Task-Oriented Solutions to Over 175 Common Problems

Covers Eclipse 3.0

Eclipse Cookbook™

O'REILLY®

Steve Holzner

Table of Contents

CHAPTER 2

Using Eclipse

2.0 Introduction

This chapter is about mastering Eclipse in everyday use. Chapter 1 gave you the basics; this chapter is designed to give you a working knowledge.

As with any complex tool, you might want to change things about Eclipse as you work with it more and more. So, in addition to teaching you how to work with Eclipse, this chapter also teaches you the many ways in which you can customize Eclipse, from moving views to creating your own perspectives.

When it comes to customizing Eclipse, one dialog stands out over the rest: the Preferences dialog, which you open by selecting Window → Preferences. This dialog is shown in Figure 2-1.

I encourage you to become familiar with this dialog. The Preferences dialog is the home of Eclipse customization, especially workbench customization. Want to automatically close all editors when you exit Eclipse (which enables Eclipse to start more quickly next time)? Select Window → Preferences → Workbench → Editors and then check the "Close all editors on exit" checkbox. Want to make editor tabs appear on the bottom of editor windows? Select Window → Preferences → Workbench → Appearance, and click Bottom in the Editor tab position box. Want to specify what editor or program Eclipse should use to open files with a certain file extension? Select Window → Preferences → Workbench → File Associations, choose a file extension, and click the Add button. Problem solved.

But as powerful as the Preferences dialog is, it's just the beginning of the customization story. Eclipse can be customized in thousands of ways. Read on to get a grip on tailoring Eclipse to match your requirements.

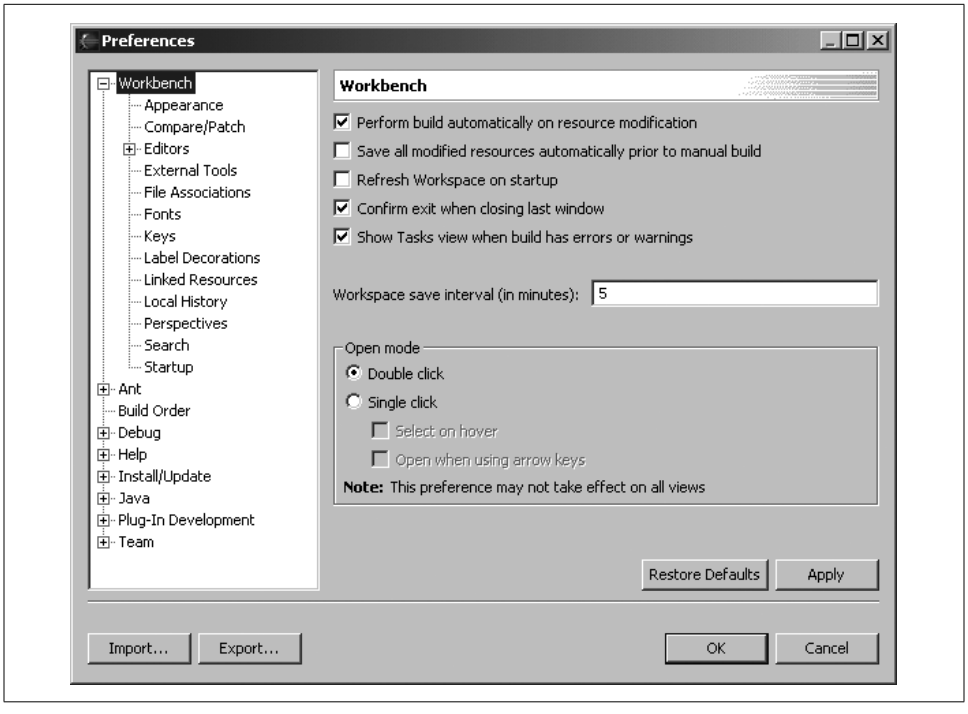

Figure 2-1. The Preferences dialog

2.1 Showing/Hiding Views

Problem

Where did the Console view go? It was here a minute ago.

Solution

To show a view, select Window → Show View, and choose the view you want to show. If the view you want isn't visible, select Window → Show View → Other, and choose a view from the dialog box that Eclipse presents of all views it knows about. To close a view, click the X button in its tab.

Discussion

It's easy to close a view accidentally or to work in a perspective that doesn't display a favorite view, such as the Console view that displays text sent to the output console. Just open the view by selecting Window → Show View.

For example, to add the Navigator view to the Debug perspective, open the Debug view (covered in Chapter 5) by selecting Window → Open Perspective → Debug.

Then add the Navigator view by selecting Window → Show View → Other → Basic → Navigator.

If you want to save the newly configured perspective, select Window → Save Perspective As. For more information, see the Recipe 2.21 later in this chapter.

Some perspectives try to stack too many views on top of each other, which can make scrolling to the right tab annoying. Now that you know you can open views again as needed, you should have no qualms about closing extra views to remove clutter.

See Also

Recipe 5.3 on starting a debugging session; Chapter 1 of *Eclipse* (O'Reilly).

2.2 Moving a View or Toolbar

Problem

The Package Explorer should be on the right, shouldn't it?

Solution

You can drag toolbars and views in Eclipse, and they'll dock on the various edges of the Eclipse window. Figure 2-2 shows the Package Explorer being dragged to a new location.

The Package Explorer now appears on the right, as shown in Figure 2-3.

Besides being dragged, toolbars also can be broken into segments. Each toolbar features a graspable handle (the upright 3D bar at the left edge of the toolbar). By dragging this handle, you can resize the segments in the toolbar, as well as show or hide additional controls.

You might not like working in an environment in which you can accidentally move toolbars, so select Window → Lock the Toolbars to hold things in place. If a perspective becomes scrambled as a result of accidental mouse movements, select Window → Reset Perspective to reset the perspective.

Discussion

You also can drag editors, but you can't mix items in the editor and view areas. The editor area is the center of the Eclipse window, and Eclipse won't allow you to drop any views there, or drop an editor on top of a view.

