



**Installation and configuration of  
IBM Rational Application Developer V7  
for developing portal applications**  
**Part of a series on portal and portlet development**

By [Kapil Gambhir](#)  
Portal Tooling Technical Lead  
IBM Corporation

Level: Intermediate

August 2007

## Contents

Abstract .....	3
Introduction .....	3
Installation .....	3
Optional portal test environment installation .....	7
Silent automated installation alternative .....	7
Configuration .....	8
Development .....	8
Deployment .....	12
Environment .....	14
Other articles in this series .....	16
Resources .....	17
About the author .....	17

## Abstract

IBM® Rational® Application Developer Version 7 provides a holistic IDE (integrated development environment) for designing, developing, testing, and deploying enterprise portal or portlet applications for IBM® WebSphere® Portal Version 5.1 and 6.0. This version introduced an installer and slimmed-down server WebSphere Portal configuration that reduces the complexity without losing the important flexibility. This article covers the installation and configuration options in the Rational Application Developer V7 environment (also called the *workspace*) to suit the specific needs of the portal developer. It also describes best practices for maximum benefit from using this software.

## Introduction

Rational Application Developer V7 is a premium product that is part of the IBM® Software Delivery Platform (SDP). Its many benefits cut across various dimensions that affect developers. It is a true integrated development environment (IDE), because it provides a single environment for designing, building, testing, and deploying, all from the same workbench.

This version provides portal and portlet development tools to increase productivity and make rapid development possible. It is based on Eclipse 3.2, thus it takes advantage of the performance and usability improvements in Eclipse, which improves response time and lowers memory requirements. If you have used Rational Application Developer V6, you will find a dramatic improvement in performance and ease of use in V7. Most of the activities have been tuned, and most of the UI (user interface) tasks have been converted to jobs (thus they don't block the main UI thread). Progress monitors have been enhanced to provide timely and appropriate feedback.

Portal tooling in Rational Application Developer V7 is designed to work with IBM® WebSphere® Portal software (see Figure 1). Like any other product, there are some configurations that help the developer derive maximum benefit from the tool, based on a particular scenario. This article first touches on how to install the portal tooling from the Rational Application Developer installer, as well as the optional portal test environment, and then outlines some of the settings that help you work faster and more efficiently.

## Installation

Rational Application Developer V7 includes an installation interface called **Installation Manager** that you can use to install optional components, too. It can be installed on an instance of Eclipse, and the installer checks for plug-in version compatibility.

Choosing to install Rational Application Developer V7 (see Figure 1), brings up the IBM Installation Manager (Figure 2), which guides through the features to be installed, which

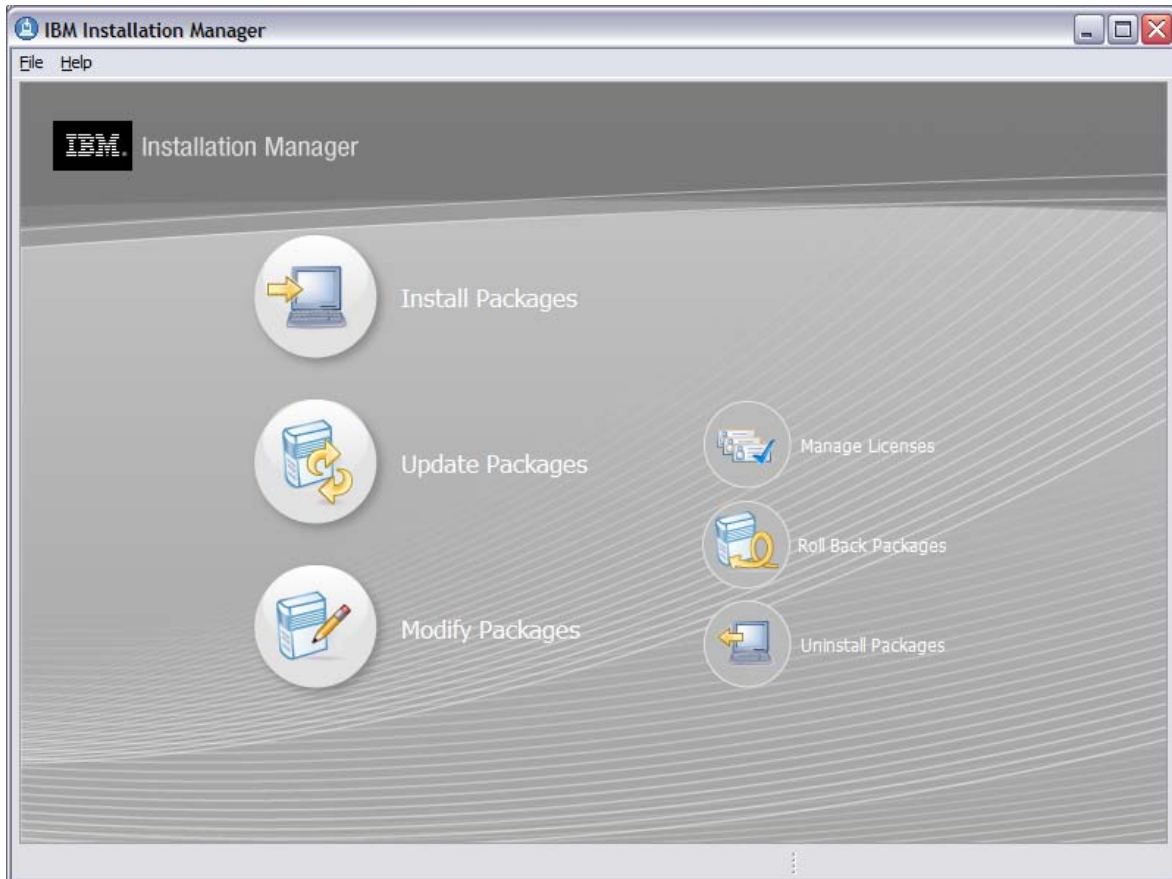
can be handpicked for installation. With Rational Application Developer V7 these features have been made more granular (with the dependency automatically taken care of) to choose from, while installing. This granularity has brought the advantage of allowing developers to keep the setup of the product suited to their individual needs. Another major shift is that developers who were using the beta version can directly upgrade to the released version without removing the beta.

