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

<table>
<thead>
<tr>
<th>Feature</th>
<th>Benefit</th>
<th>Chapter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integrated JSF with EJB 3.0</td>
<td>JBoss Seam combines the component models of EJB3 and JSF, eliminating glue code, and thus, letting the developer think about the business problem.</td>
<td></td>
</tr>
<tr>
<td>Integrated AJAX</td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>Bijection</td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>New Seam Project wizard</td>
<td>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.</td>
<td>Chapter 2, Creating a New Seam Project via the New Seam Project wizard</td>
</tr>
<tr>
<td>Seam component generations options</td>
<td>All the Seam component generations options known from Seam-gen are available as</td>
<td>Chapter 6, Seam Wizards</td>
</tr>
</tbody>
</table>
Feature | Benefit | Chapter
--- | --- | ---
Seam Editors | wizards (with sensible auto-defaulting) for creating various common Seam components. | Chapter 8, Seam Editors
Seam Validation | A configured Seam Validator provides validation of various possible problematic definitions. | Section 8.4.4, “Seam Validation”
CRUD Database Application Generation | JBoss Seam allows adding CRUD support to a Seam Web application. | Chapter 13, Generate a CRUD Database Application
TestNG Framework | TestNG (“Testing, the Next Generation”) is a Java unit testing framework that aims to overcome many limitations of JUnit. | Chapter 15, Using TestNG project

1.3. Installation into Eclipse

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


1.4. Other relevant resources on the topic

The [Seam Framework][1] - Next generation enterprise Java development.

All JBoss Developer Studio/JBoss Tools release documentation you can find at [http://docs.jboss.org/tools][2] in the corresponding release directory.

There is some extra information on [JBoss Wiki][3].

The latest documentation builds are available at [http://download.jboss.org/jbosstools/nightly-docs][4].

You can find a set of screencasts on how to search/find references and use Seam refactoring tools in the following Max Andersen’s [blog post][5].
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 to set 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.

![Image](image1.png)

**Figure 2.1. Seam Perspective Icon**

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

![Image](image2.png)

**Figure 2.2. 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 → 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.
Chapter 2. Creating a New Sea...

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 your project configuration. 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.
Chapter 2. Creating a New Sea...

Figure 2.5. Specifying Target Runtime Configurations

Clicking on Finish returns you to the Figure 2.3, “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 combox 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 a chosen server. Leave everything as it is and click on *Next*.
Figure 2.7. JBoss Runtime Summary

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 the Application Server Manager Guide to find out more about runtimes and servers.
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.
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 runtime is checked here, the *Make Primary* button won't be dimmed yet. Thus, you can make use of it to mark primary runtime.
2.2.1. Seam project with JPA facets

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

Then you should press Next and may leave the default value on Java application building page and Web Module Settings page.

After that you will see JPA facet page.
Chapter 2. Creating a New Sea...

Figure 2.13. 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.


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 2.14. 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. You know, this kind of application contains Web resources. Thus, it's important to indicate the content directory. The wizard will put all those values itself, so you can leave everything as is.
Figure 2.15. 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.
In case 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.
Chapter 2. Creating a New Sea...

Figure 2.17. Using Custom JSF implementation Library

If a new library is required, click Manage libraries... ( ) button. Then you should click New.
Adding JSF Capabilities

Figure 2.18. Using Custom JSF implementation Library

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

Figure 2.19. Entering New User Library Name

To add proper Library JARs, click Add JARs... button and select the JARs on your hard drive.
Figure 2.20. 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.
Adding JSF Capabilities

Figure 2.21. 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.
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:
Configure Seam Facet Settings

**Figure 2.23. 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 (version 1.2 in our case).
Chapter 2. Creating a New Sea...

Figure 2.24. 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.

Tip:

If you want to name your web project "MyProject-war" note that 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".

- Select a Database Type
Figure 2.25. Database Type Selecting

- and then specify a Connection profile appropriate for your database.
Figure 2.26. Connection Profile Options

You can edit chosen profile by using *Edit* button or organize a new one by clicking on *New* button and selecting necessary for you type of connection profile.
Figure 2.27. Connection Profile Selecting

On the 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 2.28. 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 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.
Figure 2.29. Code Generation Section

Click on *Finish* to generate a project.
Maven Integration for Seam Projects

Starting from 3.1 M2 version JBoss Tools includes new Jboss Maven Integration plug-in which for now allows to create mavenized Seam projects and enable Seam tools support for imported Maven projects.

The Jboss Maven Integration requires m2eclipse [http://m2eclipse.sonatype.org/] to be installed.

Note:
For M2 release of Jboss Tools it is important to use m2eclipse stable updatesite [http://m2eclipse.sonatype.org/update/] or the latest integration build [http://m2eclipse.sonatype.org/releases/0.9.9.200906081351] that M2 was developed against. If you want to use the latest development release of m2eclipse you need to use the nightly builds [http://download.jboss.org/jbosstools/updates/nightly/trunk/] of JBoss Tools.