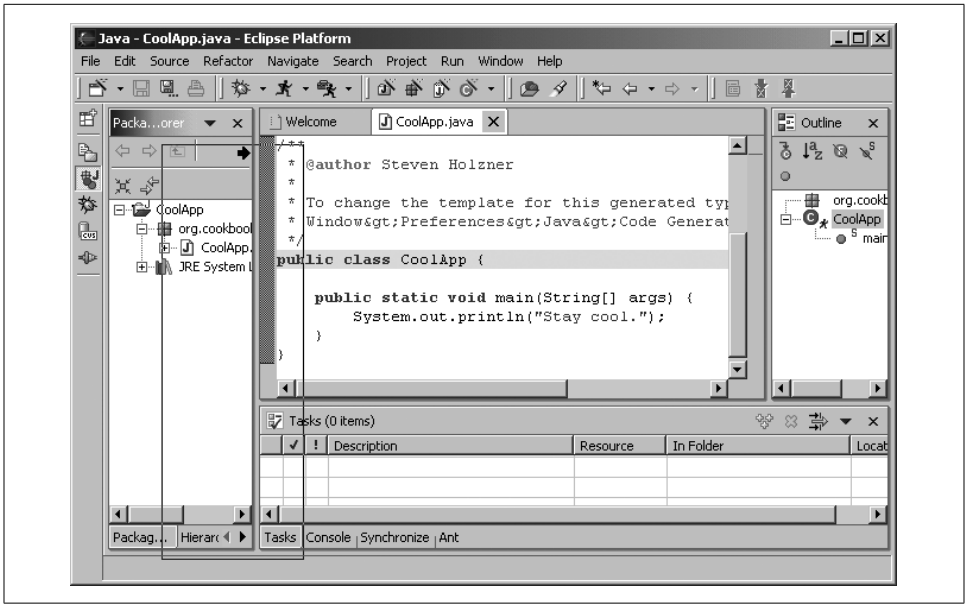

Figure 2-2. Dragging a view

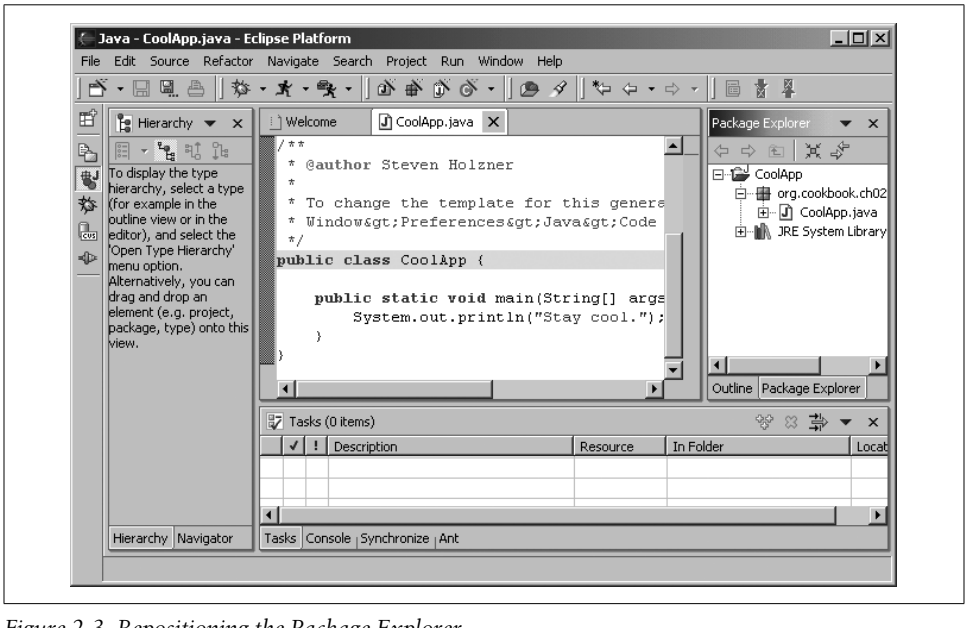

Figure 2-3. Repositioning the Package Explorer

See Also

Chapter 1 of *Eclipse* (O'Reilly).

2.3 Accessing Any Project File

Problem

Many perspectives will hide files; you want to get access to a specific file (or files) in your project.

Solution

The Navigator view in the Resource perspective gives you access to all the files in a project, without exception.

Discussion

Some perspectives hide files. For example, Eclipse stores project information in an XML file named *.project*, but many views, such as the Java perspective's Package Explorer, will hide that file. The *.project* file for a sample project open in the Resource perspective is shown in Figure 2-4.

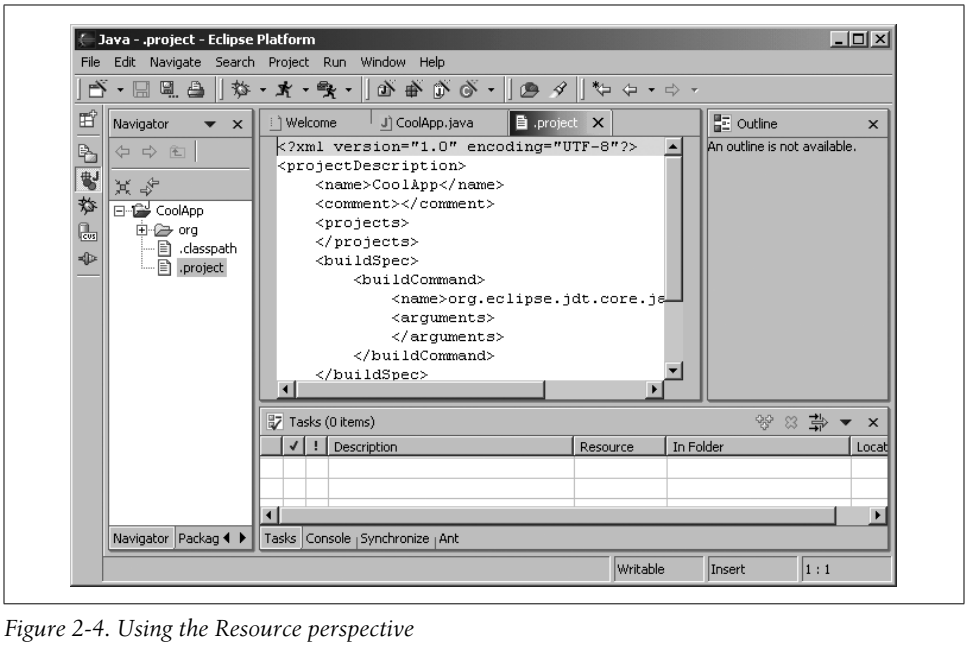

Figure 2-4. Using the Resource perspective

Some Java-oriented perspectives also support the Navigator view. Many Java programmers ignore the Resource perspective completely, but some Java-oriented views will hide various files from you. To get access to them all, don't forget about the Navigator view; it's always available in the Resource perspective as well as in some of the Java perspectives.

2.4 Tiling Editors

Problem

Although you can switch editors by clicking their tabs in the editor area, sometimes it's more desirable to have two editors open at once, e.g., when you're comparing two files visually.

Solution

You can drag editors and can tile the editor area with them.

Discussion

When you drag an editor and position it near an edge of the editor area, you'll see an arrow, as shown in Figure 2-5, which indicates that dropping the editor will dock it on that edge.

