JBoss Server Manager Reference Guide

Version: 3.3.0.M5

Quick Start with JBoss Server

This chapter covers the basics of working with the JBoss Server.

1.1. Key Features of JBoss Server

The table below lists the main features included in JBoss Server:

Table 1.1. Key Functionality for JBoss AS and Archive Tools

Feature	Benefit	Chapter
JBoss AS plugin	Using WTP the JBoss AS plugin allows you to work with the server in run or debug mode. You can easily install runtimes and servers, copy the existing runtime configuration or configure it up to your needs.	*Runtimes and servers*
JBoss AS Perspective	It provides easy management of an installed JBoss Server and includes the standard Console and Properties views and the specially added Project archives and Servers views.	*JBoss AS perspective*
Modules Deployment	Provides a number of ways, using WTP and JBoss Tools, to deploy either a project or a single file on the server.	*Deploying modules*

If you already have installed the JBoss server and runtime this document will show you how to configure, start and stop the server, as well as the deployment and archiving processes. Installing runtimes and servers is covered in more detail in *Chapter 2, Runtimes and Servers in the JBoss AS plugin*.

To start working with JBoss AS, select the **JBoss AS** Perspective via **Window** → **Show View** → **Other** → **Server** → **Servers**.

1.2. Starting JBoss Server

Starting JBoss Server™ is quite simple. You can control the server behavior with the help of a special toolbar in the **Servers** view where you can **start** it in a regular or debug mode, **stop** or **restart** it and **publish** to the server.

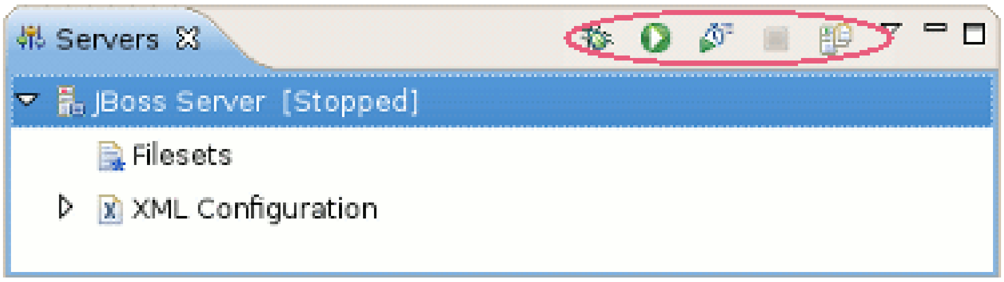


Figure 1.1. Servers Toolbar

To launch the server click the green-with-white-arrow icon on the **Servers** view or right click server name in this view and select **Start**. If this view is not open, select **Window** → **Show View** → **Other** → **Server** → **Servers**.

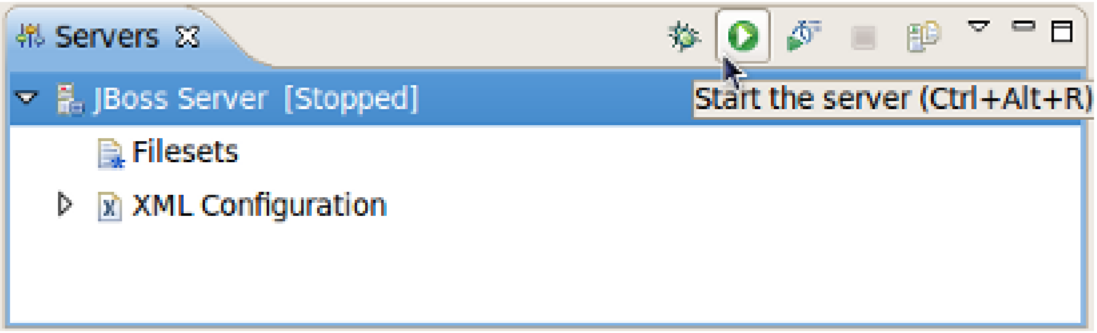


Figure 1.2. Start JBoss Server

1.3. Stopping JBoss Server

To stop the server, click the **Stop** icon in the **Servers** view or right click the server name and select the **Stop** option.

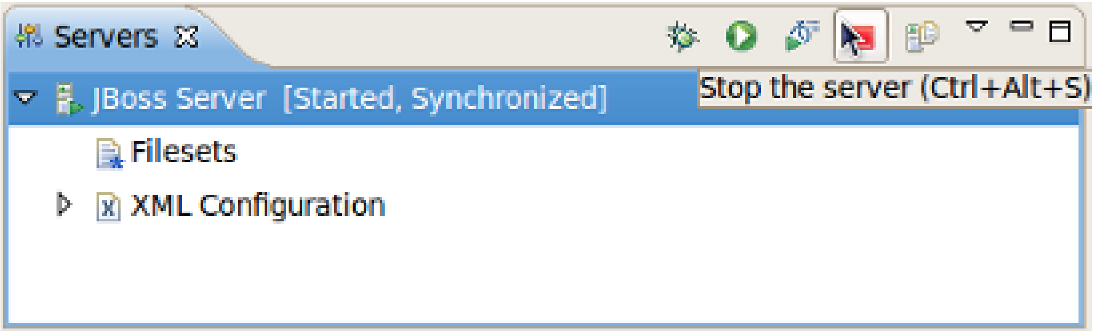

Servers
JBoss Server [Started, Synchronized]
Stop the server (Ctrl+Alt+S)
Filesets
XML Configuration

Figure 1.3. Stop JBoss Server

When the server is stopped you will see **Stopped** state next to its name in the square brackets.

Learn more about the **Servers** view in *Section 3.1, “The Servers view”*.

1.4. Project Archiving

JBoss Tools comes with our own archives tool. The Project Archives plugin consists primarily of a view to set up each packaging configuration (**Window** → **Show View** → **Other** → **JBoss Tool** → **Project archives**).

Right clicking in the **Project archives** view you can create a War, EJB War, EAR or JAR archive.

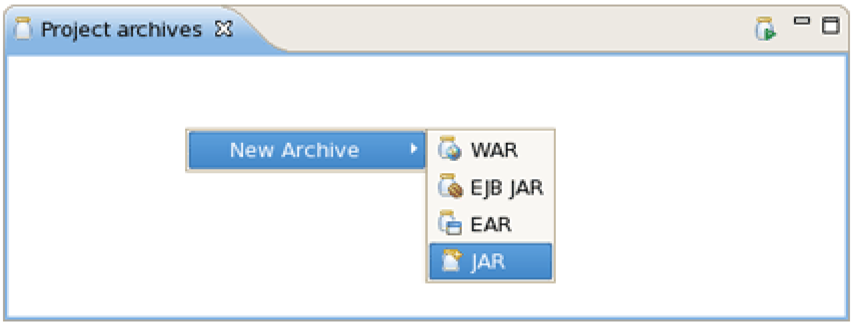


Figure 1.4. Archive Creating

Using the context menu on the item you can initiate a *full build* on an archive, *edit*,*delete* or *publish* it.

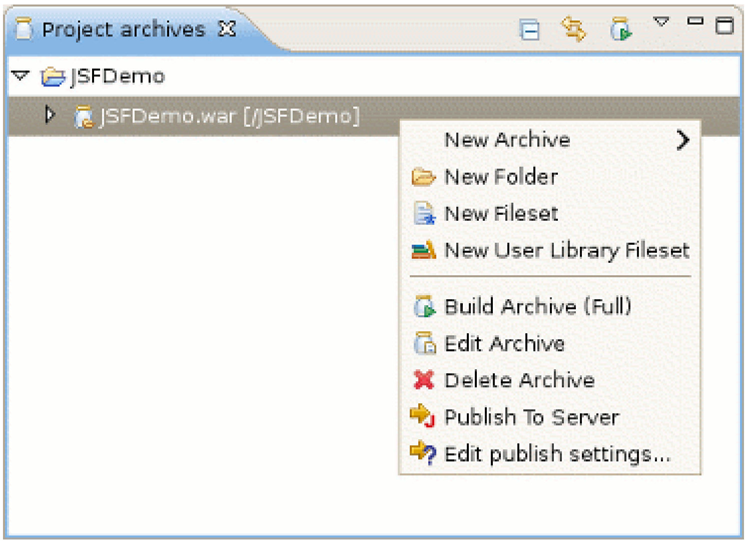
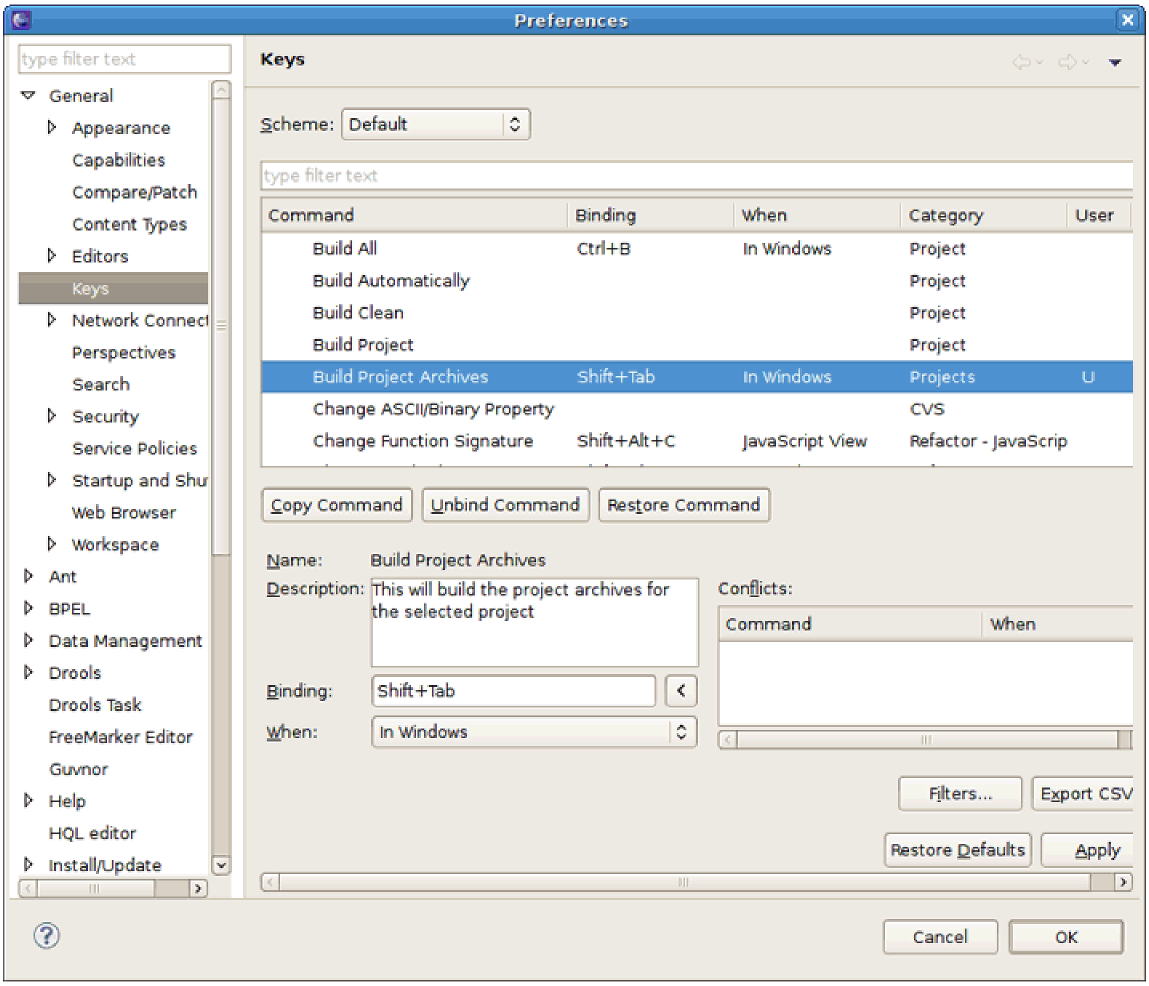


Figure 1.5. Context Menu on the Item

If you wish, you can assign a hotkey for the Build Project Archive action. A key binding can be added by selecting **Window** → **Preferences** → **General** → **Keys**.



Scheme: Default

Command	Binding	When	Category	User
Build All	Ctrl+B	In Windows	Project	
Build Automatically			Project	
Build Clean			Project	
Build Project			Project	
Build Project Archives	Shift+Tab	In Windows	Projects	U
Change ASCII/Binary Property			CVS	
Change Function Signature	Shift+Alt+C	JavaScript View	Refactor - JavaScrip	

Name: Build Project Archives

Description: This will build the project archives for the selected project

Binding: Shift+Tab

When: In Windows

Conflicts:

Command	When

Figure 1.6. Setting the Hotkey for Build Project Archive Action

To build a project archive, select the project in the **Package Explorer** view and execute the hotkey combination you assigned for this action.

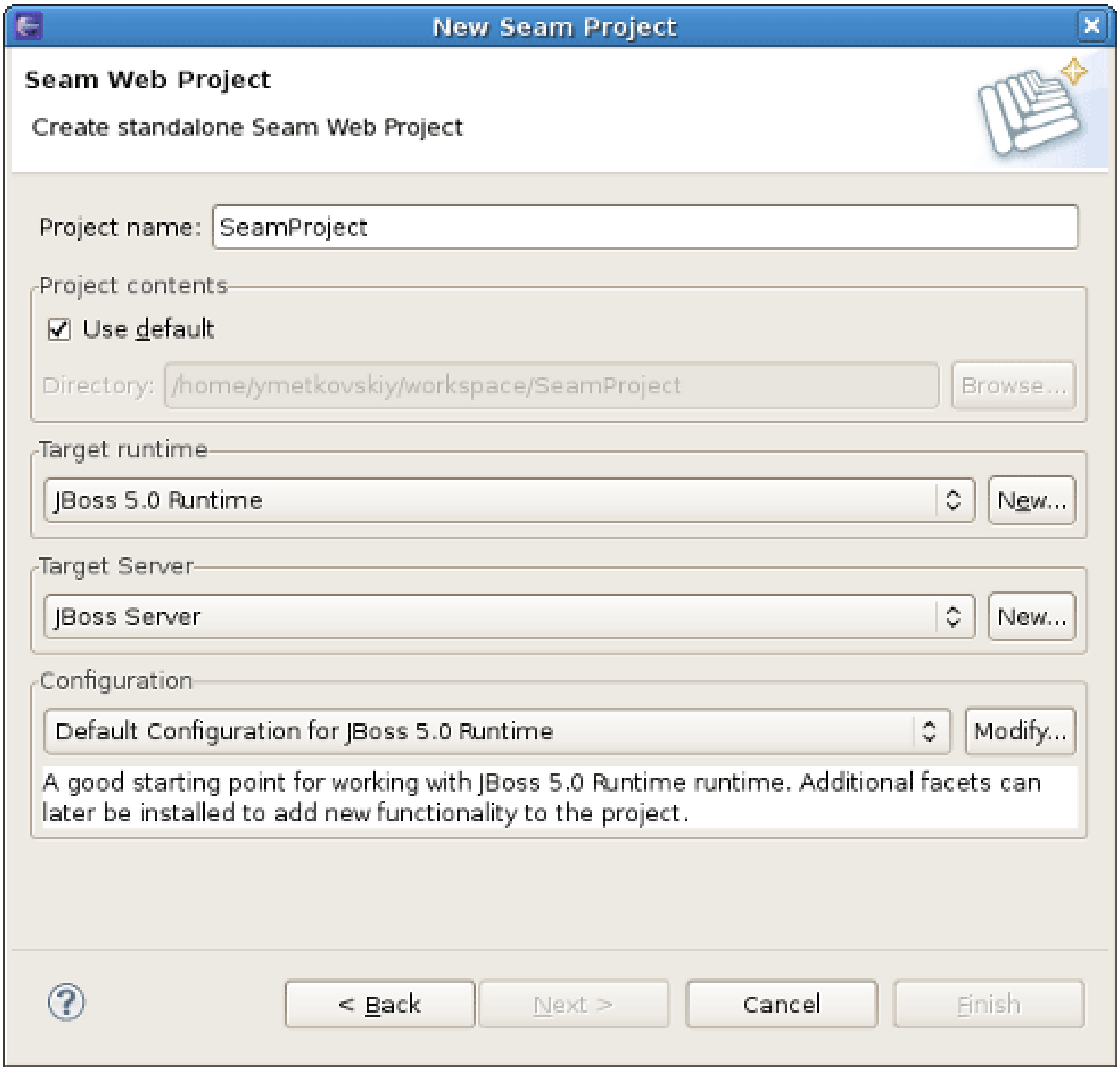
More information about the **Project Archives** view can be found in *Section 3.2, “Project Archives View”*.

1.5. Deploying an Application to a Server

There are two times to deploy your application:

- While creating it
- After it already exists

When you create a new project (Seam, JSF or Struts) with the New Project or Import Project wizards, they will include the **Target Runtime** and **Target Server** sections. You can deploy the application through the appropriate configuration in these sections.



Project name: SeamProject

☑ Use default

Directory: /home/ymetkovskiy/workspace/SeamProject

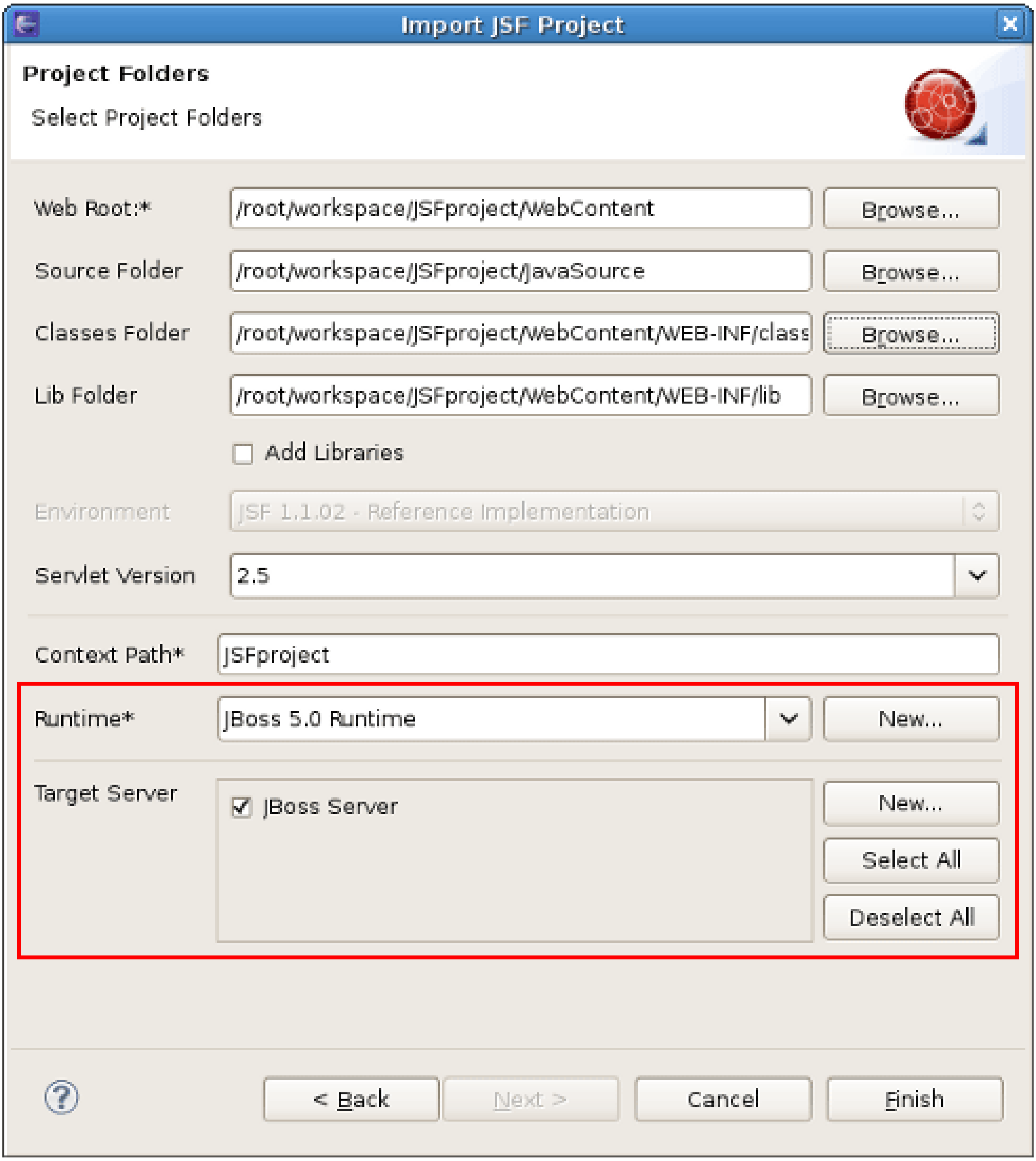
Target runtime: JBoss 5.0 Runtime

Target Server: JBoss Server

Configuration: Default Configuration for JBoss 5.0 Runtime

A good starting point for working with JBoss 5.0 Runtime runtime. Additional facets can later be installed to add new functionality to the project.

Figure 1.7. Runtime and Server Sections in the New Project Wizard



Web Root:* /root/workspace/JSFproject/WebContent

Source Folder /root/workspace/JSFproject/JavaSource

Classes Folder /root/workspace/JSFproject/WebContent/WEB-INF/class

Lib Folder /root/workspace/JSFproject/WebContent/WEB-INF/lib

□ Add Libraries

Environment JSF 1.1.02 - Reference Implementation

Servlet Version 2.5

Context Path* JSFproject

Runtime* JBoss 5.0 Runtime

Target Server ☑ JBoss Server

Figure 1.8. Runtime and Server Sections in the Import Project Wizard

You can deploy an existing application to a server by right-clicking the target defined server in the **Servers** view and then selecting **Add and Remove Projects** from the context menu.

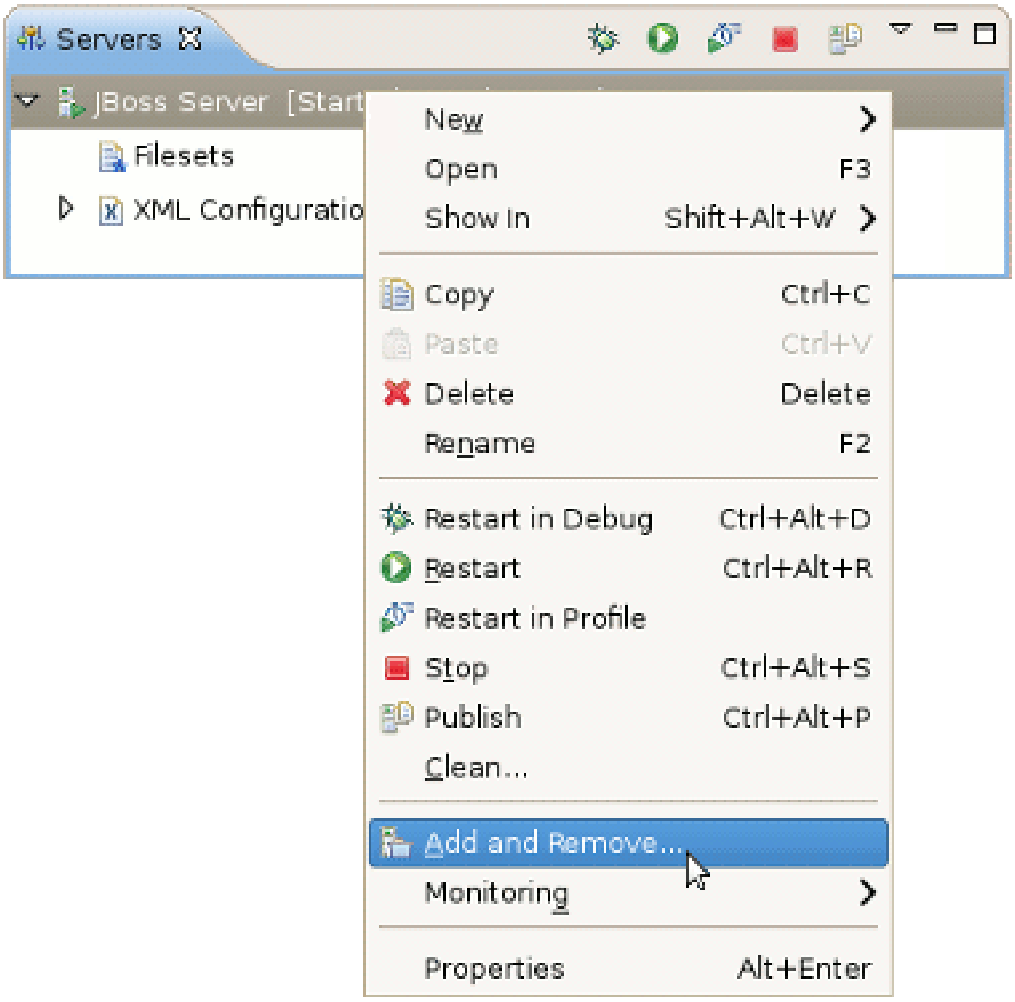
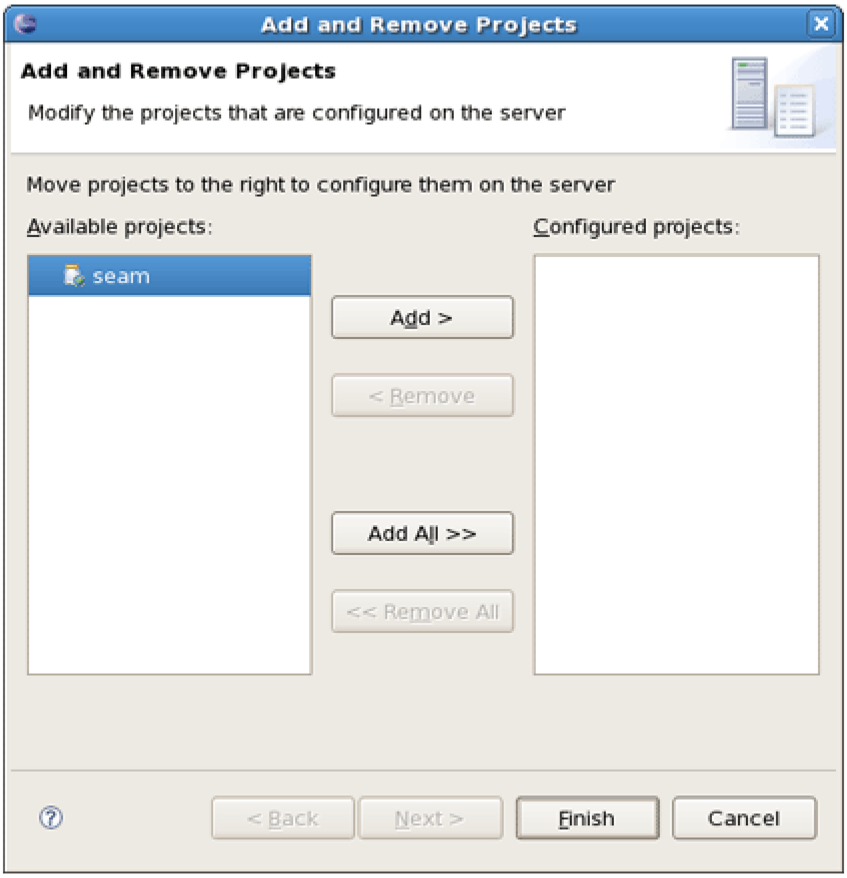


Figure 1.9. Add and Remove Projects From the Context Menu.

If this application is not assigned to a server, it will be in the left-hand available projects list. Clicking on the **Add >** button will add it to the right-hand configured projects list and deploy the application to this server.



Modify the projects that are configured on the server

Move projects to the right to configure them on the server

Figure 1.10. Modifying The Projects that are Configured on the Server

> **Note**
>
> It is now possible to deploy OSGI (Open Services Gateway initiative framework) projects to the JBoss Enterprise Application Platform 6 or JBoss Application Server 7.

1.6. Publishing to JBoss Server

The publishing of all the modules added to a Server is performed automatically when starting a Server.

Automatically publishing changes made to the workspace is enabled by default, allowing the workspace to remain in sync with the publish folder. If you need to control when to publish the changes, just disable the automatic publish in the Server Editor (see *Section 3.1.5, “Server Editor”*) and use the **Publish to Server** ( ) button which will incrementally publish the workspace.

This section has provided some basic information that will allow you to use the common features provided by the JBoss server. However, JBoss server includes a great deal more functionality, which will be discussed in subsequent chapters.

1.7. Other relevant resources on the topic

All JBoss Developer Studio/JBoss Tools documentation can be found on the *on JBoss Tools release documentation page* [http://docs.jboss.org/tools/2.1.0.GA].

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

Find answers to frequently asked questions about JBoss AS 5 usage in the *JBossAS5FAQ* [http://community.jboss.org/wiki/JBossAS5FAQ].

Information on how to monitor a remote JBoss Server from Eclipse can be found in this *wiki article* [http://www.jboss.org/community/wiki/MonitorARemoteJbossServerFromEclipse].

Runtimes and Servers in the JBoss AS plugin

In this chapter we will discuss how to install runtimes and servers.

First of all it is necessary to mention that the JBoss AS plugin makes use of WTP. This includes starting and stopping servers in run or debug mode. It also includes targeting WTP projects, such as Dynamic Web Projects, to certain server runtimes in order to ensure that the proper JARs from a specific server are added to the project's classpath properly.

In order to get started creating, running and debugging J2EE applications, we should create our runtime and server instances.

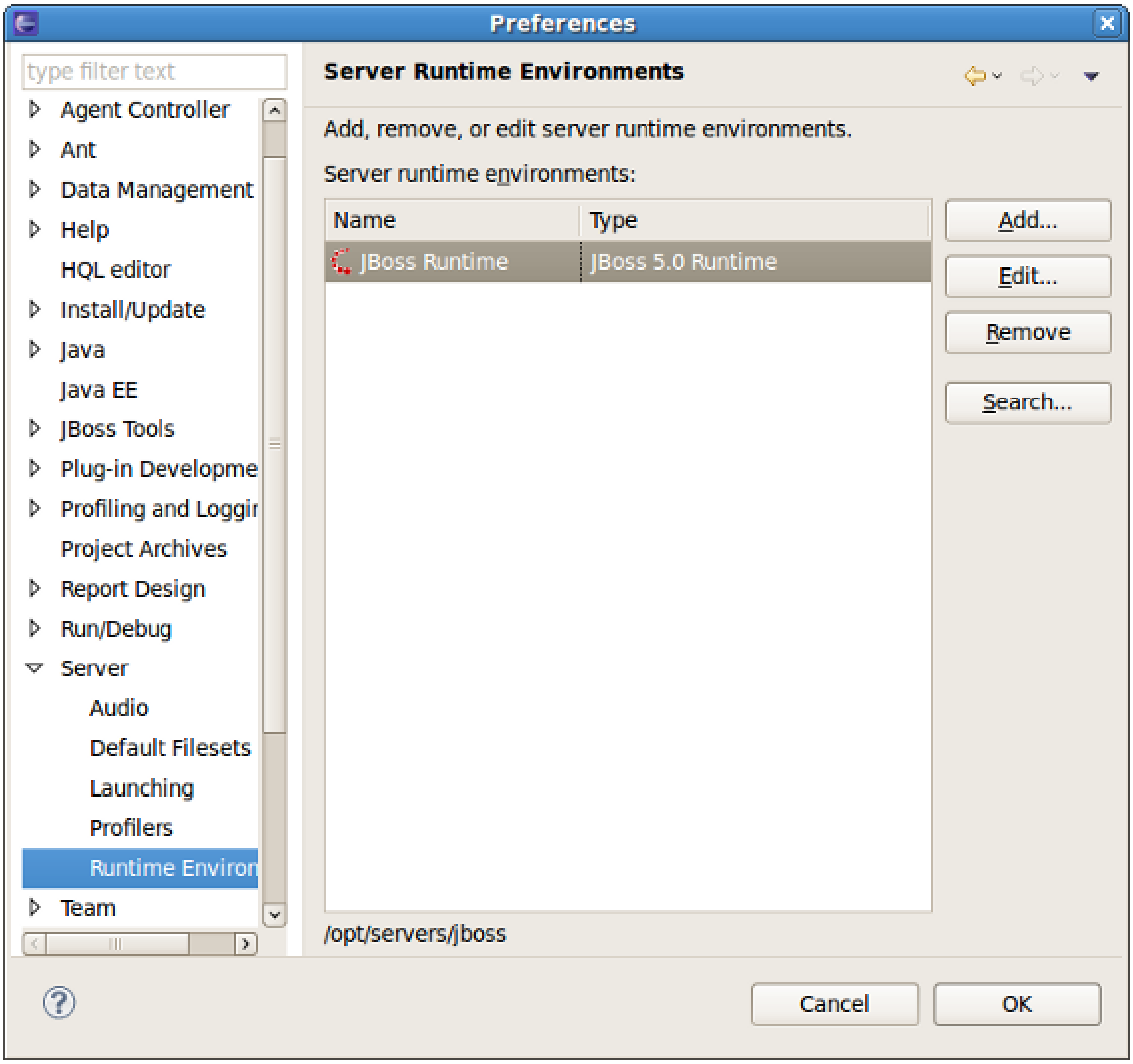
2.1. Runtimes

In JBoss Tools, the main purpose of Server Runtimes is to point to a server installation somewhere on disk. In our case, this will be a JBoss installation. It can then be used for two primary purposes:

- Providing classpath additions to WTP projects that require them.
- For JBoss server at least, it provides the necessary information to allow the server to be started and stopped and provides information on which JARs to run and which configuration to use.