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 Seam 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).
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 Maven JSF 1.2 Libraries as 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:
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.
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

3.3. Relevant Resources Links

In the *Using Maven with JBoss Tools* [http://in.relation.to/Bloggers/UsingMavenWithJBossTools] article written by Max Andersen you’ll find references to the screencasts on how to use the examples of mavenized Seam projects provided by JBoss Tools.
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:

![Project Layout for WAR projects](image)

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 Chapter 15, Using TestNG project 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.
4.2. EAR Deployment

The project layout for EAR projects is:

![Project Layout for EAR projects](image)

**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 your 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.*

![Seam Runtime Settings](image)

**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

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 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 bringing the context menu and navigating 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

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

Table 5.1. Seam Actions in the New Submenu

<table>
<thead>
<tr>
<th>Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seam Web Project</td>
<td>Runs Section 2.1, “Create standalone Seam Web Project” wizard for creating a new Seam project</td>
</tr>
<tr>
<td>Seam Action</td>
<td>Runs Section 6.1, “New Seam Action” wizard for creating a new Seam action</td>
</tr>
<tr>
<td>Seam Form</td>
<td>Runs Section 6.2, “New Seam Form” wizard for creating a new Seam form</td>
</tr>
<tr>
<td>Seam Entity</td>
<td>Runs Section 6.4, “New Seam Entity” wizard for creating a new Seam entity</td>
</tr>
<tr>
<td>Seam Conversation</td>
<td>Runs Section 6.3, “New Seam Conversation” wizard for creating a new Seam conversation</td>
</tr>
<tr>
<td>Seam Generate Entities</td>
<td>Runs Chapter 7, Seam Generate Entities wizard</td>
</tr>
</tbody>
</table>

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/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 5.2. Find Seam References/Declarations actions

<table>
<thead>
<tr>
<th>Name</th>
<th>Function</th>
<th>Keyboard Shortcut</th>
</tr>
</thead>
<tbody>
<tr>
<td>Find Seam References</td>
<td>Find all references and declarations to the selected element. It's available for EL expressions in both .java</td>
<td>Ctrl+G</td>
</tr>
</tbody>
</table>
Chapter 5. Seam Menus and Actions

<table>
<thead>
<tr>
<th>Name</th>
<th>Function</th>
<th>Keyboard Shortcut</th>
</tr>
</thead>
<tbody>
<tr>
<td>Find Seam Declarations</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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:
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

<table>
<thead>
<tr>
<th>Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open Seam Components</td>
<td>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.</td>
</tr>
</tbody>
</table>
Enter a name in the text field and see the results as it shown on screenshot:

![Figure 5.4. Open Seam Components Dialog](image)

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

6.2. 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 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.

![Form Component](image)

**Figure 6.10. Form Component**

Form component was hot-deployed.

### 6.3. 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 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](image)

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.
Chapter 6. Seam Wizards

The Entity page is:

![Entity Page in JBoss Tools HTML Editor](image)

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

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

![Customer Page](image)

**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

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

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

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.

**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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="PageName.xhtml" /></td>
<td>A yellow box with a solid border represents a <code>&lt;page&gt;</code> element.</td>
</tr>
</tbody>
</table>
Pressing on the plus icon ( + ) on the <page> 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 <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 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**

<table>
<thead>
<tr>
<th>Icon Image</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Select a page element</td>
</tr>
<tr>
<td></td>
<td>Marquee a page item</td>
</tr>
<tr>
<td></td>
<td>Add a page relationship</td>
</tr>
<tr>
<td></td>
<td>Add an exception rule</td>
</tr>
</tbody>
</table>

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
Chapter 8. Seam Editors

- **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 → Seam → Editors → Seam Pages Diagram**.

![Preferences of Seam Pages Diagram](image)

**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.
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 8.5. Outline view: Tree mode
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.
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:
Chapter 8. Seam Editors

- 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
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](image)

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

Figure 8.12. Content assist for @Named beans
Chapter 8. Seam Editors

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](image)

OpenOn is available for the following files:

- **OpenOn in XHTML files**
- **OpenOn in XML files**
- **OpenOn in Java files**

**Note:**
In this section were described only use cases special for Seam applications. For general information please read OpenOn section in Visual Web Tools Reference Guide [http://download.jboss.org/jboss/tools/nightly-docs/en/jsf/html/editors.html#OpenOnSelection4HyperlinkNavigation).

**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 (e.g. XML file) to navigate to the file you clicked on.

![Figure 8.14. OpenOn in .page.xml file](image)

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.
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;
- inside `jbpm` tags;

---

**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;
- 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 declarated 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.
Chapter 8. Seam Editors

Figure 8.23. Validation Filters
Chapter 9.

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.

![Seam Components View](image)

**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.
The Seam Components View can show a component’s 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)
• 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 dependent on other project.
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.
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

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.

Note

There are references made between each EL and variable name, so you can revalidate a particular EL if a context variable was changed. This makes parsing of each resource with referenced ELs much faster.
Figure 10.4. Severity Preference for EL Syntax

The *Revalidate unresolved ELs automatically* option is enabled by default. If necessary, you can uncheck it. It may increase a performance for big complex projects.

