



eclipse

integrated development environment

tutorialspoint

SIMPLY EASY LEARNING

www.tutorialspoint.com



<https://www.facebook.com/tutorialspointindia>



<https://twitter.com/tutorialspoint>

About the Tutorial

Eclipse is an integrated development environment (IDE) for Java and other programming languages like C, C++, PHP, and Ruby etc. Development environment provided by Eclipse includes the Eclipse Java development tools (JDT) for Java, Eclipse CDT for C/C++, and Eclipse PDT for PHP, among others.

This tutorial will teach you how to use Eclipse in your day-2-day life while developing any software project using Eclipse IDE. We will give special emphasis on Java project.

Audience

This tutorial has been prepared for beginners to help them understand basic functionality of Eclipse tool. After completing this tutorial, you will find yourself at a moderate level of expertise in using Eclipse IDE from where you can take yourself to next levels.

Prerequisites

We assume you are going to use Eclipse IDE to handle all levels of Java projects development. So it will be good if you have knowledge of software development using any programming language specially Java programming.

Copyright & Disclaimer

© Copyright 2015 by Tutorials Point (I) Pvt. Ltd.

All the content and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book can retain a copy for future reference but commercial use of this data is not allowed. Distribution or republishing any content or a part of the content of this e-book in any manner is also not allowed without written consent of the publisher.

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com

Table of Contents

About the Tutorial.....	i
Audience	i
Prerequisites	i
Copyright & Disclaimer.....	i
Table of Contents	ii
 1. OVERVIEW.....	 1
What is Eclipse?	1
Licensing	1
Eclipse Releases	1
 2. INSTALLATION	 3
Downloading Eclipse	3
Installing Eclipse.....	3
Launching Eclipse	4
 3. EXPLORE WINDOWS.....	 5
Parts of an Eclipse Window	5
Using Multiple Windows	6
 4. EXPLORE MENUS.....	 7
Typical Eclipse Menus	7
Brief Description of Menus.....	8
Customizing Menus.....	8
 5. EXPLORE VIEWS.....	 9
About Views.....	9
Organizing Views.....	9
Moving Views.....	9

Creating View Folders	10
Opening a view	10
6. PERSPECTIVES	13
What is a Perspective?	13
Opening a Perspective.....	13
Switching between Perspectives	13
Closing a Perspective.....	14
Customizing a Perspective	14
7. WORKSPACES	16
About Eclipse Workspace	16
UI Elements for Managing the Workspace	16
8. CREATE JAVA PROJECT	18
Opening the New Java Project wizard	18
Using the New Java Project wizard	18
Viewing the Newly Created Project	20
9. CREATE JAVA PACKAGE	21
Opening the New Java Package wizard.....	21
Using the New Java Package Wizard	21
Viewing the Newly Created Package	22
10. CREATE JAVA CLASS.....	23
Opening the New Java Class Wizard	23
Using the New Java Class Wizard	23
Viewing the Newly Created Java class	24
11. CREATE JAVA INTERFACE.....	25
Opening the New Java Interface Wizard.....	25

Using the New Java Interface Wizard	25
Viewing the Newly Created Java Interface	26
12. CREATE XML FILE	27
Opening the New XML File wizard.....	27
Using the New XML File wizard	28
Viewing the Newly Created XML File	29
13. JAVA BUILD PATH	30
Setting the Java Build Path	30
14. RUN CONFIGURATION	31
Creating and Using a Run Configuration	31
15. RUNNING A PROGRAM.....	33
Running a Java Program	33
16. CREATE JAR FILES	35
Opening the Jar File wizard	35
Using the Jar File wizard	35
17. CLOSE PROJECT	37
Why Close a Project?.....	37
How to Close a Project?	37
Closed Project in Package Explorer	38
18. REOPEN PROJECT	39
Reopening a Closed Project	39
19. BUILD PROJECT.....	40
Building a Java Project	40
20. DEBUG CONFIGURATION.....	42

Creating and Using a Debug Configuration	42
21. DEBUGGING A PROGRAM.....	44
Debugging a Java Program	44
22. PREFERENCES	48
Setting Preferences	48
23. CONTENT ASSIST	50
Using Content Assist.....	50
24. QUICK FIX	52
Using Quix Fix.....	52
25. HOVER HELP	54
Using Hover Help.....	54
26. SEARCH MENU	56
Searching the Workspace	56
27. NAVIGATION.....	58
Navigating the Eclipse Workspace.....	58
Open Type.....	58
Open Type in Hierarchy	60
Open Resource.....	61
28. REFACTORING	63
Refactoring using Eclipse	63
29. ADD BOOKMARKS.....	64
About Bookmarks	64
Adding a Bookmark.....	64
Opening the Bookmarks View	64

Using the Bookmarks View.....	65
30. TASK MANAGEMENT	66
Managing Tasks.....	66
Opening the Tasks View	67
Using the Tasks View.....	67
31. INSTALL PLUGINS.....	69
Locating and Installing Plug-ins	69
32. CODE TEMPLATES.....	73
Using Code Templates	73
Modifying/Adding code templates.....	74
33. SHORTCUTS.....	75
About Shortcuts	75
34. RESTART OPTION.....	78
Restarting Eclipse	78
35. TIPS & TRICKS	79
36. WEB BROWSERS.....	81
Internal Web Browser	81

1. OVERVIEW

What is Eclipse?

