

Rational Application Development certification prep, Part 1: Workbench basics

Skill Level: Introductory

[Mr. Bobby McChesney \(mcchesne@us.ibm.com\)](mailto:mcchesne@us.ibm.com)
Senior Learning Specialist
IBM

28 Feb 2006

Prepare for the IBM Certification Test 255, Developing with IBM Rational Application Developer for WebSphere Software V6. This tutorial covers setting workbench preferences, working with views and perspectives, use of the import and export wizards, using the Help features to aid in development activities, the resource Local History feature, and managing your workspaces in Rational Application Developer for WebSphere® Software. It is the first tutorial in a [series of seven tutorials](#).

Section 1. Before you start

About this series

Rational® Application Developer for WebSphere Software is the IBM Software Development Platform that allows you to quickly design, develop, analyze, test, profile and deploy Web, Web services, Java™, J2EE, and portal applications. This series of seven tutorials helps you prepare to take the IBM certification Test 255, Developing with IBM Rational Application Developer for WebSphere Software V6 to become an IBM Certified Associate Developer. This certification targets entry level developers and is intended for new adopters of IBM Rational Web Developer or IBM Rational Application Developer for WebSphere Software V6.0, specifically professionals and students entering into Web development using IBM products.

About this tutorial

This tutorial is the first in the series designed to help you prepare for the IBM Certification Test 255: Developing with IBM Rational Application Developer for WebSphere Software V6. This tutorial highlights features in the product, but does not replace the knowledge and familiarity that you obtain from using the product on a regular basis. This tutorial takes a basic approach to using the Rational Application Developer for WebSphere Software workbench. It begins with starting the workbench and presents workbench basics that simplify a developer's every day tasks. After you complete this tutorial, continue with the second tutorial, which presents an overview of the Java development environment that is included with workbench.

Objectives

After completing this tutorial you will know how to set workbench preferences, manage workspaces, and how to work with views and perspectives. You will also become familiar with the import and export wizards, and the local history and help features.

Prerequisites

This tutorial is written for developers whose skills and experience are at a beginning to intermediate level. You should have a general familiarity with using an integrated development environment.

System requirements

To run the examples in this tutorial, you need to install Rational Application Developer for WebSphere Software or Rational Web Developer for WebSphere Software. Download a free trial version of [Rational Application Developer for WebSphere Software](#) if you don't already have a copy of it.

The hardware and software requirements for this software can be located at [IBM Rational Application Developer System Requirements](#).

Section 2. Getting started

This tutorial assumes you have installed a Rational Software Development-based workbench. The Rational Application Developer family of products is based on the

Eclipse workbench. For the sake of brevity, the Rational Application Developer for WebSphere Software is referred to as Application Developer.

An installation directory for the Application Developer workbench was specified as part of the installation process. Refer to this directory as `<RAD_Install>`.

This tutorial begins with a description of how to open the workbench. If you installed the Windows version of Application Developer, the Windows Start menu has an entry for it. To start the workbench from the Windows Start menu, select:
Programs > IBM Rational > IBM Rational Application Developer V6.0 > Rational Application Developer.

On either the Linux or the Windows versions of the application, open a command window and start it from the directory:

```
<RAD_Install>.
```

On Windows, execute:

```
rationalsdp.exe.
```

On Linux, execute:

```
rationalsdp.sh.
```

Because Application Developer is based on Eclipse 3.0, there are options you can specify on the command line when you start the workbench to modify its behavior or enable features. Some of the parameters are inherited from the Eclipse base and some are added by the IBM Rational Software Development Platform plug-ins. Two good parameters to know are:

- `-data <workspace directory>` (opens the workbench to the specified workspace directory)
- `-showlocation` (shows the workspace directory in the workbench title bar)

A new feature in Eclipse 3.0 is the ability to switch workspaces from within the workbench. If you need to switch workspaces while in the workbench, just select **File > Switch Workspace** from the main menu bar and you can switch to an existing workspace, or create a new workspace. The workbench automatically closes and reopens into the indicated workspace.

The first time you open Application Developer in a new workspace location it creates a `.metadata` directory in that workspace location, which is the workbench metadata directory. In most cases you should never need to access this directory directly. After opening a new workspace, the Rational Software Development Platform Welcome page opens.

Figure 1. The Workbench Welcome Page



The Welcome Page provides access to various features. The six icons in the center of the page allow access to the Overview, What's New, Tutorials, Samples, First Steps and Web Resources content. The icon that looks like a person in the lower right corner of the page is the Enable Roles button.

The **Overview** is a *must see* if you are new to the Rational Software Development Platform. It contains links to a Flash Player tour of the product and links that describe the Java, Web, EGL, XML, Team programming, and Test development tools in the workbench. The workbench contains many simulations, some with audio, that explain how to use the workbench tools.

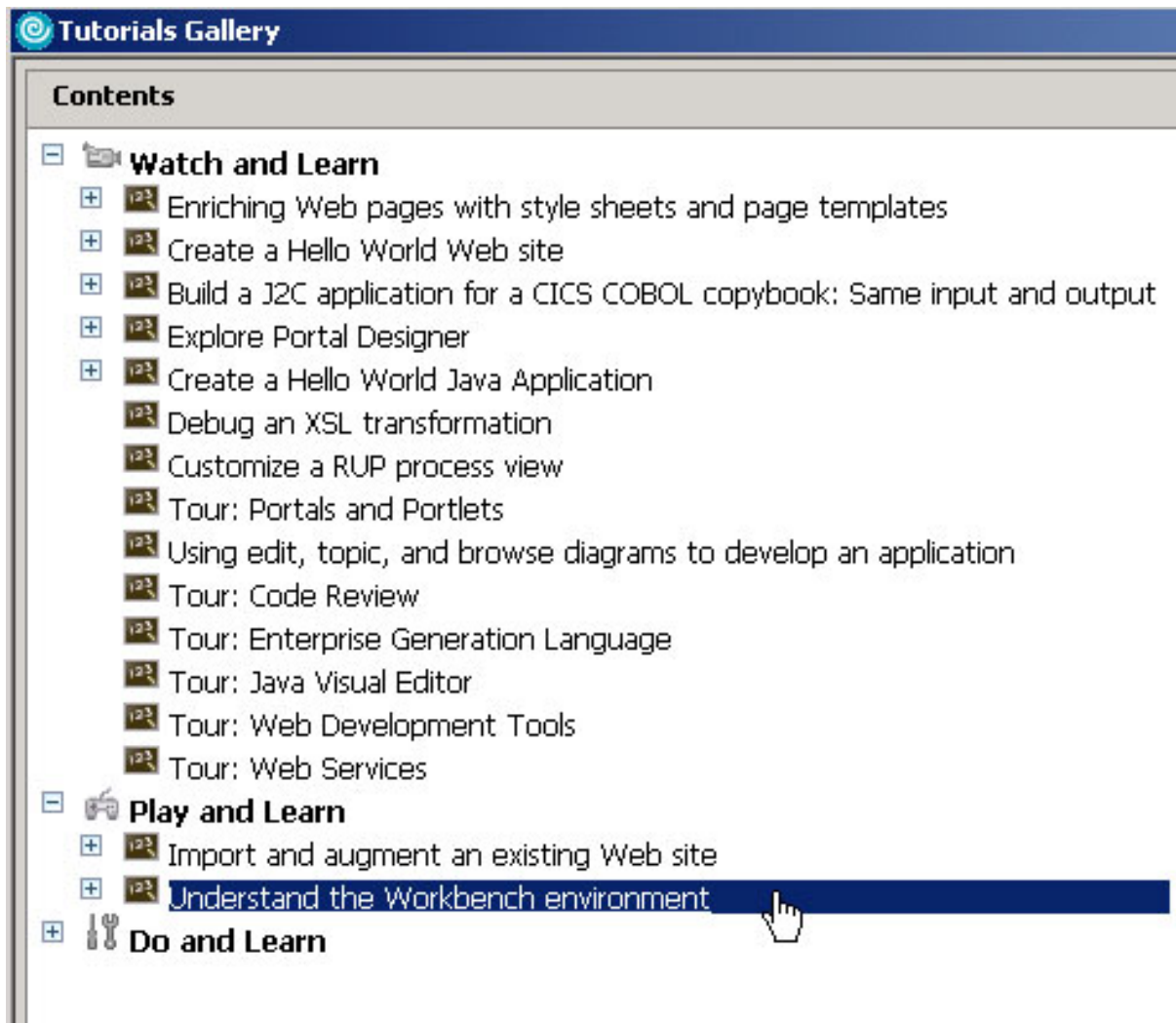
What's New (the star icon next to the Overview icon) highlights features for developers that might be familiar with the WebSphere Studio family of products.

Figure 2. What's New



Tutorials (click the chalkboard icon next to the What's New icon) contains more in-depth content that describes how to perform development tasks. Inside Tutorials, there is a Launch the Tutorials Gallery option that includes **Watch and Learn**, **Play and Learn**, and **Do and Learn** tasks that can accelerate your workbench-skill learning curve.