2.1.1. Installing a new runtime

You can install runtimes into Eclipse by selecting **Window** → **Preferences** menu and then selecting **Server** → **Runtime Environments** from the categories available on the left.



Server Runtime Environments

Add, remove, or edit server runtime environments.

Server runtime environments:

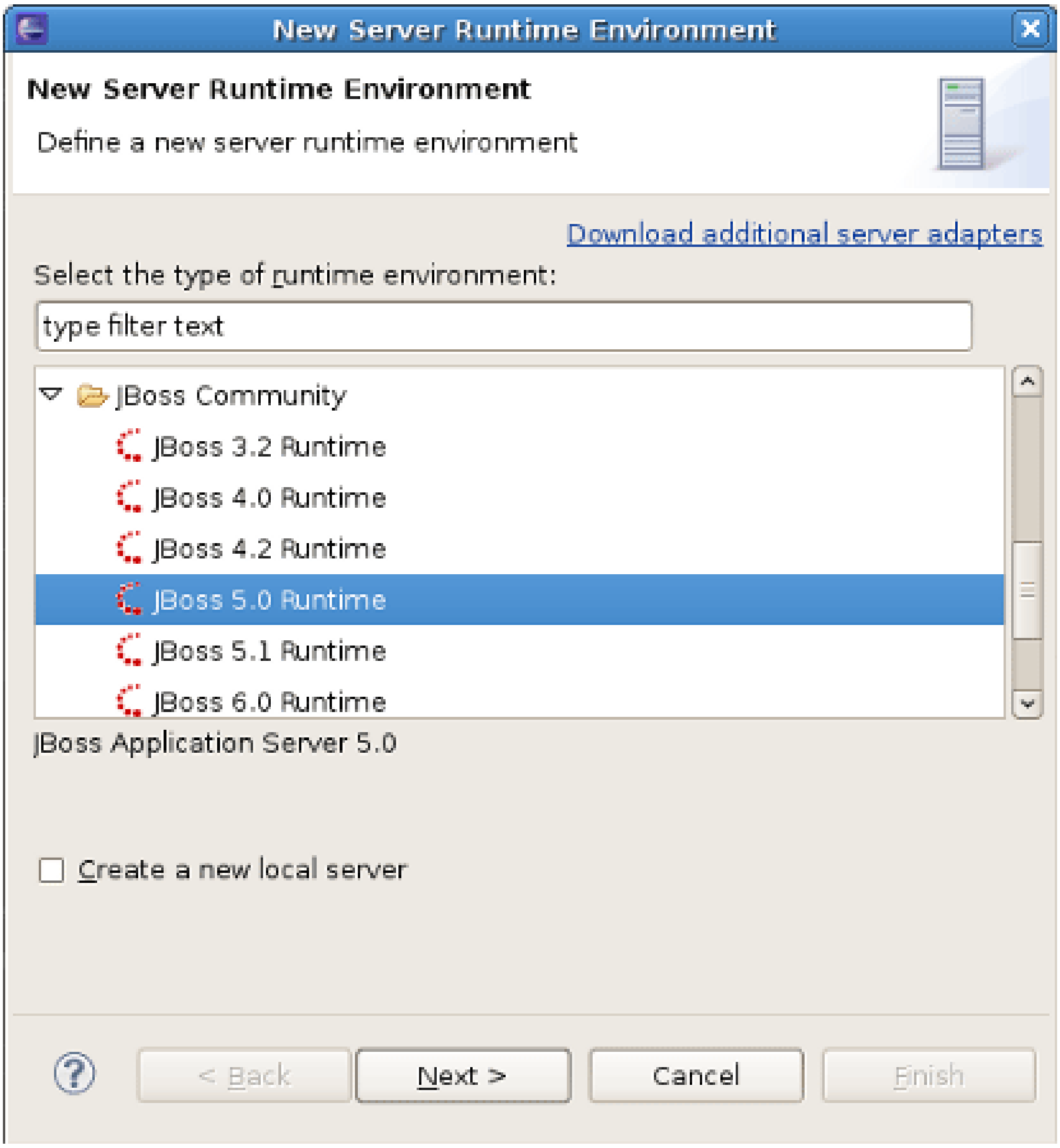
Name	Type
JBoss Runtime	JBoss 5.0 Runtime

/opt/servers/jboss

Figure 2.1. Installed Runtimes

From this preference page you can see all declared runtimes along with their types. Here, it is possible to edit or remove existing runtimes, as well as add a new one.

To create a JBoss runtime click the **Add** button and choose the appropriate type of runtime from the **JBoss Community** category.



New Server Runtime Environment

New Server Runtime Environment

Define a new server runtime environment

Download additional server adapters

Select the type of runtime environment:

type filter text

- JBoss Community
  - JBoss 3.2 Runtime
  - JBoss 4.0 Runtime
  - JBoss 4.2 Runtime
  - JBoss 5.0 Runtime
  - JBoss 5.1 Runtime
  - JBoss 6.0 Runtime

JBoss Application Server 5.0

☐ Create a new local server

Figure 2.2. Adding a Runtime

Note:

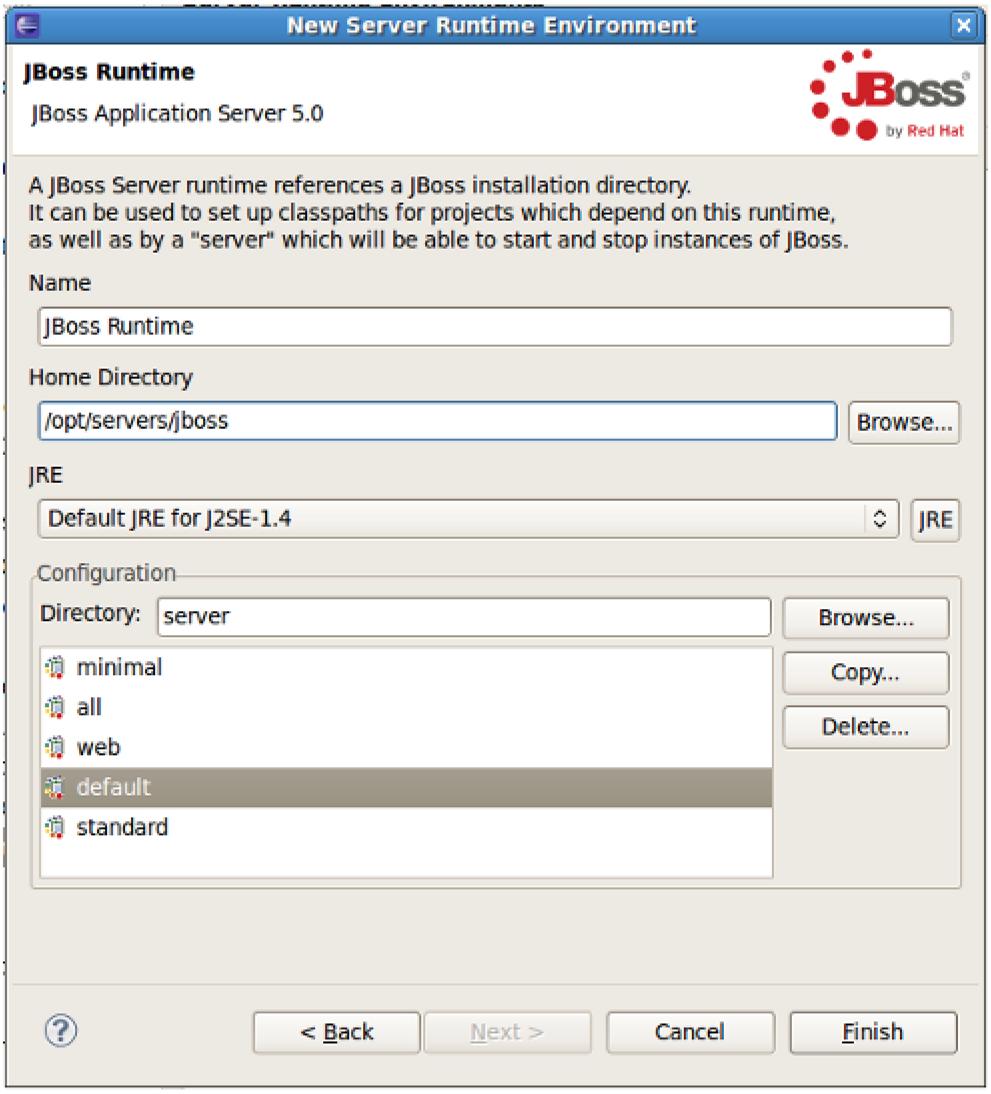
Now there is a separation between .org servers (the **JBoss Community** category) and product server that comes with JBoss EAP in JBDS (the **JBoss Enterprise Middleware** category).

As you can see, JBoss Tools™ provide its own adapters such as JBoss 3.2, 4.0, 4.2 and 5.0 as well. The last one comes a new safer incremental deployment feature, which prevents partial deployments to be picked up by the server. It means that scanning for auto-deployment is suspended while files are being copied to the deployment location and resumed when the copy is completed.

Note:

Currently we recommend you to use a fully supported JBoss 5.0 server adapter.

You will also note a Deploy-Only Runtime type. This type does not provide a classpath for WTP projects. It is used solely by it's server type for the purpose of setting up a deploy directory for users who do not wish to make use of starting, stopping, or debugging their projects inside Eclipse.



New Server Runtime Environment

JBoss Runtime

JBoss Application Server 5.0

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

Name: JBoss Runtime

Home Directory: /opt/servers/jboss

JRE: Default JRE for J2SE-1.4

Configuration

Directory: server

- minimal
- all
- web
- default
- standard

Figure 2.3. Adding a JBoss 5.0 Runtime

The following table describes all the available options of the current wizard page.

Table 2.1. Server Runtime Wizard Parameters

Name	Description
Name	The name of a new Runtime for a chosen server. We suggest that you do not leave the default value for this field. It is better to give descriptive names that will help to distinguish one runtime from another.
Home directory	The path to a directory where the runtime is installed.
JRE	The proper Java Runtime Environment. Because of the open-source nature of JBoss, a user is likely to want to modify and repackage some of the configuration-specific JBoss JARs and create their own configuration. So rather than forcing you to copy your entire JBoss installation, the structure of the wizard allows to create a new configuration instead.
Directory	The path to a directory where the configurations are installed.
Configuration	The list of configurations (all, default, minimal), which is updated as soon as you browse to a valid runtime installation folder.

As a result of having each runtime represent a specific configuration rather than the server installation as a whole, it is very likely you will create several different runtimes to test each of your configurations. So it becomes important to ensure your runtimes and later your servers, are given descriptive names that help you to remember which is which.

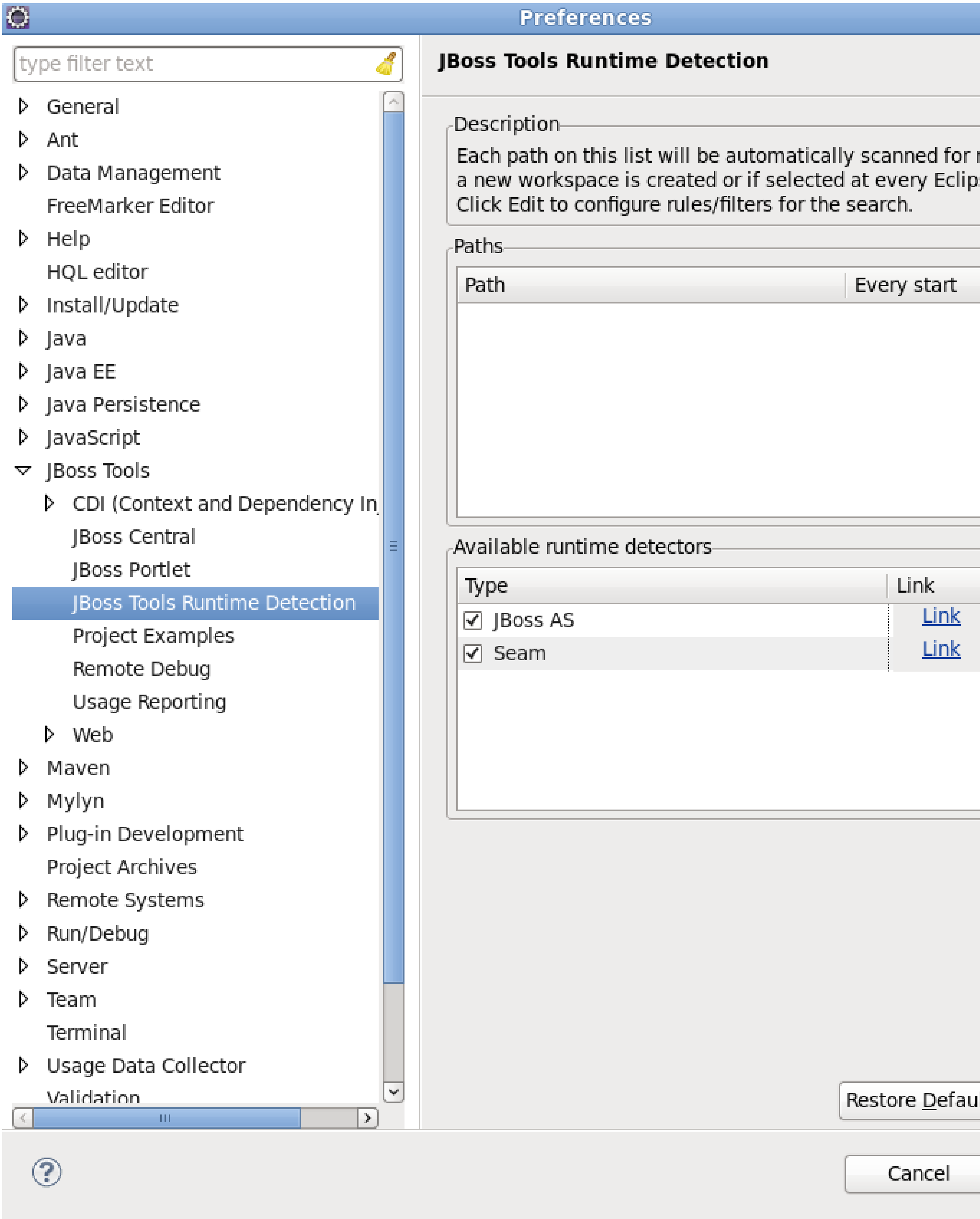
Click the **Finish** button to see your new runtime in the list.

Note:

If you edit the configuration of a runtime, the changes don't affect the settings of the servers that currently use the runtime. To apply the changes to them also, you should double left click on the server, select **Overview** → **Runtime Environment**, make sure that necessary configuration is chosen, click the **Finish** button and then **Save** button.

2.1.2. Detecting an existing runtime

JBoss Tools features the ability to search, detect and add existing JBoss server runtimes installed on your system. If you don't have an existing runtime you can download one through the **Download** option or *Section 2.1.1, “Installing a new runtime”* will guide you through the creation process. To begin searching for your existing JBoss runtime select **Window** → **Preferences** → **JBoss Tools** → **JBoss Runtimes**.



JBoss Tools Runtime Detection

Description

Each path on this list will be automatically scanned for r
a new workspace is created or if selected at every Eclip
Click Edit to configure rules/filters for the search.

Paths

Path	Every start

Available runtime detectors

Type	Link
☑ JBoss AS	Link
☑ Seam	Link

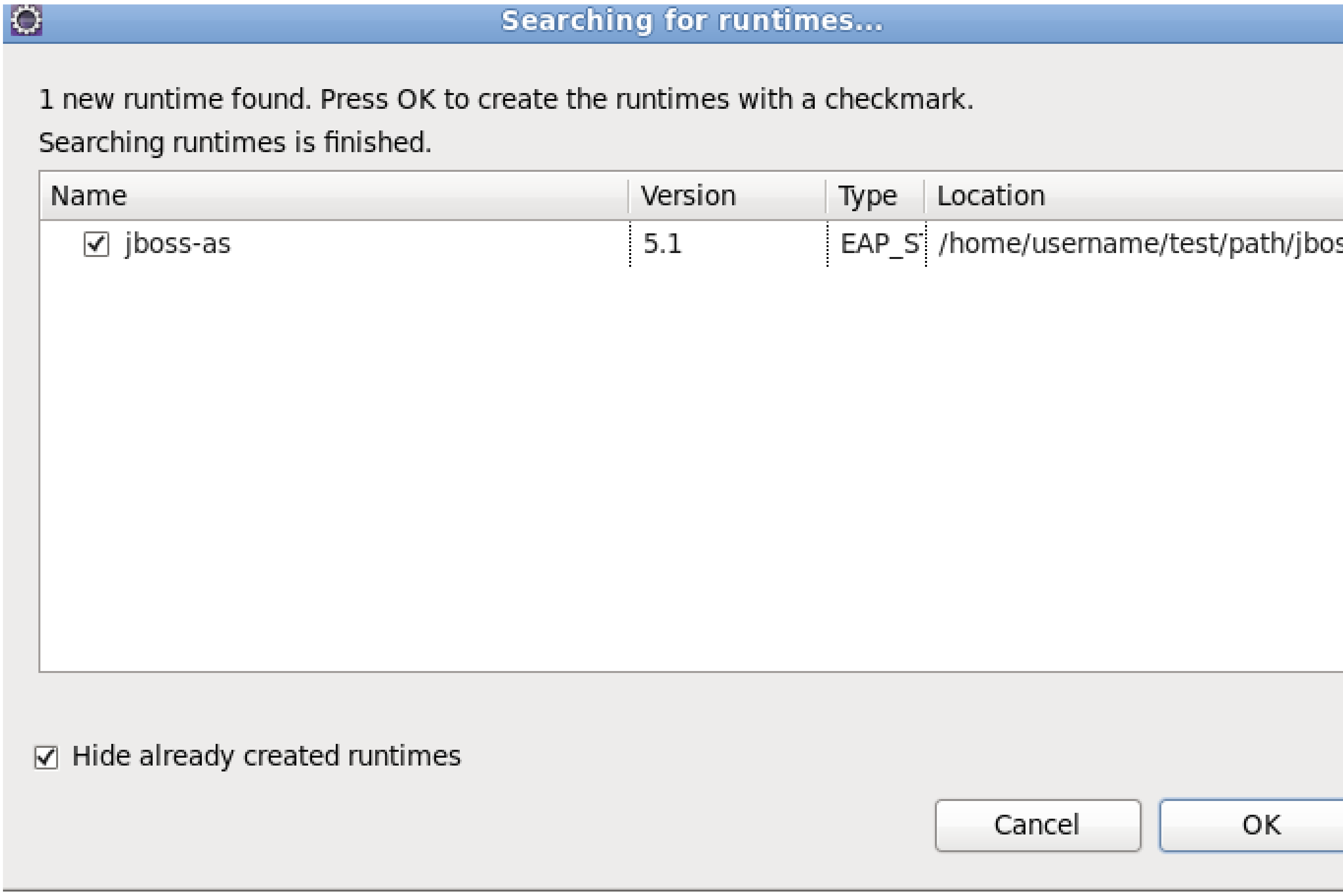
Figure 2.4. Preference page for JBoss Runtimes

The JBoss Tools Runtimes preference page is split into two different sections. One section defines **Paths** to be searched for installed server runtimes, the other section defines the runtime detectors available when the paths in the previous section are checked.

The **Add** button in the **Paths** section opens a file system browser window. Select the directory you wish to have recursively searched for JBoss runtimes. The directory will be searched and all found servers will be displayed as a list in the **Searching for runtimes** dialog. From the returned list, choose the runtimes you wish to make available by clicking the box beside each runtime and clicking the **OK** button.

Note

Server runtime detection now recognizes the ESB runtime distributed as part of the JBoss Service-Oriented Architecture Platform.



Searching for runtimes...

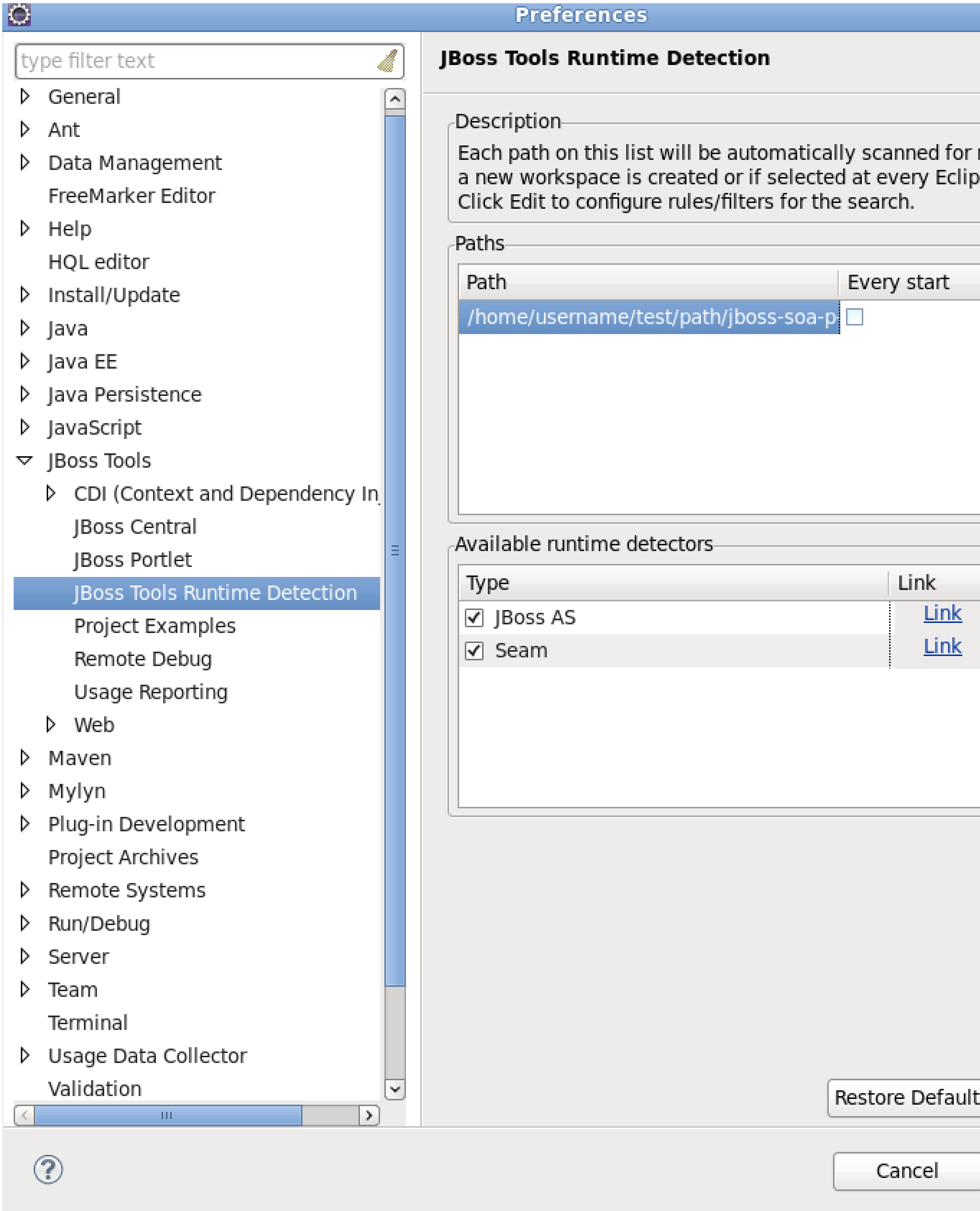
1 new runtime found. Press OK to create the runtimes with a checkmark.
Searching runtimes is finished.

Name	Version	Type	Location
☑ jboss-as	5.1	EAP_S	/home/username/test/path/jbos

☑ Hide already created runtimes

Figure 2.5. JBoss Runtime search results

The path you searched is now added to a list in the **JBoss Tools Runtime Detection** dialog **Paths** section. All the paths in this section will be automatically searched when a new workspace is created. If you wish for a path to be searched on startup then check the checkbox in the **Every start** column associated with it.



JBoss Tools Runtime Detection

Description

Each path on this list will be automatically scanned for r
a new workspace is created or if selected at every Eclip
Click Edit to configure rules/filters for the search.

Paths

Path	Every start
/home/username/test/path/jboss-soa-p	□

Available runtime detectors

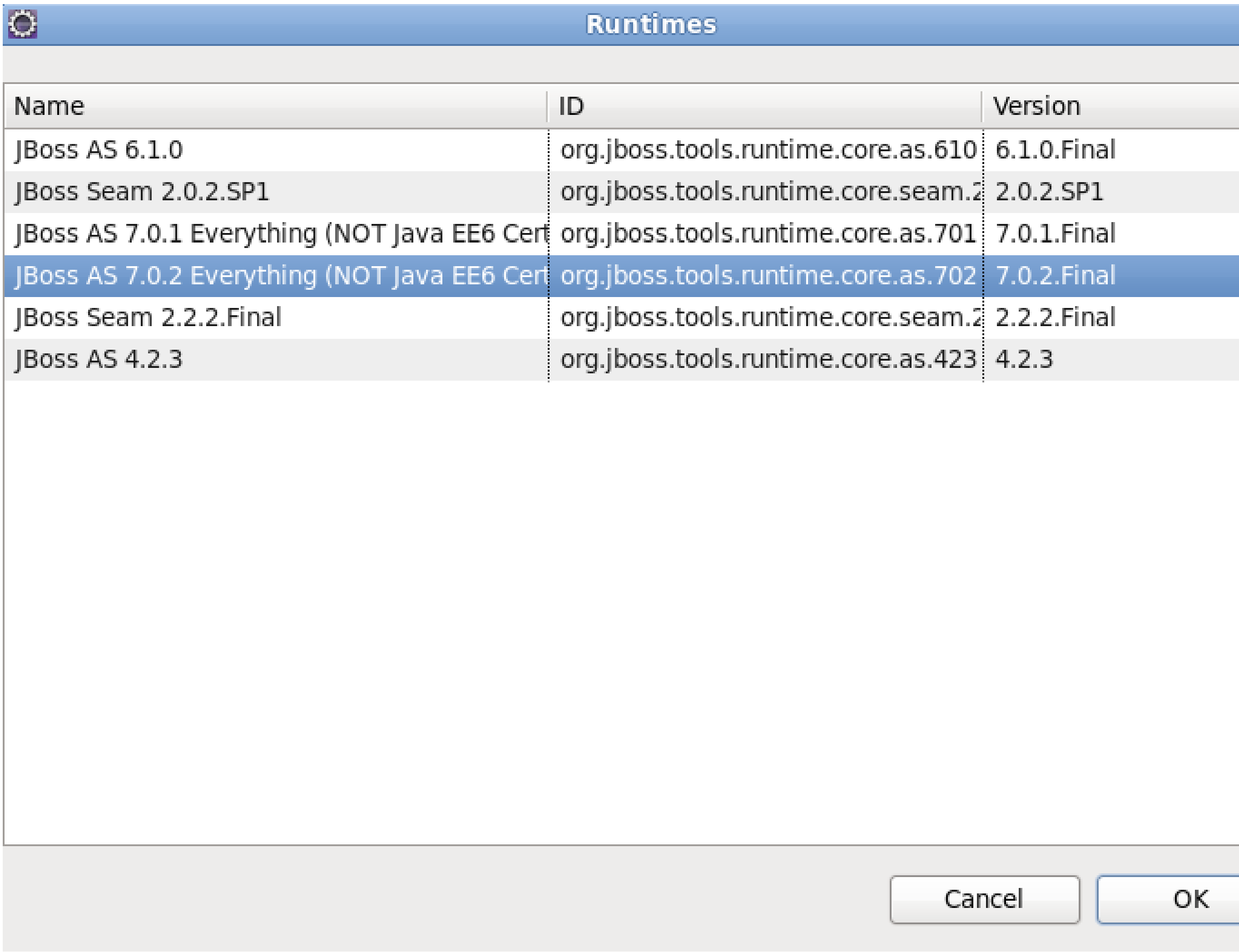
Type	Link
☑ JBoss AS	Link
☑ Seam	Link

Figure 2.6. JBoss Runtime search results

If you don't have a runtime already downloaded, you can download a free community application server through the **Download** button.

> **Important**
>
> No official support is available for community application servers (this includes enterprise customers using JBoss Developer Studio).

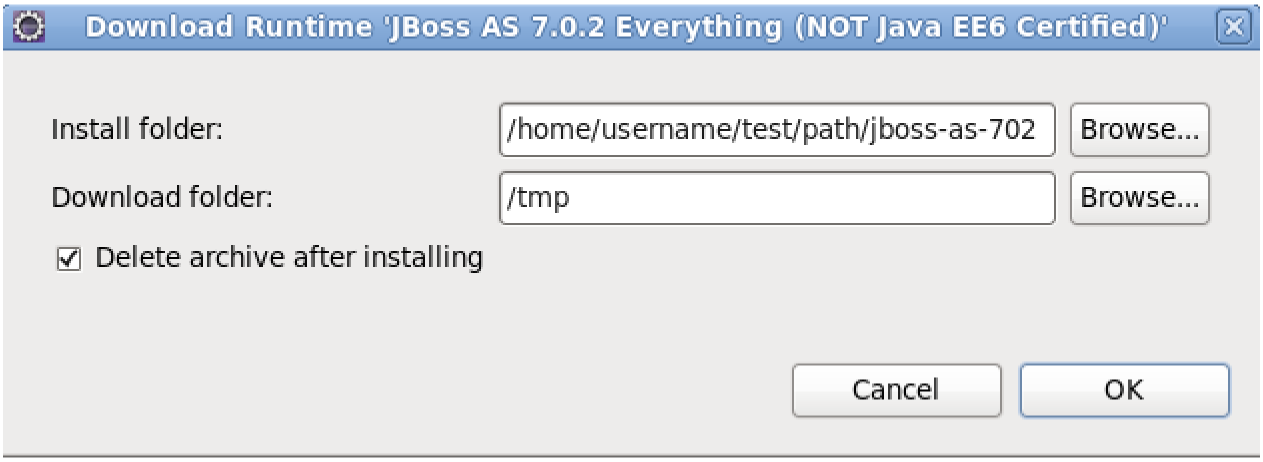
Clicking on the **Download** button will display a new screen of available runtimes that can be downloaded. Highlight the server you wish to download and install, and click the **OK** button.



Name	ID	Version
JBoss AS 6.1.0	org.jboss.tools.runtime.core.as.610	6.1.0.Final
JBoss Seam 2.0.2.SP1	org.jboss.tools.runtime.core.seam.2	2.0.2.SP1
JBoss AS 7.0.1 Everything (NOT Java EE6 Cert	org.jboss.tools.runtime.core.as.701	7.0.1.Final
JBoss AS 7.0.2 Everything (NOT Java EE6 Cert	org.jboss.tools.runtime.core.as.702	7.0.2.Final
JBoss Seam 2.2.2.Final	org.jboss.tools.runtime.core.seam.2	2.2.2.Final
JBoss AS 4.2.3	org.jboss.tools.runtime.core.as.423	4.2.3

Figure 2.7. JBoss Runtime search results

A new dialog will appear asking you to specify an **Install folder** and **Download folder**; the option to **Delete archive after installing** is checked by default. Once you have specified the two paths above, click the **OK** button and the server will begin downloading.