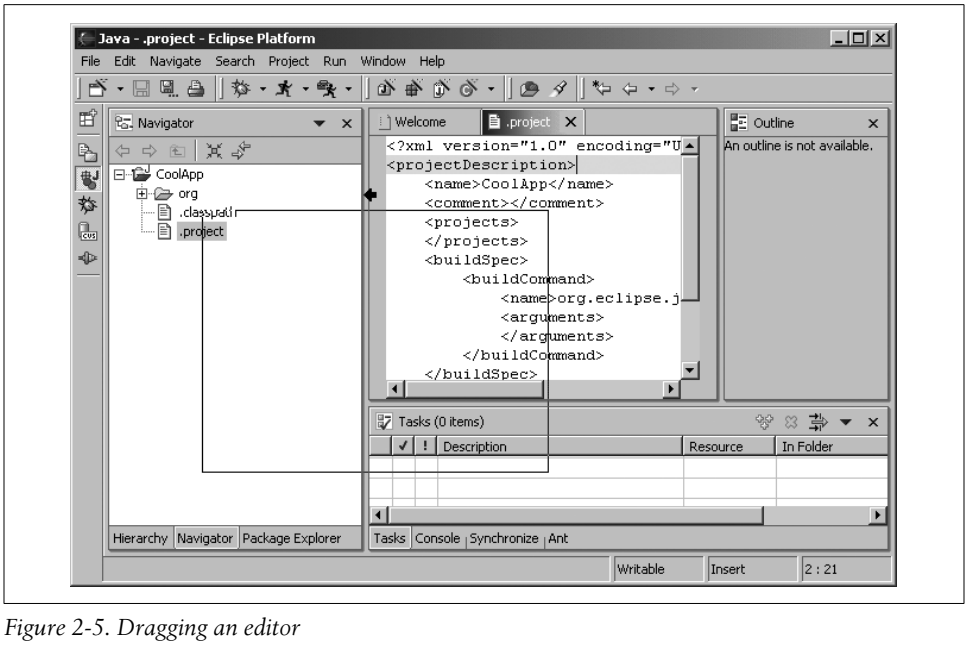

Figure 2-5. Dragging an editor

The new, tiled editor presentation is shown in Figure 2-6.

To restore an editor to its original position, just drag it back to where it was.

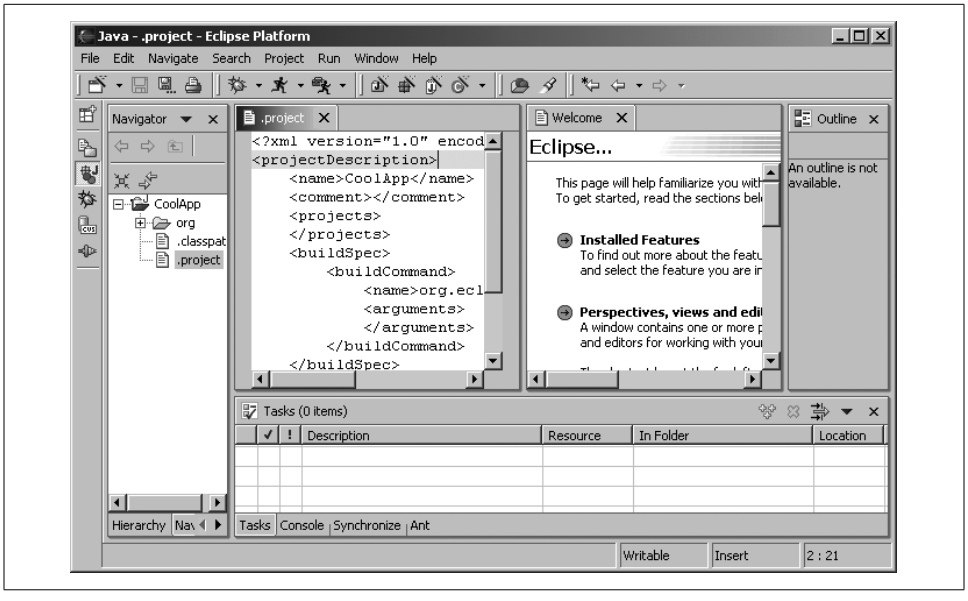

Figure 2-6. Tiling editors

You can open an item in an editor by dragging the item from a view such as the Package Explorer and dropping it on the editor area.

2.5 Maximizing Views and Editors

Problem

The Eclipse window is crammed with menu bars, toolbars, views, and editors. Sometimes, editing your code seems too cramped an experience.

Solution

You can maximize views or editors simply by double-clicking the view's titlebar or the editor's tab, something few people know.

Discussion

Figure 2-7 shows a maximized view of the JDT editor.

Double-click the view's titlebar or the editor's tab to restore it to its original size.

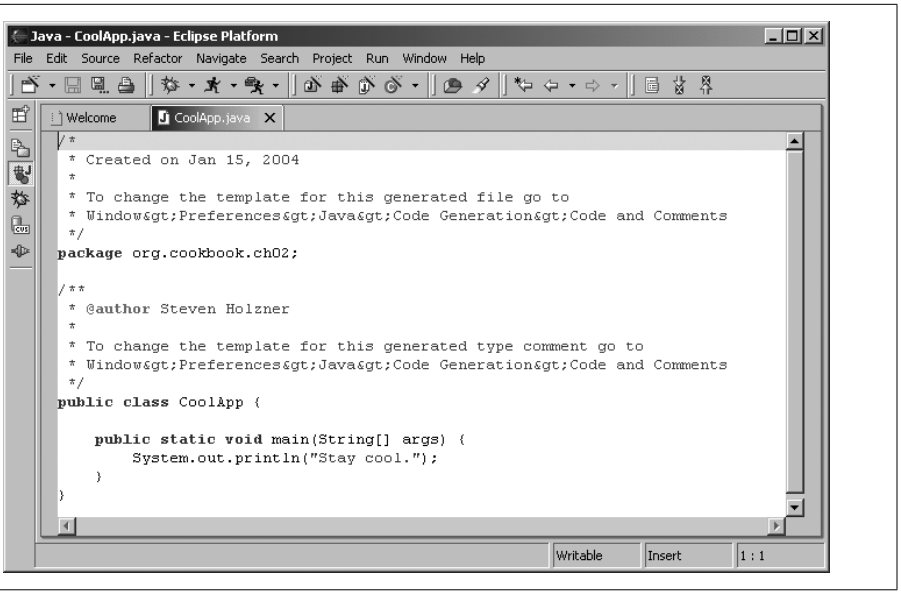

Figure 2-7. Maximizing an editor

As with many IDEs, Eclipse can feel crammed at times, especially when you're working on large documents. You might end up feeling like you're peering through a nest of distractions at the code you need to edit. To solve this problem, just double-click the editor's tab to maximize the editor.

2.6 Going Back to the Previous Editor

Problem

You've got 20 editors open, and scrolling through all those editor tabs to navigate between them is taking a long time.

Solution

Use the workbench editor's navigation history, which works much like a web browser's history. Just click the back arrow in the workbench toolbar as you would in a web browser to go back to the previous editor. (Or use Navigate menu items such as Navigate → Backward, or keyboard shortcuts such as Alt-Left Arrow.)

Discussion

When you work with Eclipse on extended projects, you'll find the editor area filling up with editors. And as the editors get more cramped, Eclipse makes their tabs smaller and the text in them more abbreviated. The more editors you have open, the

more difficult it becomes to move between them. One way to ameliorate this problem is to bear in mind that you can use navigation controls to move between recently used editors.

2.7 Going Back to the Previous Edit Location

Problem

While editing, you switched to another editor to make sure that you had the name of a variable right. But you have 15 editors open. How do you get back to the exact line of code you were editing?

Solution

Select Navigate → Go to Last Edit Location (Ctrl-Q), which moves you back to the location where you last made a change in an editor. You'll also see a button for this function in the toolbar when working with editors.

Discussion

Eclipse lets you easily move between edit locations, a valuable addition to any programmer's toolset. If you can't see these buttons in your current perspective, you might need to add them by selecting Window → Customize Perspective → Other → Editor Navigation).

2.8 Linking Views to Editors

Problem

The items selected in a view are not tied closely to whichever editor is open. Switching editors, or even closing an editor, does not change the selection in a view by default.