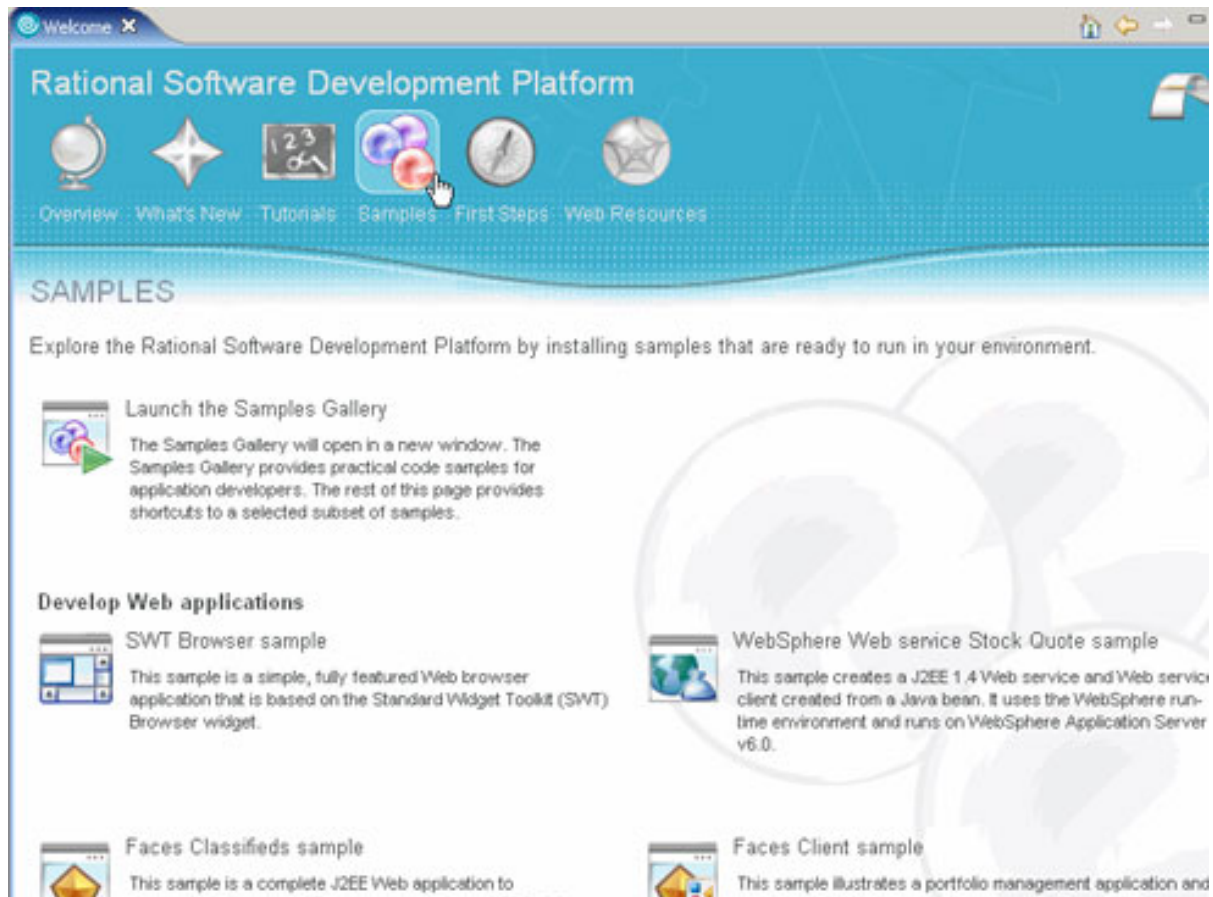
Figure 3. Launch the Tutorials Gallery



Be sure to watch the "Understand the Workbench environment" **Play and Learn**, which is an excellent companion to this tutorial.

Samples contains sample code for applications based on Swing, SWT, Portal, Faces, Faces Clients, Web Services, and J2C Connector Architecture, just to name a few. It is amazing how many people are not aware of all of the learning tools that are available in the Application Developer product! It definitely is worth your while to take the time to see all of the additional content that is shipped with the Application Developer workbench.

Figure 4. Samples



First Steps is a valuable tool for programmers who are new to J2EE development, need to migrate from WebSphere Studio Application Developer to Application Developer, or want to learn about the Import feature in the workbench. You will learn more about importing content later.

The last link in the row, **Web Resources**, takes you to the Web sites of technologies, publications, standards, and IBM tools.

Figure 5. Web Resources



Finally, we have that person icon, the Enable Roles button. It is the direct response to feedback from customers who used Eclipse 2.0 or the WebSphere Studio family of products. Many developers thought that even though a development tool had features to support multiple roles representative of tasks on a J2EE development team, including wizards and toolbar icons for tasks they did not perform *cluttered* their workbench or made the workbench more difficult to use and navigate. The icon lets you enable *Capabilities* in the product based on a developer's role! An Enable roles panel displays when the icon is clicked, as in Figure 6.

The 12 defined roles are:

- Advanced J2EE
- Database Developer
- Eclipse Developer
- EGL Developer
- Enterprise Java
- Java Developer
- Team

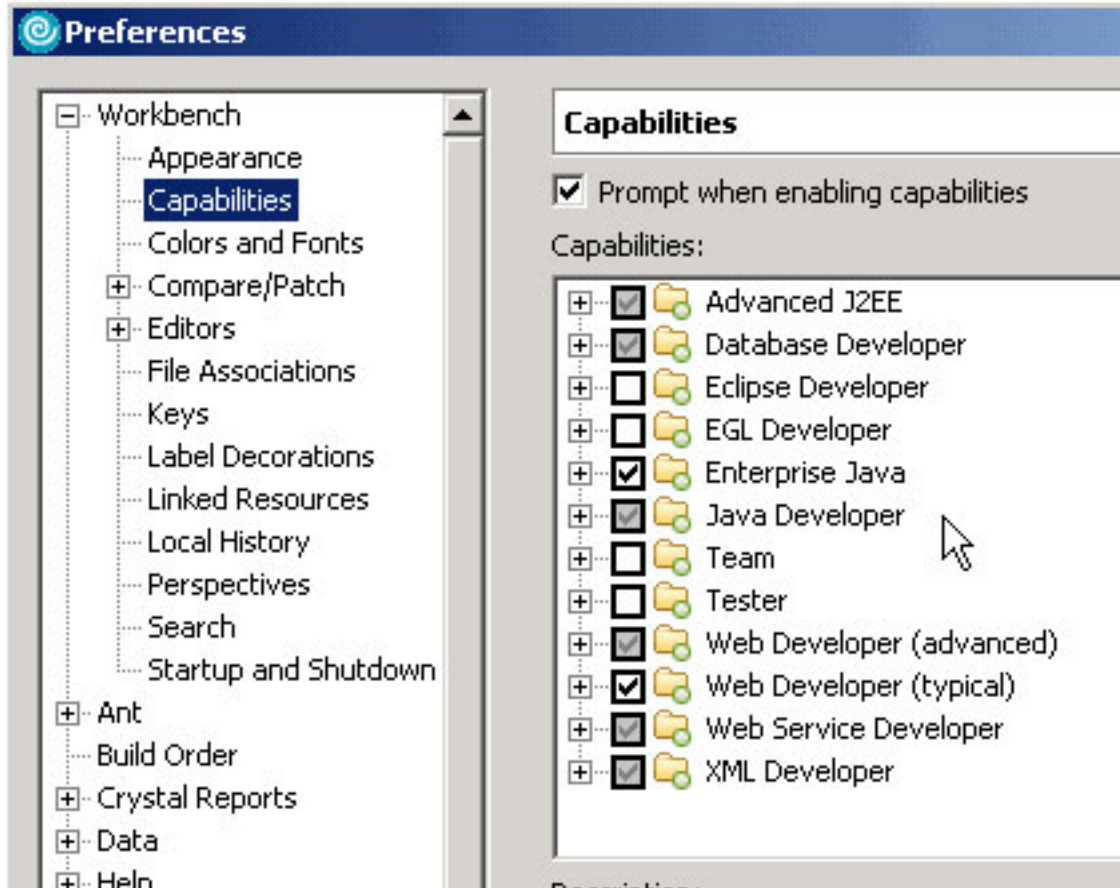
- Tester
- Web Developer (advanced)
- Web Developer (typical)
- Web Service developer
- XML Developer

Figure 6. Enable roles window



These capabilities can be enabled by role through the Welcome Page, and by role or individual capability through the Workbench Preferences window, which is discussed later. Figure 7 shows the Workbench Preferences method of enabling capabilities. Notice the same 12 roles, and that each role may be expanded to see the capabilities which are included under each role.

Figure 7. Enabling Capabilities



When you attempt to perform an action for a capability that has been disabled, or has yet to be enabled in the preferences page, a Confirm Enablement prompt might appear asking if you want to enable the required capability.

Section 3. Perspectives and views

Perspectives

Now that you have the workbench up and running, let's understand what it displays. Go ahead and close the Welcome page (click the **X** on the Welcome tab). The workbench window is organized into perspectives to assist in development tasks for different development roles on your team. Each perspective defines an initial layout and set of views.

The default perspective when you install the Application Developer product is the

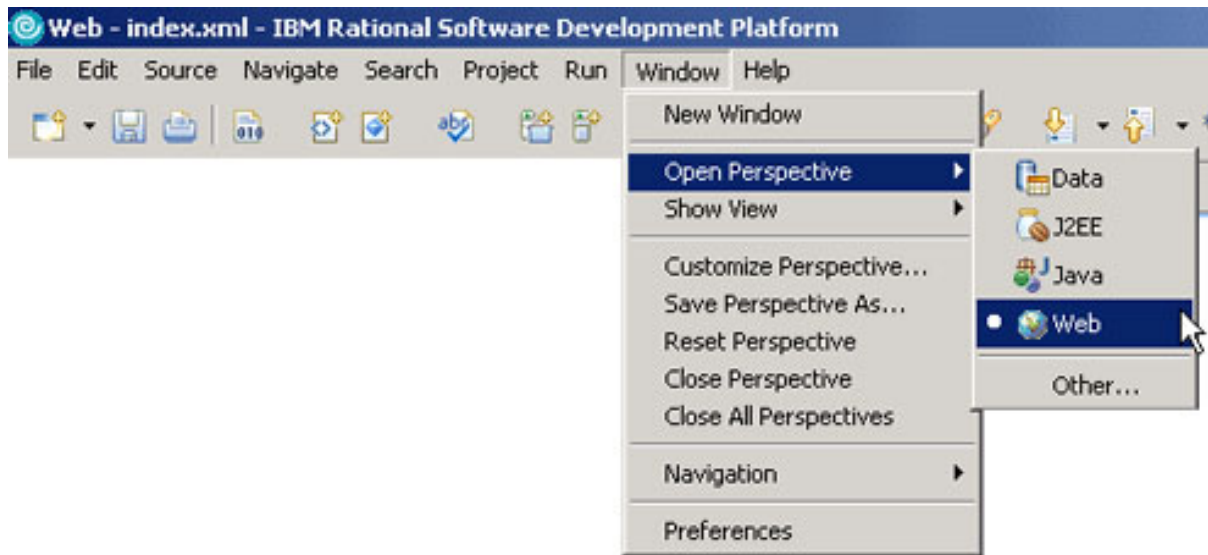
J2EE perspective. The default perspective always shows up on the short menu **Window > Open Perspective**. The default perspective is configurable through the workbench preferences.