Figure 1. Initial installation screen



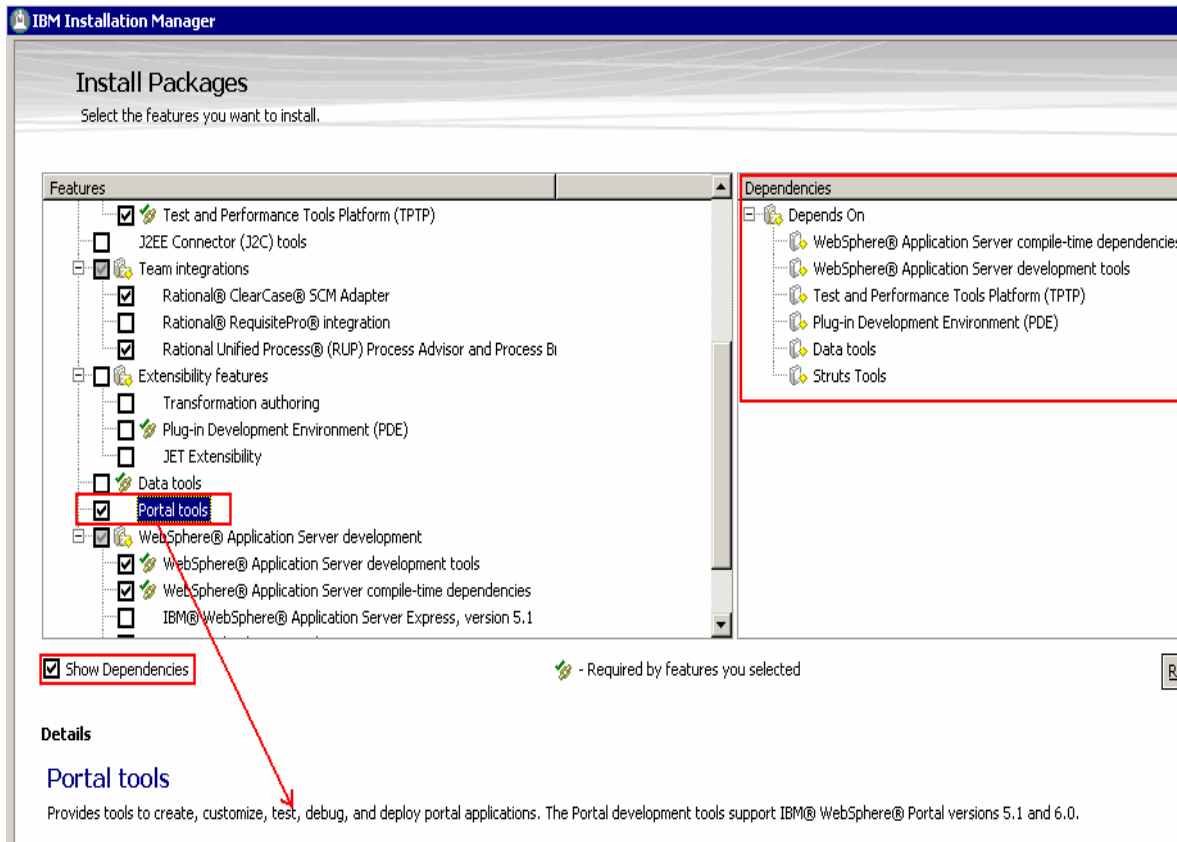
You can use the Installation Manager to uninstall, rollback, and modify the packages installed, as well to manage your licenses (see Figure 2). This enables you to add or remove packages that you don't need while working. This installer is based on the concept of a repository that can be either within your local file system or available on the Web.

