JBoss Server Manager Reference Guide

Version: 2.1.0.GA

1. Quick Start with JBoss Server 1
1.1. Key Features of JBoss Server 1
1.2. Starting JBoss Server 1
1.3. Stopping JBoss Server 2
1.4. Project Archiving 2
1.5. Deploying an Application to a Server 4
1.6. Publishing to JBoss Server 8
1.7. Other relevant resources on the topic 8
2. Runtimes and Servers in the JBoss AS plugin 10
2.1. Runtimes 10
2.1.1. Installing a New Runtime 10
2.1.2. Copying the existing runtime configuration 14
2.2. Servers 16
2.2.1. Creating a New Server 16
3. JBoss AS Perspective 20
3.1. The JBoss Server View 20
3.1.1. JBoss Server View Toolbar 20
3.1.2. JBoss Server View Structure 21
3.1.3. Drag-n-Drop to JBoss Server View 31
3.1.4. Server Log View 32
3.1.5. Server Editor 32
3.1.6. Relevant Resources Links 37
3.2. Project Archives View 38
3.2.1. Overview 38
3.2.2. Creating an Archive 39
3.2.3. Archive Actions 47
3.2.4. Publishing to Server 47
3.2.5. Relevant Resources Links 48
4. Projects 49
4.1. Faceted Projects Overview 49
4.2. Adding Facets to a Project 49
4.3. Relevant Resources Links 54
5. Deploying Modules 55
5.1. Deploying on the Package Explorer 55
5.1.1. Deploying with Run On Server Wizard 55
5.1.2. Deploying single files 56
5.2. Deploying with JBoss Server View 58
5.2.1. Top part of JBoss Server View 58
5.2.2. Bottom part of JBoss Server View 59
5.3. Deploying with Project Archives View 60
6. TPTP Support 62
6.1. TPTP Profiling 62
6.2. Relevant Resources Links 63

Quick Start with JBoss Server

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

1.1. Key Features of JBoss Server

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

Table 1.1. Key Functionality for JBoss AS and Archive Tools

Feature	Benefit	Chapter
JBoss AS plugin	Using WTP, JBoss AS plugin allows working 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 makes managing installed JBoss Server quite easy and includes the standard Console and Properties views, and specially added Project archives View and JBoss Server View for that purpose.	*JBoss AS perspective*
Modules Deployment	A number of ways, provided by WTP and JBoss Tools, to deploy either a project or a single file on the server.	*deploying modules*
TPTP Support	Test and Performance Tools Platform (TPTP) profiling.	*TPTP support*

If you already have installed JBoss server and runtime you can quickly learn how to configure, start, stop the server, to know deployment and archiving processes. How to install runtimes and servers read in the *Runtimes and Servers in the JBoss AS plugin* chapter.

To start working with JBoss AS, select a JBoss AS Perspective via *Window > Open Perspective > Other > JBoss AS*.

1.2. Starting JBoss Server

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

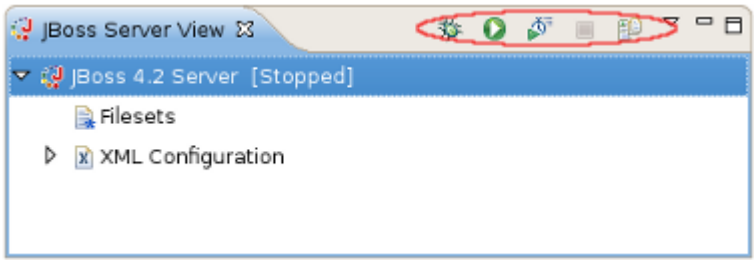

Figure 1.1. JBoss Server Toolbar

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

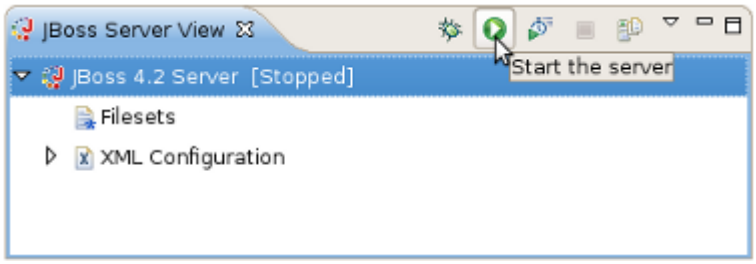

Figure 1.2. Start JBoss Server

1.3. Stopping JBoss Server

To stop the server, click the *Stop* icon in the JBoss Server View or right click the server name and press *Stop*.

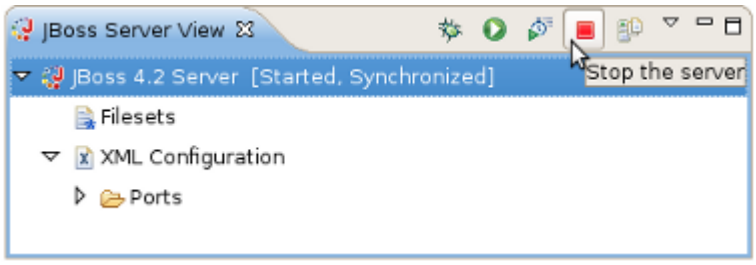

Figure 1.3. Stop JBoss Server

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

Learn more about the JBoss Server View *here*.

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 Tools > Project archives)*.

Right clicking in the Project archives view you can create 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 archive, *edit*, *delete* or *publish* it.

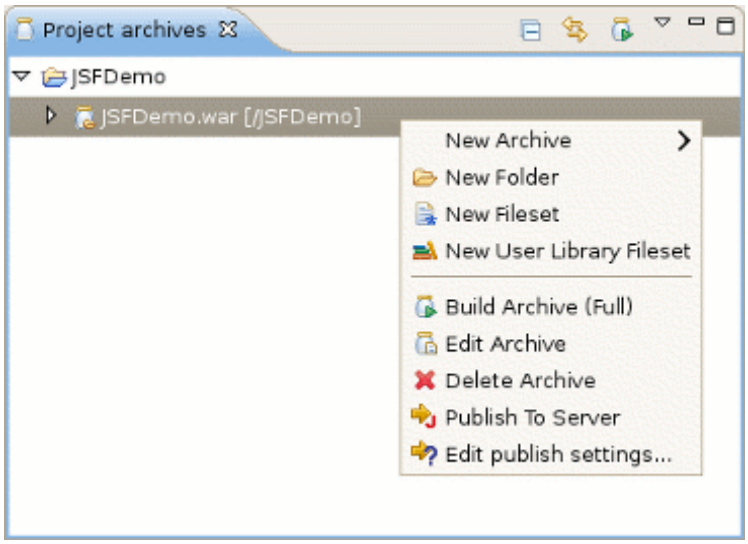

Figure 1.5. Context Menu on the Item

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

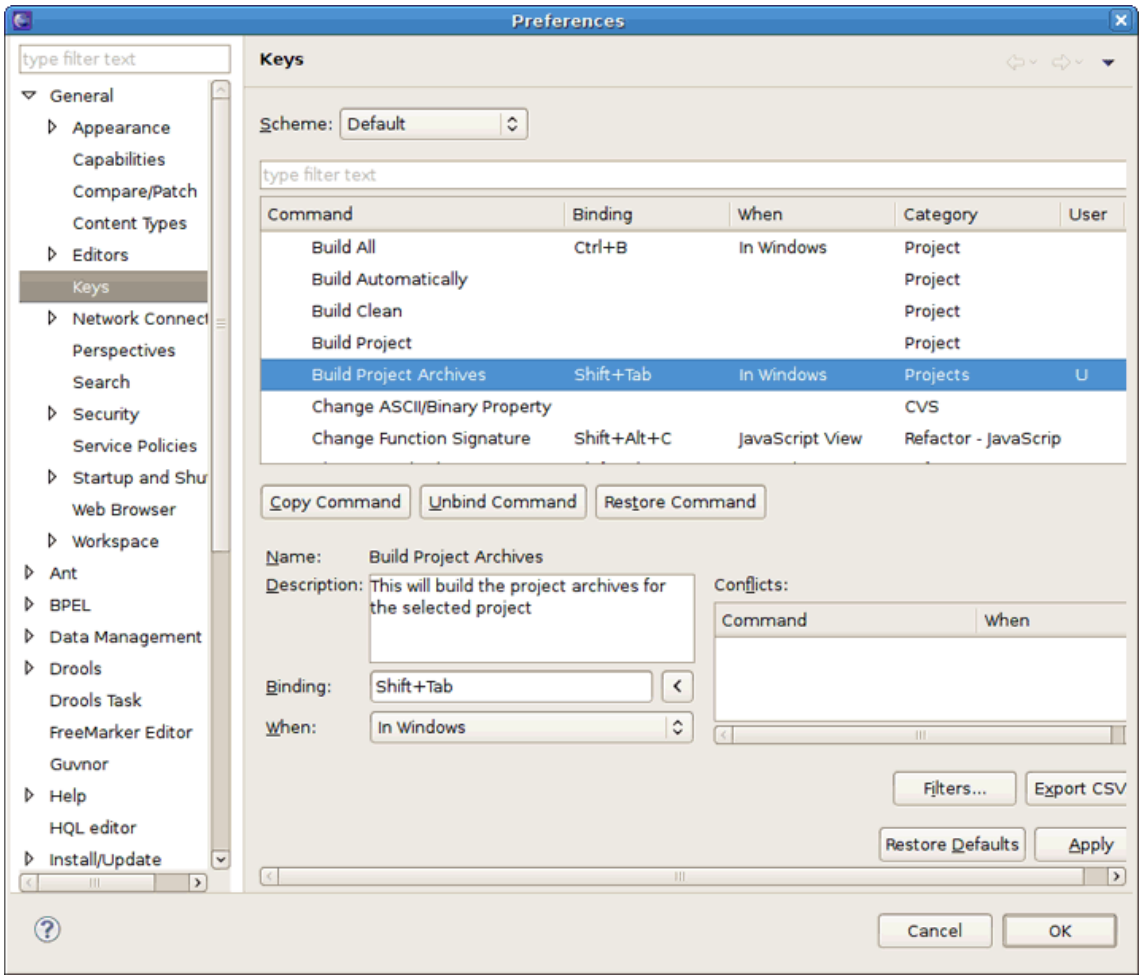

Figure 1.6. Setting the Hotkey for Build Project Archive Action

To perform the building of the project archives, select the project in the Package Explorer view and execute the hotkey combination you assigned for this action.

Learn more about the Project Archives View *here*.

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, the one of wizards steps has a *Target Runtime* and *Target Server* sections. You can deploy the application through the appropriate selection in these sections.

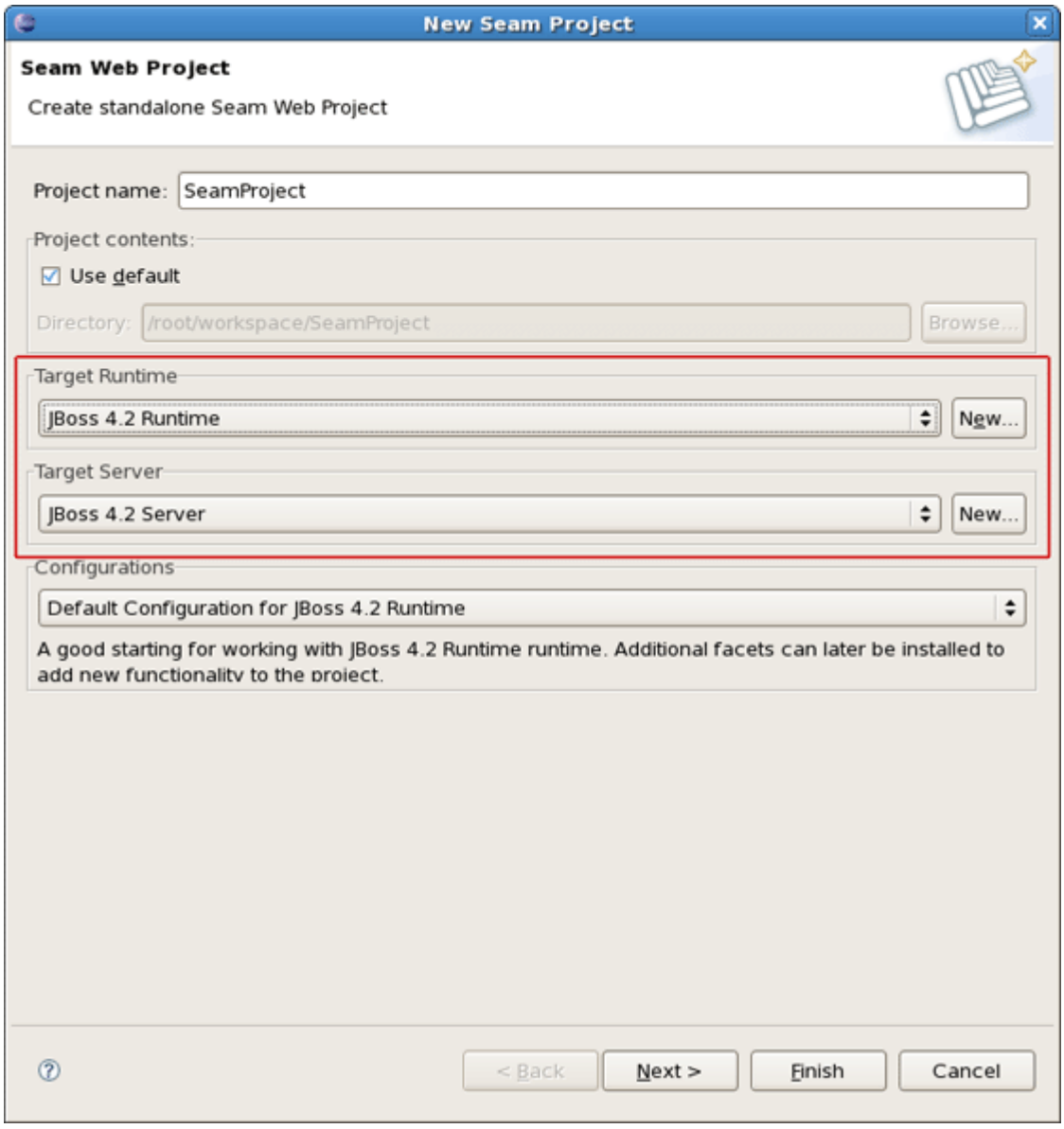

Figure 1.7. Runtime and Server Sections in the New Project Wizard

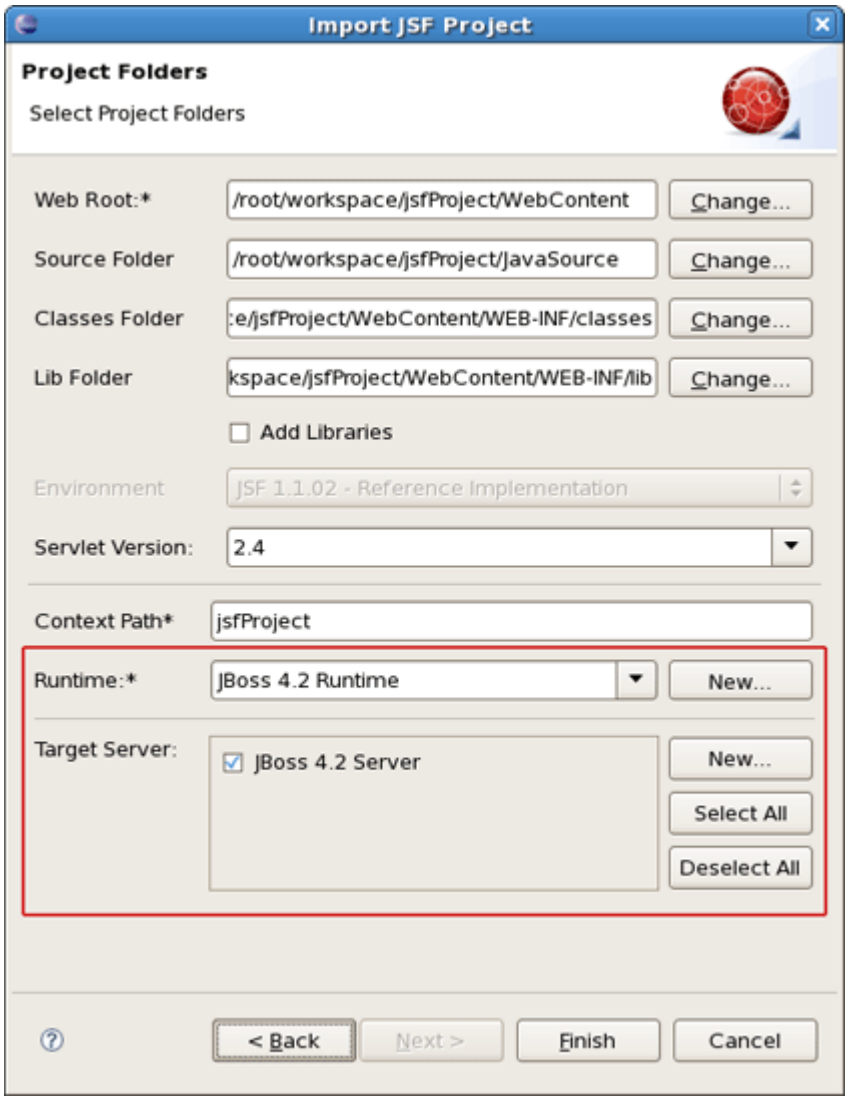

Figure 1.8. Runtime and Server Sections in the Import Project Wizard

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

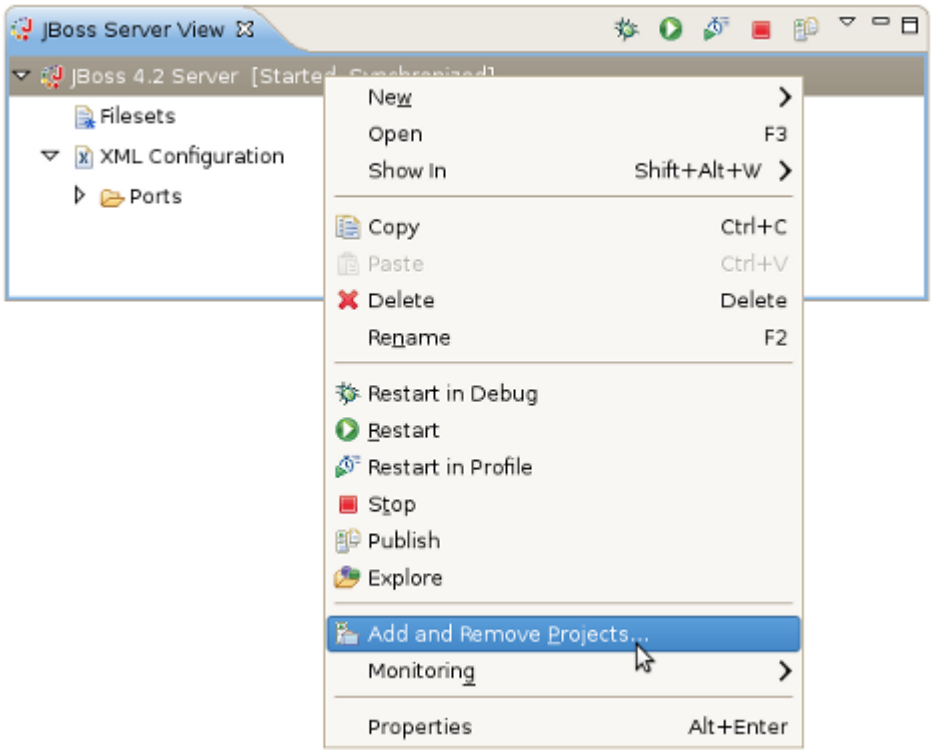

Figure 1.9. Add and Remove Projects From the Context Menu.

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.