Download Runtime 'JBoss AS 7.0.2 Everything (NOT Java EE6 Certified)'

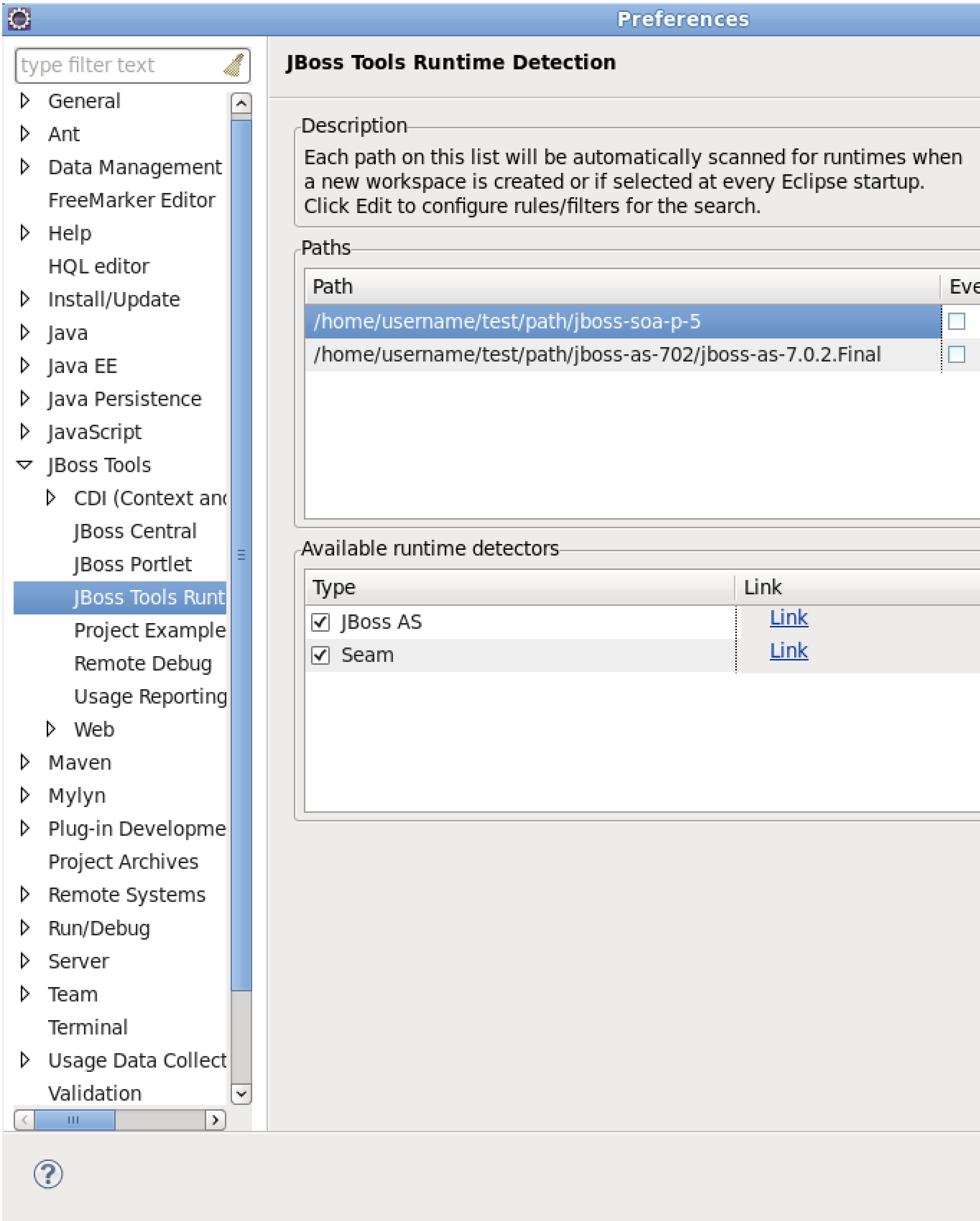
Install folder: /home/username/test/path/jboss-as-702

Download folder: /tmp

☑ Delete archive after installing

Figure 2.8. JBoss Runtime search results

Once the server has been downloaded and installed, you will notice that the path to the new server now appears in the **Paths** section of the **JBoss Tools Runtime Detection** dialog.



JBoss Tools Runtime Detection

Description

Each path on this list will be automatically scanned for runtimes when a new workspace is created or if selected at every Eclipse startup. Click Edit to configure rules/filters for the search.

Paths

Path	Eve
/home/username/test/path/jboss-soa-p-5	□
/home/username/test/path/jboss-as-702/jboss-as-7.0.2.Final	□

Available runtime detectors

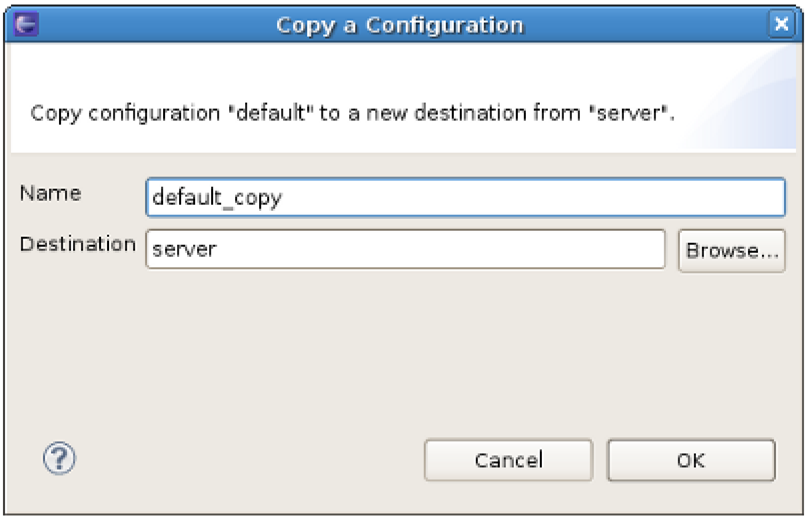
Type	Link
☑ JBoss AS	Link
☑ Seam	Link

Figure 2.9. JBoss Runtime search results

2.1.3. Duplicating a runtime configuration

While installing a new runtime you can copy the configuration from the existing one. To do this you should perform all the steps in *Section 2.1.1, “Installing a new runtime”*, with the exception of clicking the **Finish** button in the New Server Runtime Environment menu.

Make sure that you browse to a valid runtime folder and can see the list of configurations (all, default, minimal) in the Configuration section. Then choose appropriate Configuration from the list and click the **Copy** button.The next dialog should appear.



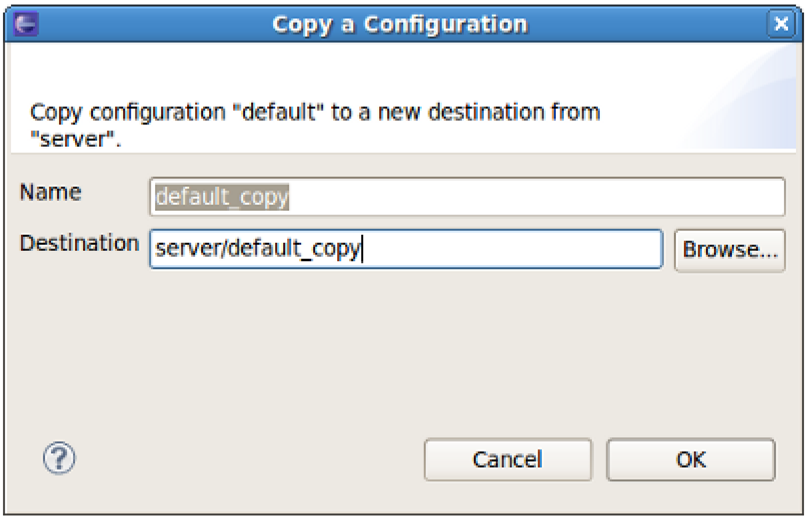
Copy configuration "default" to a new destination from "server".

Name: default_copy

Destination: server

Figure 2.10. Copy the existing configuration

Change the name, click the **Browse** button and select your configuration location or leave as it is if you want it to be located together with other runtime configurations.



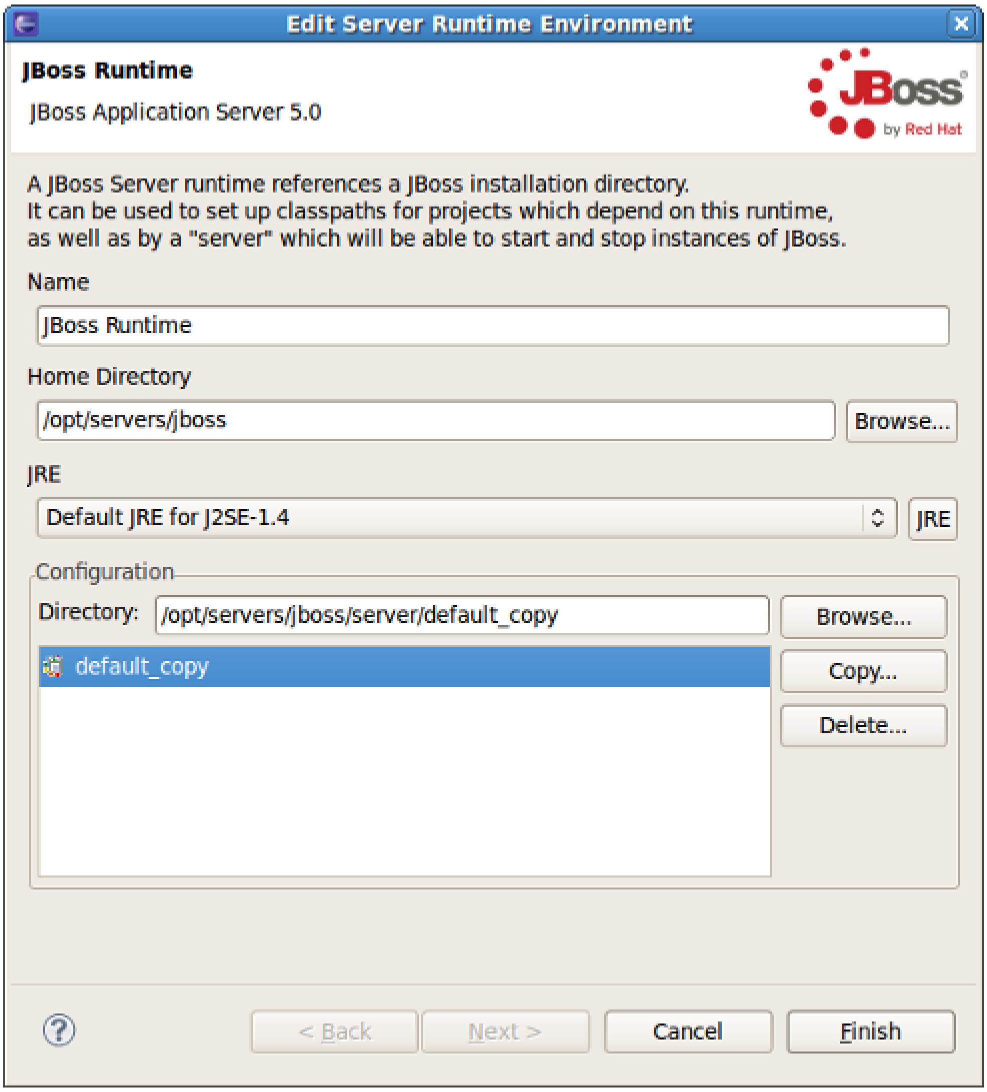
Copy configuration "default" to a new destination from "server".

Name: default_copy

Destination: server/default_copy

Figure 2.11. Copy the existing configuration

Click the **OK** button and you should see the next wizard with the newly copied configuration.



Edit Server Runtime Environment

JBoss Runtime

JBoss Application Server 5.0

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

Name: JBoss Runtime

Home Directory: /opt/servers/jboss

JRE: Default JRE for J2SE-1.4

Configuration

Directory: /opt/servers/jboss/server/default_copy

default_copy

Figure 2.12. Runtime with copied configuration

Click the **Finish** button and you will see your new runtime in the list.

You can also change the configuration of existing runtime to a copied one in the same way by selecting **Window** → **Preferences** → **Server** → **Runtime Environments** and clicking the **Edit** button.

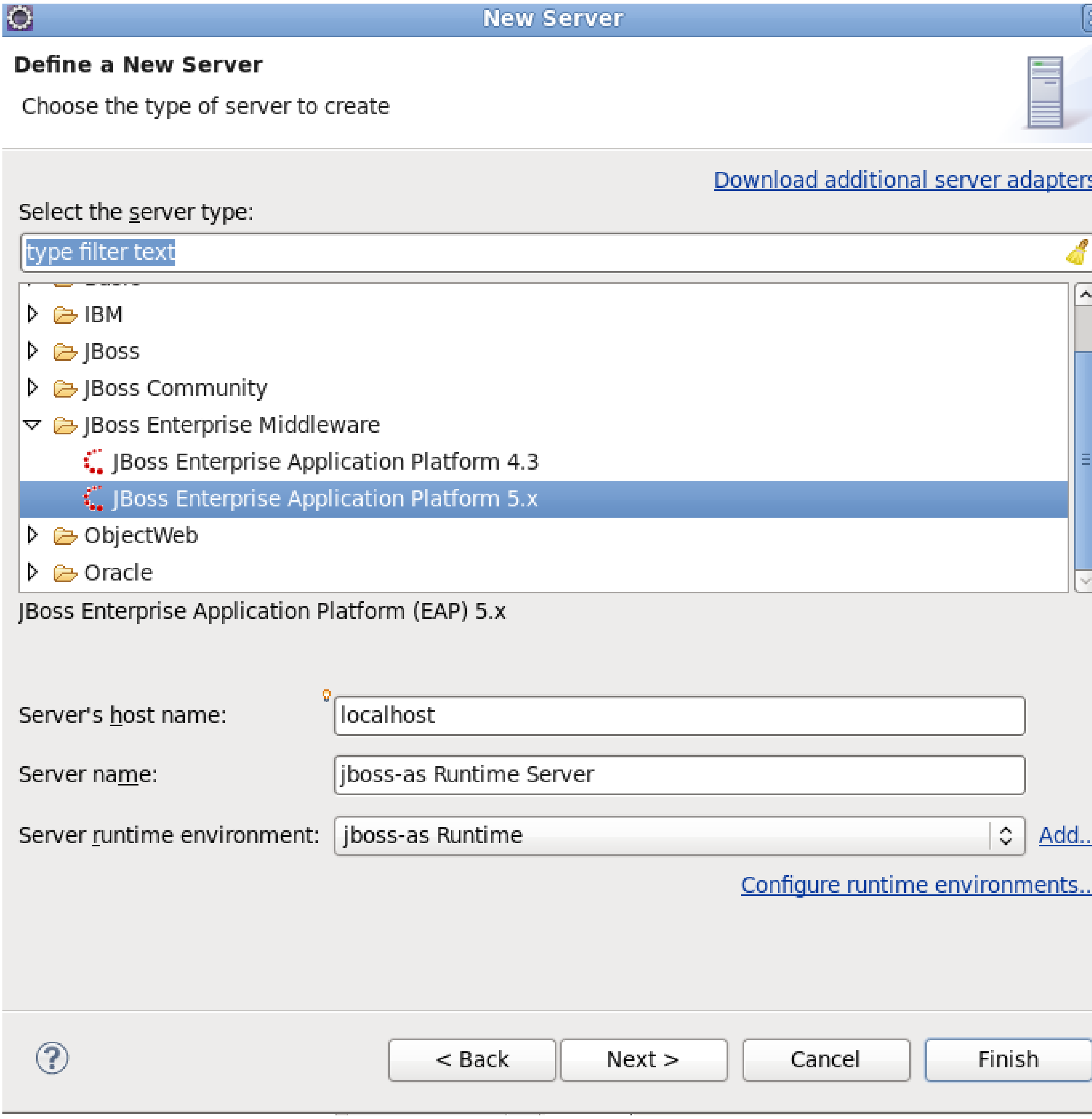
2.2. Servers

WTP servers are Eclipse-representations of a back end server installation. They are used to start or stop servers, deploy to servers, or debug code that will run on the server. They keep track of the modules (JARs, WARs, etc) you deploy to the server and also allow you to undeploy those modules (see *Section 5.1.1, "Deploying with Run On Server Wizard"*).

Servers can be started or stopped with different *command-line arguments [46]*. They are often backed by a runtime object representing that server's location.

2.2.1. Creating a New Server

There are many ways to get to the new server wizard. One way is to select **File** → **New** → **Other...** → **Server**. This should show the wizard like below.



New Server

Define a New Server

Choose the type of server to create

Select the server type:

JBoss Enterprise Application Platform (EAP) 5.x

Server's host name: localhost

Server name: jboss-as Runtime Server

Server runtime environment: jboss-as Runtime

Download additional server adapters
type filter text
IBM
JBoss
JBoss Community
JBoss Enterprise Middleware
JBoss Enterprise Application Platform 4.3
JBoss Enterprise Application Platform 5.x
ObjectWeb
Oracle
Add...
Configure runtime environments...
< Back
Next >
Cancel
Finish

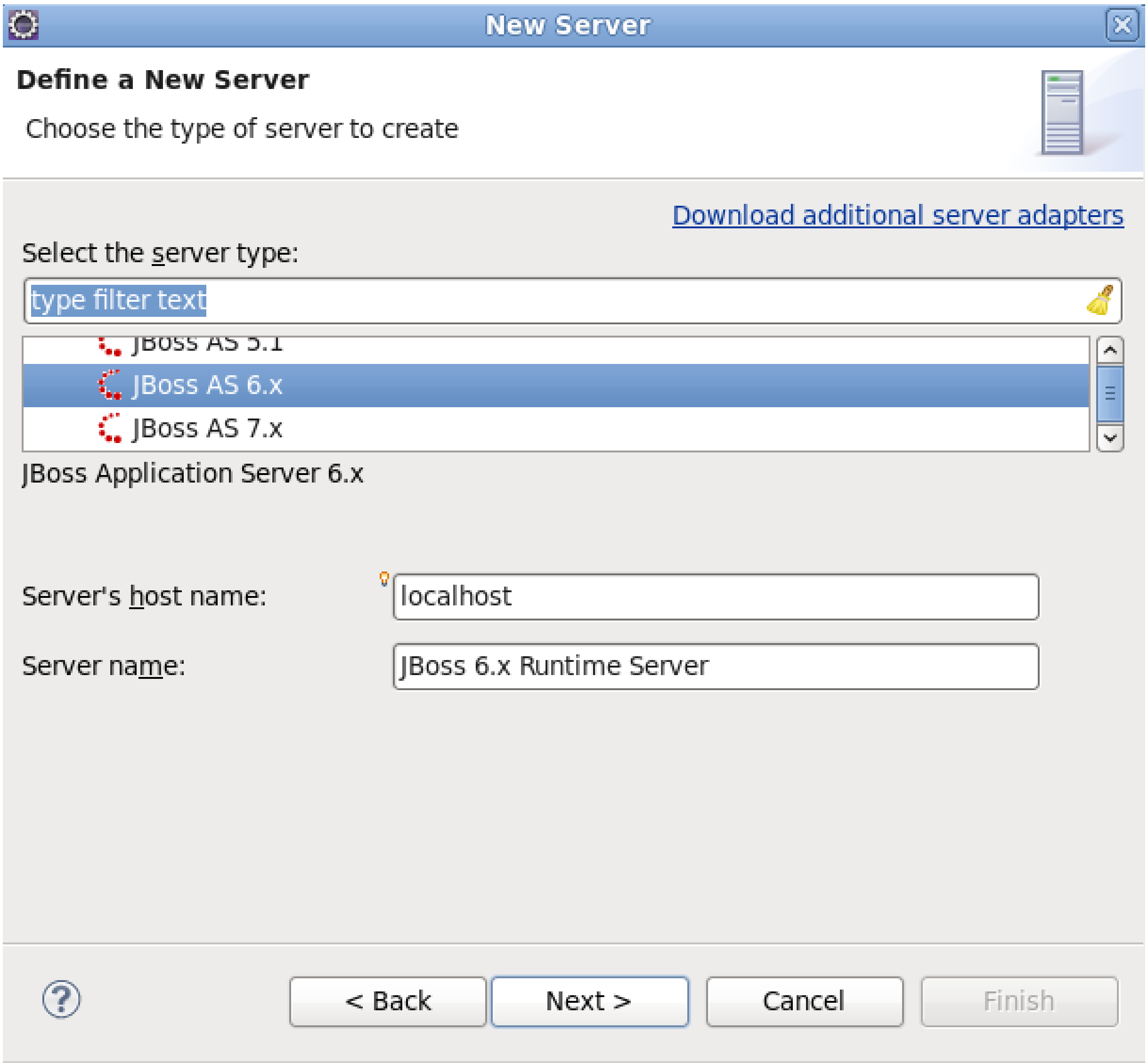
Figure 2.13. Adding a JBoss Server

A server object keeps track of the command line arguments when starting or stopping and the runtimes keep track of the location of the installation. This means that each server instance must be backed by an appropriate runtime.

The **New server wizard** allows you to name the server via the **Server name** field, or you can use a generated default name.

You can select the appropriate runtime from the **Server runtime environment** combo box. If there is no runtime that matches your needs, press the **Add...** link nearby to bring up the wizard for creating a new runtime (see *Figure 2.3, “Adding a JBoss 5.0 Runtime”*). To configure an existing runtime, go to server preferences by pressing the **Configure runtime environments...** link.

If the server you want to create does not have any installed runtime yet, the combobox and the links are absent.



Define a New Server

Choose the type of server to create

Select the server type:

type filter text

JBoss AS 5.1
JBoss AS 6.x
JBoss AS 7.x

JBoss Application Server 6.x

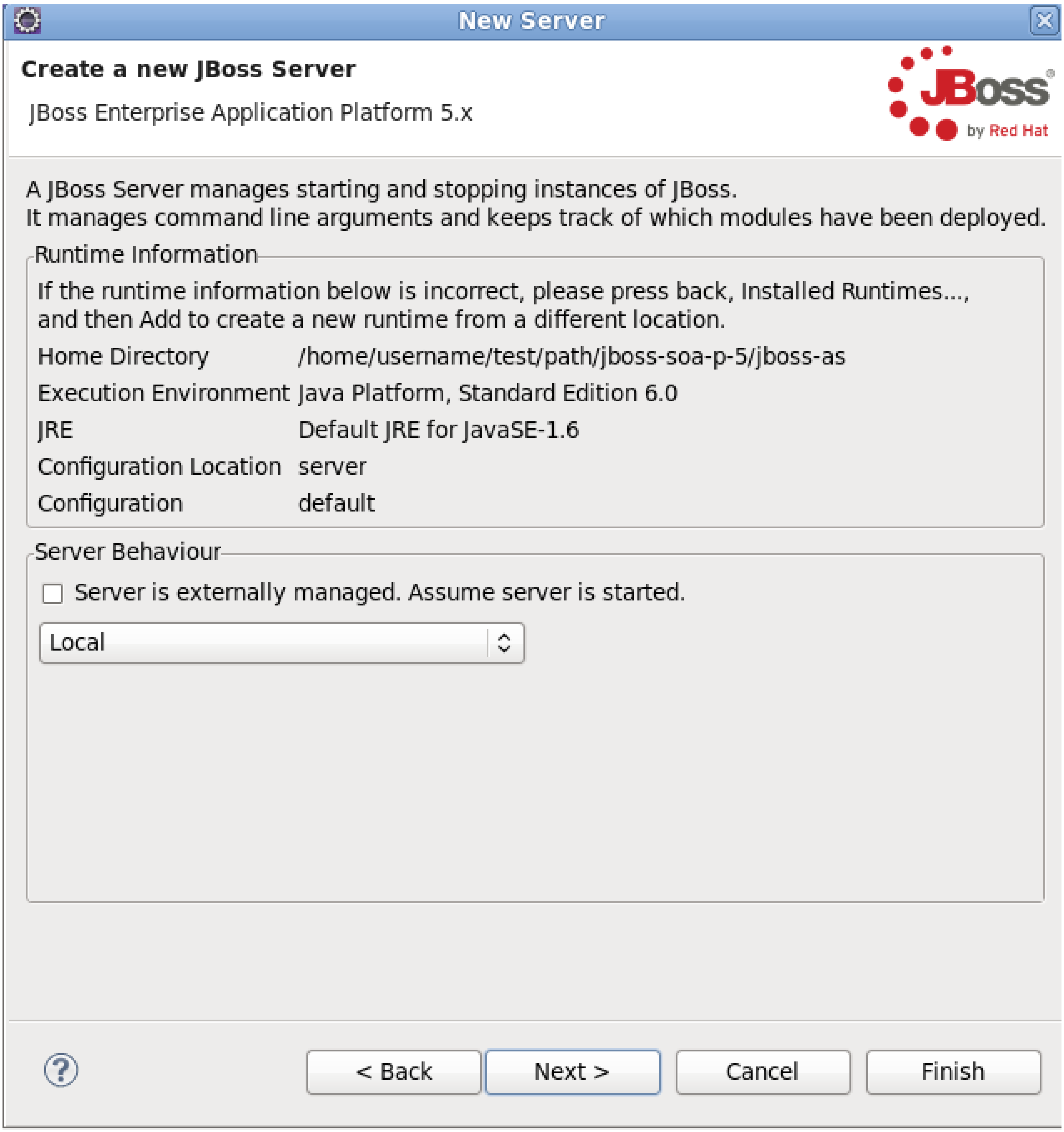
Server's host name: localhost

Server name: JBoss 6.x Runtime Server

Figure 2.14. Installed Server Runtime Environments

In this case the next page of the wizard which has the same form as in *the previous section* and will ask you to create the associated runtime.

After targeting your server to a runtime, the final screen in this wizard presents a summary of the selected options, giving you a chance to verify that you have selected the appropriate runtime.



New Server

Create a new JBoss Server

JBoss Enterprise Application Platform 5.x

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

Runtime Information

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

Home Directory	/home/username/test/path/jboss-soa-p-5/jboss-as
Execution Environment	Java Platform, Standard Edition 6.0
JRE	Default JRE for JavaSE-1.6
Configuration Location	server
Configuration	default

Server Behaviour

□ Server is externally managed. Assume server is started.

Local

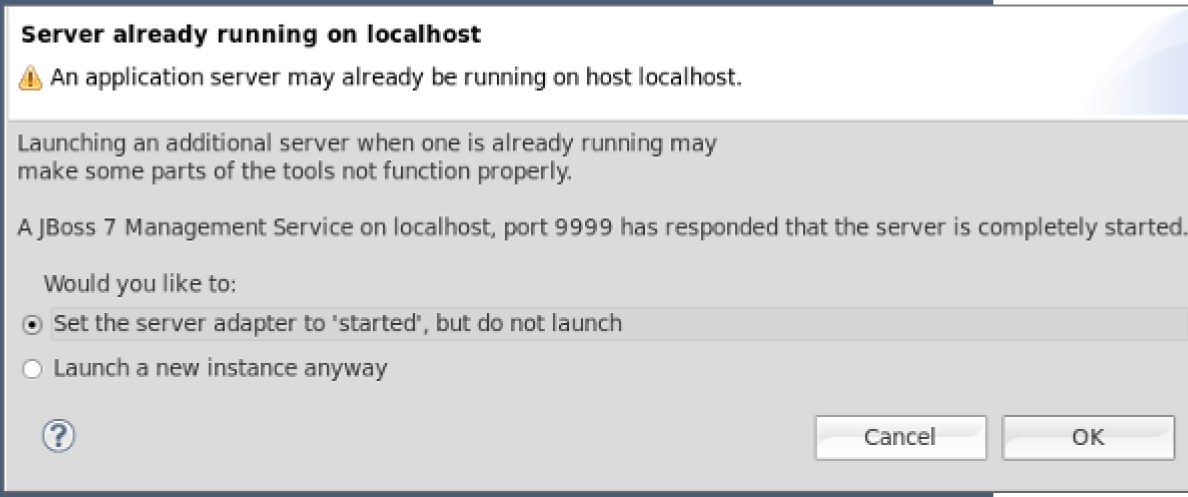
Figure 2.15. Installed Server Runtime Environments

Click the **Finish** button to complete the process of the server creation.

Now that we have created our runtimes and servers, we can explore the services and tools provided by the JBoss Server Manager.

Important

It is not recommended to run two servers on the same host, at the same time as you may experience a conflict in ports. If a server is already running on the same host a warning will appear indicating this and will ask if you wish to **Set the server adapter to 'started', but do not launch** or **Launch a new instance anyway**



Server already running on localhost

⚠ An application server may already be running on host localhost.

Launching an additional server when one is already running may make some parts of the tools not function properly.

A JBoss 7 Management Service on localhost, port 9999 has responded that the server is completely started.

Would you like to:

● Set the server adapter to 'started', but do not launch

○ Launch a new instance anyway

Figure 2.16. Multiple server adaptors warning

JBoss AS Perspective

This chapter describes how to manage installed JBoss Servers™ via the **JBoss AS** perspective.

The **JBoss AS** perspective is similar to the **Java** perspective, but it contains a few additional views. The **Console view** and the **Properties view** are standard views. The other two views that are added are the **Project archives** view and the **Servers** view.

3.1. The Servers view

The **Servers** view is built on the Common Navigator Framework allowing extensions and is using label decorators that make the UI compact enough without loosing the vital information.

Let's have a detailed look at the **Servers** view and its constituent components.

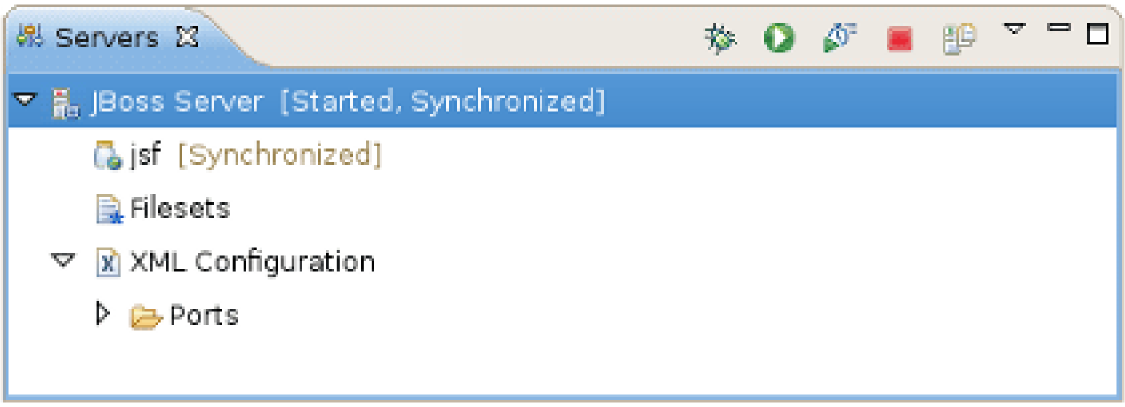


Figure 3.1. The Servers view

3.1.1. Servers view Toolbar

In the right top corner of the **Servers** view there is a special toolbar which provides a quick access for starting a server (in the debug mode, run mode, or profile mode), restarting a server, stopping a server and a publishing to a server.

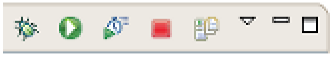


Figure 3.2. The Servers view Toolbar

In order to debug your applications or EJB's that are deployed to the server, the server must be started in debug mode. By starting the server in debug mode, Eclipse will allow you to set breakpoints on code in your workspace and step through the code.

The **Publish to the server** button will republish any modules where it has determined that the workspace is out of synchronization with the server. It will attempt to do an incremental publish if the module in question is capable of doing one.

3.1.2. Servers view Structure

The **Servers** view displays all defined servers as well as their current status (that is whether they are started or stopped) in square brackets next to the server name.

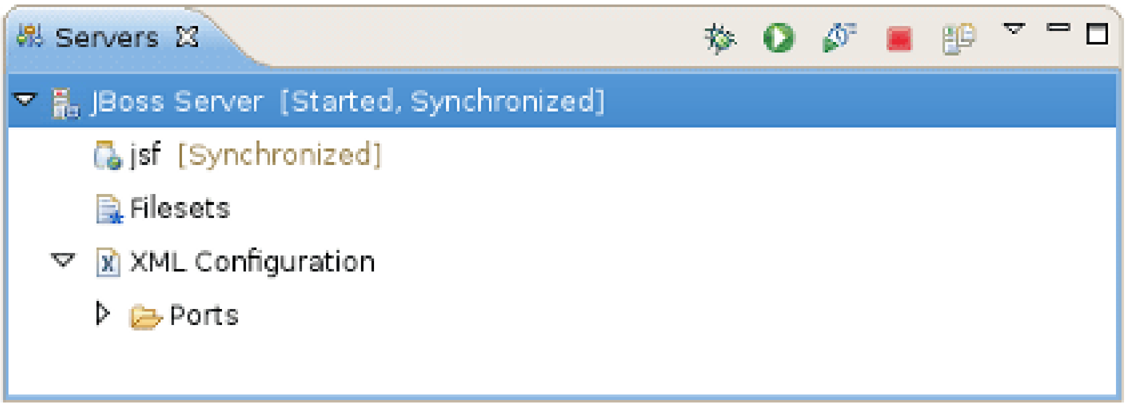


Figure 3.3. The Servers view

The following table lists possible server statuses.

Table 3.1. Server Publish Status

Status	Description
Republish	The status which allows you to see if changes are awaiting
Publishing...	The status which shows if changes are being updated
Synchronized	The status which allows you to see if changes are synchronized

You can control a server behavior as well as adjust a number of server preferences through the context menu.