Solution

Click the Link with Editor button.

Discussion

Many views, such as the Resource perspective's Navigator view and the Java perspective's Package Explorer, enable you to synchronize with editors. To synchronize, click the Link with Editor button shown in Figure 2-8.

By linking a view with an editor, you tie that view to the editor so that the view will always show the file currently being edited.

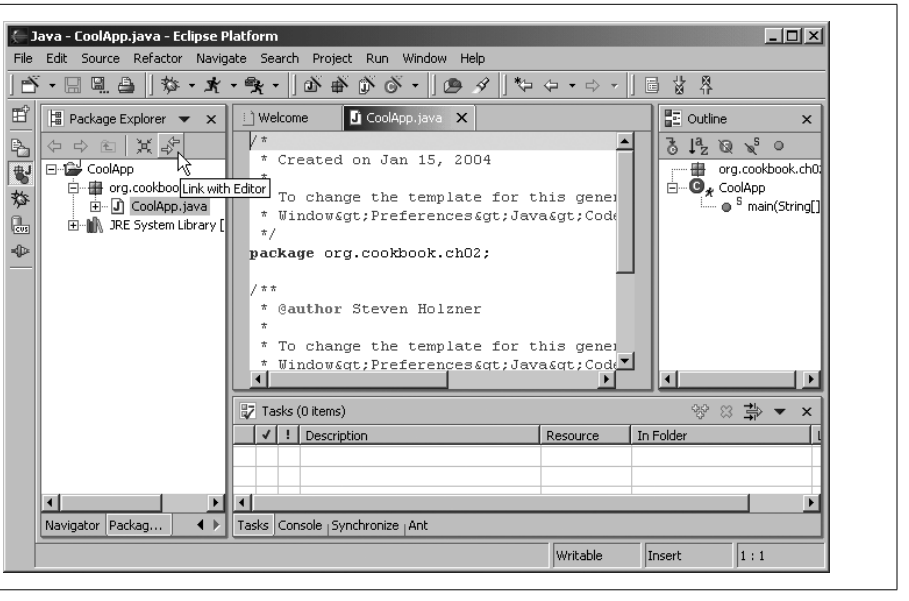

Figure 2-8. The Link with Editor button

See Also

Chapter 1 of *Eclipse* (O'Reilly).

2.9 Reordering View and Editor Tabs

Problem

You've got more than a dozen editors open, and the two you're working with are at opposite ends of the stack. As a result, you must scroll back and forth over all the editor tabs to work with the two editors you need. Isn't there a better way?

Solution

You can rearrange the order of the view and editor tabs as you prefer in Eclipse. Just drag the tab(s) to the order you want.

Discussion

This is a useful skill that few Eclipse developers know about, even those who use Eclipse everyday. When you open a large number of editors, as when you're working with multiple projects, you might find yourself surfing the editor tabs to find the ones you want. To avoid that, you can drag and drop those tabs in the order you want, grouping the editors you're currently working with.

2.10 Navigating from an Editor to a View

Problem

It's easy to navigate from a file in a view to an editor showing the file. Just double-click the file in the view. However, what if you want to go backward; from the editor showing a file to the view showing that file?

Solution

Select Navigate → Show In, and select the view you want from those that appear in the submenu.

Discussion

For example, if you're editing a Java file, select Navigate → Show In, and the Package Explorer will open. The file you're currently editing will be selected.

2.11 Creating a Key Binding

Problem

You find yourself performing a specific task, such as adding a bookmark or adding tasks to the task list, over and over. You want to assign a simple key combination to such tasks.

Solution

You can customize Eclipse by adding key combinations to dozens of Eclipse tasks.

Discussion

To see which tasks are available, select Window → Preferences → Workbench → Keys, and choose the task you want to automate, such as adding a bookmark. Then choose a key sequence you want to associate with the task, as shown in Figure 2-9.

2.12 Displaying More Resource Information with Icons

Problem

You want to turn on label decorators to get additional information about icons and buttons.

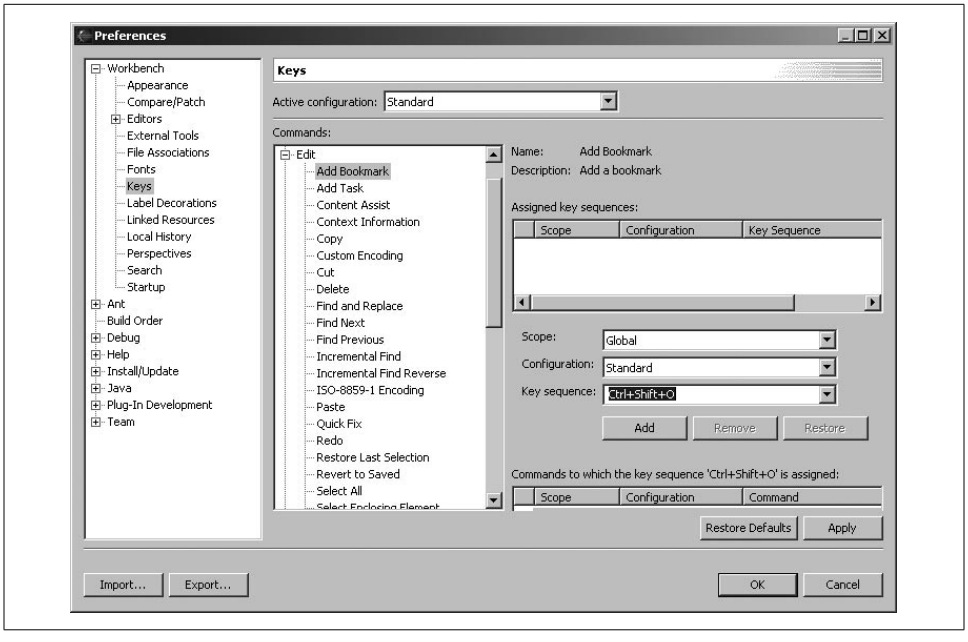

Figure 2-9. Assigning a key combination to a task

Solution

Select Window → Preferences → Workbench → Label Decorations.

Discussion

Label decorations augment the standard Eclipse icons displayed in various views. For example, if you archive a code file in a CVS repository and have label decorations turned on, the file's icon displays a small gold cylinder that isn't there otherwise.

To turn on label decorations, select Window → Preferences → Workbench → Label Decorations, and select which decoration you want to use, as shown in Figure 2-10.

We're going to use label decorations with Eclipse and CVS in Chapter 6. Once you've turned on label decorations, you can see at a glance which files have been stored in the CVS repository.

See Also

Recipe 6.6 on labeling files using version control.