**Figure 2. Installation Manager options**



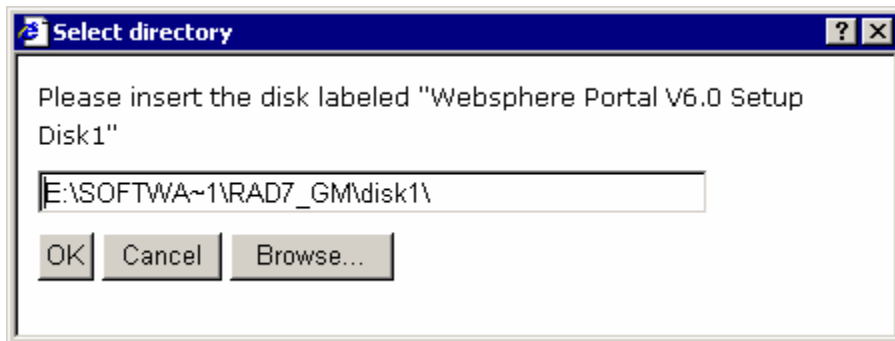
For portal tooling, select **Install Packages** and then **Portal tools** (the tools are unselected by default, as Figure 3 shows).

Figure 3. Select *Portal tools* if you want to install portal tooling



After installing Rational Application Developer, select the appropriate portal test environment version from the Installation Manager. This will prompt you for location of the first disk to start the test environment installation (Figure 4).

Figure 4. Popup window to specify the location of portal test environment



## ***Optional portal test environment installation***

A portal server can be installed as an optional component of Rational Application Developer V7 through the Installation Manager.

### **Differences between Version 6 and Version 7**

Rational Application Developer 6 provided a WebSphere Portal Test environment (V5), which was a slimmed-down version of the actual portal server and had limited support. This was a separately supported product and was managed through Rational Product Updater, which was the common application for managing product feature updates or rollback (including the optional ones) of Rational products.

With Rational Application Developer V7, the best concepts of a portal test environment have been carried forward and those limitations eliminated. It still offers automatic reloading of artifacts and hot code replacement (when it points to a local server) in addition to support for other critical functions of the server. The point of it being called a test environment is debatable, but it continues to be called that to be associated with all of the advantages of using the local server.

To manage the server startup time and memory use, there are scripts that tweak the server environment to achieve faster startup and lower memory use. These scripts are for a development environment and not supported for production purposes. They make the server cut down on lot of prepackaged portlet applications and other non-critical elements that are typically not required during development.

Rational Application Developer also offers tips to improve publishing time (see [Resources](#)). These guidelines help operate the portal server on a typical development machine without the resource demands of a full server.

## ***Silent automated installation alternative***

Although the GUI is quite intuitive and suitable for a single desktop installation, it is cumbersome and impractical for a wider user base. To solve the problem of installing and managing updates for a wider user base, Rational Application Developer provides a “silent install” option. This automates the whole task and works in the background.

By using a response file, you can install Rational Application Developer and WebSphere Portal server without going through the installation screens. Although the silent installation alternative can be used for a single machine or user by using a response file, its effectiveness for a wider base can be achieved by having a local server with local install and update images. The desktops can invoke the silent install (response file) to capture the image from the local (intranet) server and install them on local machines (see [Resources](#)).

## Configuration

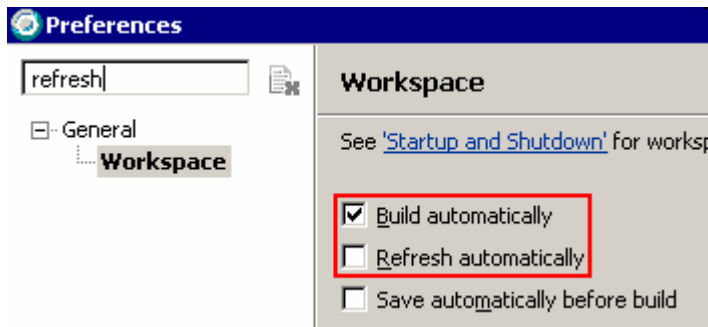
After the installation is complete and the product is ready to use, there are several important settings that can make things easier. Some of them relate to Rational Application Developer V7, and others are on the peripheral interface between it and the server (WebSphere Portal V6).

