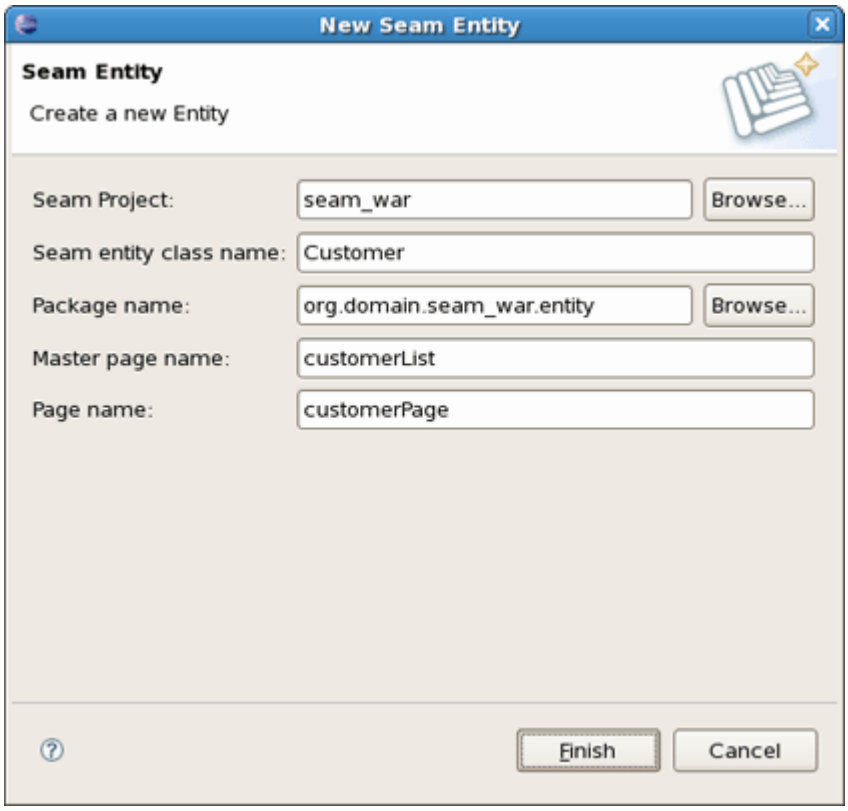


Figure 5.15. New Seam Entity Wizard

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

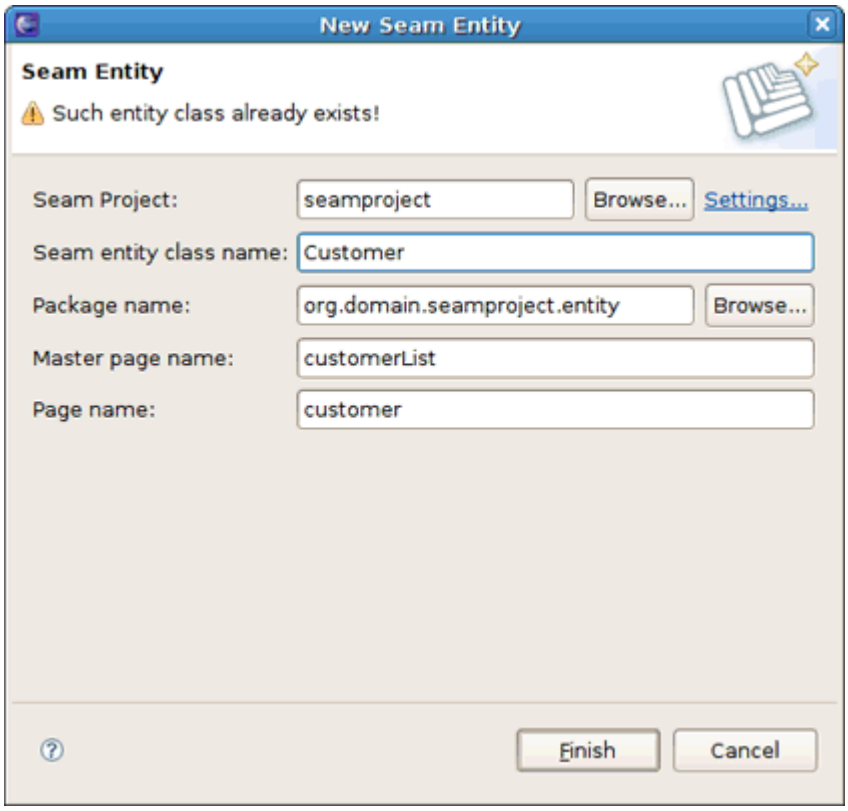


Figure 5.16. Wizard Warning against Existing File

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

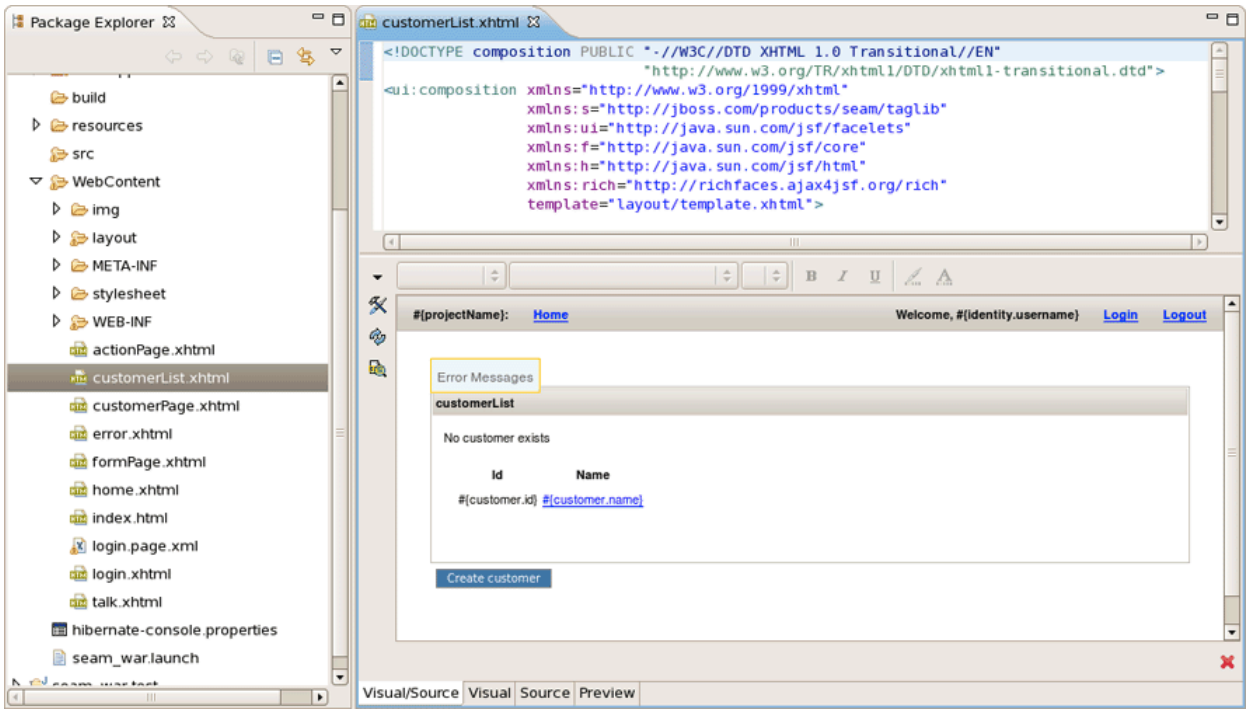


Figure 5.17. Master Page in JBoss Tools HTML Editor.

The Entity page is:

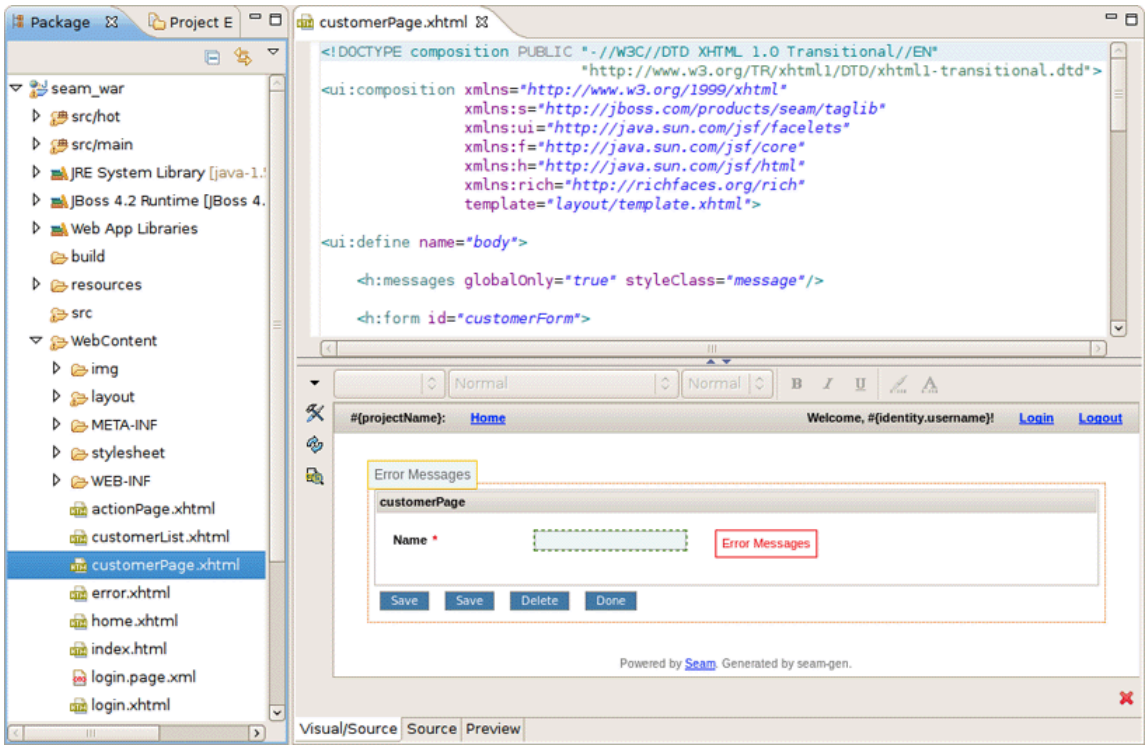


Figure 5.18. Entity Page in JBoss Tools HTML Editor.

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

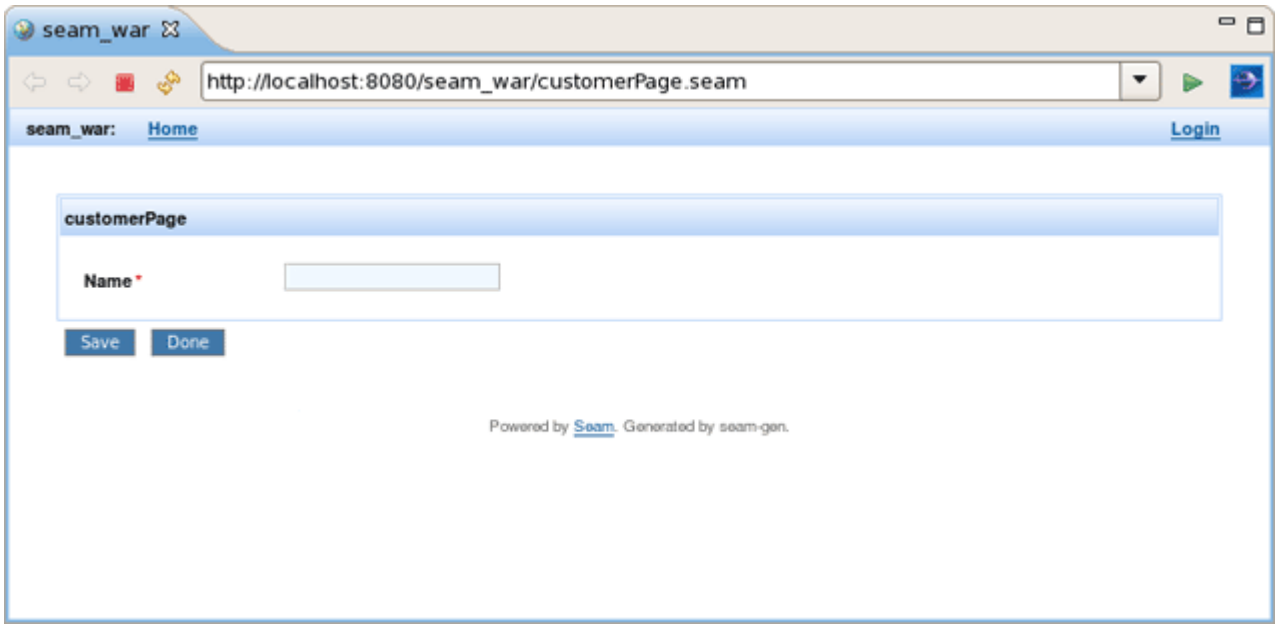


Figure 5.19. Customer Page

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:

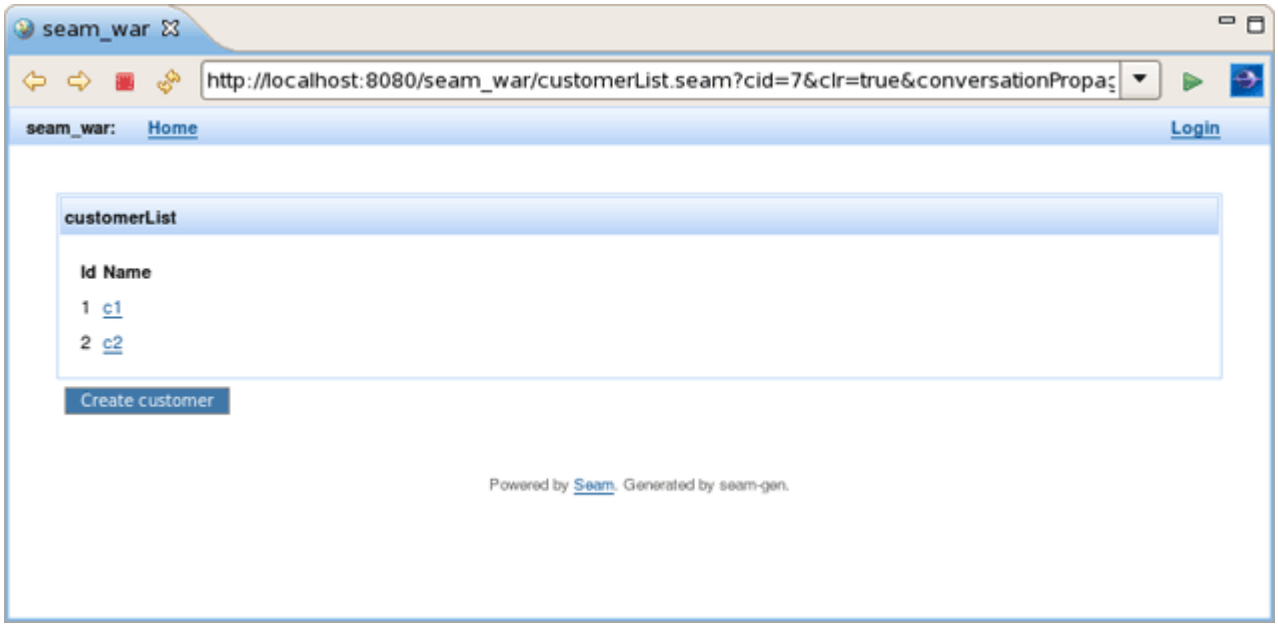


Figure 5.20. Two Customers Are Created

Seam Generate Entities

The main purpose of this chapter is to tell you about Seam Generate Entities.

Generate Entities is available directly from within Eclipse using Hibernate Tools plugin for the standard seam-gen generation.

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

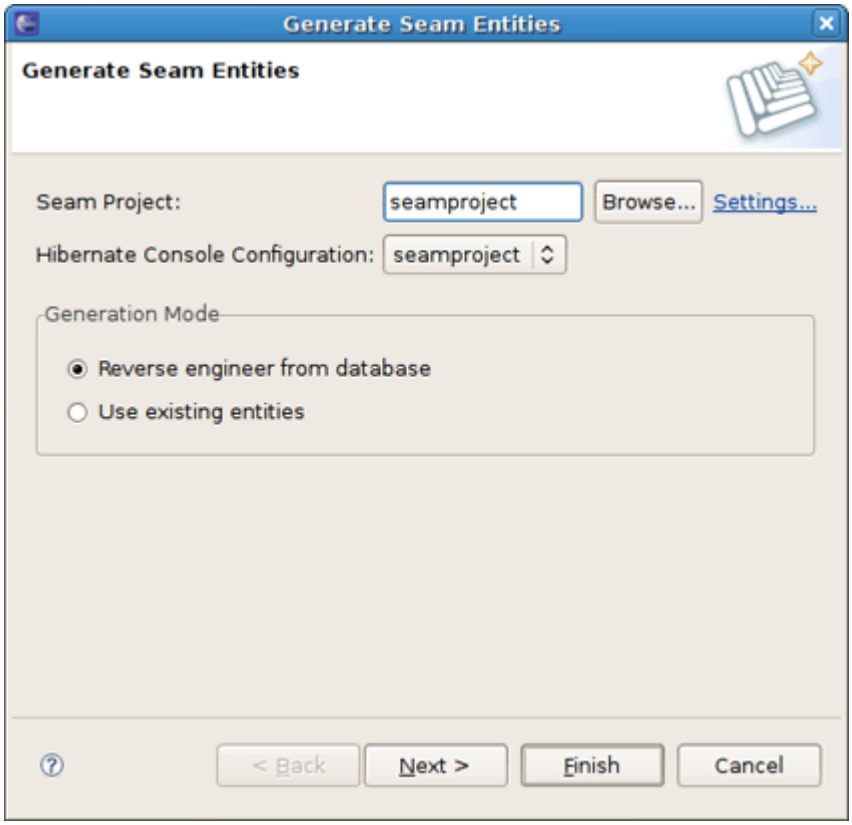


Figure 6.1. Generate Seam Entities Wizard

In the Generate Seam Entities wizard there are two generation modes: Reverse Engineer from database and Use existing entities.

The Reverse Engineer from database mode can be described in four steps:

1. The wizard gets in database, extracts the tables and their connections
2. On basis of this metainfomation 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.

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

Read the *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.

7.1. Visual Page Editor

Visual Page Editor fits perfectly for authoring view Seam pages. The major features of VPE are listed in *Main Features of Seam Editors*

You can also read more about Visual Page Editor in *Visual Page Editor* [http://download.jboss.org/jbosstools/nightly-docs/en/jsf/html/editors.html#visual_page] chapter of "Visual Web Tools Reference Guide".

7.2. 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.

7.2.1. Graphical Mode

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

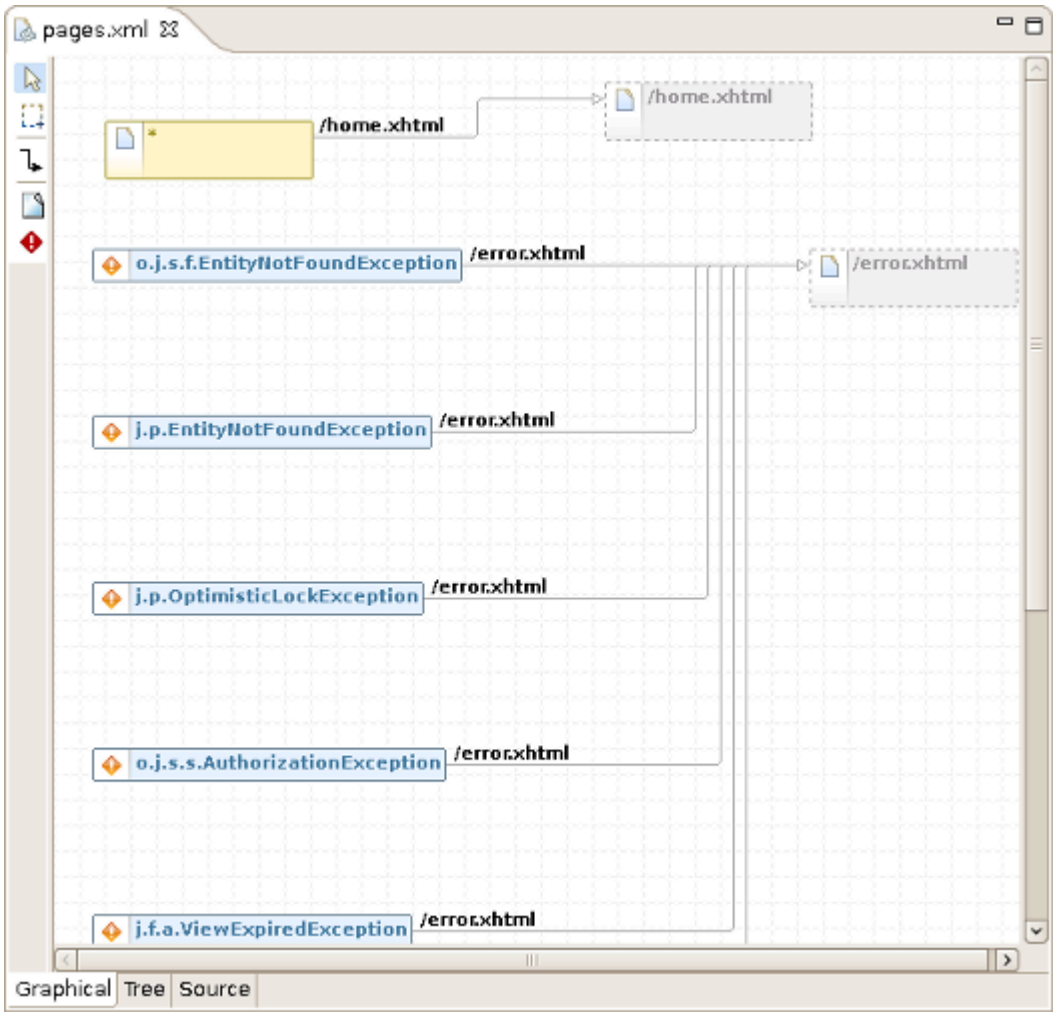
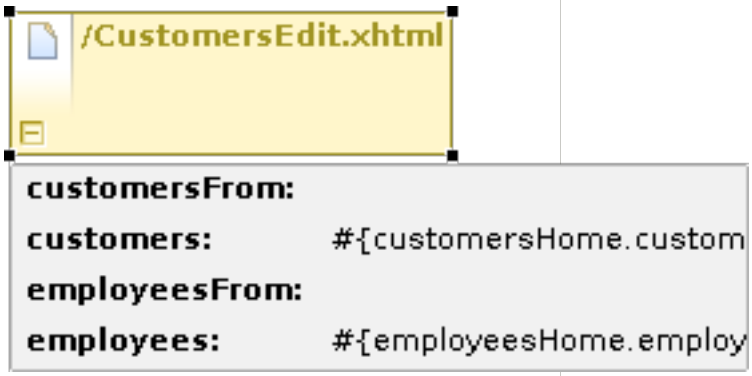


Figure 7.1. Seam Pages Editor: Graphical View

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

Table 7.1. Pages Editor: Graphical View. Visual elements

Element	Description
	A yellow box with a solid border represents a *<page>* element.
	Pressing on the plus icon (on the *<page>* element reveals a box that lists the parameters for the page.

Element	Description
	
	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 <exception> is represented by a blue box with a orange diamond.

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

On the lefthand 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 7.2. 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 file

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

- **Rule** defines a new navigation rule for the element
- **Param** adds a *<param> tag* to the *<page>* element

7.2.2. Tree Mode

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

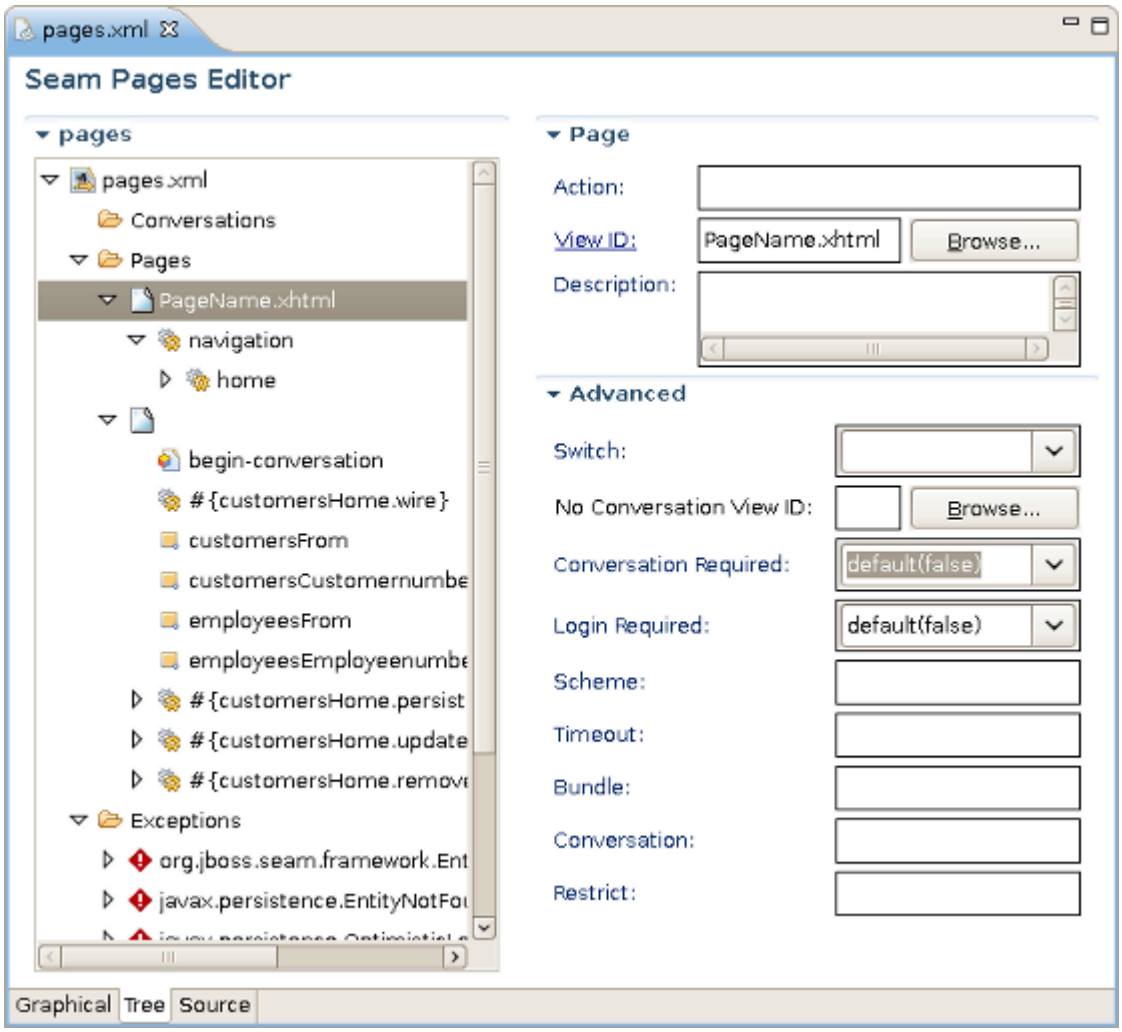


Figure 7.2. Seam Pages Editor: Tree View

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

7.2.3. 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.