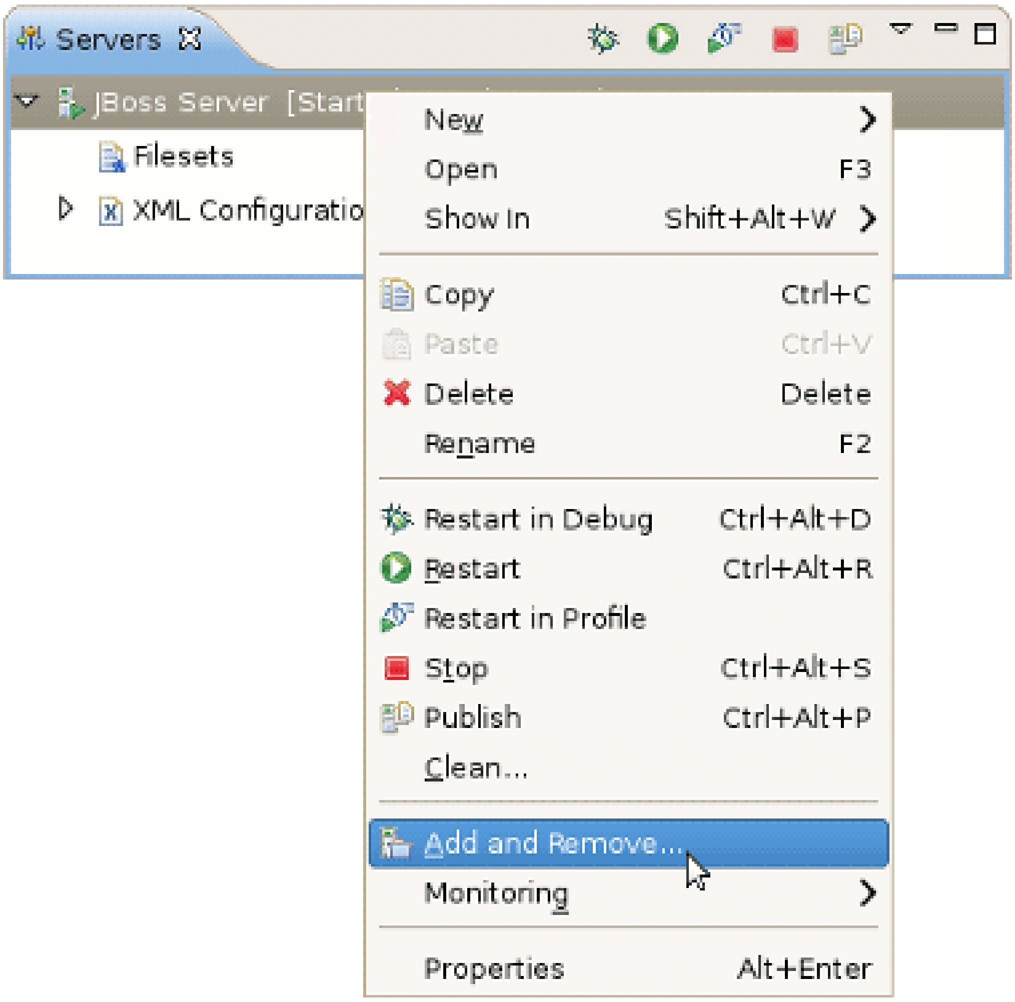


Figure 3.4. Context Menu Commands

All available context menu commands are described in the following table.

Table 3.2. Server Properties through the Context Menu

Name	Description
New Server	The option allows you to define a new server
Open	The option opens the Server editor
Show in	This option provides easy access to the Console, Debug, Server Log or MBean Explorer views
Delete	Standard option that allows you to delete the chosen server
Start	This will start the server in a run mode
Debug	This will start the server in a debug mode
Stop	This will stop the server
Publish	This will synchronize the publish information between the server and workspace

Name	Description
Explore	This action uses the native OS file explorer to browse the deploy destination. Note: The option is also available for deployed resources and projects (see the figure below).
Add and Remove Projects	This option will publish a new project to the server (if it's type is supported)
Monitoring	Allows you to add ports to be monitored on the current server
Properties	Opens a window that allows you to adjust the current server preferences

Under the server element in the **Servers** view, you can see modules that are currently deployed to the server and some server extensions that provide additional information on the server.

The context menu for any module allows you to remove it from the server and force a full or incremental republish upon it.

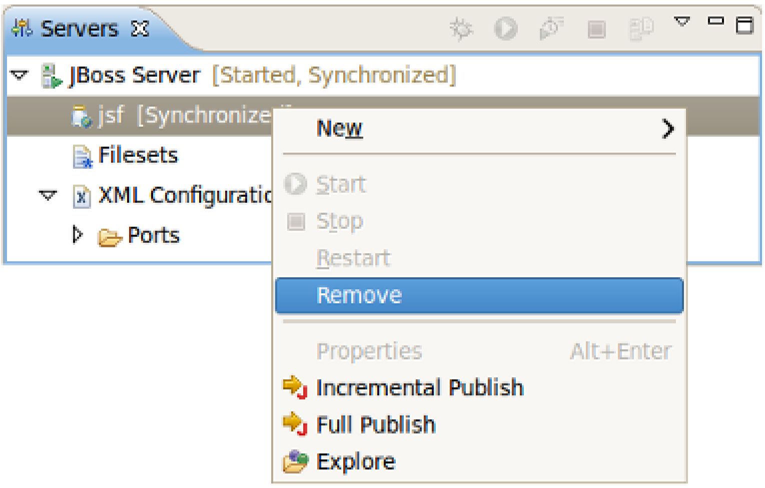


Figure 3.5. Modules Action

3.1.2.1. Filesets

The **Filesets** category in the **Servers** view provides a way to filter files.

To add a new file filter, right-click the **Filesets** category and select the **Create File Filter** option.

The **New File Filter wizard** should appear.

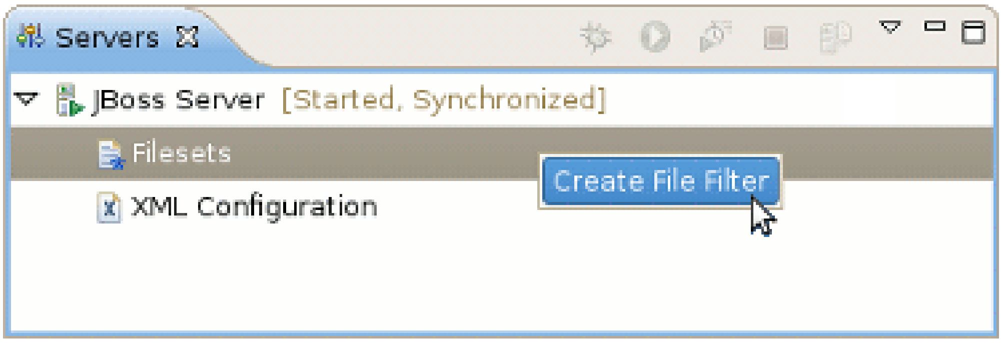
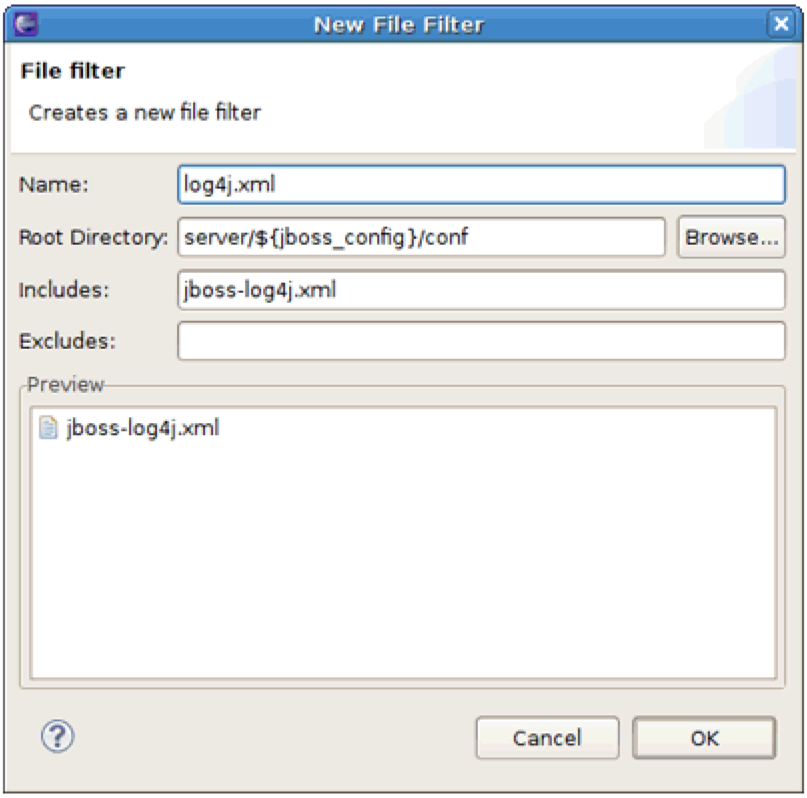


Figure 3.6. Creating a New File Filter

The wizard asks you to enter the filter name and add includes and excludes patterns. The preview box underneath provides a list of files matched to the defined patterns (see the figures bellow).

In order to set up a default fileset relative to the fixed configuration of the server runtime, use the following variable: `${jboss_config}`, i. e. you should enter `server/${jboss_config}/` in the **Root Directory** option. This allows you to modify the runtime's configuration and not have to manually update paths.



New File Filter

File filter

Creates a new file filter

Name: log4j.xml

Root Directory: server/${jboss_config}/conf

Includes: jboss-log4j.xml

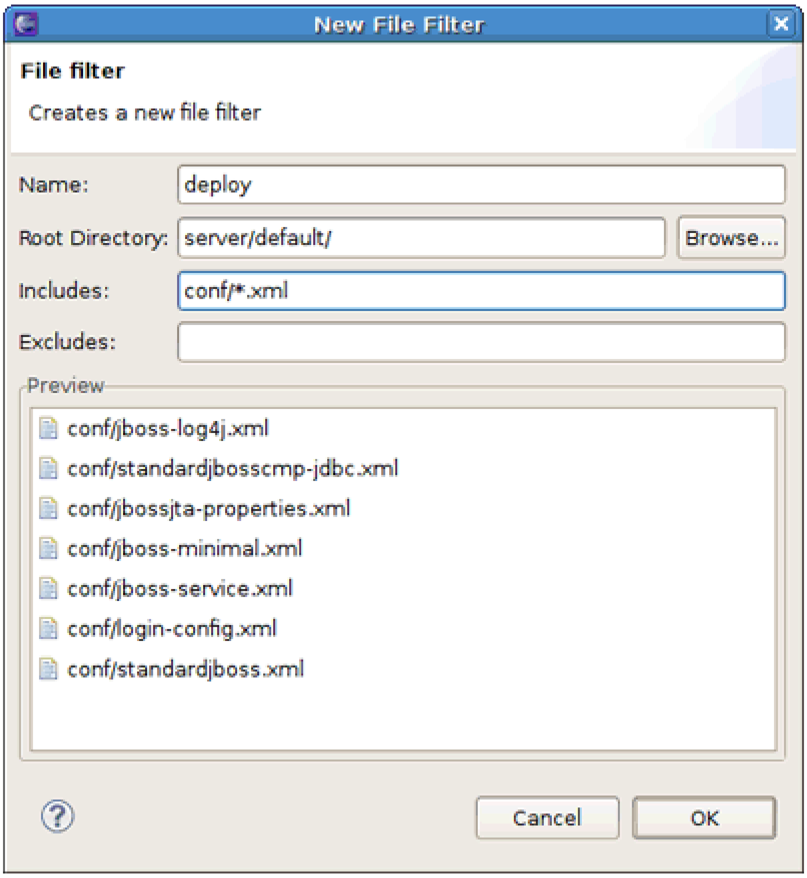
Excludes:

Preview

jboss-log4j.xml

Figure 3.7. New File Filter Wizard

Notice, that the *Browse* button still returns an absolute path:



Name: deploy

Root Directory: server/default/

Includes: conf/*.xml

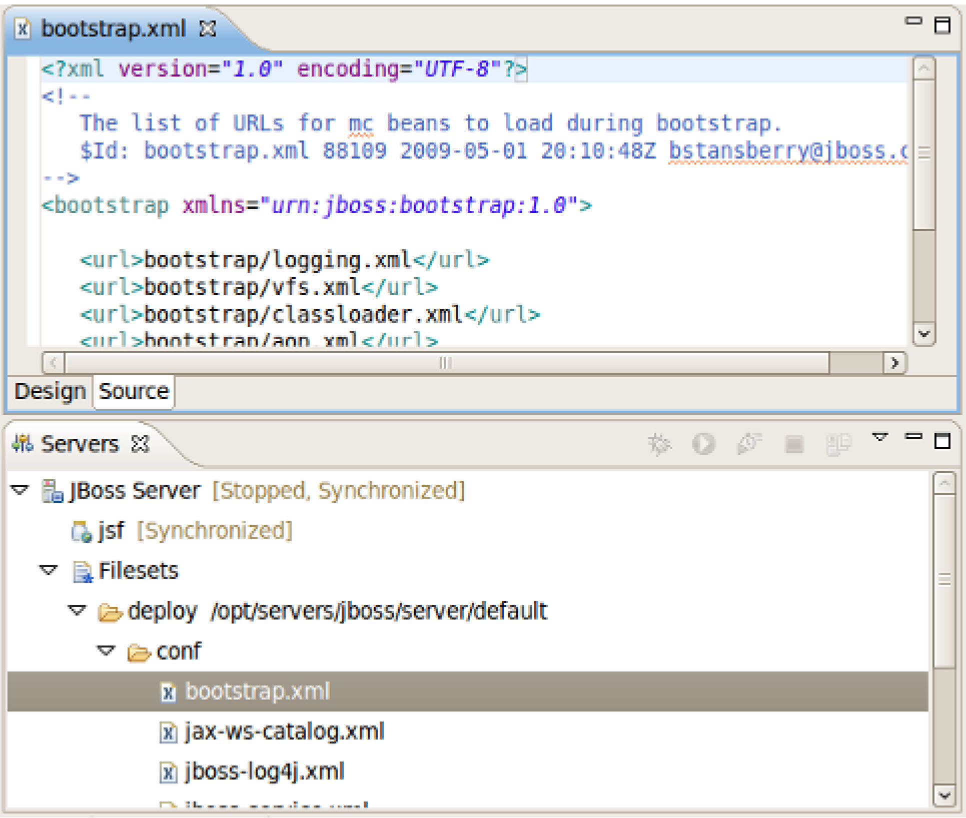
Excludes:

conf/jboss-log4j.xml
conf/standardjbosscmp-jdbc.xml
conf/jbossjta-properties.xml
conf/jboss-minimal.xml
conf/jboss-service.xml
conf/login-config.xml
conf/standardjboss.xml

Figure 3.8. New File Filter Wizard

After the filter is created, you can explore it by expanding the **Filesets** category in the **Servers** view.

It is now possible to edit files directly from the **Filesets** category. Double clicking on a file from **Filesets** opens up the editor automatically, or you can use the **Edit File** context menu command.



```
<?xml version="1.0" encoding="UTF-8"?>
<!--
   The list of URLs for mc beans to load during bootstrap.
   $Id: bootstrap.xml 88109 2009-05-01 20:10:48Z bstansberry@jboss.c
-->
<bootstrap xmlns="urn:jboss:bootstrap:1.0">

   <url>bootstrap/logging.xml</url>
   <url>bootstrap/vfs.xml</url>
   <url>bootstrap/classloader.xml</url>
   <url>bootstrap/aop.xml</url>
```

Figure 3.9. Direct Editing from the Filesets

To delete a file filter (or just a file) from the **Filesets**, right-click a file filter or file and select the **Delete File Filter** or **Delete File** command.

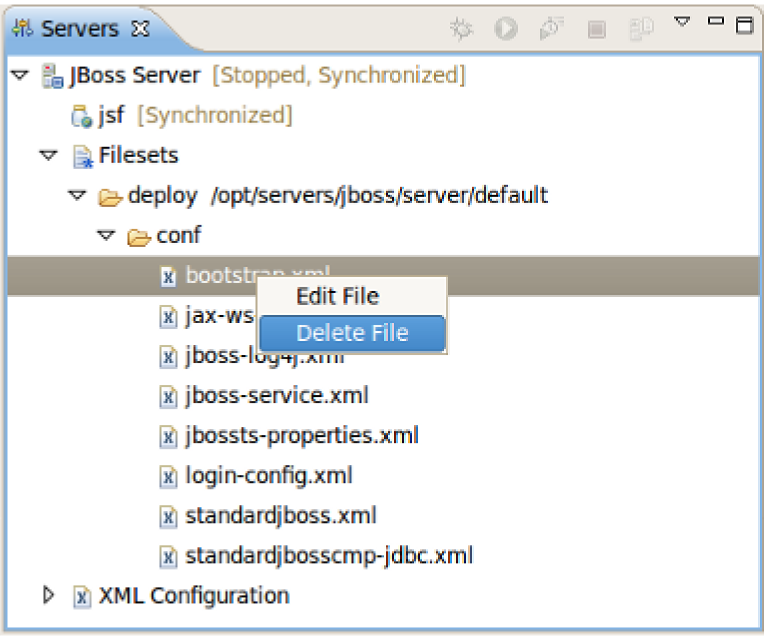


Figure 3.10. Deleting the File from the Filesets

If you want to set filesets for some server types, select **Window** → **Preferences** and then select **Server** → **Default** from the categories available on the left.

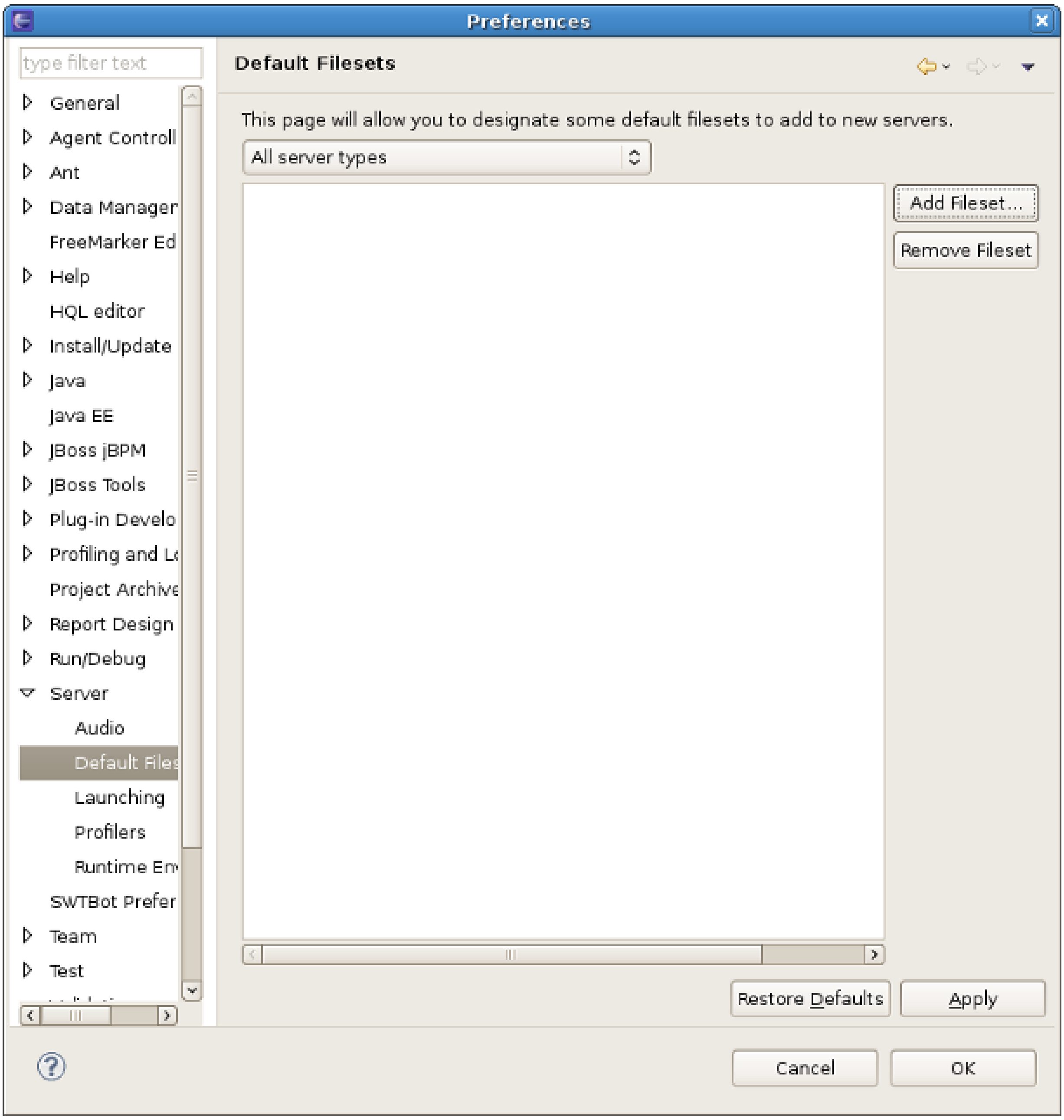


Figure 3.11. Deleting the File from the Filesets

On this preference page you can add a fileset to any server type or to all servers at once. To do this you should select the server type in the combo box and click the **Add fileset...** button. In the opened **New File Filter wizard** follow the steps described in *Section 3.1.2.1, "Filesets" [32]* and finally click the **Apply** button on the preference page.

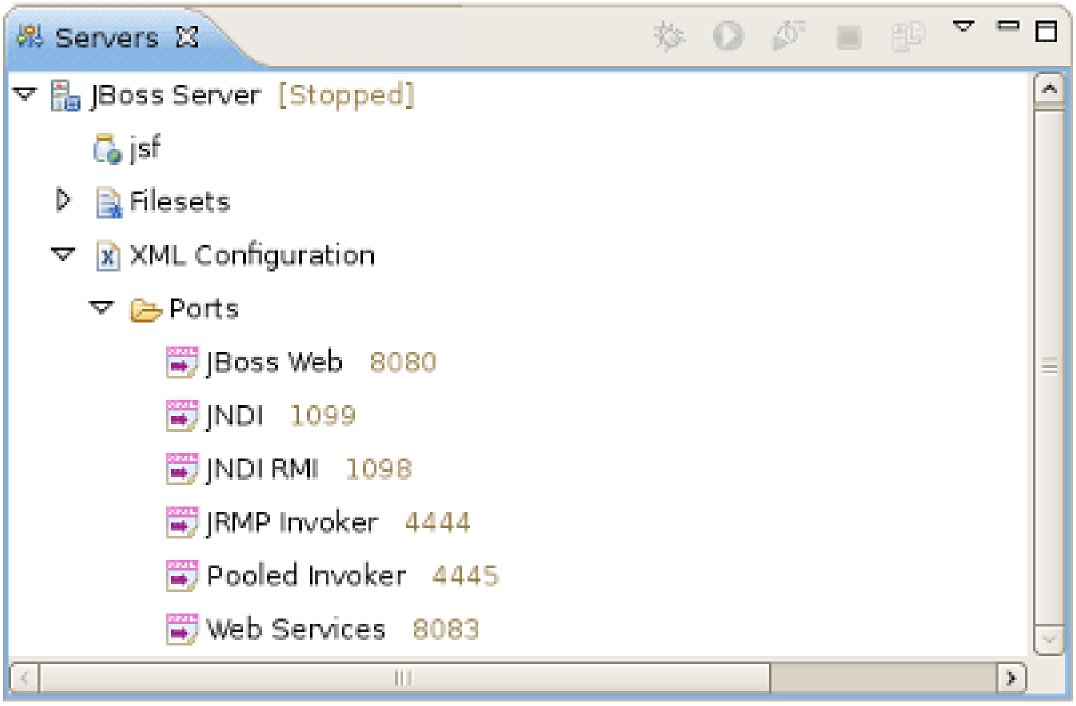
The defined file filter will be automatically added to new servers during their creation.

3.1.2.2. XML Configuration

The **XML Configuration** category allows you to quickly browse to descriptor files in your server's deploy directory and check or change the values. Basically, **XML Configuration** includes XML XPaths, where an XPath is a path used to access some specific part of an XML document.

> **Note:**
>
> This document assumes that you are familiar with XPath. If not, we highly suggested that you look through an appropriate manual or tutorial on the topic.

The **XML Configuration** category itself contains only a list of categories. **Ports** are provided by default and display many of the most commonly used ports in the JBoss Server™.



Servers

JBoss Server [Stopped]

jsf

Filesets

XML Configuration

Ports

JBoss Web 8080

JNDI 1099

JNDI RMI 1098

JRMP Invoker 4444

Pooled Invoker 4445

Web Services 8083

Figure 3.12. XML Configuration

By right-clicking on the **XML Configuration** node you can create a new category. Besides, context menu for **XML Configuration** category makes possible to disable it. You can disable any category in the bottom part of the **Servers** view. Look for them in the **Inactive Categories** afterwards to re-enable.

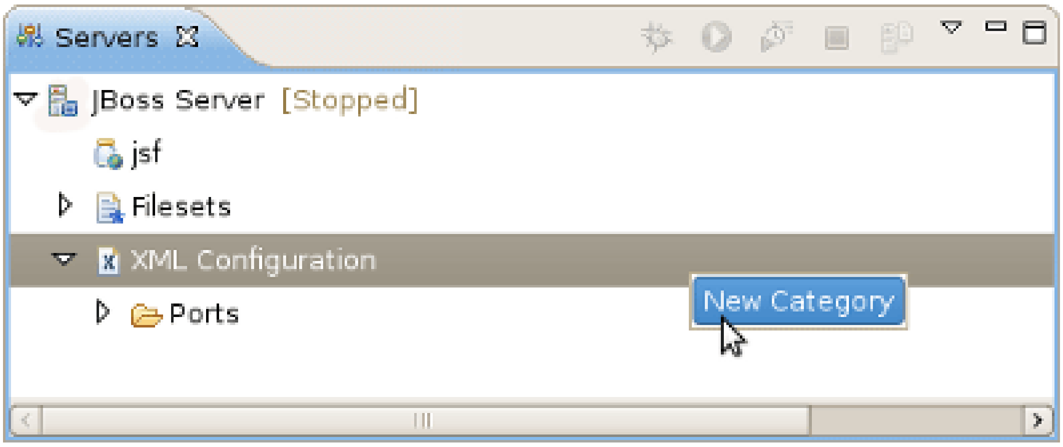


Figure 3.13. Adding New Category

By right-clicking on the **Ports** category, or any other category in **XML Configuration**, you can create a new XPath.

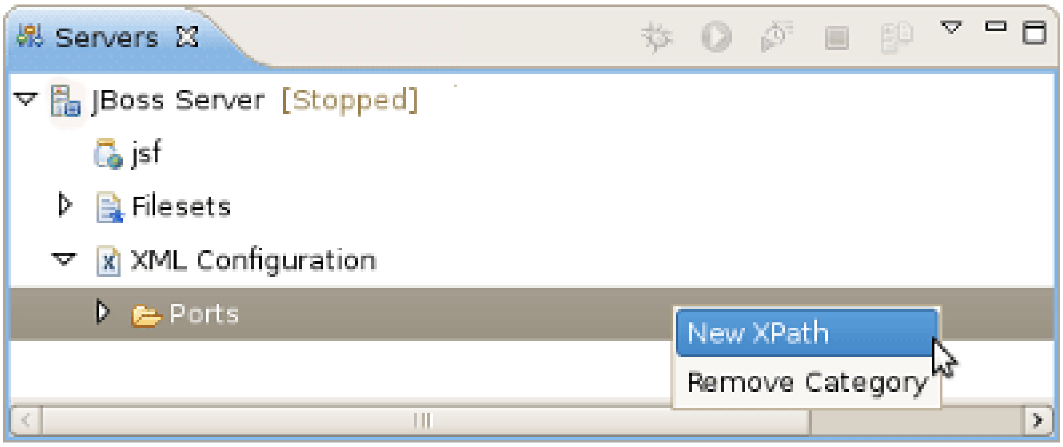
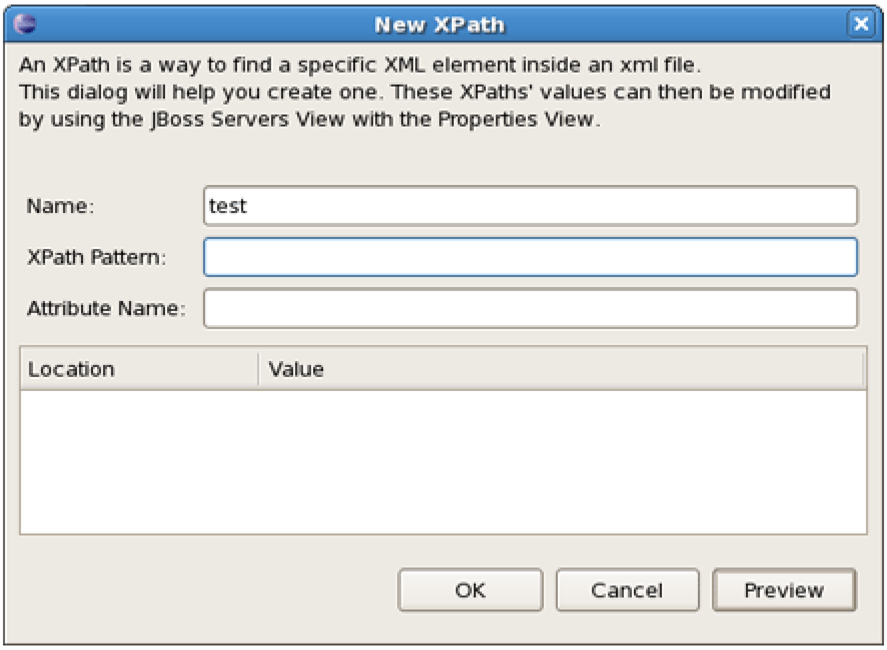


Figure 3.14. Adding New XPath

After that, the dialog shown below will appear.



New XPath

An XPath is a way to find a specific XML element inside an xml file.
This dialog will help you create one. These XPaths' values can then be modified by using the JBoss Servers View with the Properties View.

Name: test

XPath Pattern:

Attribute Name:

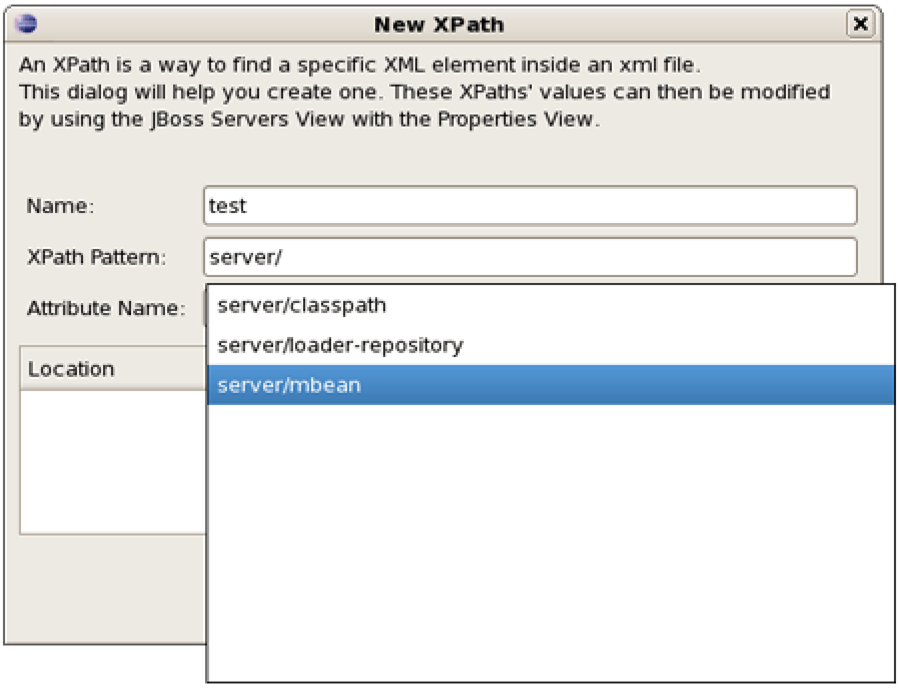
Location	Value

Figure 3.15. Adding New XPath

The goal here is to get an end result where the XPath matches up with a necessary property. With that in mind, let's look how it works. If the property you want to reach is the value of the `name` attribute in the element `<mbean>`, then your **XPath Patten** should end with `mbean` and your **Attribute Name** should be `name`, as demonstrated in the next figure.

```
...
<server>
...
    <mbean code="org.jboss.ejb.EJBDeployer"
                   name="jboss.ejb:service=EJBDeployer" xmbean-dd="">

     <!-- Inline XMBean Descriptor BEGIN -->
         <xmbean>
             <description>
               The EJBDeployer responsible for ejb jar deployment</description>
               ...
         </xmbean>
     </mbean>
</server>
```



An XPath is a way to find a specific XML element inside an xml file.
This dialog will help you create one. These XPaths' values can then be modified by using the JBoss Servers View with the Properties View.