## Development

While developing portals and portlets, there are few settings that can enhance your productivity and make your work easier. Some of them are relevant only to a given situation, but others are valid for all scenarios. Initial settings are generic and valid for non-portal application development through Rational Application Developer V7.

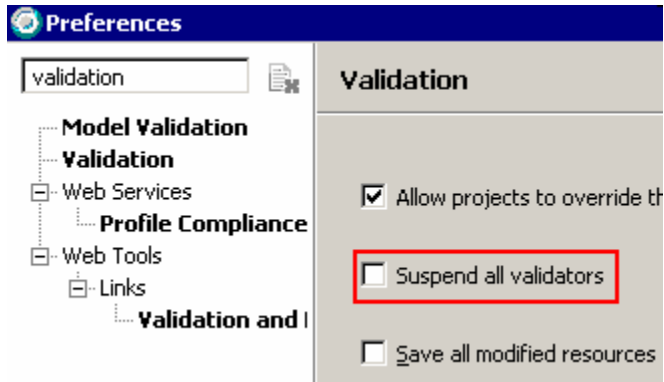
**Build and refresh automatically:** Values of these preferences (Figure 5) are dependent on the purpose and the user's preference. If the Save operation is taking a lot of time, then Automatic Build can be turned off. Similarly, to generally improve the responsiveness of the system, Refresh Automatically can be turned off (this function tries to synchronize between the workspace and file system image).

Figure 5. Options to build automatically, refresh automatically, both or neither



**Validators:** There are more than 30 validators working behind the scene, providing instant feedback about changes made to resources. All redundant validators can be switched off, and the rest can be used judiciously. These settings can also be project-specific. (See Figure 6.)

Figure 6. Validation options



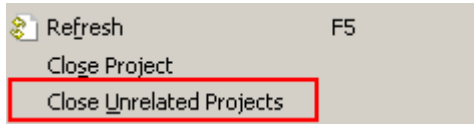
**Sharing preferences:** Normally within a development team, a set of preferences are reused by all developers. Also, it is desirable for developers to have the same set of preferences settings across their individual workspaces. This is achievable through exporting the Preferences file and importing it into the target workspace. Similarly, the break points can also be imported and exported between workspaces and developers. (See Figure 7.)

Figure 7. Preference settings export options



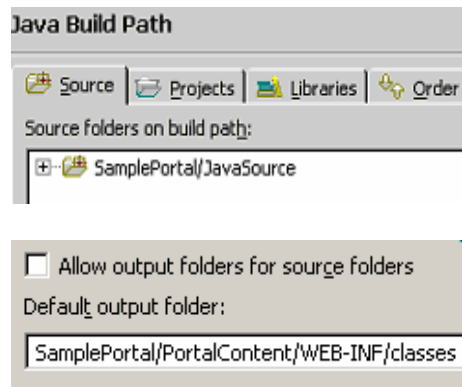
**Reducing active projects:** You can either manually select inactive projects and close them, or select any specific project and close all unrelated projects (Figure 8). It is advisable to break the workspaces into logical groups to lower the memory needs. The more a particular workspace is used, the more the metadata directory grows over time. Using a new workspace helps keep a check on its growth. Yet having too many workspaces takes away the advantage of using a single workbench. Therefore, it is advisable to switch to a new workspace only when the existing one becomes sluggish.

**Figure 8. Closing projects that you don't need**



**Hot swap:** For the hot swapping feature to work, the build folder of the project should be set as `<project_name>/WebContent/WEB-INF/classes` for portlets and as `<project_name>/PortalContent/WEB-INF/classes` (Figure 9) for portals. These are the default and the advisable values. They ensure that projects from the local workspace are used for deployment (for the local server when you use Run on Server).