The perspectives that ship with Application Developer are:

- CVS Repository Exploring
- Data
- Debug
- EGL
- Generic Log Adapter
- J2EE
- Java Browsing
- Java Type Hierarchy
- Plug-in Development
- Profiling and Logging
- Resource
- Team Synchronizing
- Test
- Web

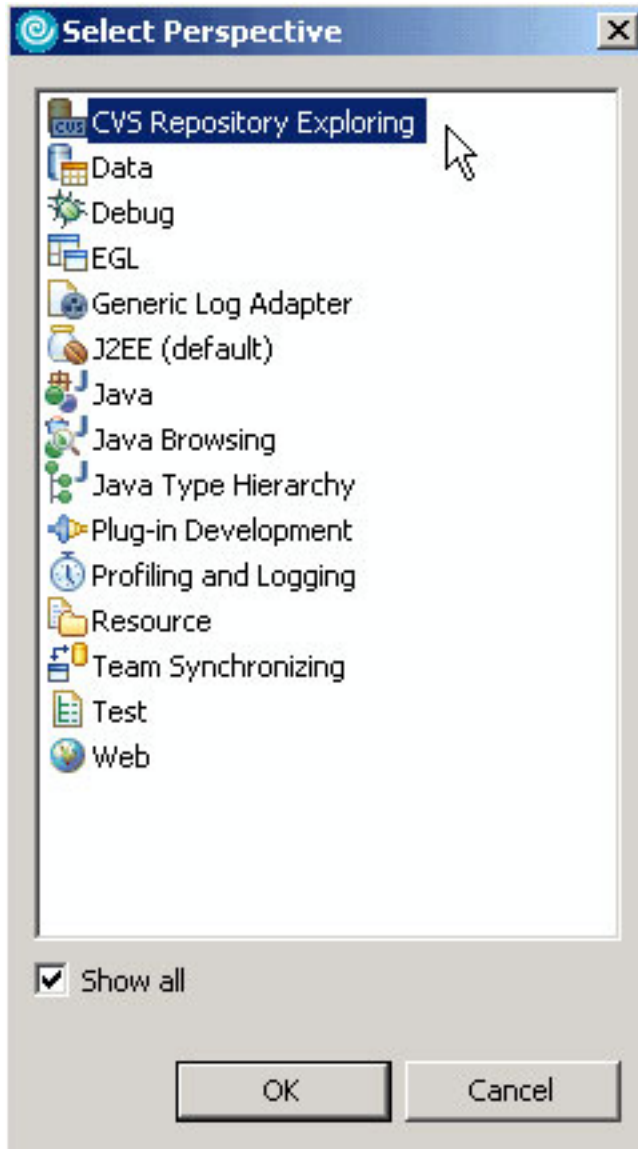
As you perform your job, you can open other perspectives to facilitate your tasks. One way to open another perspective is to select **Window > Open Perspective** from the main menu bar. This opens a short list from which you can select a perspective.

Figure 8. Open Perspective



If you select **Other...** and then check **Show all**, you get a complete list of the perspectives that ship with the product, and any you have created.

Figure 9. Perspectives

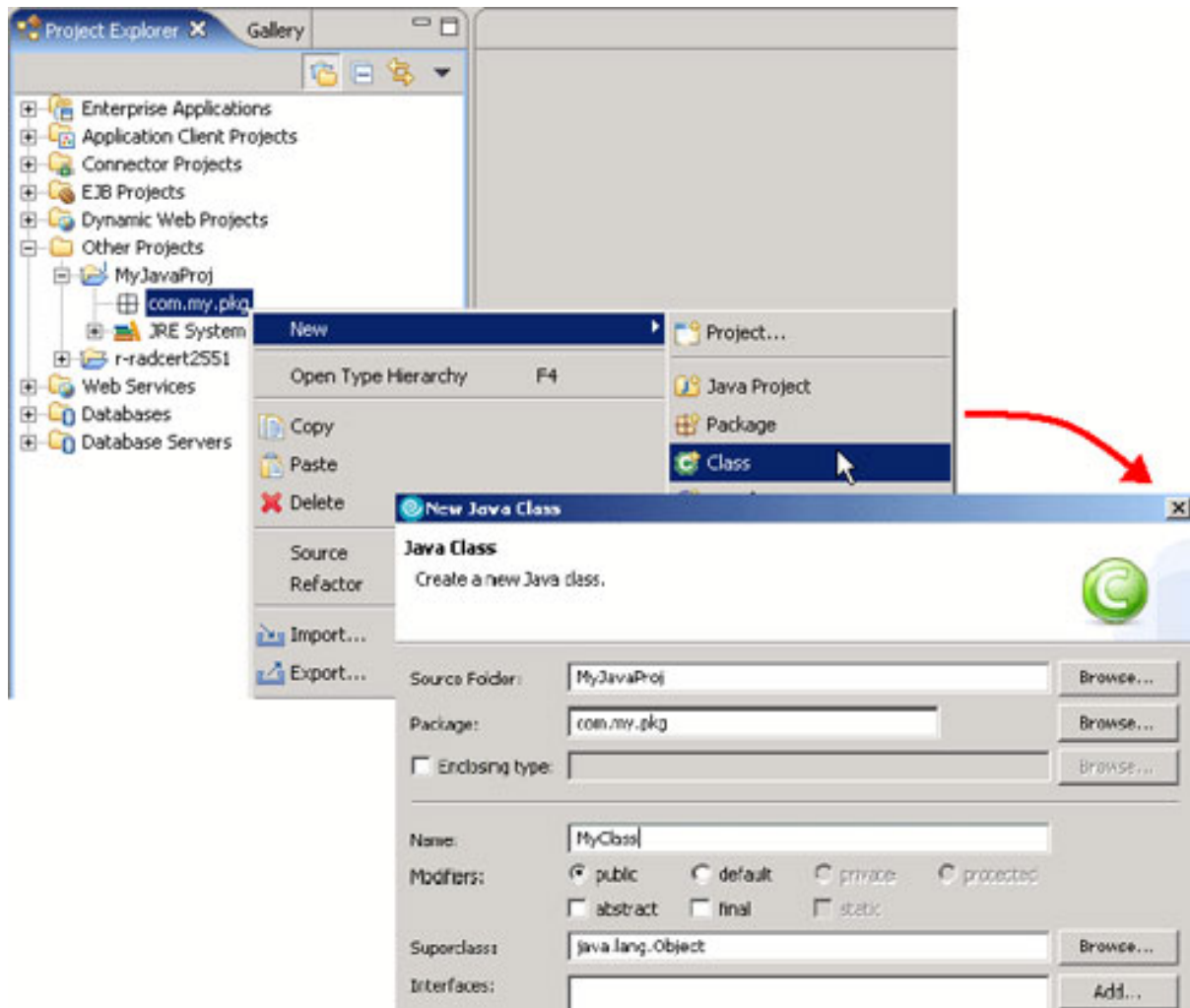


By default, when you open other perspectives they open in the same window as the workbench. If you prefer to open perspectives in a new window, change that setting in your workbench preferences. If you set the perspectives to open in the same window, the first word in the title bar displays the name of the current active perspective. You can switch between the open perspectives through the perspective switcher bar (highlighted in red below). Even the switcher bar position is configurable! Right-click the switcher bar to see your position options under the Dock On option. It is docked on the top right position below. Your other options are to dock it on the top left and left position. Notice that the perspectives icons are represented by both a descriptive icon and text. You can disable the Show Text option and remove the text that accompanies each perspective icon. The icon in the yellow box is the Open Perspective button and is a second way to open perspectives.

Figure 10. The Perspective Switcher Bar

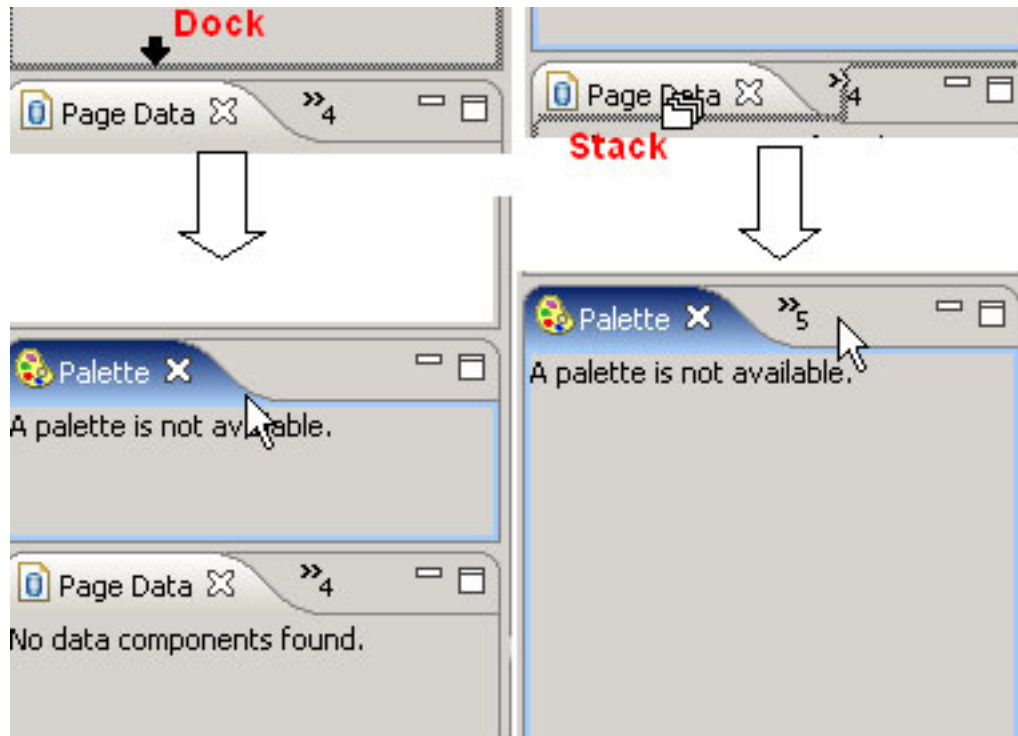
Views are designed to display, allow easy access to, and create different types of resources that developers work with as part of their job. Views let you create new resources during your development efforts. Selecting existing projects, packages, or folders prepopulates content when you invoke different wizards that guide you in creating or editing your resources. For instance, in the following figure a package is selected in an existing Java project and, from the package element, select the context menu (right-click) > **New** > **Class**. The resulting New Java Class wizard has the **Source Folder** and the **Package** entries filled in with our selected content, automatically.

Figure 11. Filling wizard entries through selection



Once you have selected a perspective to work in, customize it by dragging and dropping the views, by their title bars, to another position on the workbench. When you drag and drop a view, the cursor changes shape to indicate whether the view will overlay a current view or insert between two existing views. Two views that are overlaid is a *stack*. A view inserted between two existing views is a *dock*. When the cursor looks like a file folder icon, it stacks the views, and when it is a black arrow, it docks the views. Notice the number in the top right corner under the chevron >> symbol. That indicates the number of views that are stacked under that view. If you click on it, a list of the views stacked in that location appears. In the following figure, the Palette view is docked above the Page Data view on the left, and the Palette view is stacked on the Page Data view on the right.