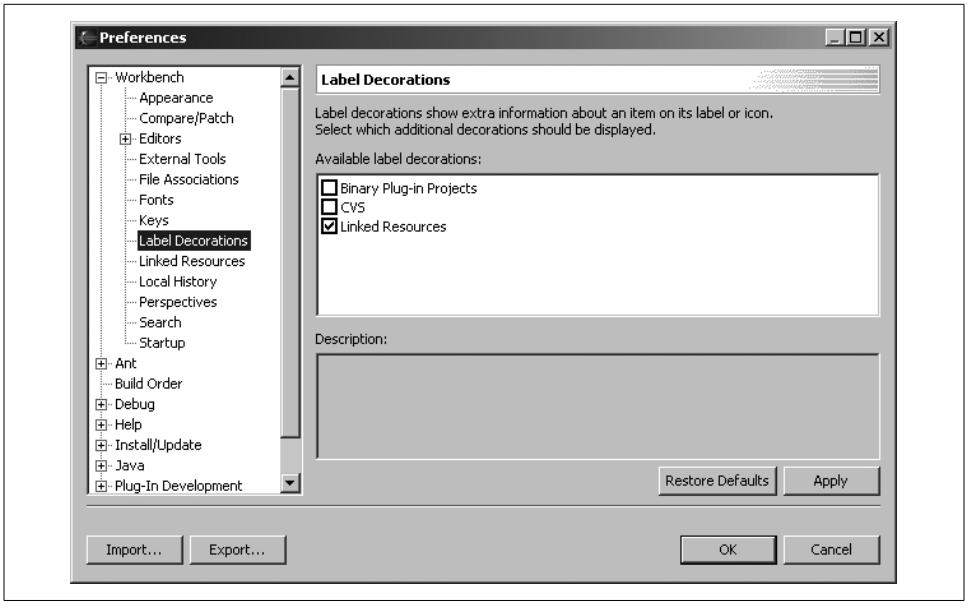

Figure 2-10. Selecting label decorations

2.13 Using a Different Workspace

Problem

You want to use a nondefault workspace for an Eclipse project.

Solution

Uncheck the "Use default" checkbox when you name the project in the first pane of the New Project dialog, and fill in the directory you want to use instead.

Discussion

When you create a new Eclipse project, Eclipse stores it in its *workspace* directory by default. You can, however, make it use another directory, regardless of the type of Eclipse project you're creating. To select a different location, uncheck the "Use default" checkbox when you name the new project in the first pane of the New Project dialog. Then fill in the directory you want to use instead.

You can see how this works in Figure 2-11, in which a new Java project is created and stored in the directory *c:\myworkspace*.

Using a nondefault workspace is a good idea in many cases. For example, you might be working with code on a network. In this case, you should store the code where other software can use it (in Chapter 8, we're going to store and develop code where

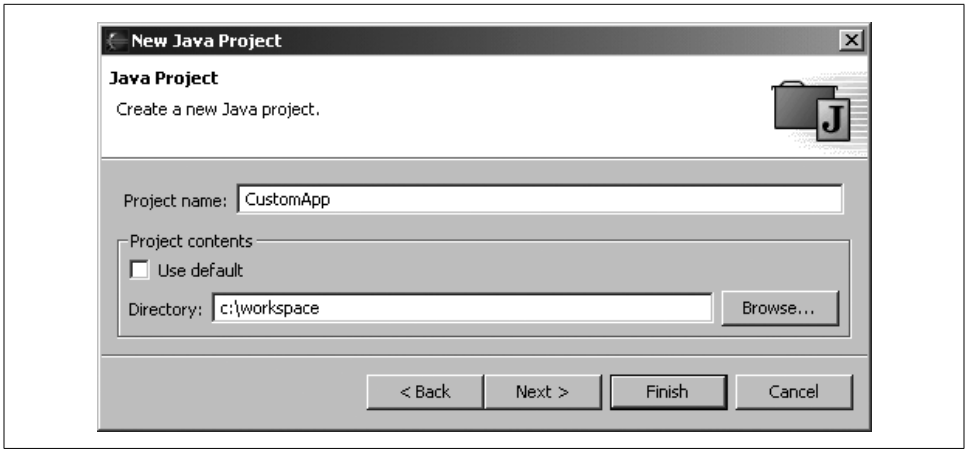

Figure 2-11. Creating a custom workspace

the Tomcat web server can find it, which means using the Tomcat installation's *webapps* directory as the workspace). It's also a good idea to use a nondefault workspace if you want to keep some software, as in a beta version you're developing, separate from the current version of the software.

2.14 Creating a Task

Problem

Eclipse tasks in the Tasks view serve as reminders of things you need to do. For example, compilation errors appear in the Tasks view; you can click them to jump to the lines containing the errors. So, how can you add and manage your own tasks to the Tasks view?

Solution

You can create a task in the Tasks view by right-clicking the view and selecting New Task. This opens the New Task dialog, in which you can create the task.

Discussion

The New Task dialog is shown in Figure 2-12. Enter the name of the new task, select a priority level, and click OK.

You also can create a new task by right-clicking the marker bar in an editor and selecting Add Task. You can also click the button with three + signs in the Tasks view's toolbar.

The new task will appear in the Tasks view, as shown in Figure 2-13.

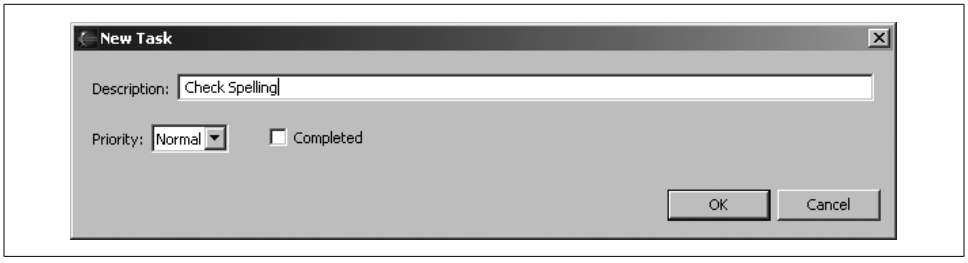

Figure 2-12. Creating a new task

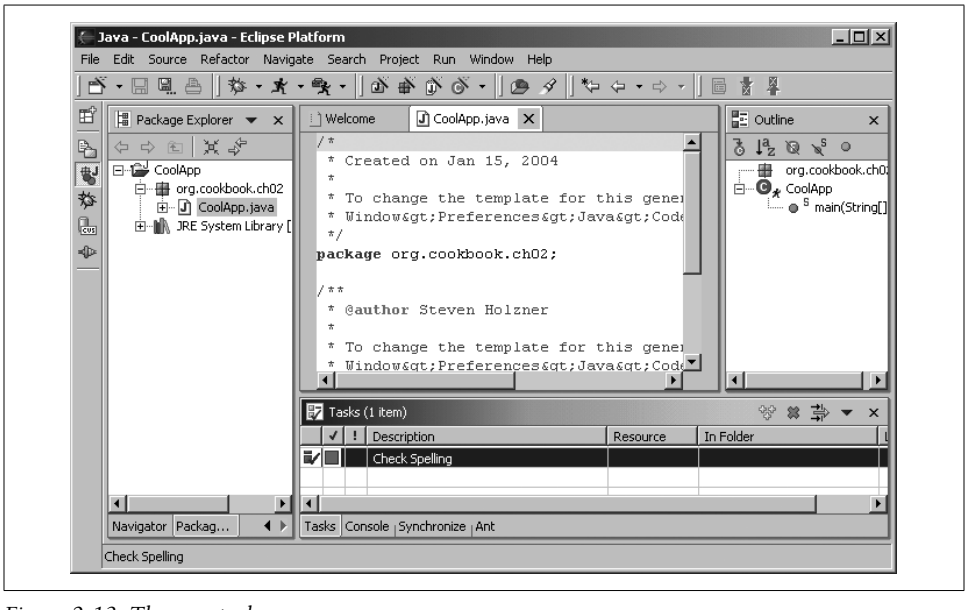

Figure 2-13. The new task

You can delete tasks you add to the Tasks view by right-clicking them and clicking the Delete button or by selecting the tasks and pressing the Delete key.