**Figure 9. Setting up build path for portals**



## Portal project creation and synchronization

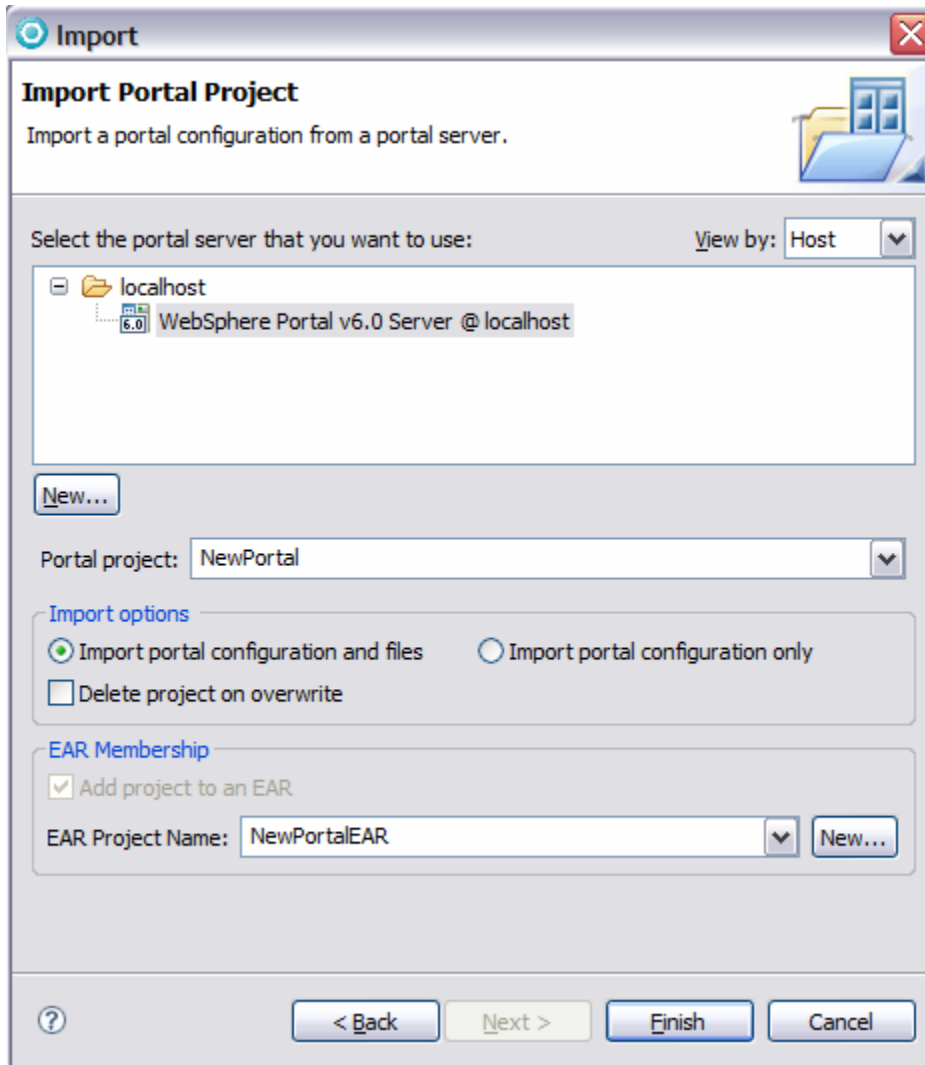
Instead of creating a new portal project in the workspace, it is advisable to import it from the server (**File > Import > Portal**). This ensures that the artifacts (themes, skins, and screens) and the configuration (layout and navigation) on the server are reflected in the workspace.

When you create a portal project from the **File > New > Project** menu (as a new project), it uses a static PortalContent setup and portal configuration, which might not be in synch with the default or customized settings on the server.

### Important

If a portal project already exists in the workspace, there is an option to import only the configuration for the portal project (see Figure 10). It is quite useful when resources on the server (portlets, for example) have been used on those pages and it is desirable to have the same configuration in the new workspace. Otherwise, these configurations will be overwritten the next time that you publish the portal.

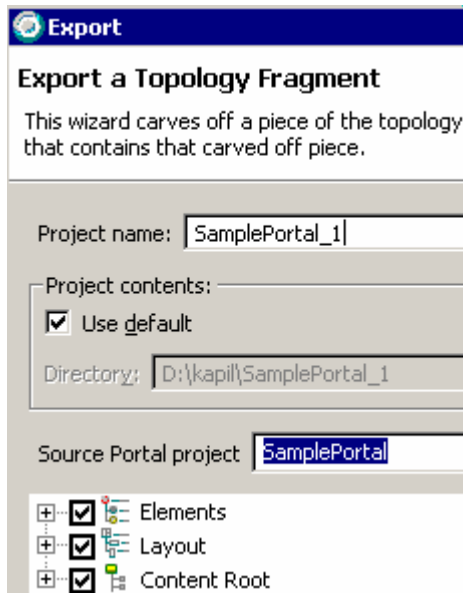
Figure 10. Importing only the configuration of a previous portal project



## Logical breakup of a portal project

A portal project is a full site development, thus it is a daunting task for one person to handle. Typically, it is best to break up medium to large projects into multiple, logical segments for separate development. Later, you aggregate them as a single project again. To do this, you use the **Portal Topology Fragment** feature, which can be exported (see Figure 11). You can select specific nodes to export, and then import the portal topology fragment later, when you are ready to combine them again. This is a new feature introduced in Rational Application Developer V7.