Figure 12. View Actions



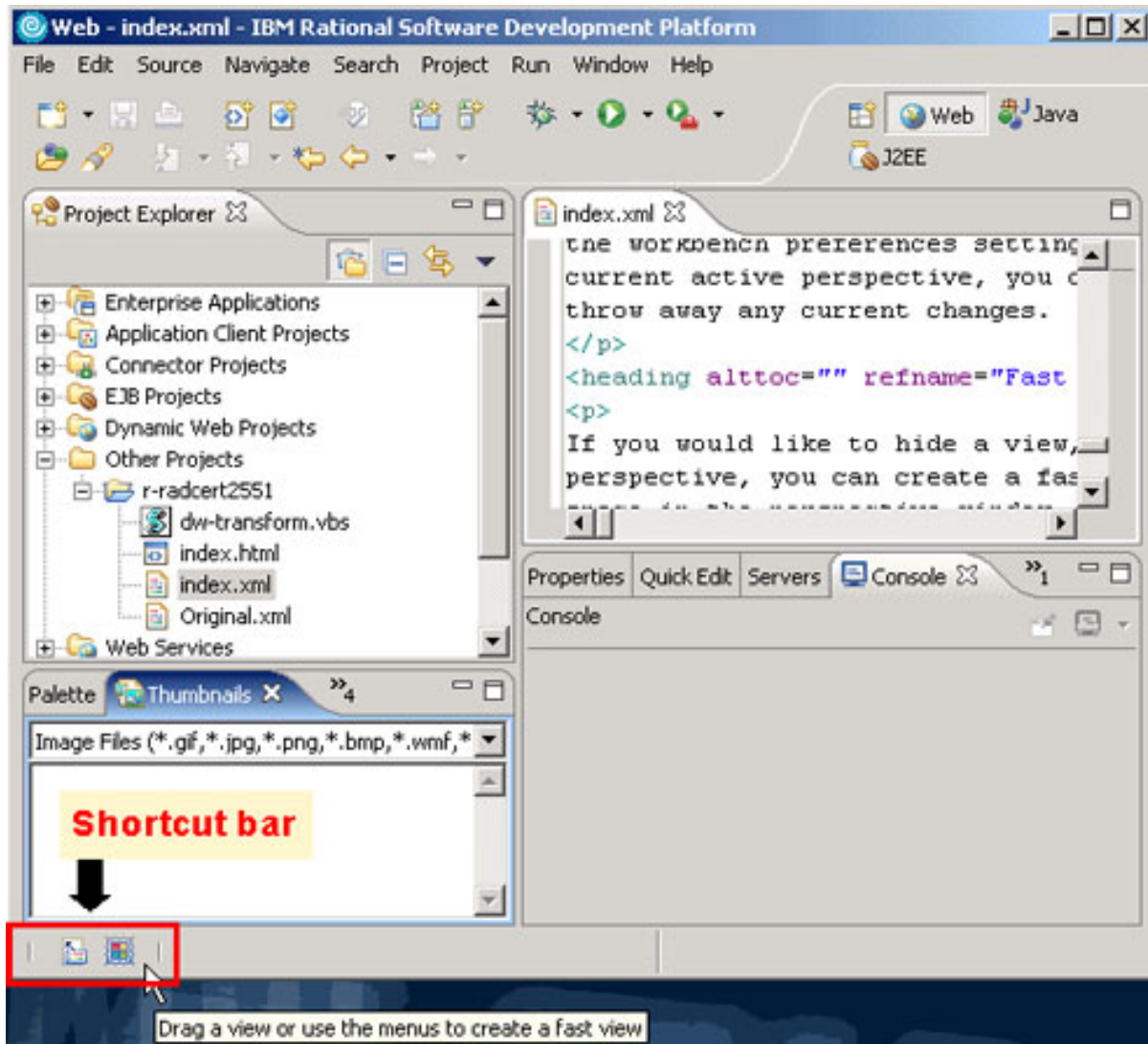
Aside from moving current views, to add a view to your active perspective by selecting **Window > Show View** from the main menu bar. After you have customized a perspective, save it as a new perspective or overwrite a Application Developer default perspective. So, yes, you can customize and overwrite the default perspectives that ship with Application Developer. You don't even have to worry about playing with the ability to customize the default perspectives because at any time you can get back to the default perspective's original settings shipped with the product through the workbench preferences settings. To discard any changes and modifications you made to the current active perspective, select **Window > Reset Perspective** to throw away the current changes.

Fast Views

To hide a view but do not close it or remove it from the perspective, create a fast view. *Fast views* are hidden views that do not take up space in the perspective window. The two ways to create fast views:

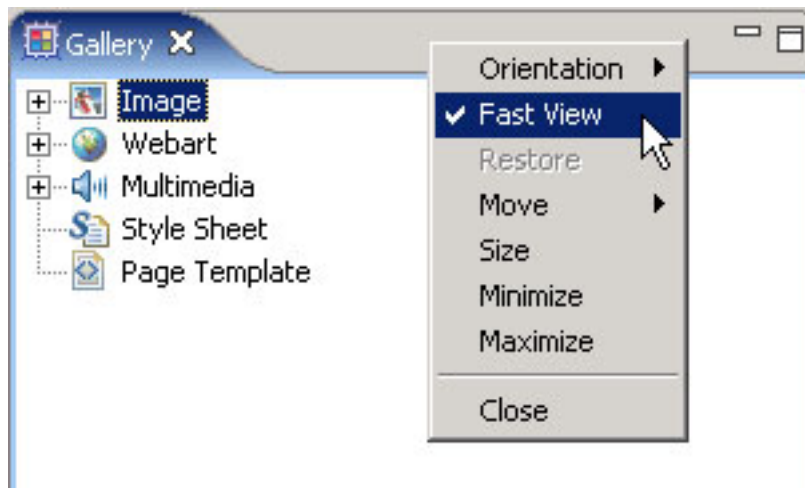
1. Left-click the title bar of the view that you want. Hold the mouse button down.
2. Drag the view to the shortcut bar and release the mouse button. By default the shortcut bar is located in the lower left corner of the workbench.

Figure 13. Shortcut bar



Right-click the view's title bar and select **Fast View** from the context menu.

Figure 14. Fast View menu



Once you have created a fast view from any of the workbench views, use it by clicking on the fast view icon. It acts as a toggle switch hiding and restoring the view each time you click on it.

Important Views

Previous users of WebSphere Studio Application Developer may remember using two views that required frequent switching between the two, depending on the task you wanted to perform: the J2EE Hierarchy view and the Project Navigator view. A major usability issue has been solved by combining these two views into the new Project Explorer view! The Project Explorer view provides an integrated view of all project resources.

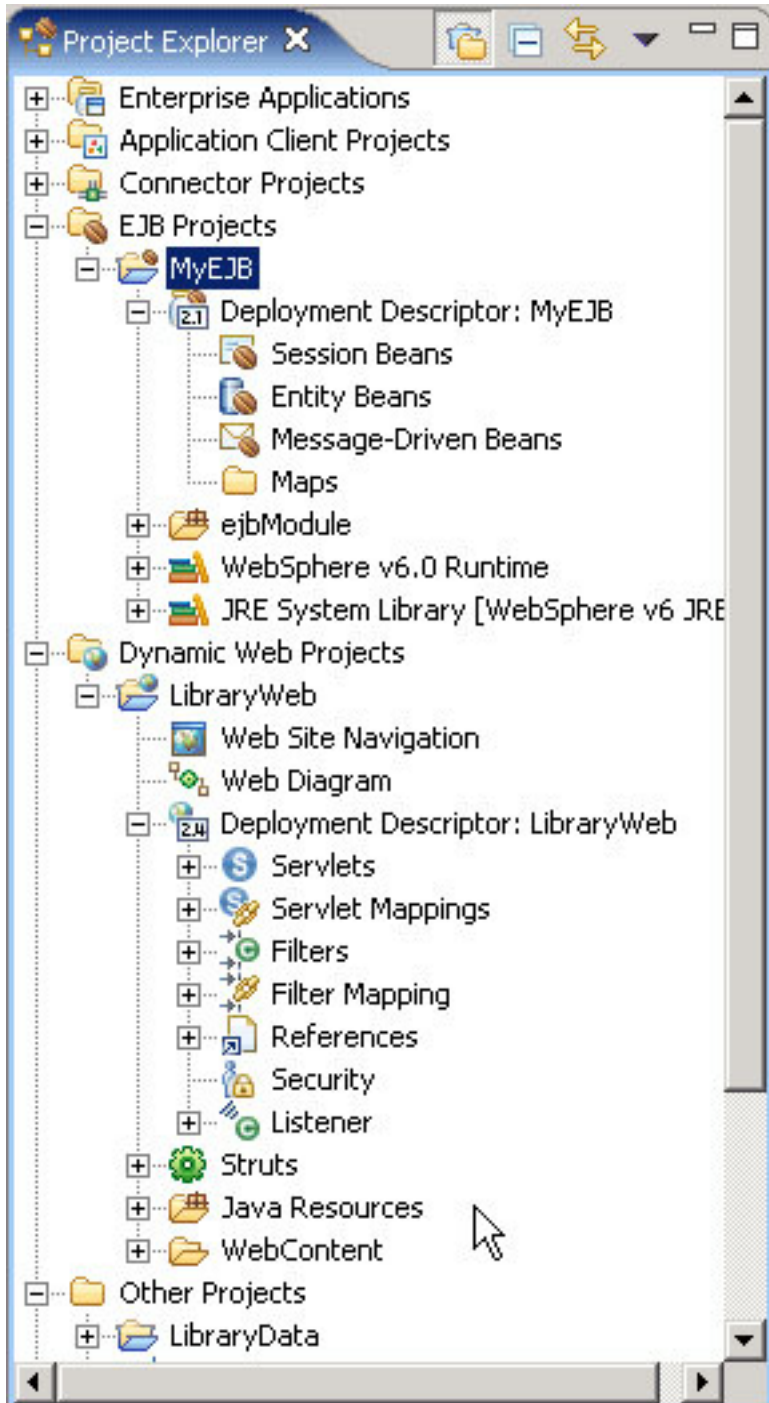
Notes on the Project Explorer view:

- Project folders are organized by J2EE type and the Java projects are stored under the Other Projects folder.
- Enterprise JavaBean (EJB) components in EJB Projects can be found by type by expanding the EJB deployment descriptor element.
- EJB source and generated deploy code is located under the ejbModule folder in the Project Explorer.
- J2EE Web resources can be found by expanding the Web deployment descriptor element.
- In Dynamic Web Projects, a developers' Java servlet, filter and utility source code can be found by expanding the Java Resources element.
- In Dynamic Web Projects, HTML files, JSPs and the Web deployment descriptor are created by developers in directories in the WebContent folder. Compiled code from the Java Resources element is automatically copied by the workbench into folders in the WebContent folder. Basically, the WebContent folder represents Web content

that needs to be published to the application server.

- There is a Struts element to view Struts resources.