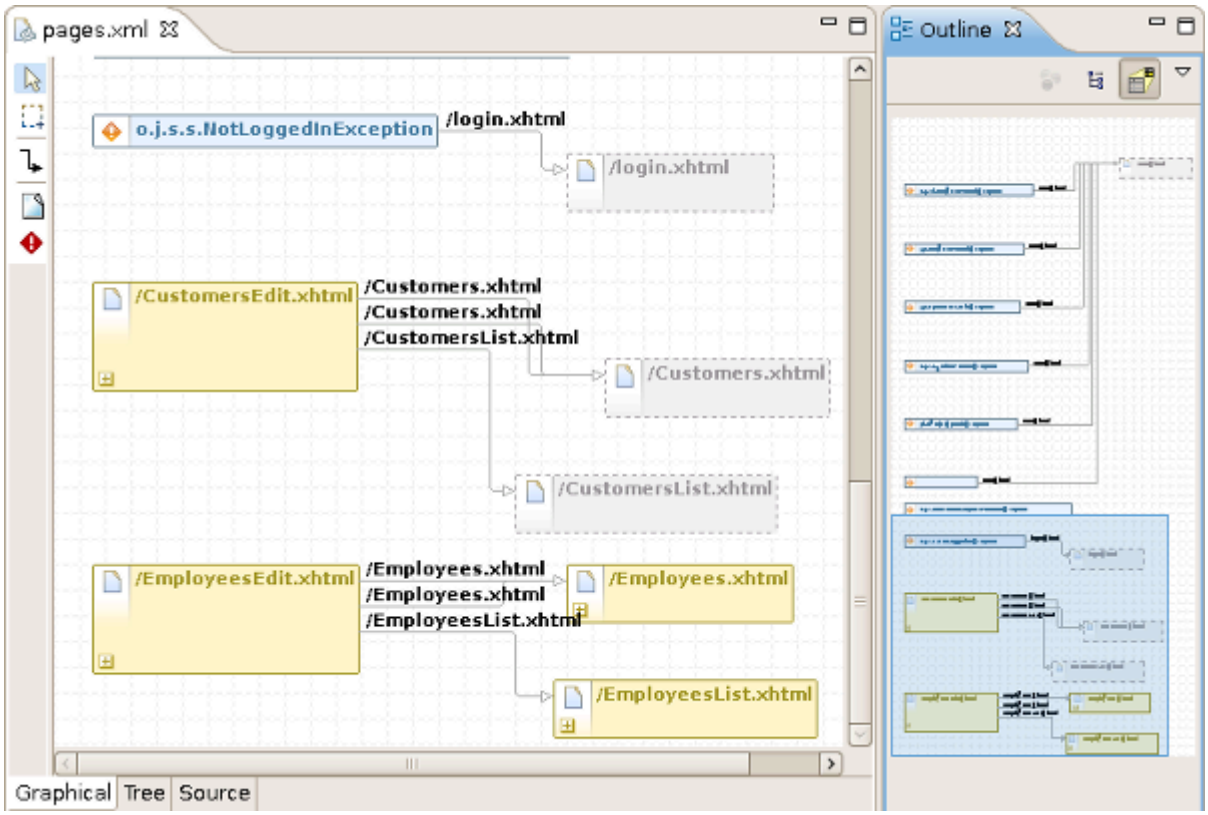


Figure 7.3. Outline view: Diagram Navigation mode

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

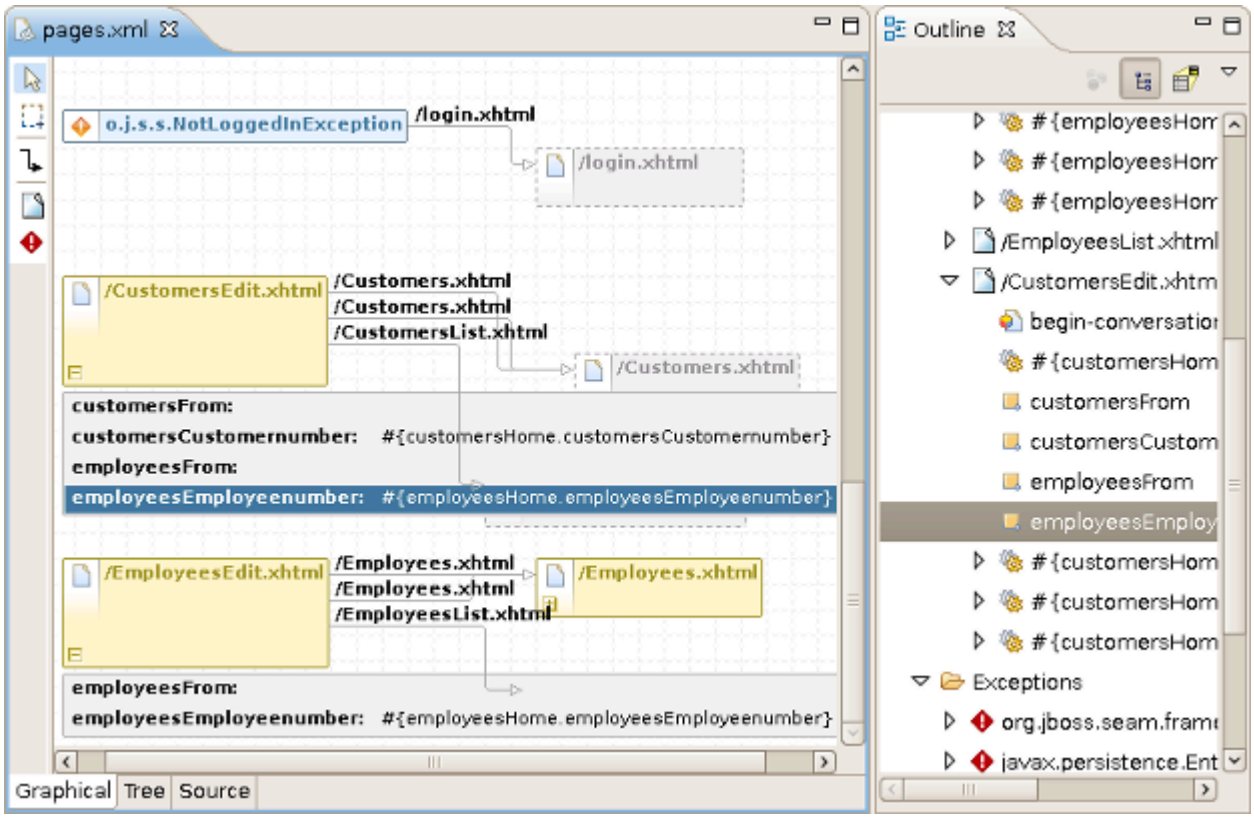


Figure 7.4. Outline view: Tree mode

7.3. 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).

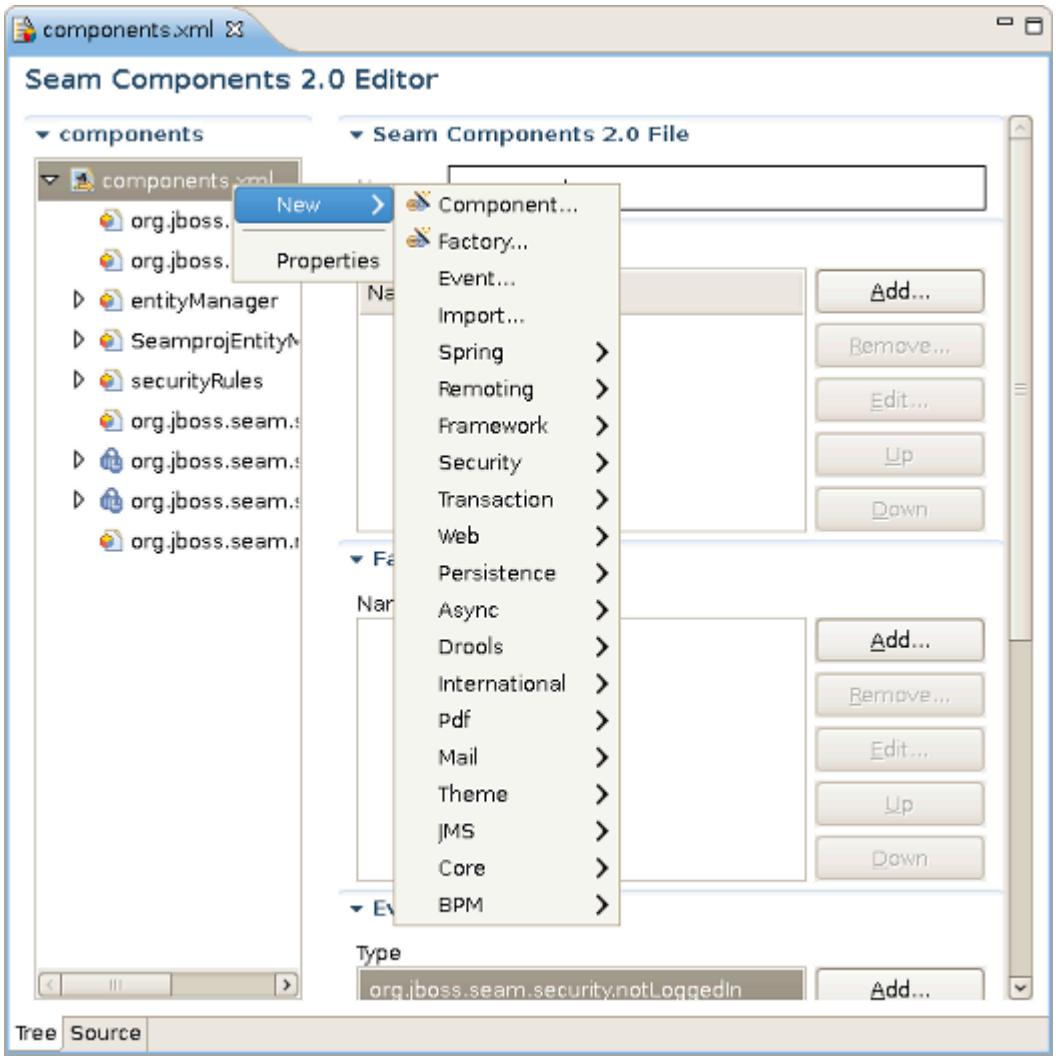


Figure 7.5. component.xml Editor

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.

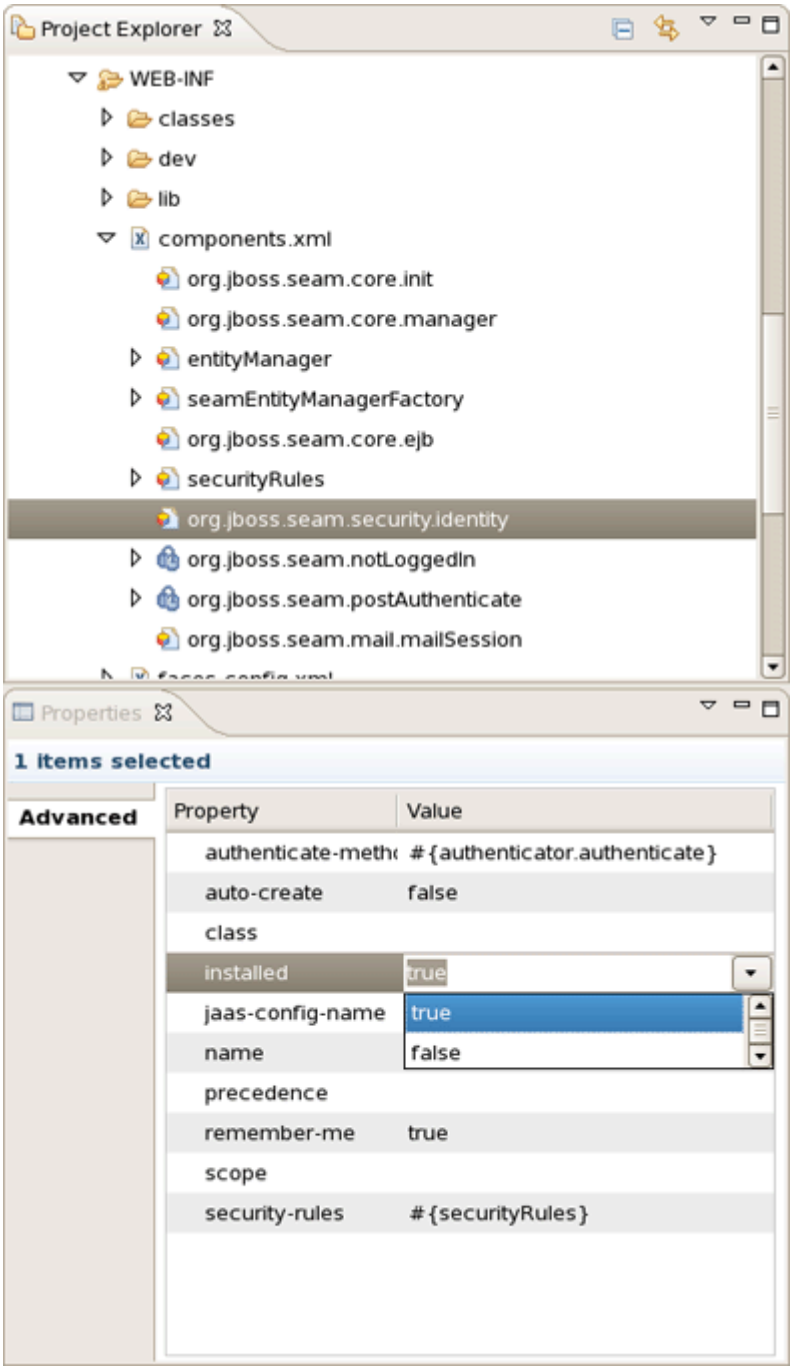


Figure 7.6. component.xml Editor

7.4. Main Features of Seam Editors

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

7.4.1. 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.

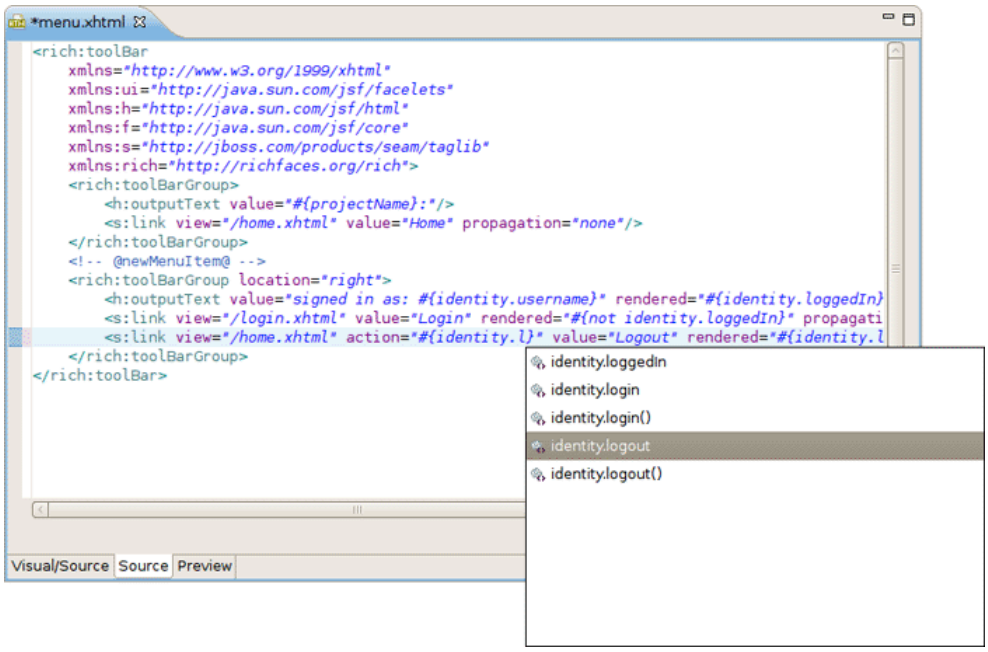


Figure 7.7. Content Assist

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 *Project Preferences*.

7.4.1.1. 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.

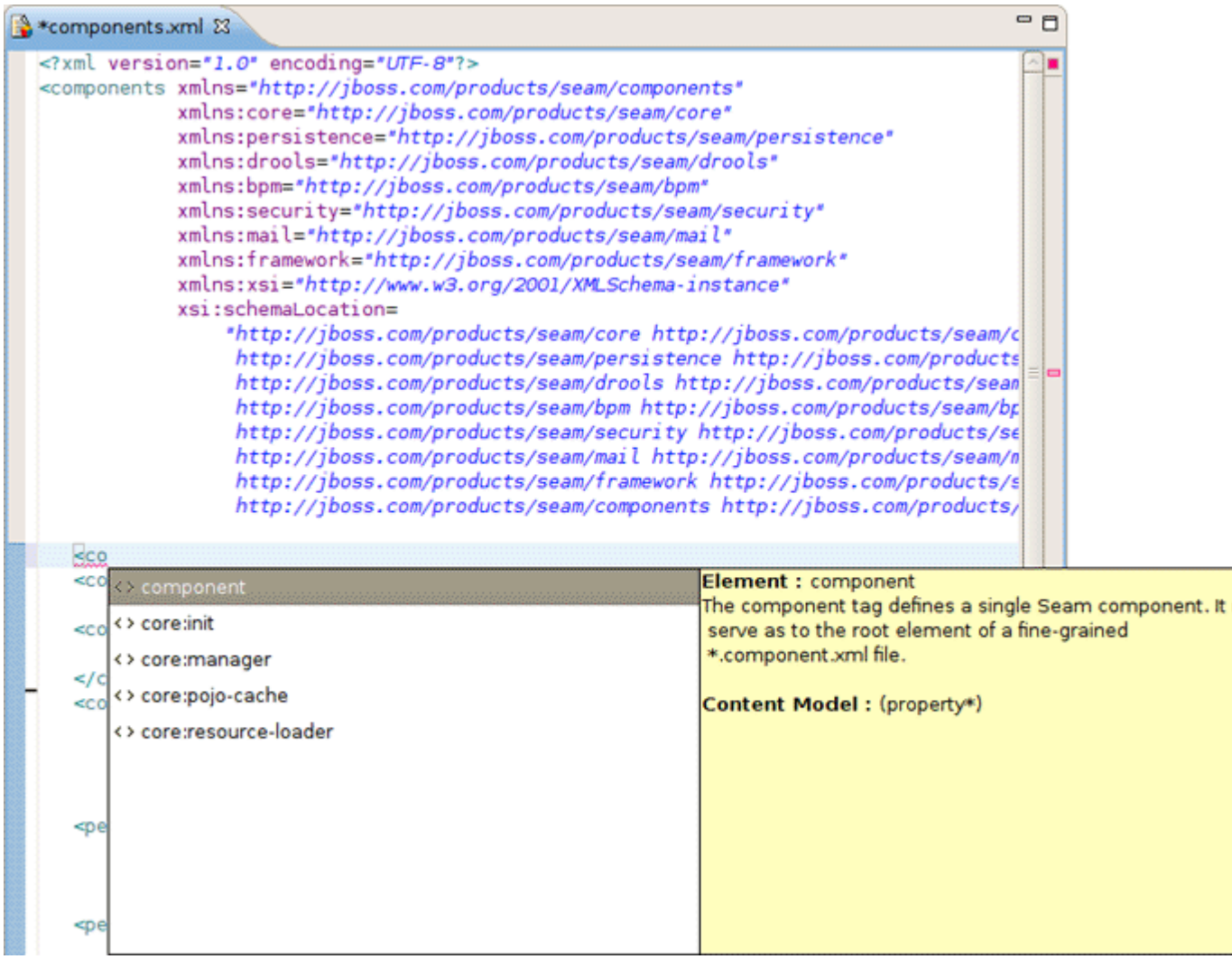


Figure 7.8. Content Assist in the component.xml Editor

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

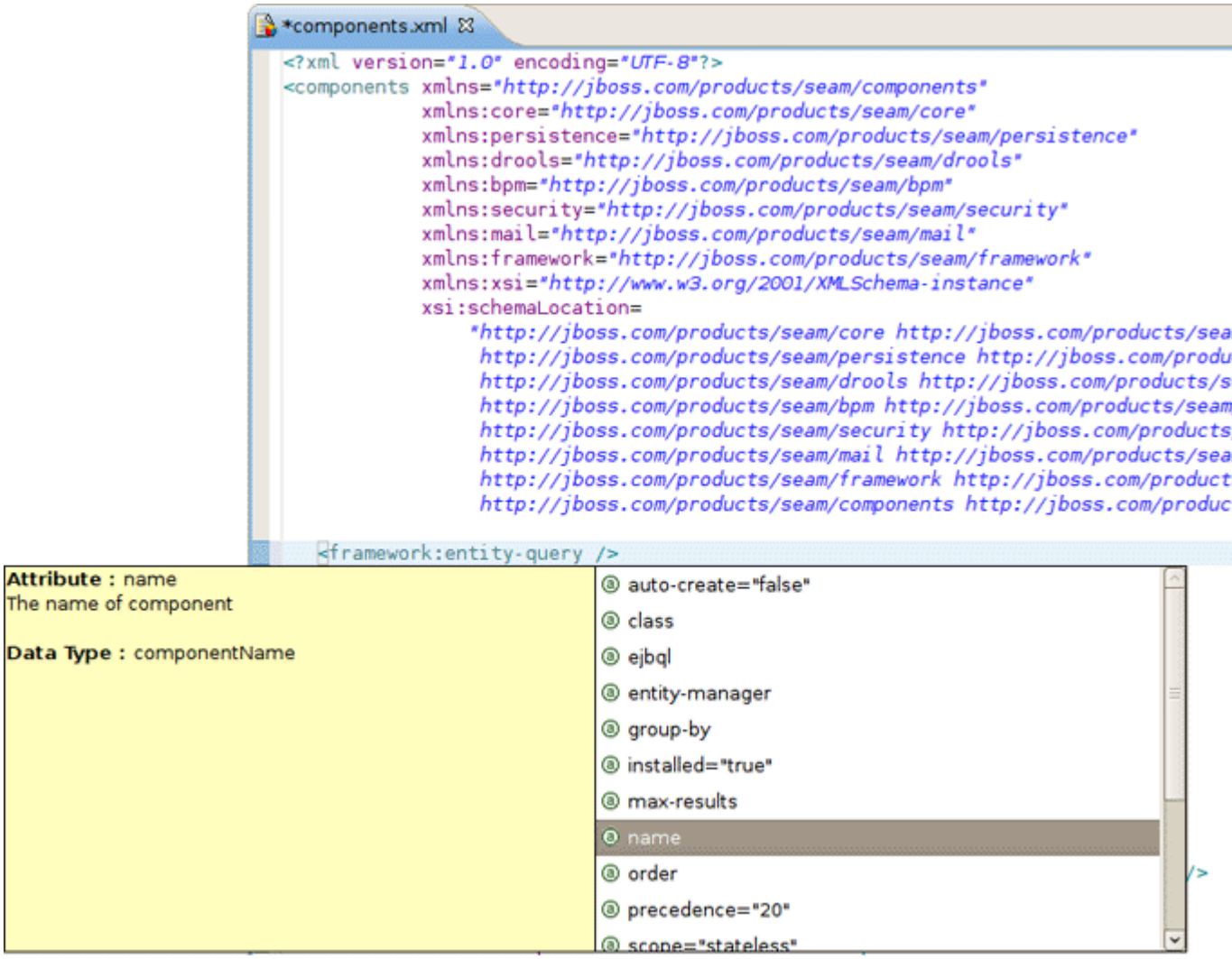


Figure 7.9. Content Assist for EntityQuery Component

7.4.2. OpenOn

OpenOn let's 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 7.10. OpenOn

OpenOn is also supported in Page Descriptors(page.xml). You can Ctrl + left click on the bean to open navigate to the bean you clicked on.

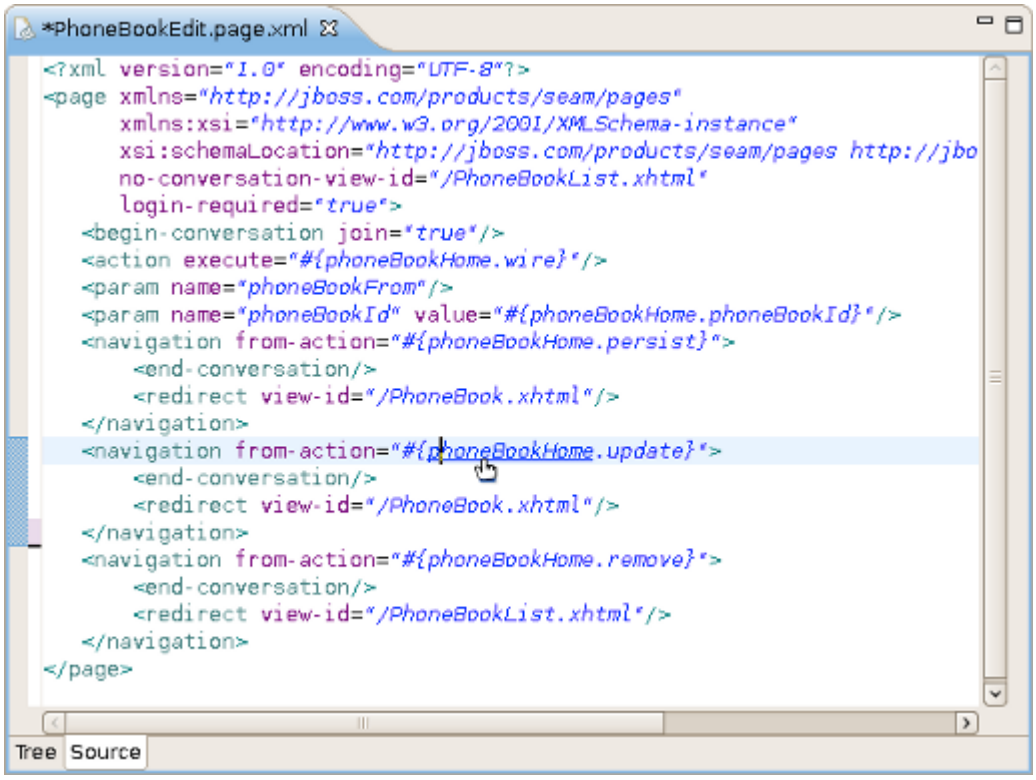


Figure 7.11. OpenOn in page.xml file

7.4.3. 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.

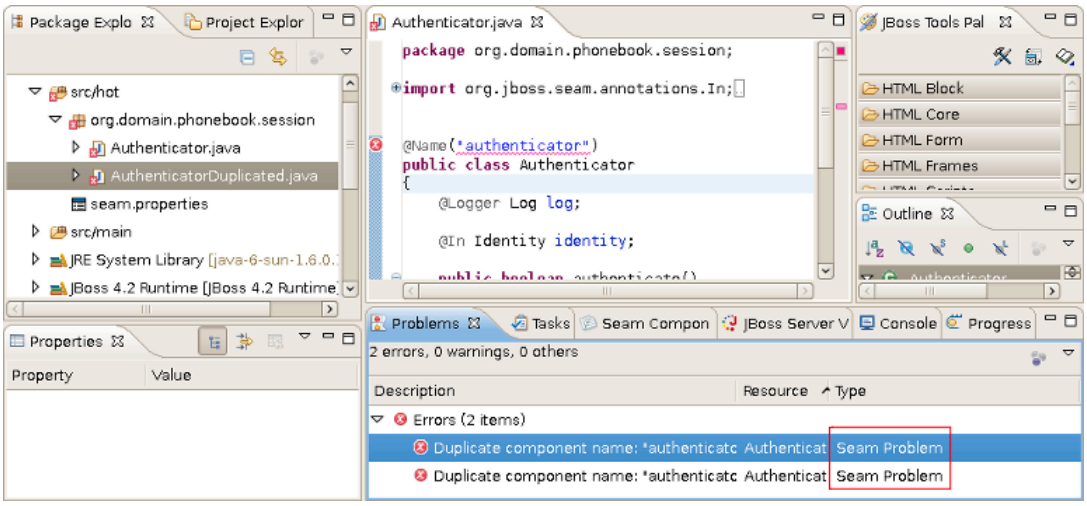


Figure 7.12. Seam Validation

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.

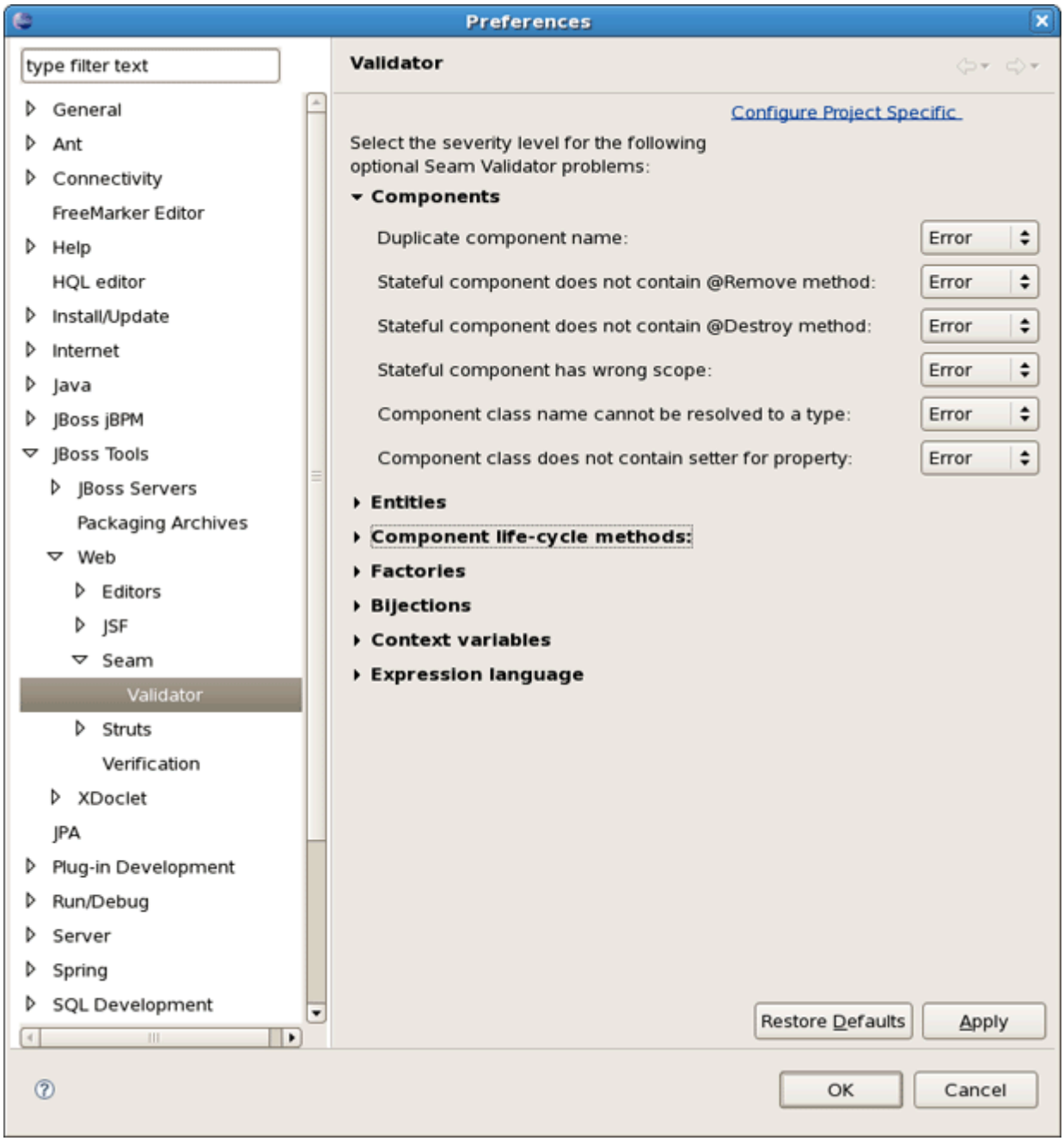


Figure 7.13. Seam Validator Preferences

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. The validations can be run manually by clicking Validate via the context menu on your project which will execute all the active WTP validations.