Figure 11. Portal Topology Fragment



## WebSphere Application Server 6 removal

In a new workspace, Rational Application Developer V7 automatically creates an instance of WebSphere Application Server 6 during installation. This is created on the first click of the server view, but only once (only once as long as the workspace exists). It can be removed if it is not desirable to develop applications targeted at the WebSphere server.

### ***Deployment***

While working with the server, there are multiple options which can drive the user experience of deploying portal artifacts to the server.

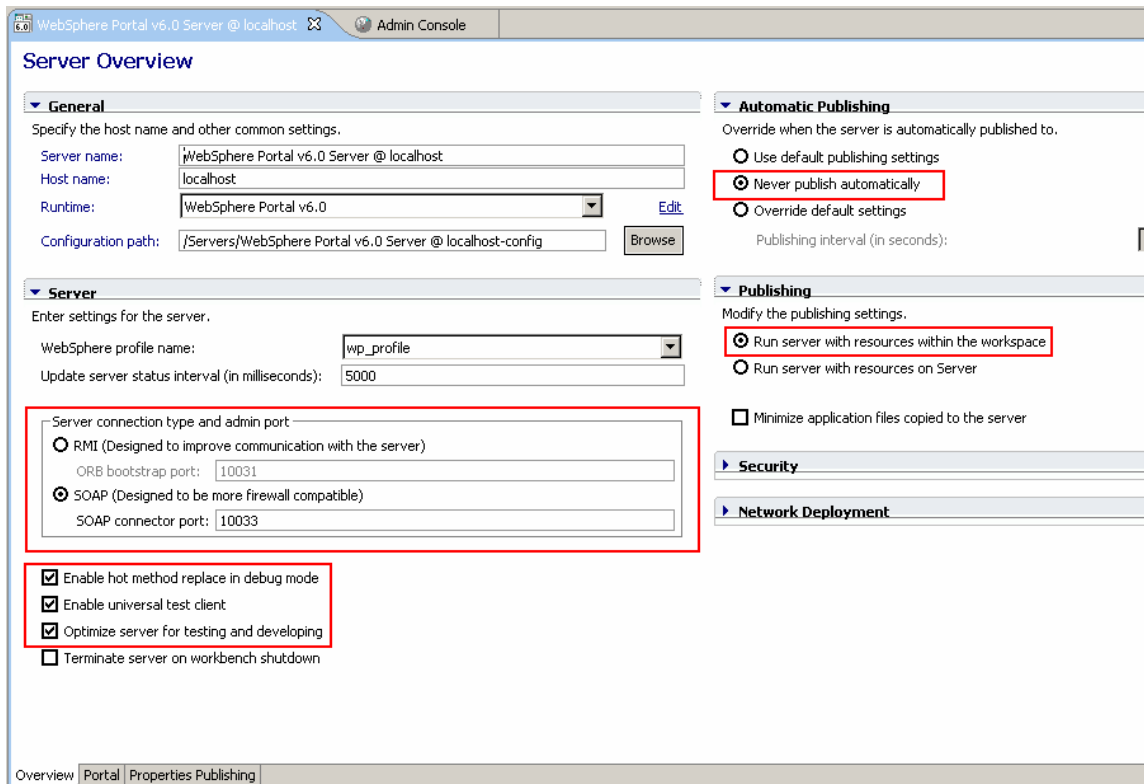
A WebSphere Portal server can be created in the server view of Rational Application Developer V7. It is a configuration placeholder of a server runtime which is used for any interaction of portal or portlet projects with the corresponding server. Double clicking of the server brings up the server configuration for corresponding instance. This has all the properties required to connect to the server instance and also defines various parameter used for publishing.

In the **Overview** tab of the server configuration, the following settings can be useful:

- **SOAP and RMI protocols:** These are the two protocols available to communicate with the server instance. RMI is optimized for Java to Java communication and is recommended for a local server scenario. SOAP is XML based and is recommended for a remote server scenario (firewall friendly). The corresponding ports are automatically deduced for a local server.

- **Automatic publishing:** For automatic publishing it is recommended to use “Never Publish Automatically”, this option never triggers off an automatic publish because of a change done on any resource in the project and avoids unsolicited publishing. If it is required to publish the changes whenever they are made (automatically) then “Override Default settings” can be used with utmost care, as it will trigger off a publish on every resource change in the project (after the duration specified in “Publish interval”).
- **Enable hot method replace in debug mode:** This option enables hot replace and facilitates just-in-time compilation while in debug mode. It speeds up the application deployment and also improves server startup time. This option affects only debug mode and has no effect when running the server in normal mode.