Tasks also have a completion status, which you can set by right-clicking the tasks and clicking Mark Completed. When you want to delete all completed tasks, right-click the Tasks view, and click Delete Completed Tasks.

See Also

Chapter 1 of *Eclipse* (O'Reilly).

2.15 Creating a Bookmark

Problem

You want to come back to a location in a file at a later time.

Solution

Create a bookmark by right-clicking the marker bar in an editor and specifying a name for the bookmark in the New Bookmark dialog.

Discussion

You're editing code, and suddenly it's quitting time. How can you save your place? Create a bookmark, and come back to it later by double-clicking that bookmark in the Bookmarks view. You create the bookmark by right-clicking the marker bar and naming the bookmark in the New Bookmark dialog. The bookmark will appear in the marker bar, as shown in Figure 2-14.

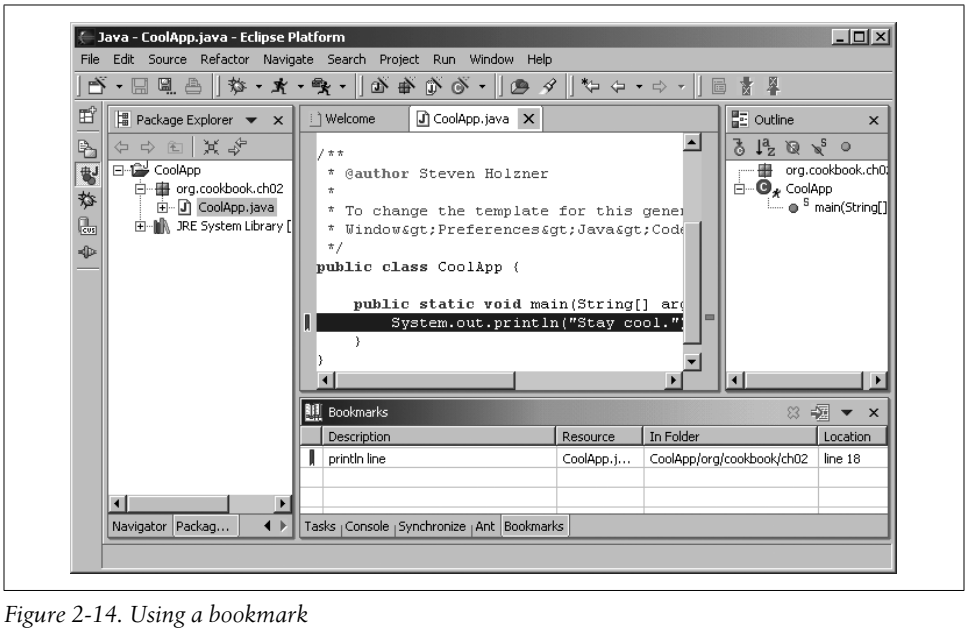

Figure 2-14. Using a bookmark

When you want to come back to the bookmarked location later, just open the Bookmarks view by selecting Window → Show View → Other → Basics → Bookmarks. You can see the bookmark we created in the Bookmarks view at the bottom of Figure 2-14; double-clicking it opens the bookmarked line in an editor.

As you'd expect, bookmarks work even if the file with the bookmark is closed; when you double-click the bookmark, that file opens automatically.

2.16 Creating a Fast View

Problem

Territory is always at a premium in IDEs; views can be stacked, but when you've got too many of them, it's frustrating to have to search for the right tab.

Solution

Use fast views to free some space in the Eclipse window.

Discussion

Transforming a view into a fast view adds it as an icon in the shortcut bar (that's the bar at extreme left in Eclipse, where the perspective icons appear). Clicking a fast view's icon makes it slide open until you're done with it. When you click outside it, it will slide closed automatically. For example, you can see the Bookmarks view made into a fast view in Figure 2-15.

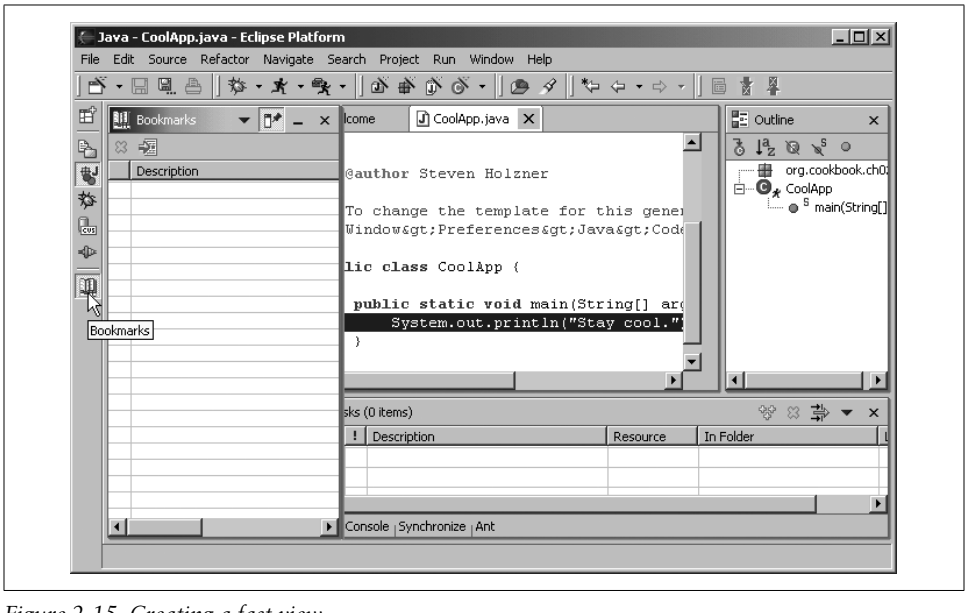

Figure 2-15. Creating a fast view

You turn a view into a fast view by selecting Fast View in its system menu, and you reach a view's system menu by right-clicking its name in its titlebar. When you turn a normal view into a fast view, an icon for the fast view appears on the shortcut bar. Right-clicking a fast view icon in the shortcut bar and deselecting Fast View restores it.

2.17 Customizing Help

Problem

You're browsing the Eclipse help system, and some of the help pages are very long. You want to search the text in those pages, but you notice that the Eclipse help browser doesn't let you search the current page.

Solution

Use your own browser instead.

Discussion

Eclipse uses a custom version of the Tomcat web server to display help documentation. While the help system is open, you can find the direct URL of the help topic you're looking at. After launching your own browser, you can navigate to that URL.

To get the URL of a help topic, right-click the topic, and select Create Shortcut, which creates a shortcut to the help topic. For example, my URL for help on bookmarks is *http://127.0.0.1:1203/help/index.jsp?topic=/org.eclipse.platform.doc.user/concepts/cbookmrk.htm*. You can double-click the shortcut or enter the URL in a browser to view the help page in your browser, as shown in Figure 2-16. You're free to search individual Eclipse help pages with your browser's Edit menu.