Seam Views

8.1. 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.

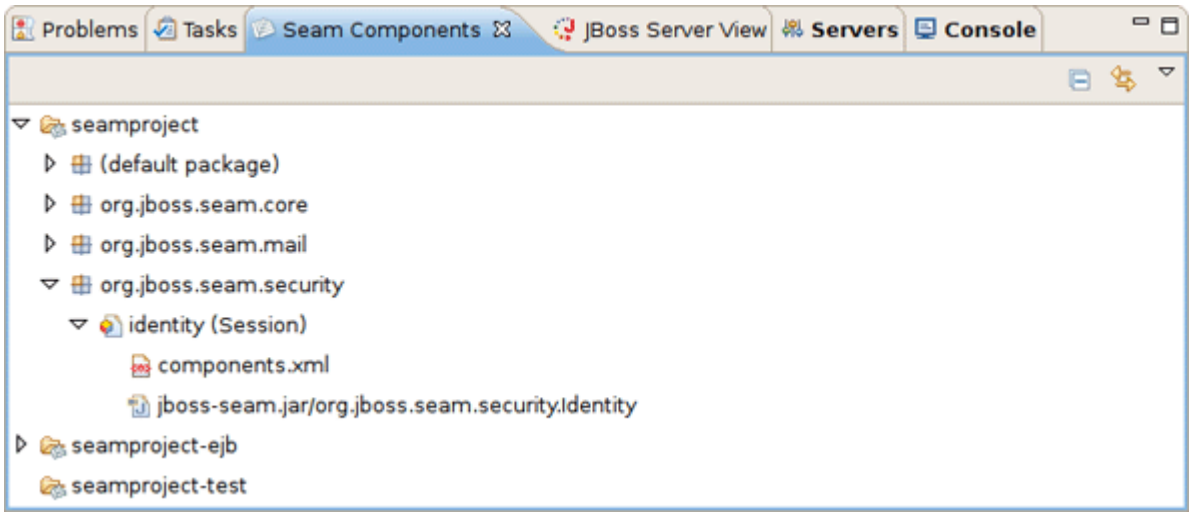


Figure 8.1. Seam Components View

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*)

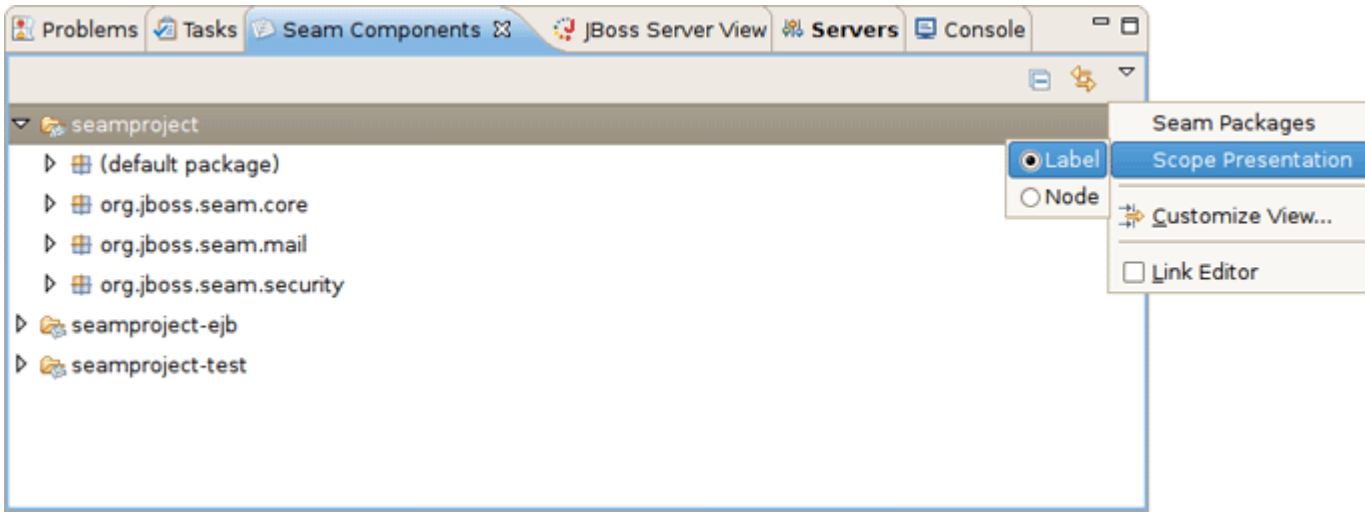


Figure 8.2. As label

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

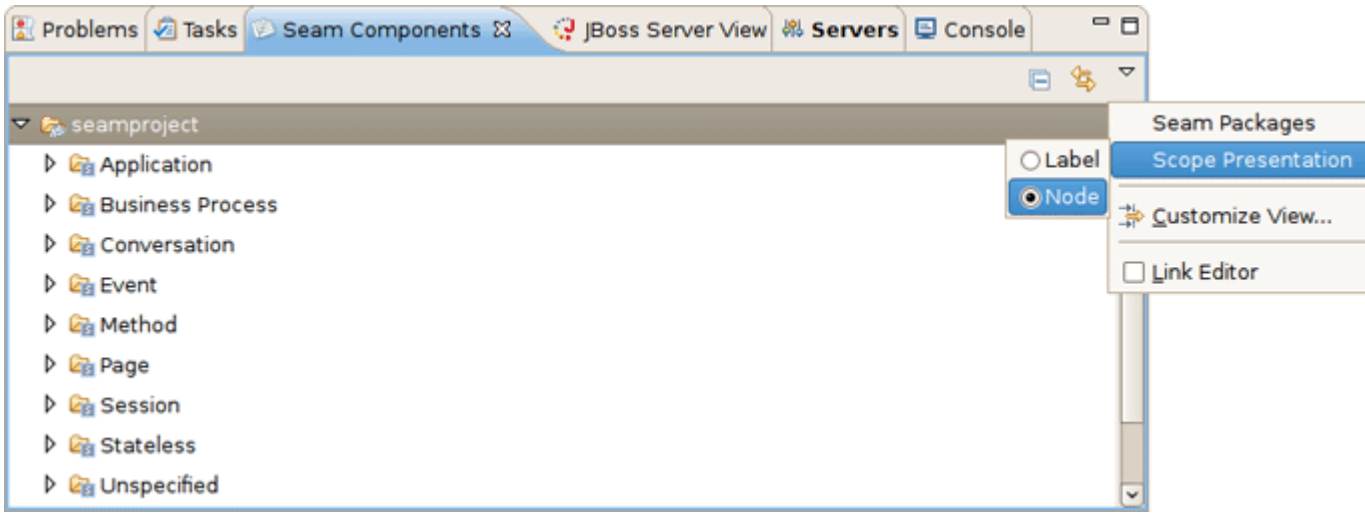


Figure 8.3. As node

The Seam Packages can be presented in two ways:

- Flat

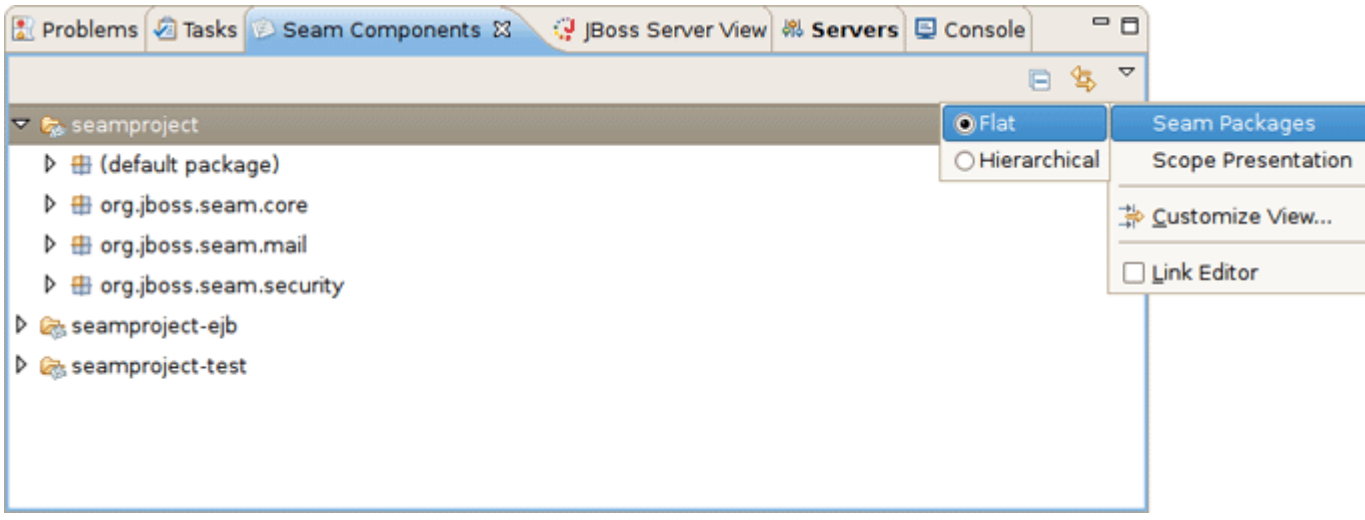


Figure 8.4. Flat Presentation of Seam Packages

- Hierarchical

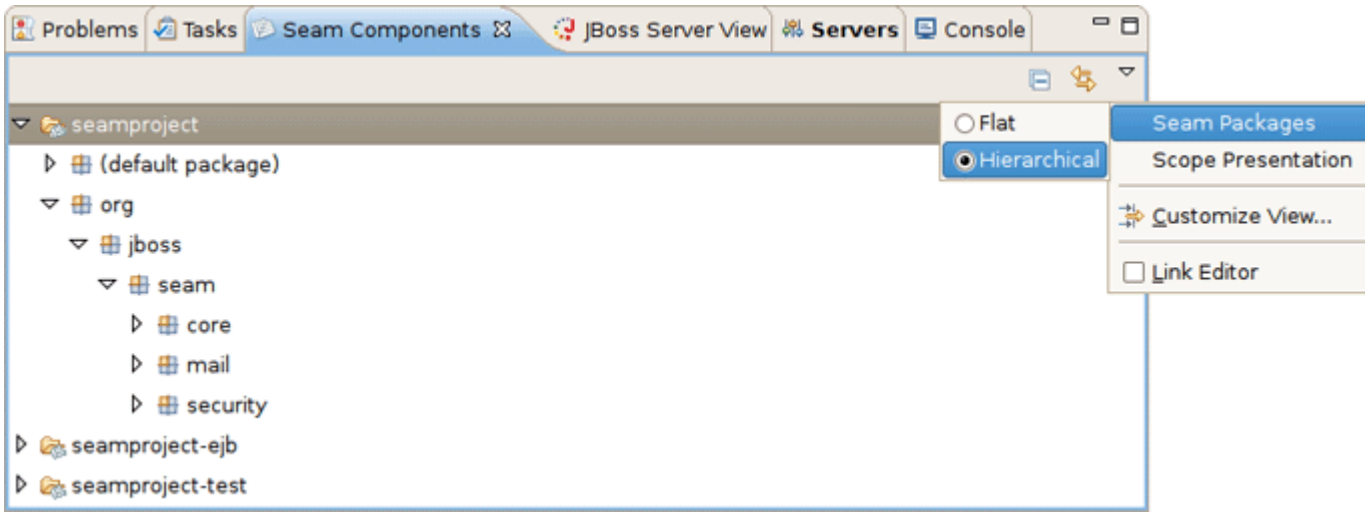


Figure 8.5. Hierarchical Presentation of Seam Packages

The Seam Component View can be filtered by choosing Customize View.

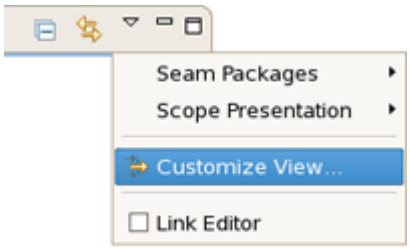


Figure 8.6. Customize View

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 dependent on other project.

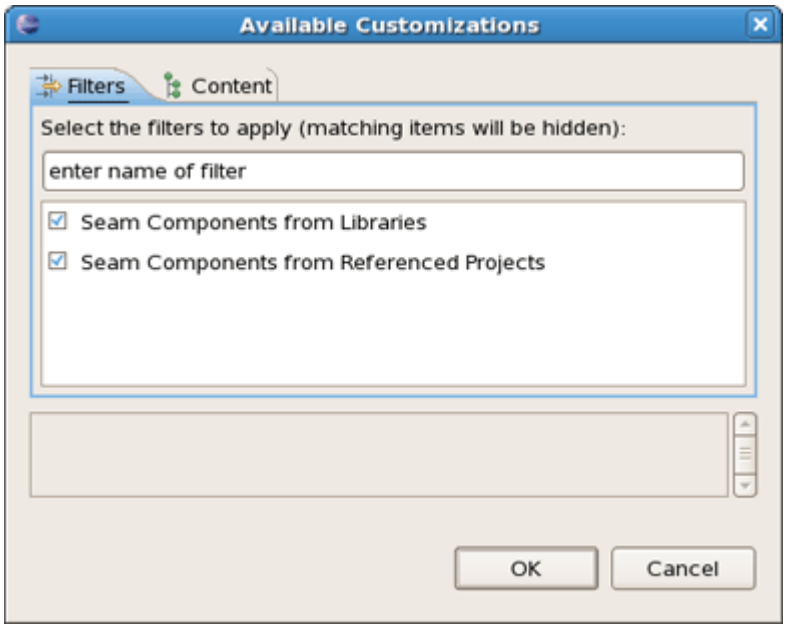


Figure 8.7. Available Custimozations

8.2. 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.

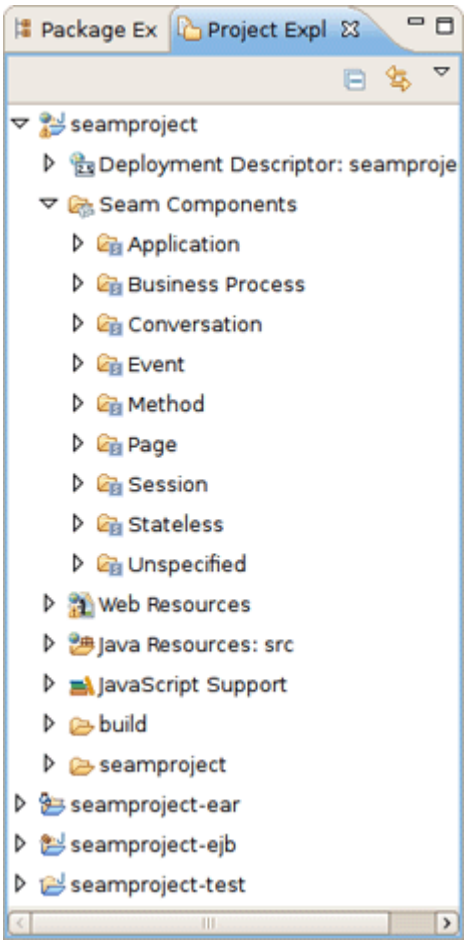


Figure 8.8. Seam Components in Project Explorer

Seam Preferences

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

9.1. 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.

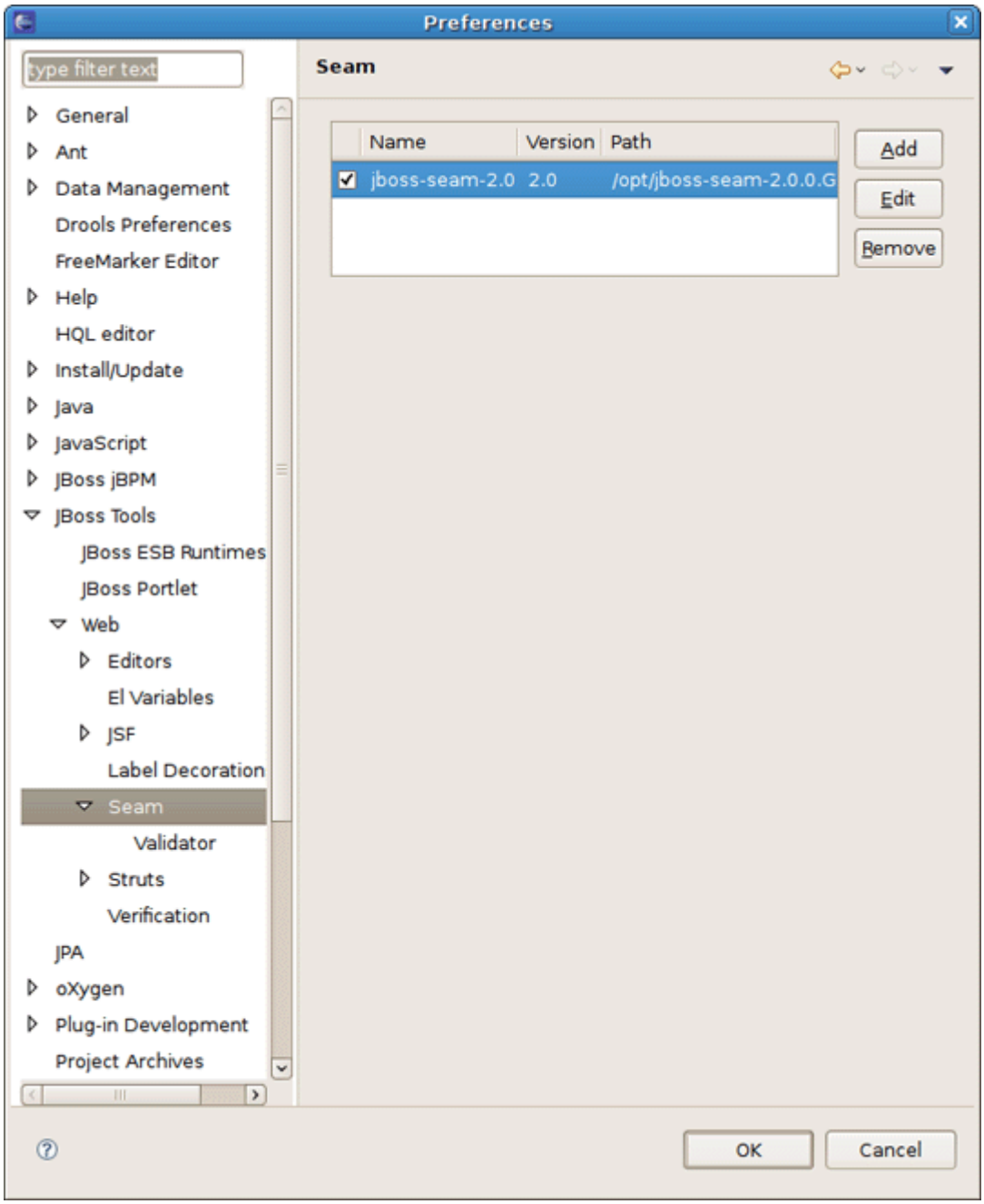


Figure 9.1. Seam Preferences Page

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.

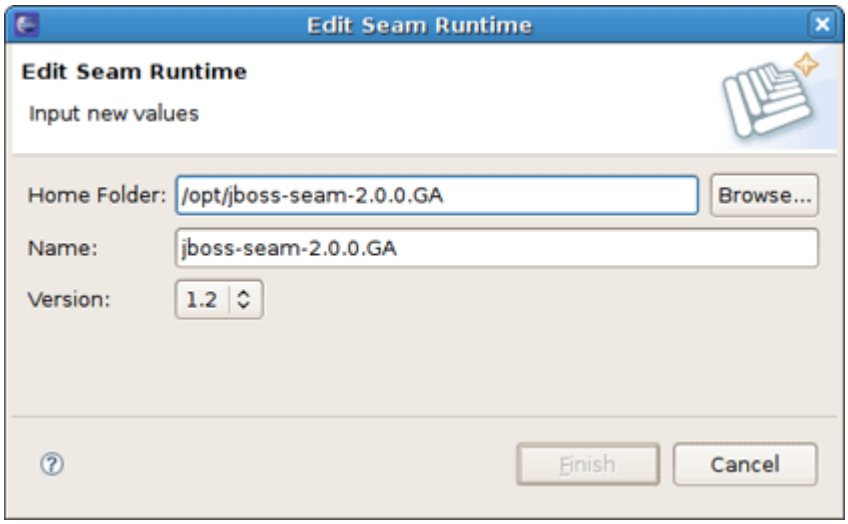


Figure 9.2. Edit Seam Runtime

9.1.1. 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.

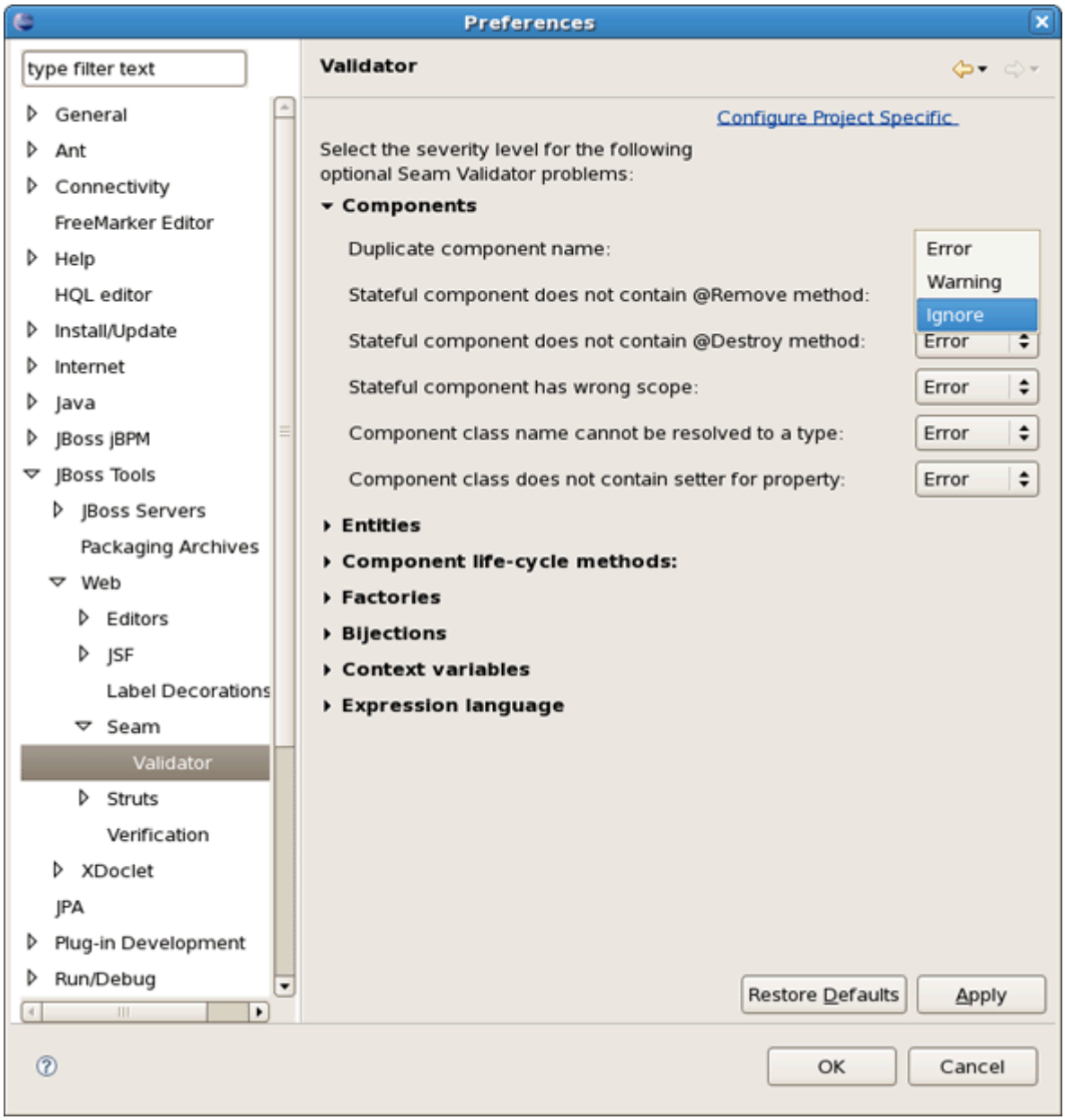


Figure 9.3. Seam Validator Preference Page

Notice, that in the *Expression language* section there is now a preference for setting severity of EL Syntax. You can select whether the Seam validator displays an error, a warning or just ignore the EL Syntax error.

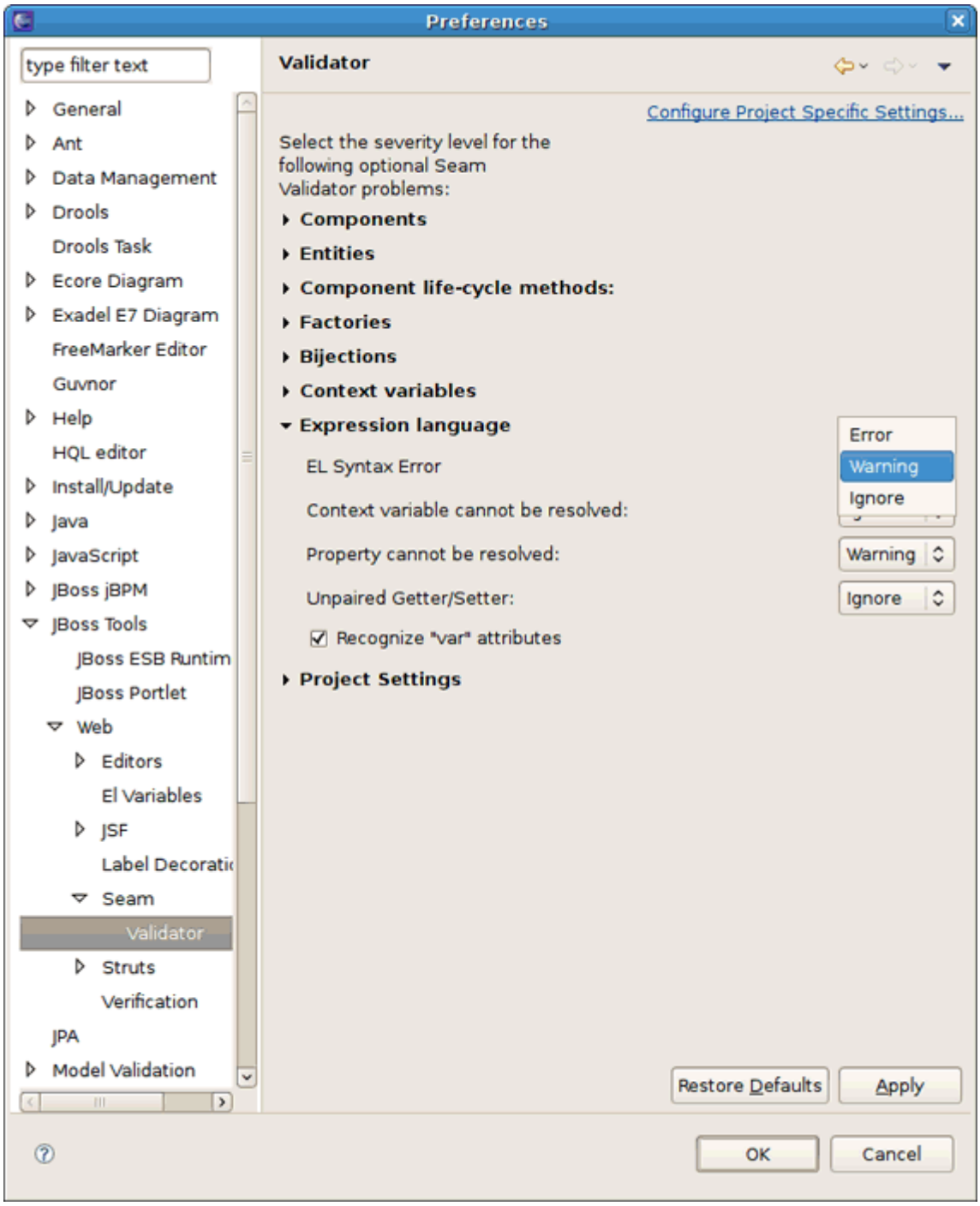


Figure 9.4. Severity Preference for EL Syntax

In the upper right corner of the 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.

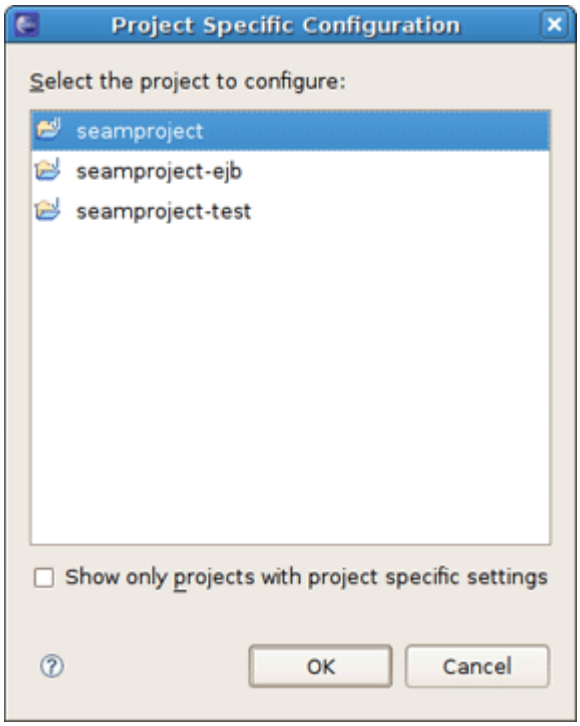


Figure 9.5. Project Specific Configuration

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*.