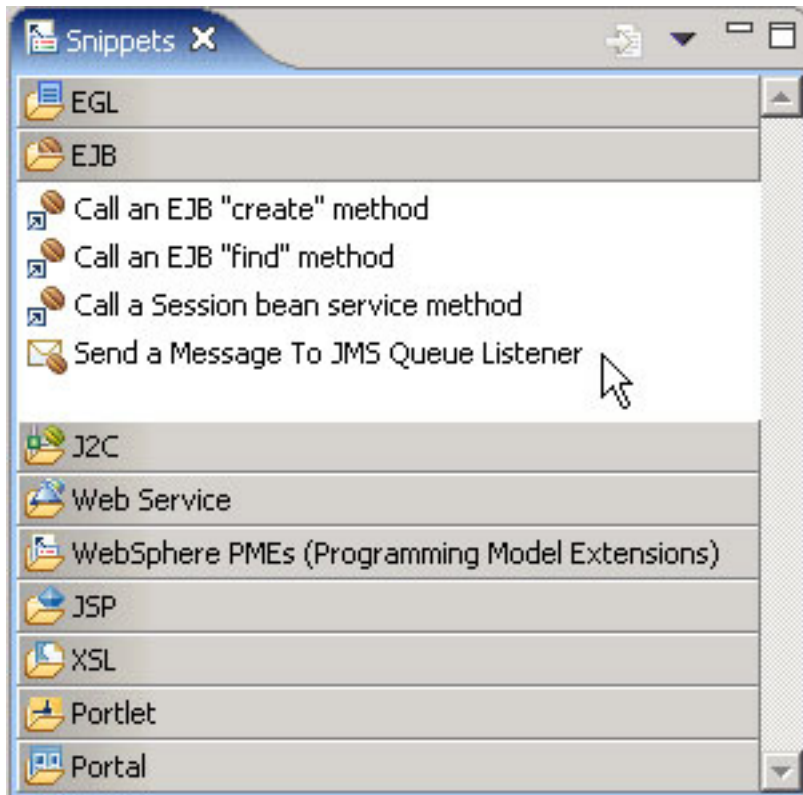
Figure 15. Project Explorer view



The Snippets view catalogs reusable programming objects, by type, into drawers,

that developers may use while developing Java code or Web resources. The Snippets view comes with some predefined drawers that include EGL, EJB components, J2C, Web Service, WebSphere PMEs, JSP, XSL, Portlet, and Portal. Not only do these drawers contain reusable objects, but some, such as the "Call an EJB create method", have been implemented to use well-known design patterns, such as the Service Locator pattern.

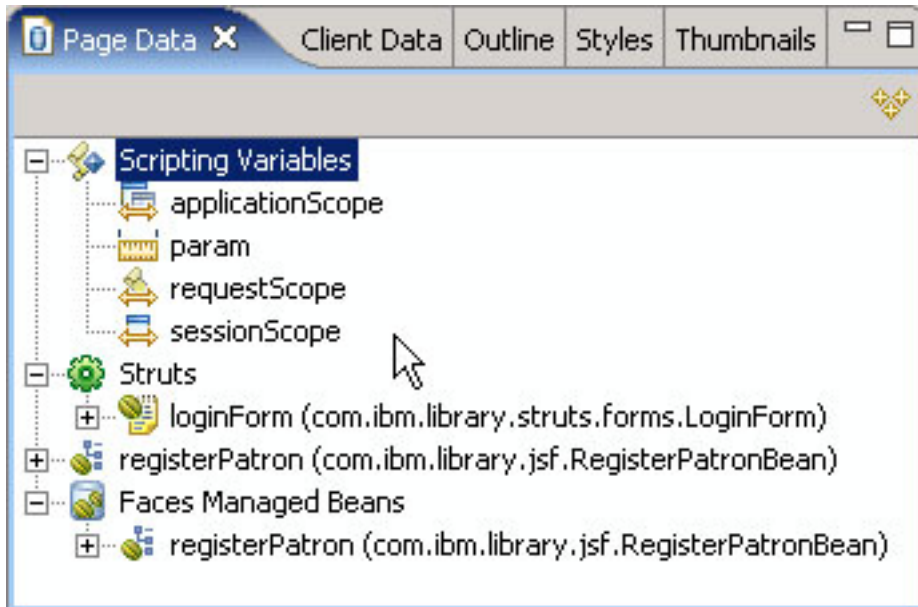
Figure 16. Snippets view



When developing Struts, Faces, or regular JSPs you will find that the Page Data view may come in handy. The Page Data view allows access to standard scripting variables that represent the application, session, request and page objects (applicationScope, sessionScope, requestScope, and param, respectively).

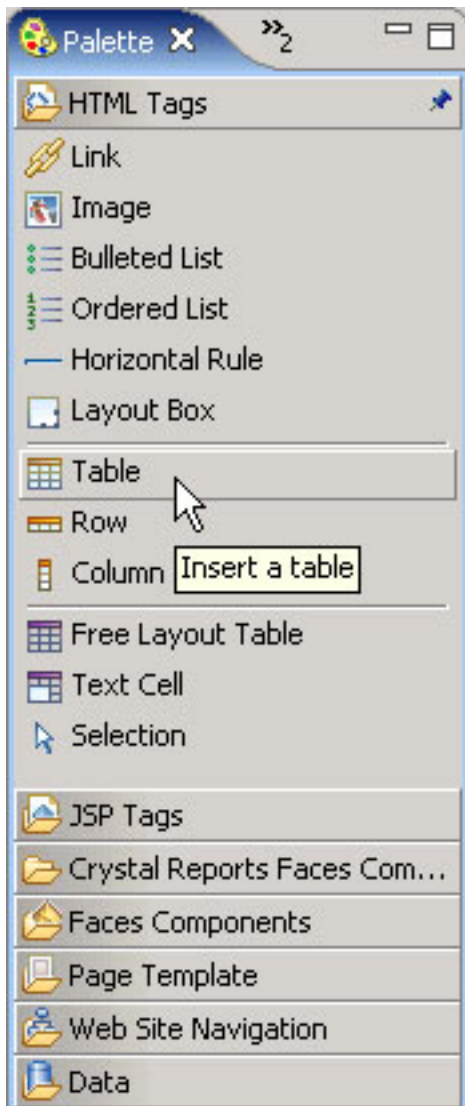
Other Page Data objects, depending on the type of JSP file:

- JavaBeans
- Session beans
- Web Services
- Relational Records and Relational Record Lists
- Portlet data objects
- Struts Form Beans
- EGL data items and records

Figure 17. Page Data view

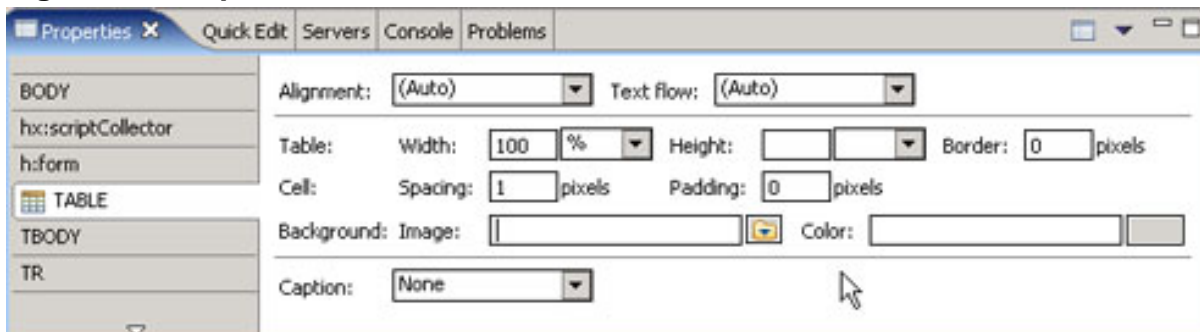
The Palette view assists in generating content for your JSPs and HTML files. When you expand the drawers you can find items to drag and drop into the active editor, like the Page Designer.

Figure 18. Palette view



With the Properties view you can set required and optional properties of selected elements in an editor, such as Page Designer.

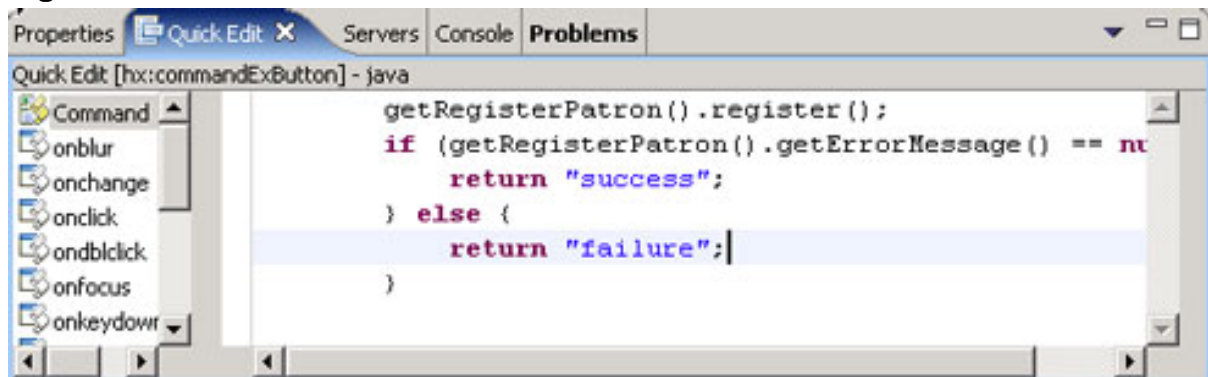
Figure 19. Properties view



The Quick Edit view is integrated with the Page Designer and lets you add short

scripts to your HTML and JSP files. If you select a button tag on a JSP page, you can then associate a click event with that button and add a script to it. It has also been integrated with the JavaServer Faces features in the workbench, allowing you to add code to the pagecode classes associated to your Faces Component JSPs.