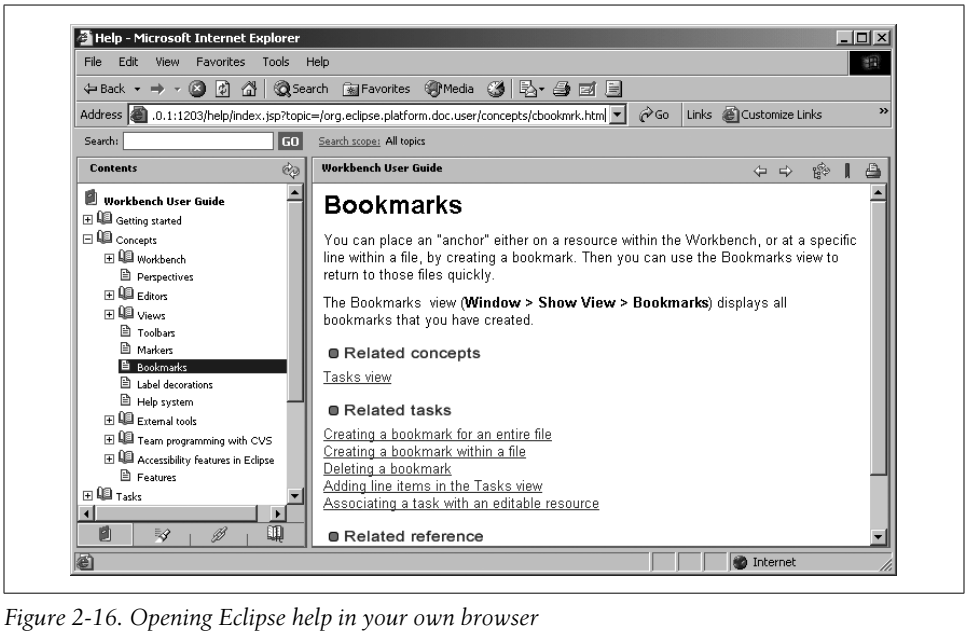

Figure 2-16. Opening Eclipse help in your own browser

A recent addition to Eclipse enables you to bookmark help topics in the Eclipse help system. To create a bookmark, click the Bookmark Document button on the toolbar, and the bookmark will appear in the Bookmarks tab. Just double-click a bookmark to go back to the help topic indicated.

You can even customize the Eclipse help system to use your favorite browser by default. Just select the browser option you want in the Window → Preferences → Help page, as shown in Figure 2-17.

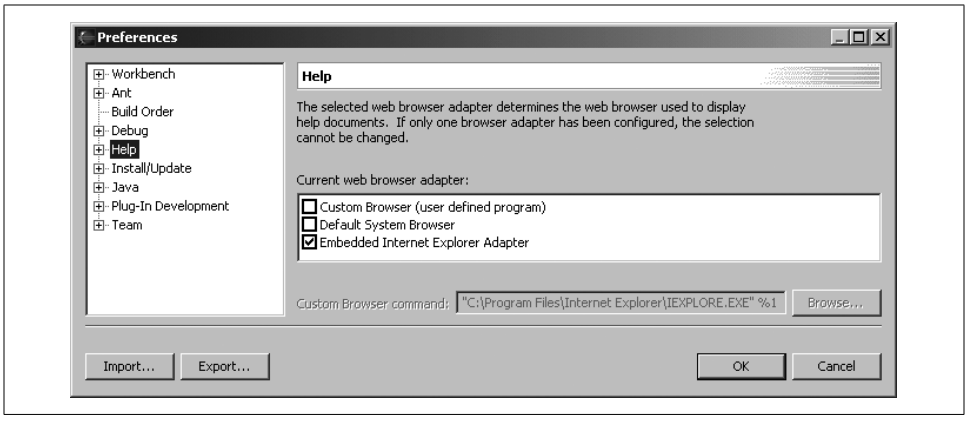

Figure 2-17. Configuring the help browser

You say you don't want to wait for the help system to open? What you're actually waiting for is the Tomcat web server to start so that it can serve help documents to the Eclipse Help browser. If you just want to check whether some topics match what you're looking for, click the Search button in the Eclipse toolbar and the Help Search button in the Search dialog. Enter the text you want help on, and click Search. The Search view then appears with matches in the help system. Here, you can see if your search returned any interesting matches to your query, all without having to launch Tomcat. However, if you see a match you're interested in, the help browser has to start up for you to view the topic's contents.

2.18 Restoring Deleted Resources

Problem

A needed file was deleted by mistake, and you want to restore it.

Solution

No problem. Restore the file using the container project's Restore from Local History context menu item.

Discussion

To restore a deleted file, right-click the file's project, and select Restore from Local History, opening the dialog shown in Figure 2-18. In the figure, a *.class* file that was deleted is restored.

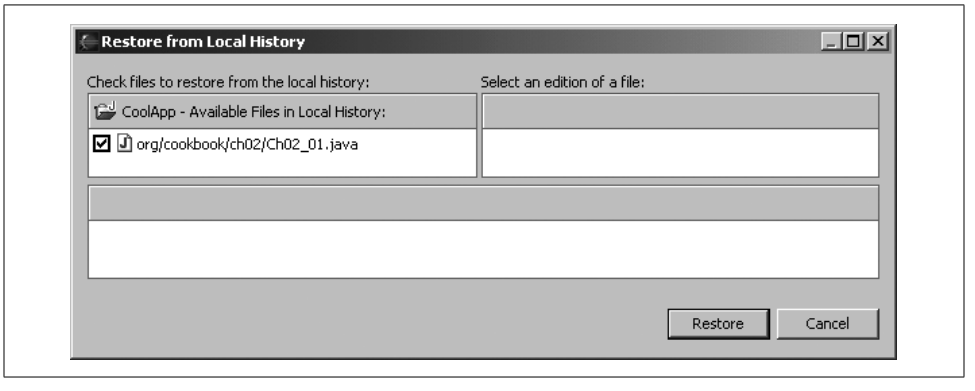

Figure 2-18. Restoring a deleted file

Because the Java perspective's Package Explorer doesn't display all files, including *.class* files, you'll need to switch to the Navigator view in the Java perspective or use the Resource perspective if you want to restore *.class* files.

2.19 Customizing a Perspective

Problem

You want to change the menu choices and toolbar items in a perspective.

Solution

Use the Customize Perspective dialog, which you can open by selecting Window → Customize Perspective. The dialog is shown in Figure 2-19.

After you've set Eclipse preferences, you can export those preferences so that others can use them as well. To do this, click the Import and Export buttons in the Preferences dialog.