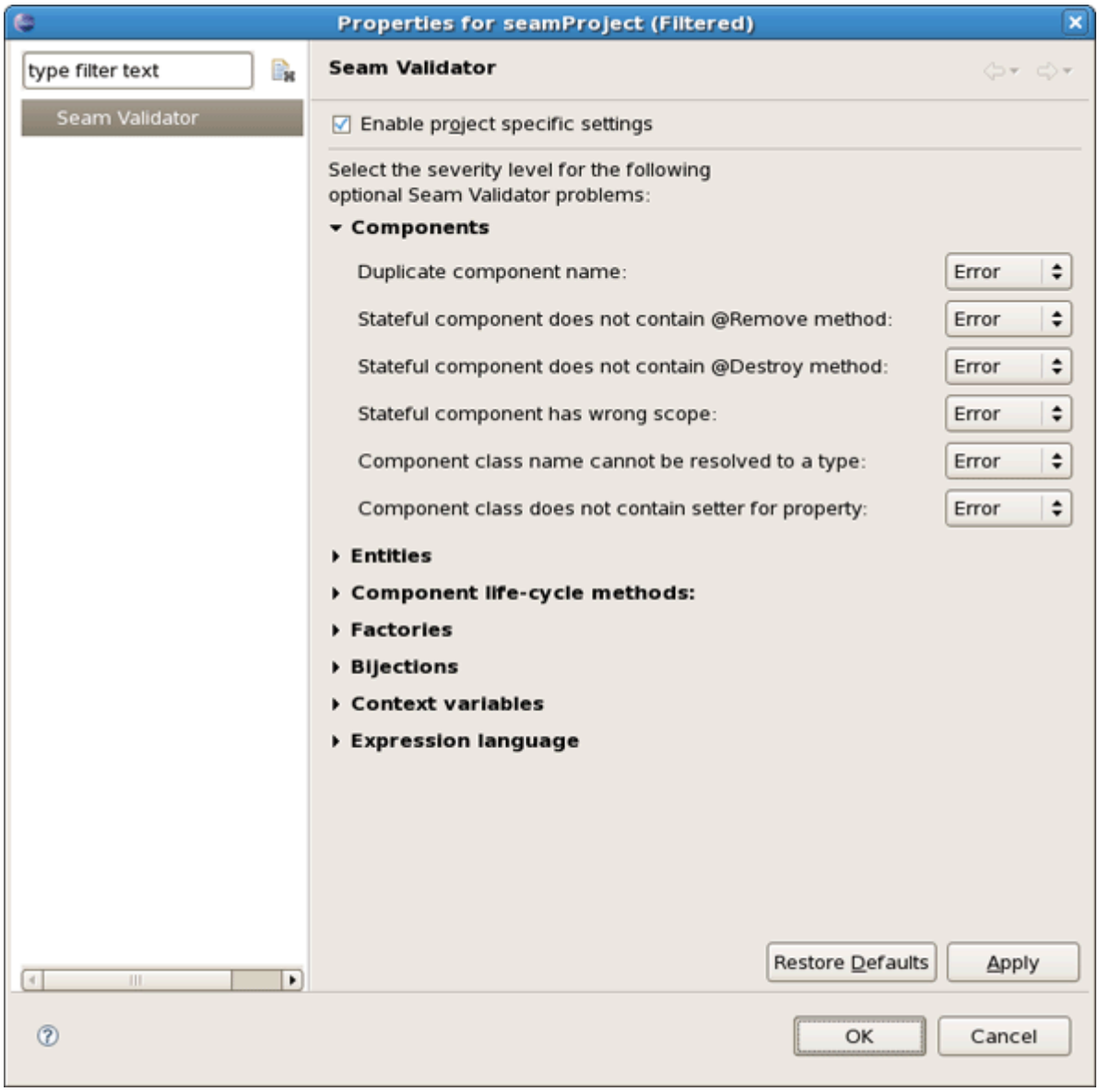


Figure 9.6. Validator Properties Page for Chosen Project

9.2. 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.

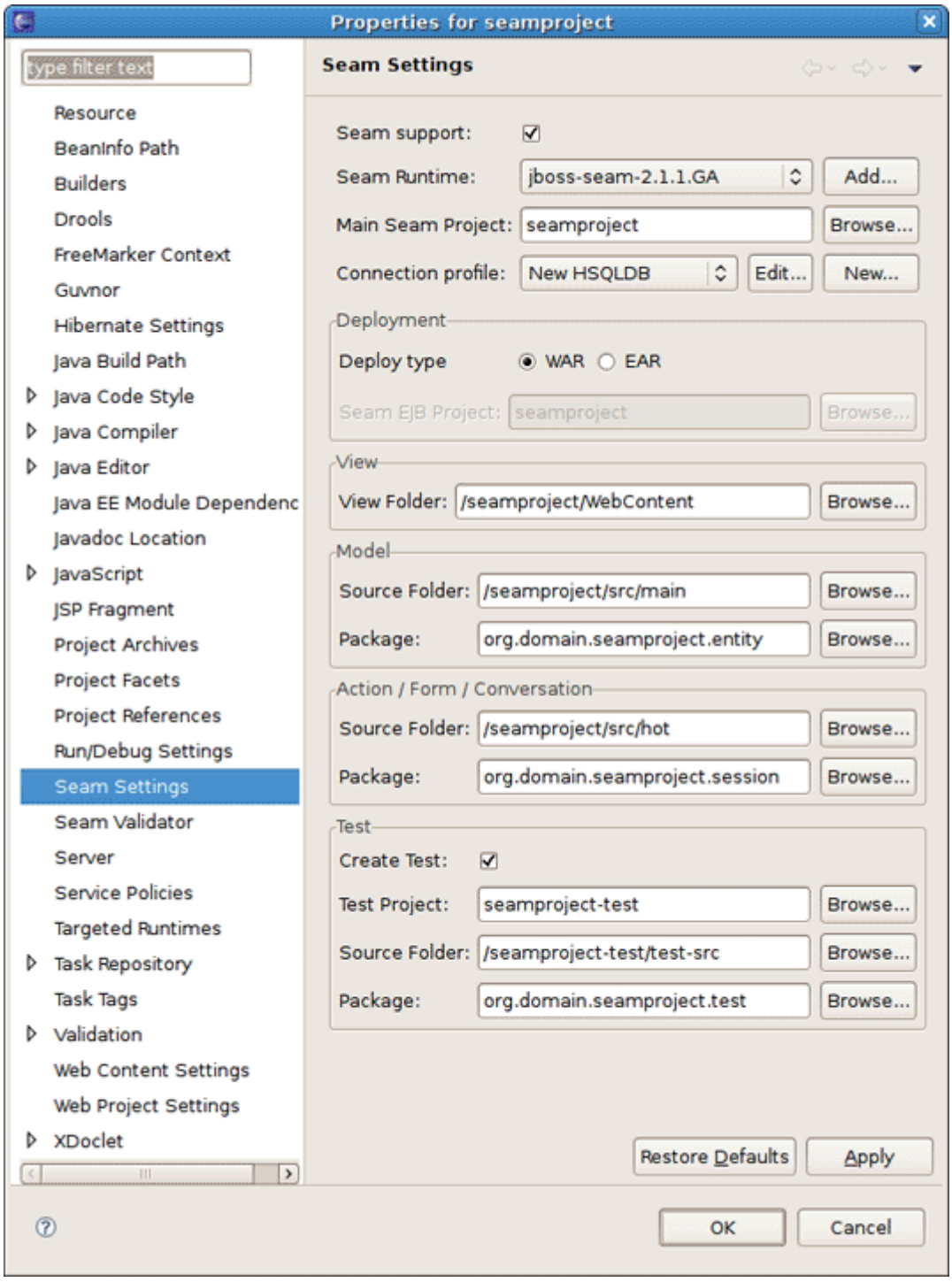


Figure 9.7. Properties for Seam Project

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 *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.

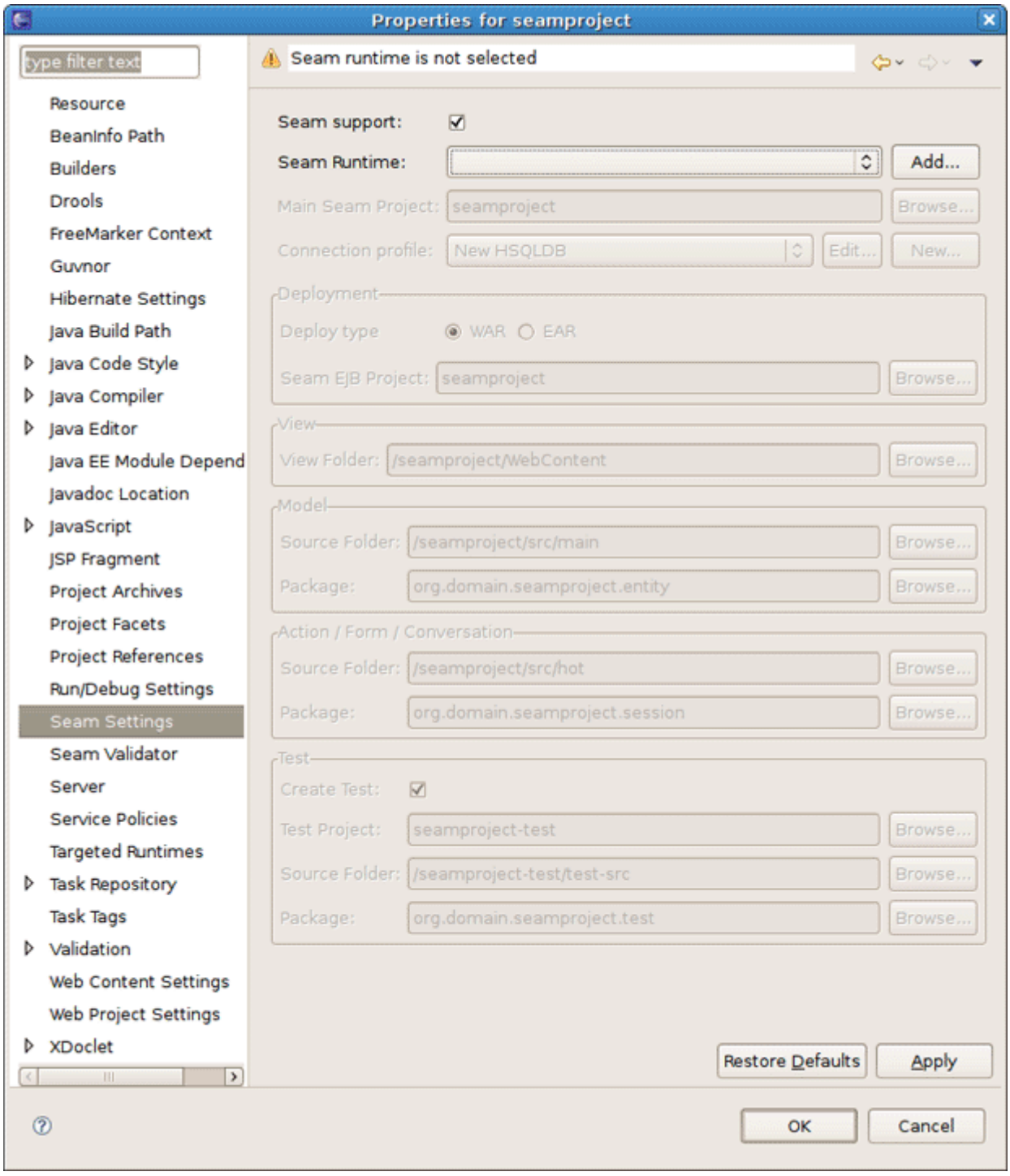


Figure 9.8. Properties for Seam Project

Tip:

On the figure above fields for configuring Seam artifacts are dimmed 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.

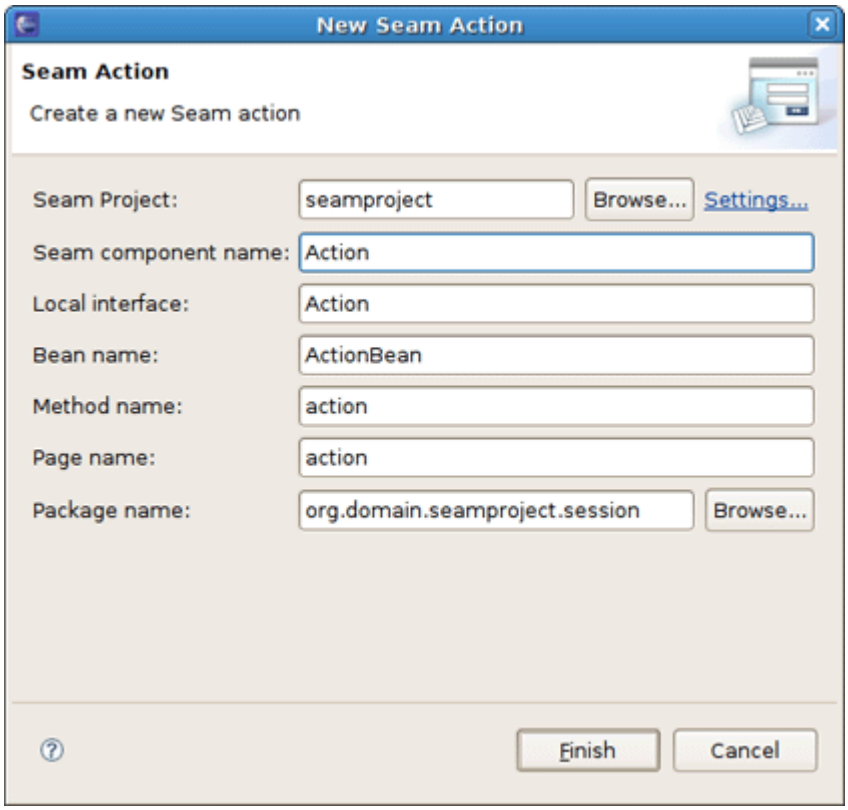


Figure 9.9. Settings Link

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.

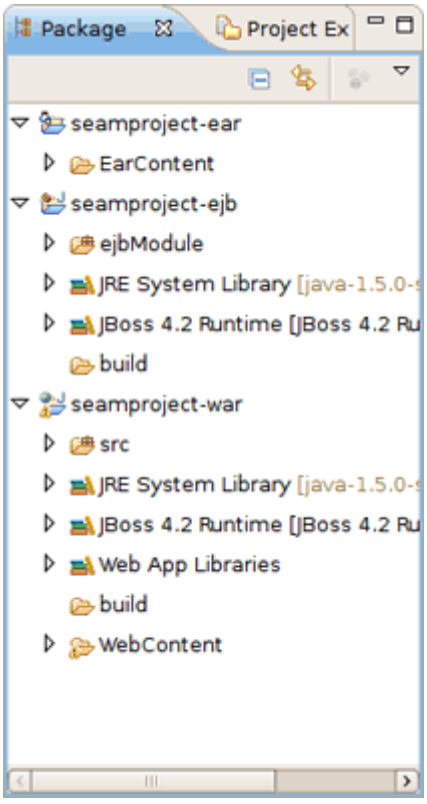


Figure 10.1. WTP Projects

Make sure EJB and WAR are included in EAR as modules (Properties for seamproject-ear -> Java EE Module Dependencies)

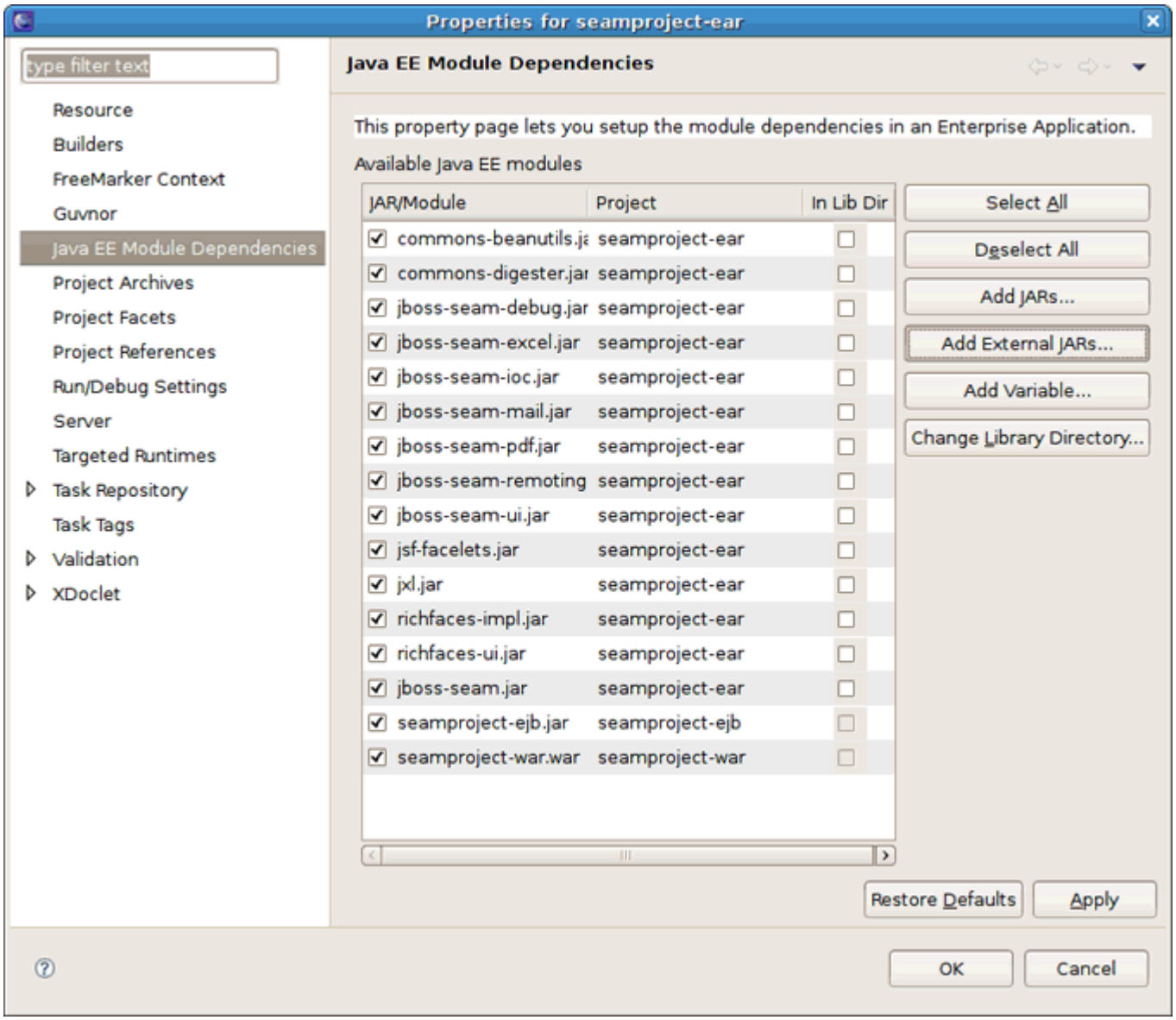


Figure 10.2. Java EE Module Dependencies

Make sure there is jboss-seam.jar in application.xml as well:

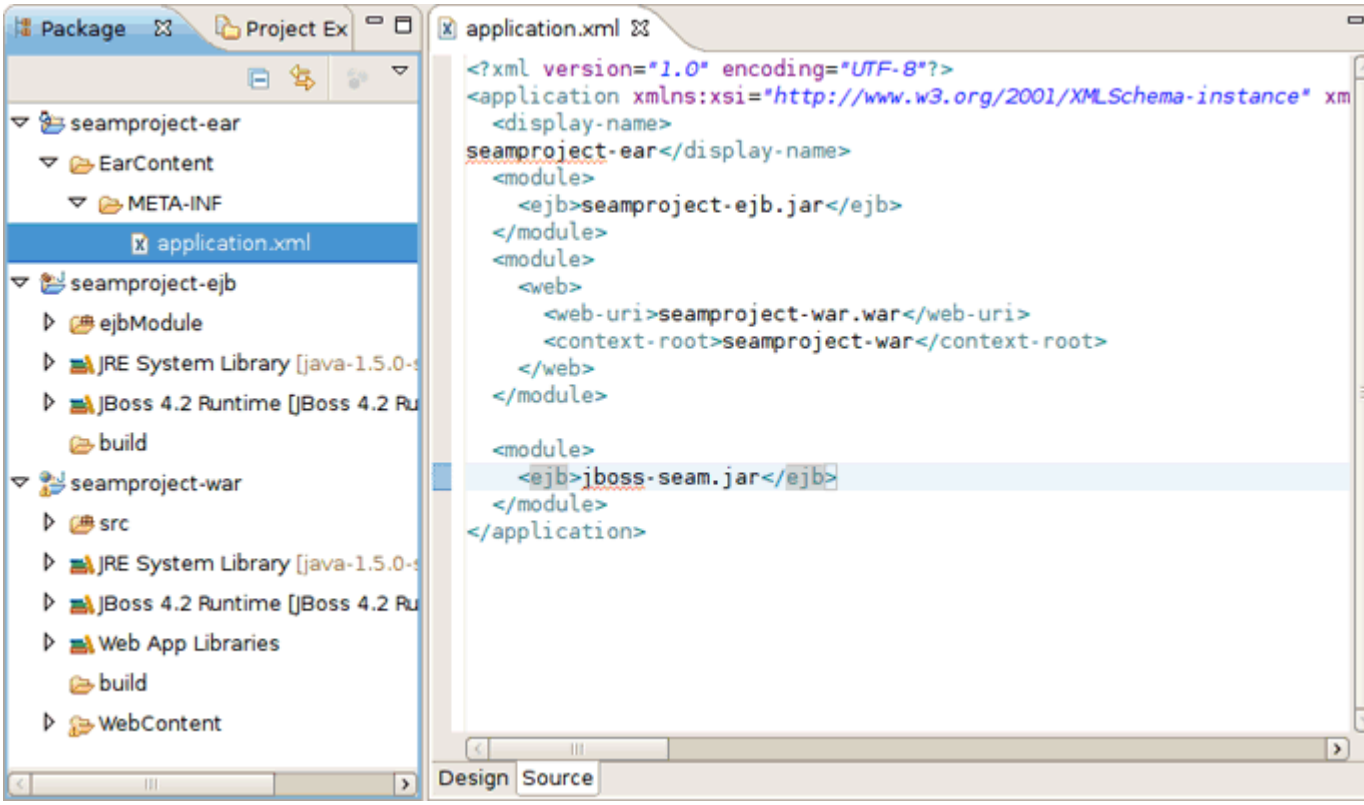


Figure 10.3. Java EE Module Dependencies

Then include libs from EAR to EJB Manifest Class-Path (Properties for seamproject-ejb -> Java EE Module Dependencies)

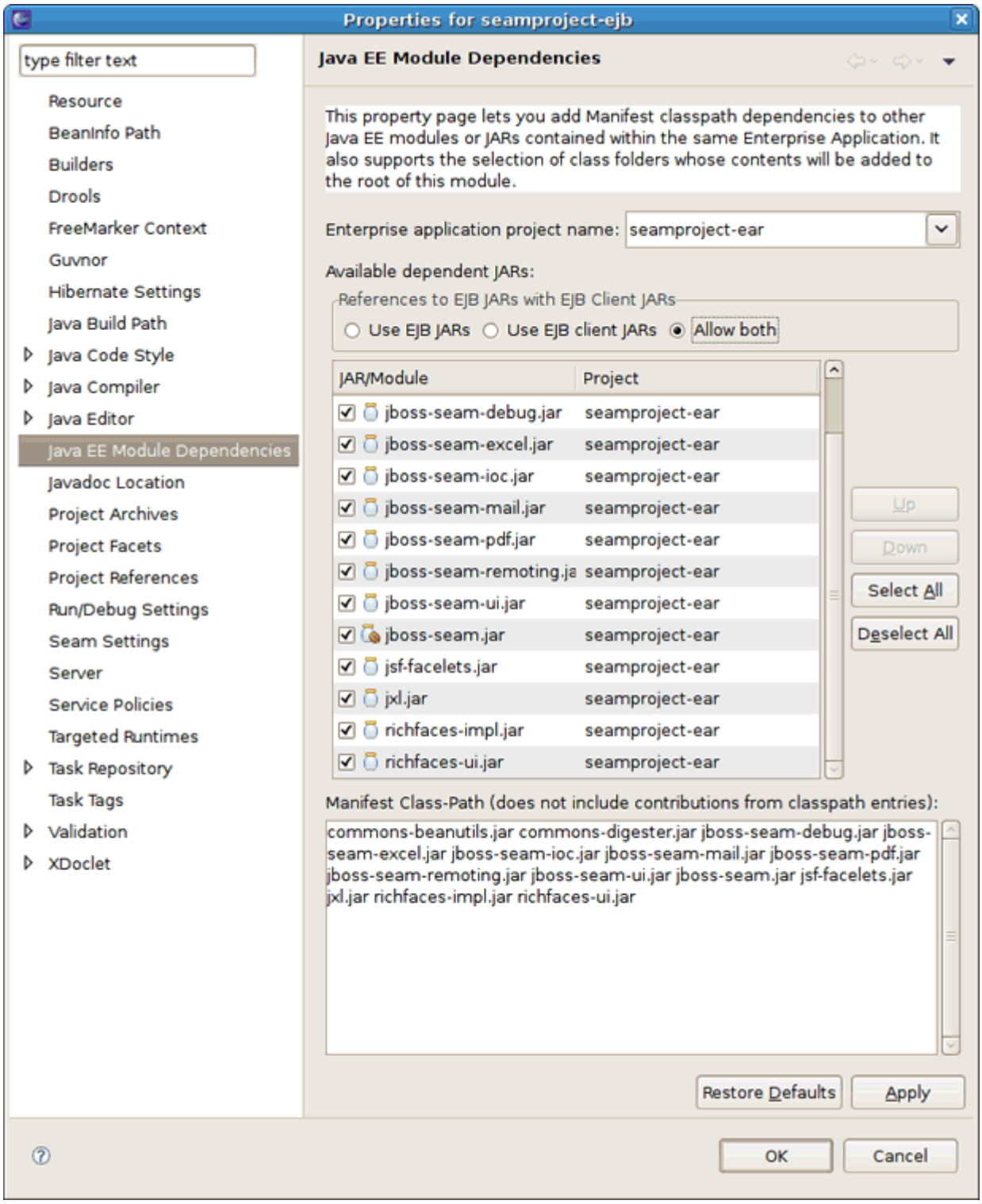


Figure 10.4. Including Libs from EAR to EJB Manifest Class-Path

Include libs from EAR and seamproject-ejb.jar to WAR Manifest Class-Path (Properties for seamproject-war -> Java EE Module Dependencies)

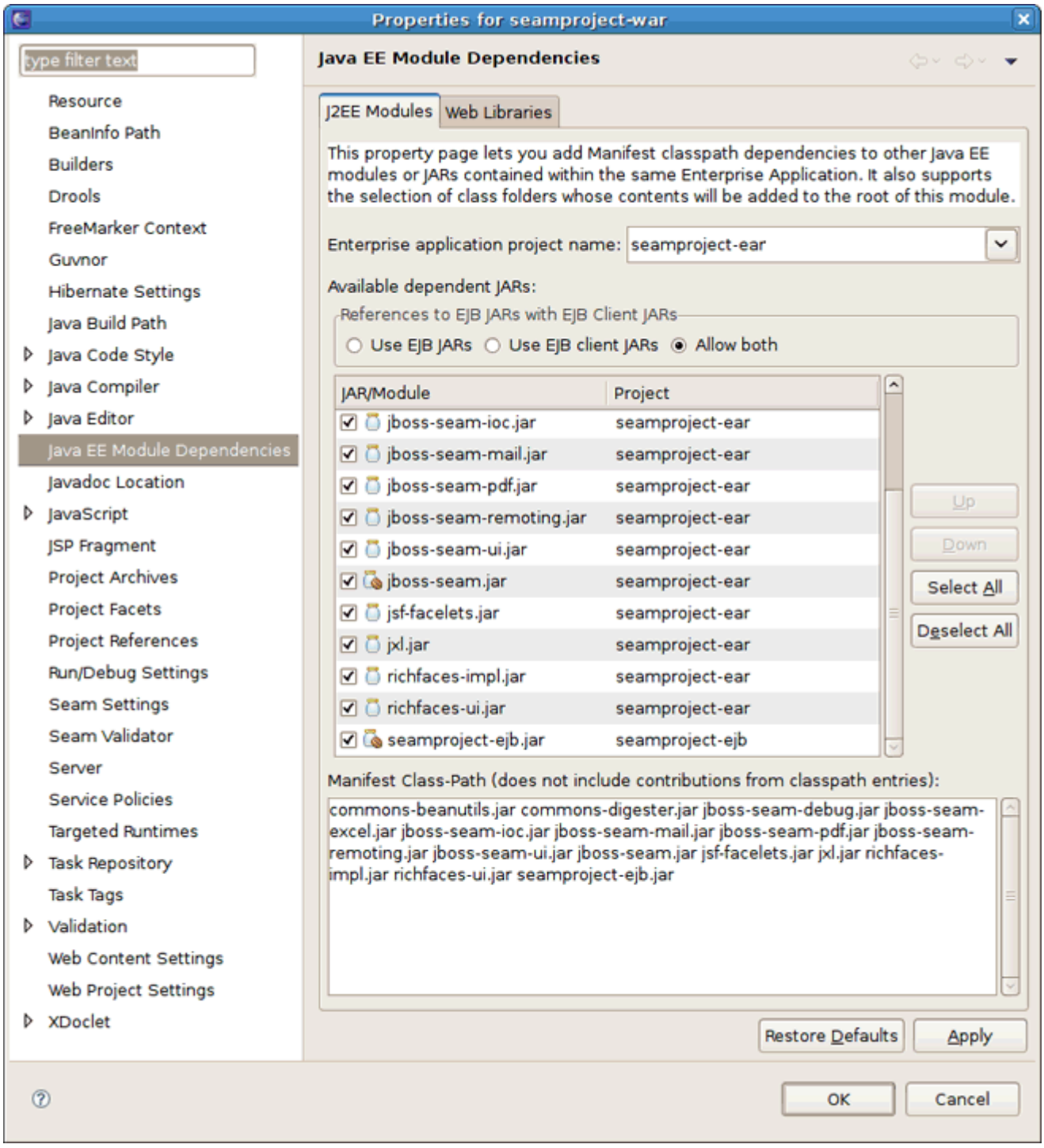


Figure 10.5. Including Libs from EAR and seamproject-ejb.jar to WAR Manifest Class-Path