Figure 12. WebSphere Portal V6 configuration in Rational Application Developer V7

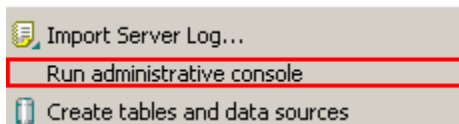


- **UTC option:** Universal test client (UTC) is a Web-based test client that can be used to test the Enterprise Java™Beans (EJBs) and other Java™ objects running on a local or remote server. It is more useful when developing EJB-based components and can be switched off if not needed (it is enabled by default).
- **Optimize server for testing and developing:** This sets Java™ Virtual Machine (JVM) parameters that improve the server startup time.
- **Remote server use:** If the desktop machine has less memory and portal or portlet deployment is desired, remote server can be used to cater to this case. It

needs to be prudently used as concurrency issues could cause unexpected results (for example, if everyone is publishing a portal project, the portal configuration of the latest publication is going to be visible).

- **Administrative console:** WebSphere administrative console can be invoked from Rational Application Developer (Figure 13) to configure other J2EE-related settings. It can also be used to specify other runtime settings for the WebSphere Portal runtime instance.

Figure 13. Launching the WebSphere administrative console



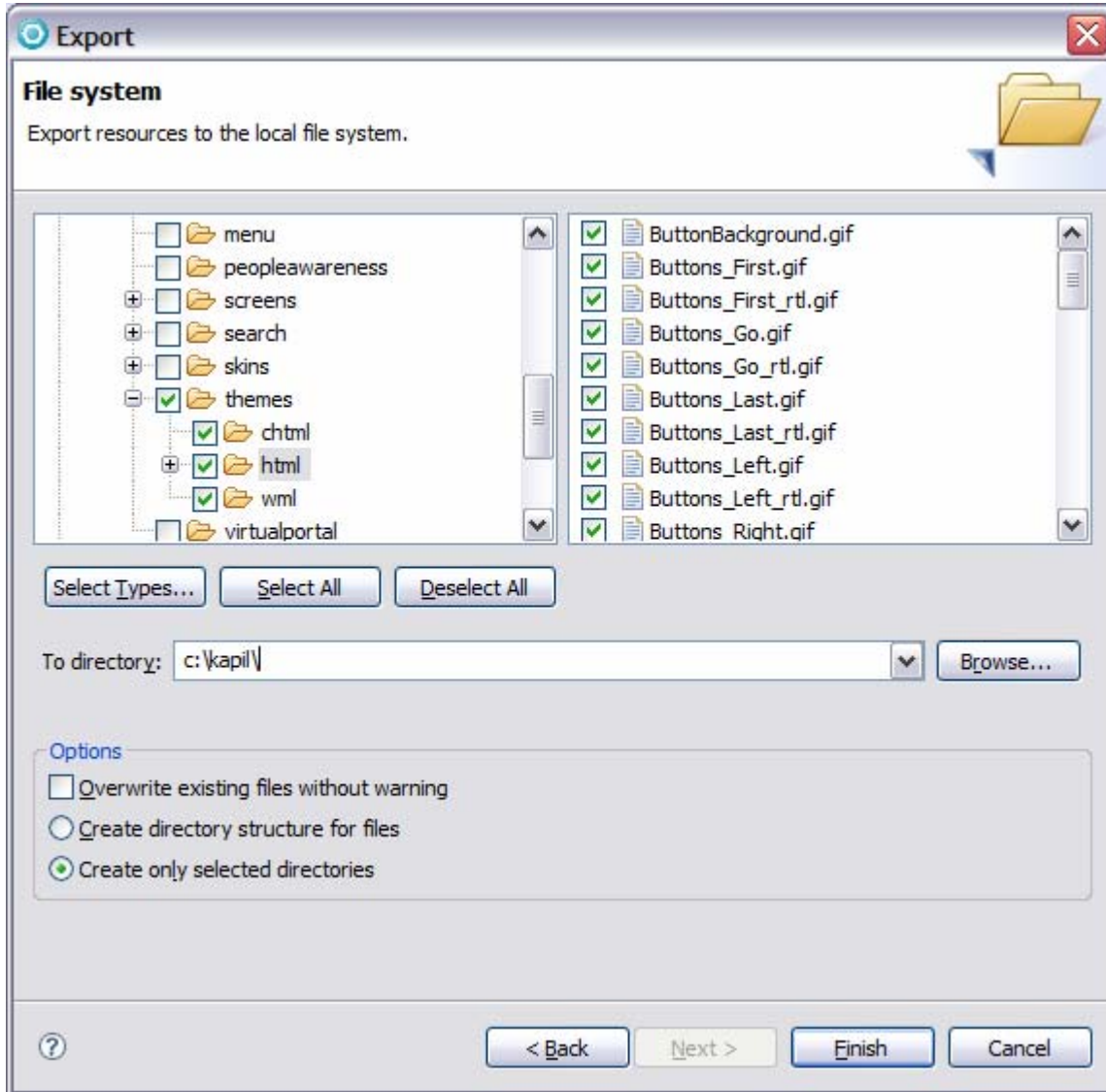
## Environment

So far, this article has covered the installation, development, and deployment phases. In a typical project, there are various environments defined (from a server runtime perspective) that are controlled at different levels.