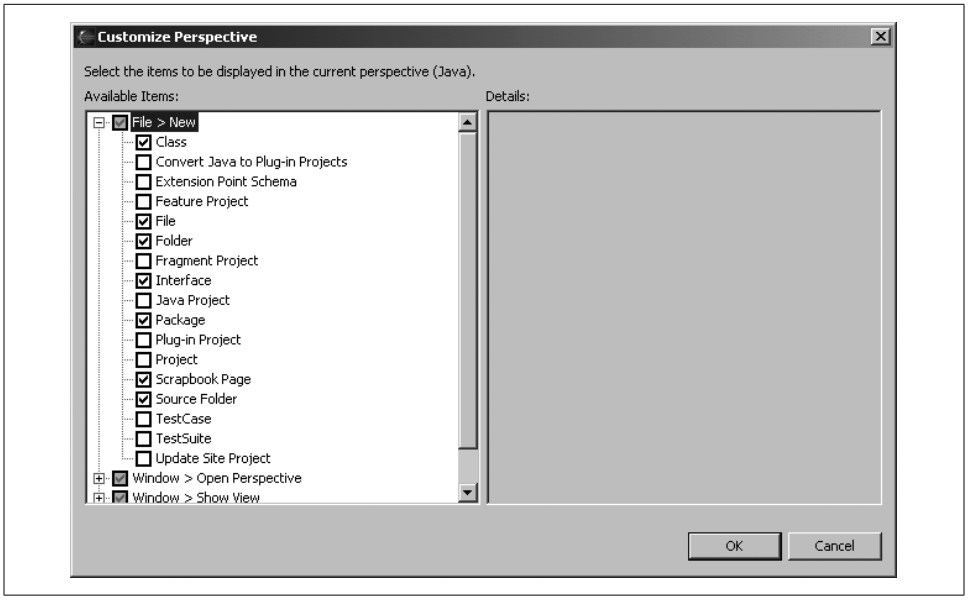

Figure 2-19. Customizing a perspective

The Preferences dialog enables you to customize menu choices and toolbar items, but it doesn't really enable you to "customize" perspectives (and in fact, the items you can add to menus usually are available under the Other menu choice anyway). To learn how to customize a perspective, see Recipe 2.21.

See Also

Recipe 2.20 on restoring a perspective; Recipe 2.21 on creating a new perspective.

2.20 Restoring a Perspective

Problem

One of Eclipse's perspectives was rearranged, and too many views were closed. How do you restore the perspective to its original condition?

Solution

If a perspective becomes scrambled to the point of being unrecognizable, you can restore it to its original state by selecting Window → Reset Perspective.

Discussion

Eclipse always keeps numerous backups and restore points in its memory. In addition to those platform- and file-specific backups, Eclipse also has default starting points for the windows and perspectives of editors and other components. Reset Perspective gives you that behavior with just a mouse-click and is helpful when things become cluttered and confused.

2.21 Creating a New Perspective

Problem

None of the built-in perspectives is quite right for you. You want to mix and match to create your own custom perspective.

Solution

No problem. Just open a perspective that's close to the one you want to create, add new views and close the ones you don't want, and save the new perspective by selecting Window → Save Perspective As.

Discussion

The Save Perspective As dialog is shown in Figure 2-20. This new perspective, named Debug2, adds the Navigator view to the Debug perspective.

Now you're free to open your new perspective whenever you want, as shown in Figure 2-21. Very cool.

To delete a custom perspective, select Window → Preferences → Workbench → Perspectives, choose the perspective you want to get rid of, and click Delete.

Being able to create your own perspectives is very cool, and is something virtually no other IDE offers. You've got the opportunity here to mix and match views and create something totally new. Get creative!

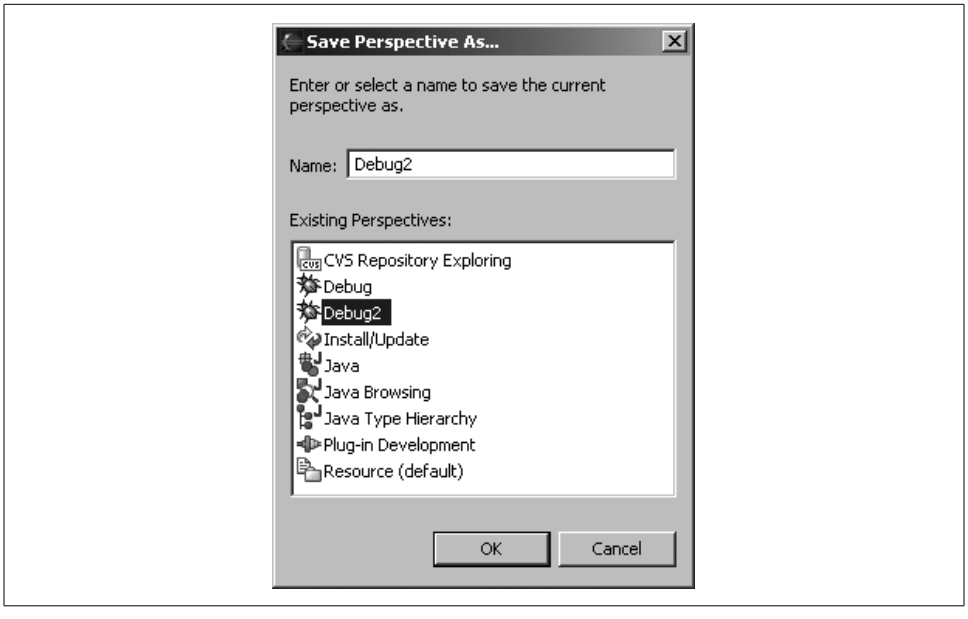

Figure 2-20. Creating a new perspective

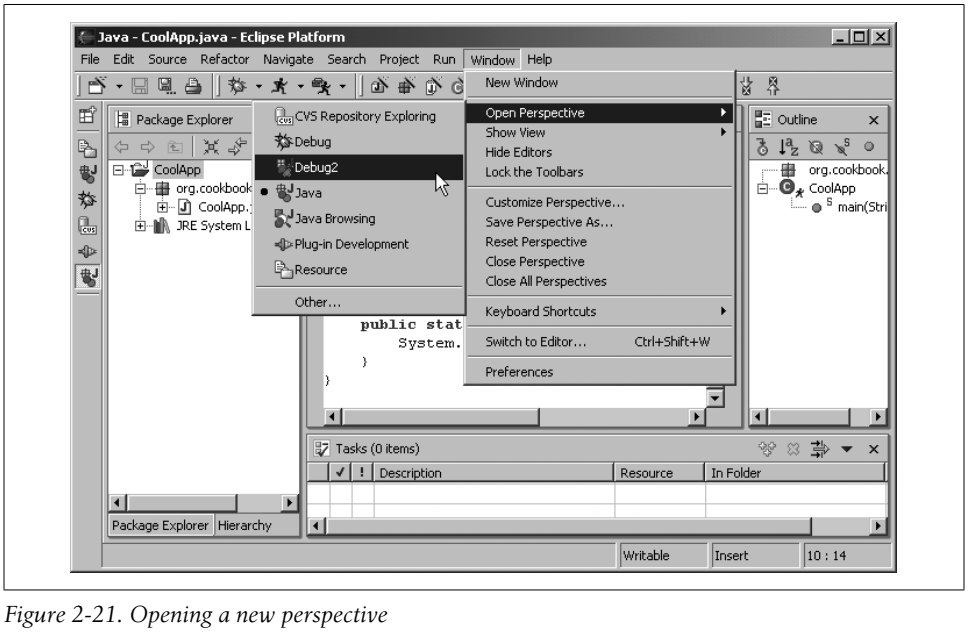

Figure 2-21. Opening a new perspective