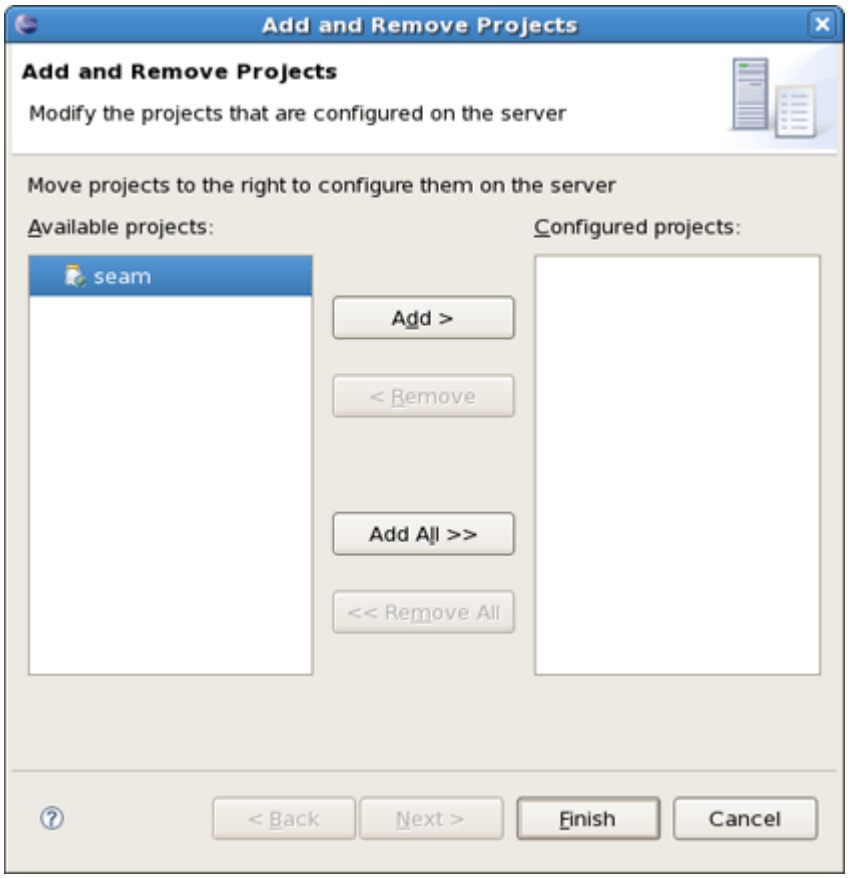

Figure 1.10. Modifying The Projects that are Configured on the Server

1.6. Publishing to JBoss Server

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

The automatic publish of the changes made to the workspace is turned on by default in order to permanently keep the workspace 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* and use the *Publish to Server* ( ) button which initiates an incremental publish.

Here, we have just performed the basic steps you should know to quick start with JBoss server. In fact, there are more functionalities which you can make use of. Further we will talk about them in detail.

1.7. Other relevant resources on the topic

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

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

Find out the answers on the frequently asked questions about JBoss AS 5 usage in the *JBoss5FAQ* [http://www.jboss.org/community/wiki/jboss5faq].

The information on how to monitor a remote JBoss Server from Eclipse you can find in the following *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's 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, and it can then be used for two primary purposes:

- it provides classpath additions to WTP projects that require them.
- for JBoss server at least, it provides information necessary for the starting and stopping of the server, it tells which jars to run and which configuration to use.

2.1.1. Installing a New Runtime

You can install runtimes into eclipse from the *Window > Preferences* menu, and then select *Server > Runtime Environments* from the categories available on the left.

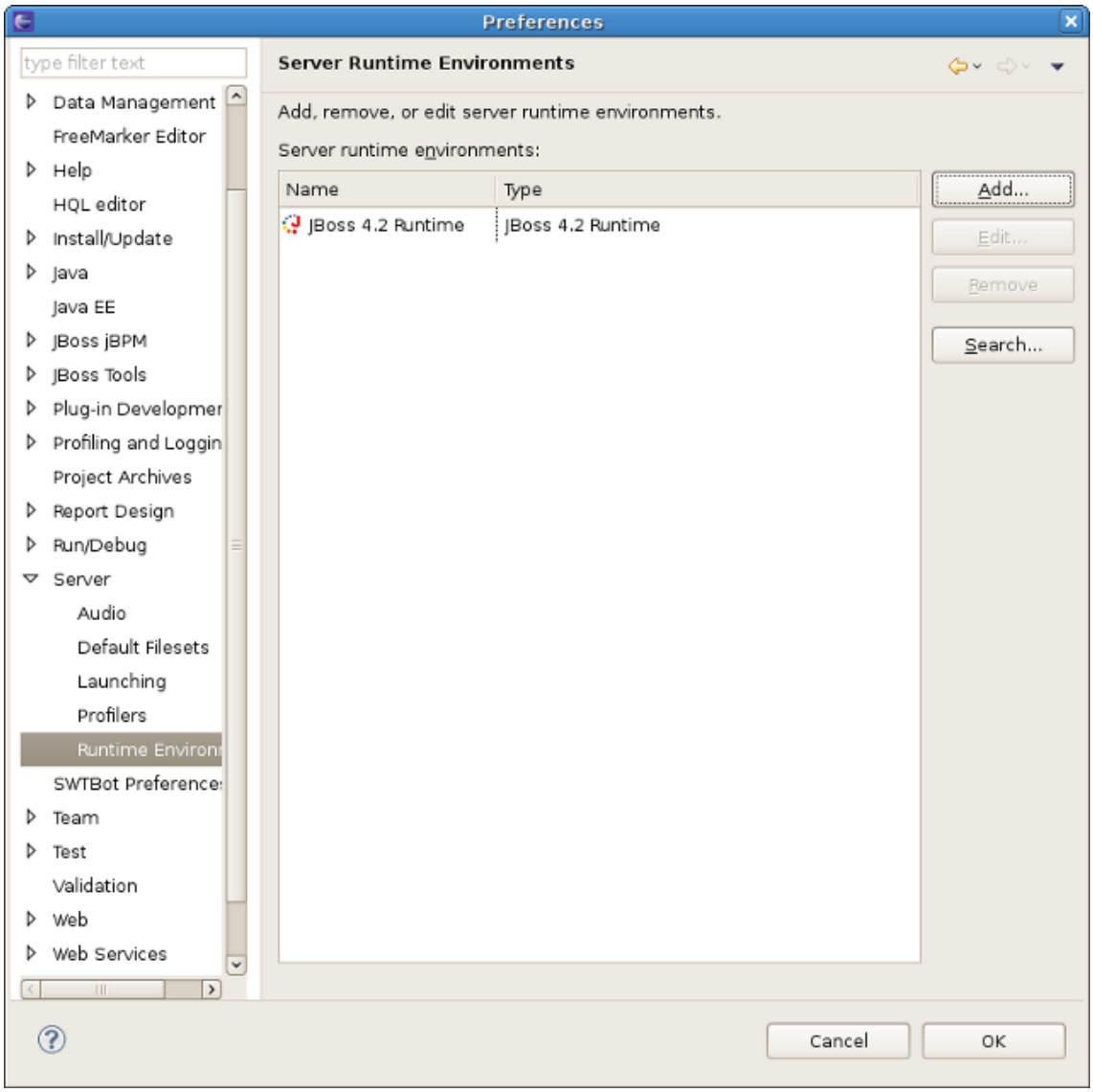

Figure 2.1. Installed Runtimes

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

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

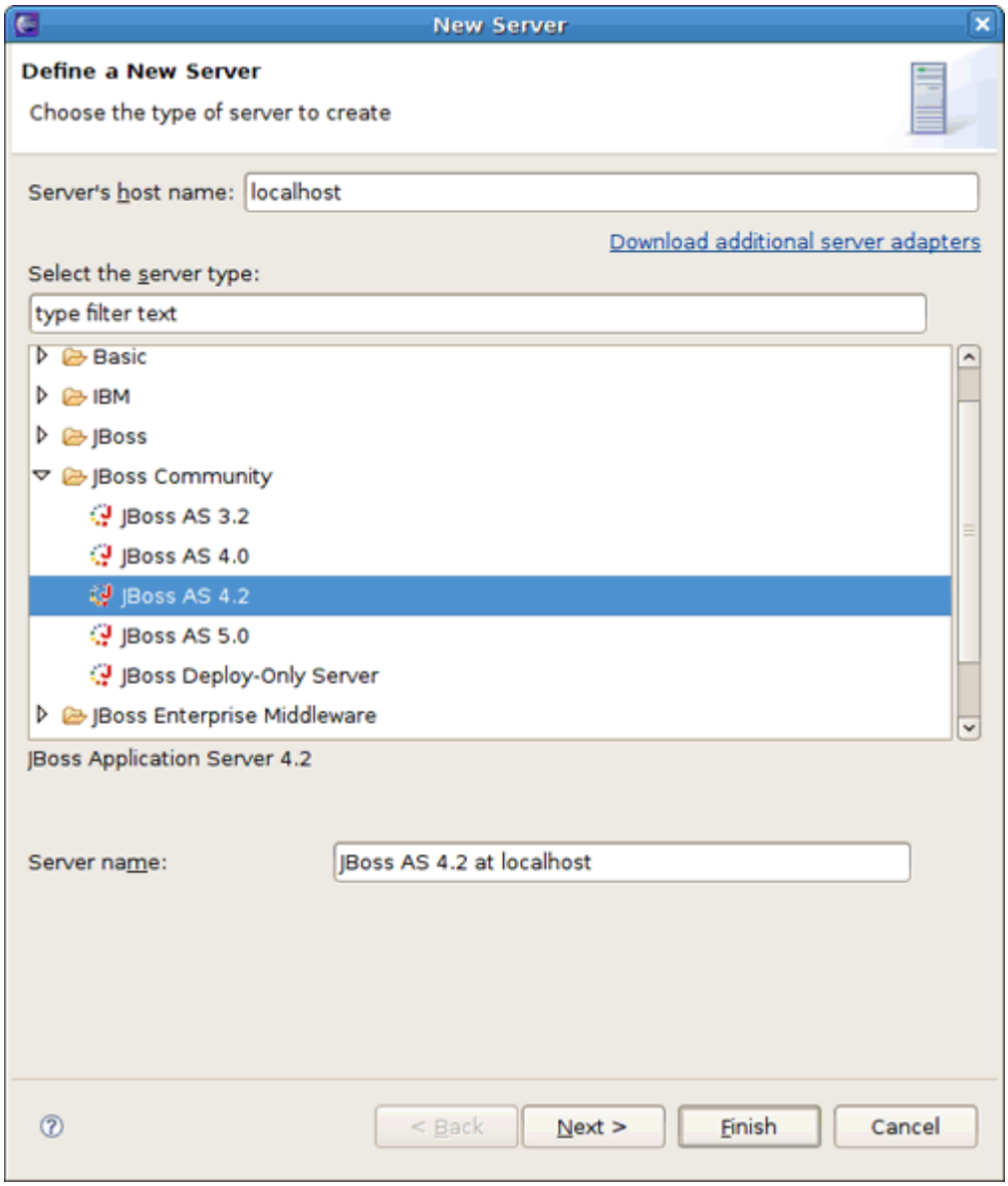

Figure 2.2. Adding a Runtime

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 with its own new feature, that is a safer incremental deployment, 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 4.2 server adapter.

You'll also note a Deploy-Only Runtime type. This type provides no classpath for WTP projects. It is used solely by its server type for the purpose of setting up a deploy directory for users who don't wish to make use of starting, stopping, or debugging their projects inside eclipse.

Figure 2.3. Adding a JBoss 4.2 Runtime

The following table describes all the available options of the currant 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 don't leave a default value. It's better to give descriptive names that will help to distinguish one runtime from another.

Name	Description
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. Thus, rather than forcing you to copy his entire JBoss installation, the structure of the wizard allows to create only a new configuration instead.
Directory	The path to a directory where the configurations are installed.
Configuration	The list of configurations (all, default, minimal) that 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'll create several different runtimes to test each of your configurations. It becomes important to ensure your runtimes, and later your servers, are given descriptive names that help you to remember which is which.

Press *Finish* 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 right click the server, *Overview>Runtime Environment* ,make sure that necessary configuration is chosen,click *Finish* and then *Save* button.

2.1.2. Copying the existing 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 the *previous* section except pressing *Finish* 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 necessary Configuration from the list and click *Copy*.The next dialog should appear.

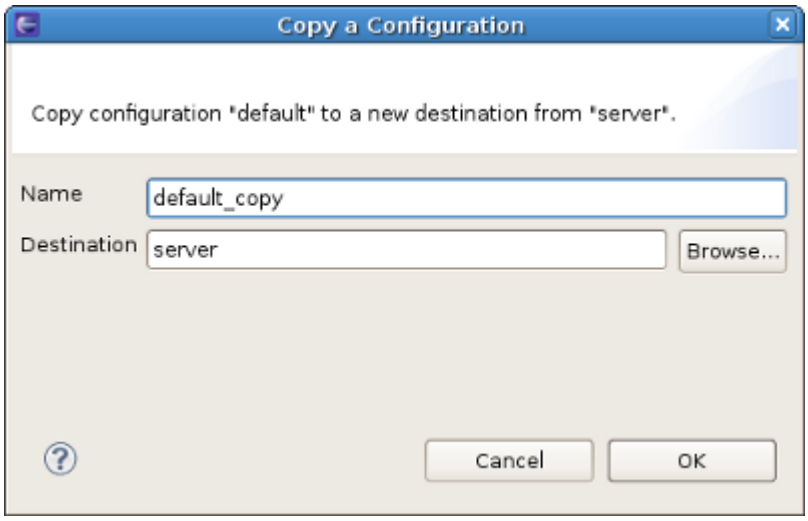

Figure 2.4. Copy the existing configuration

Change the name Click on *Browse* and select your configuration location or leave as it's if you want it to be located together with other runtime configurations.

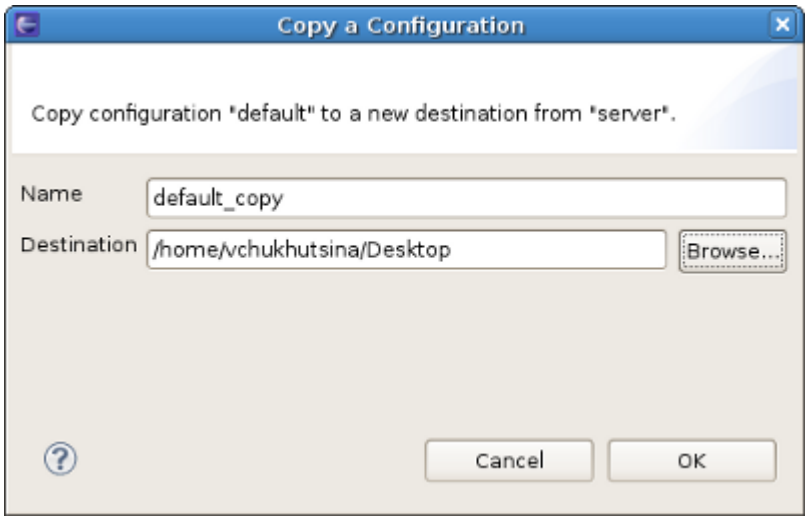

Figure 2.5. Copy the existing configuration

Click *Ok* and you should see the next wizard with the just added copied configuration.

Figure 2.6. Runtime with copied configuration

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

You can also change a configuration of existing runtime to a copied one in the same way from the *Window > Preferences* menu *Server > Runtime Environments* and clicking *Edit*.

2.2. Servers

WTP servers are eclipse-representations of a backing 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 *Deploying with Run On Server Wizard* section).

Servers can be started or stopped with different *command-line arguments [35]*. 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 use the old standard *File > New > Other...* and then *Server* . This should show the wizard like below.

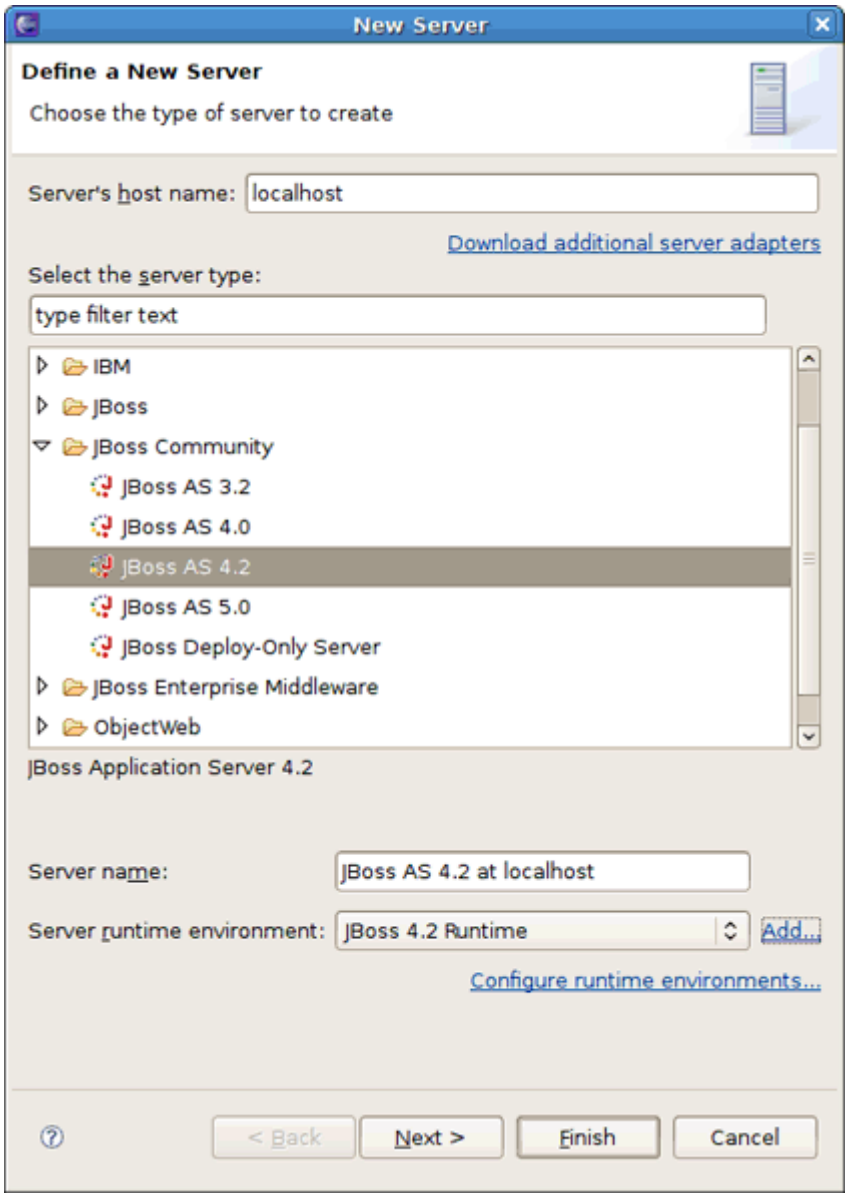

Figure 2.7. Adding a JBoss Server

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

New server wizard allows to name the server appropriately in the *Server name field* or you can use a generated default name.If it's nessecary to restore the default name,after you defined yours, click the Reset default button( ).
From the list of already declared runtimes in the combo box below the view it's possible to select which runtime you want your server to be backed by. If there is no runtime that matches your needs just press the *Add...* link nearby to bring up the wizard for creating a new runtime (see the

Installing a New Runtime section). To configure the already installed runtimes you should go to server preferences that you can easily do by pressing the *Configure runtime environments...* link.

If the server you want to create doesn't have any installed runtime yet, the combo box and the links are absent.

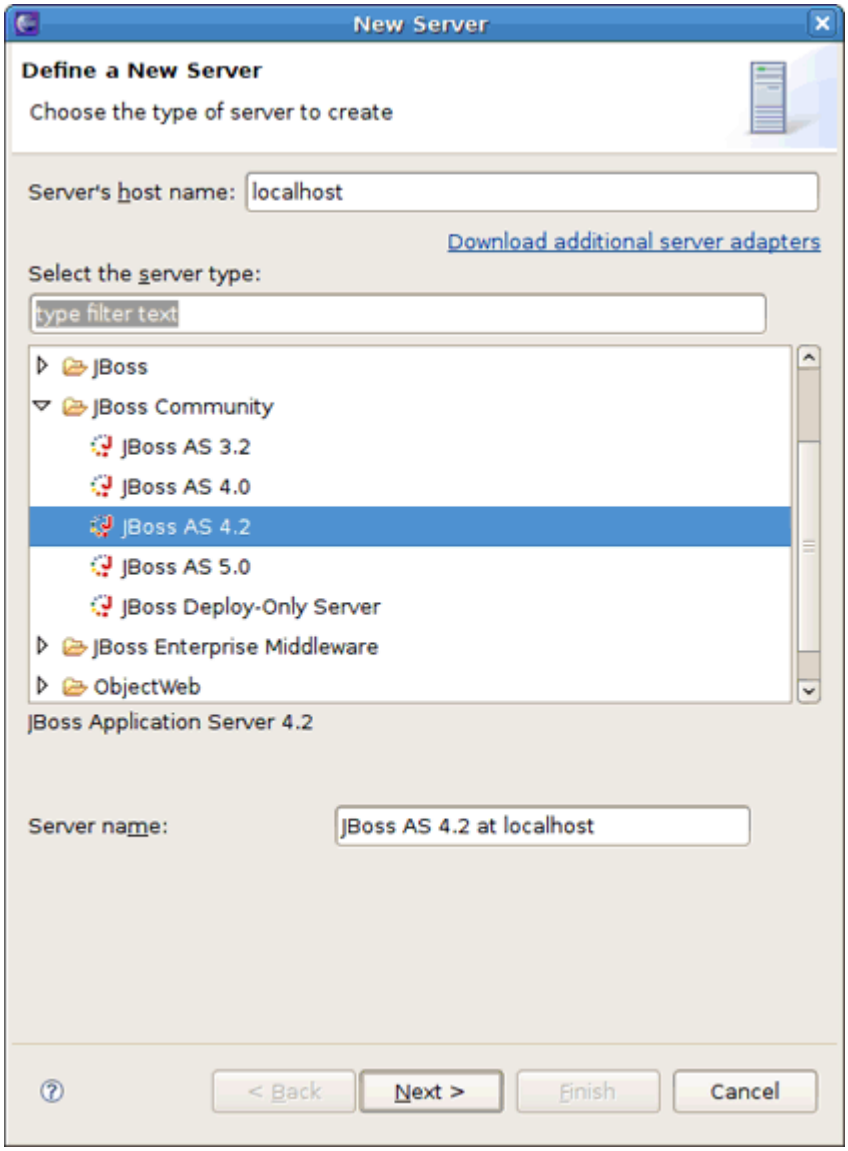

Figure 2.8. Installed Server Runtime Environments

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

Either way, after targeting your server to a runtime, the final screen in this wizard is largely confirmational, giving you a chance to verify that you've selected the appropriate runtime.