- The least-controlled is the **development** environment, which typically resides on the developer's machine to facilitate development and testing of the artifacts for the relevant environment.
- After the project is in a better shape, it can be deployed to a more-controlled **staging** environment (typically synonymous with *production* environment from an infrastructure and topology perspective). Considering that this would be a shared instance, it typically resides on a remote server.
- The highest control is for a **production** environment, which is the set of machines being used live by the end users of the portal.

For the **staging server**, portlet developers can create a remote server instance and use **Deploy** (instead of **Run on Server**) to publish the portlet project to the server. Unlike the Run on Server alternative, Deploy publishes only the artifacts and does not change with the portal configuration. Similarly, for portal projects, if the configuration needs to be retained, the portal project can be exported and then deployed on the portal server by using the WebSphere administrative console. The **Portal Export** tool in Rational Application Developer V7 offers an option to export only themes and skins if you don't want to export the whole wps.ear file. For selective exports, such as themes only, choose **Export > General > File System**, and then select the specific resources, which can be patched on the server.

Figure 14. Selective exporting

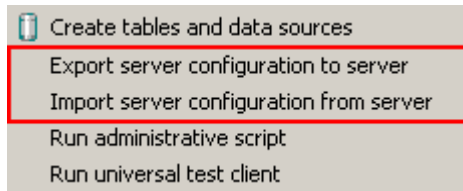


Typically, no direct publishing should be attempted to the **production environment** from Rational Application Developer V7 and the changes should be published in a very controlled manner (through scripts which are well tested). For a production environment, a utility called **Release Builder** can be used to compare the staging and production environments and decide which changes must be pushed to the server.

From a backup perspective, the portal configuration and server artifacts (part of the profile directory) should be archived. A copy of the WAS profile directory (which is the dynamic part of the portal artifacts) and the portal configuration (which is maintained in the portal DB) should be backed up for this purpose. The WAS profile can be exported or imported by right-clicking the server and choosing **Import [or Export] server**

**configuration to server** and storing it in the workspace. It can be version-controlled with other files in the workspace and can be restored somewhere else, with the benefit being that this method is file-system and OS-neutral.

**Figure 15. Importing or exporting the server configuration**



## Other articles in this series

Check the [One-stop guide to portal and portlet application development using Rational Application Developer V7 and WebSphere Portal V6](#) for summaries of the nine other articles in this series and links to download each of them in PDF format. Topics covered include the predevelopment resources, portlet tooling features, portal design tools, and the testing, deploying, and debugging capabilities of Rational Application Developer.

## Resources

### Learn

- Get information on IBM Rational Application Developer and how to use it:
  - [What's new in IBM Rational Application Developer V7.0](#)
  - [IBM Rational Application Developer Version 7.0 Information Center](#)
  - [Rational Application Developer](#) page on IBM® developerWorks®
- Read [Automating installation and update of an Eclipse-based IBM Rational Software Development Platform product from a network on Windows](#), a developerWorks article by Michel R. Morin and Chris Jeffs (August 2006), which explains how to automate the tasks of silently installing and updating an Eclipse-based IBM Rational Software Development Platform product on Windows, so that administrators who want to install the product on multiple machines over a network can run these tasks in a single step.
- Read [Installing and configuring WebSphere Portal V6.0 Servers for development with Rational Application Developer V7.0 and Rational Software Architect V7.0](#), a developerWorks article by Donal Riordan (March 2007, for more about tweaking portal server for local installation.
- Study [Best practices: Developing portlets using JSR 168 and WebSphere Portal](#).
- Read [Creating a new portal](#), an IBM® developerWorks® series of articles.
- Get product documentation for WebSphere Portal in the [IBM WebSphere Portal Version 6.0 Information Center](#).

### Discuss

- Participate in the Rational Software Architect, Data Architect, Software Modeler, Systems Developer, Application Developer and Web Developer [forums](#) and the [WebSphere Portal forum](#), where you can ask questions, share your experiences, and discuss problems and solutions.

### Download

- Get the [evaluation version](#) of Rational Application Developer.
- Get WebSphere evaluation versions from the [WebSphere downloads](#) page.

## About the author

Kapil Gambhir is the Technical Lead for the Portal Tooling component in Rational Application Developer in the WebSphere PortalLC group at India Software Labs in Delhi, India.. He has overall of 11 years of experience in the IT industry and has experienced a wide gamut of technologies, industries and platforms. He has keen interest in the latest and disruptive technologies.