Name: test

XPath Pattern: server/

Attribute Name:

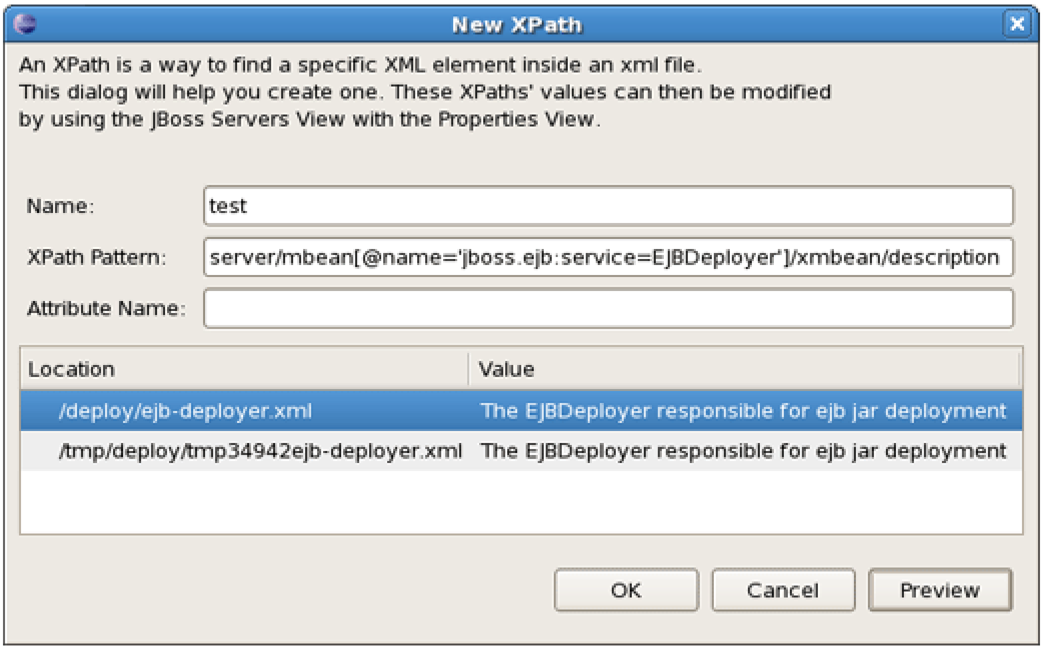
server/classpath
server/loader-repository
server/mbean

Figure 3.16. XPath Preview

Tip:

Notice when you type the fields autocomplete to help you locate exactly what XPath you're looking for.

If your desired field is the text of an element `<description>`, your **XPath Patten** should end with `description` and **Attribute Name** field should be left blank. When finished, click the **Preview** button to see how many matches are found for that particular XPath.



An XPath is a way to find a specific XML element inside an xml file.
This dialog will help you create one. These XPaths' values can then be modified
by using the JBoss Servers View with the Properties View.

Name: test

XPath Pattern: server/mbean[@name='jboss.ejb:service=EJBDeployer']/xmbean/description

Attribute Name:

Location	Value
/deploy/ejb-deployer.xml	The EJBDeployer responsible for ejb jar deployment
/tmp/deploy/tmp34942ejb-deployer.xml	The EJBDeployer responsible for ejb jar deployment

Figure 3.17. XPath Preview

3.1.3. Drag-n-Drop to Servers view

Starting from JBoss AS Tools™ 2.0.0.CR2 the **Servers** view supports drag-n-drop of deployable and runnable projects and resources.

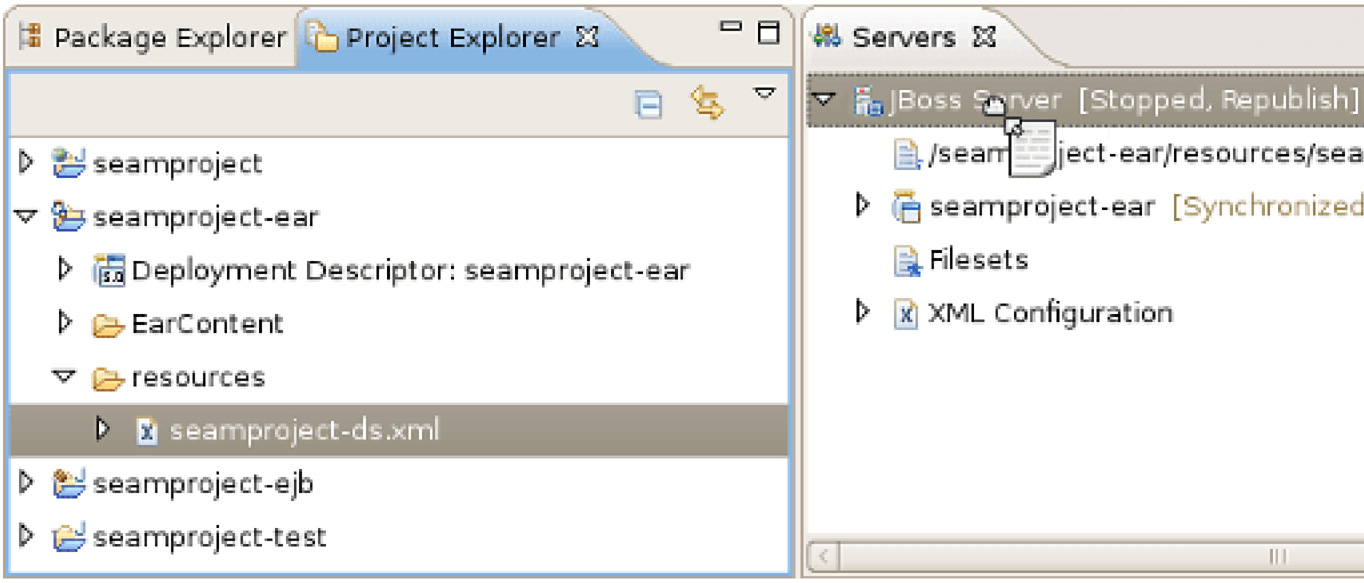


Figure 3.18. Dragging to the Servers view

With drag-n-drop the following actions can be performed:

- Dragging a project to a server will deploy it to the server and run it by showing the main page in a browser.

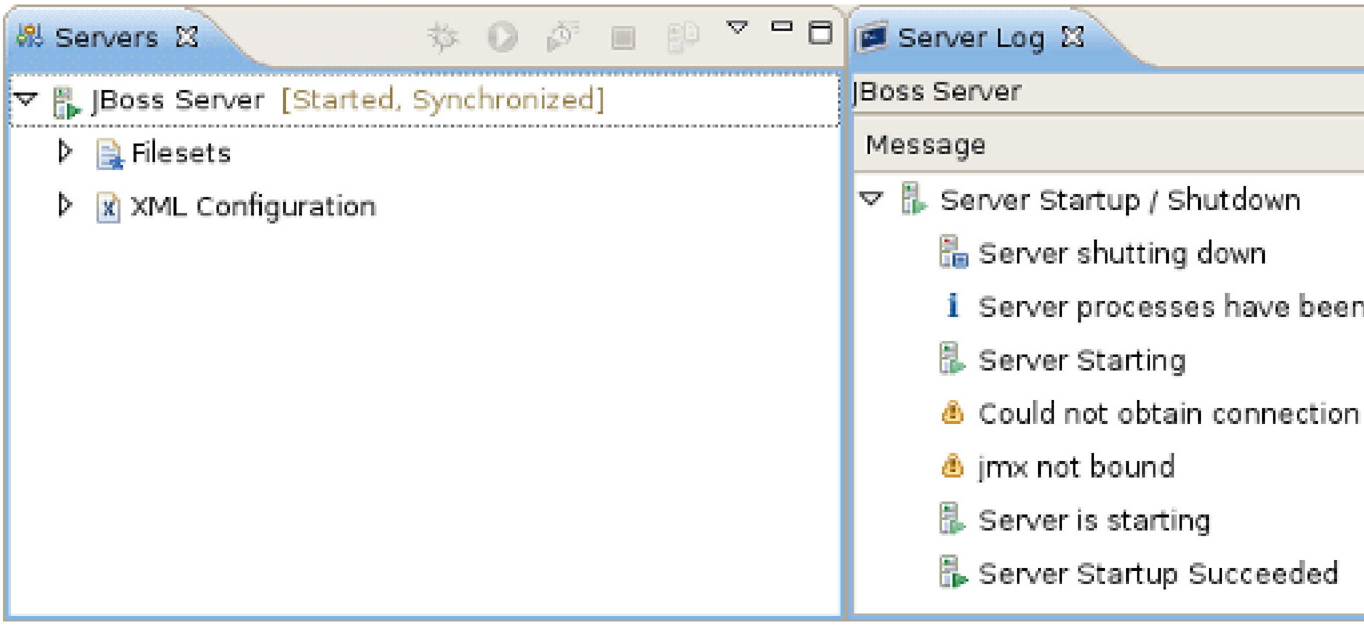
- Dragging an `.xhtml` file from the `WebContent` folder will do the same and show the corresponding page in a browser.
- Dragging a deployable resource (i.e. a datasource `-ds.xml` file that has been made deployable) will simply deploy that resource directly to the server.

In short, the feature does the same thing as if you used the **Run On Server** or **Add and Remove Projects** option in the context menu of the server.

3.1.4. Server Log View

You can monitor the current server behavior with the help of the Server Log. To open a server in the Server Log view you should right-click on the server and follow to **Open in** → **Server Log**.

The **Server Log** view shows relevant information to your server's startup, shutdown and publish processes. This allows you to keep an eye on what's going on (such as automatic incremental deployment if you have it enabled).



Server Startup / Shutdown
Server shutting down
Server processes have been
Server Starting
Could not obtain connection
jmx not bound
Server is starting
Server Startup Succeeded

Figure 3.19. Event Log Actions

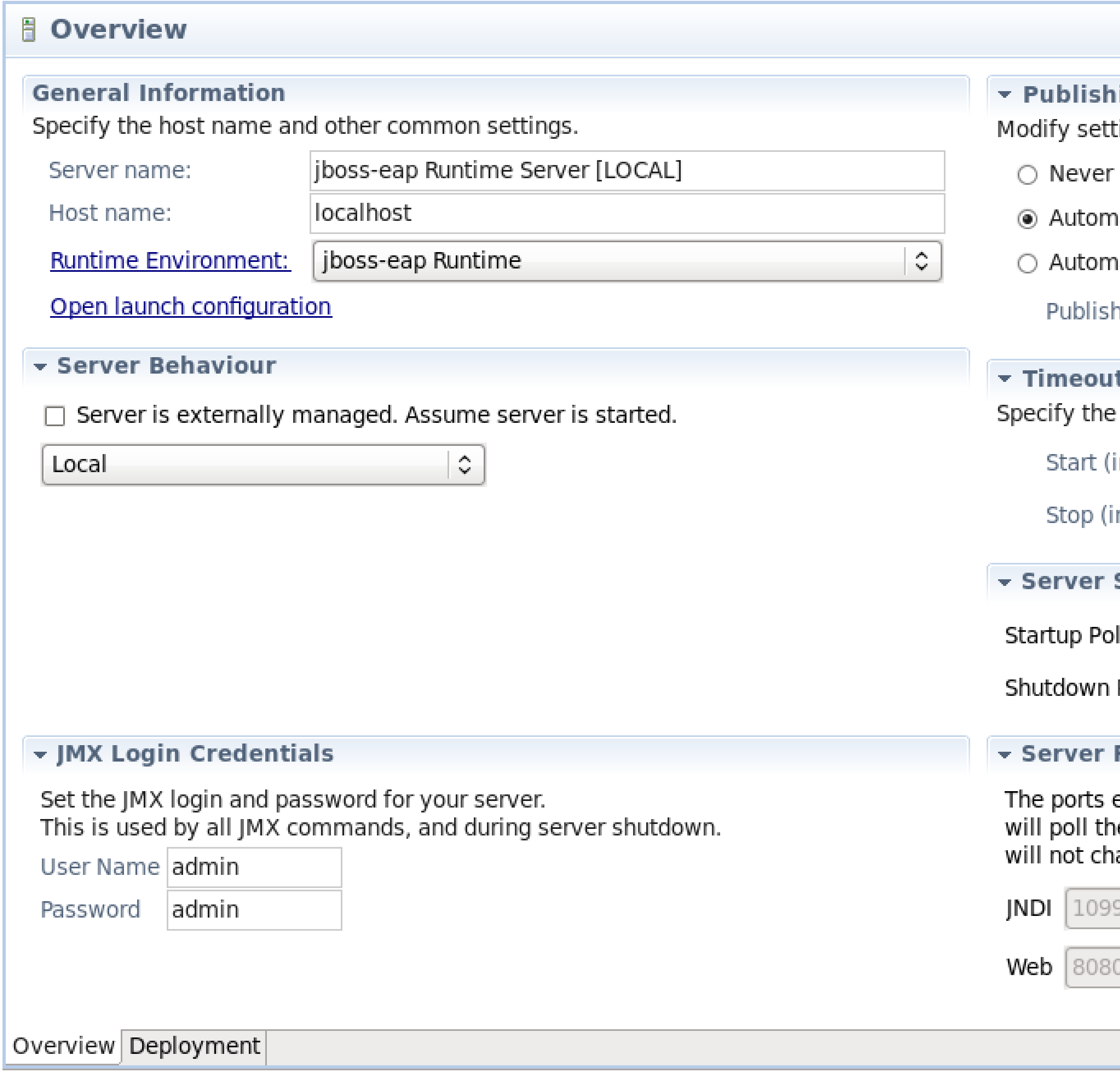
The **Server Log** view toolbar contains several icons that perform the following actions:

Table 3.3. Server Log Toolbar Icons

Name	Description
Export Log	Allows you to export the log into a text file
Clear Log Viewer	This option clears the current server log
Delete Log	Click to delete the server log
Open Log	Click to open the server log text file
Restore Log	Click to restore the server log

3.1.5. Server Editor

By double-clicking on any server, an window will appear allowing you to edit the servers settings.



Overview

General Information

Specify the host name and other common settings.

Server name: jboss-eap Runtime Server [LOCAL]

Host name: localhost

Runtime Environment: jboss-eap Runtime

Open launch configuration

Server Behaviour

☐ Server is externally managed. Assume server is started.

Local

JMX Login Credentials

Set the JMX login and password for your server.
This is used by all JMX commands, and during server shutdown.

User Name admin

Password admin

Publishi

Modify setti

○ Never

● Autom

○ Autom

Publish

Timeout

Specify the

Start (i

Stop (ir

Server S

Startup Pol

Shutdown I

Server I

The ports e
will poll the
will not cha

JNDI 1099

Web 8080

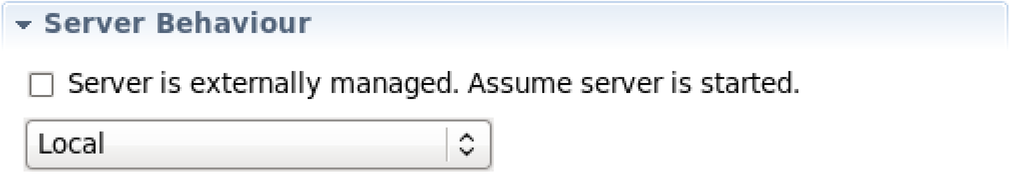
Figure 3.20. Preferences Page for the Chosen Server

Tip:

On the figure you can see that the username and password fields are available. If you get a SecurityException when trying to launch the server, it is most likely because your server is protected, which requires that you need to fill the username and password fields with appropriate values.

The **Server Behaviour** settings tab allows you to set how tool interaction with the server should be undertaken.

When you created the server, if you selected that it was a local server then you will notice that the option **Server is externally managed. Assume server is started** is unchecked and the combo-box displays **Local**.



Server Behaviour

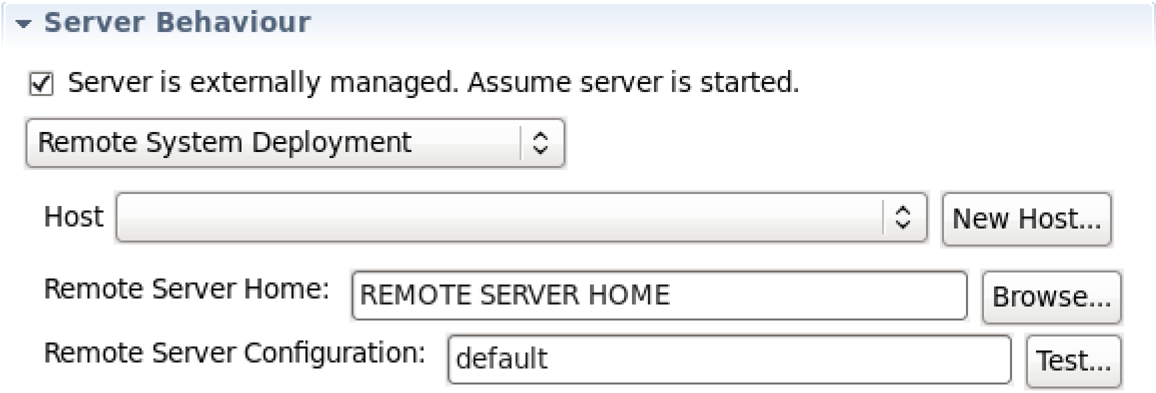
□ Server is externally managed. Assume server is started.

Local

Figure 3.21. Server Behaviour - Local

If you created a remote server then you will see that the checkbox, **Server is externally managed. Assume server is started** is checked and the combo-box displays **Remote System Deployment**. Also populated will be the **Host**, **Remote Server Home** and **Remote Server Configuration** fields.

You are also able to change a servers behaviour from **Local** to **Remote System Deployment** through this settings tab. In doing so you will see that the **Host** is not set by default, but the other fields contain default values. Once your remote server behaviour is configured, click the **Test** button to confirm the settings are correct.



Server Behaviour

☑ Server is externally managed. Assume server is started.

Remote System Deployment

Host:

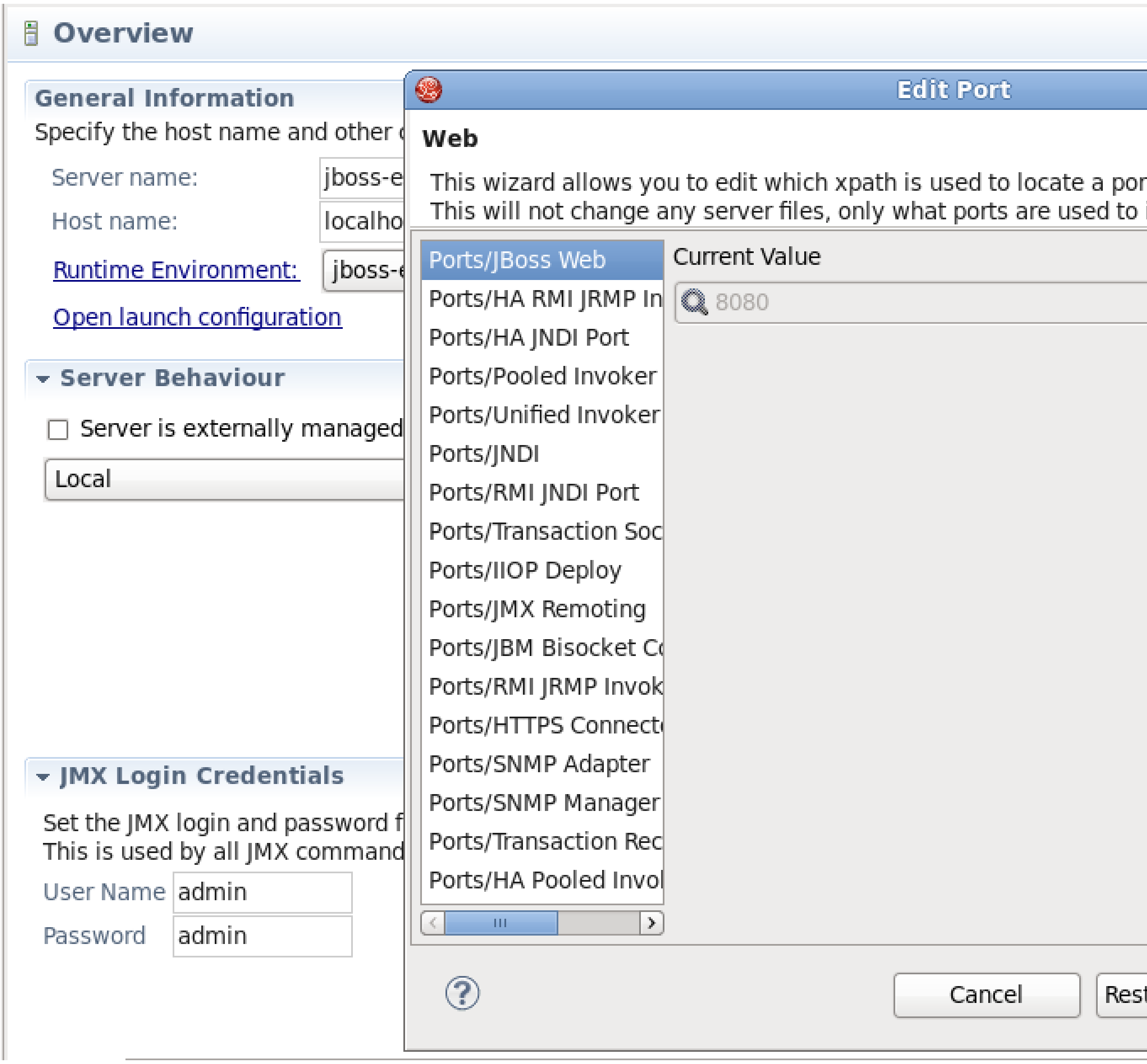
Remote Server Home: REMOTE SERVER HOME

Remote Server Configuration: default

Figure 3.22. Server Behaviour - Remote

Under the **Publishing** section it is possible to disable or enable the automatic publishing of the changes in the workspace.

It should be pointed out that the server adapter tries to automatically detect the ports it needs for integrating with a JBoss Server by default. Sometimes it is necessary to override this automatic detection if you are using a custom configuration. The **Server Ports** section in the **Server editor** provides fields to customize port settings. Click the **Configure...** link to bring up the wizard for adjusting the settings for the ports.



Overview

General Information

Specify the host name and other c

Server name: jboss-e

Host name: localho

Runtime Environment: jboss-e

Open launch configuration

Server Behaviour

□ Server is externally managed

Local

JMX Login Credentials

Set the JMX login and password f
This is used by all JMX command

User Name admin

Password admin

Edit Port

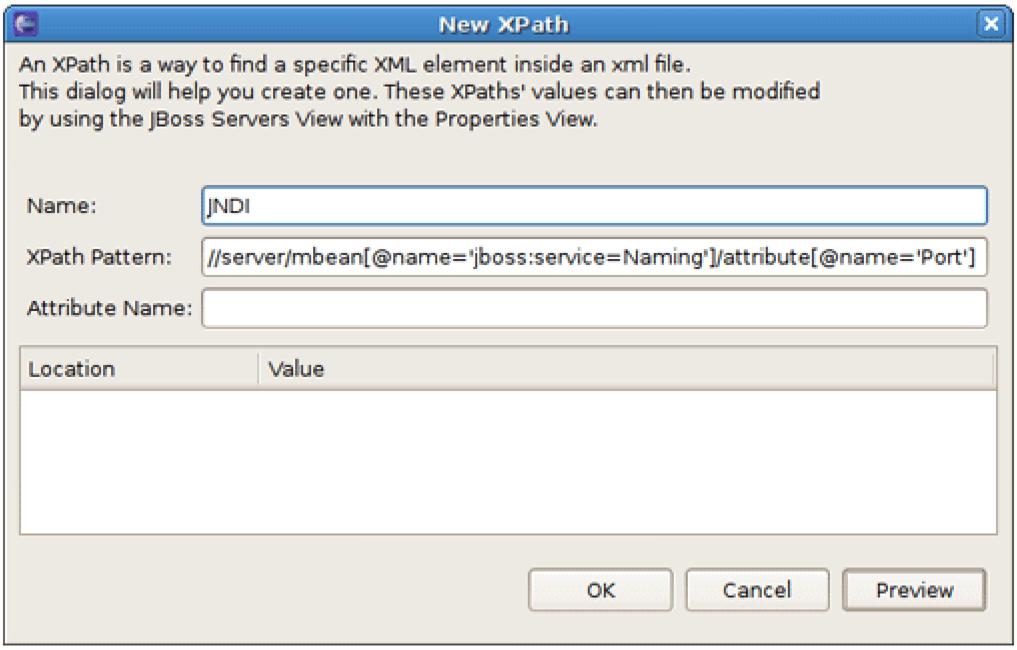
Web

This wizard allows you to edit which xpath is used to locate a por
This will not change any server files, only what ports are used to

Current Value 8080

Figure 3.23. Server Ports Preferences

Click the **Edit XPath** button for the chosen port to configure its XPath's values.



New XPath

An XPath is a way to find a specific XML element inside an xml file.
This dialog will help you create one. These XPaths' values can then be modified
by using the JBoss Servers View with the Properties View.

Name: JNDI

XPath Pattern: //server/mbean[@name='jboss:service=Naming']/attribute[@name='Port']

Attribute Name:

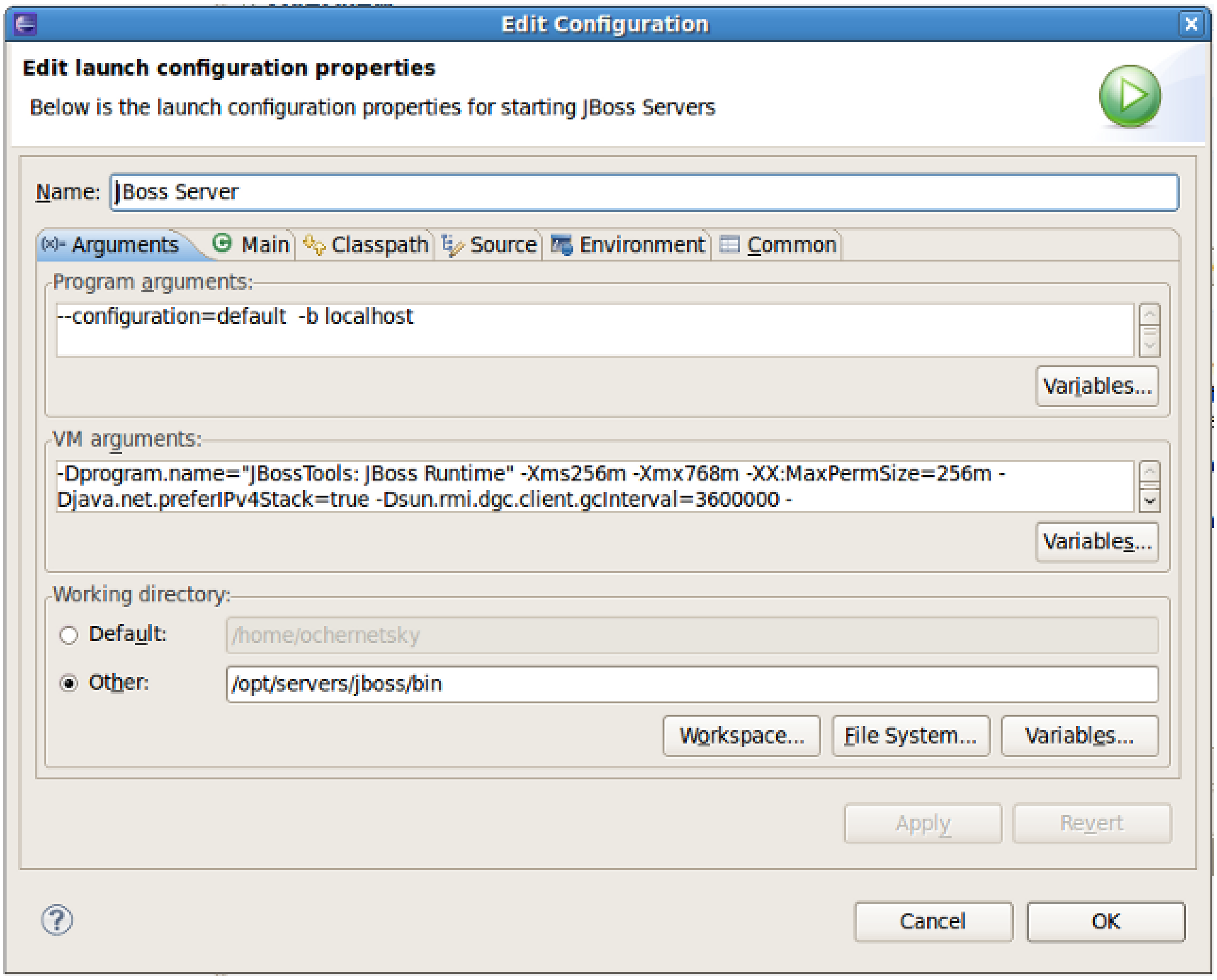
Location	Value

Figure 3.24. XPath Pattern for a Server Port

In the **Server editor** you are able to edit the timeouts and the server pollers to use.

> **Note:**
>
> By default, the Startup poller is set to JMX Poller. If you change the Startup poller to Timeout Poller (which may be required if you are using the minimal configuration for your server), this will do no polling at all and will only set the server state to **"Started"** after your startup timeout is reached.

The **Server editor** window also allows you to modify the server's launch configuration. The settings is available by clicking the the **Open launch configuration** link. The resulting window provides tabs for setting command line arguments, main, classpaths and other things that are relevant to launching the server.



Edit Configuration

Edit launch configuration properties

Below is the launch configuration properties for starting JBoss Servers

Name: JBoss Server

Program arguments:

```
--configuration=default  -b localhost
```

VM arguments:

```
-Dprogram.name="JBossTools: JBoss Runtime" -Xms256m -Xmx768m -XX:MaxPermSize=256m -
Djava.net.preferIPv4Stack=true -Dsun.rmi.dgc.client.gcInterval=3600000 -
```

Working directory:

○ Default: /home/ochernetsky

● Other: /opt/servers/jboss/bin

Figure 3.25. Launch Configuration Properties

The first tab shows the JBoss server arguments

See the *AS Installation Guide* [http://docs.jboss.org/jbossas/guides/installguide/r1/en/html/start-stop.html] to find the parameters that can be specified for JBoss Server™.

Note:

Please note that the values in the Launch Configurations for JBoss Servers are strictly enforced in order to avoid inconsistencies between server's and their configured runtime.