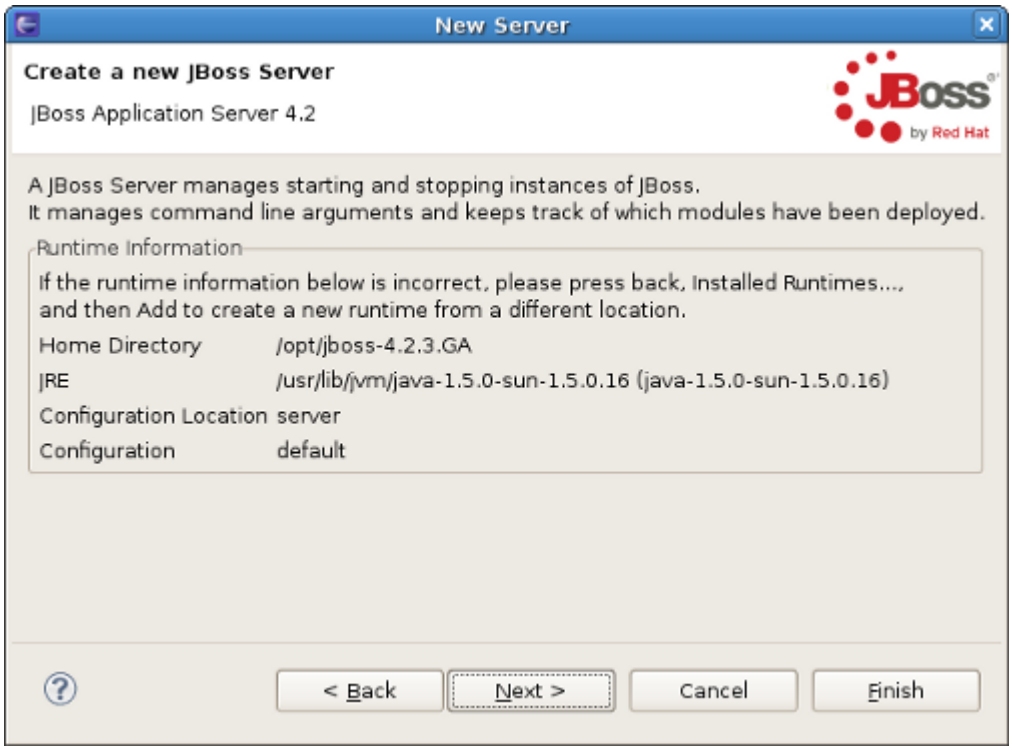

Figure 2.9. Installed Server Runtime Environments

Press *Finish* to complete the process of the server creation.

Now that we've created our runtimes and servers, we can dwell on all services and tools that JBoss Server Manager provides.

We hope, this guide helped you to get started with the JBoss Tools server manager. Besides, for additional information you are welcome on *JBoss forum* [http://www.jboss.com/index.html?module=bb&op=viewforum&f=201].

JBoss AS Perspective

This chapter tells how to manage installed JBoss Server via JBoss AS perspective.

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

3.1. The JBoss Server View

The JBoss Server View is built on Common Navigator Framework allowing extensions and is using label decorators what makes the UI enough compact without loosing the vital information.

Let's have a look at the JBoss Server View and inspect in detail all parts it consists of.

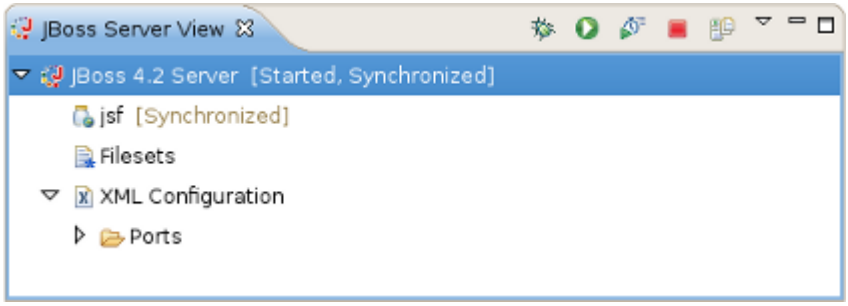

Figure 3.1. The JBoss Server View

3.1.1. JBoss Server View Toolbar

In the right top corner of the JBoss Server View there is a special toolbar which provides a quick access to starting a server (in the debug mode, run mode, or profile mode), restarting a server, stopping a server and a possibility to publish to a server.

Figure 3.2. The JBoss Server View Toolbar

In order to debug your applications or EJB's that are deployed to the server, you must start the server 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.

Start the server in profiling mode button allows to enable profiling actions for your application. For more details on how to start using TPTP profiling with JBoss Tools refer to *TPTP Support* chapter.

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

3.1.2. JBoss Server View Structure

The JBoss Server View displays all declared servers as well as their current states (that is whether they are started or stopped) and statuses in the square brackets next to a server name.

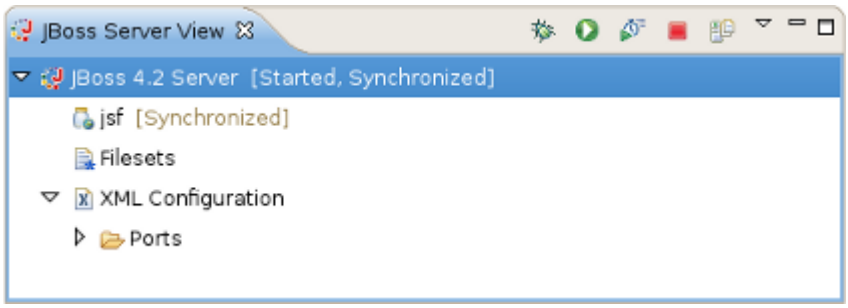

Figure 3.3. The JBoss Server View

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 in-sync

You can control a server behavior as well as adjust some server preferences with the help of the context menu commands.

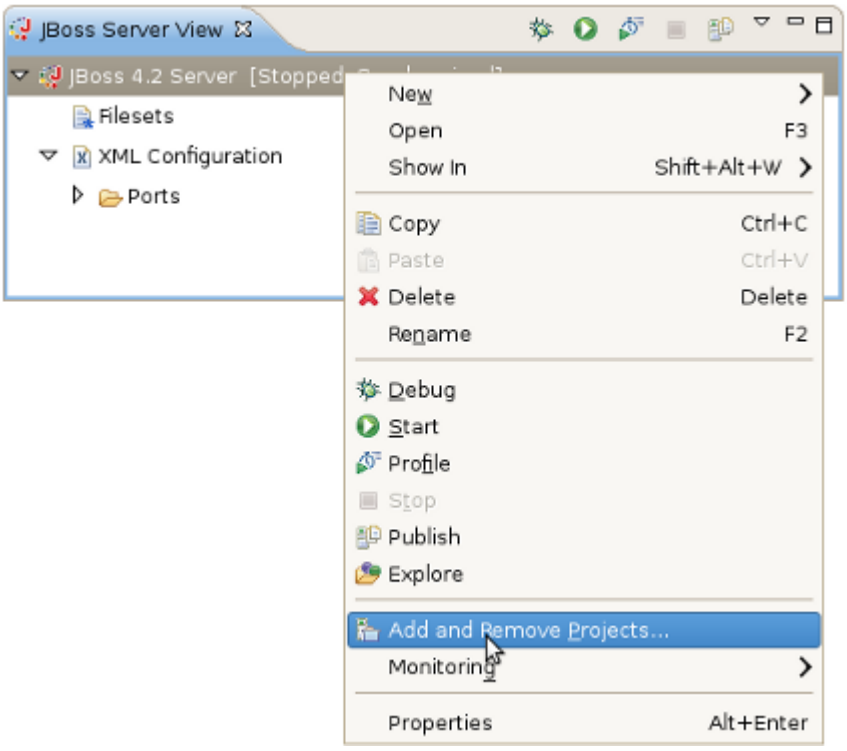

Figure 3.4. Context Menu Commands

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 to define a new server
Open	The option opens the Server editor
Show in	This option gives an easy access to the next views: Console, Debug view, Server Log or MBean Explorer
Delete	Standard option that allows to delete the chosen server
Start	The action for stating a server in a run mode
Debug	The action for stating a server in a debug mode
Stop	The action for stopping a declared server
Publish	The action for synchronizing the publish information between the server and workspace
Explore	This action uses the native OS file explorer to browse the deploy destination. Note: The option is also available for deployed resources/ projects (see the figure below).
Add and Remove Projects	The option allows to publish a new project to the server (if its type is supported)

Name	Description
Monitoring	Possibility to add ports to be monitored on the current server
Properties	Opens the window to adjust the current server preferences

Under the server element in the JBoss Server View, you can see currently deployed to the server modules and some server extensions which provide the additional information on the server.

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

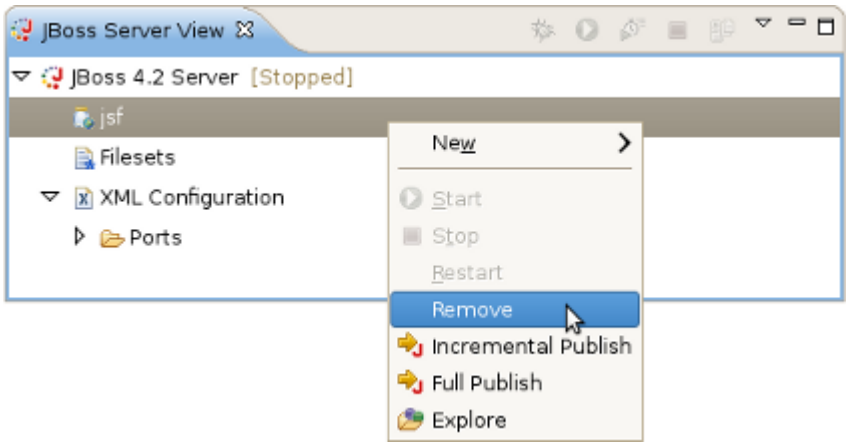

Figure 3.5. Modules Action

3.1.2.1. Filesets

The *Filesets* category in the JBoss Server View is intended for files filtering.

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

The New File Filter wizard should appear.

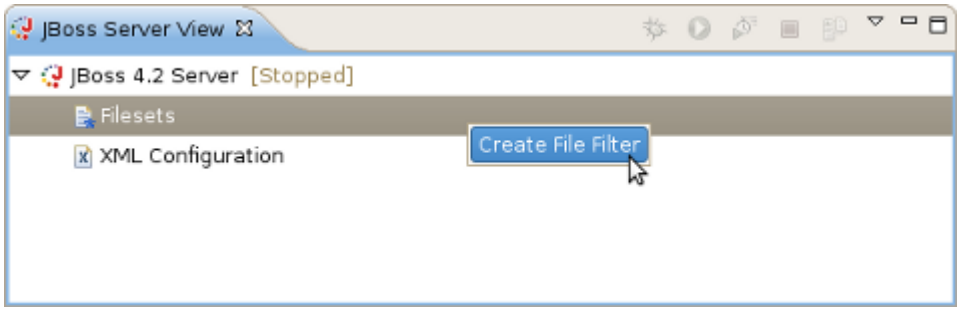

Figure 3.6. Creating a New File Filter

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

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

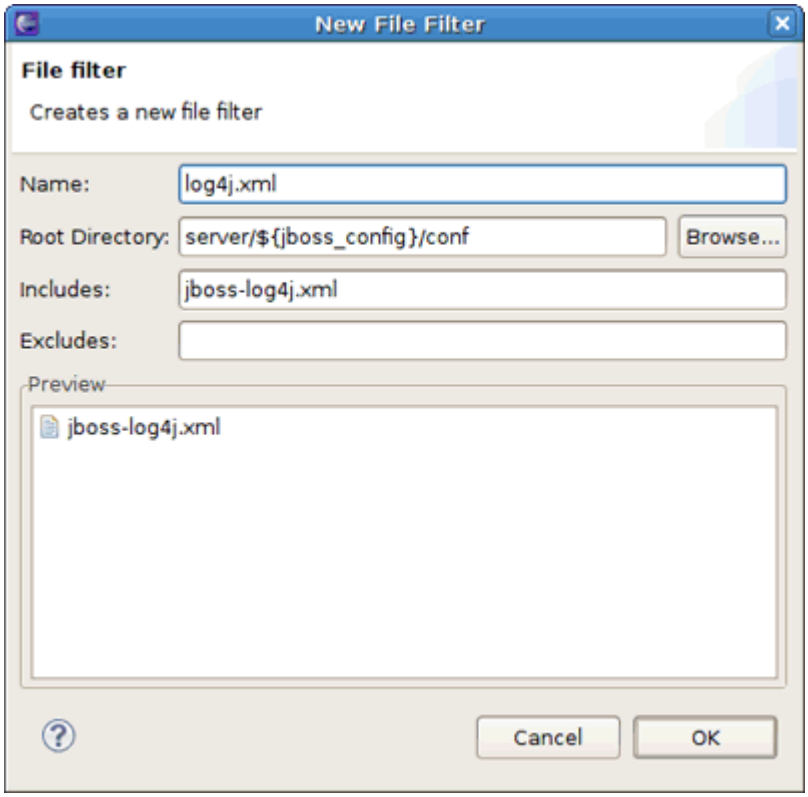

Figure 3.7. New File Filter Wizard

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

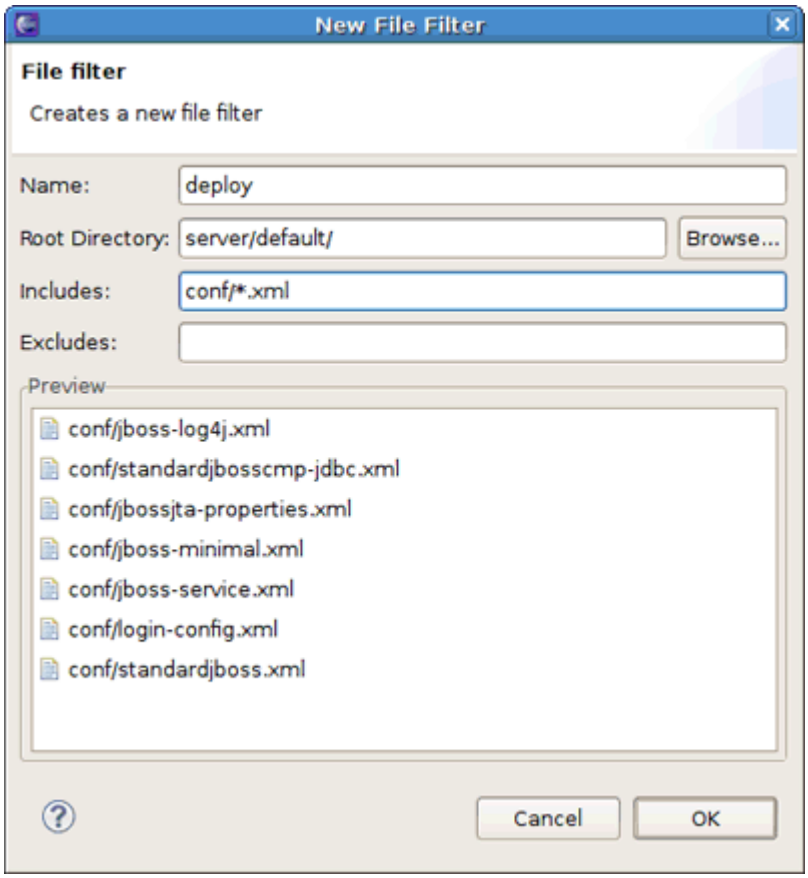

Figure 3.8. New File Filter Wizard

After the filter is created, you can observe it by expanding the *Filesets* category in the JBoss Server View.

It's possible now 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.