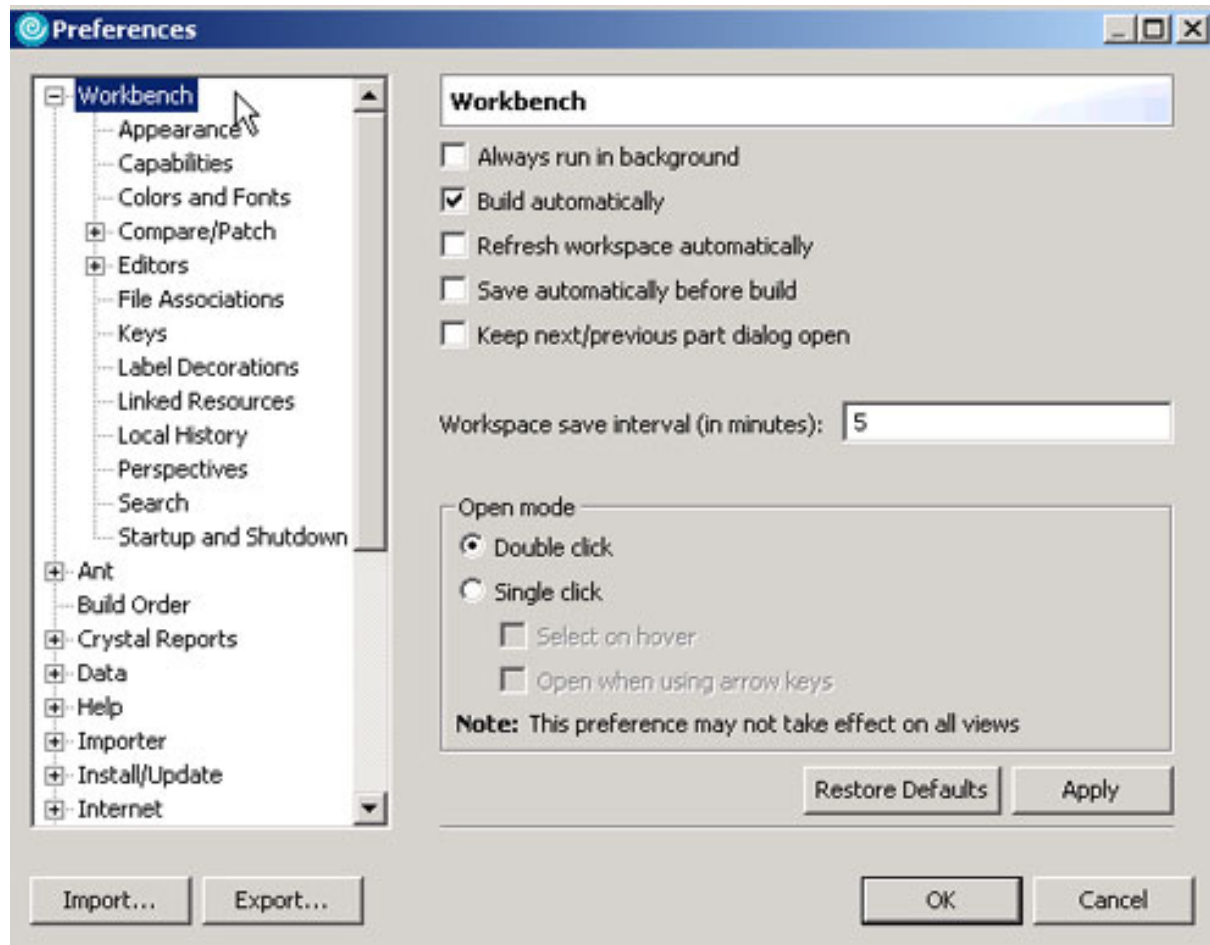
Figure 20. Quick Edit view



Section 4. Workbench preferences

Now that you have the workbench up and running, the tutorial discusses how to set the development environment to meet your needs. The workbench features and default behavior can be modified through the Workbench Preferences window. Access the workbench preferences dialog from the main menu bar by selecting **Window > Preferences**.

Figure 21. Workbench Preferences



Because this tutorial focuses on the Application Developer workbench basics, the Preferences window has the **Workbench** section expanded. Notice that the Build automatically option is enabled by default. Whenever a resource is modified, the Workbench automatically performs an incremental build. There is also a workspace save interval that indicates how often the state of the workspace is automatically saved to disk. If you are developing large projects, these options may take time and resources to complete if major revisions or changes are being made. If you select the Workbench preferences section and then click the F1 function key, the Help System's contents for the Workbench preferences section is displayed. This displays some details you might need when selecting your own preferences settings. The F1 function key can do this for many features in Application Developer.

Preferences

Appearance

Controls the appearance of the workbench. It organizes the position of the view tabs, editor tabs, and the location of the perspective switcher.

Capabilities

Allows you to enable and disable workbench features.

Colors and Fonts

Allow you to change the default colors and fonts for the workbench view and editor title text, editor and console text, console output text, and so forth.

Compare/Patch

Control options when you are comparing resources and applying patches (fixes) to code.

Editors

Decide default text file encoding, the color and location of annotations in the editor, QuickDiff settings, enable line numbers, and other text editor features.

File Associations

Add or remove file types recognized by the Workbench and associate editors with file types in the file types list.

Color and Fonts

Control fonts and colors and used by Eclipse components.

Keys

Customize how key strokes and key sequences are assigned to invoke particular commands.

Label Decorations

Show extra information about an item by modifying its label or icon.

Local History

Allows you to specify how many days to keep files, the number of entries per file, and the maximum file size that is maintained in history.

Perspectives

Control whether perspectives open in a new window, if new views open in a window or as a Fast View, and whether to prompt or switch to an appropriate perspective when creating new project.

Search

Allows the user to set preferences for searches.

Startup and Shutdown

Allows you to select whether to prompt for a workspace during startup and to select plug-ins to be automatically activated during workbench startup.

Section 5. Import and export wizards

Application Developer ships with import and export wizards that can assist you in various development tasks. Remember that using drag and drop or copy/paste to import files relies on operating system support that is not necessarily available on all platforms. If the platform you are using does not have this support, you can always use the import wizard. The tutorial highlights some of the more useful wizards. Access the wizards by selecting **File > Import** or **File > Export** from the main menu bar.

App Client JAR file

J2EE application client projects (that contain a deployment descriptor) are deployed as JAR files. The wizards can export a J2EE application client project JAR file or import a J2EE application client that had been deployed into a JAR file (see the Zip option below for working with basic JAR files).

EAR file

An existing J2EE enterprise application archive file may be exported or imported into a workspace with this option. After importing an EAR file you may then navigate the contained J2EE project types that were archived, using the Project Explorer view.

EJB JAR file

EJB projects are deployed as JAR files. These wizards can import an enterprise bean project that has been deployed into a JAR file or export an EJB project you have developed as an EJB JAR file.

WAR file

Web archive file projects are deployed as WAR files. The import wizard can import a dynamic Web project that has been deployed as a WAR file into an enterprise application project. The import wizard prompts you for a context root. You can also export a dynamic Web project as a WAR file.

RAR file

Java 2 Connectors are deployed as resource archive files. These wizards allow you to export a RAR file or import a connector project that has been deployed into a RAR file.

Existing Project into Workspace

This wizard can be used to import projects created with WebSphere Application Developer version 5.1.X. It can also be used to add projects that have been previously deleted from a workspace (you can delete a project from a workspace, but not from the file system, so that it does not consume

workbench resources). If at any time you suspect that the workbench metadata directory has been corrupted, you can recover by closing the workbench, deleting the workspace's metadata directory, then opening the workspace (it appears empty) and using the Import > Existing Project into Workspace wizard for each project in the workspace to recreate the workspace metadata.

File system

This is the option to select if you want to copy JAR files into your lib directory without extracting them from the JAR file. Copy files from or to the local file system and your workspace.

FTP

Use the file transfer protocol to transfer complete Web sites into Web projects. It provides options to limit the scope of the import. You can also export a Web project to a URL.

HTTP

Use the HTTP protocol to transfer complete Web sites into Web projects. It provides options to limit the scope of the import.

Project Interchange

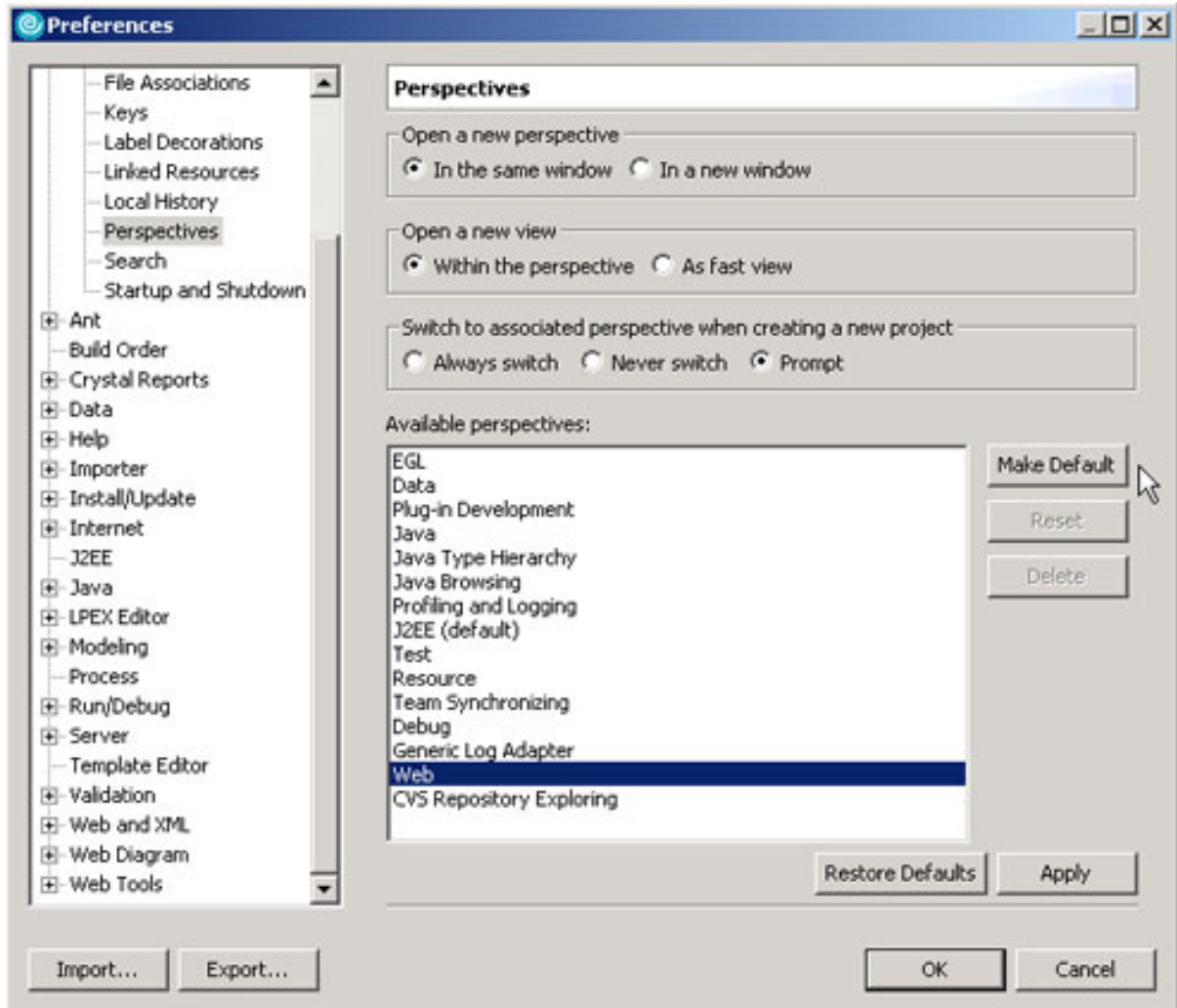
A great alternative to sharing data with other developers, where they get the content from your workspace, plus any metadata associated to it.

Zip file

The import Zip file option imports and extracts files from an archive file. It works on both ZIP and JAR files. The export wizard has an export ZIP file option and a separate export JAR file option.