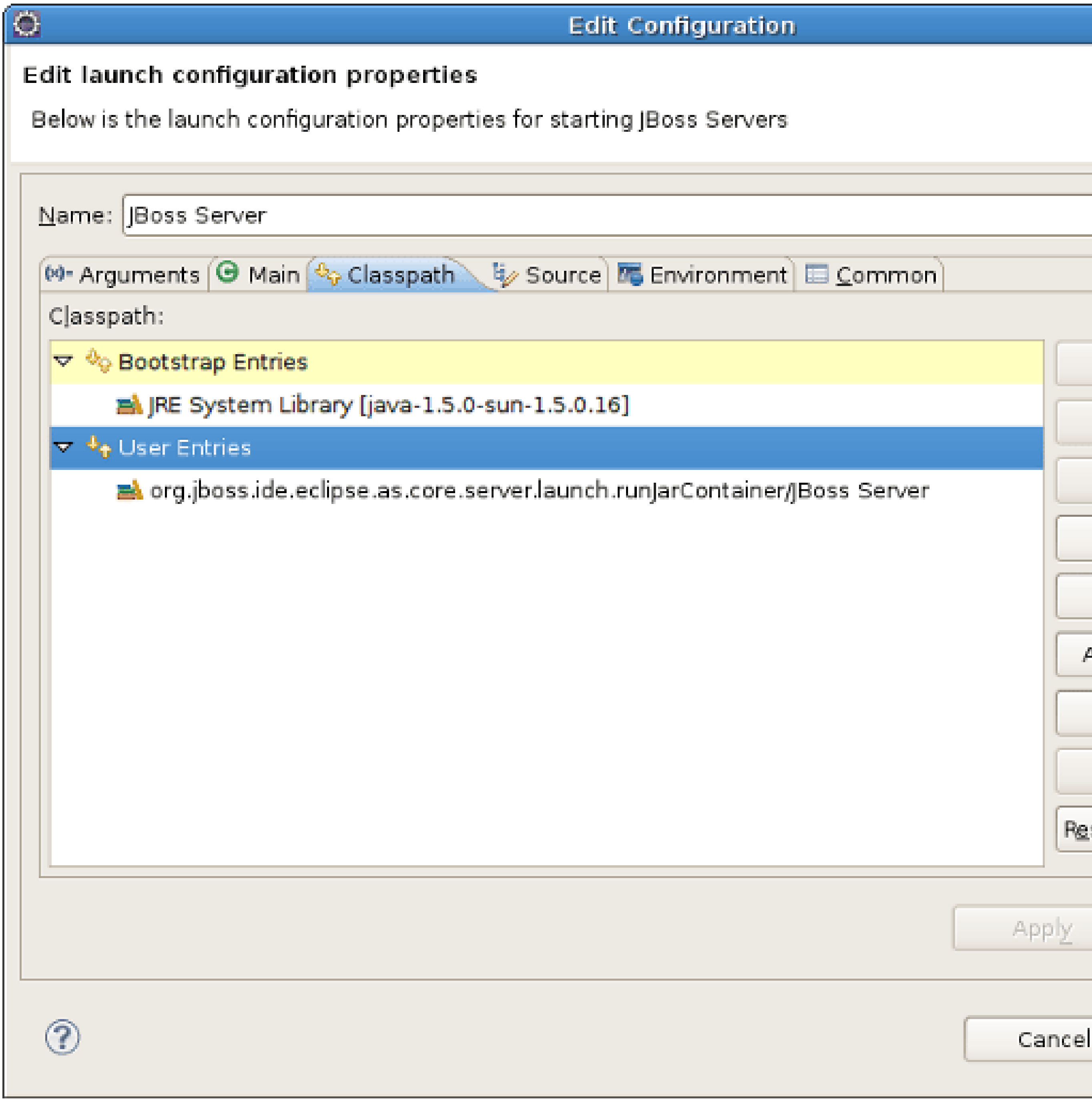
For example, if you change the launch configuration program arguments to `"-c myConfig"` but do not change the targeted runtime configuration, then your program arguments will be ignored. The configuration of the server runtime "wins" so to speak. This ensures consistency and if you change the location of the runtime, your launch configurations will automatically pick that up.

Values are not controlled by the server and its runtime setup will be passed on unaltered.

On the second tab you find the main class used for launching JBoss AS (the default is `org.jboss.Main`). This value can be changed if necessary.

Until JBoss Tools 3.0.0.GA the servers classpath was read only, but that caused problems for users wanting to add their own JARs in the startup classpath. That is relevant if you need to patch the server, add a custom charset or other tweaks that require early access to the classpath.

Now all servers have a custom 'server runtime classpath container', which is there by default and point to the default JARs in JBoss. You can now adjust the classpath. Then just make sure this container is there if you want the classpath to be picked up.



Edit Configuration

Edit launch configuration properties

Below is the launch configuration properties for starting JBoss Servers

Name: JBoss Server

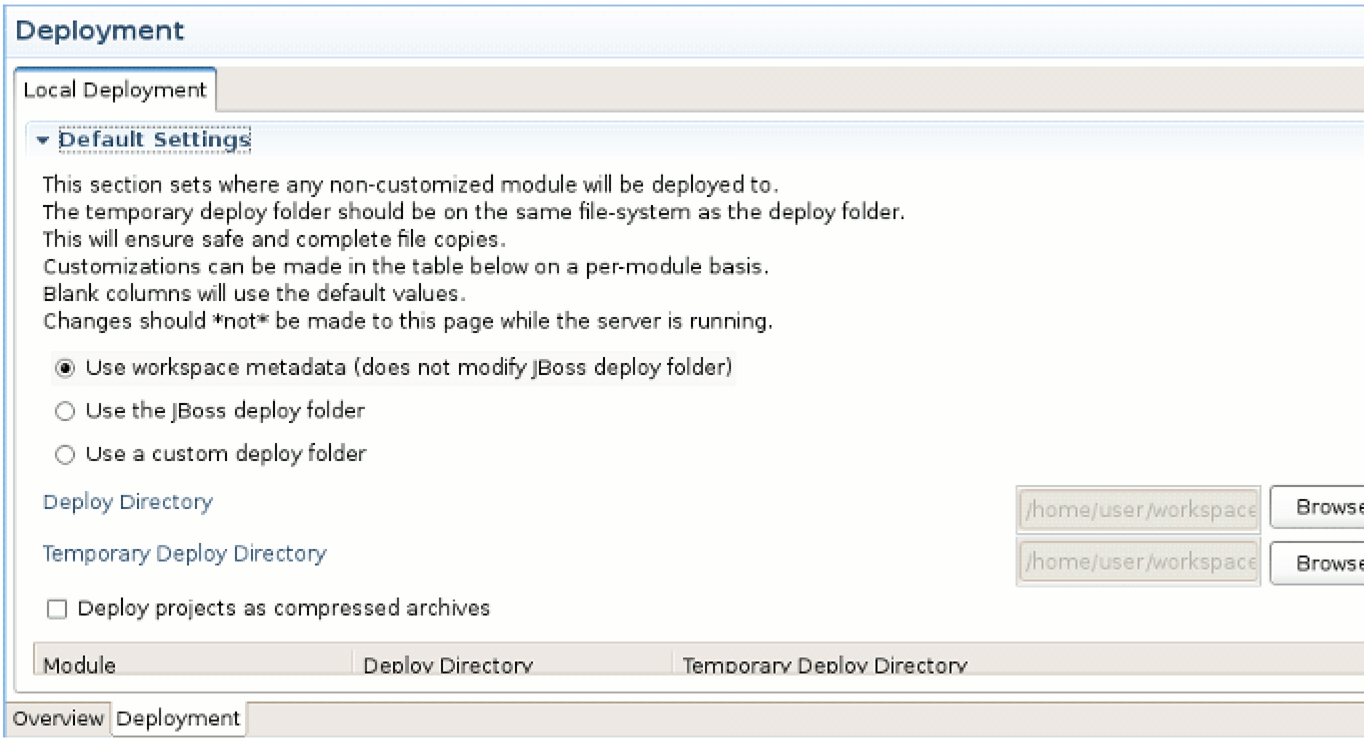
Classpath:

- Bootstrap Entries
  - JRE System Library [java-1.5.0-sun-1.5.0.16]
- User Entries
  - org.jboss.ide.eclipse.as.core.server.launch.runJarContainer/JBoss Server

Figure 3.26. Server Classpaths

If for some reason you have a launch configuration without this container, the **Restore Default Entries** button should add it properly. Also, the **Restore Default Entries** button will remove any extra entries you added yourself.

Using **Deployment tab** you configure local deployment settings.



Deployment

Local Deployment

Default Settings

This section sets where any non-customized module will be deployed to.
The temporary deploy folder should be on the same file-system as the deploy folder.
This will ensure safe and complete file copies.
Customizations can be made in the table below on a per-module basis.
Blank columns will use the default values.
Changes should *not* be made to this page while the server is running.

● Use workspace metadata (does not modify JBoss deploy folder)

○ Use the JBoss deploy folder

○ Use a custom deploy folder

Deploy Directory: /home/user/workspace

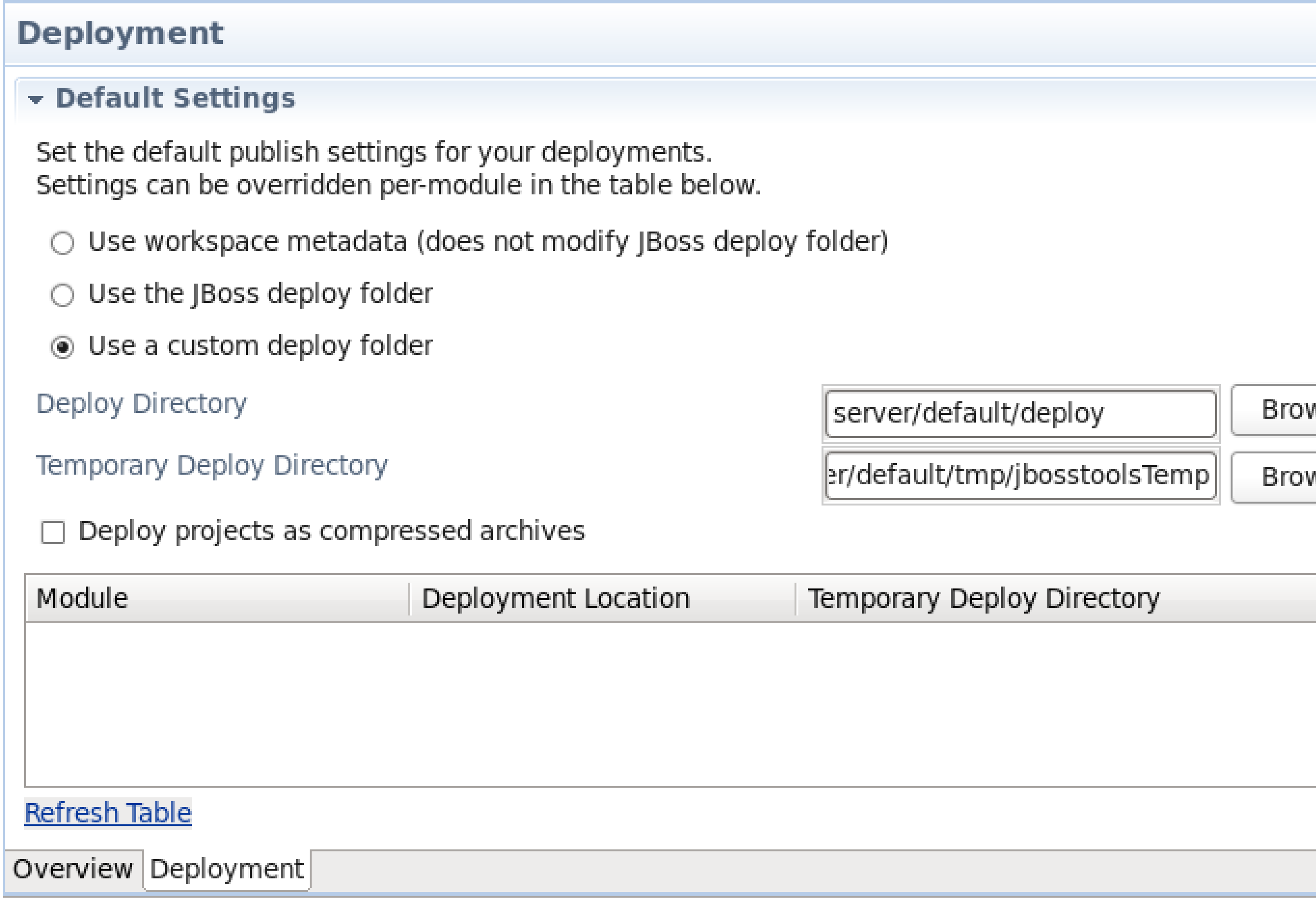
Temporary Deploy Directory: /home/user/workspace

□ Deploy projects as compressed archives

Module	Deploy Directory	Temporary Deploy Directory

Figure 3.27. Deployment tab

Using the group of radio buttons in the **Default Settings** section a user can set where the application will be deployed to. By default it is deployed to the user's workspace folder, `[workspaceDirecotry]\.metadata\.plugins`. If you would like the application to be deployed to your JBoss server deploy folder select the **Use the JBoss deploy folder** option. The option to specify a custom deploy folder is also available.



Deployment

Default Settings

Set the default publish settings for your deployments.
Settings can be overridden per-module in the table below.

○ Use workspace metadata (does not modify JBoss deploy folder)

○ Use the JBoss deploy folder

● Use a custom deploy folder

Deploy Directory: server/default/deploy

Temporary Deploy Directory: er/default/tmp/jbosstoolsTemp

□ Deploy projects as compressed archives

Module	Deployment Location	Temporary Deploy Directory

Figure 3.28. Deployment tab - custom deploy folder

3.1.6. Relevant Resources Links

Find more about XPath in the *XPath Documentation* [http://www.w3.org/TR/xpath20/].

3.2. Project Archives View

Every application, whether Plain Old Java, J2EE, or some other language altogether, needs to be packaged in some way. In Java-related projects, many people use ANT.

> **Note:**
>
> Those who use ANT will appreciate how the Project Archives Ant task is now improved: it supports variables and gives more informative error/logging messages when something goes wrong.

But JBoss Tools™ comes with our own Archives tool with simpler and less-verbose XML and a handy user interface. The Project Archives plugin consists primarily of the **Project Archives** view to set up each packaging configuration.

Let's look through all functionality that the **Project Archives** view provides.

3.2.1. Overview

The packaging configuration for each project is stored in the project's root folder in a file named `.packages`, which has a fairly simple XML structure. Modifying the file by hand is neither required nor recommended, as the UI is the only supported way to modify your packaging structure.

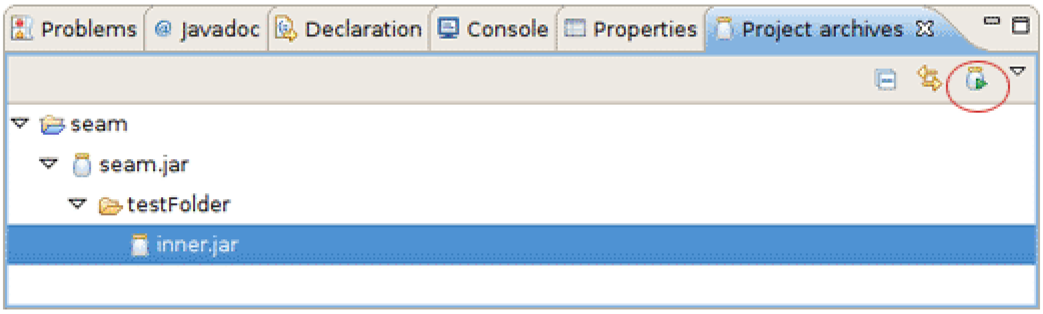


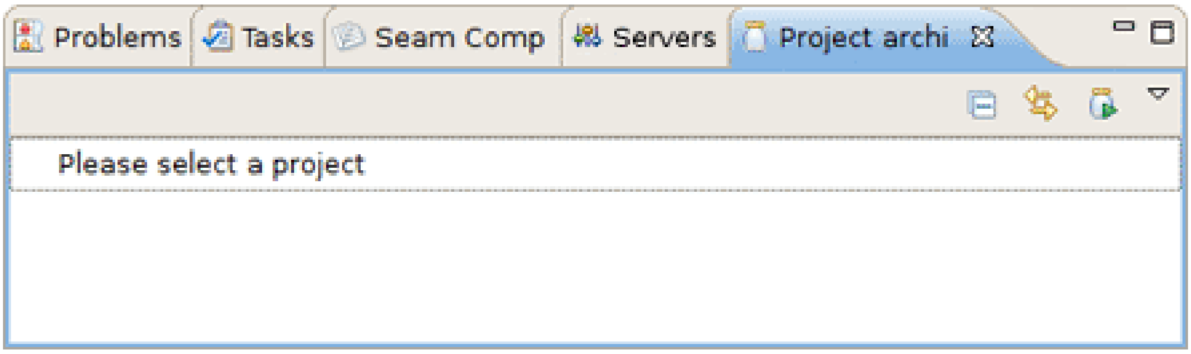
Figure 3.29. Archives View

A project's configuration contains archives. As you can see on the image above a project can contain more than one archive. Internal archives and filesets can be directly inside of an archive, or in a sub-folder of that archive.

In the upper right corner of the view you can see an icon which, when clicked, will build the selected top-level archive. Additionally, you can select **Project** → **Build Packages** when a project is selected in the **Packages View** to build all declared packages in that project's `.packages` file. This will execute a full build on all declared archives.

3.2.2. Creating an Archive

When you open the **Project archives** view for the first time, it asks you to select the project for which you want to create an archive.



Please select a project

Figure 3.30. Archives View

When creating a new archive for selected project, you have some different options at your disposal. You need right-click inside the view and select **New Archive** to see your archive type options.

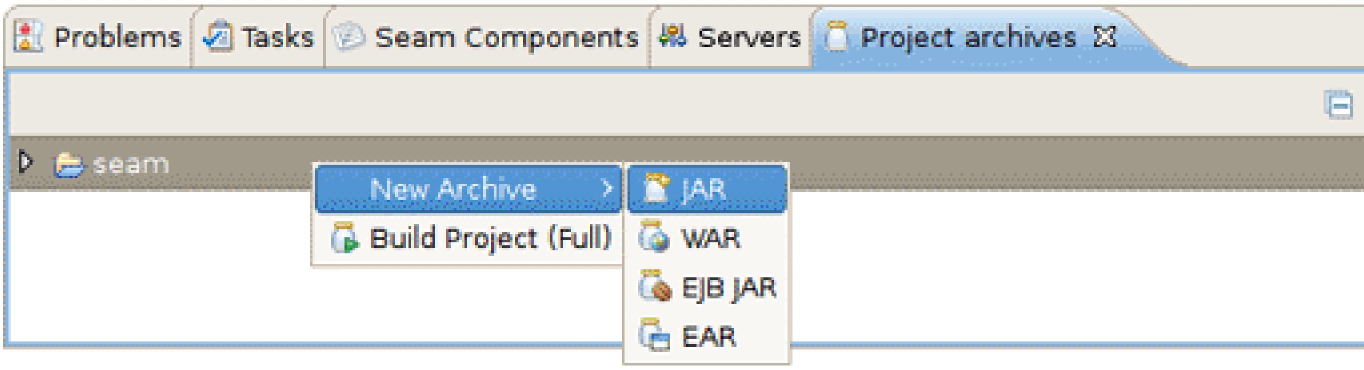


Figure 3.31. Create an Archive

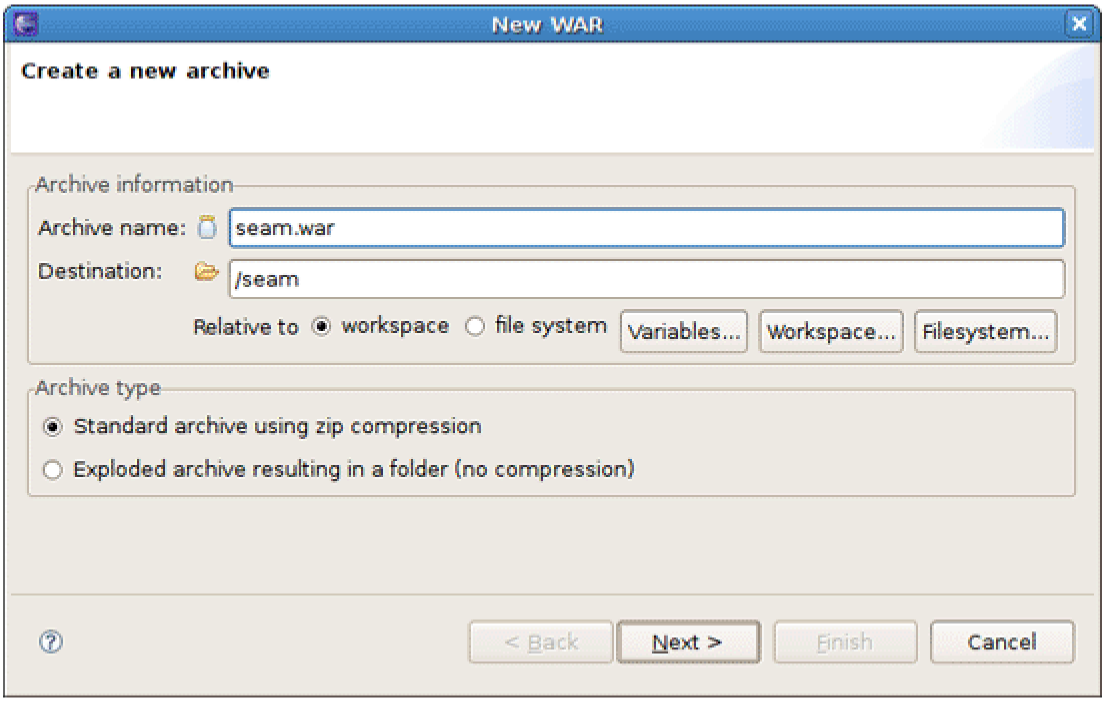
Note:

If you see only JAR from the list of available archive types, you should verify whether AS Tools plugins are installed. EAR, EJB JAR and WAR options are contributed by the AS Tools independently from webtools and the virtual project model. So without them only the JAR option will show up.

`JAR` is the standard archive type and does very little configuration, leaving most of the work up to you. You can customize the name, add folders, filesets and inner JARs to it.

The other types, for the most part, simply start off with a default setting, usually the JAR with some specific children based on an expected structure of the project. For example, if the project is a Dynamic Web Project and you create a `WAR` archive, the archive will be created with a few filesets relevant to the known structure of the project.

Here is the first page of all New archive wizards. It is the same for any archive type and the only page in the **New JAR** wizard.



New WAR

Create a new archive

Archive information

Archive name: seam.war

Destination: /seam

Relative to ● workspace ○ file system

Archive type

● Standard archive using zip compression

○ Exploded archive resulting in a folder (no compression)

Figure 3.32. New WAR Wizard

The page is pretty simple. First it prompts you to set the name of your new archive and a destination.

The destination of an archive can be anywhere on the file system, anywhere in the workspace, inside another archive, or inside a folder declared inside an archive. Select the appropriate checkbox (either **workspace** or **file system**) to specify that the destination is related to either the workspace or filesystem. You can browse to workspace or filesystem destinations by clicking on their respective buttons. To select a destination inside some other archive, you'll need to click the **Workspace** button. At the bottom of the list, you will see the archives that have been declared in the workspace.

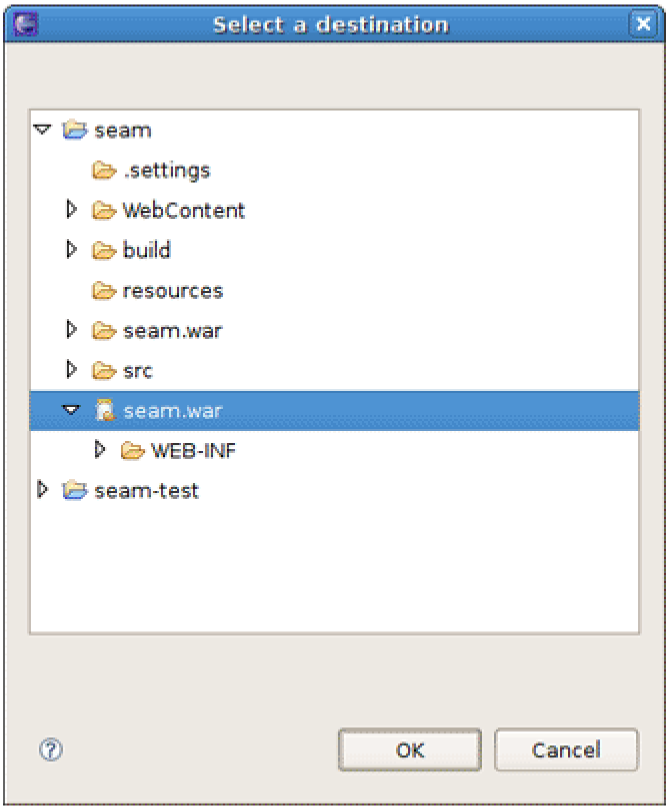
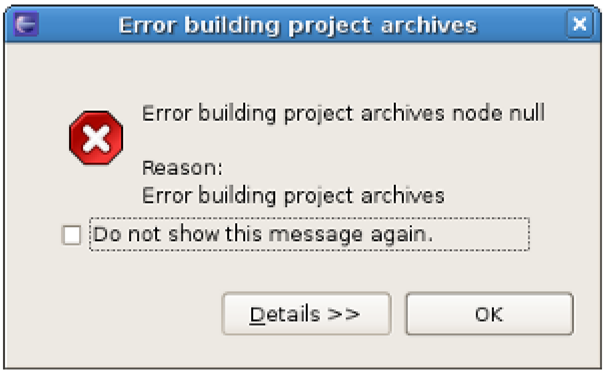


Figure 3.33. Selecting the destination in the workspace

Also in the wizard for creating a new archive you can choose whether an archive to be compressed or exploded into a folder (without compression). You need just select proper checkbox in the **Archive type** section.

If a build or incremental update fails Project Archives will show an error dialog:



Error building project archives node null

Reason:
Error building project archives

□ Do not show this message again.

Figure 3.34. Selecting the destination in the workspace

Click the **Details** button to view detailed information about the cause of the error.

In the **Package Explorer** you can view the created archive.

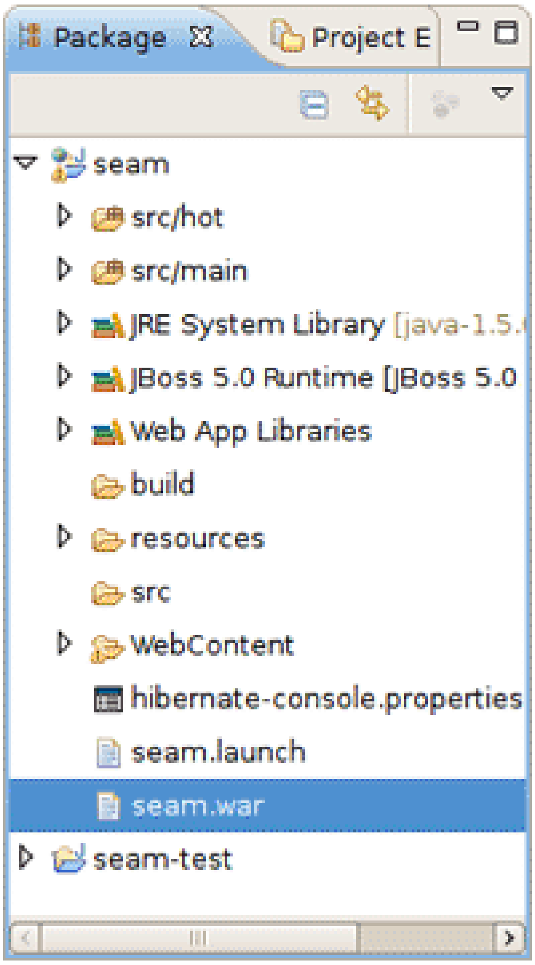


Figure 3.35. The Archive in the Package Explorer

If you use the exploded type of archiving, instead of a single file archive the result put into a folder is displayed in the **Package Explorer**.

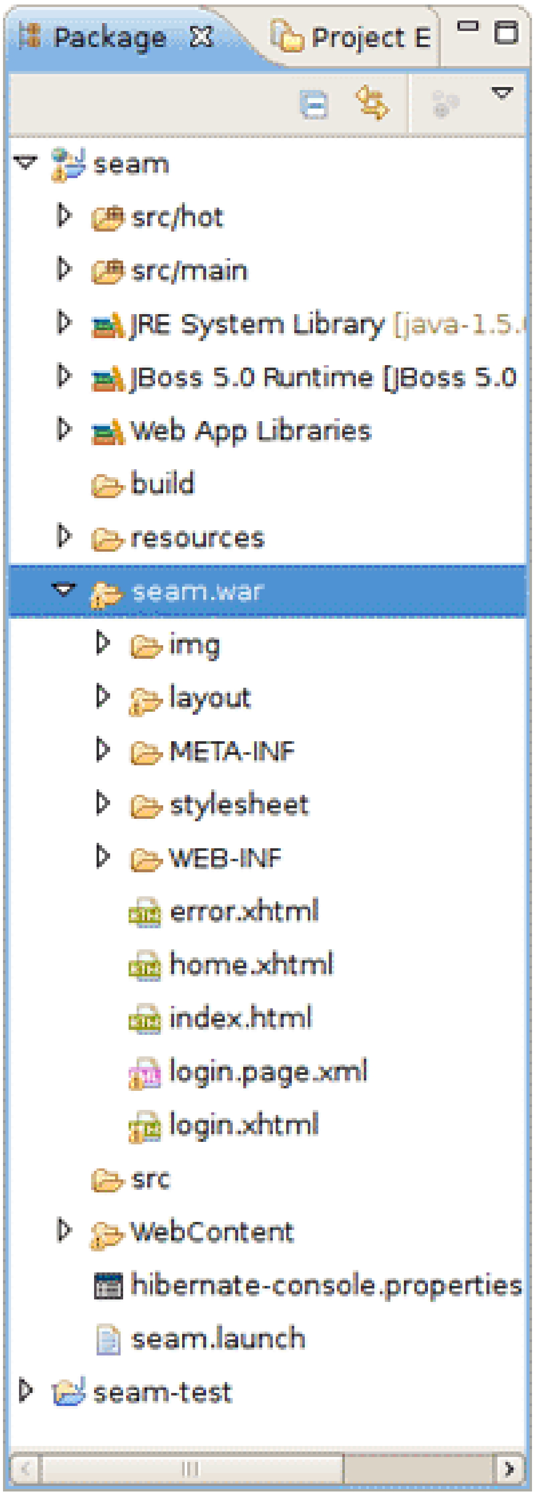


Figure 3.36. The Exploded Archive in the Package Explorer

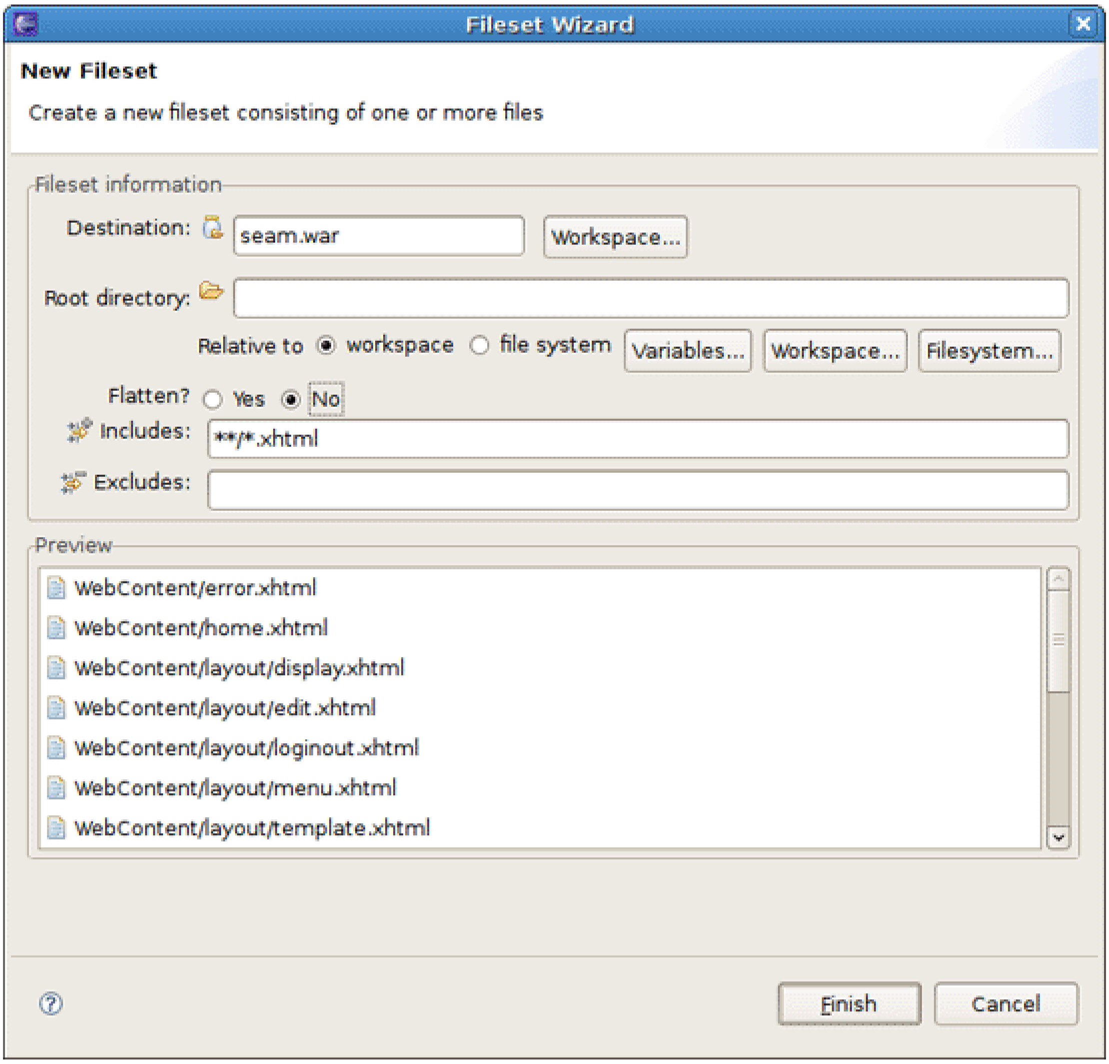
3.2.2.1. Creating a Folder

To create a folder right-click on an archive or folder you want your new folder to be a child of. The only piece of required information the folder name.