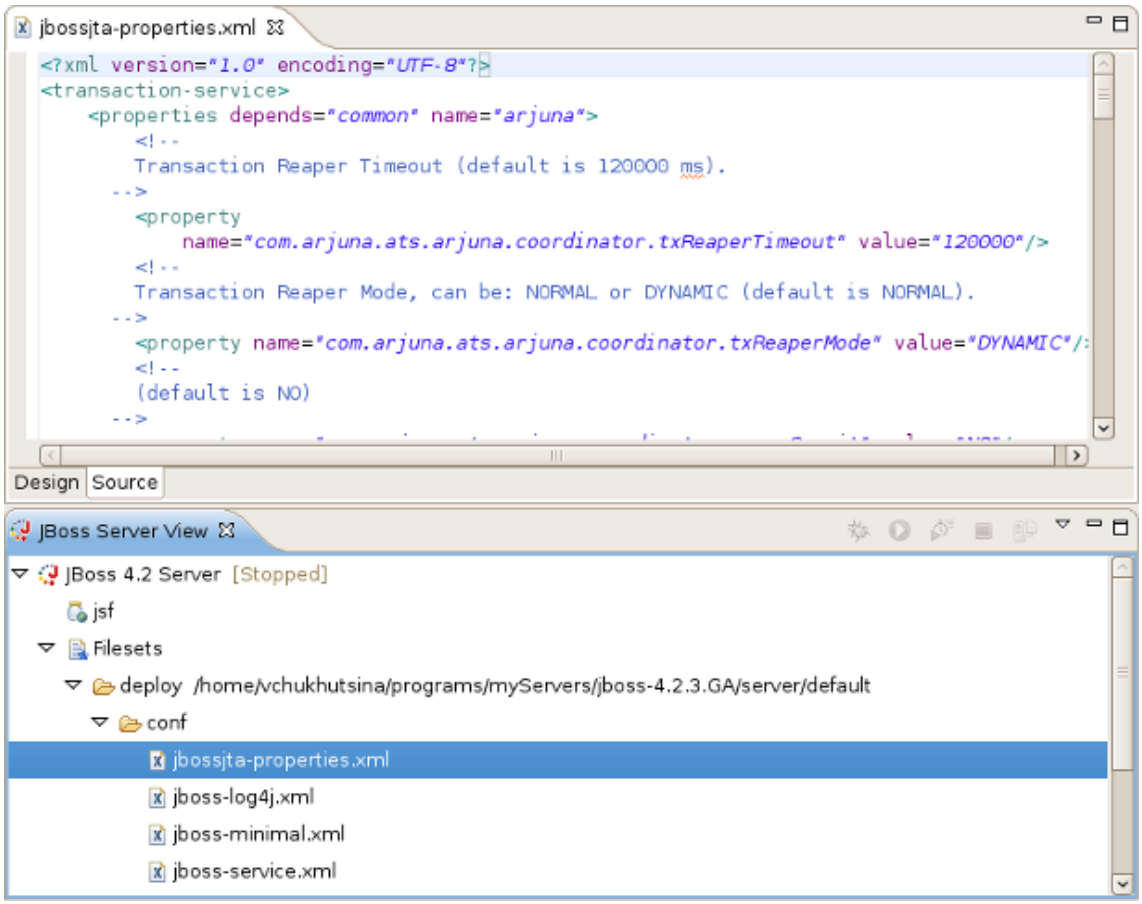

Figure 3.9. Direct Editing from the Filesets

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

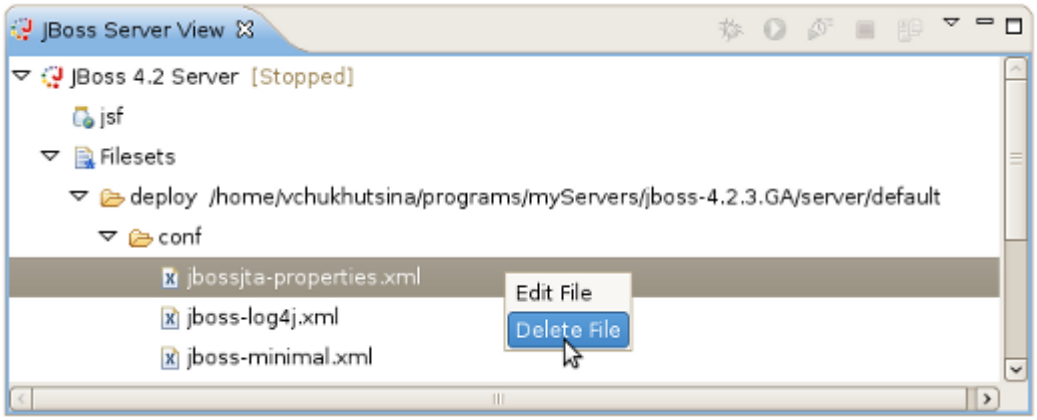

Figure 3.10. Deleting the File from the Filesets

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

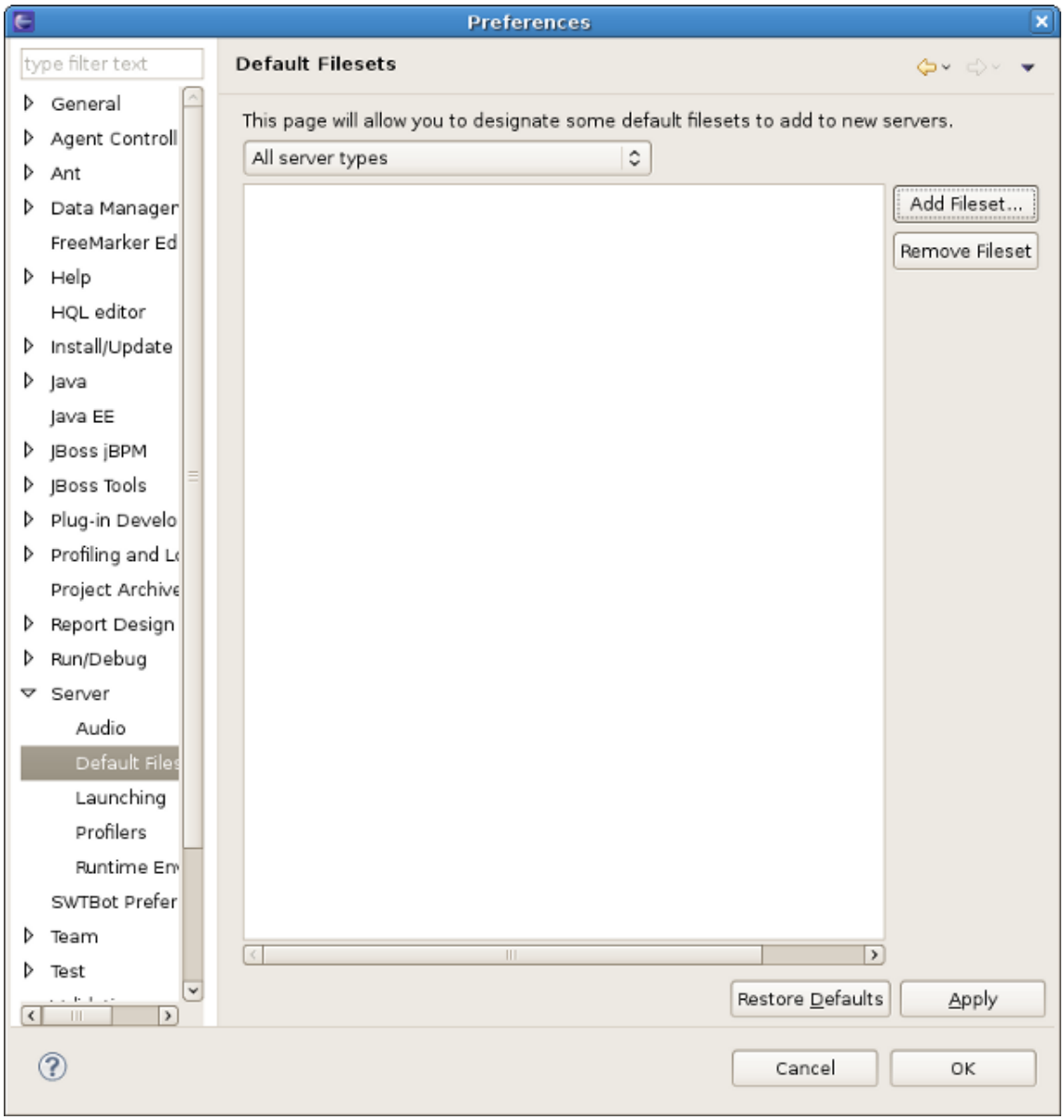

Figure 3.11. Deleting the File from the Filesets

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 *Add fileset...* button. In the opened New File Filter wizard follow the steps described *before [23]* and finally click *Apply* button on the preference page.

The defined file filter will be automatically added to new servers during 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:

You are assumed to be 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 is filled with many of the most commonly used ports in the JBoss Server.

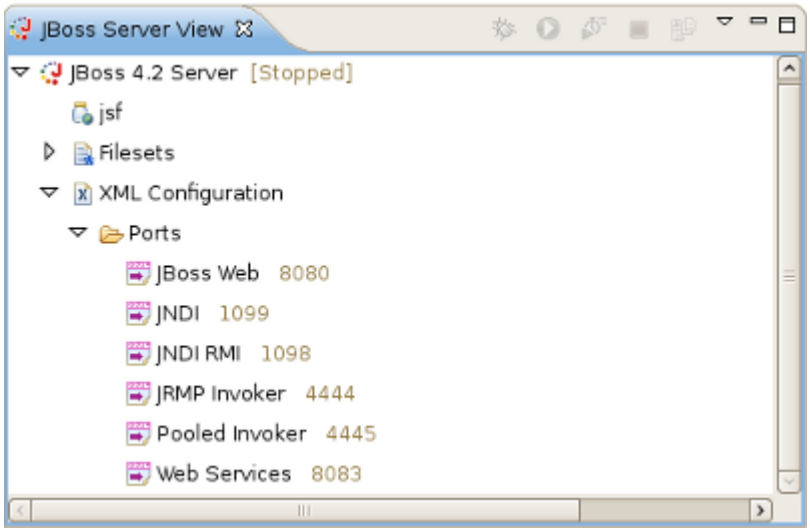

Figure 3.12. XML Configuration

By right-clicking on *XML Configuration*, 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 *JBoss Server View*. Look for them in the *Inactive Categories* afterwards to re-enable.

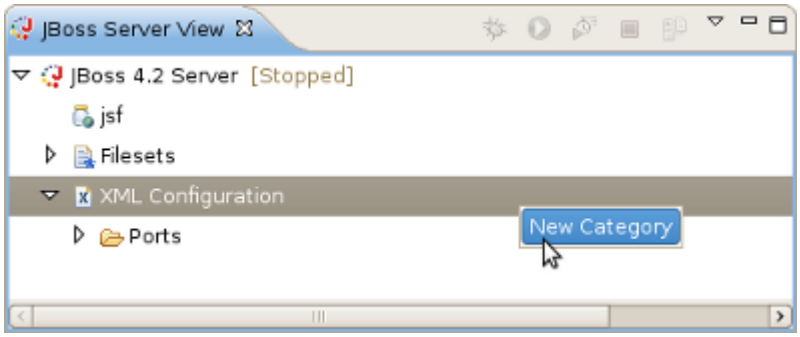

Figure 3.13. Adding New Category

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

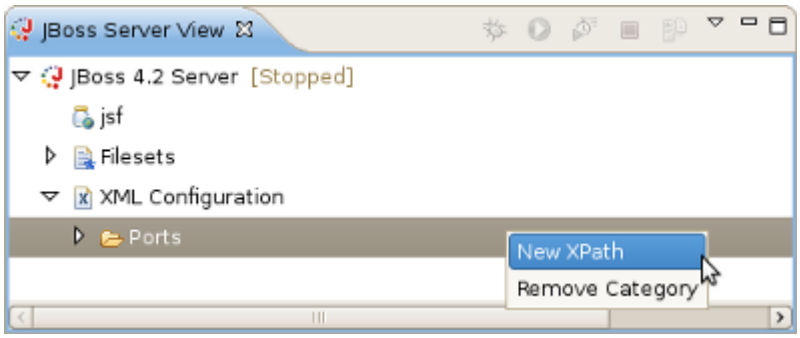

Figure 3.14. Adding New XPath

After that, the dialog shown below will appear.