Thus we have set our WTP EAR/EJB/WAR projects and now we are ready to add Seam support to them:

First you have to add Seam support to WAR project: Properties for seamproject-war -> Seam Settings

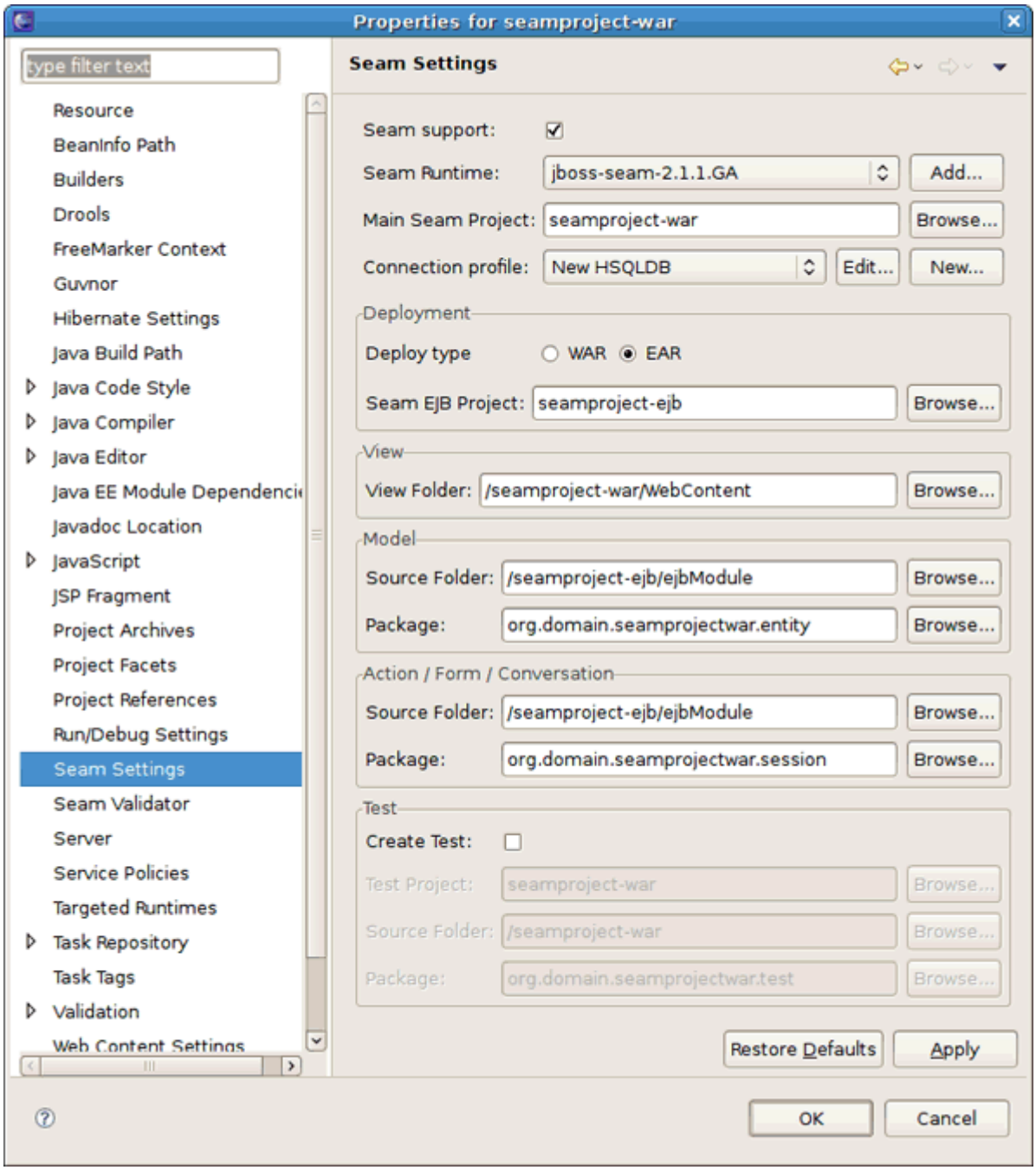


Figure 10.6. Adding Seam Support to WAR Project

Set seamproject-war as Main Seam project. All other settings mostly are used by New Seam Entity/Action/Form/Conversation/... Wizards. Thus you can set them as you wish.

Then you can add Seam support to EJB project: Properties for seamproject-war -> Seam Settings

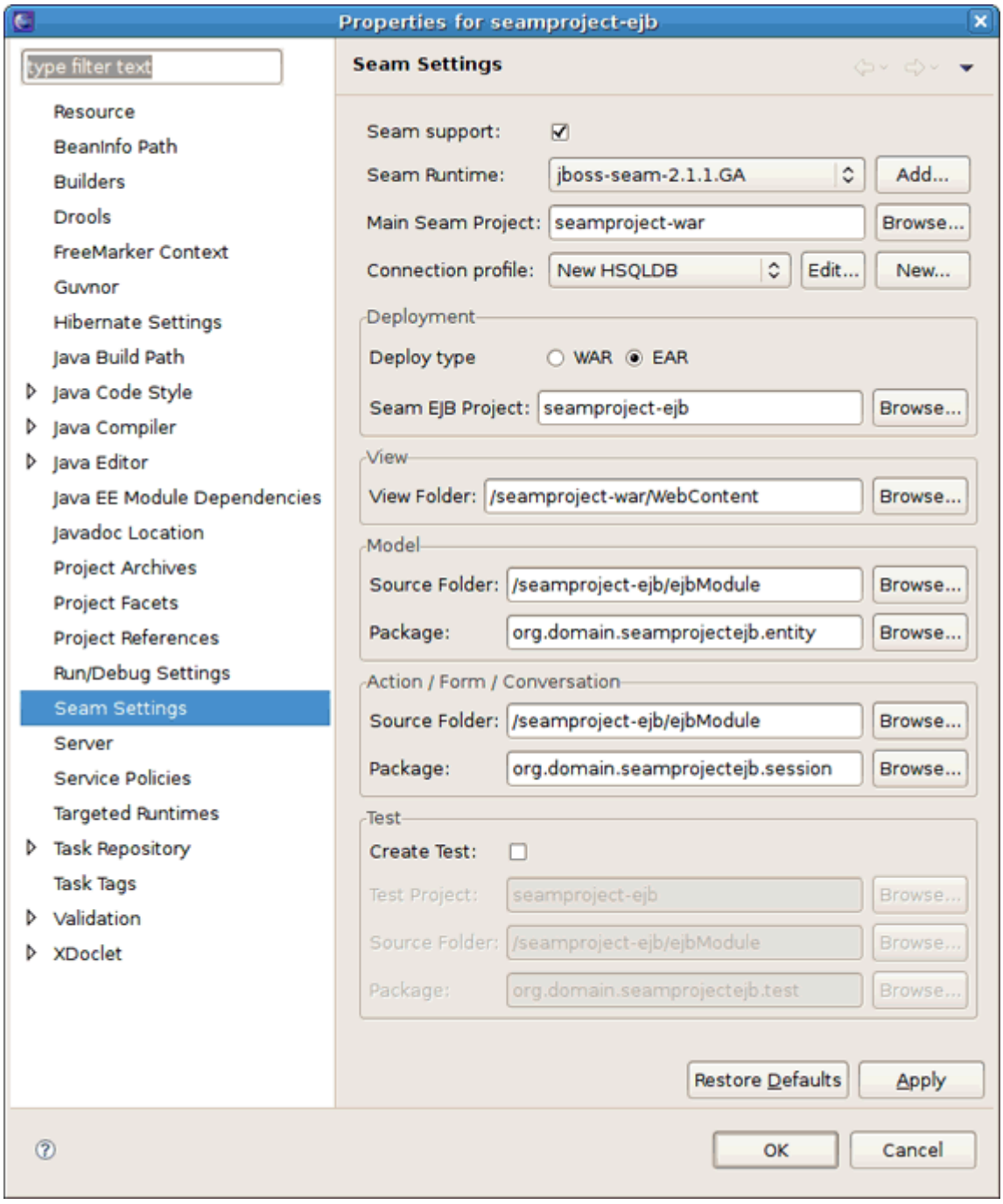


Figure 10.7. Adding Seam Support to EJB Project

It&s important to set seamproject-war project as main Seam project there.

It will allow Seam Tools to use one common Seam model for seamproject-ejb and seamproject-war projects.

The last step is to clean/build the projects

That is all. You have added Seam support to your EAR project.

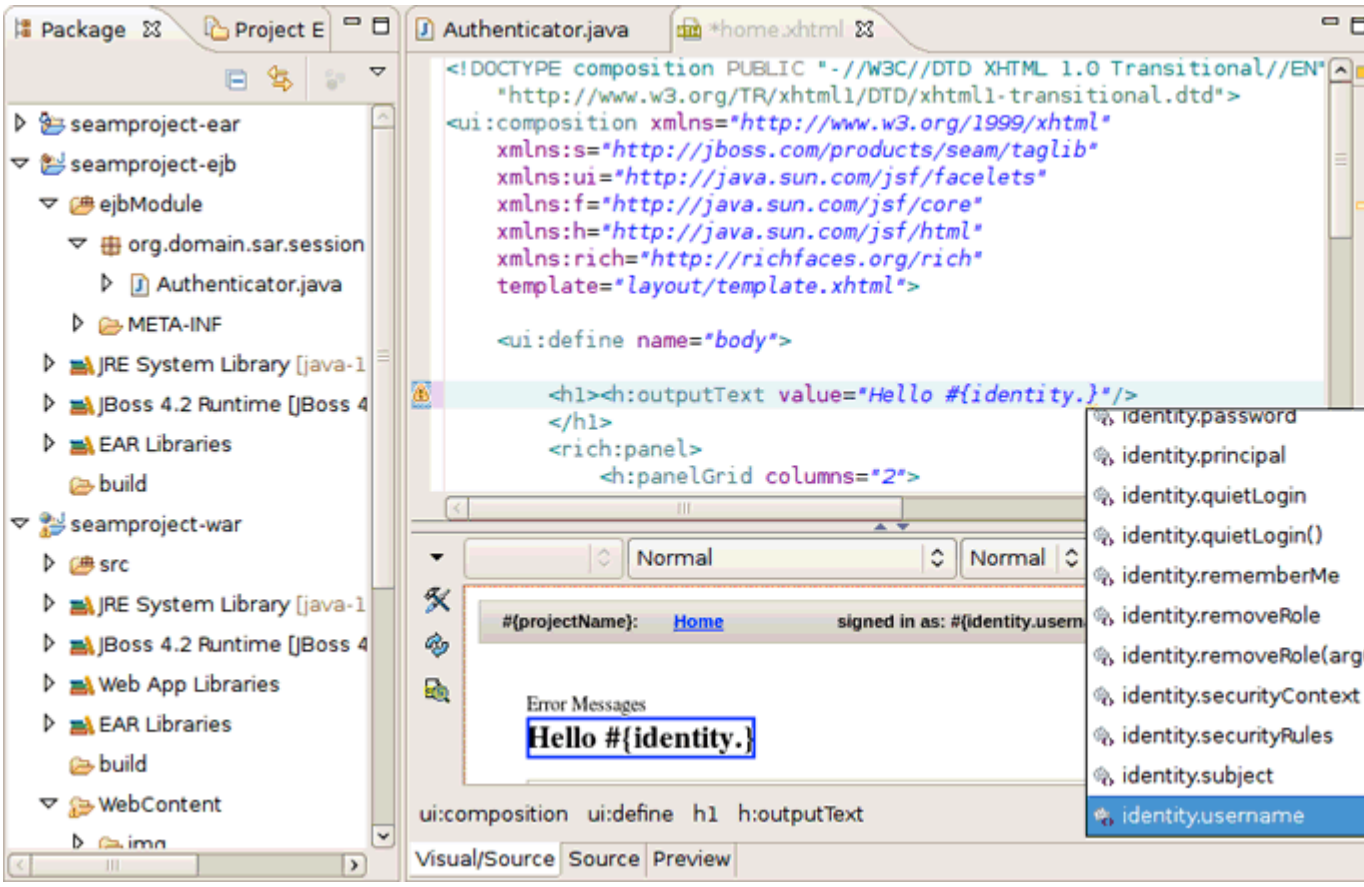


Figure 10.8. Added Seam Support to EAR Project

Generate a CRUD Database Application

11.1. 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 *Seam* Web application, connect it to the *HSQL* [http://www.hsqldb.org/] database and add CRUD support.

11.2. How to create the CRUD Database Application with Seam

- First, you should *download* [http://docs.jboss.org/tools/resources/] a database 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*.

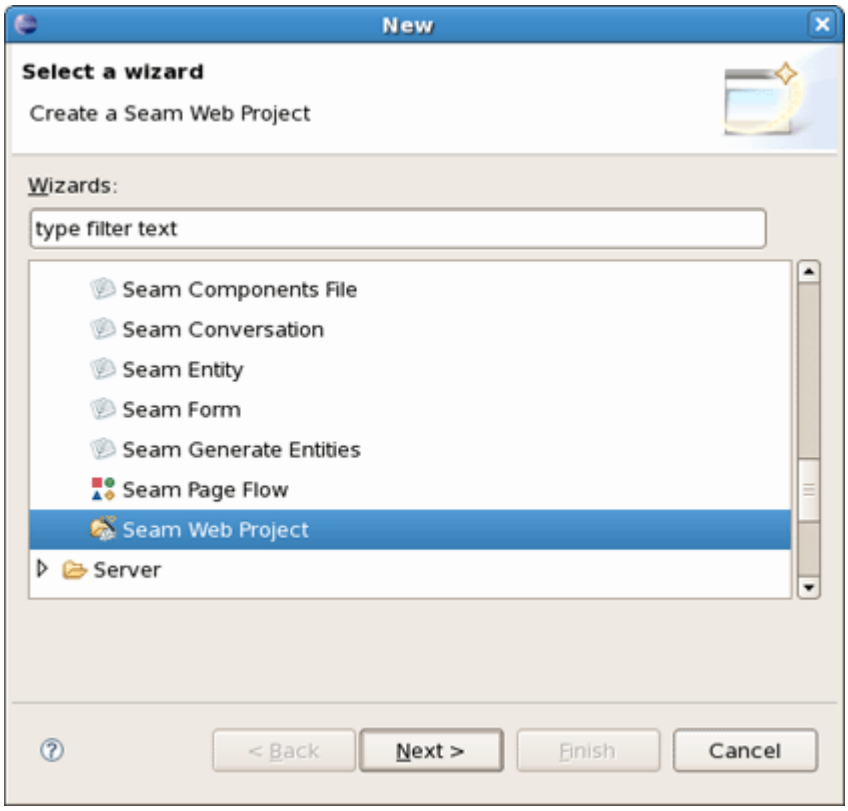


Figure 11.1. Seam Web Project Creation

- 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.

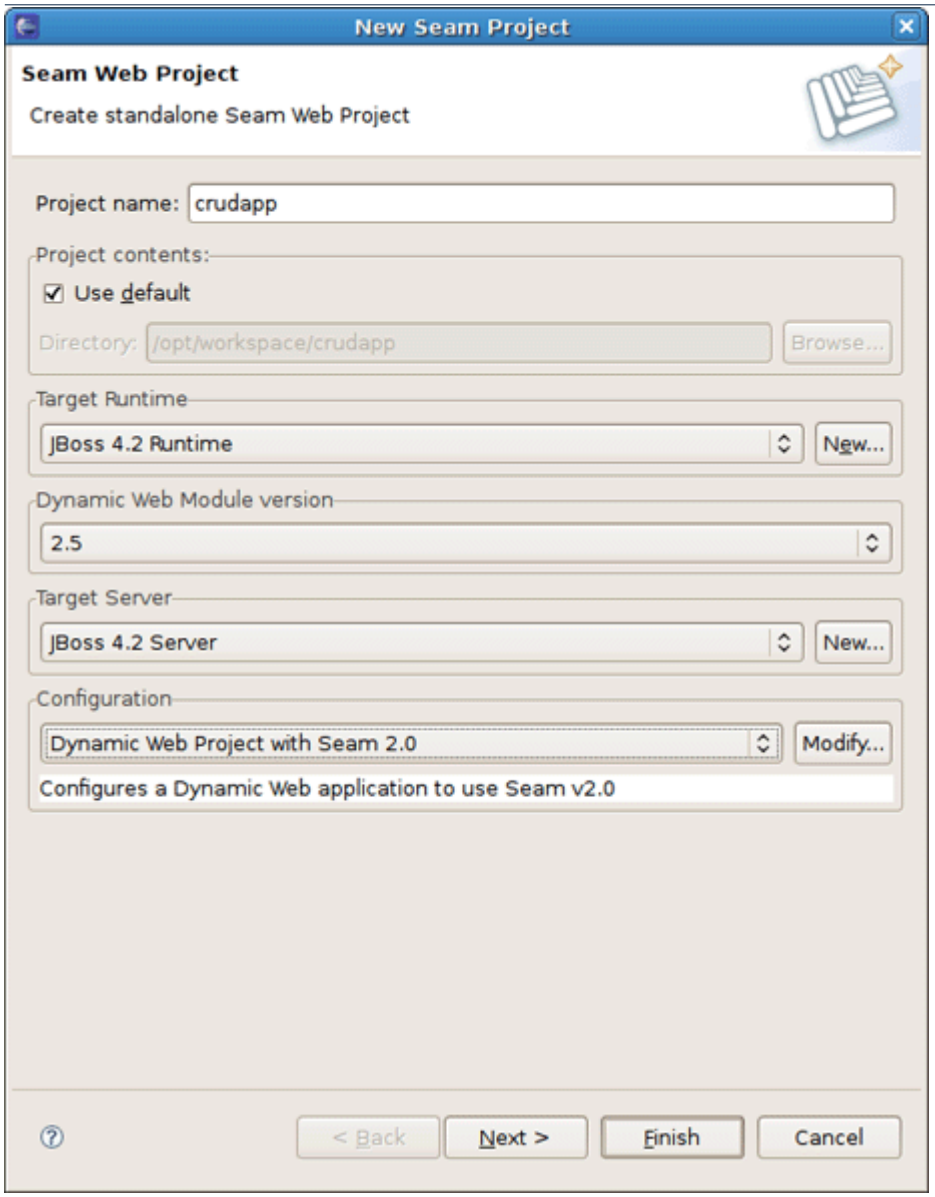


Figure 11.2. New Seam Project Wizard

Tip:

Please have a look *here* 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.

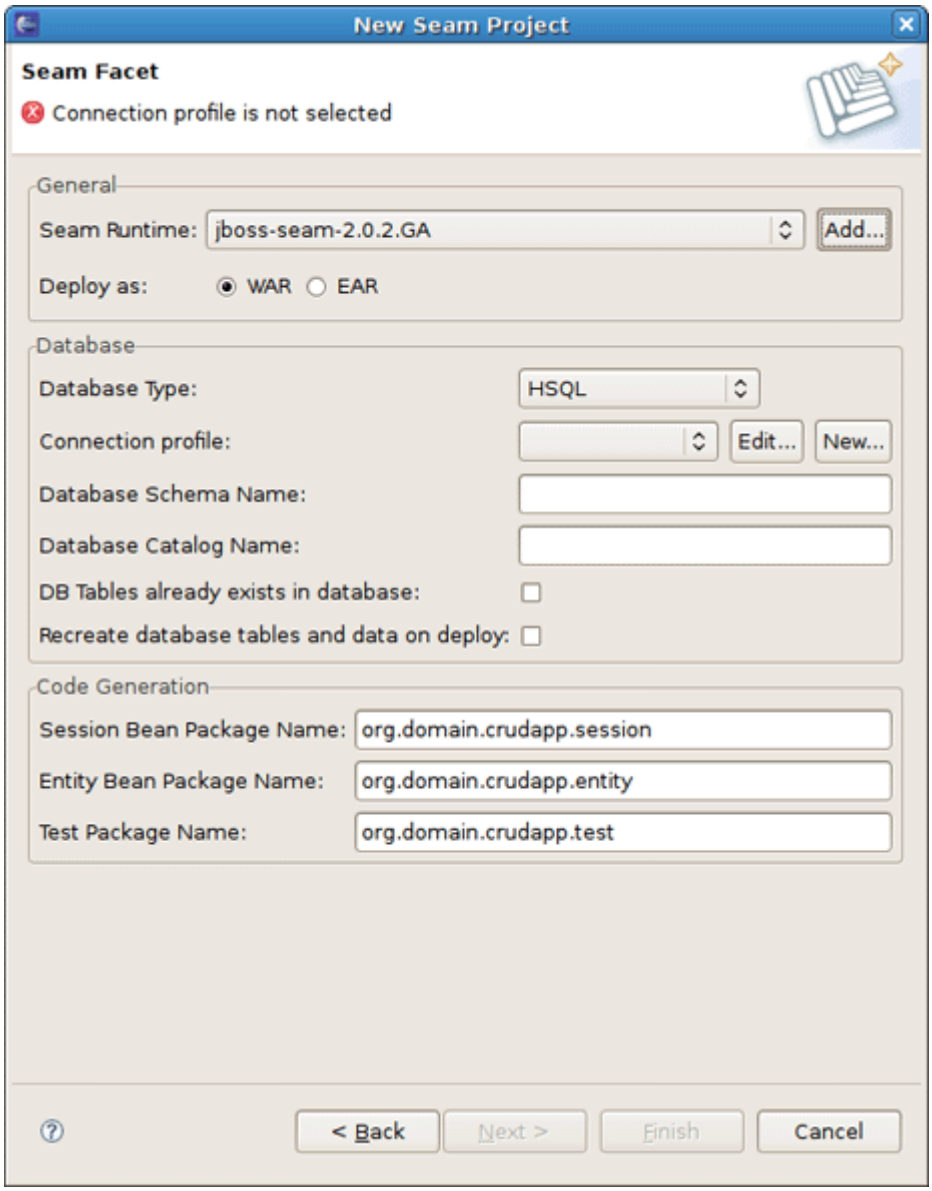


Figure 11.3. Seam Facet page

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

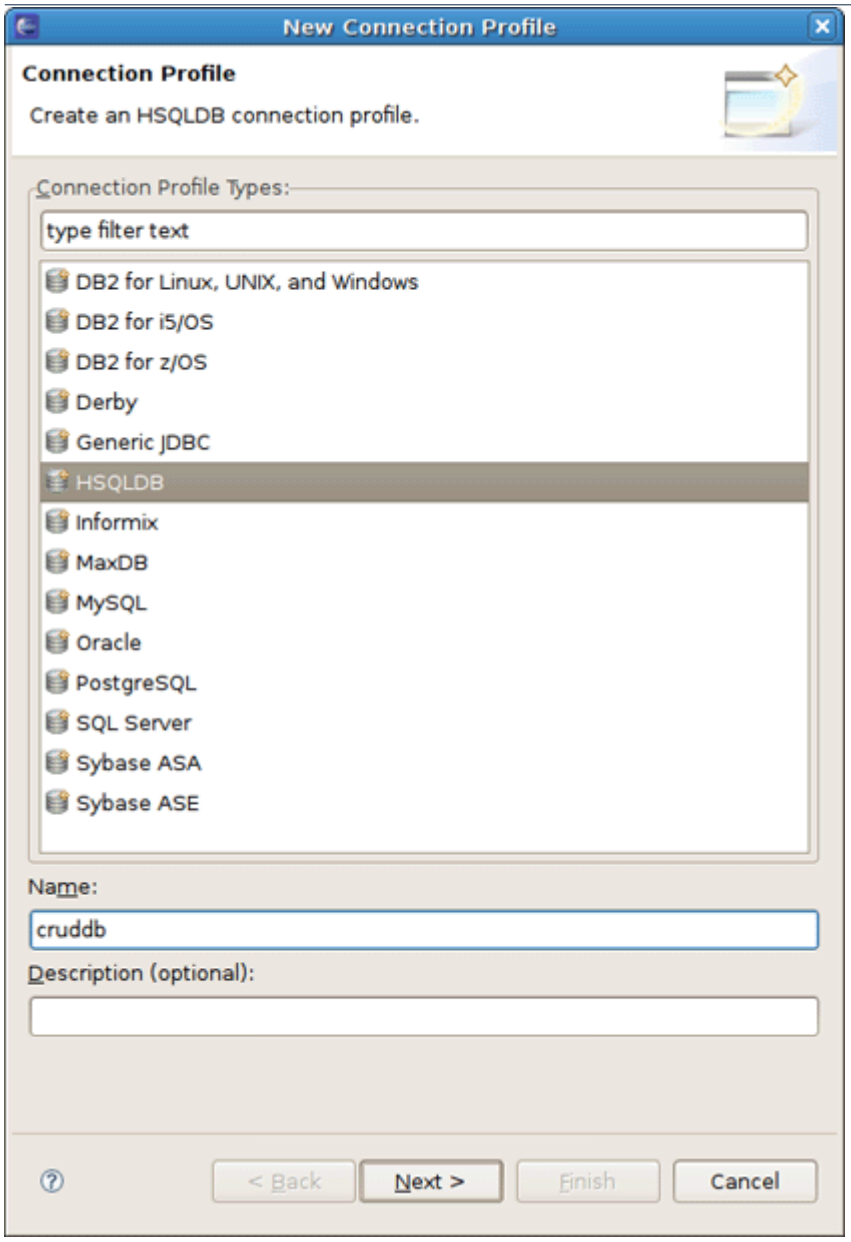


Figure 11.4. New JDBC Connection Profile

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

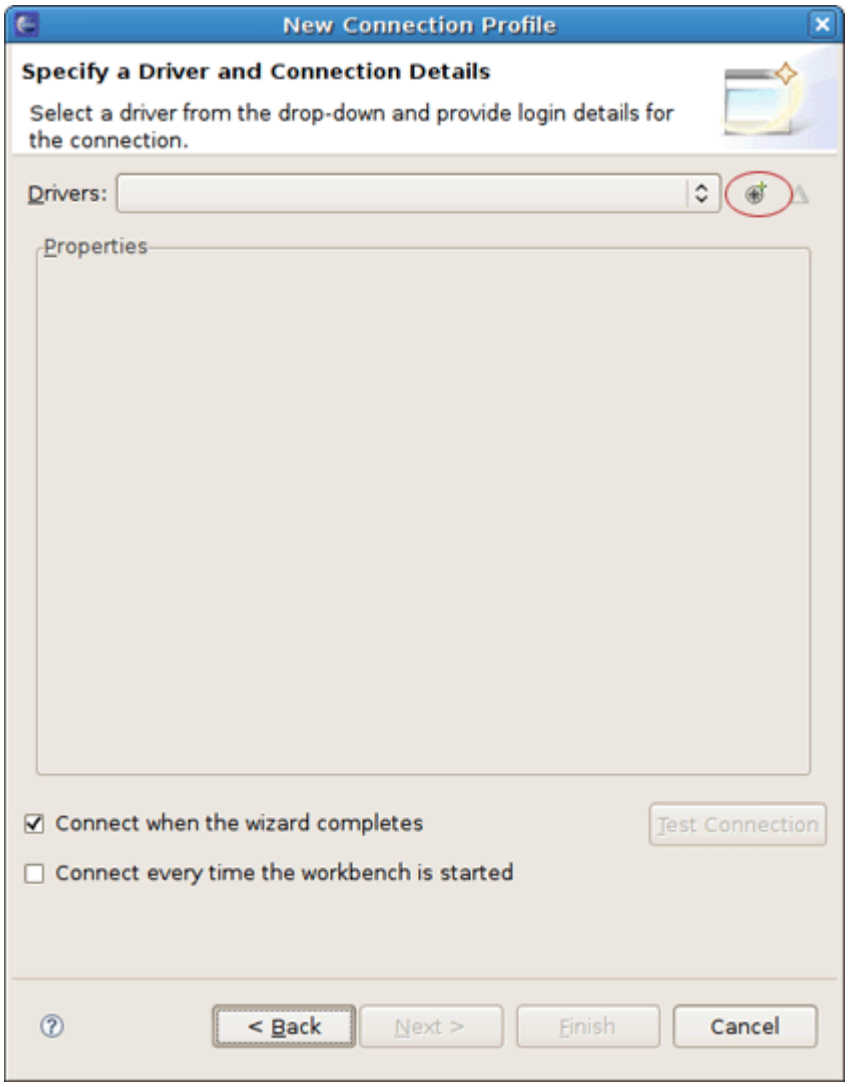


Figure 11.5. New JDBC Connection Profile Database Driver

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

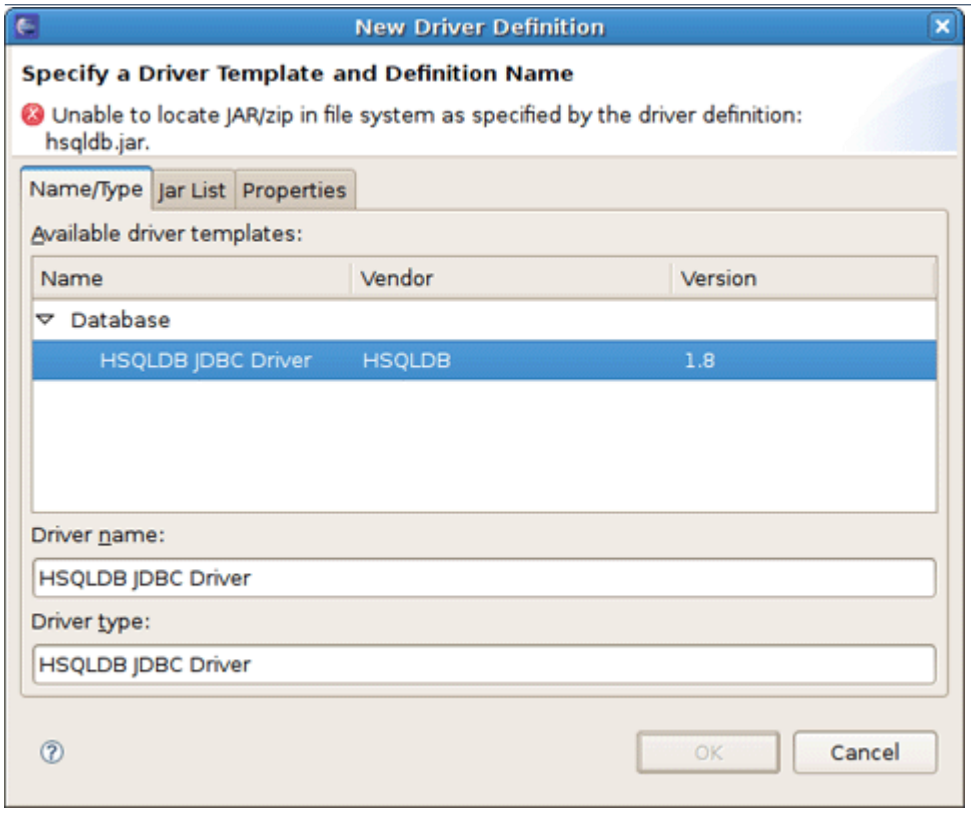


Figure 11.6. Database Driver Type

- 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 switching to the next tab and press *Add JAR/Zip* button.