Let's take a closer look at a one of the workbench preferences. The following figure shows the Perspectives preferences selected. Notice that the Web perspective has been selected. If you click on **Make Default**, the Web perspective becomes the default perspective. Notice also that if you select any of the perspectives, and then select the Reset button, that perspective loses any customization that you applied. This is just like selecting **Window > Reset Perspective** from the main menu bar of a perspective. The Restore Defaults button resets *ALL* perspectives to their Application Developer product defaults. This option is only applicable to built-in perspectives that have been overwritten using the **Window > Save Perspective As...** option. Last but not least, the Import and Export buttons in the lower left corner allow you to share your customized Application Developer preferences with other developers.

Figure 22. The Perspective Preferences



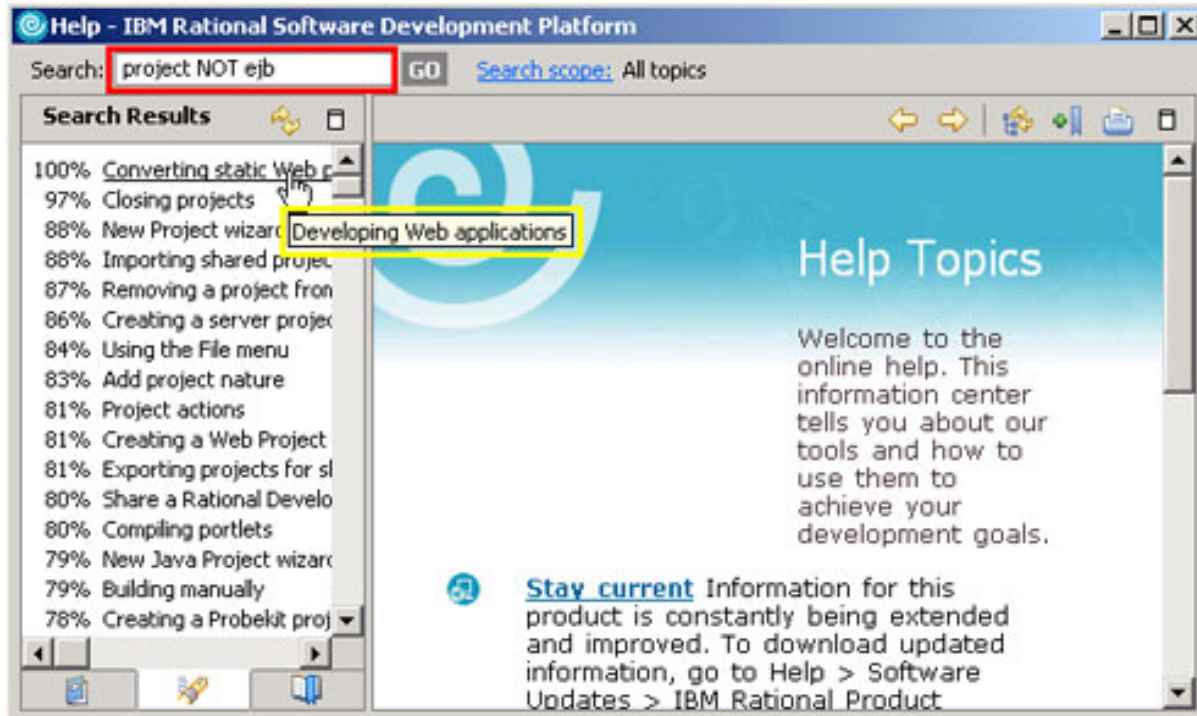
Section 6. Help System

The help system included with Application Developer lets you search product documentation, browse through the search results, and print documentation. Select **Help > Help Contents** from the main menu bar to invoke the help browser. The help system is constantly being updated and improved, so to stay current, the help system is updated when you update the Application Developer product software.

Learning to use the help system appropriately saves you the time of having to sort through search results that are not relevant to your help needs. For instance, when you type multiple terms into the Search field at the top of the Search browser, there is an implied AND between the terms. If you want to exclude certain results, you can

type the keyword **NOT** before a term to have those results filtered.

Figure 23. Search Keywords



Notice in the search browser above (in the red box) that the target of the search is the term *project*, but requests search results that do not include the term *ejb*. Because the search is not case sensitive, this also excludes the term *EJB*. Other keywords that you might use in your searches are **AND**, **OR**, **?** for a single-character wildcard, and ***** for a multi-character wildcard. Use quotation marks around terms that you want used as a phrase, encountered together, as typed.

Also, if you look at the hover help that appears when you place the cursor over a specific search result (notice the yellow box above), it displays the location of that search result in the help system topics. If you knew which topics you wanted to search, or, at least, which topics you did not want to search, the results could be filtered using **Search scope** next to the Search text entry field. In the following figure, we have clicked **Search scope**, then the Search only the following topics radio button and then **New**. Once you create topic scopes, you can use them to limit searches.

Figure 24. Search scope

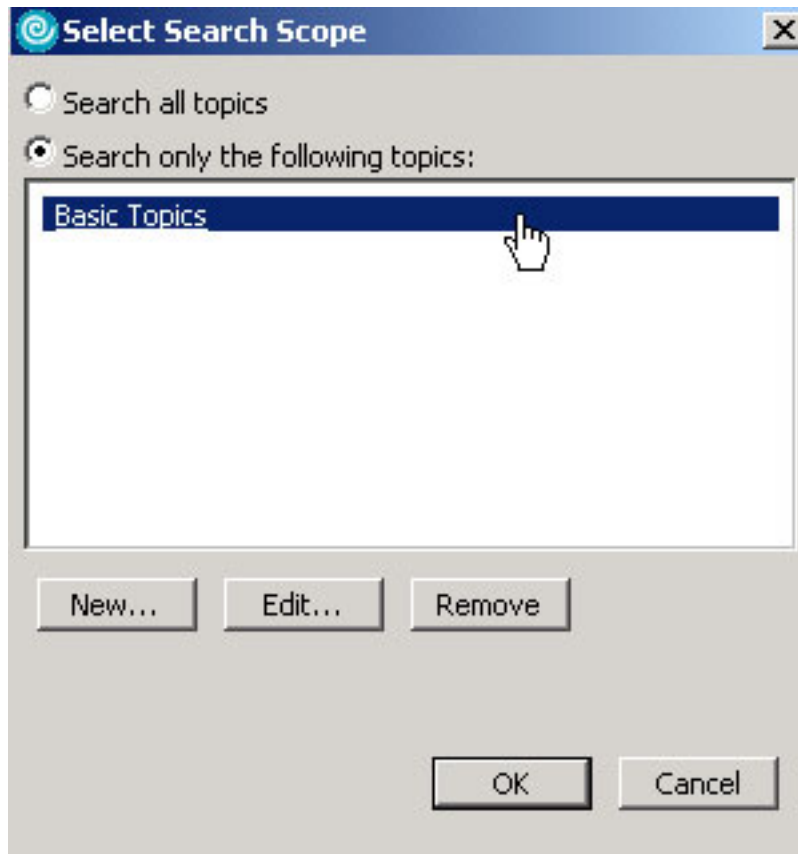
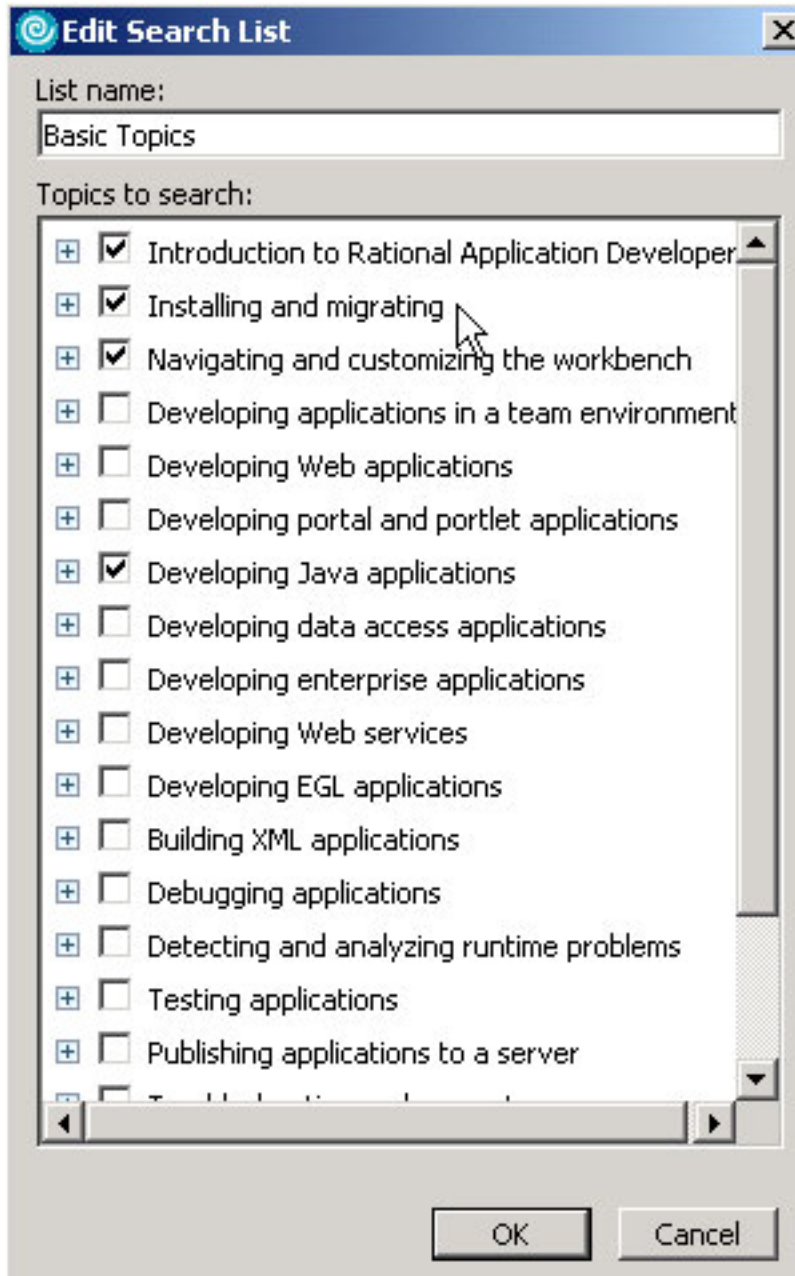