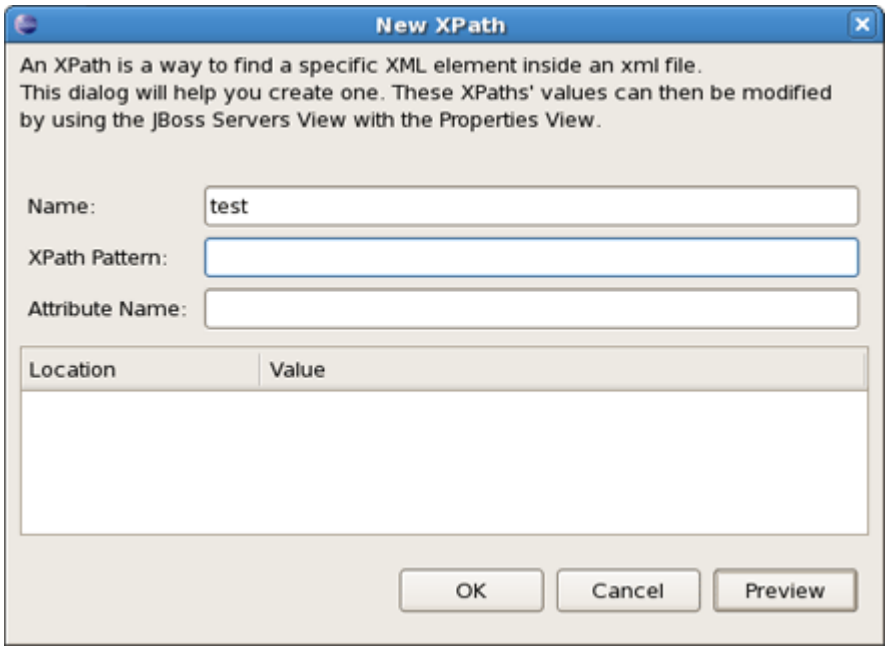

Figure 3.15. Adding New XPath

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* like on 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>
```

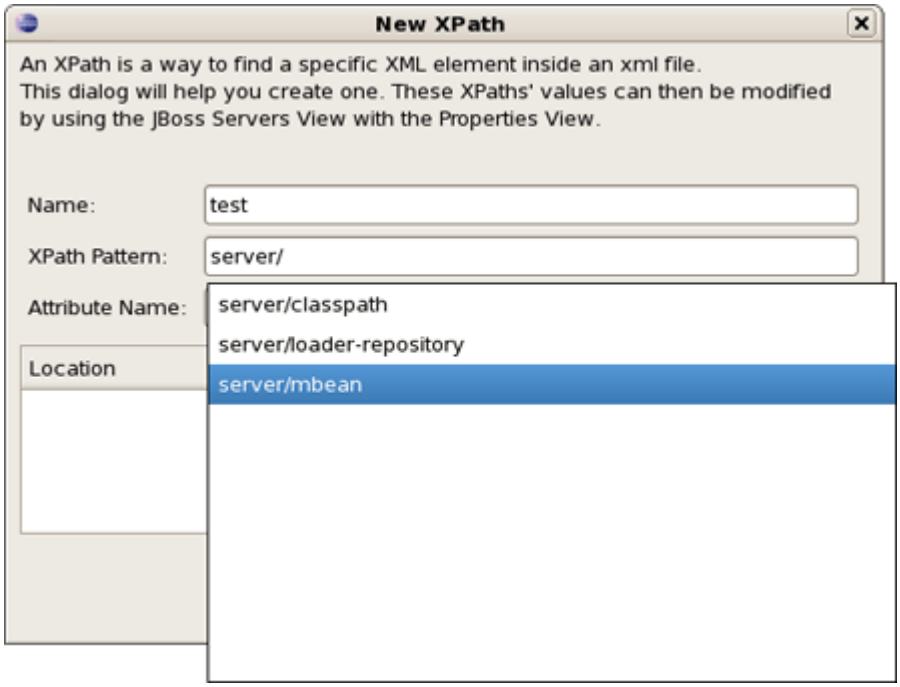

Figure 3.16. XPath Preview

Tip:

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

Then, on the other hand, 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 *Preview* to see how many matches you have for that particular XPath.

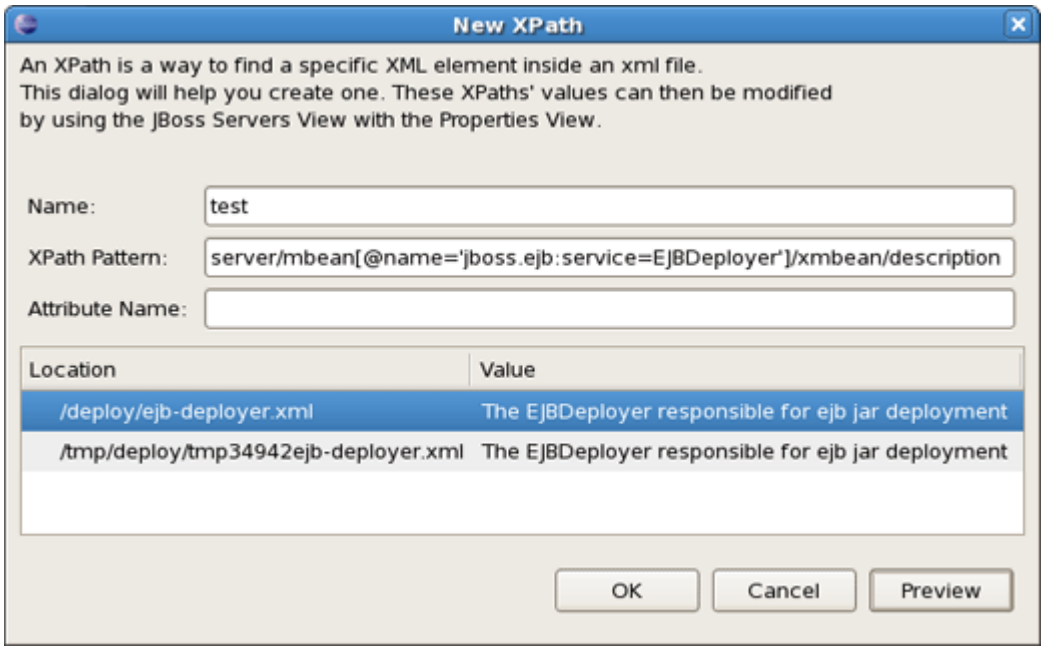

Figure 3.17. XPath Preview

3.1.3. Drag-n-Drop to JBoss Server View

Starting from 2.0.0.CR2 version of JBoss AS Tools the JBoss Server View supports drag-n-drop of deployable and runnable projects/resources.

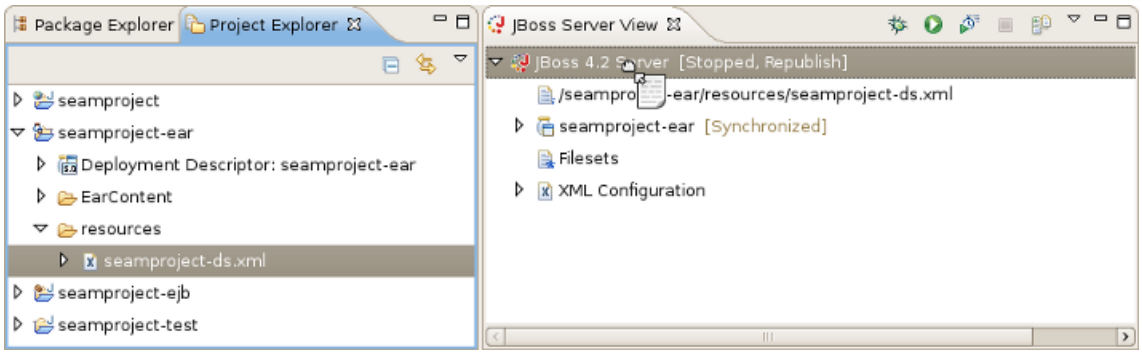

Figure 3.18. Dragging to the JBoss Server View

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 *WebContent* 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* option 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* 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).

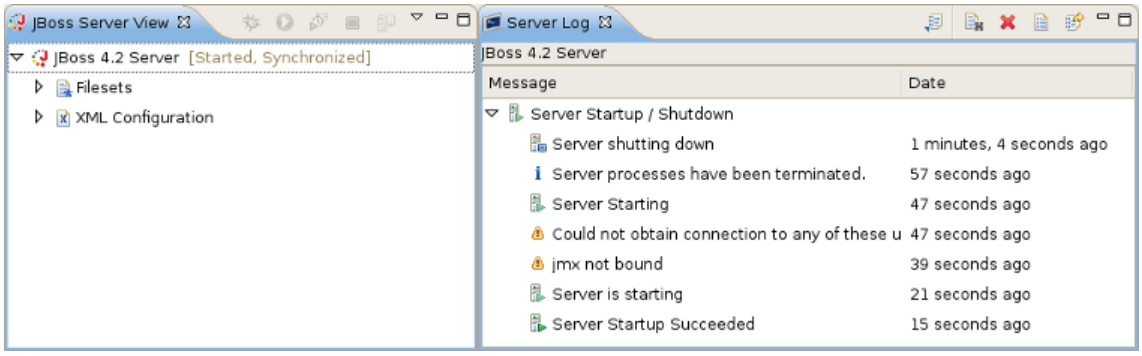

Figure 3.19. Event Log Actions

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

Table 3.3. Server Log Toolbar Icons

Name	Description
Export Log	Possibility to export the log into a text file
Clear Log Viewer	The 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 editor window will appear allowing you to edit parts of that server.

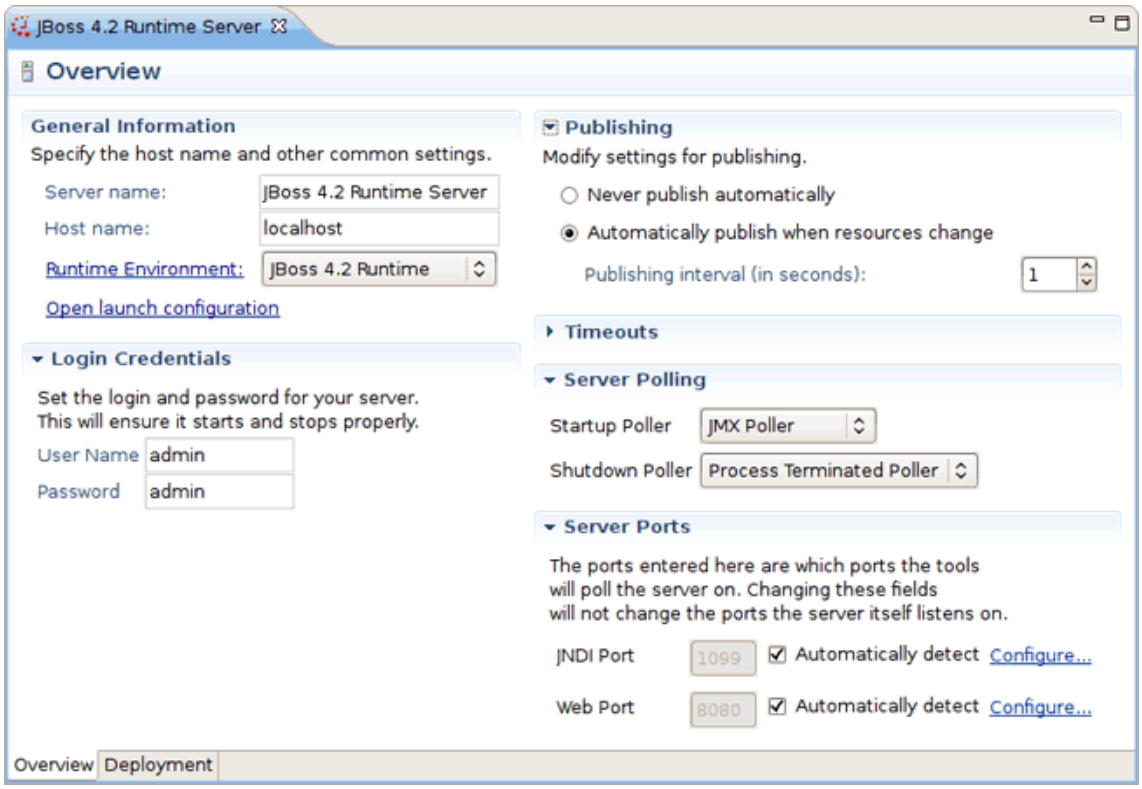

Figure 3.20. Preferences Page for the Chosen Server

Tip:

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

Under the *Publishing* section it' possible to disable/inable the automatic publishing of the changes in the workspace.

It should be pointed out that our server adapter by default tries to automatically detect the ports it needs for integrating with a JBoss Server. Sometimes it can though be relevant to override this automatic detection if you are using some custom configuration. For this purposes the *Server Ports* section in the Server editor is provided where the port settings are configurable. Click the *Configure...* link to bring up the wizard for adjusting the settings for the ports.

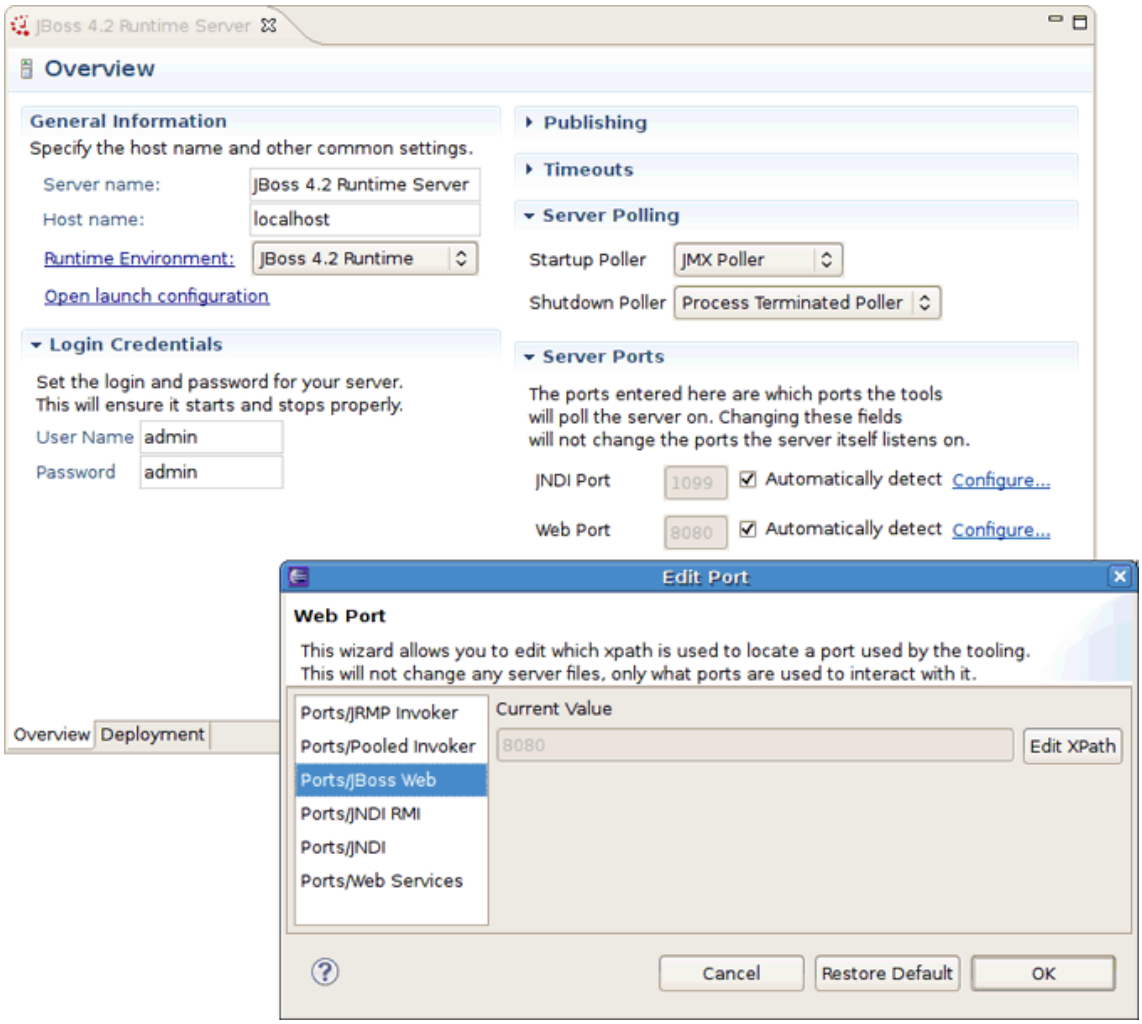

Figure 3.21. Server Ports Preferences

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

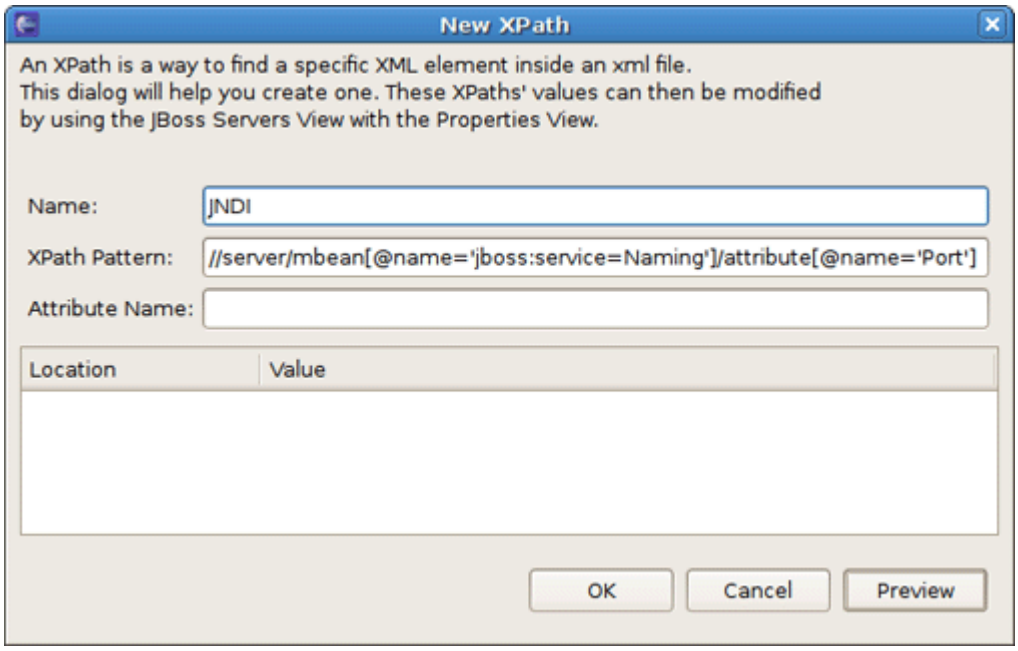

Figure 3.22. XPath Pattern for a Server Port

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 (see the *Server Polling* section). If you change the Startup poller to Timeout Poller (it may need in case, for example, you're 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.

Server editor makes it also possible to modify the server's launch configuration. It's just after clicking *Open launch configuration* link. In the open window there are the tabs for setting command line arguments, main, classpaths and other things that are relevant to launching the server.

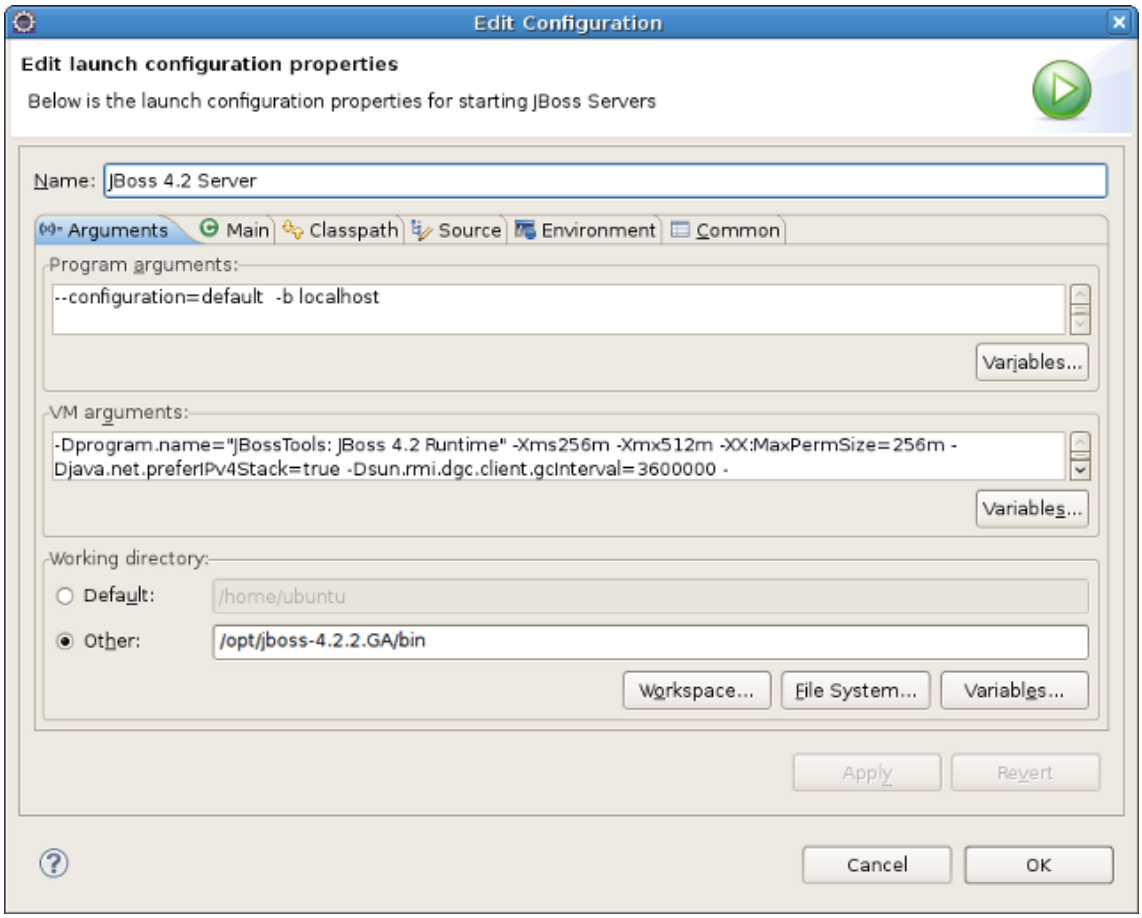

Figure 3.23. Launch Configuration Properties

The first tab shows the Jboss server arguments

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

Note:

Please note, that the Launch Configurations for JBoss Servers are enough strict in enforcing the configured values in the server 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 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 (default is org.jboss.Main), you can change it, if necessary.

Until 3.0.0.GA release of JBoss Tools, the servers classpath was readonly, 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.

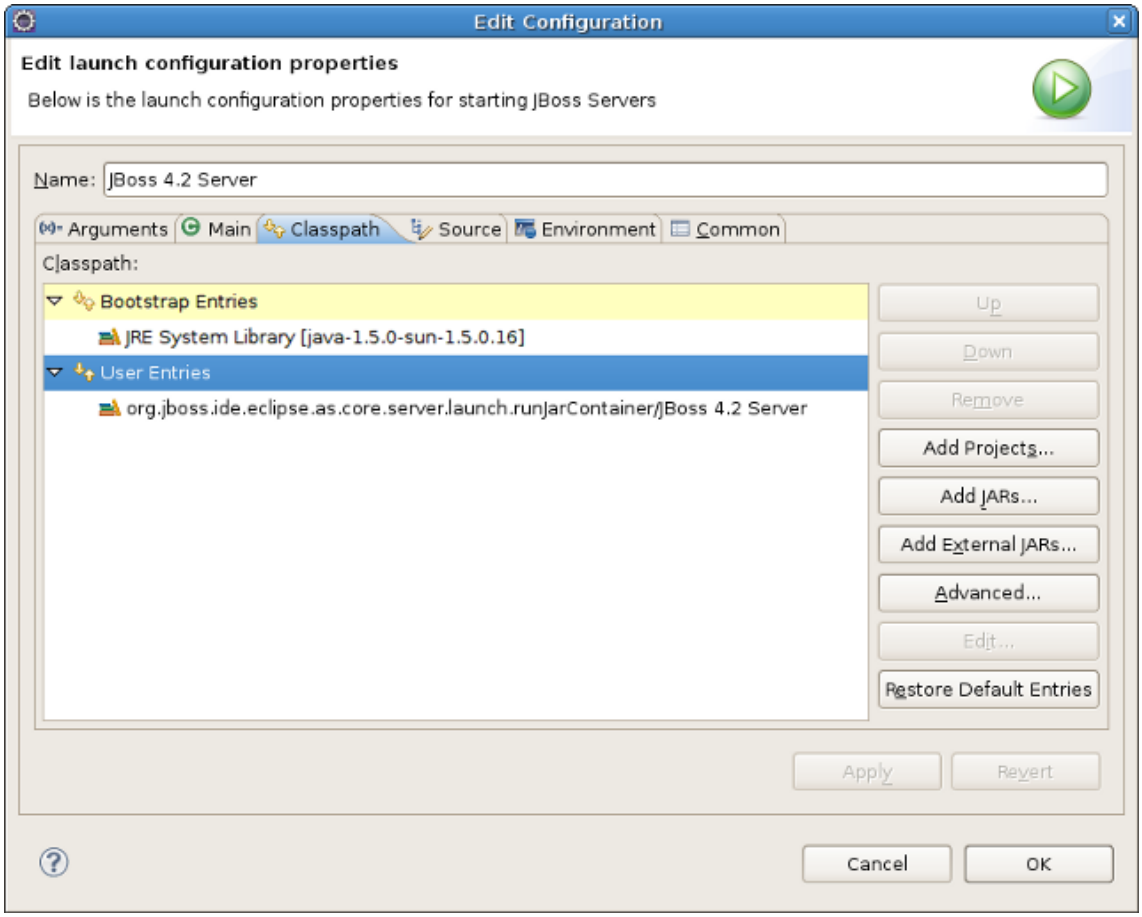

Figure 3.24. Server Classpaths

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

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 come with our own Archives tool with a bit easier and less-verbose XML and a handy user interface. The Project Archives plugin consists primarily of a view, that is Project Archives view, to set up each packaging configuration.