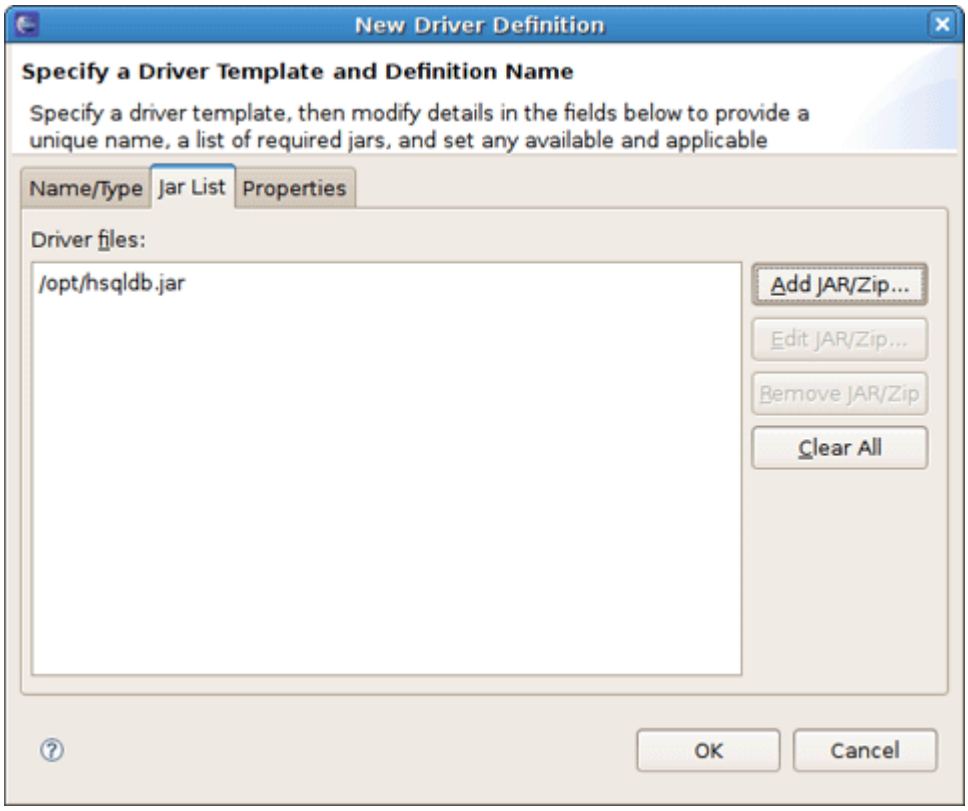


Figure 11.7. Driver Definition

- On the Properties tab set the *Connection URL*, *Database Name* and *User ID* and click *OK*.

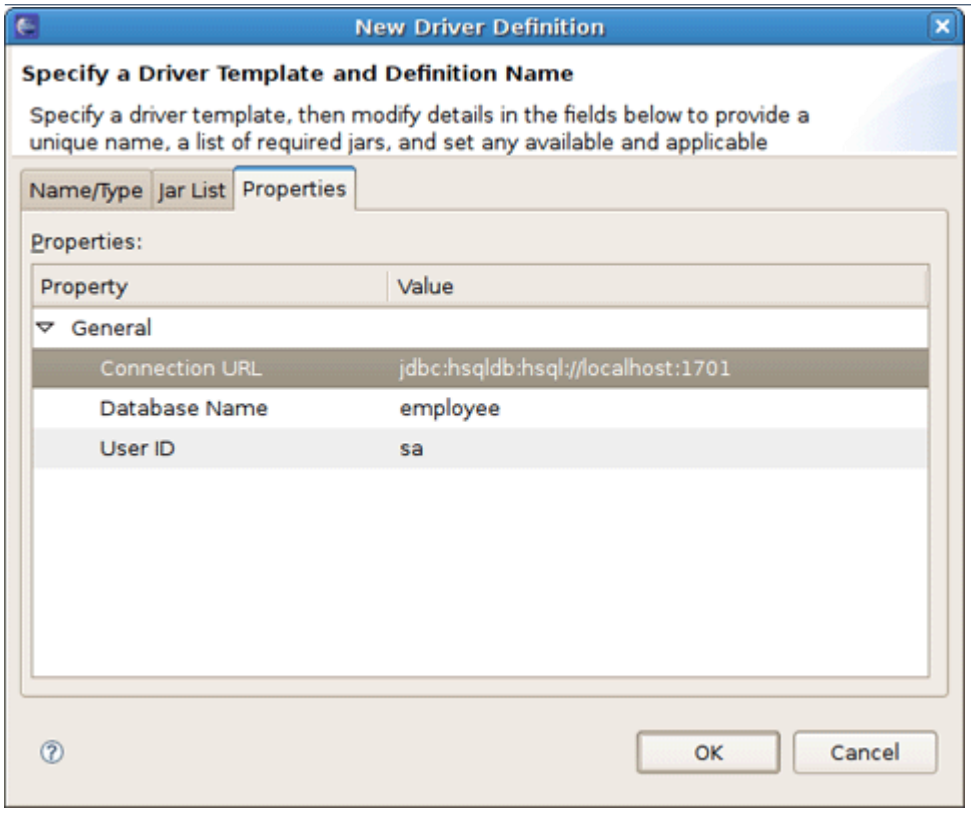


Figure 11.8. Driver Definitions Properties

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

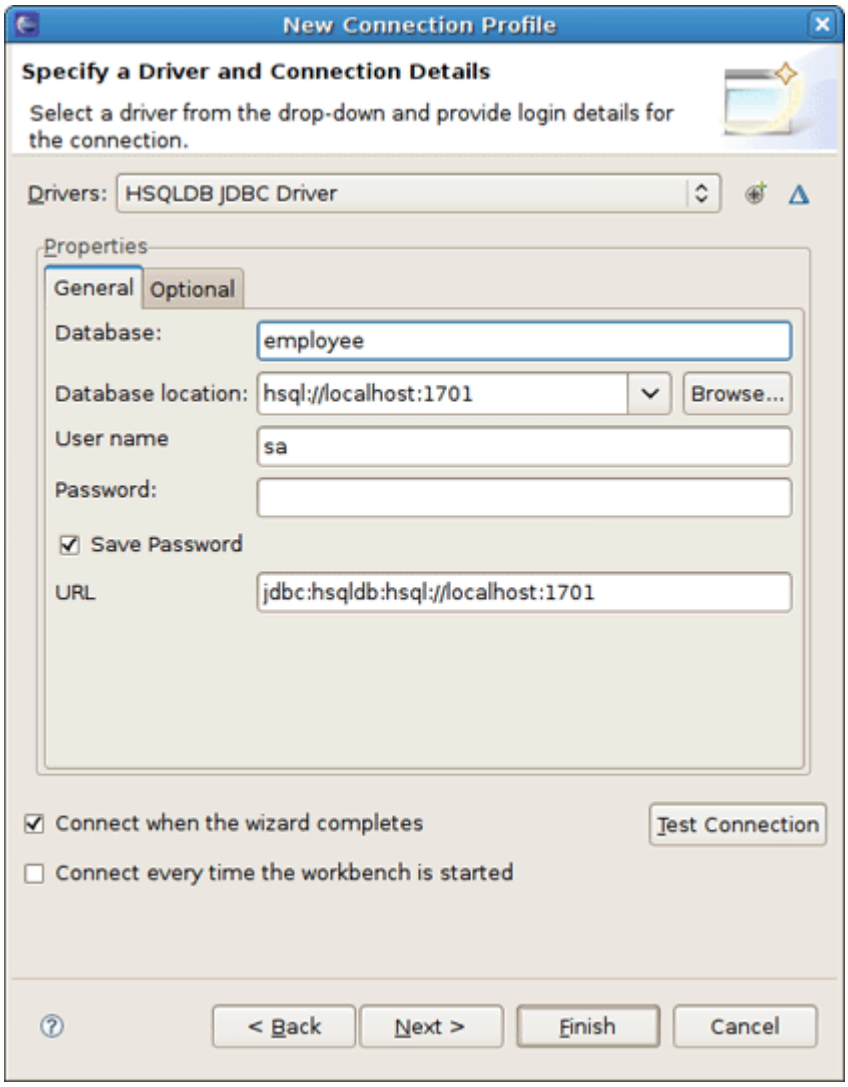


Figure 11.9. Driver and Connection Details

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

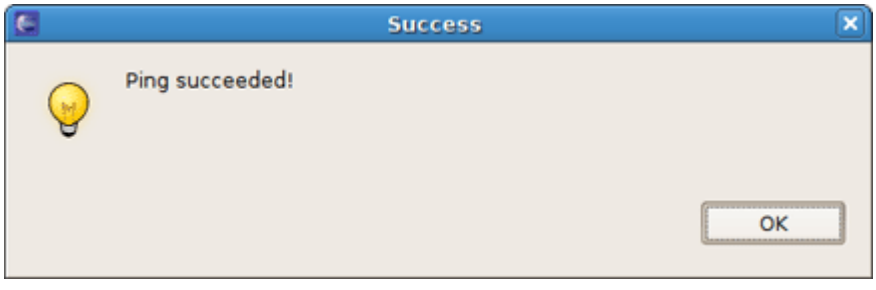


Figure 11.10. JDBC Connection is OK

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

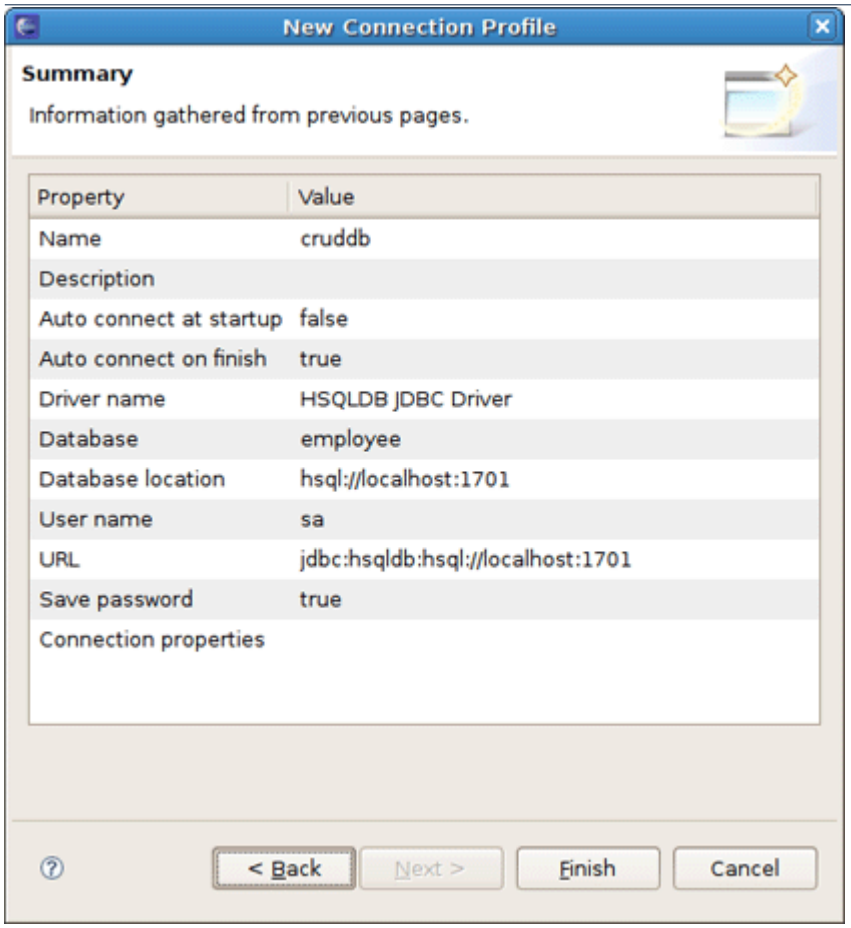


Figure 11.11. Validate JDBC Connection settings

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

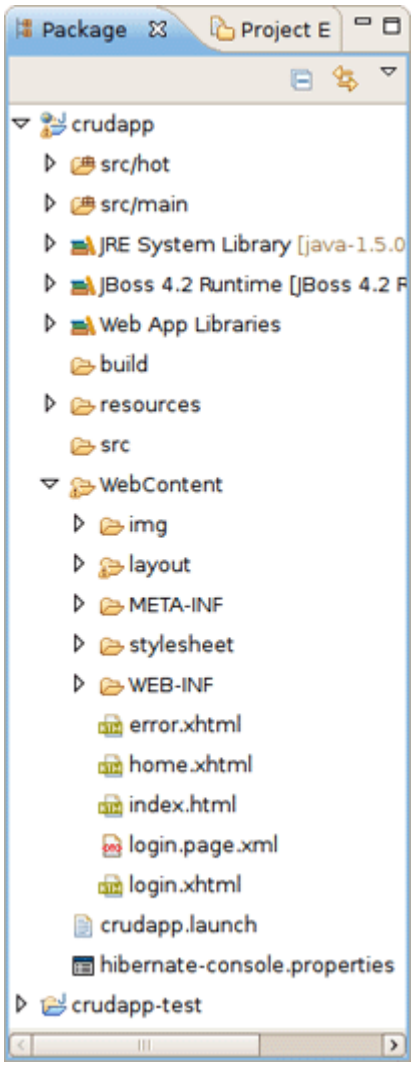


Figure 11.12. CRUDAPP Seam Project

- Have a look at the created projects. You can expand WEB_CONTENT folder and open home.xhtml or login.xhtml with JBoss Visual Editor.

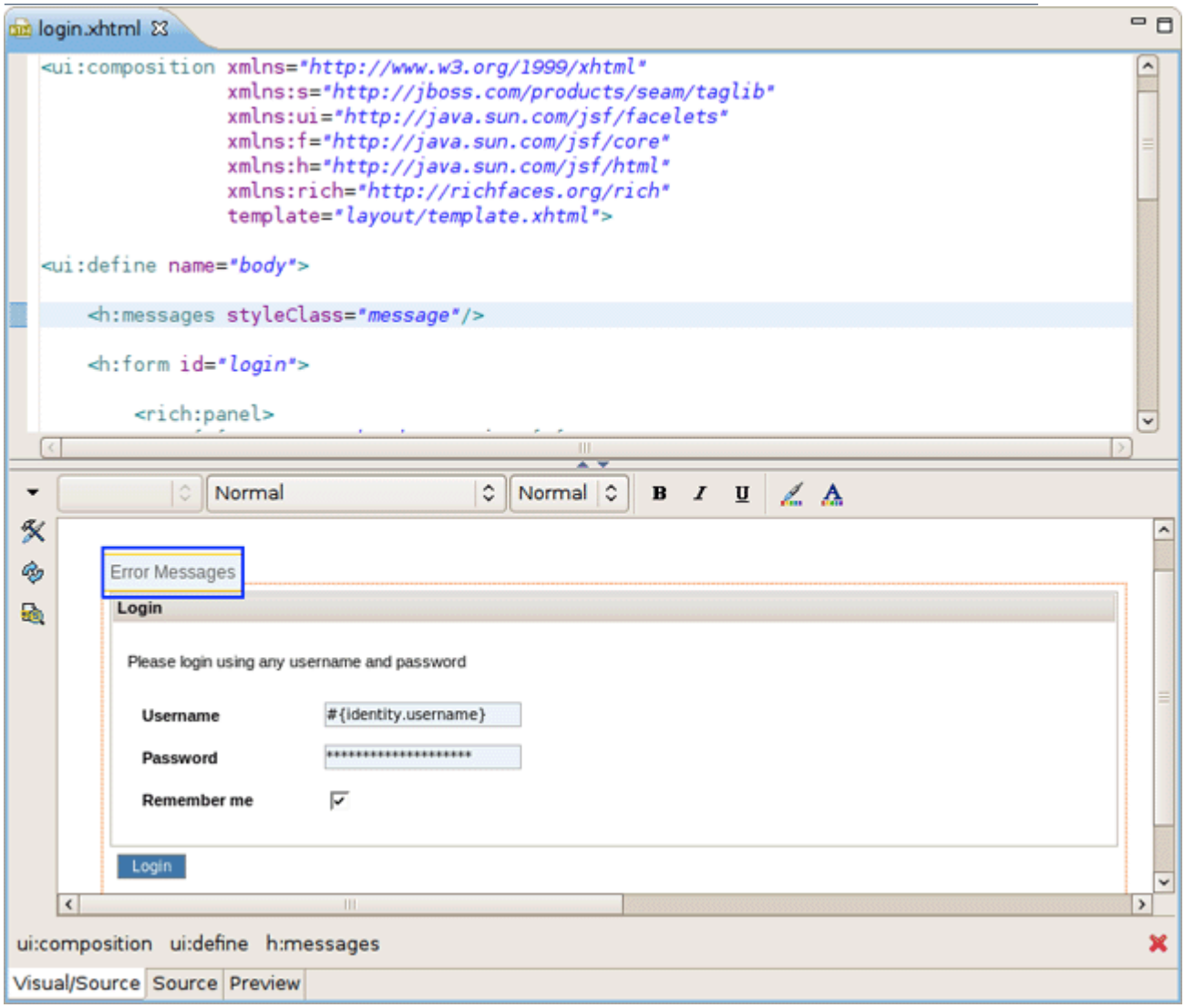


Figure 11.13. login.xhtml in VPE

- Switch to Database Development perspective with *Window->Open Perspective->Other...* and connect to the *cruddb* database.

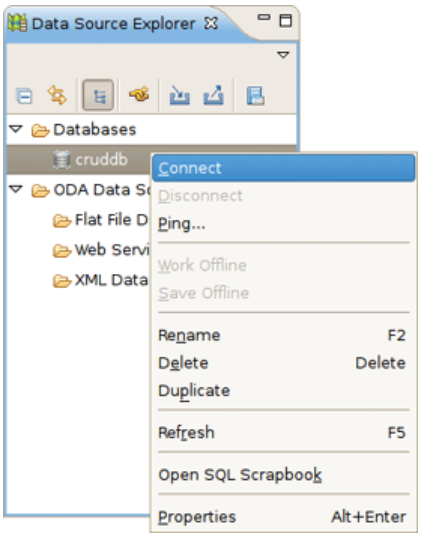


Figure 11.14. Connecting to the CRUDDB database

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

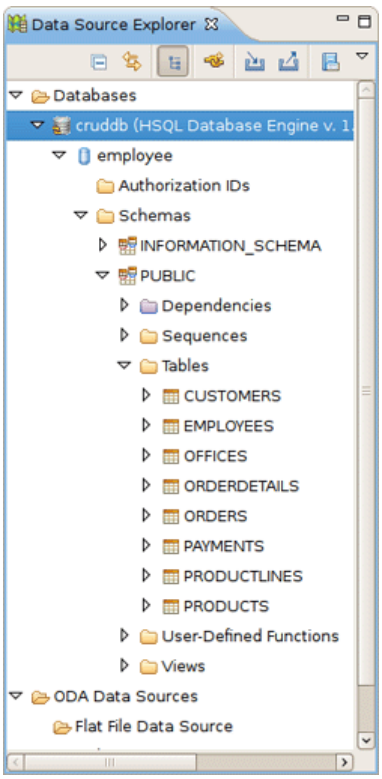


Figure 11.15. CRUDDB Database

- 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 keep everything by default and press *Next*.

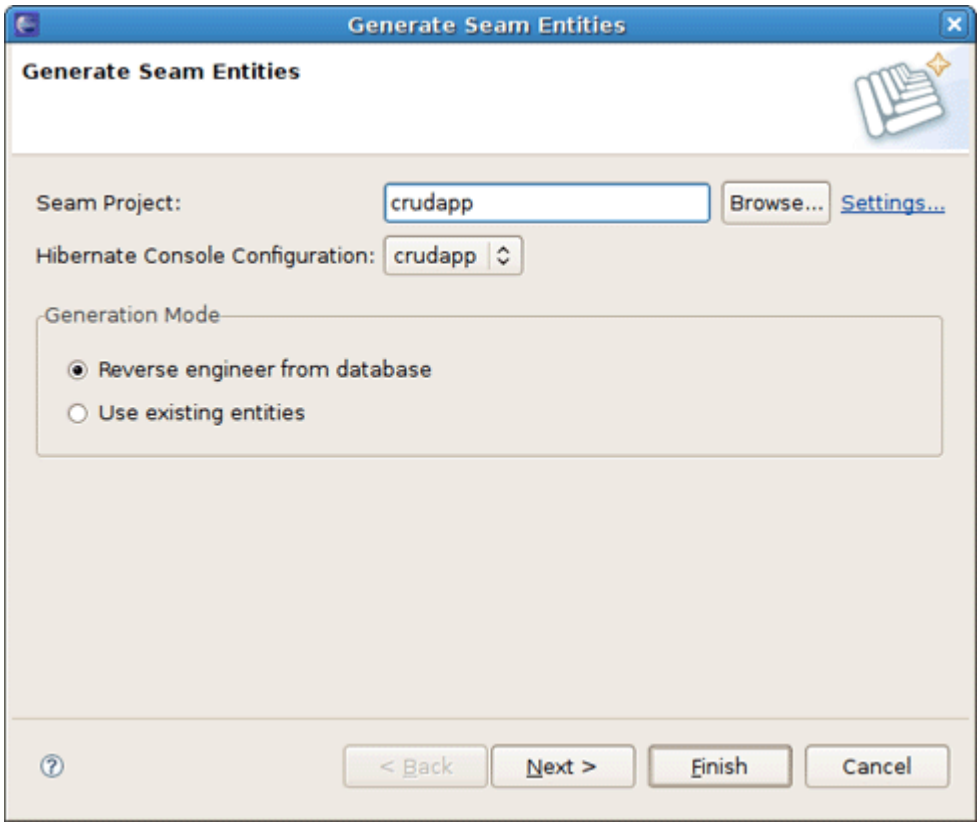


Figure 11.16. Generate Seam Entities

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

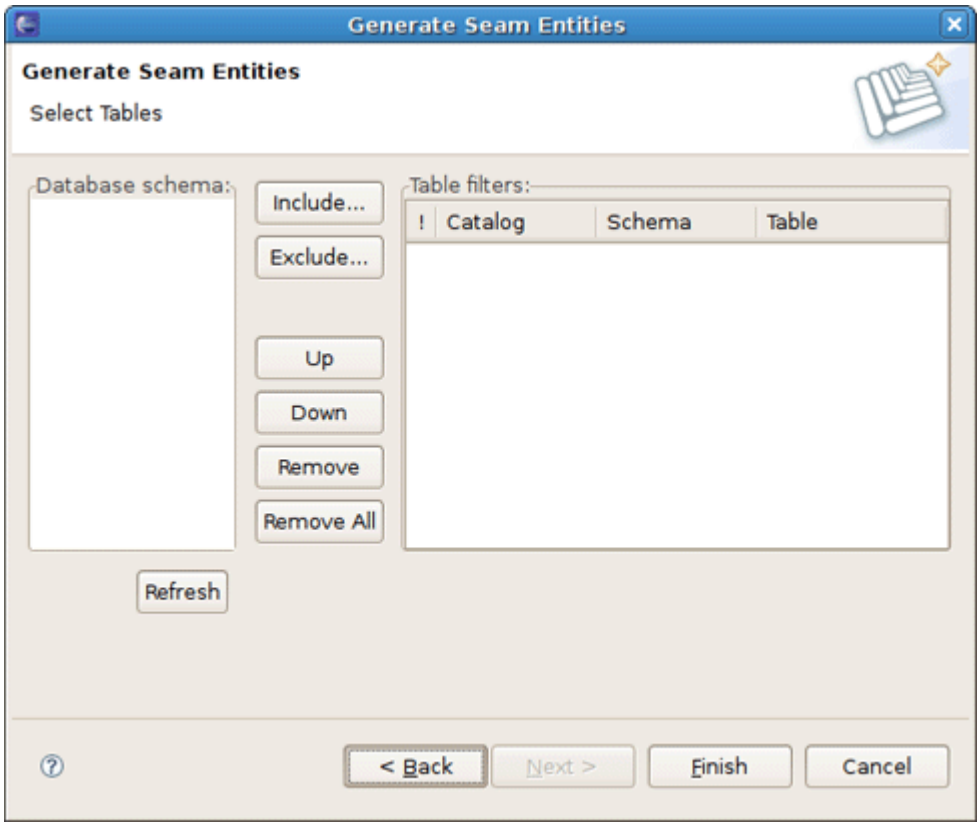


Figure 11.17. Dialog for Selecting Tables

- Now you can filter the tables. Use the *Include* button to include only necessary ones.