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 10.5. 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.**
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.
Chapter 10. Seam Preferences

Figure 10.7. 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 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.

![Properties for Seam Project when Seam runtime is not selected](image)

**Figure 10.8. 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 10.9. 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 JBoss Tools → Add Custom Capabilities. Then check the needed modules and press Finish.
Figure 10.10. Custom Capabilities be added to Seam Project

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

Figure 10.11. Adding Custom Capabilities to Seam Project
Chapter 11.

Seam Refactoring Tools

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

Starting from Jboss Tools M4 refactoring functionality became also available for managed beans, method and properties used in ELs. Thus now Seam refactoring tools include the next 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 and use a standard Eclipse refactoring wizard. For details, see Section 11.1, “Using the Refactoring Wizard” later in this chapter.

Note:

Seam refactoring could be only performed in the file that is not marked as Read only in its properties.
11.1. Using the Refactoring 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:
Chapter 11. Seam Refactoring ...

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.

![Changes to be performed](image)

**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**.
Chapter 11. Seam Refactoring ...

Figure 11.5. Opening Refactoring Wizard in Java Editor

For details, see also the movie, “Renaming a Seam component in the Java Editor” [http://docs.jboss.org/tools/movies/demos/rename_seam_components_java/rename_seam_components_java.htm].

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.
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.

For details, see also the movie, *“Renaming a Seam component in the Seam Components view”* [http://docs.jboss.org/tools/movies/demos/rename_seam_components_comp_view/rename_seam_components_comp_view.htm].
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](image)
Figure 11.9. Opening Refactoring wizard in components.xml file
Figure 11.10. Opening Refactoring wizard in .html file

For details, see also a set of movies about renaming Seam context variables in EL in different locations:

- In the Java editor [http://docs.jboss.org/tools/movies/demos/rename_context_variables_java(rename_context_variables_java.htm)]

- In the JBoss Tools Properties Editor [http://docs.jboss.org/tools/movies/demos/rename_context_variables_properties/rename_context_variables_properties.htm]

- In the XML editor [http://docs.jboss.org/tools/movies/demos/rename_context_variables_xml/rename_context_variables_xml.htm]

- In the Visual Page Editor [http://docs.jboss.org/tools/movies/demos/rename_context_variables_vpe/rename_context_variables_vpe.htm]
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
Chapter 12. Adding Seam support...

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

Figure 12.2. Java EE Module Dependencies

Make sure there is `jboss-seam.jar` in application.xml as well:
Figure 12.3. Java EE Module Dependencies

Then include libs from EAR to EJB Manifest Class-Path (Properties for seamproject-ejb → Java EE Module Dependencies)
Figure 12.4. 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 12.5. 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
Chapter 12. Adding Seam support...

Figure 12.6. 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 12.7. 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 12.8. Added Seam Support to EAR Project
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/](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 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.
Figure 13.3. Seam Facet page

- On New Connection Profile dialog select the HSQLDB connection profile type, name it cruddb 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.
• 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 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 switching to the next tab and press Add JAR/Zip button.
Figure 13.7. Driver Definition

- On the Properties tab set the Connection URL, Database Name and User ID 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

- 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 13.13. login.xhtml in VPE

• Switch to Database Development perspective with **Window → Open Perspective → Other...** and connect to the *cruddb* database.
Chapter 13. Generate a CRUD D...

Figure 13.14. Connecting to the CRUDDB database

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

Figure 13.15. 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*.

![Generate Seam Entities](image)

**Figure 13.16. Generate Seam Entities**

- Click *Refresh* to display all the tables from the database.
Chapter 13. Generate a CRUD D...

Figure 13.17. Dialog for Selecting Tables

- Now you can filter the tables. Use the *Include* button to include only necessary ones.
Figure 13.18. Selecting Tables

- Under *WebContent* folder you can find all generated xhtml files:
Chapter 13. Generate a CRUD D...

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.

![Image of Customers.java file in the java editor]

**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.
Chapter 14.

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 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.

![Employees List Page](image)

**Figure 14.4. Employees List Page**

Use Employees search parameters fields to filter the selected list.

Press Select opposite one of employees.
Figure 14.5. Employee details Page

Press Edit to edit employee fields.

Enter Login and Password to login. (Use "crudapp-user"/"secret" for example)
Figure 14.6. Login page
Figure 14.7. 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.

15.1. 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)

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

15.3. 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 15.1. Seam-test Project](image)

- Add Seam Action to your project via File → New → Seam Action.
How to use the generated Seam-test project to run Seam tests?

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

* 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.
Chapter 15. Using TestNG project

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

Figure 15.8. Test Results in Browser

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

Figure 15.9. Test Information

Select a result on the left-hand pane and its details will be displayed on the right-hand one.
Chapter 15. Using TestNG project

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.

In conclusion, the main goal of this document is to get you know with a full feature set 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 the article Overview of the improvements required by JBossTools/JBDS Docs users [http://wiki.jboss.org/wiki/JBossToolsDocsFuture].

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