So far, 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, and is in a file named *.packages* , which has a fairly simple XML structure. Modifying the file by hand is neither required nor recommended, and using the UI is the official way of modifying your packaging structure.

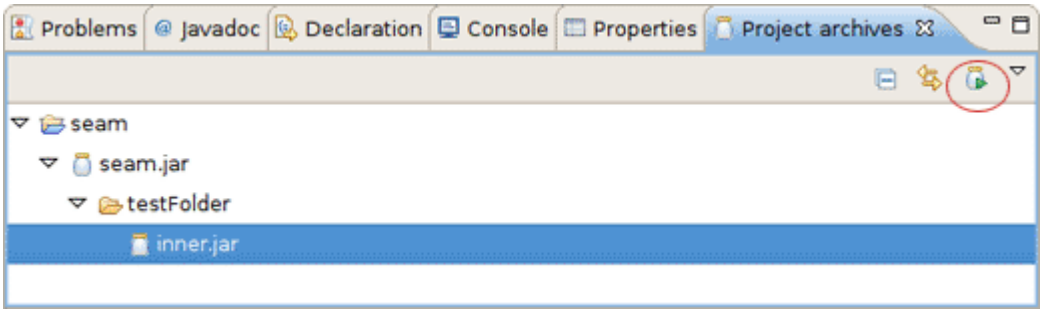

Figure 3.25. Archives View

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 some 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 what you want to create an archive.

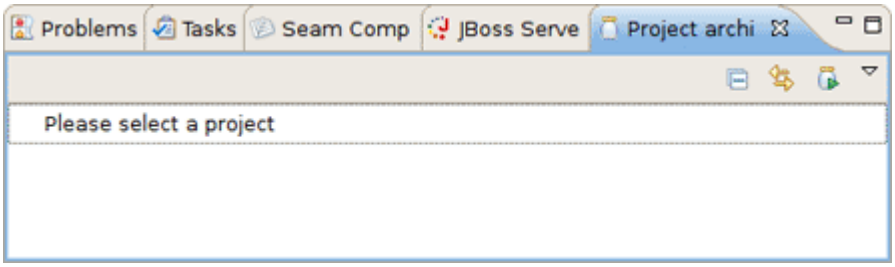

Figure 3.26. Archives View

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.

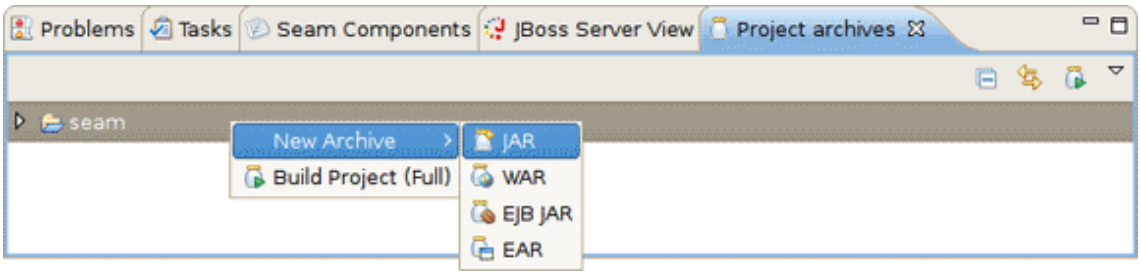

Figure 3.27. Create an Archive

Note:

If you see only JAR from the list of available archive types, you should verify whether AS Tools plugins/features are in place. EAR, EJB JAR and WAR options are contributed by the AS Tools independently from webtools and the virtual project model. Thus, without them only JAR 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.

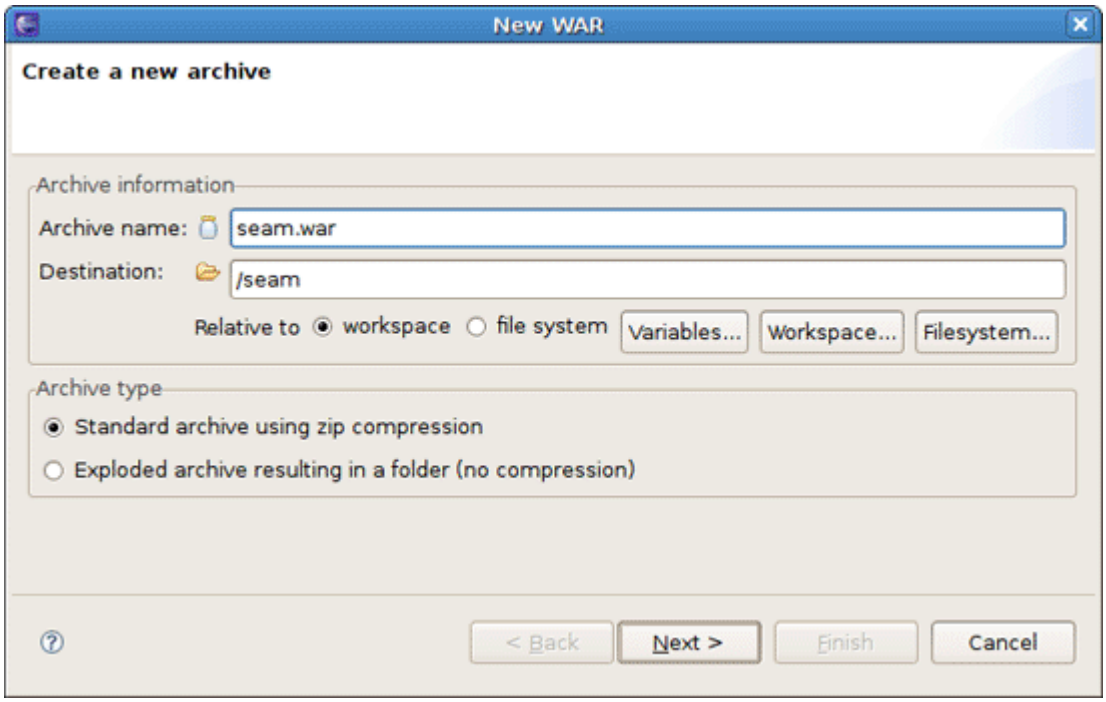

Figure 3.28. New WAR Wizard

The page is pretty simple. First it asks 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 some other archive, or inside a folder declared inside an archive. Select the necessary checkbox (either *workspace* or *file system*) for marking the destination as related to either workspace or file system. 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 press the *Workspace* button. At the bottom of the list, you'll see archives that have been declared in the workspace.

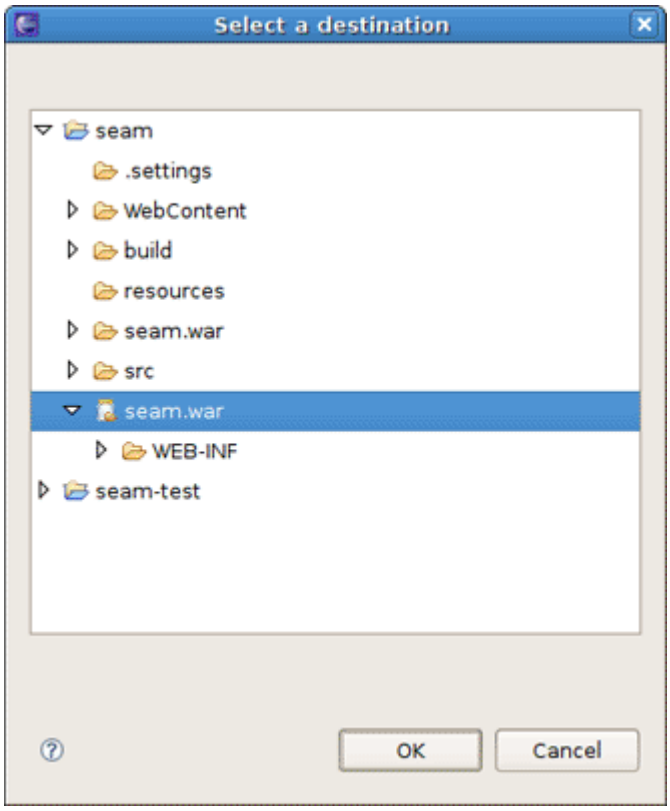

Figure 3.29. Selecting the destination in the workspace

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 chechbox in the *Archive type* section.

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

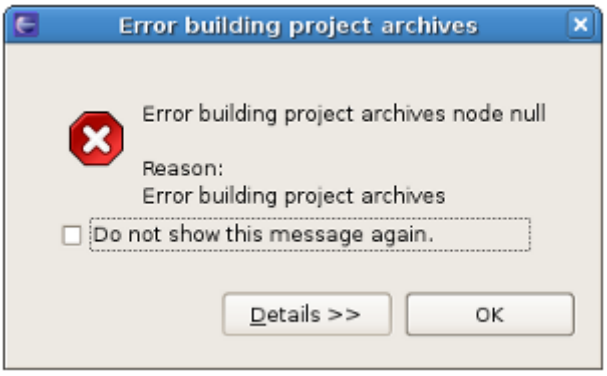

Figure 3.30. Selecting the destination in the workspace

Click in *Details* to see detailed information about what caused the error.

In the Package Explorer you can observe the created archive.

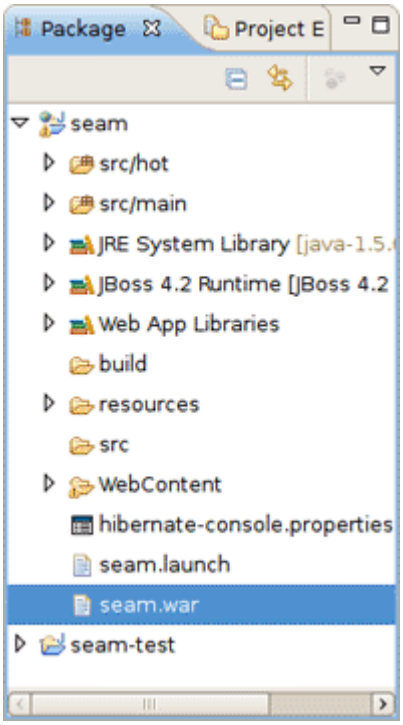

Figure 3.31. The Archive in the Package Explorer

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.

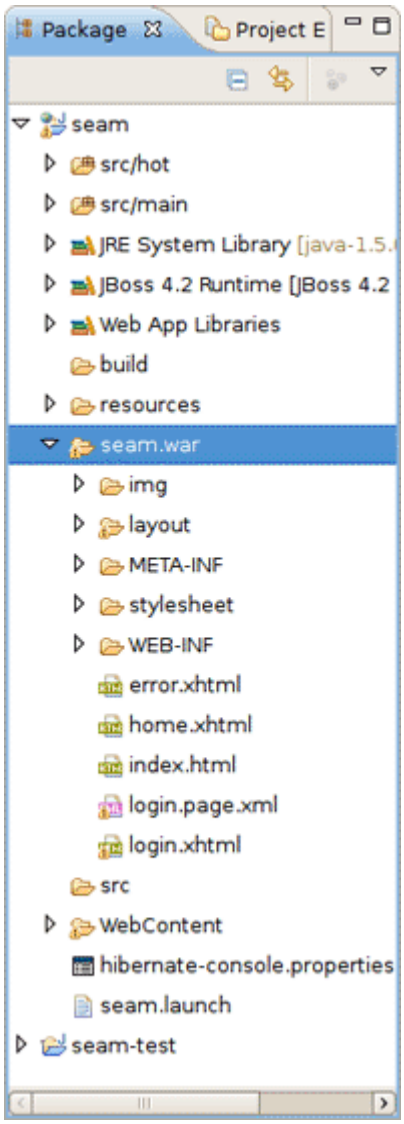

Figure 3.32. The Exploded Archive in the Package Explorer

3.2.2.1. Creating a Folder

Creating a folder is much easier. You simply right-click on an archive or folder you want your new folder to be a child under. The only piece of required information is naming the file.

3.2.2.2. Creating a FileSet

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

The New Fileset wizard requires a destination (where the files will go), 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.

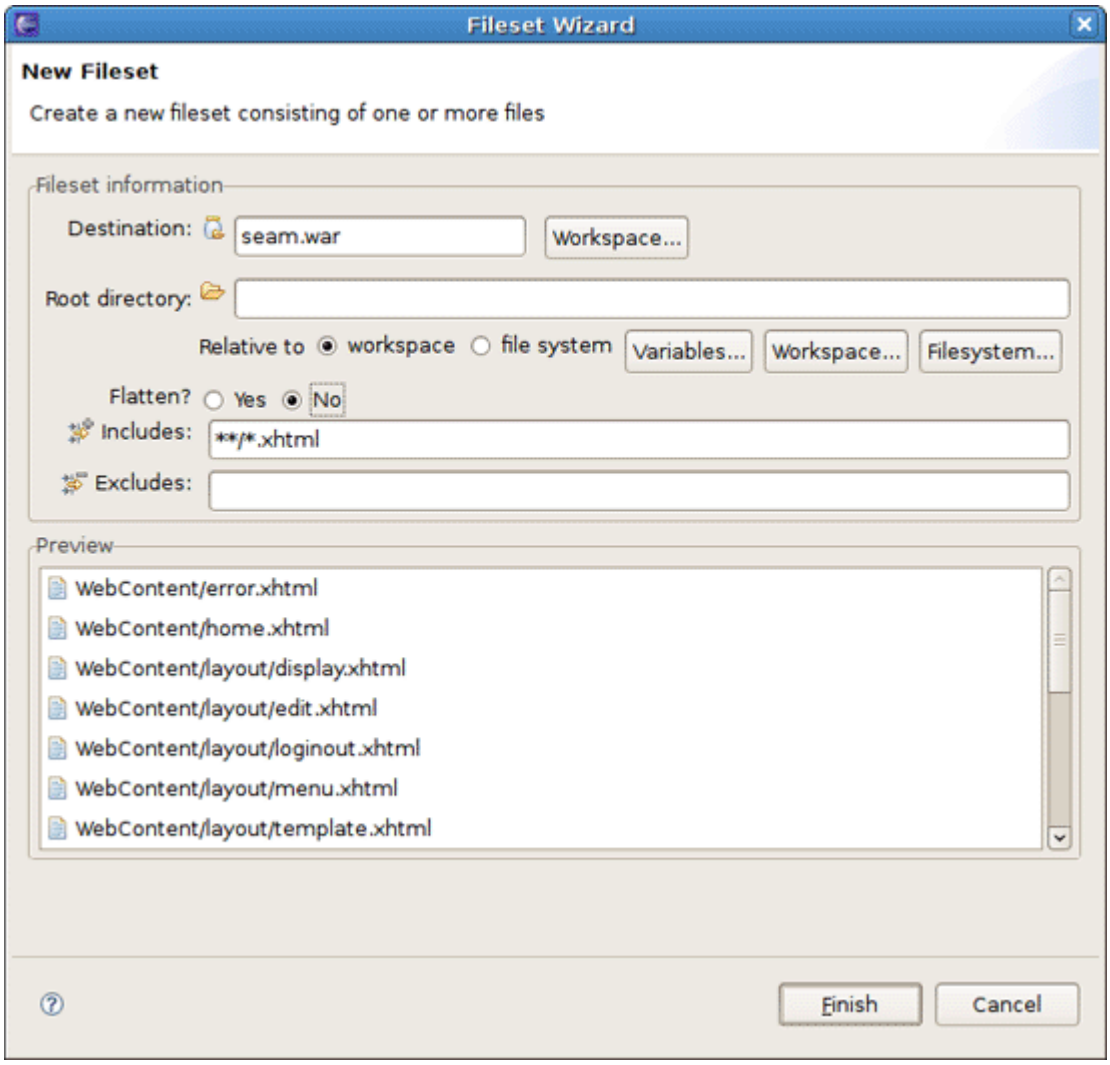

Figure 3.33. Adding a New FileSet

Below that, the fileset requires only an includes pattern and an excludes pattern. As you type in either of these fields, the preview viewer should update itself with which files are matched.

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

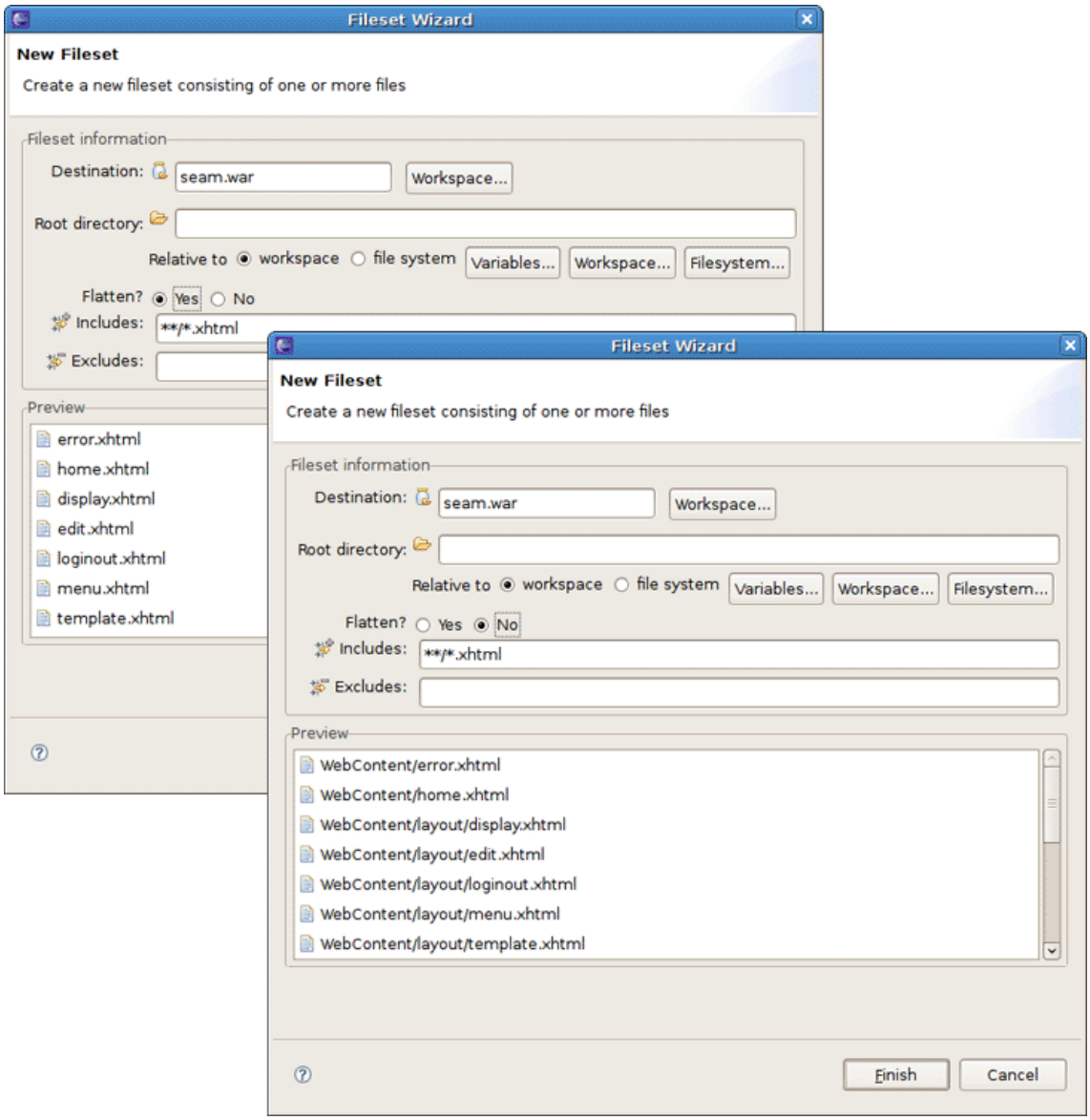

Figure 3.34. The FileSet with flattening and without it

3.2.2.3. Creating User Library FileSet

If you use user libraries in your projects you can also refer to these from project archives and have all the .jar/.zip files they refer included into the archive.

To add a new user libraries file set, call the right-click menu on the necessary archive and choose " New User Libraries FileSet" item

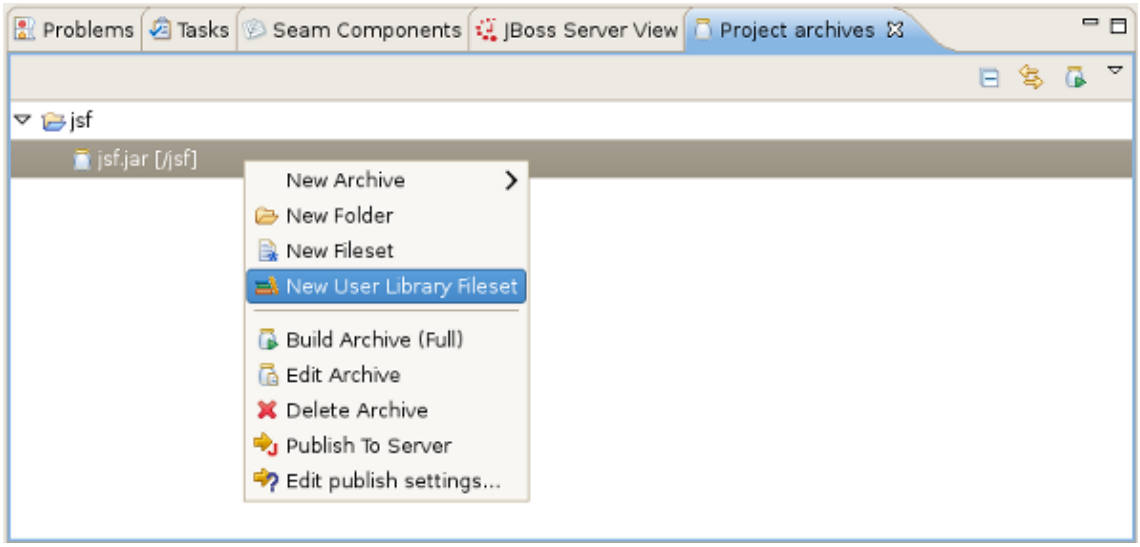

Figure 3.35. Adding New User Library Fileset

You can edit the existing user libraries as well using "User Libraries Fileset Wizard". Call the right-click menu on the library fileset and choose "Edit Fileset " item