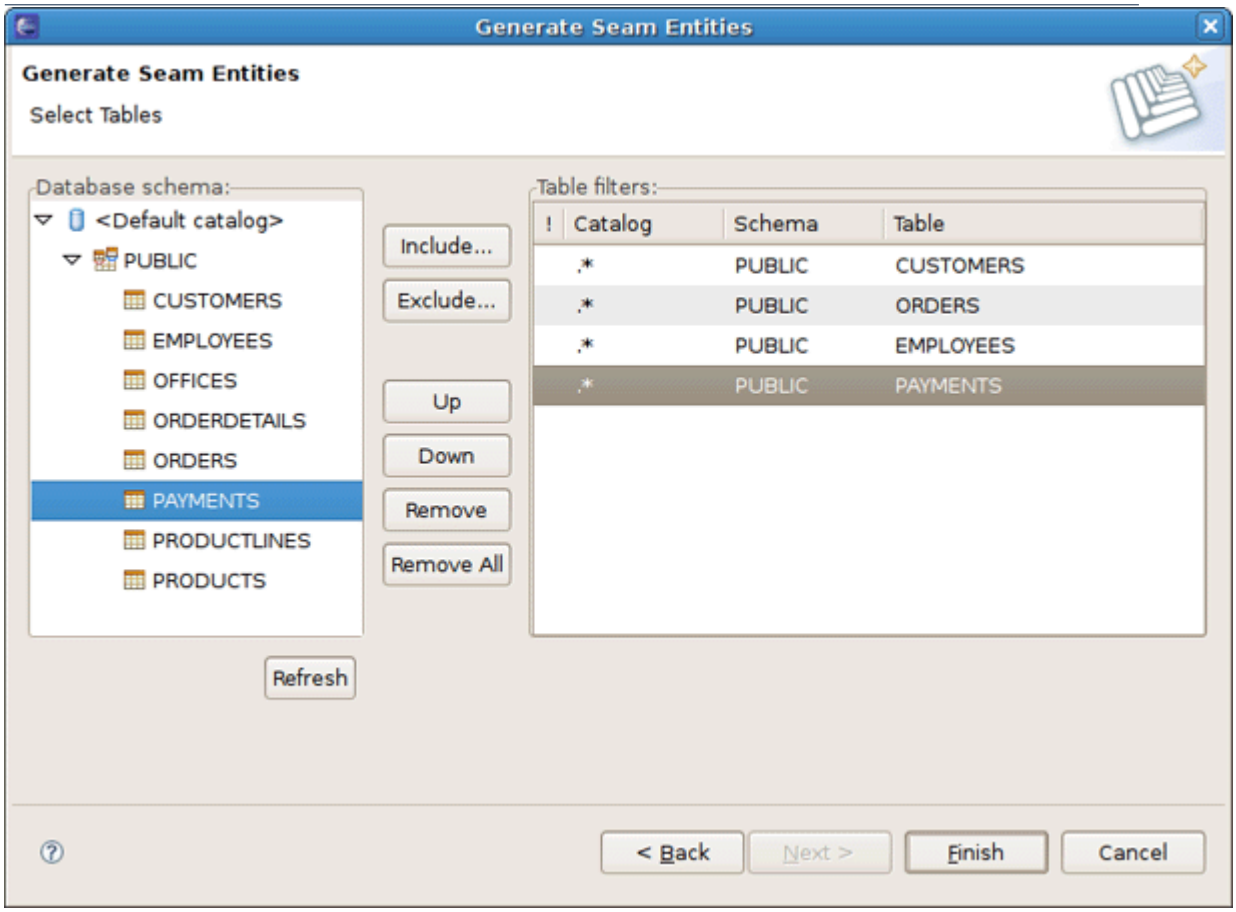


Figure 11.18. Selecting Tables

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

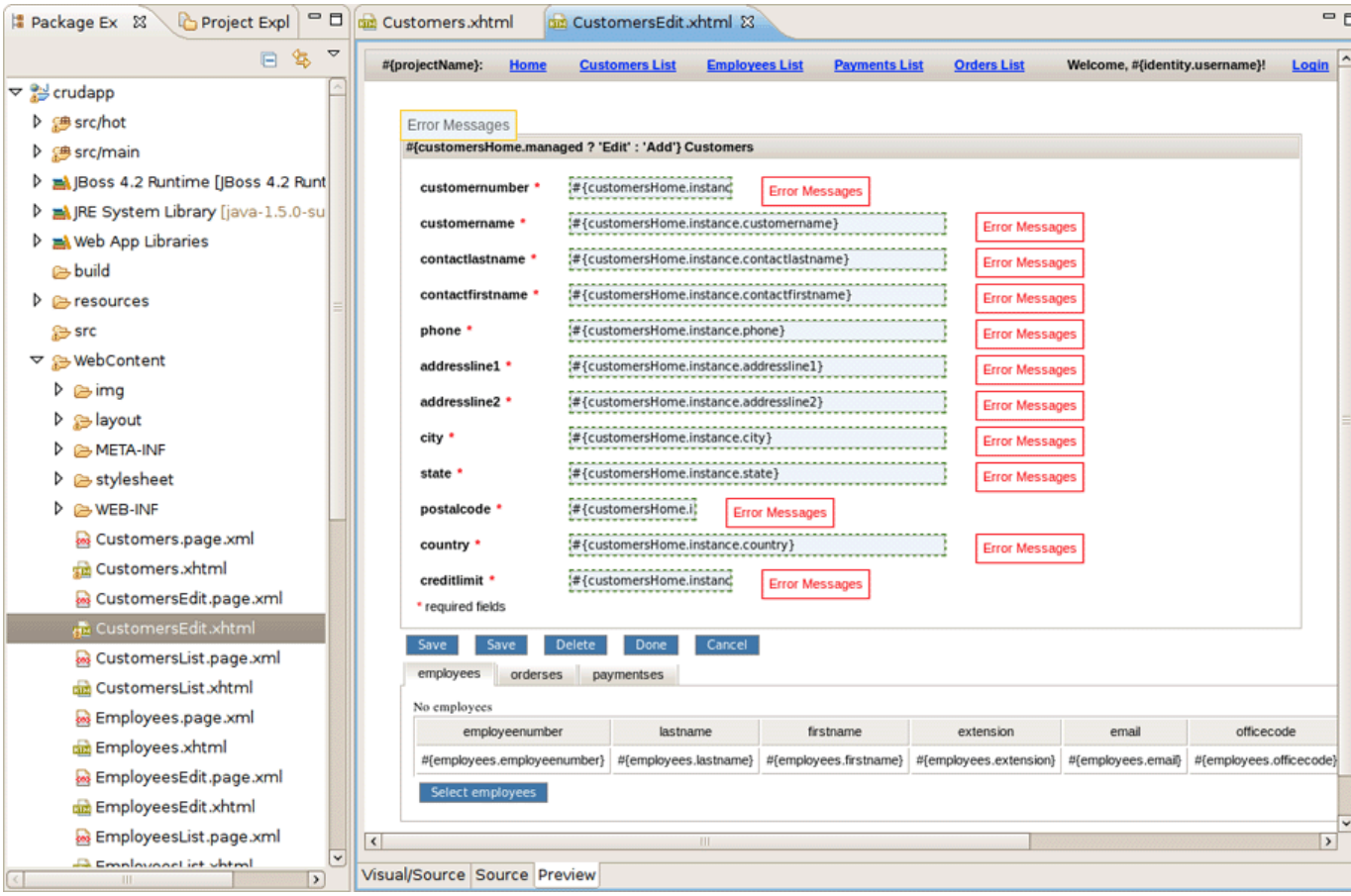


Figure 11.19. Entities Web Pages

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

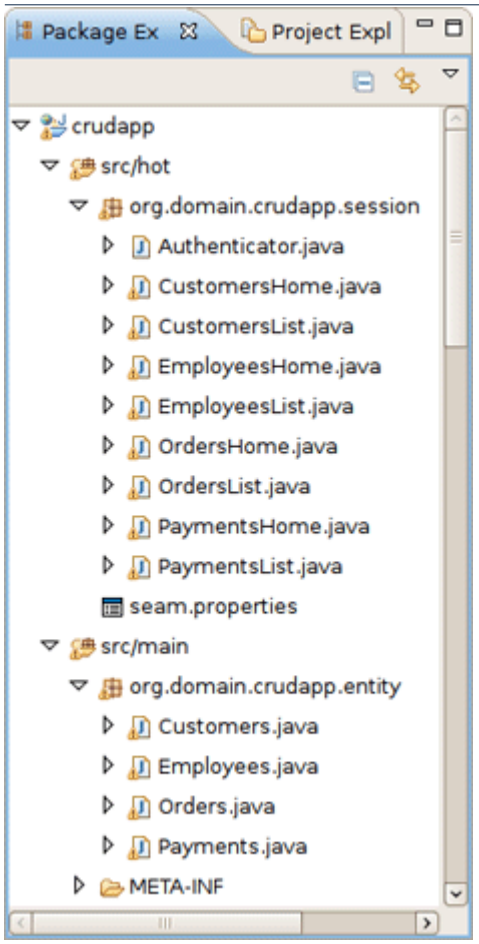


Figure 11.20. Entities Java Classes

- 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.

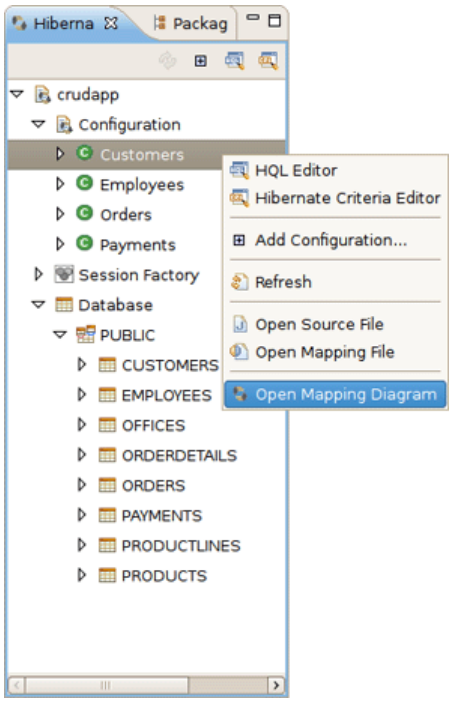


Figure 11.21. Hibernate Configurations View

- 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.

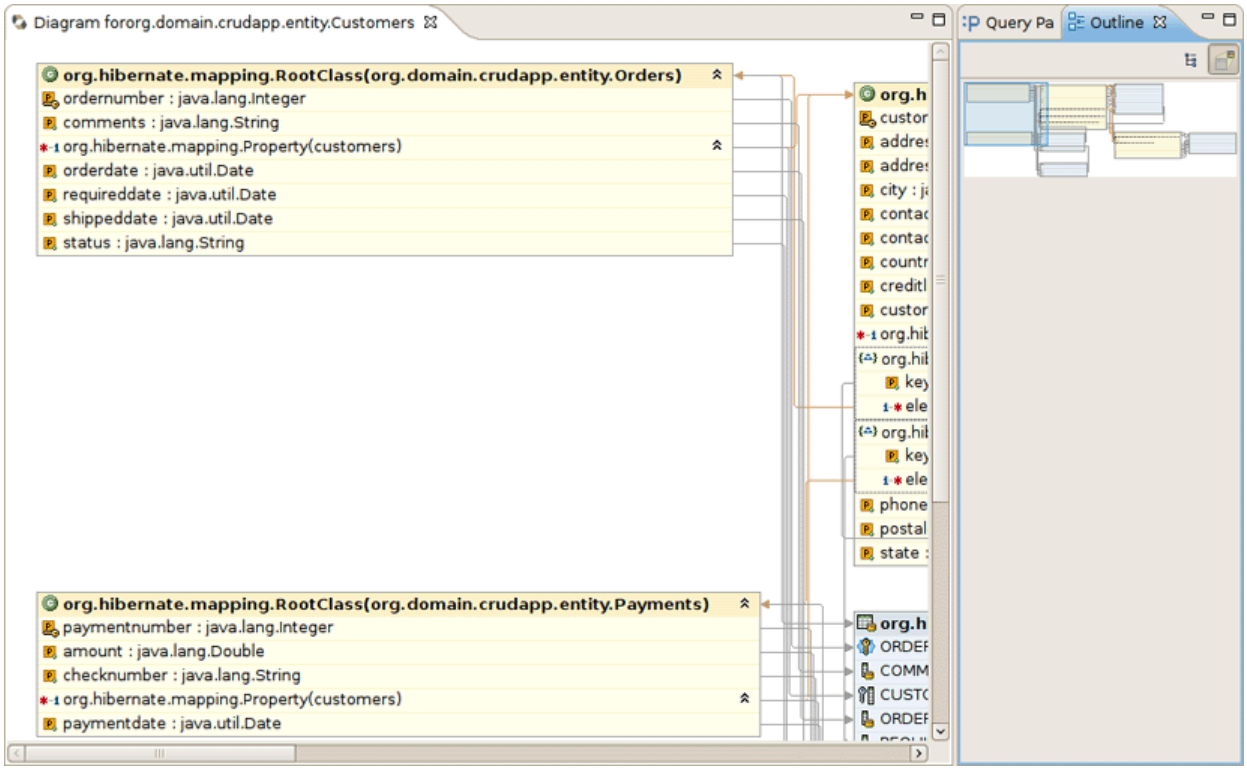


Figure 11.22. Mapping Diagram

- 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.

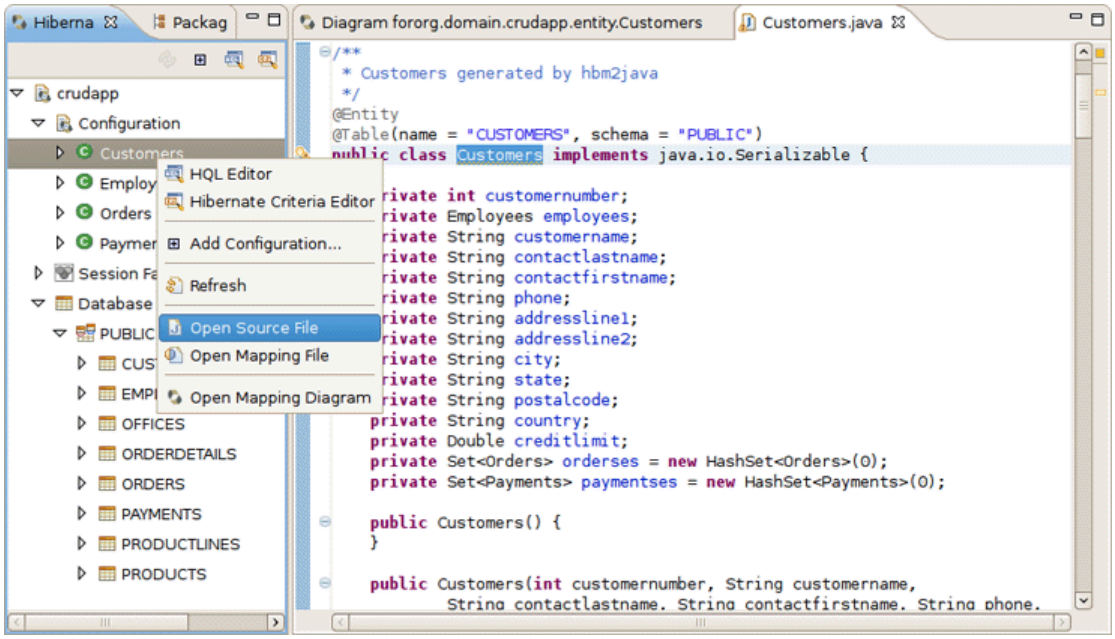


Figure 11.23. Entity class from Mapping Diagram

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.

12.1. 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*.

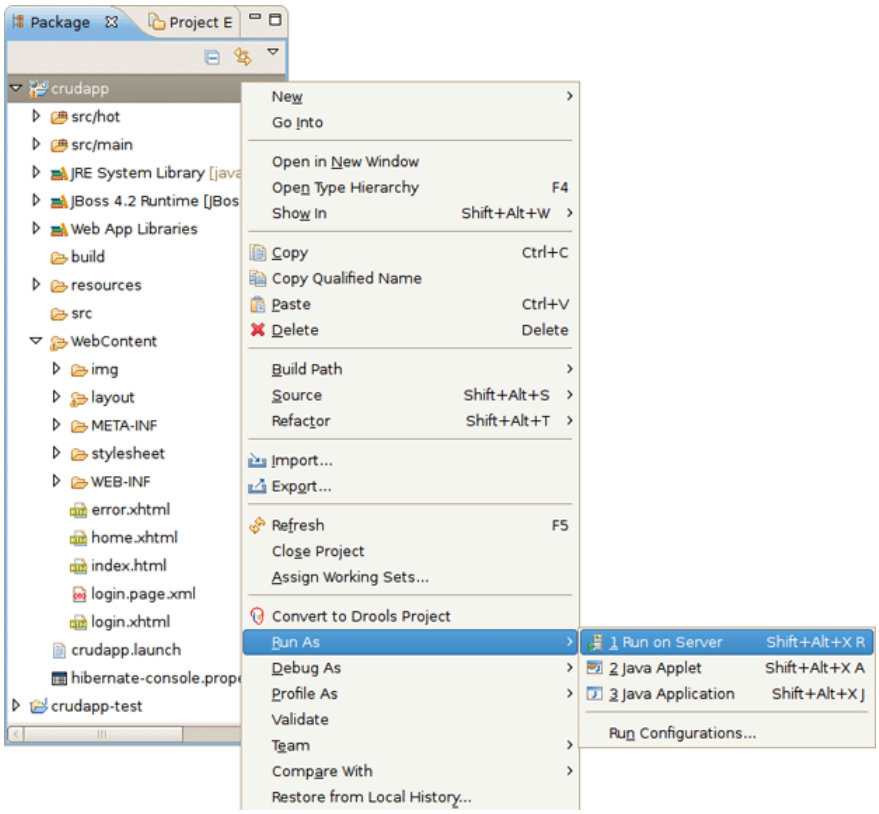


Figure 12.1. Run Project on Server

Select a Server and click Finish

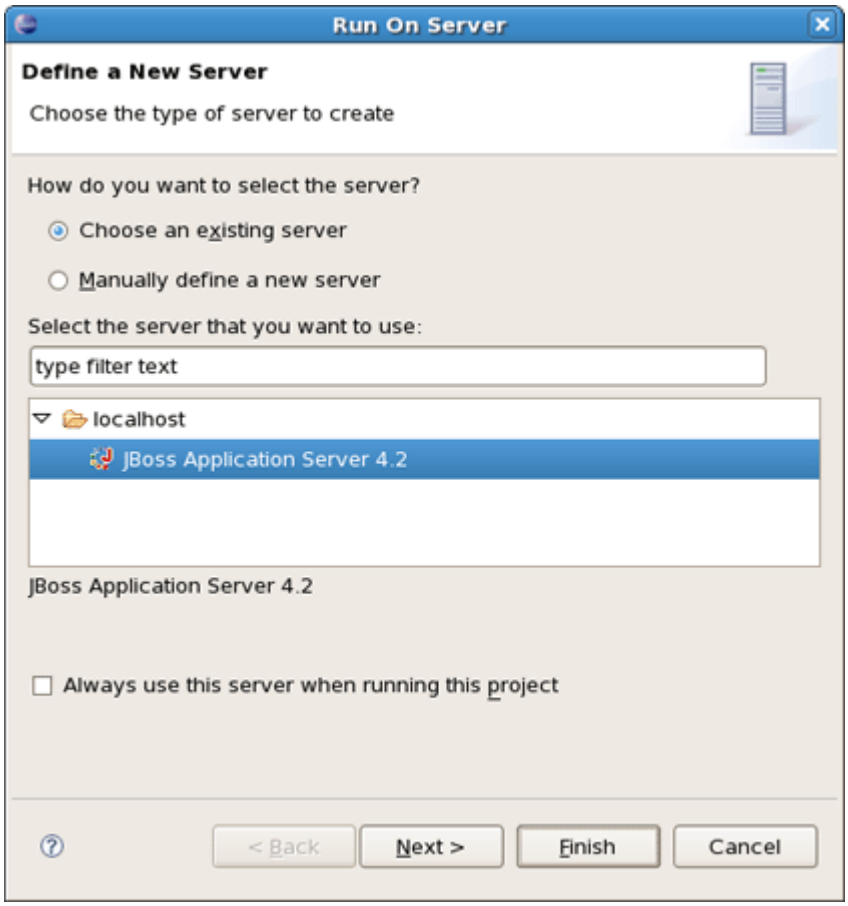


Figure 12.2. Select a Server

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

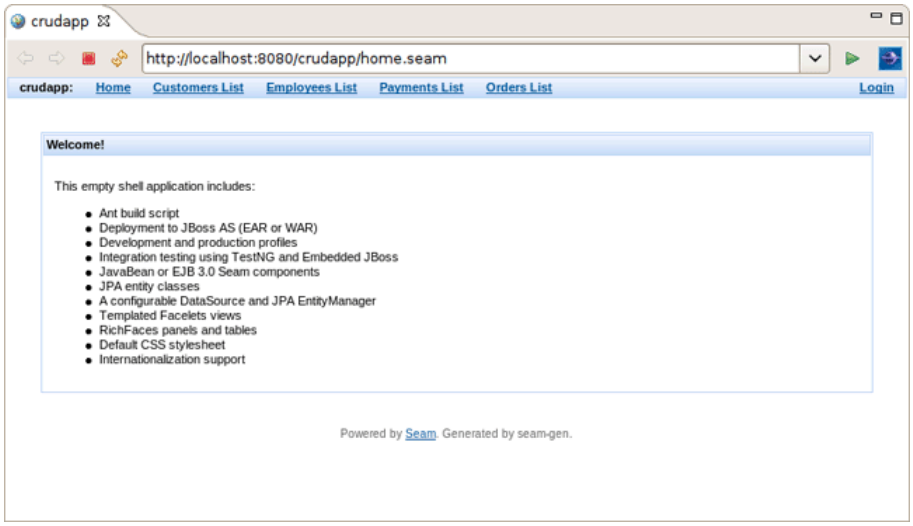


Figure 12.3. Home page

After that you can use CRUD application with "employee" database.

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

Click on the *Employees List* link and observe that data from *employee* database is displayed.

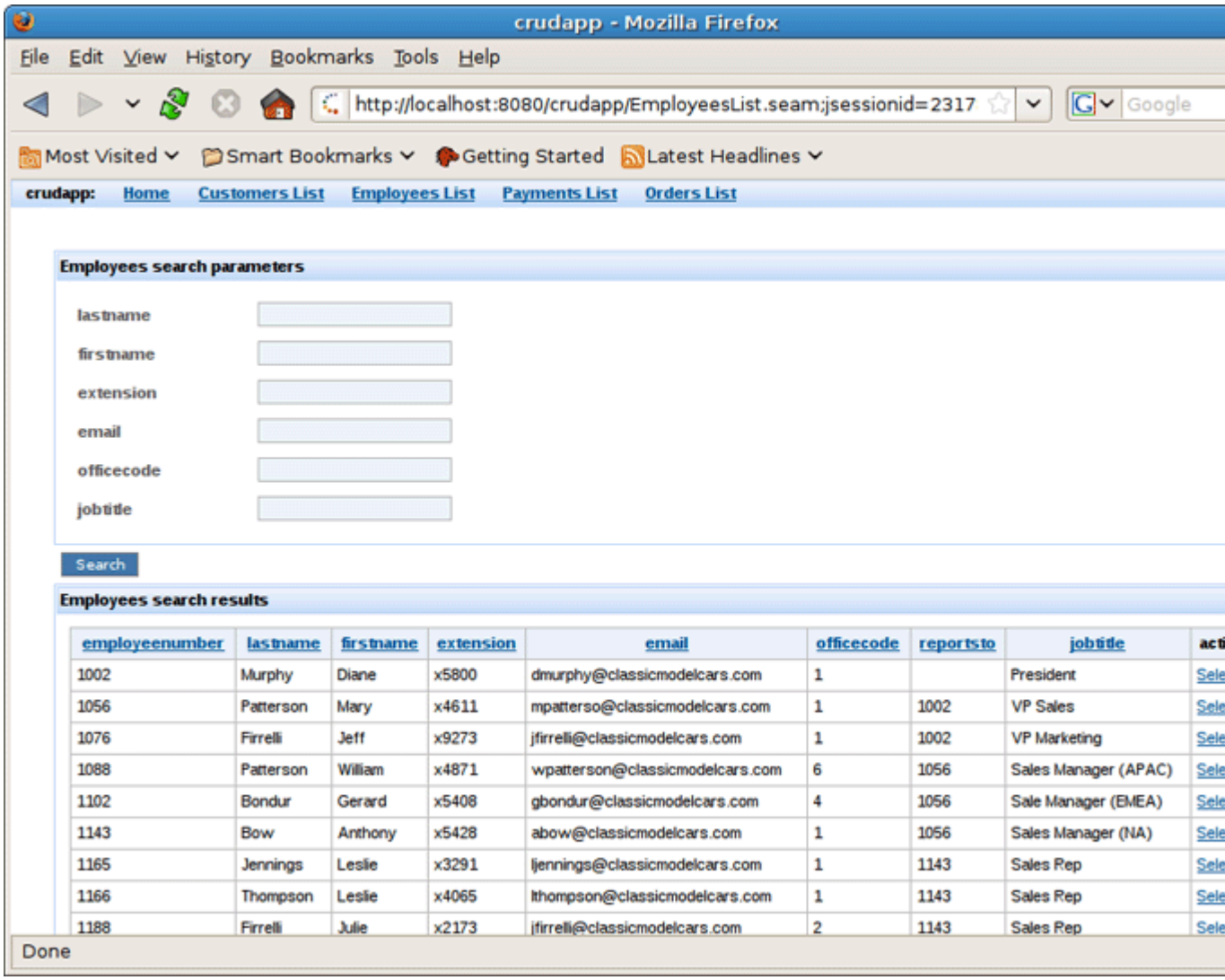


Figure 12.4. Employees List

Use Employees search parameters fields to filter the selected list.

Press Select opposite one of employees.

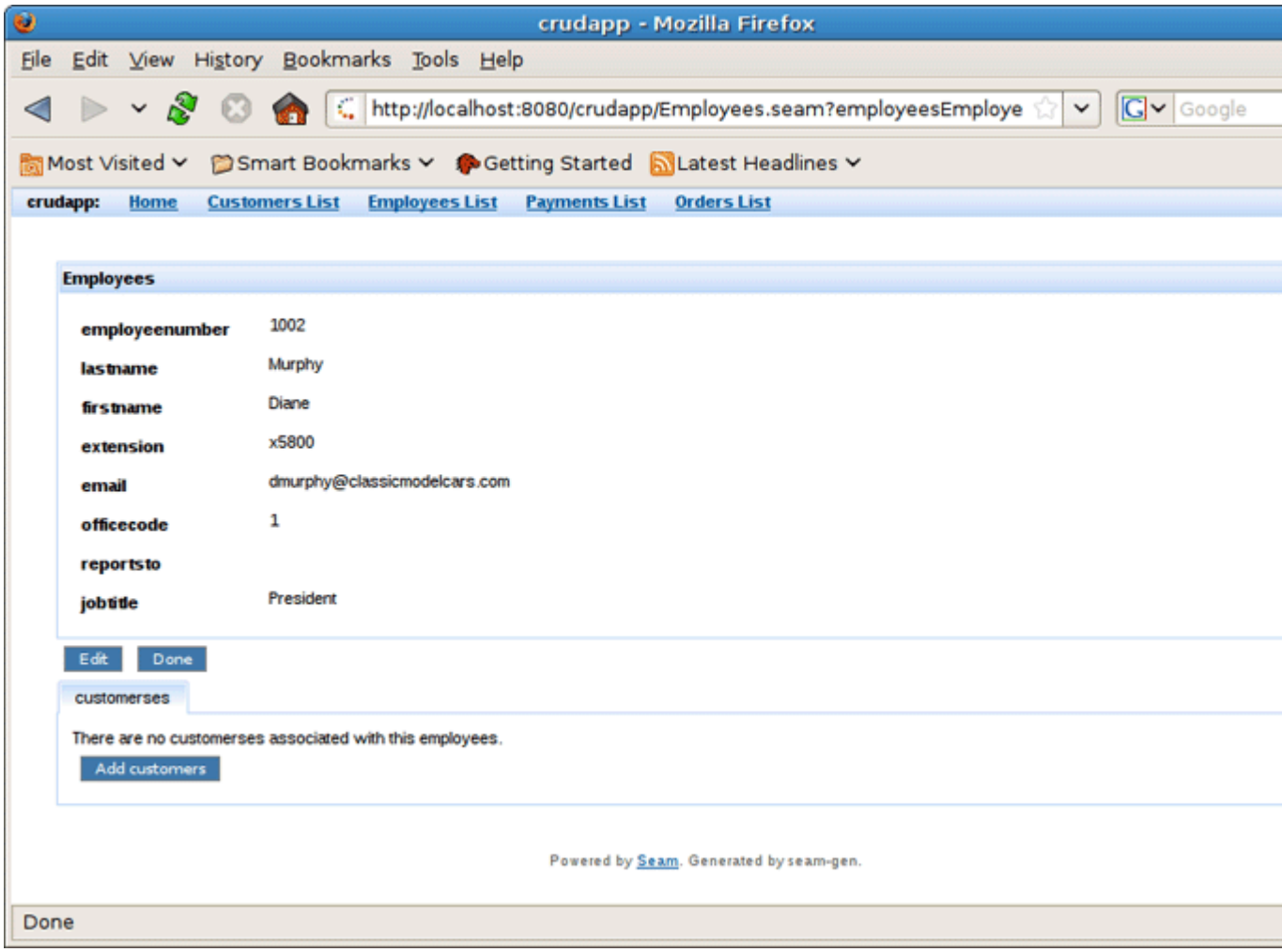


Figure 12.5. Employee details

Press Edit to edit employee fields.