3.2.2.2. Creating a FileSet

To create a new fileset, right click on an available target location such as an archive, a nested archive, or a folder within an archive and select the **New Fileset** option.

The **New Fileset** wizard requires a destination (where the files will be located) and a root directory (or where the files are coming from). The source can be anywhere in the workspace or from the filesystem at large.



Fileset Wizard

New Fileset

Create a new fileset consisting of one or more files

Fileset information

Destination: seam.war

Root directory:

Relative to ● workspace ○ file system

Flatten? ○ Yes ● No

Includes: **/*.xhtml

Excludes:

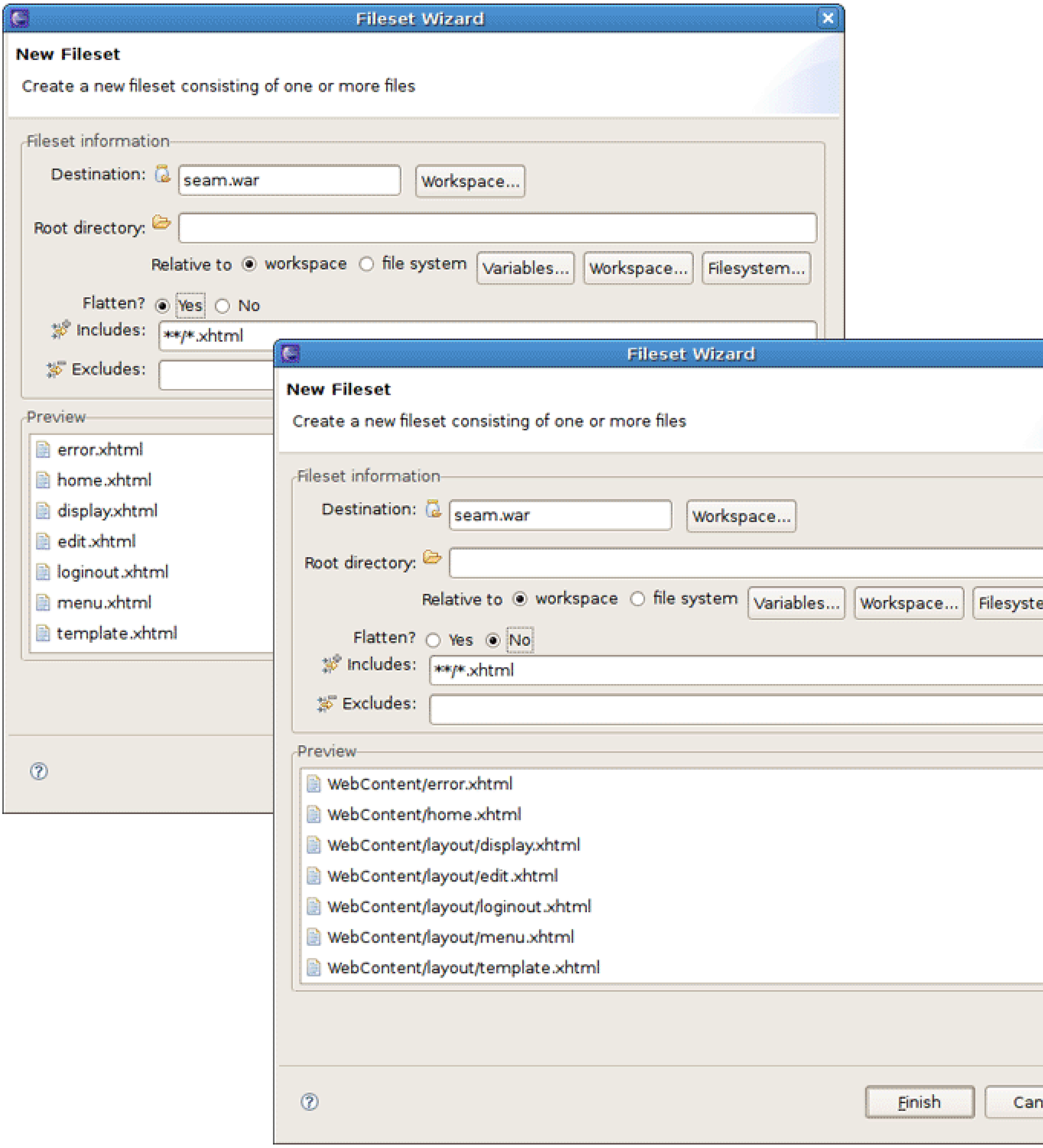
Preview

WebContent/error.xhtml
WebContent/home.xhtml
WebContent/layout/display.xhtml
WebContent/layout/edit.xhtml
WebContent/layout/loginout.xhtml
WebContent/layout/menu.xhtml
WebContent/layout/template.xhtml

Figure 3.37. Adding a New FileSet

Below that, the fileset requires only an **Includes** and **excludes** pattern. As you type in either of these fields, the preview viewer will list those files that are matched.

You can create a Fileset with flattening or without it. Look at the difference on the figure below.



Fileset Wizard

New Fileset

Create a new fileset consisting of one or more files

Fileset information

Destination: seam.war

Root directory:

Relative to ● workspace ○ file system

Flatten? ● Yes ○ No

Includes: **/*.xhtml

Excludes:

Preview

error.xhtml
home.xhtml
display.xhtml
edit.xhtml
loginout.xhtml
menu.xhtml
template.xhtml

Fileset Wizard

New Fileset

Create a new fileset consisting of one or more files

Fileset information

Destination: seam.war

Root directory:

Relative to ● workspace ○ file system

Flatten? ○ Yes ● No

Includes: **/*.xhtml

Excludes:

Preview

WebContent/error.xhtml
WebContent/home.xhtml
WebContent/layout/display.xhtml
WebContent/layout/edit.xhtml
WebContent/layout/loginout.xhtml
WebContent/layout/menu.xhtml
WebContent/layout/template.xhtml

Figure 3.38. The FileSet with flattening and without it

3.2.2.3. Creating User Library FileSet

If you make use of user libraries in your projects you can also refer to these from project archives and have all the JAR and ZIP files they refer included into the archive.

To add a new user libraries file set, right-click on the necessary archive and select the **New User Libraries FileSet** option.

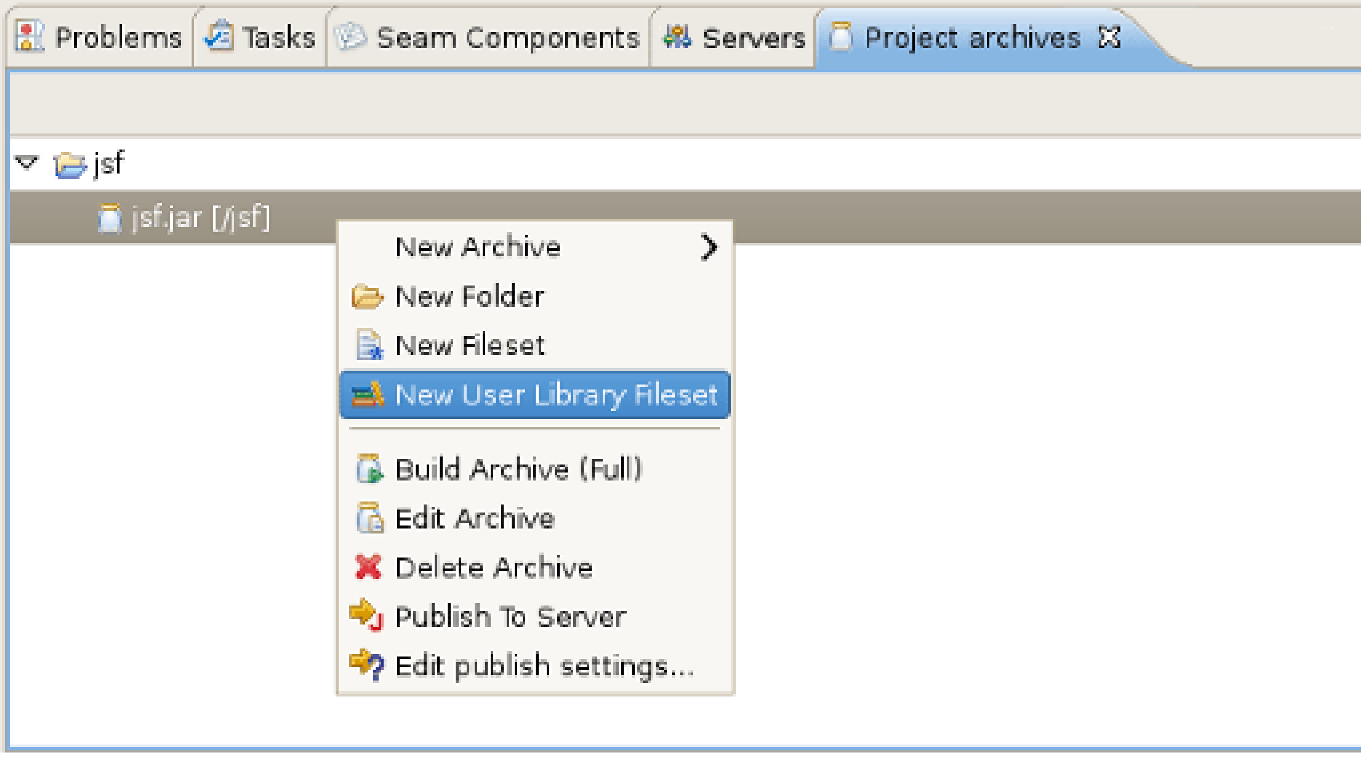


Figure 3.39. Adding New User Library Fileset

You can edit the existing user libraries as well using **User Libraries Fileset** Wizard. Right-click on the library fileset and select the **Edit Fileset** option.

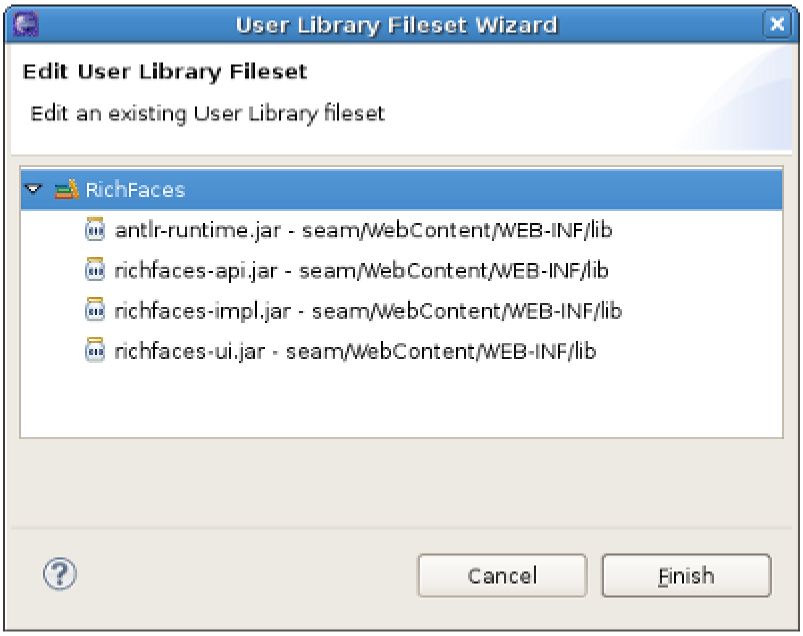

User Library Fileset Wizard
Edit User Library Fileset
Edit an existing User Library fileset
RichFaces
antlr-runtime.jar - seam/WebContent/WEB-INF/lib
richfaces-api.jar - seam/WebContent/WEB-INF/lib
richfaces-impl.jar - seam/WebContent/WEB-INF/lib
richfaces-ui.jar - seam/WebContent/WEB-INF/lib
Cancel
Finish

Figure 3.40. Editing User Library Fileset

3.2.3. Archive Actions

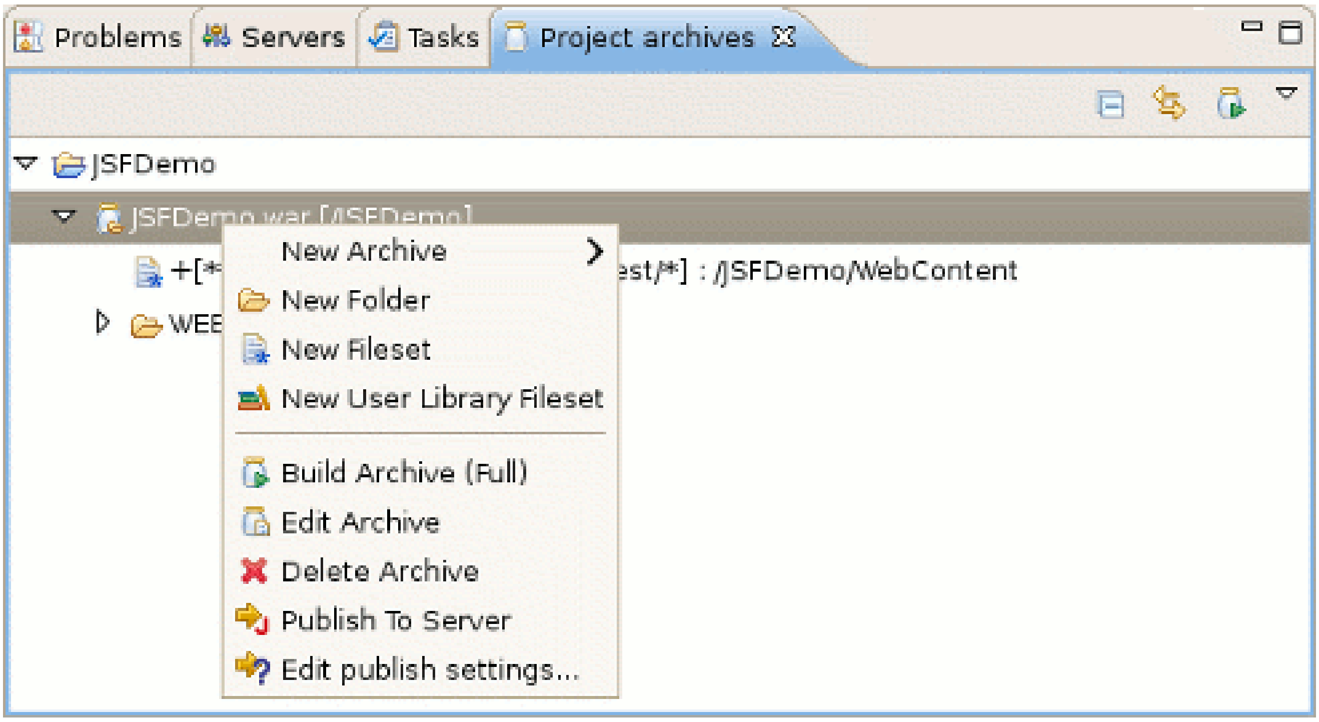


Figure 3.41. Context Menu on the Item

There are a number of variable options in the context menu, but there are also several that come standard.

Table 3.4. Context Menu on the Item

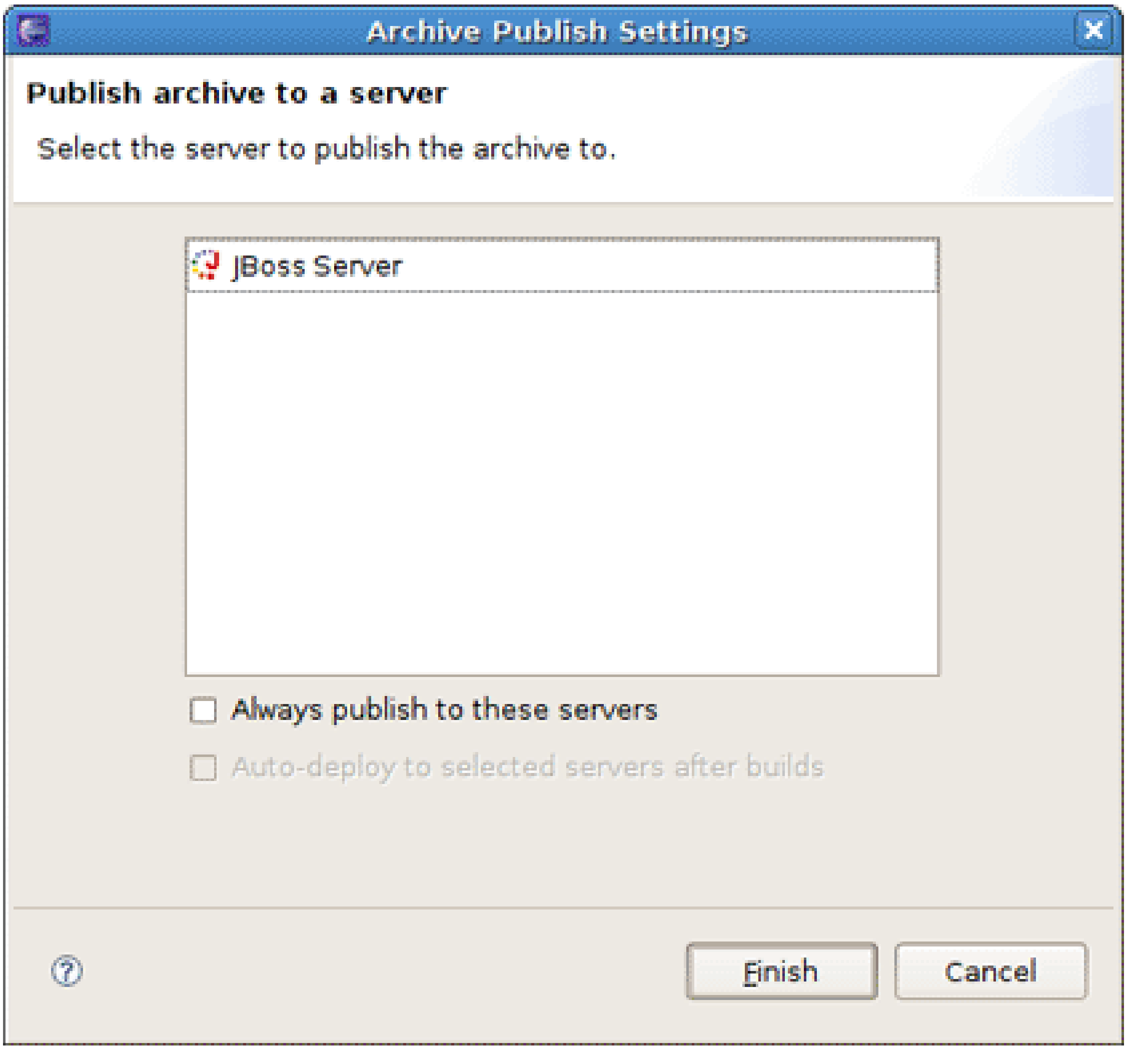
Name	Description
Build Archive (Full)	This action is enabled only on top-level archives and initiates a full build on that archive
Edit Archive	Standard action that brings up the wizard associated with that particular node type and allows the details to be changed
Delete Archive	This option deletes the selected node
Publish To Server	This action will publish to a declared server
Edit publish settings	This option edits the archives publish settings

Note:

When editing an archive, it is also updated in all folders and other archives where it is nested.

3.2.4. Publishing to Server

Finally, you will need to publish your application to a server. This section describes how to do it with the help of the **Archives** View.



Archive Publish Settings

Publish archive to a server

Select the server to publish the archive to.

JBoss Server

□ Always publish to these servers

□ Auto-deploy to selected servers after builds

Figure 3.42. Context Menu on the Item

The dialog shown above appears after selecting the **Publish To Server** option. To publish once, select the server(s) that you want and click the **Finish** button. If you want the **Publish to Server** action on that particular Archive to always publish to that set of servers, then check the appropriate checkbox. To enable automatic publishing upon build events, check the last checkbox.

The automatic publishing feature is nice if, for example, your package's destination (where it is built) is a temporary folder and you want the archive published to several servers. If you only need your archive published to one server, it might be easier to have the archive's destination folder be the deploy folder of the server.

3.2.5. Relevant Resources Links

Refer to the *Ant manual* [http://ant.apache.org/manual/index.html] to find more on how to build your applications using Ant.

We also recommend that you watch this *movie* [http://docs.jboss.org/tools/movies/demos/archiving/archiving.htm] which demonstrates the powerful archiving functionality in JBoss Tools™.

This chapter has covered the functionality provided by the JBoss AS perspective. The next chapter will explore working with different kinds of projects.

Projects

The most popular of the projects we deal with are the J2EE ones, such as Dynamic Web Project, EJB Project, or EAR project. JBoss Tools™ web projects include Struts, JSF and Seam projects. These are referred to as faceted projects. This chapter will cover facets, which are used to provide a consistent structure and packaging features to any type of project.

4.1. Faceted Projects Overview

The idea behind faceted projects is that each project can accept units of functionality, or facets, which can be added or removed by the user. These facets either add to the project's classpath, enable a builder, or watch the project in some other fashion. Typically every project concerned has at least one facet when it is created. As an example, a Web project has a WebDoclet facet, or an EJB Project has an EJB Module facet as prerequisites.

WTP projects have been criticized for being over-engineered or too restrictive in their design. WTP projects are set up in a tree-relationship to each other, where one project can be a child of another. For example, an EAR project may have a Web Project child, an EJB project child, or other types.

However, the benefit of this is that the structure of your projects is then known and packaging it up *should* be trivial. If your project is non-standard, or you feel too confined by such rigid structural requirements, you can still choose to package your project using the Archives plugin (see *Section 3.2, “Project Archives View”*).

4.2. Adding Facets to a Project

This section will cover the facets added by JBoss Tools and show how you can configure them in a project by adding new ones or modifying existing facet configurations.

One way to configure the facets is doing it while organizing a new project. To demonstrate this let's create a new **Dynamic Web Project** by selecting **File** → **New** → **Other...** → **Web** → **Dynamic Web Project**.

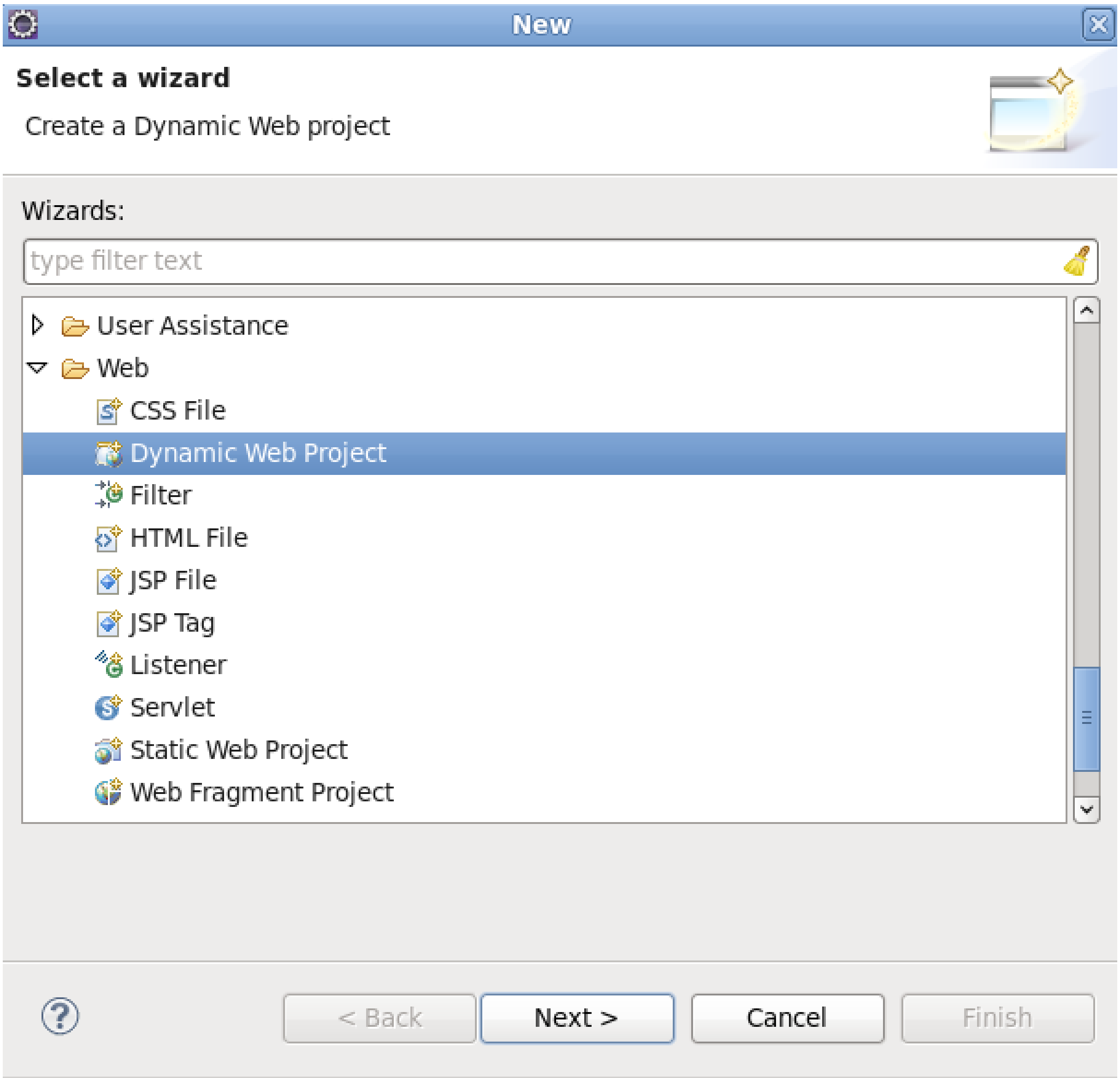
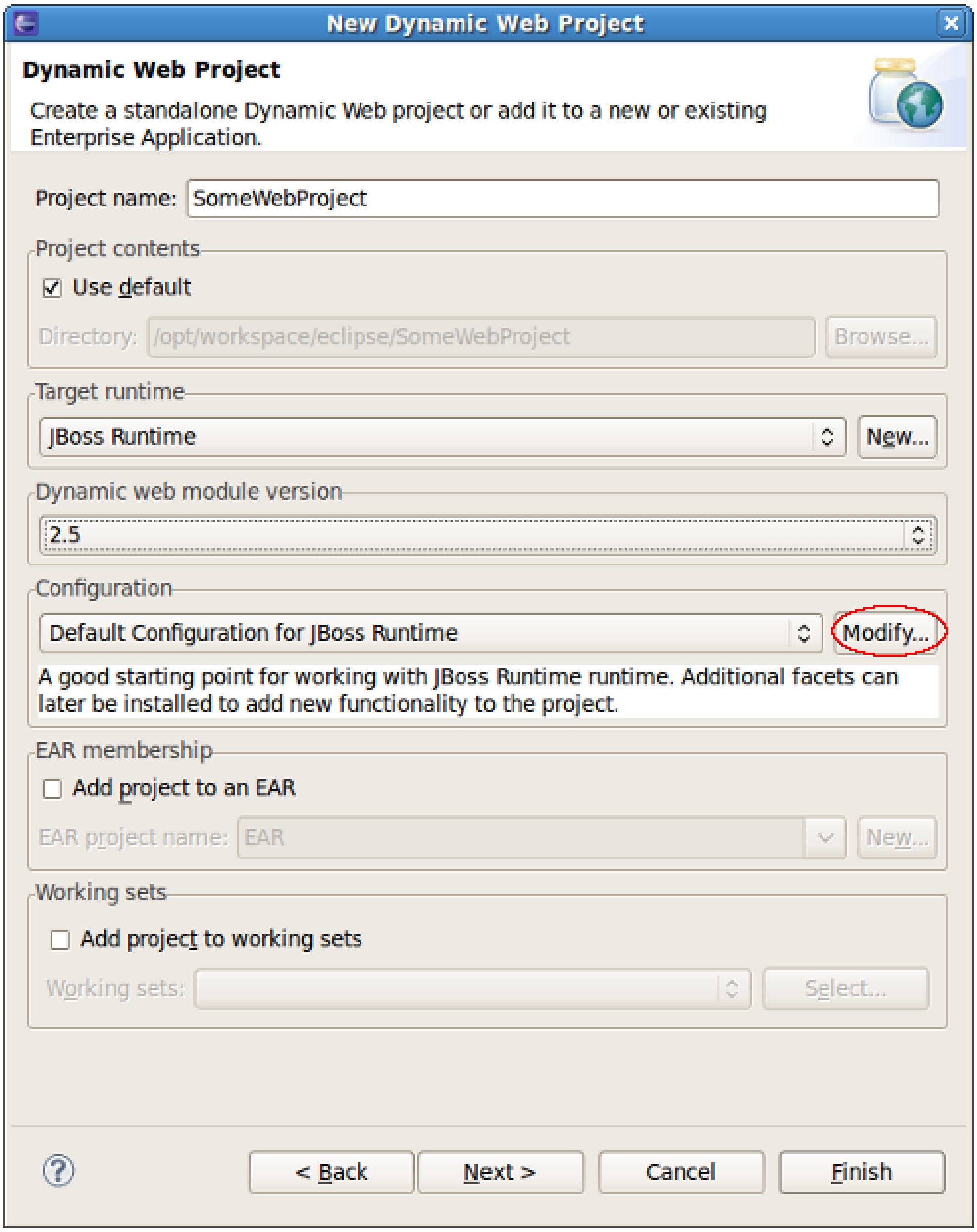


Figure 4.1. New Dynamic Web Project

Click the **Next** button and you will see a Dynamic Web Project page like on the figure below.

The first page of most WTP projects allows you to target a specific runtime, which represents a server's library location. It will also provide you the ability to add this project to an EAR project and select a preselected default set of facets, called a configuration, rather than manually select each required facet.

Selecting the runtime allows the project to install the proper classpaths to the project so it knows what code to compile against.



New Dynamic Web Project

Dynamic Web Project

Create a standalone Dynamic Web project or add it to a new or existing Enterprise Application.

Project name: SomeWebProject

Project contents

☑ Use default

Directory: /opt/workspace/eclipse/SomeWebProject

Target runtime: JBoss Runtime

Dynamic web module version: 2.5

Configuration: Default Configuration for JBoss Runtime

A good starting point for working with JBoss Runtime runtime. Additional facets can later be installed to add new functionality to the project.

EAR membership

☐ Add project to an EAR

EAR project name: EAR

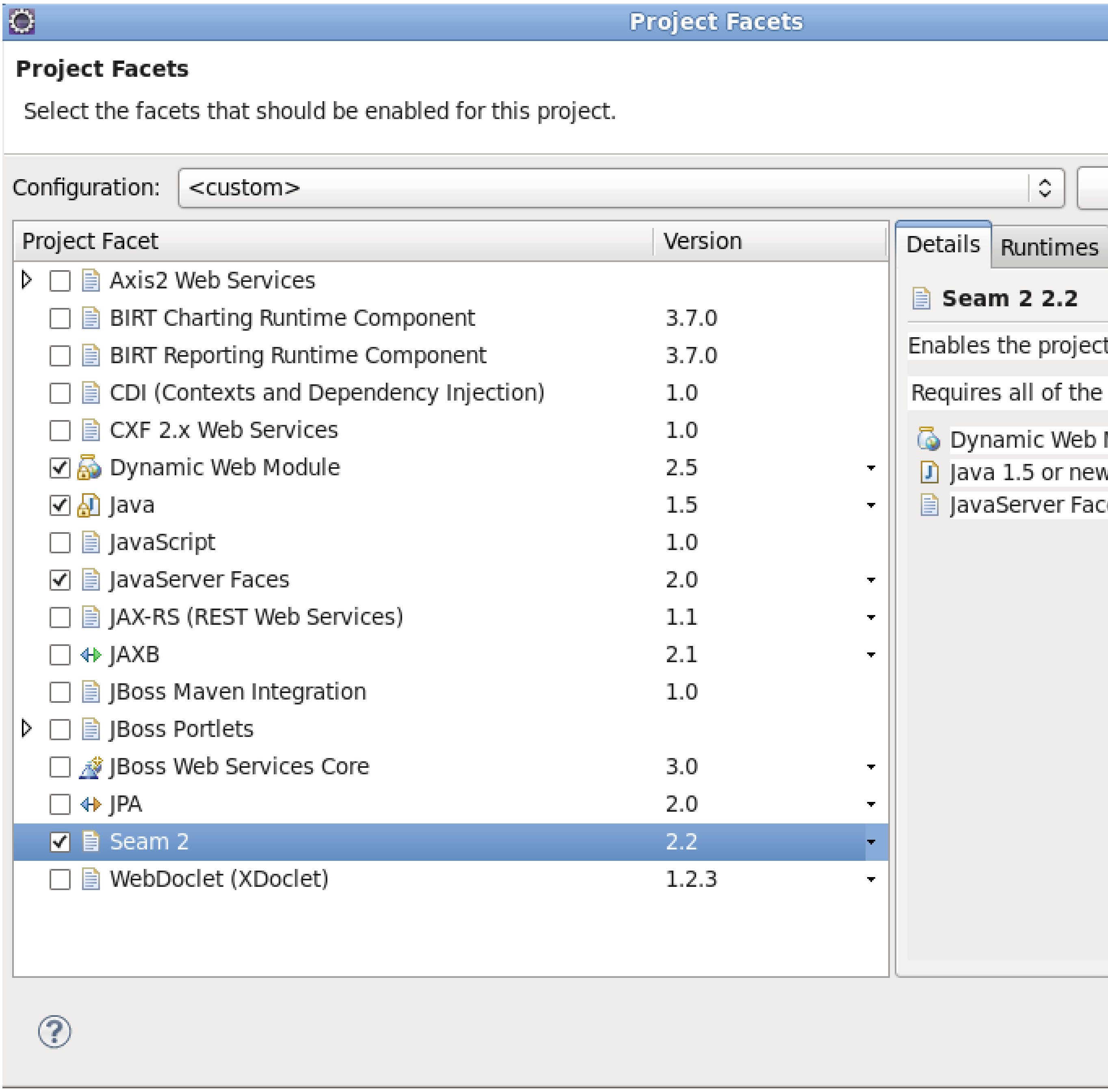
Working sets

☐ Add project to working sets

Working sets:

Figure 4.2. New Dynamic Web Project

Click the **Modify** button next to the **Configuration** section to open a wizard which allows you to modify the chosen configuration. The wizard is shown in the image below.