Figure 25. Edit search list

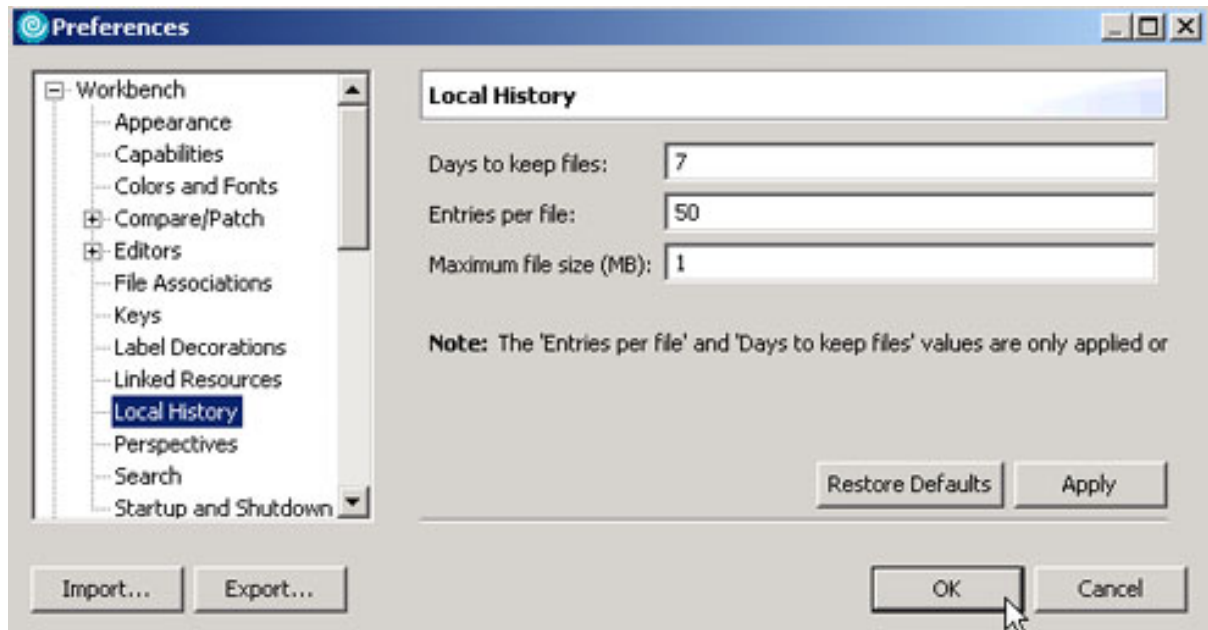


Section 7. Local History

The Application Developer workbench maintains copies of your locally developed resources each time you save a resource. This feature comes in very handy when you wish to revert back to a previous copy of a resource, or compare a current

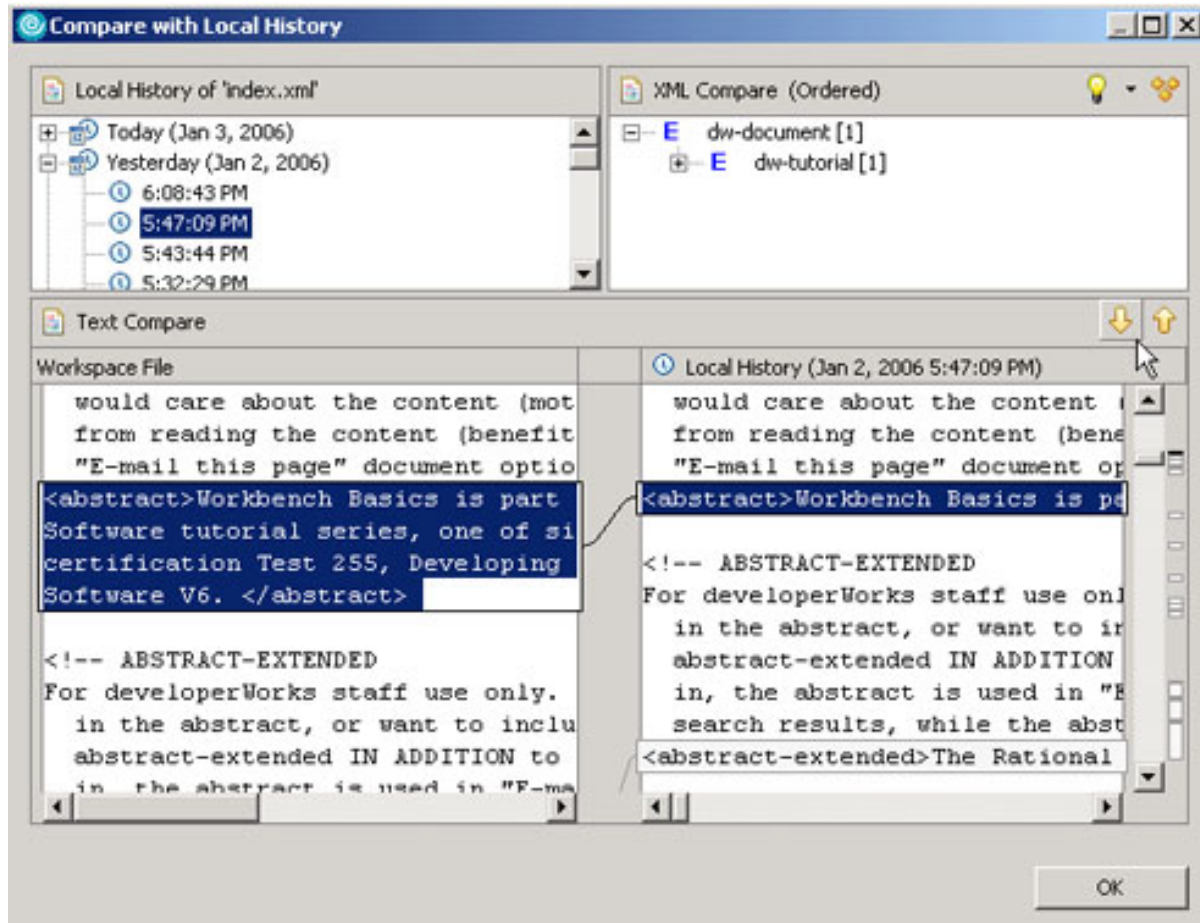
resource to a copy that existed last night when you went home. The feature is enabled through the workbench preferences, under **Workbench > Local History**. The settings include **Days to keep files**, **Entries per file**, and **Maximum file size (MB)**. If a file is over the maximum file size indicated, a history file is not stored for it.

Figure 26. History Preferences



In some instances you have access to your version control system server and this ability seems redundant, but what if you have been making changes and wish to compare or revert to a local copy that was never checked in to your version control system? Just select the file, and from its context menu, select **Compare with > Local History** or **Replace With > Local History**. This brings up a window that shows all the file saves you have performed, up to the limit you specified in the Local History preferences.

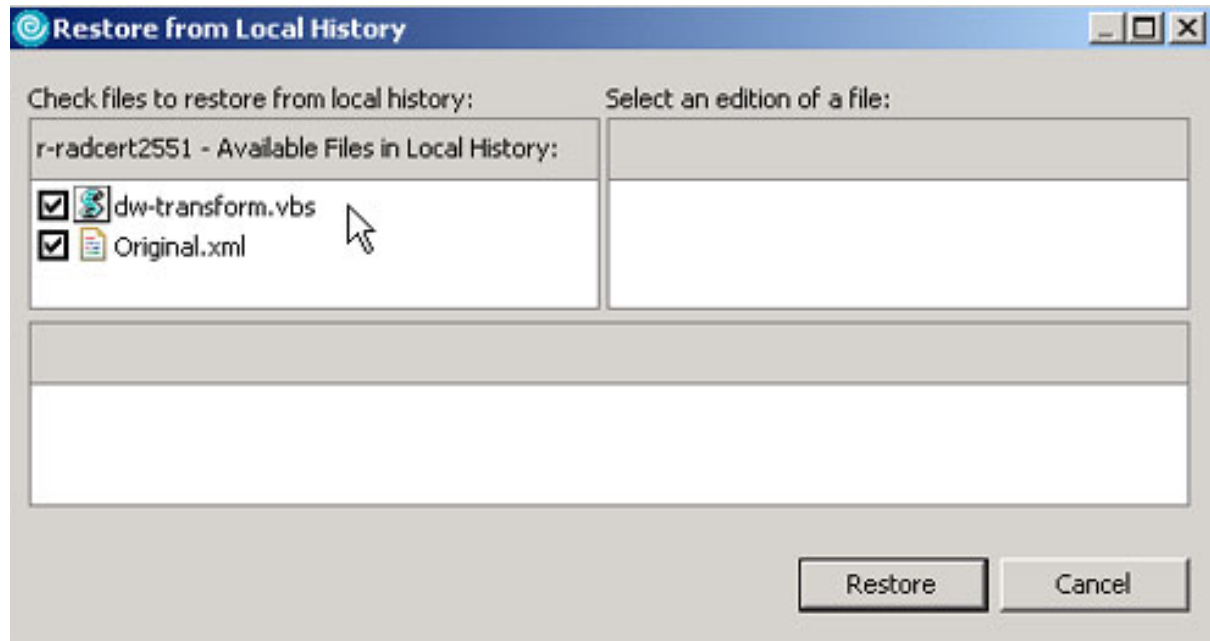
Figure 27. Local History



Notice that there are versions of your resources stored on the local file system, identified by their time stamps, from today, yesterday, and up to the number of days you specify in the workbench preferences. The yellow up and down arrows allow you to navigate back and forth from the next or previous difference between the file and its local history copy. If you select two files from the Project Explorer, you can also choose **Compare With > Each Other**. This is only useful on similar files.

The local history is also useful when you accidentally delete a resource that you did not save to your version control system. Just select the folder or project that contains it, and from its context menu select **Restore from Local History....** If you deleted multiple files, you can choose which files to restore.

Figure 28. Restore from Local History



Section 8. Conclusion

This tutorial covered the basics, such as what are views and perspectives, how to customize your perspectives, a very small subset of the Application Developer preference settings, the import and export wizards, the local history and the search features. Completing the seven tutorials in this series can help you gain the knowledge you need to prepare for Test 255: Developing with IBM Rational Application Developer for WebSphere Software V6, but nothing can replace the experience and knowledge that is obtained from using the product.

I sincerely hope you have found this tutorial helpful, and wish you luck as you prepare for your certification test.

Resources

Learn

- Get certified as an ["IBM Certified Associate Developer"](#). Check out the [objectives](#), [sample assessment tests](#), and [training resources](#) for test 255, "Developing with IBM Rational Application Developer for WebSphere Software V6".
- Stay current with [developerWorks technical events and webcasts](#).

Get products and technologies

- Download a free trial version of [IBM Rational Application Developer](#).
- Build your next development project with [IBM trial software](#), available for download directly from developerWorks.

Discuss

- [Participate in the discussion forum for this content](#).
- Participate in [developerWorks blogs](#) and get involved in the developerWorks community.

About the author

Mr. Bobby McChesney



Bobby McChesney was an Education Specialist with IBM's Application Development team. He specialized in WebSphere Application Server Version 6 and Rational Application Developer for WebSphere® Software courses. Bobby also taught the Java Servlet, JavaServer Pages, Struts, JavaServer Faces, and Enterprise JavaBeans programming courses. He is currently a developer on the WebSphere Administrative console development team.