Enter *Login* and *Password* to login. (Use *"crudapp-user"/"secret"* for example)

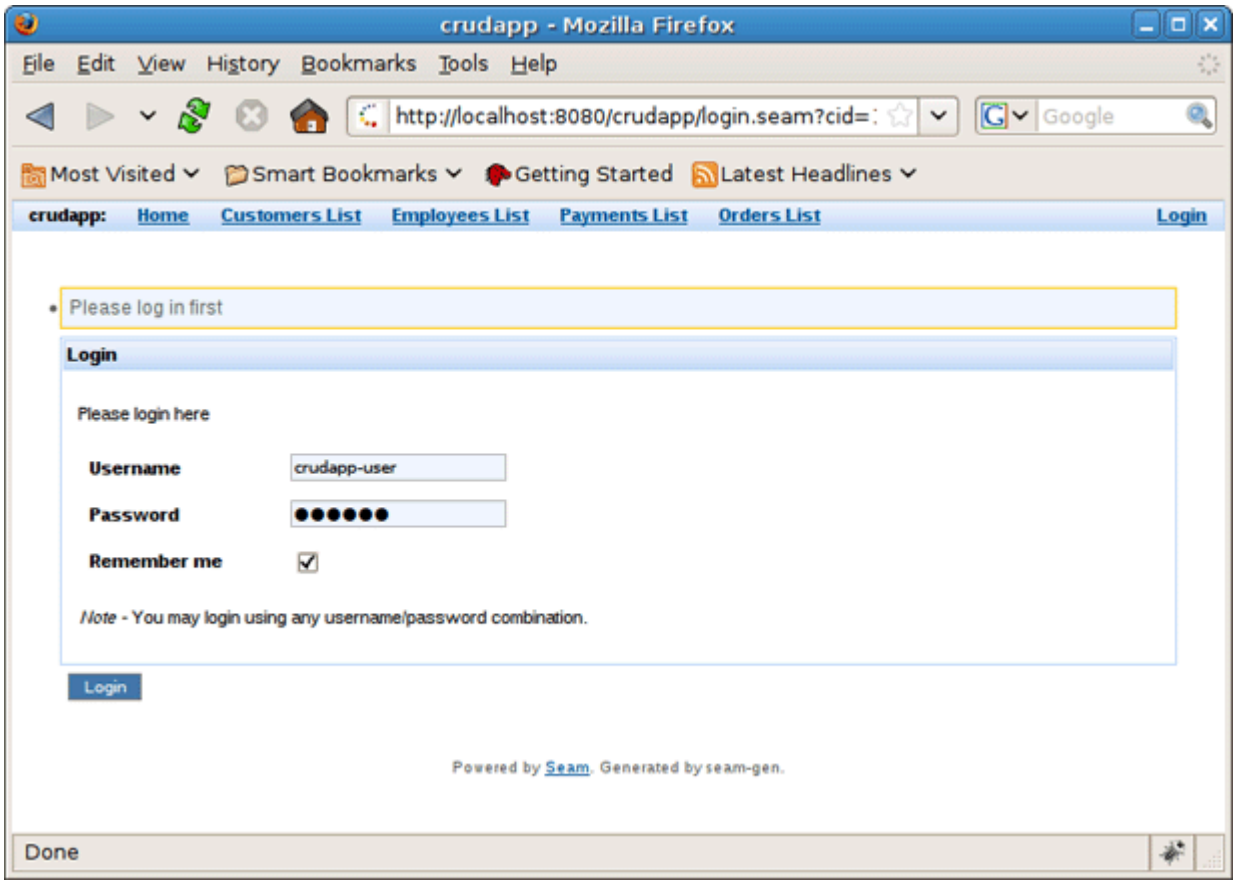


Figure 12.6. Login page

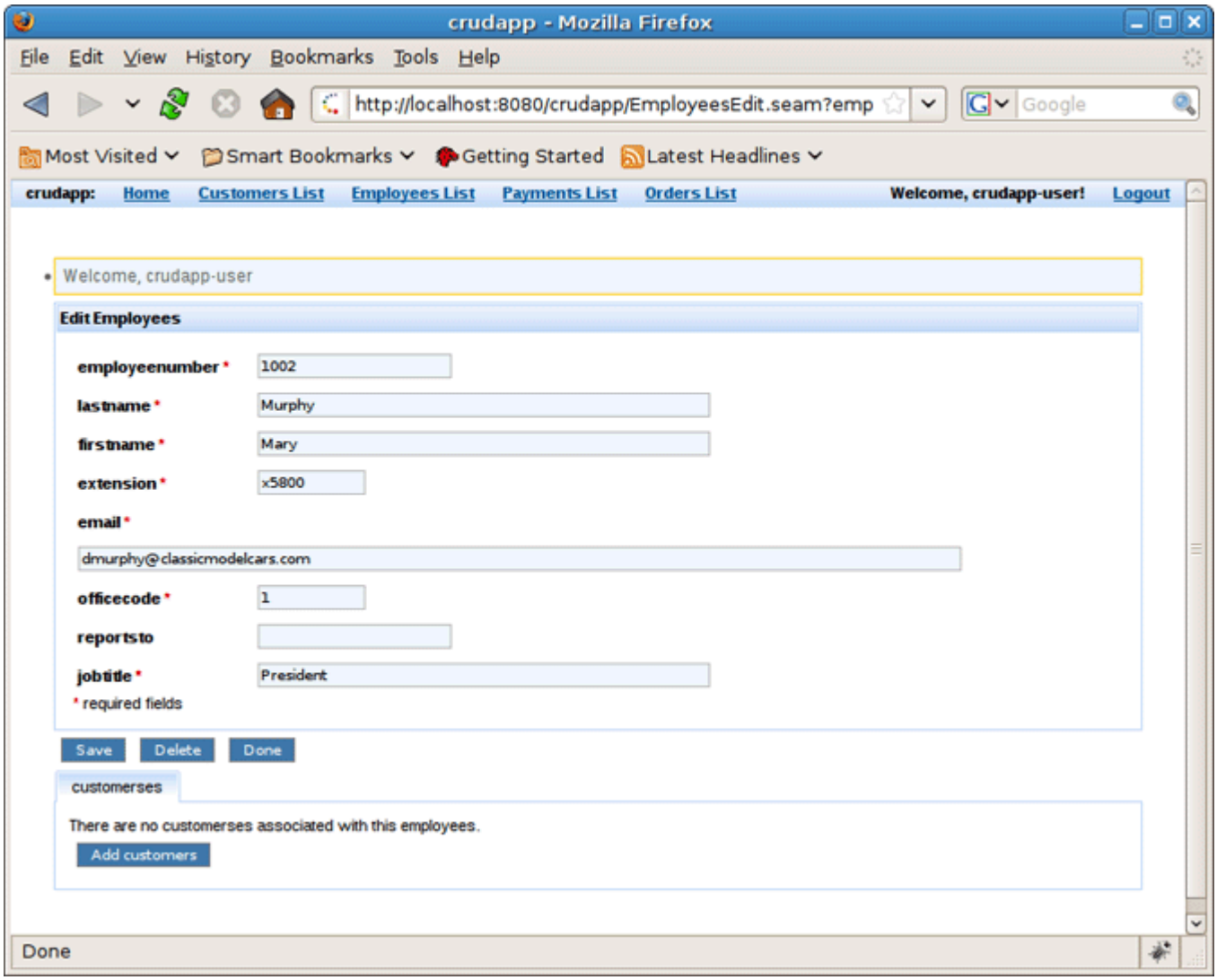


Figure 12.7. Edit Employee

Fill in *firstname* and press Save.

Database will be updated.

Using TestNG project

With the help this chapter you will get to know with TestNG.

13.1. 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

More information can be found on home page: *www.testng.org* [http://testng.org]

13.2. Other relevant resources on the topic

Next-Generation Testing with TestNG (An Interview with Cedric Beust) [http://www.artima.com/lejava/articles/testng.html]

TestNG: The next generation of unit testing [http://www.javaworld.com/javaworld/jw-04-2005/jw-0404-testng.html]

Test Categorization Techniques with TestNG [http://dev2dev.bea.com/pub/a/2006/09/testng-categorization.html]

TestNG makes Java unit testing a breeze [http://www-128.ibm.com/developerworks/java/library/j-testng/]

In pursuit of code quality: JUnit 4 vs. TestNG [http://www-128.ibm.com/developerworks/java/library/j-cq08296/index.html]

13.3. 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.

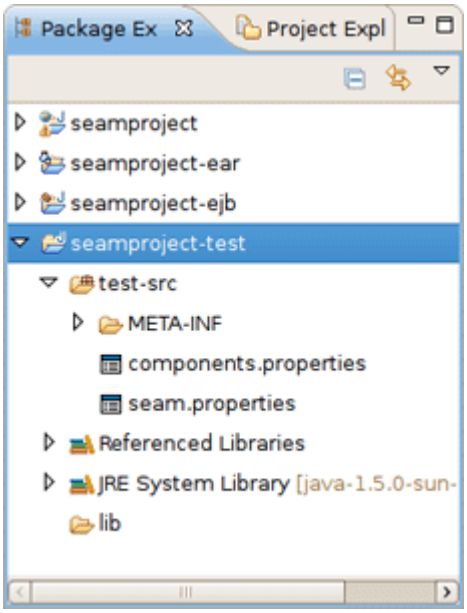


Figure 13.1. Seam-test Project

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

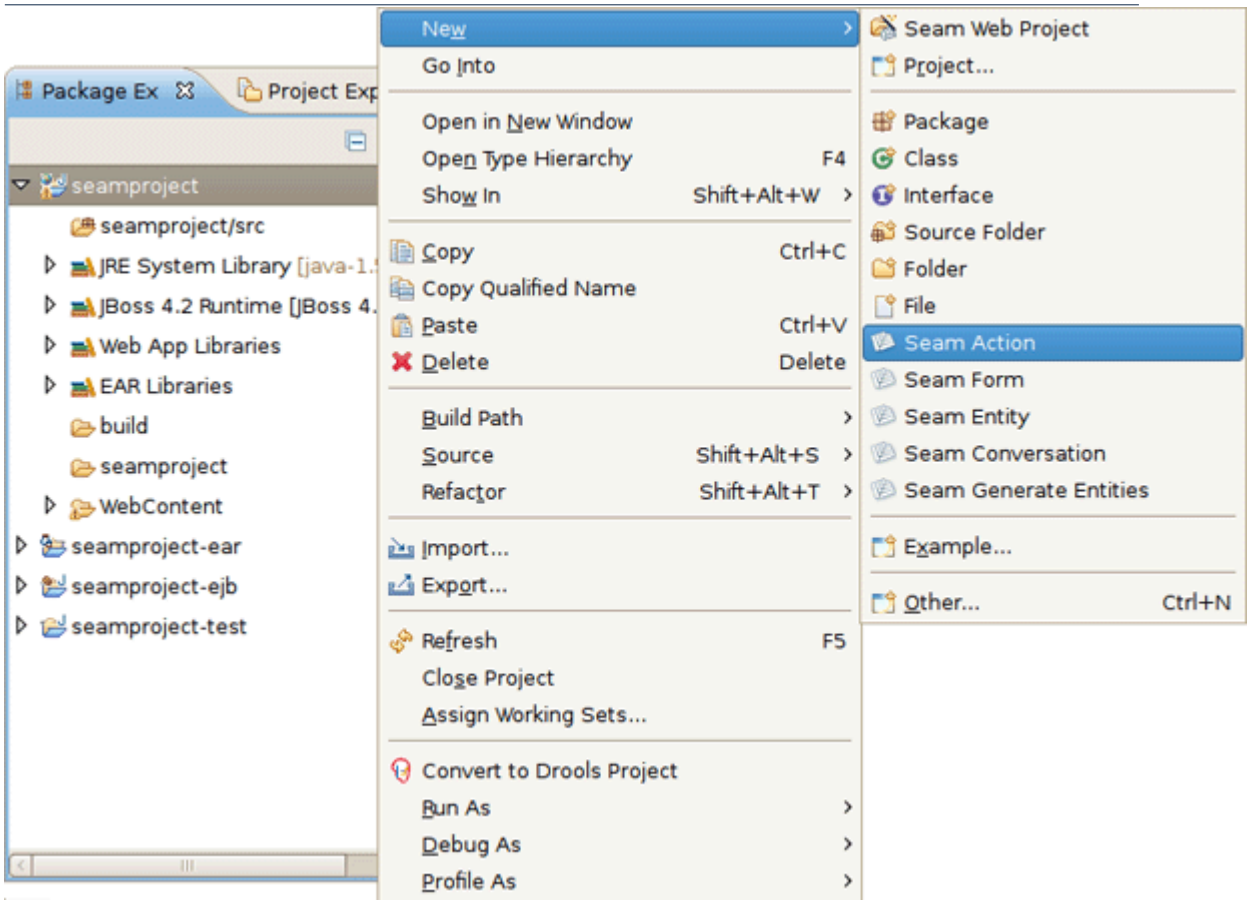


Figure 13.2. Seam Action Creation

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

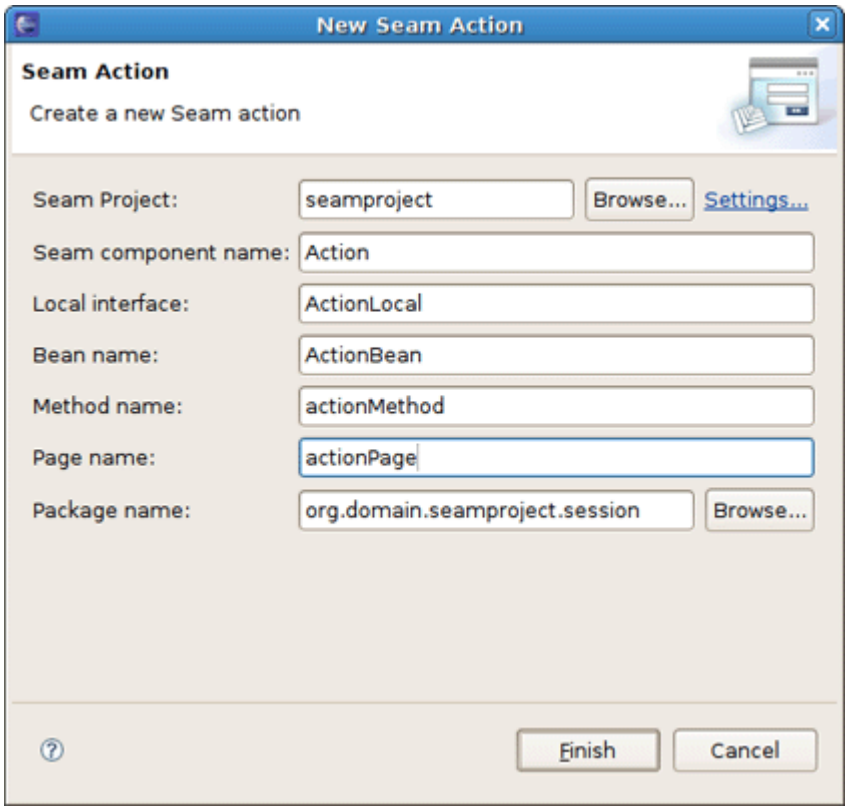


Figure 13.3. New Seam Action Wizard

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

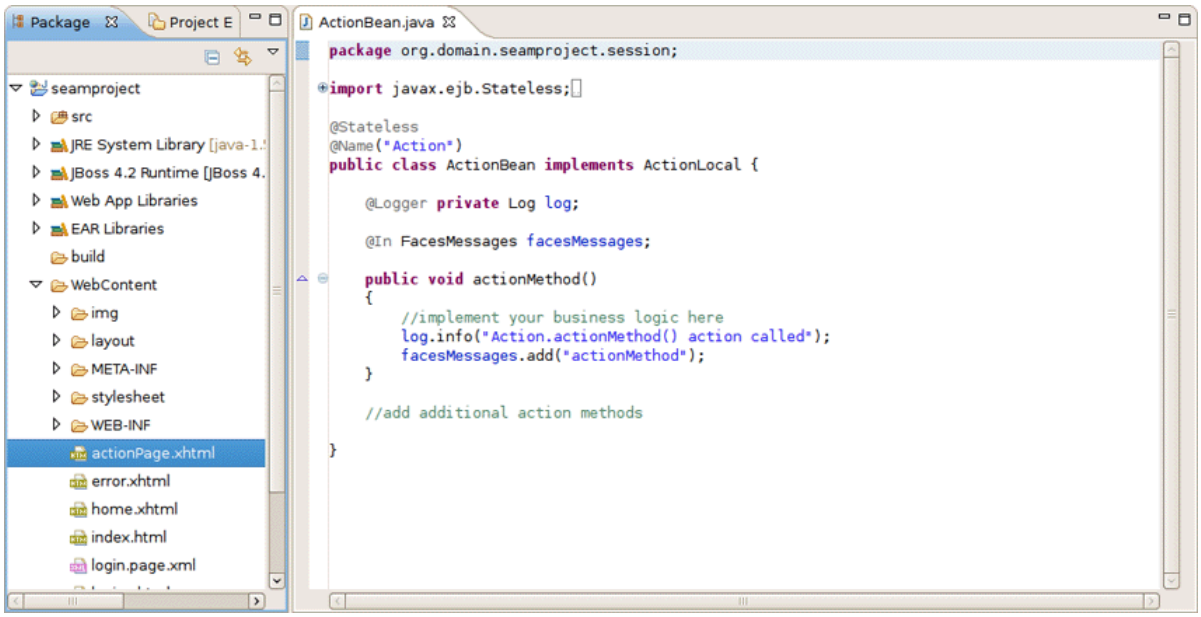


Figure 13.4. Created Action

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

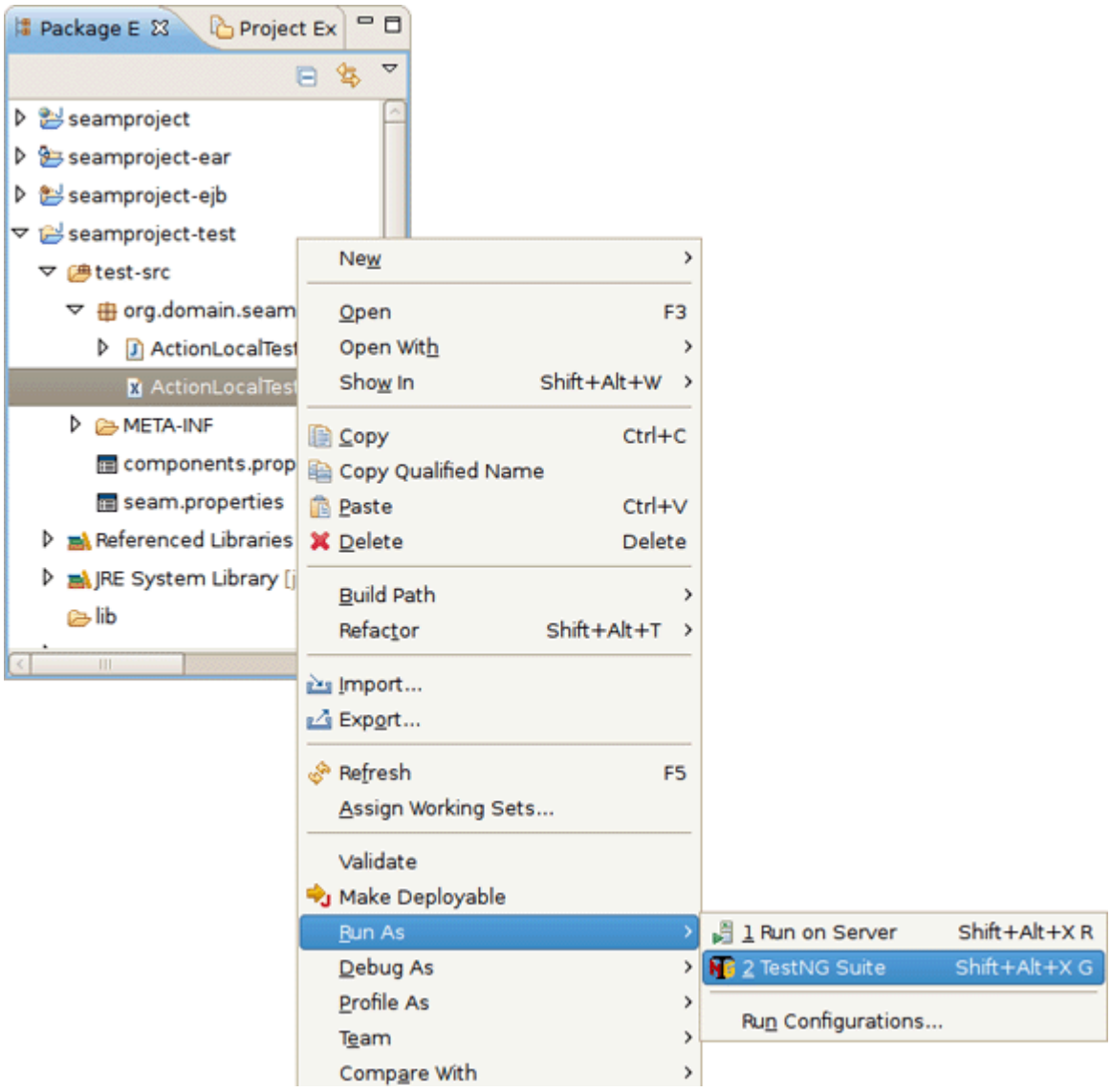


Figure 13.5. Running TestNG

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

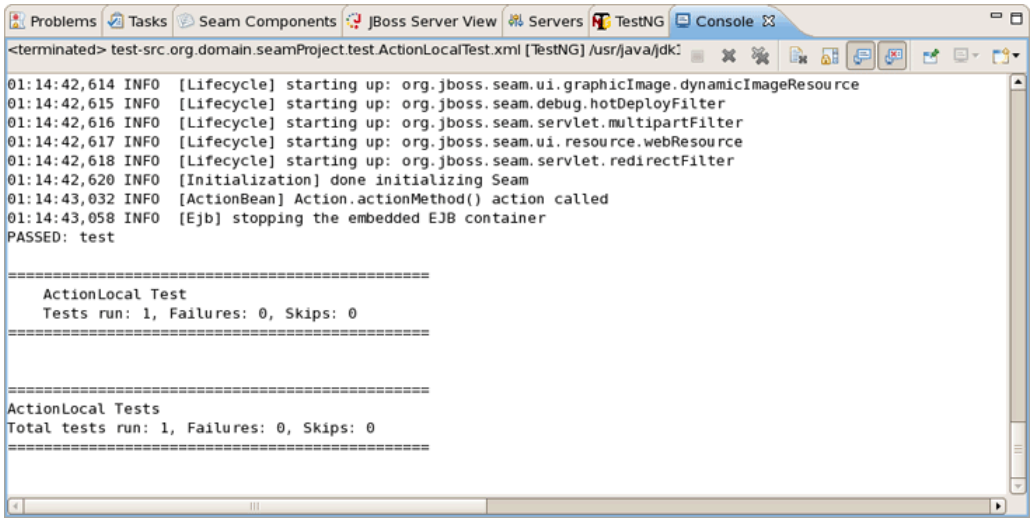


Figure 13.6. Test is Finished

- 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.

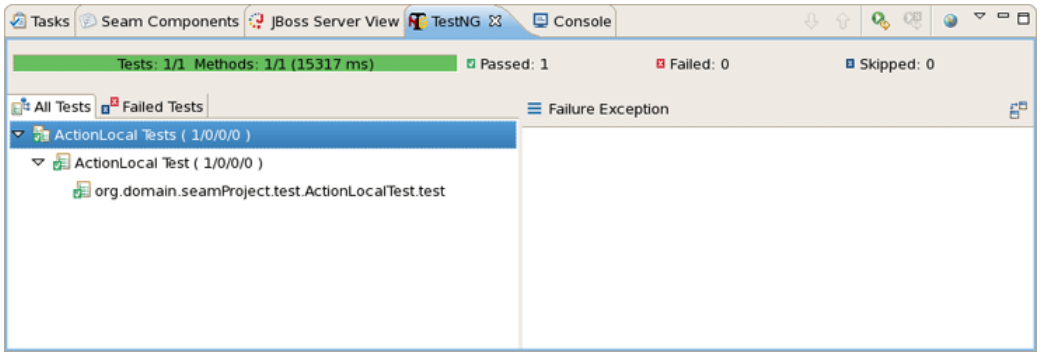


Figure 13.7. Viewing the Test Results

You can see the test results in Web Browser.

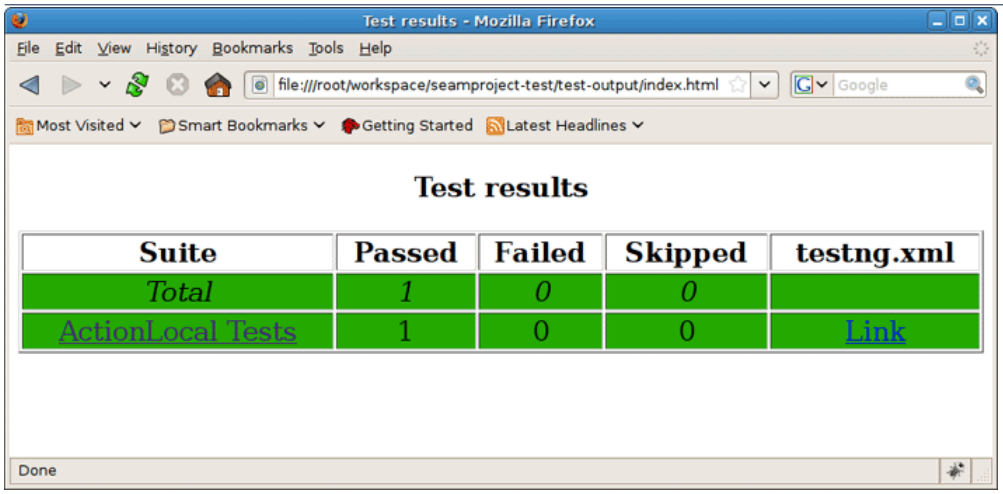


Figure 13.8. Test Results in Browser

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

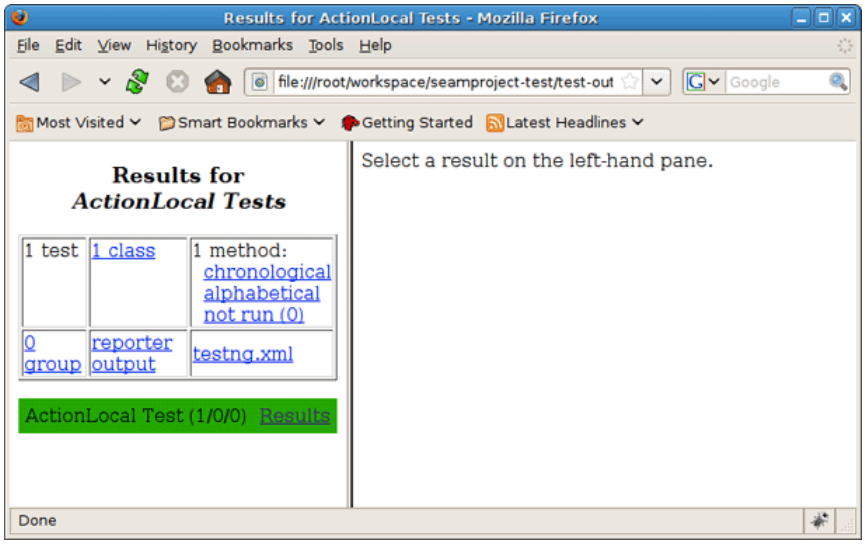


Figure 13.9. Test Information

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

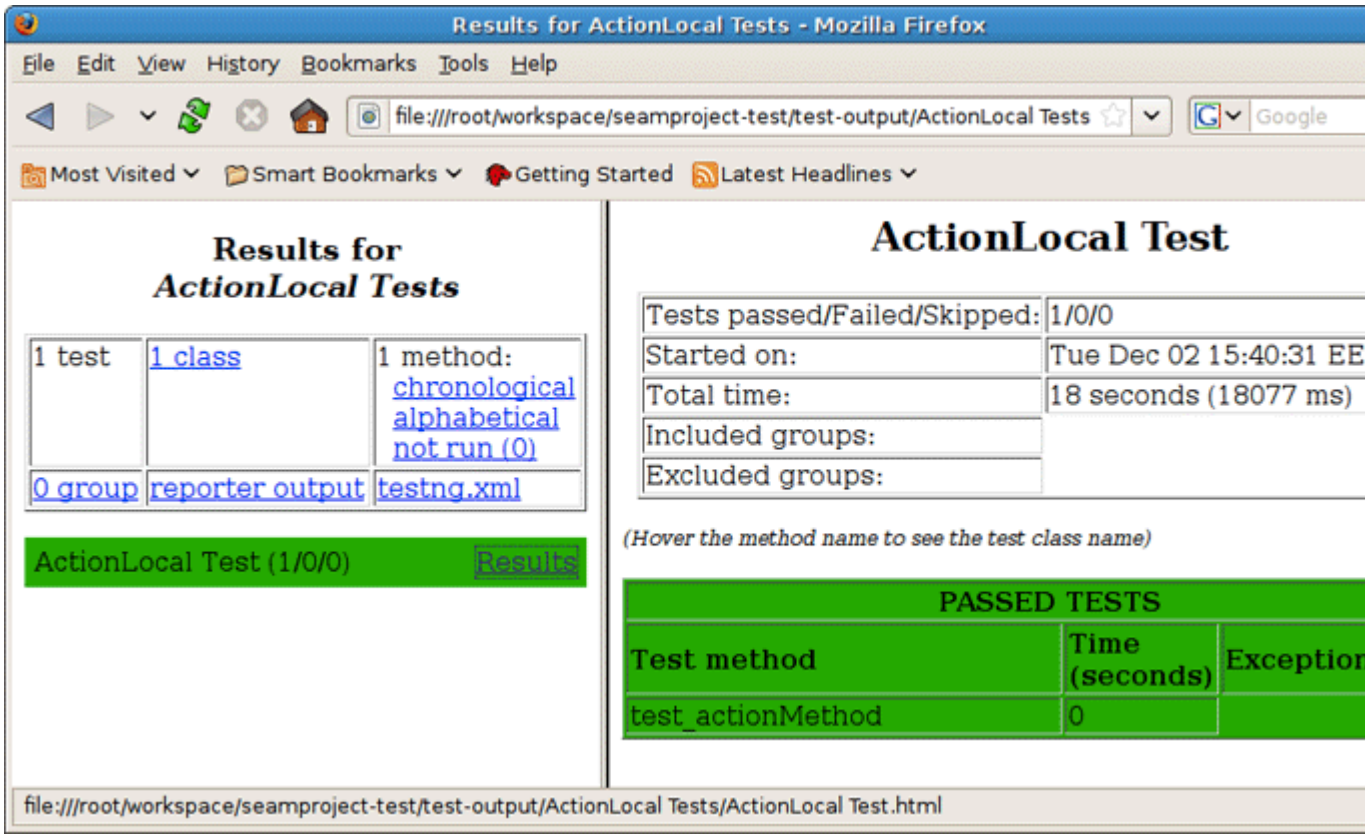


Figure 13.10. ActionLocal Test Details

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.

In conclusion, the main goal of this document is to get you know with a full featureset that JBoss Tools provides to support Seam development. Thus if you have some questions, comments or suggestions on the topic, please fell free to ask in the *JBoss Tools Forum* [http://www.jboss.com/index.html?module=bb&op=viewforum&f=201]. You can also influence on how you want to see JBoss Tools docs in future leaving your vote on our page *here* [http://wiki.jboss.org/wiki/JBossToolsDocsFuture].

A set of movies on Seam tooling is available *here* [http://docs.jboss.org/tools/movies].