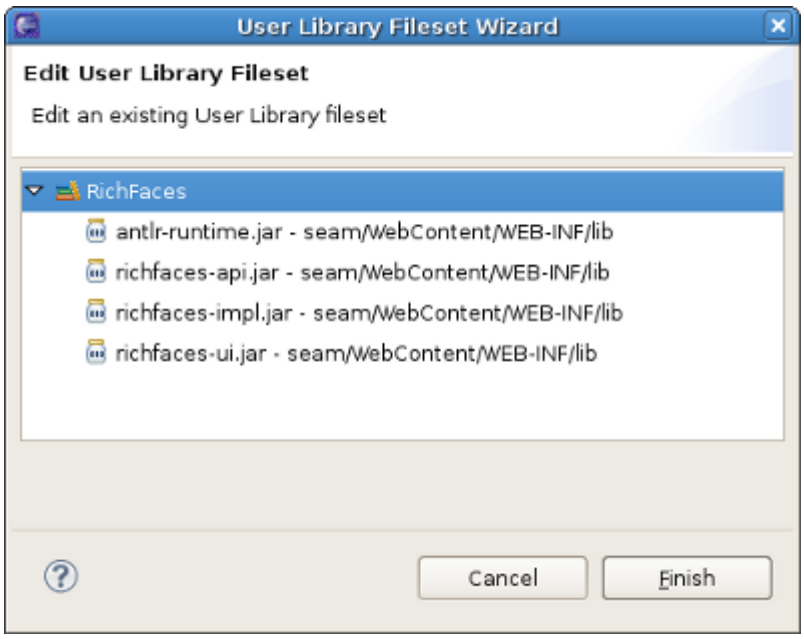

Figure 3.36. Editing User Library Fileset

3.2.3. Archive Actions

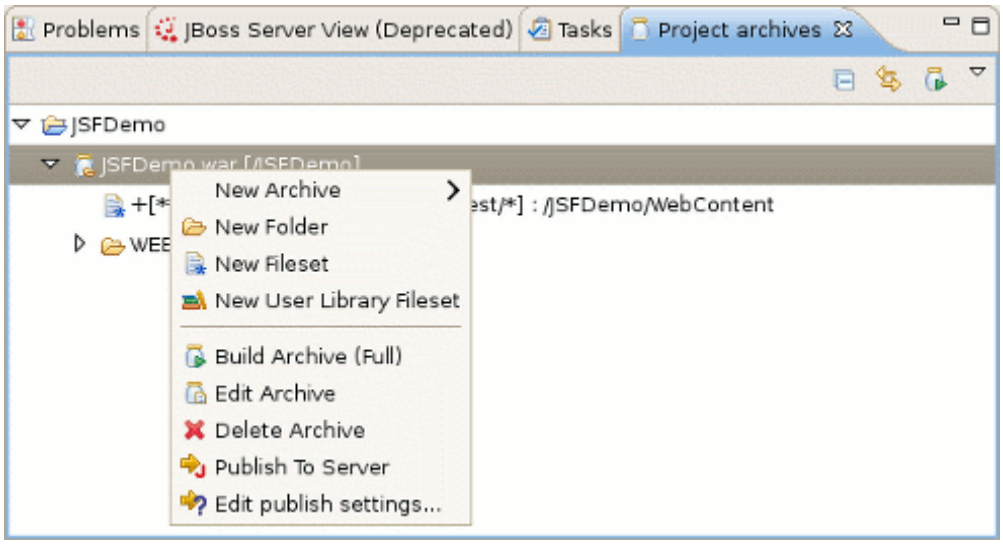

Figure 3.37. Context Menu on the Item

The context menu on the items in the view is extendable, but there are several that come standard.

Table 3.4. Context Menu on the Item

Name	Description
Build Archive (Full)	The action enabled only on top-level archives, which 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	Deleting node is standard action with deletion not needing an explanation
Publish To Server	The action means the ability to publish to a declared server
Edit publish settings	Gives possibility to edit an archive 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'll need to publish your application to a server. Here, we show you how to do it with the help of *Archives View*.

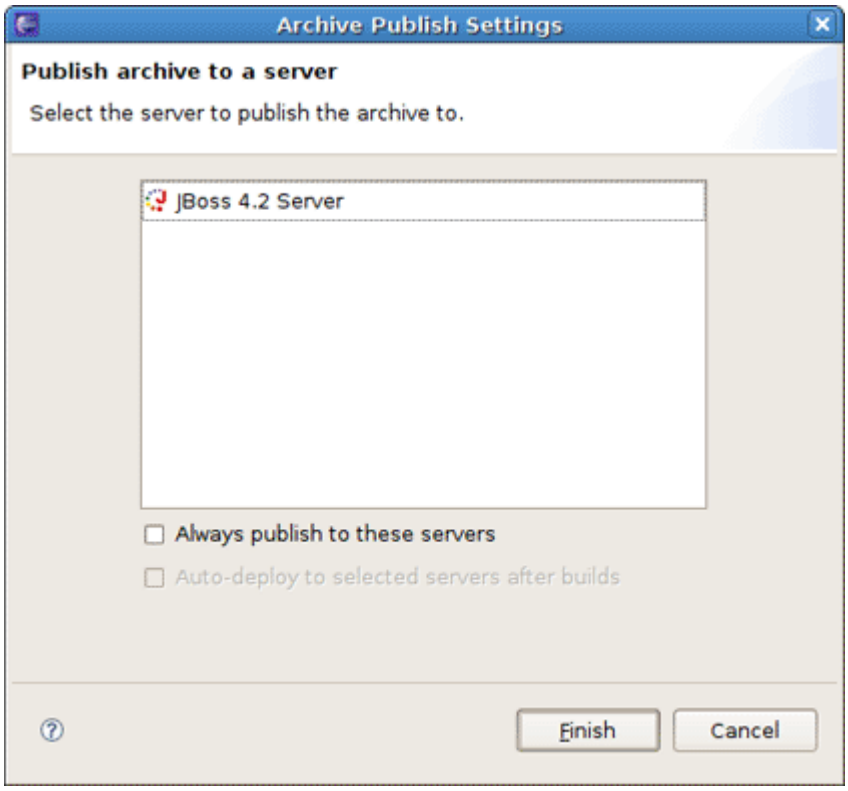

Figure 3.38. Context Menu on the Item

The dialog above appears after selecting *Publish To Server.* To simply publish once, you just select the server(s) that you want, and finish. If you want the *Publish to Server* action on that particular Archive to publish always to that set of servers, then check the appropriate checkbox. And finally, 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 really want 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 *Ant manual* [http://ant.apache.org/manual/index.html] to find more on how to build your applications with help of Ant.

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

At this point, you are guessed to be familiar with JBoss AS perspective and your next step now is to explore how to work 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. Web projects of JBoss Tools are Struts, JSF and Seam projects. All of them are called faceted projects. Thus, in this chapter we are going to tell you about facets the main benefit of which to provide proper structuring and packaging for 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. Most often, these facets either add to the project's classpath, enable a builder, or watch the project in some other fashion. Generally, every project concerned has at least one facet when it's 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 undergone some criticism as 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. Apparently, 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*.

4.2. Adding Facets to a Project

In this section we're going to consider the facets added by JBoss Tools and show how you can configure them in a project: add new ones or modify already existing configuration of the facets.

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...* and then *Web > Dynamic Web Project*.

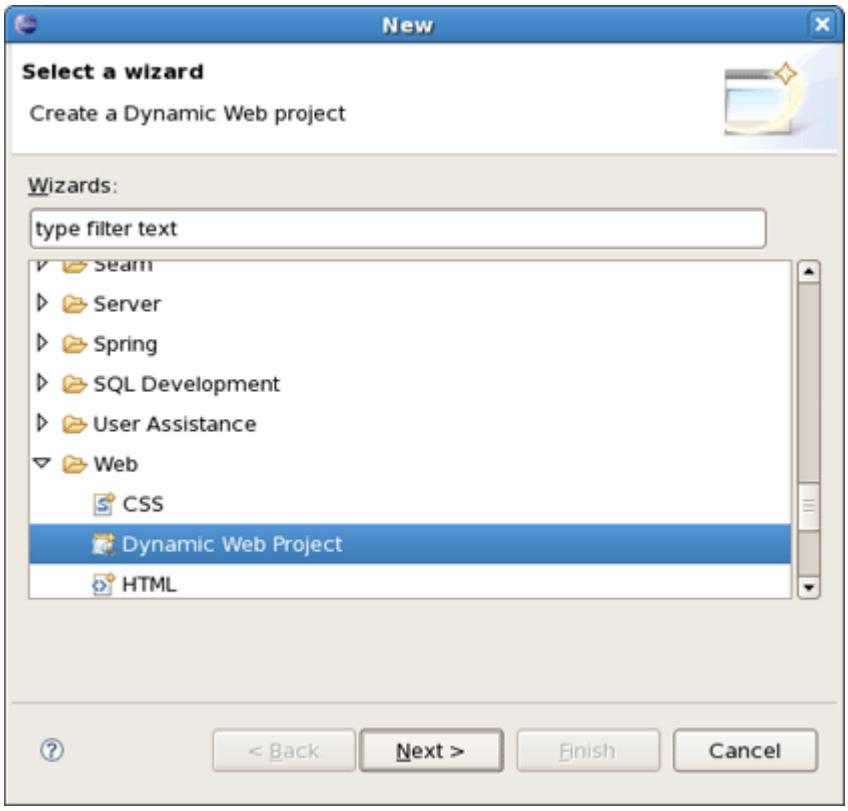

Figure 4.1. New Dynamic Web Project

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

The first page of most WTP projects allows you to target a specific runtime, representing 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 facet you might want.

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

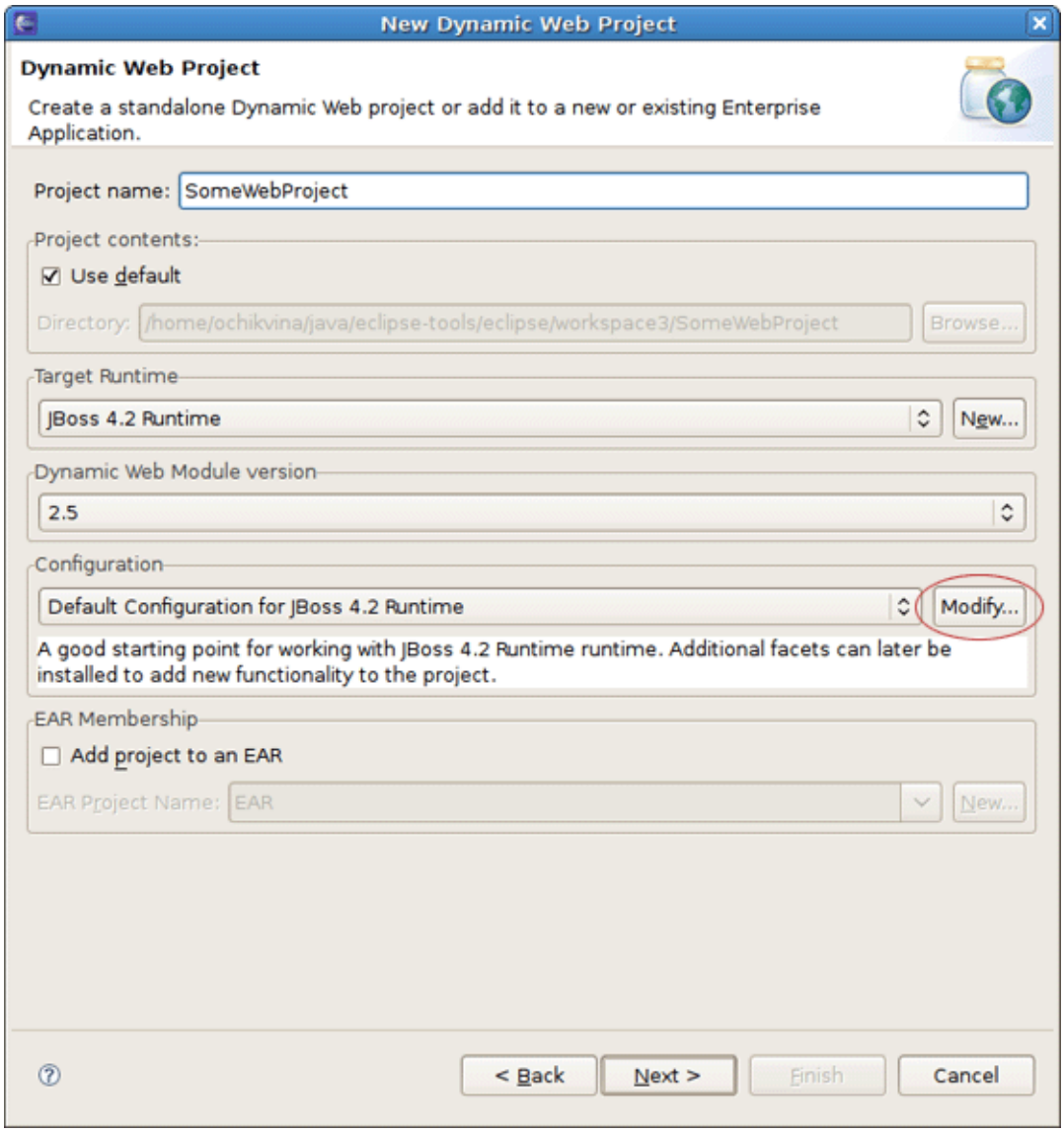

Figure 4.2. New Dynamic Web Project

Click on the *Modify* button next to Configuration section in order to open the wizard which allows you to modify a chosen configuration. The wizard looks as follows.

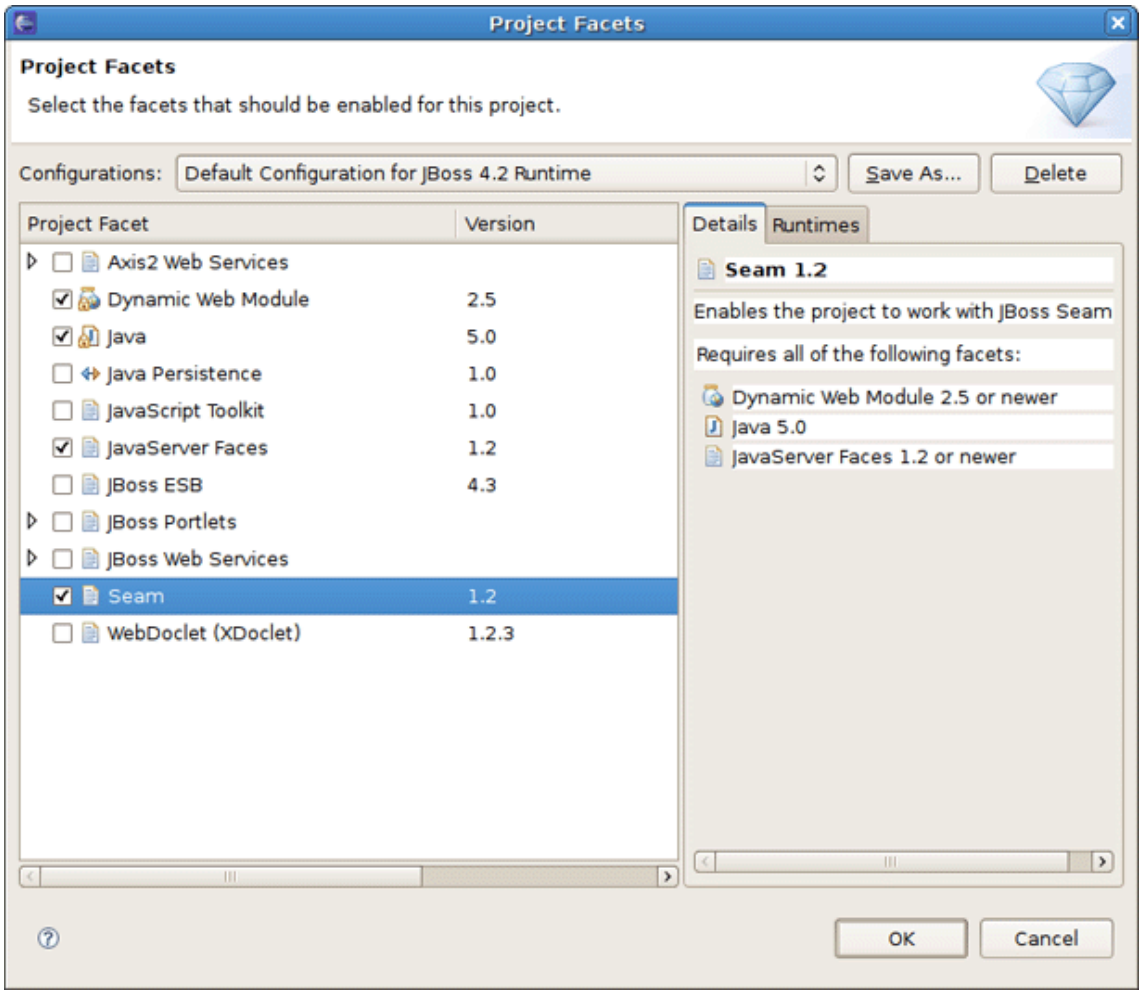

Figure 4.3. Project Facets Wizard

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
- JBoss ESB
- JBoss Portlets
- JBoss Web Services
- Seam

On this wizard page you can enable or disable any facet as well as change it version. What you should note here is that some facets or facets versions may conflict with each other. In case of incompatibility you'll be prompted about this in the combo box underneath.

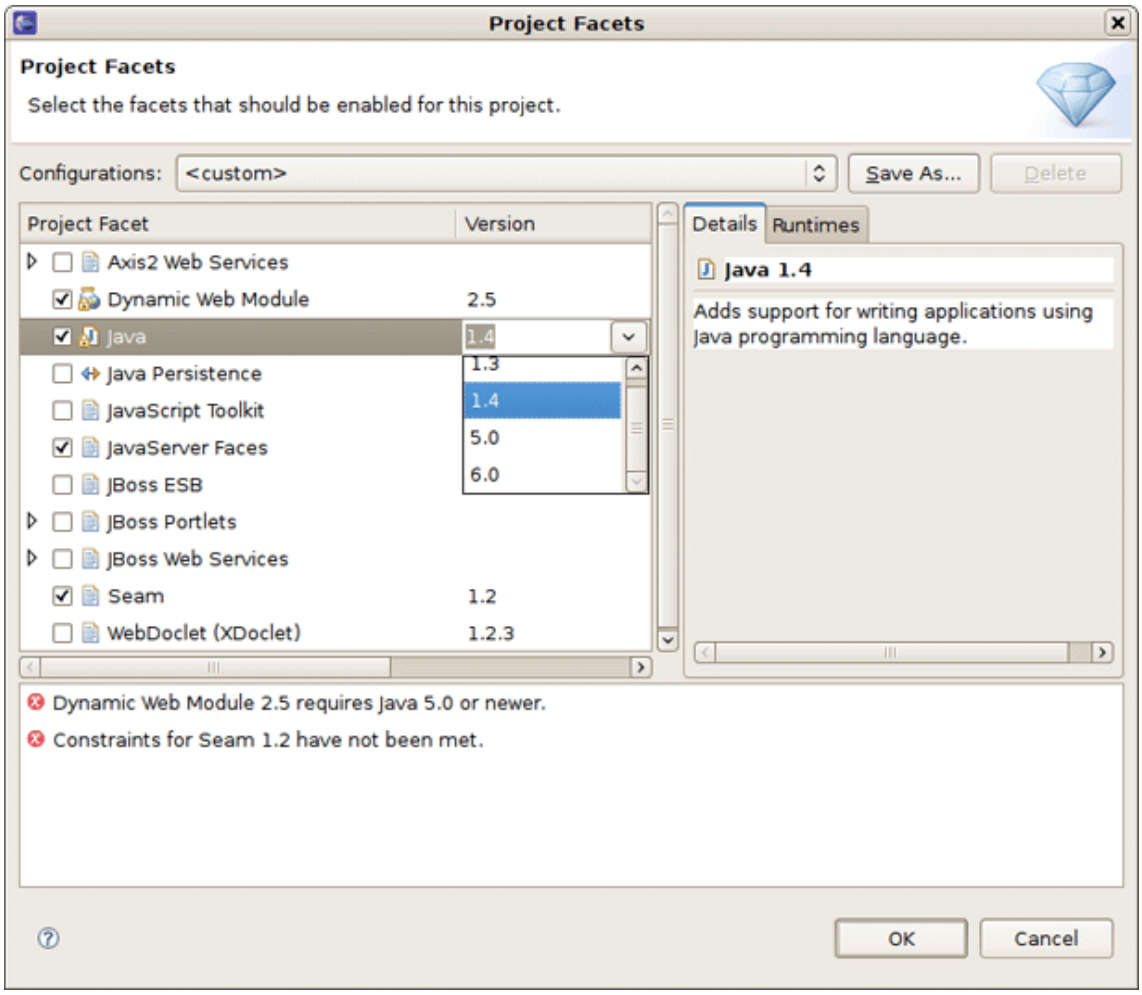

Figure 4.4. Facet Constraints