Project Facets

Project Facets

Select the facets that should be enabled for this project.

Configuration: <custom>

	Project Facet	Version
▷ ☐	Axis2 Web Services	
☐	BIRT Charting Runtime Component	3.7.0
☐	BIRT Reporting Runtime Component	3.7.0
☐	CDI (Contexts and Dependency Injection)	1.0
☐	CXF 2.x Web Services	1.0
☑	Dynamic Web Module	2.5
☑	Java	1.5
☐	JavaScript	1.0
☑	JavaServer Faces	2.0
☐	JAX-RS (REST Web Services)	1.1
☐	JAXB	2.1
☐	JBoss Maven Integration	1.0
▷ ☐	JBoss Portlets	
☐	JBoss Web Services Core	3.0
☐	JPA	2.0
☑	Seam 2	2.2
☐	WebDoclet (XDoclet)	1.2.3

Seam 2 2.2

Enables the project

Requires all of the

Dynamic Web M

Java 1.5 or new

JavaServer Fac

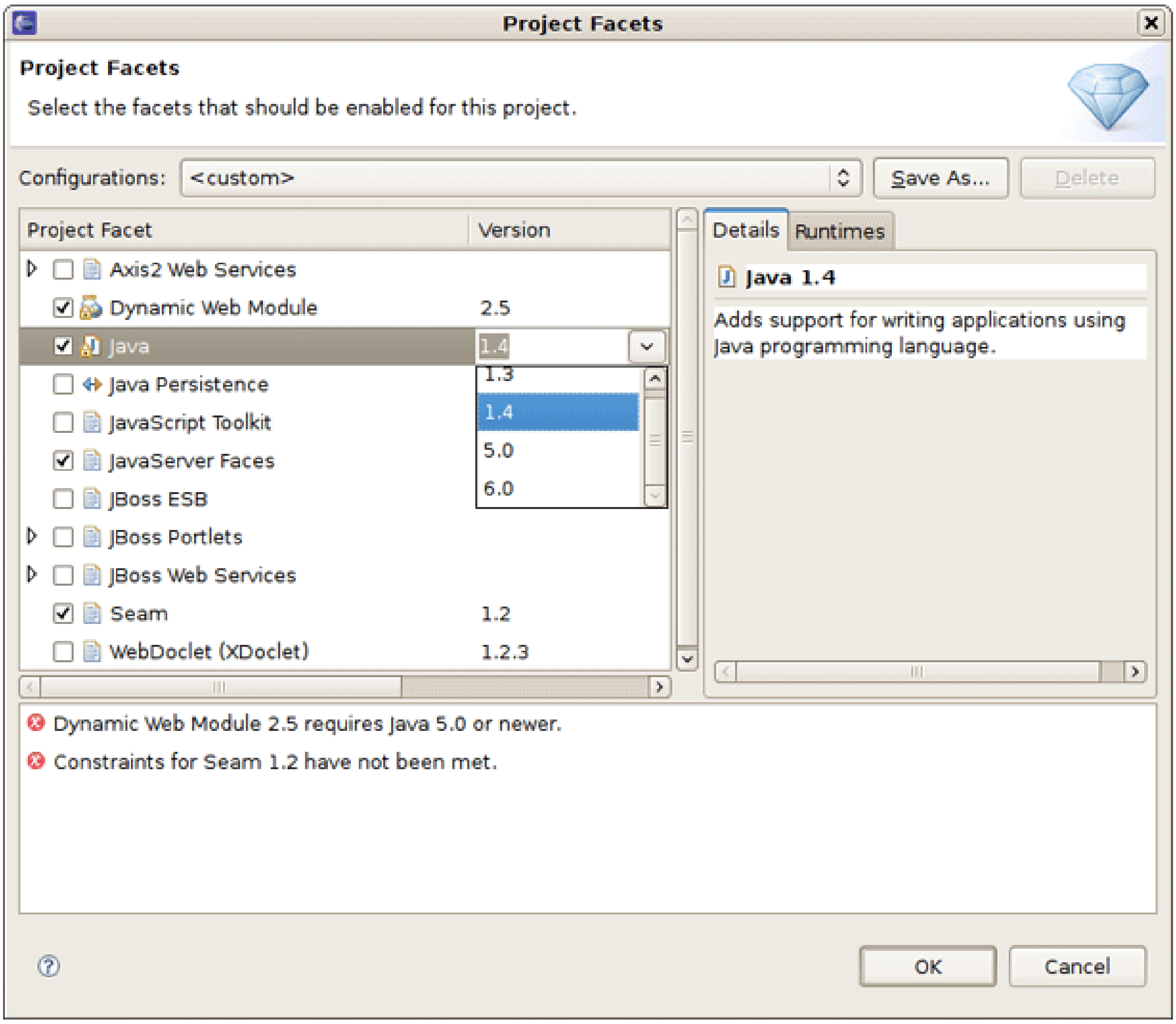
Figure 4.3. Project Facets Wizard

Here part of the listed facets are those which are provided by WTP. Some of them are added by JBoss Tools. They are:

- BIRT Charting Runtime Component
- BIRT Reporting Runtime Component
- CDI (Contexts and Dependency Injection)

- CXF 2.x Web Services
- JAX-RS (REST Web Services)
- JAXB
- JBoss Portlets
- JBoss Web Services Core
- JPA
- Seam 2

On this wizard page you can enable or disable any facet as well as change its version. What you should note here is that some facets or facets versions may conflict with each other. In case of incompatibility you will be notified in the combobox underneath.



Project Facets

Select the facets that should be enabled for this project.

Configurations: <custom>

Project Facet	Version
☐ Axis2 Web Services	
☑ Dynamic Web Module	2.5
☑ Java	1.4
☐ Java Persistence	
☐ JavaScript Toolkit	
☑ JavaServer Faces	
☐ JBoss ESB	
☐ JBoss Portlets	
☐ JBoss Web Services	
☑ Seam	1.2
☐ WebDoclet (XDoclet)	1.2.3

1.3
1.4
5.0
6.0

Java 1.4

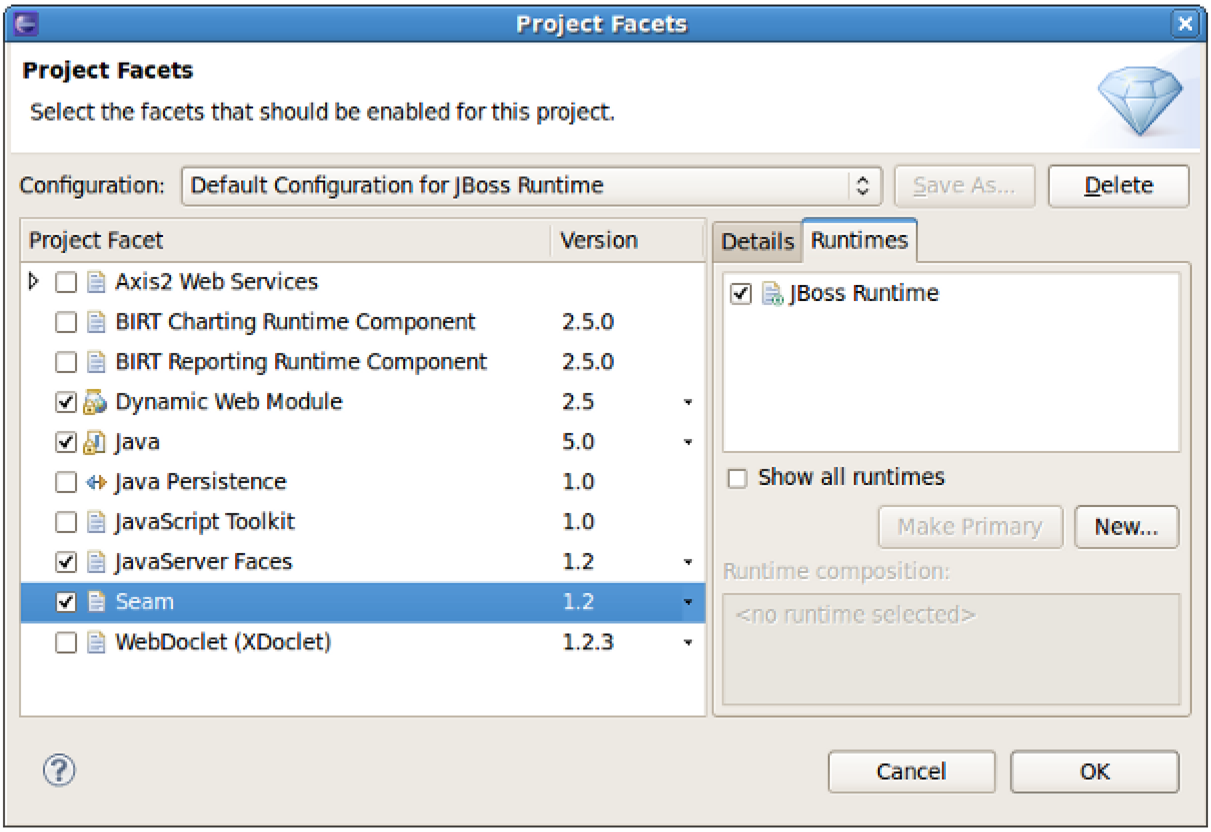
Adds support for writing applications using Java programming language.

Dynamic Web Module 2.5 requires Java 5.0 or newer.

Constraints for Seam 1.2 have not been met.

Figure 4.4. Facet Constraints

When switching on the **Runtimes** tab on the right you will see the current server Runtime.



Project Facets

Select the facets that should be enabled for this project.

Configuration: Default Configuration for JBoss Runtime

Project Facet	Version
□ Axis2 Web Services	
□ BIRT Charting Runtime Component	2.5.0
□ BIRT Reporting Runtime Component	2.5.0
☑ Dynamic Web Module	2.5
☑ Java	5.0
□ Java Persistence	1.0
□ JavaScript Toolkit	1.0
☑ JavaServer Faces	1.2
☑ Seam	1.2
□ WebDoclet (XDoclet)	1.2.3

☑ JBoss Runtime

□ Show all runtimes

Runtime composition:

<no runtime selected>

Figure 4.5. Runtimes on the Project Facets Wizard

On this tab you can also create a new Server Runtime and make it primary by enabling it and then clicking the **Make Primary** button.

Clicking on the **OK** button will save the chosen configuration of the facets and return you to the Dynamic Web Project wizard (see *Figure 4.2, “New Dynamic Web Project”*). Additional pages in the wizard are specific to either the project type or the facets selected.

If you need to configure the facets for an existing project, right click on the project, select **Properties** and then select **Project Facets**. This will bring up the Project Facets wizard (see *Figure 4.3, “Project Facets Wizard”*), where you can create your own custom facets configuration.

4.3. Relevant Resources Links

More information on the WTP facets can be found in the *Eclipse help* [http://help.eclipse.org/ganymede/index.jsp?topic=/org.eclipse.jst.j2ee.doc.user/topics/cfacets.html].

Deploying Modules

In this chapter it will be described how to deploy modules onto the server.

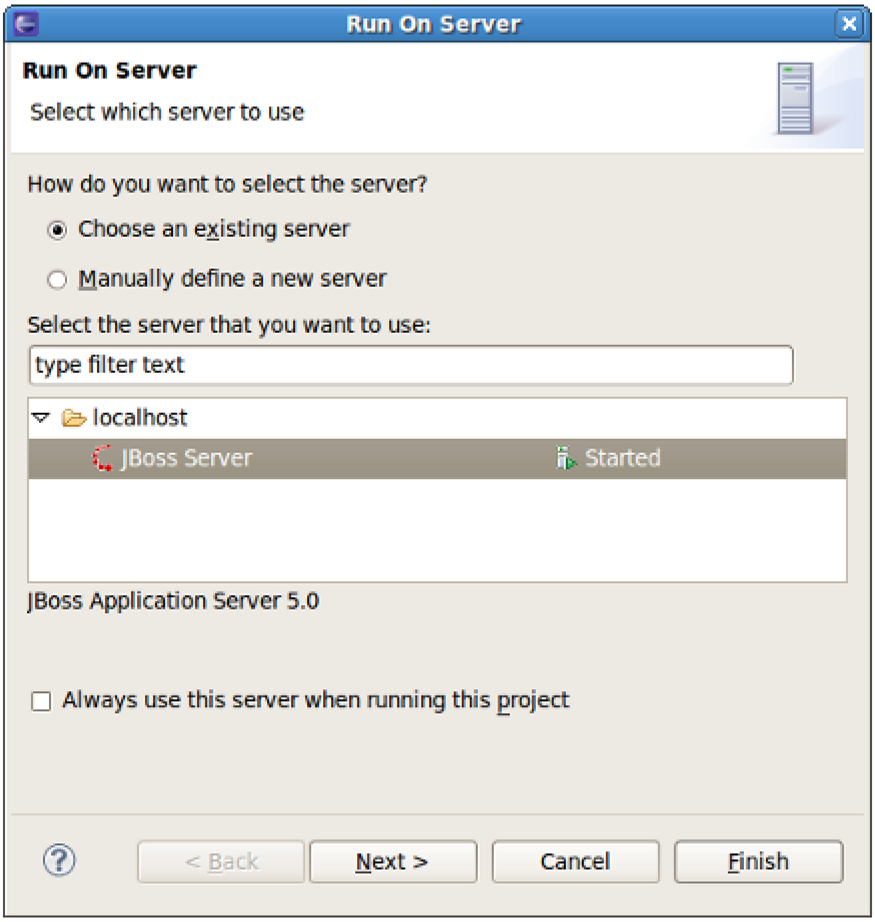
First of all it is necessary to say that deploying to a server is mostly painless. There are several ways to do it provided by WTP and some additional methods provided by JBoss Tools. These methods are described further in this chapter.

5.1. Deploying on the Package Explorer

On the package explorer it is possible to publish either a project to a server or just a single file. Let's look at how to do this.

5.1.1. Deploying with Run On Server Wizard

The first WTP method is to right-click on a project, such as a Dynamic Web project, EJB project, or EAR project and then select **Run As** → **Run on Server**. The resulting dialog allows you to select which supporting server the project can be published to.



Run On Server

Run On Server

Select which server to use

How do you want to select the server?

● Choose an existing server

○ Manually define a new server

Select the server that you want to use:

type filter text

localhost

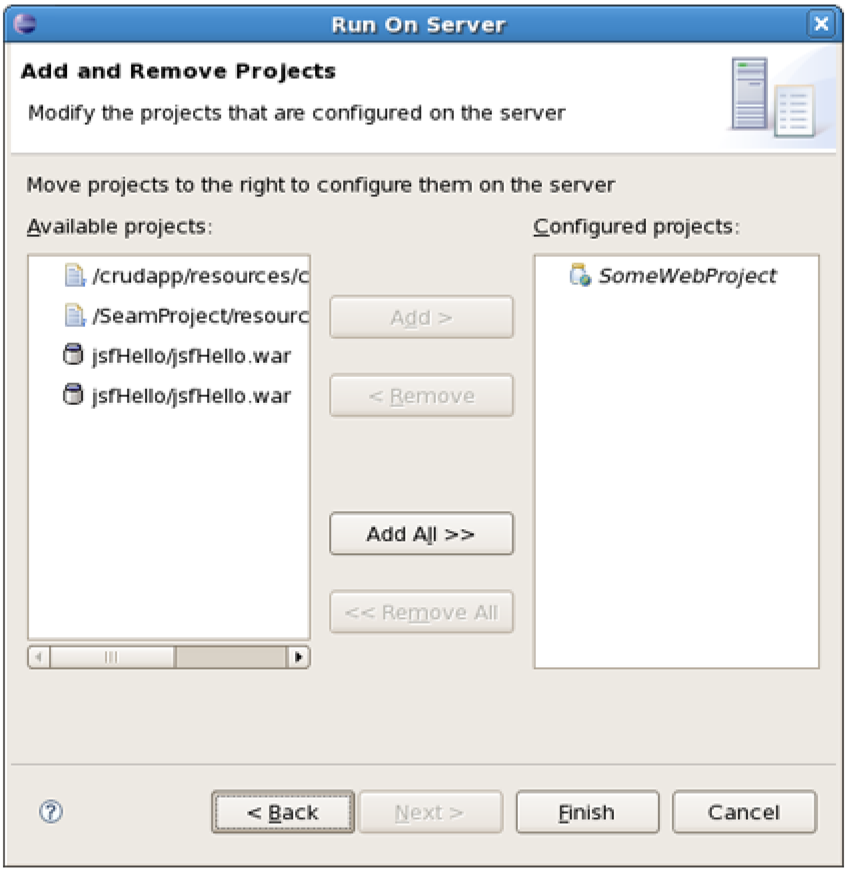
JBoss Server Started

JBoss Application Server 5.0

□ Always use this server when running this project

Figure 5.1. Define a New Server

Click the **Next** button to see add or remove projects page where you can choose projects to configure them on server.



Run On Server

Add and Remove Projects

Modify the projects that are configured on the server

Move projects to the right to configure them on the server

Available projects:

- /crudapp/resources/c
- /SeamProject/resourc
- jsfHello/jsfHello.war
- jsfHello/jsfHello.war

Configured projects:

- SomeWebProject

Figure 5.2. Add or Remove Projects

This page of the wizard also allows to undeploy modules from the server. For that choose proper module(s) from the right and click the **< Remove**. The modules will be completely undeployed after restarting your server or republishing.

Generally, for the JBoss AS Server Adapters, publishing using this method will force a default, best-guess, packaging configuration for your project. This best-guess does not publish incrementally, but instead repackages your entire project into a `.war`, `.jar`, or `.ear` as appropriate and then copies that file into the proper deploy directory. For quicker smarter deployment, you will need to create archives using the Project Archives view (see *Section 3.2, “Project Archives View”*) and customize packaging yourself.

5.2. Deploying with Servers View

As it has been already mentioned **Servers** view contains two parts: the top part that displays all defined servers and the bottom part which provides categories with additional information. Thus, in this section we suggest two more ways to deploy resources onto the server.

5.2.1. Top part of Servers view

In the top part of the **Servers** view you should right click on a server and select the **Add and Remove** menu item.

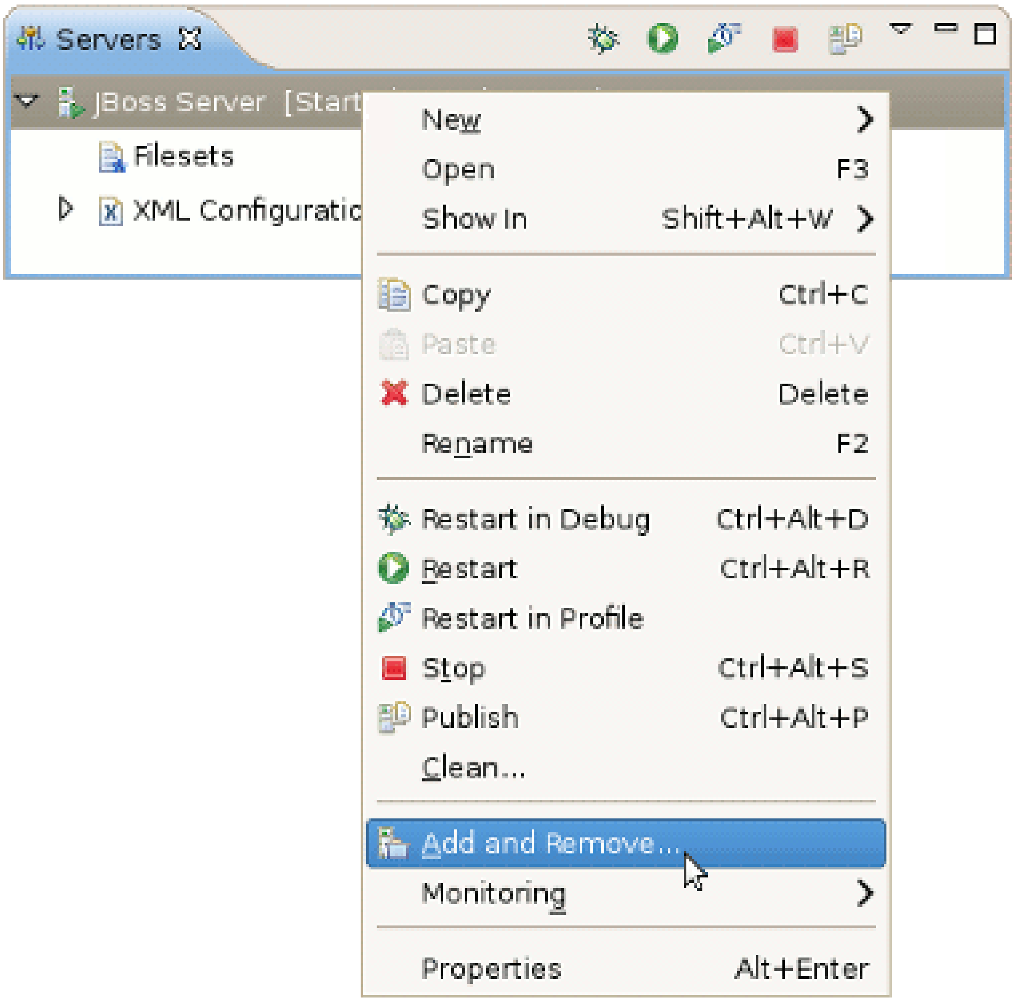


Figure 5.3. Add and Remove Projects

This will bring up a dialog (see *Figure 5.2, “Add or Remove Projects”*) that allows you to either publish projects or modules to a server, or remove them from the server. If the selected module is a project like a Dynamic Web project, EJB project, or EAR project, it will be published as through **Run on Server** wizard, with a best-guess full package. If, however, the selected element is an archive from the Project Archives view (see *Section 3.2, “Project Archives View”*), it will be published according to the rules of that module type.

5.2.2. Bottom part of Servers view

In the bottom part of **Servers** view there is a category called **Modules** which should display all currently-published modules on the server. Right-clicking on the desired module and selecting **Full Publish** will force a full rebuild of the entire module.

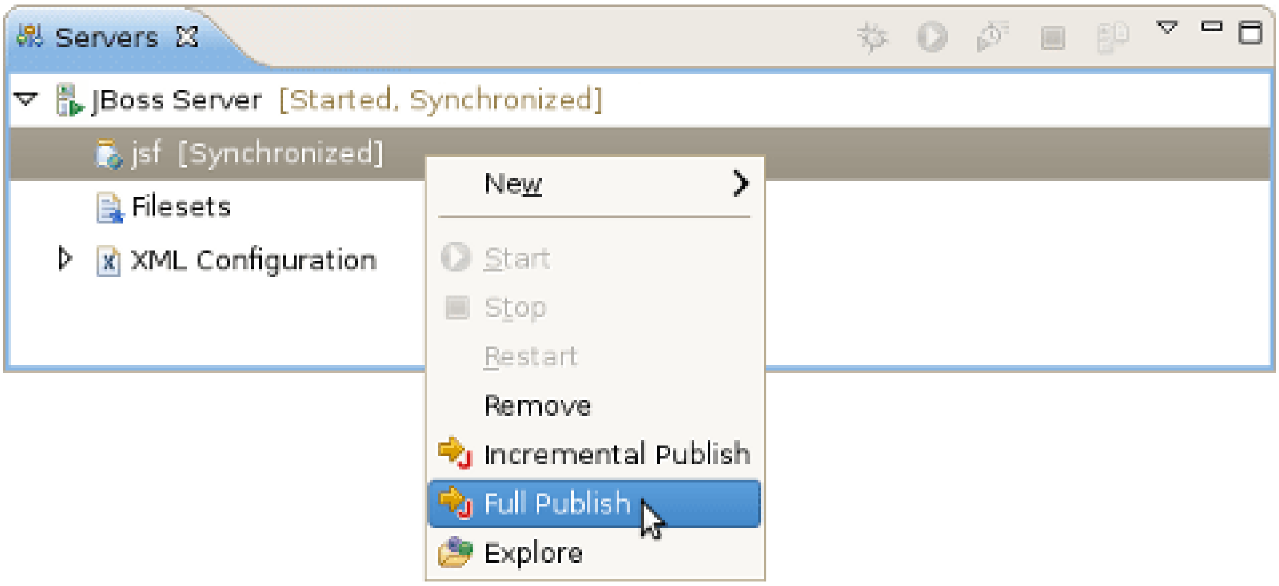


Figure 5.4. Full Publish

Here, **Incremental Publish** is meant to enable publishing of only those parts where changes have been made.

5.3. Deploying with Project Archives View

In the **Project Archives View** you can right-click on any declared archive and select the **Publish To Server** element. For more on this subject, see *Section 3.2.4, "Publishing to Server"* in the Project Archives View section.

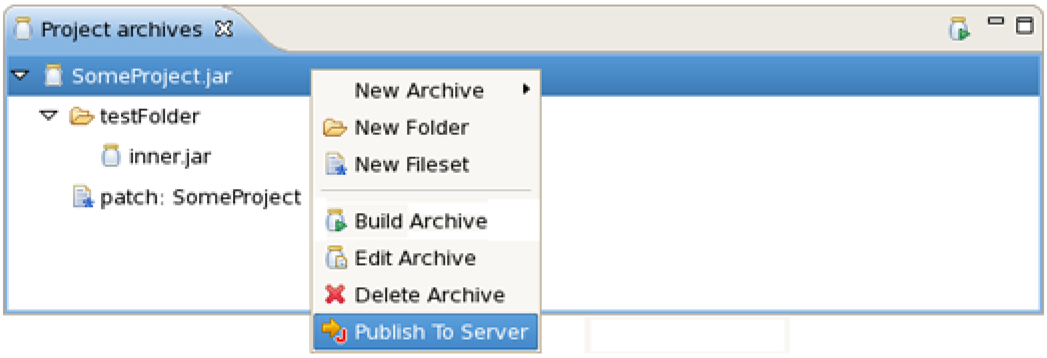


Figure 5.5. Publish to Server

The only way to ensure an **Incremental Build**, such as changes to one `.jsp`, `.html`, or `.class` file, is to enable the builder for that project. This is done by either changing the global preferences for the **Archives View**, or by enabling project-specific preferences and ensuring the builder is on.

You can also use the "Finger touch" button for a quick restart of the project without restarting the server:

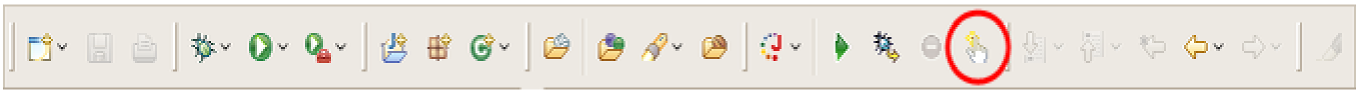


Figure 5.6. Finger Touch button

The "Finger" touches descriptors dependent on project (i.e. `web.xml` for WAR, `application.xml` for EAR, `jboss-esb.xml` in ESB projects).

The last chapter covers a variety of methods on how you can deploy needed modules onto a server.

TPTP Support

This chapter provides an overview on how to enable TPTP Profiling for JBoss AS™ adapters in JBoss Tools™.

6.1. TPTP Profiling

To get TPTP profiling work on JBoss Application Server™ you should do the following:

- Download *TPTP Runtime* [http://www.eclipse.org/tptp/home/downloads/] and install it, i. e. just add the content of `plugins/features` folders from downloaded directory to the same folders in your eclipse installation directory or use the **Help** → **Install New Software** command.
- Install JBoss TPTP Tools which provide TPTP support for JBoss AS servers (find the latest stable version of the JBoss TPTP profile feature at *http://www.jboss.org/tools/download/stable*).

And now all profile actions should work for you. To start JBoss AS™ in profiling mode use **Start the server in profiling mode** button or select **Profile As** → **Profile on Server** from the context menu of the project.

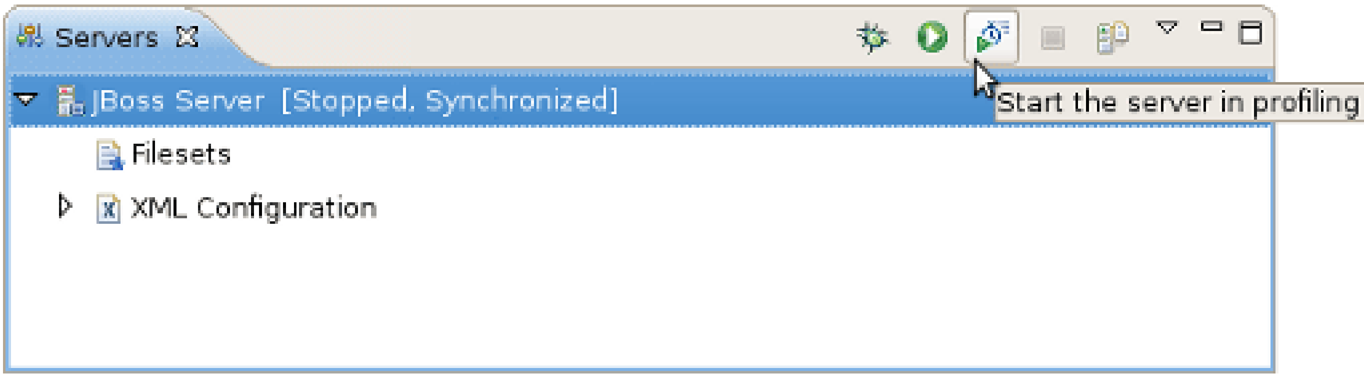

Servers
JBoss Server [Stopped, Synchronized]
Filesets
XML Configuration
Start the server in profiling

Figure 6.1. Start the Server in Profiling mode

To enable TPTP features in your workbench use Profiling and Logging Perspective that you can find in the list of proposed perspectives: **Window** → **Open Perspective** → **Other...**

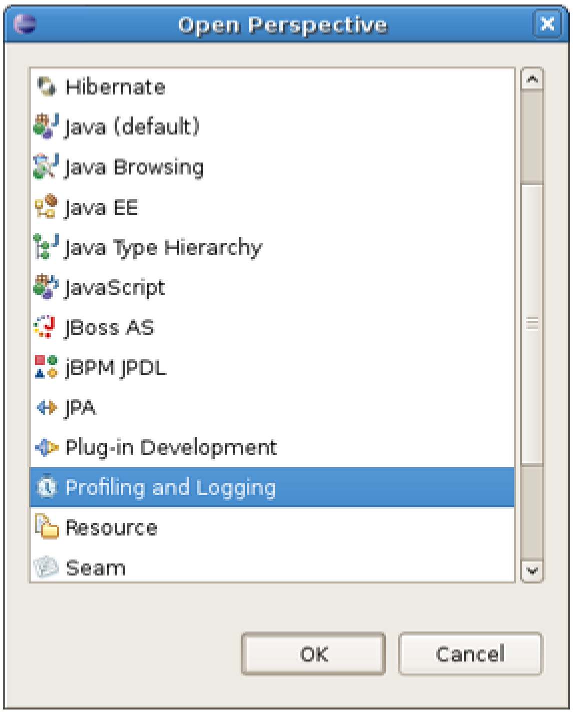


Figure 6.2. Profiling and Logging Perspective

6.2. Relevant Resources Links

All additional information on TPTP (Test and Performance Tools Platform) can be found in the *Eclipse documentation* [http://www.eclipse.org/tptp/home/downloads/4.5.0/documents/quicktour/quick_tour.html].