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

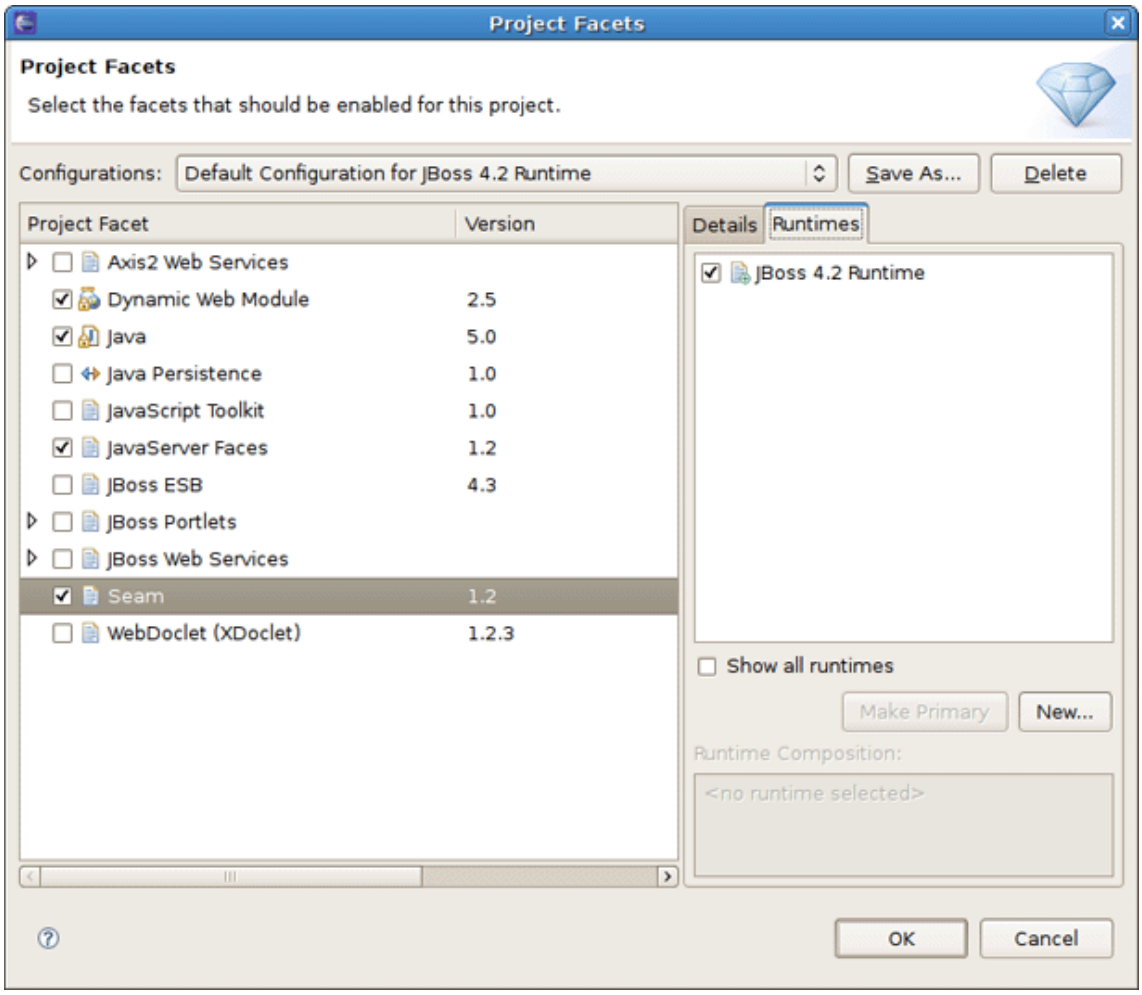

Figure 4.5. Runtimes on the Project Facets Wizard

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

Clicking on *OK* will save the chosen configuration of the facets and return you to the *Dynamic Web Project wizard*. Further pages in the wizard are specific to either the project type or the facets selected.

If you need to configure the facets for the existing project, you should bring up the context menu for selected project and click *Properties* and then *Project Facets*. This will bring up the familiar *Project Facets wizard*, where you can create your own custom facets configuration.

4.3. Relevant Resources Links

More on the WTP facets you can read 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.

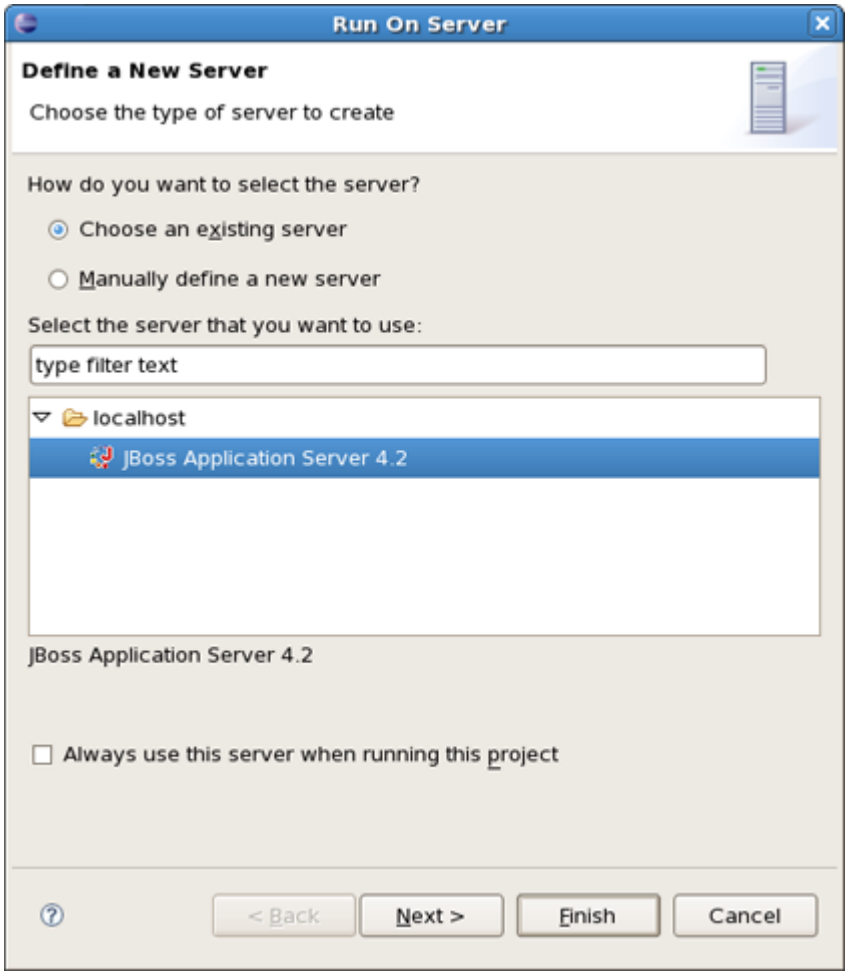

Figure 5.1. Define a New Server

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

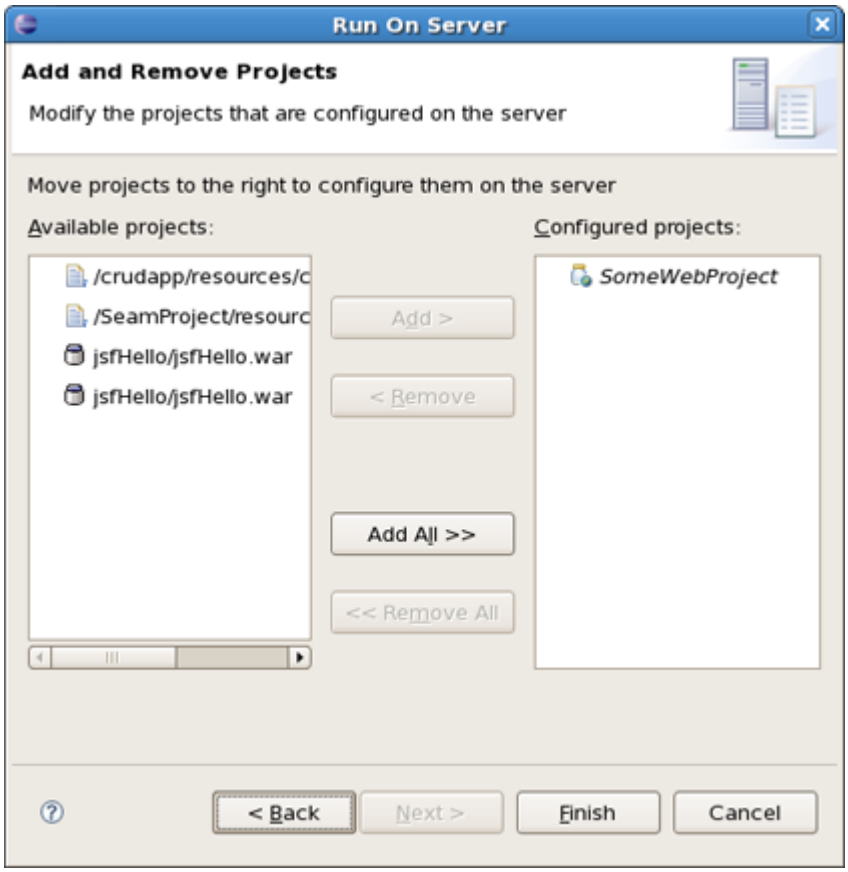

Figure 5.2. Add or Remove Projects

This page of the wizard also allows to undeploy modules from the server. For that choose proper module(s) from the right and click *< 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* and customize packaging yourself.

5.1.2. Deploying single files

Sometimes it becomes necessary to deploy one or more files to a server. For that in order not to do a full republish in the context menu of files a *Deploy To Server* option is provided that allows a single file deployment. To deploy these non-WTP files/projects right click on the file (*-ds.xml*, *.ear*, *.jar* etc.) and select *Deploy To Server* and it will be automatically deployed.

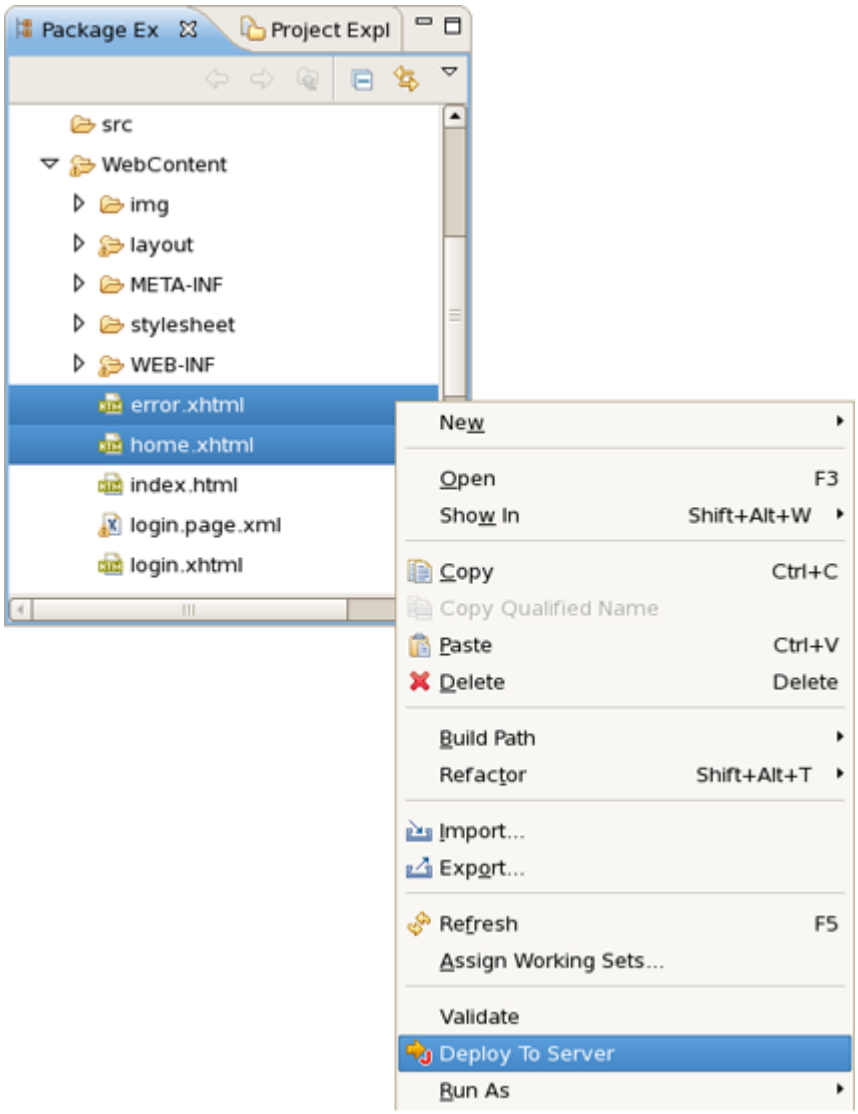

Figure 5.3. Deploy to Sever

The deployed files are listed side-by-side with other modules that are deployed to the server.

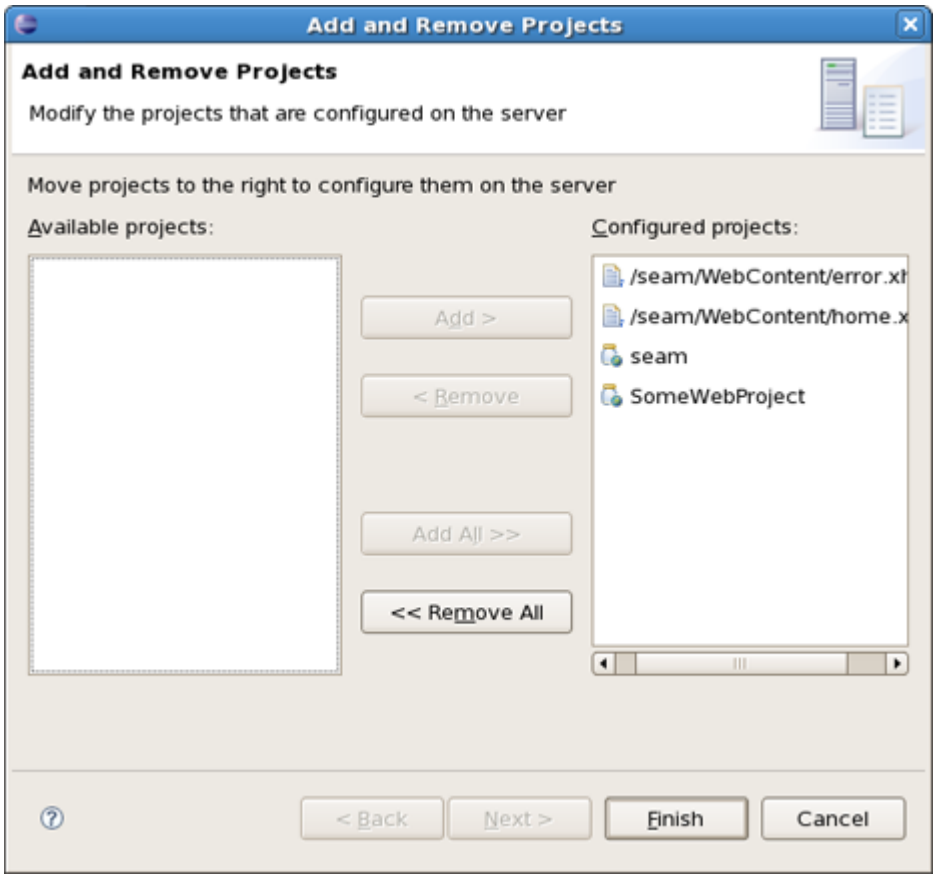

Figure 5.4. Deployed files on the Server

5.2. Deploying with JBoss Server View

As it has been already mentioned JBoss Server 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 JBoss Server View

In the top part of the JBoss Servers View like in the Servers View you should right click on a server and select the *Add and Remove Projects* menu item.

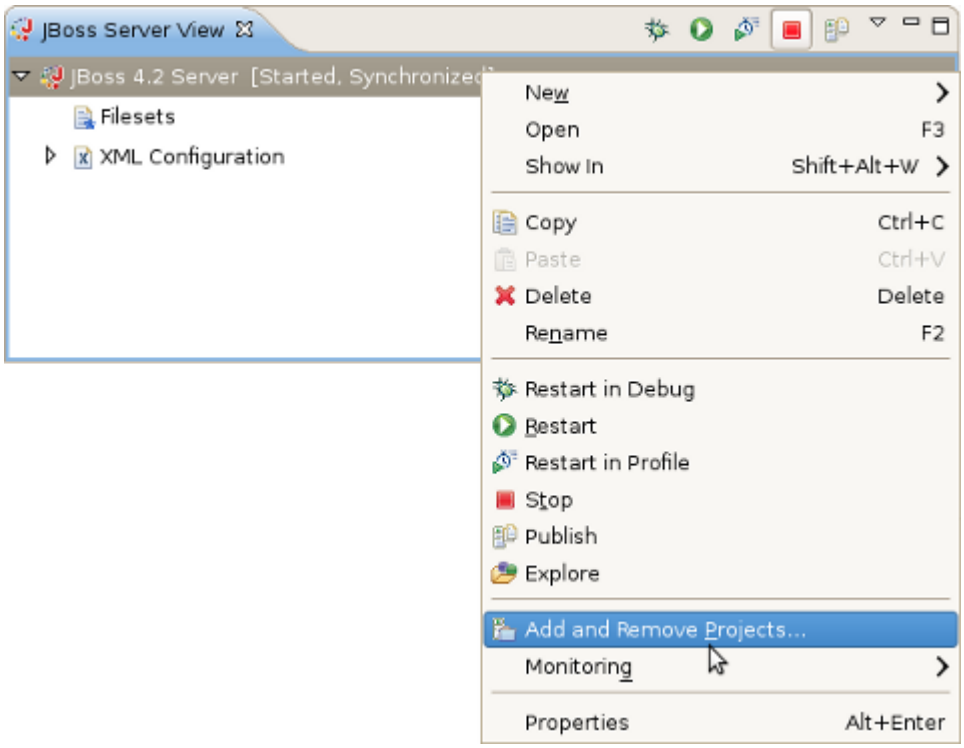

Figure 5.5. Add and Remove Projects

This will bring up a *familiar dialog* allowing 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*, it will be published according to the rules of that module type.

5.2.2. Bottom part of JBoss Server View

In the bottom part of *JBoss Server 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.

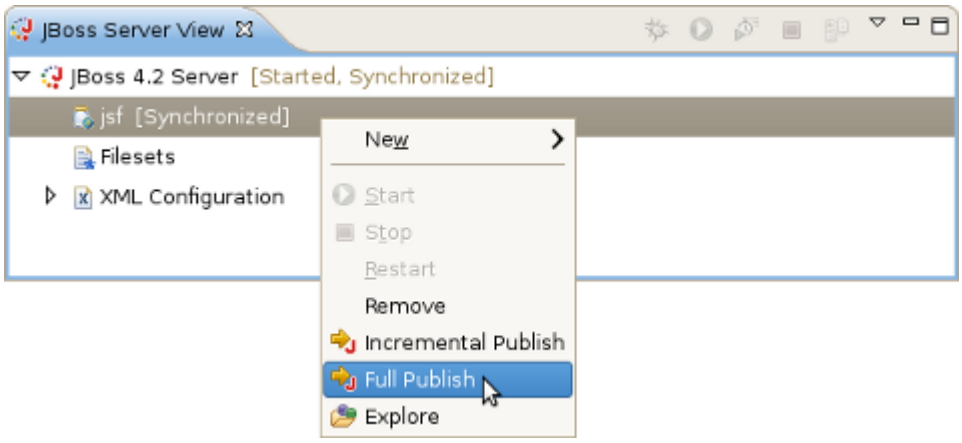

Figure 5.6. Full Publish

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 *Publishing to Server* in the Project Archives View section.

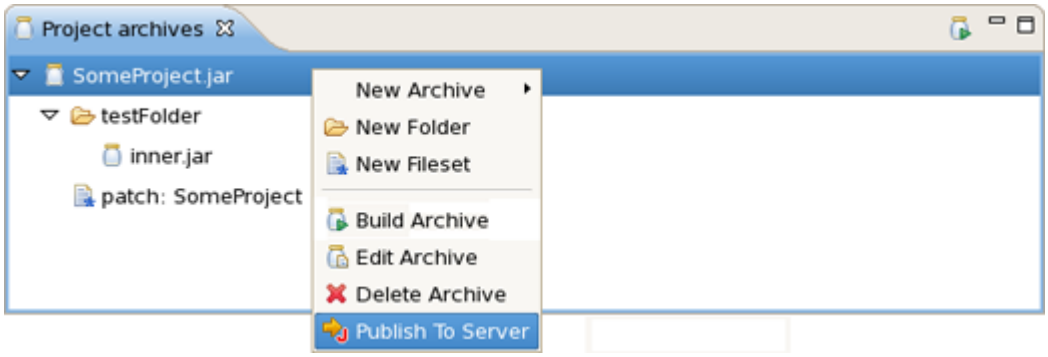

Figure 5.7. Publish to Server

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:

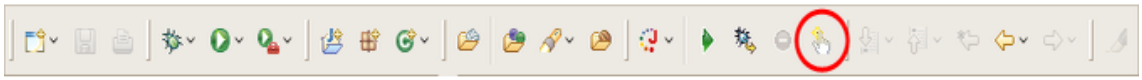

Figure 5.8. Finger Touch button

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 first download *TPTP 4.5.0 Runtime* [http://www.eclipse.org/downloads/download.php?file=/tptp/4.5.0/TPTP-4.5.0/tptp.runtime-TPTP-4.5.0.zip] 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.

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 *Profile As > Profile on Server* option in the context menu of the project.

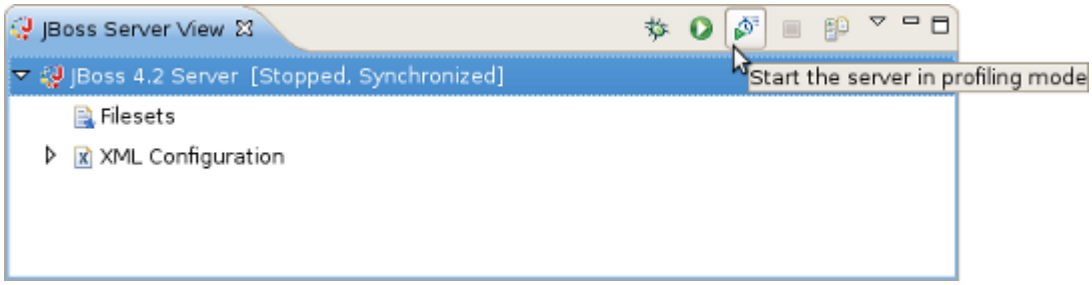

Figure 6.1. Start the Server in Profiling mode

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…*

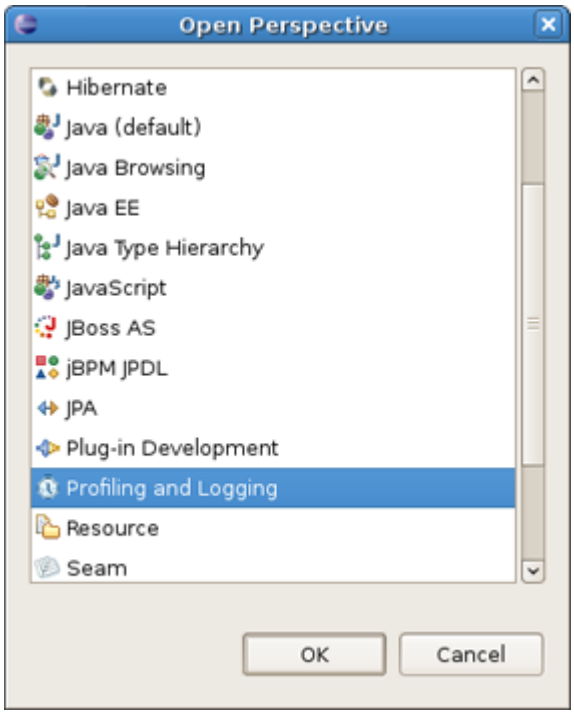

Figure 6.2. Profiling and Logging Perspective

6.2. Relevant Resources Links

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

In summary, this reference should help you to start with *JBoss AS* [http://www.jboss.org/jbossas] and get to know with functionality for work with it.