In the context of computing, Eclipse is an integrated development environment (IDE) for developing applications using the Java programming language and other programming languages such as C/C++, Python, PERL, Ruby etc.

The Eclipse platform which provides the foundation for the Eclipse IDE is composed of plug-ins and is designed to be extensible using additional plug-ins. Developed using Java, the Eclipse platform can be used to develop rich client applications, integrated development environments, and other tools. Eclipse can be used as an IDE for any programming language for which a plug-in is available.

The **Java Development Tools (JDT)** project provides a plug-in that allows Eclipse to be used as a Java IDE, **PyDev** is a plugin that allows Eclipse to be used as a Python IDE, **C/C++ Development Tools (CDT)** is a plug-in that allows Eclipse to be used for developing application using C/C++, the **Eclipse Scala** plug-in allows Eclipse to be used an IDE to develop Scala applications and **PHPEclipse** is a plug-in to eclipse that provides complete development tool for PHP.

Licensing

Eclipse platform and other plug-ins from the Eclipse foundation is released under the Eclipse Public License (EPL). EPL ensures that Eclipse is free to download and install. It also allows Eclipse to be modified and distributed.

Eclipse Releases

Every year, since 2006, the Eclipse foundation releases the Eclipse Platform and a number of other plug-ins in June.

Codename	Year	Platform Version
Callisto	2006	3.2
Europa	2007	3.3
Ganymede	2008	3.4
Galileo	2009	3.5
Helios	2010	3.6

Indigo	2011	3.7
Juno	2012	3.8 and 4.2
Kepler	2013	4.3
Luna	2014	4.4.0

2. INSTALLATION

Downloading Eclipse

You can download eclipse from <http://www.eclipse.org/downloads/>. The download page lists a number of flavors of eclipse.



The capabilities of each packaging of eclipse are different. Java developers typically use Eclipse Classic or Eclipse IDE for developing Java applications.

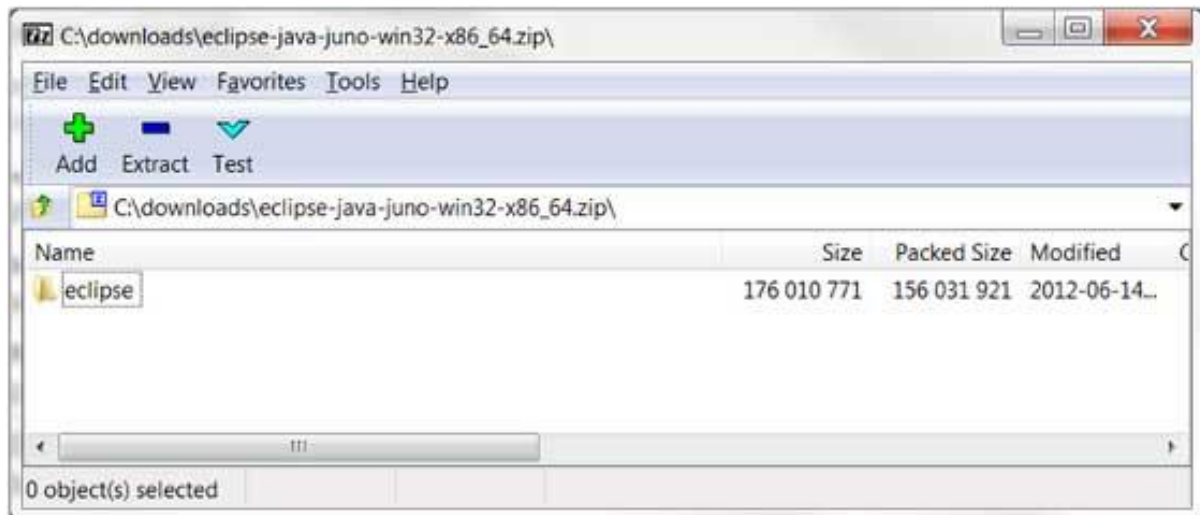
The drop down box in the right corner of the download page allows you to set the operating system on which eclipse is to be installed. You can choose between Windows, Linux and Mac. Eclipse is packaged as a zip file.

Installing Eclipse

To install on windows, you need a tool that can extract the contents of a zip file. For example you can use:

- **7-zip**
- **PeaZip**
- **IZArc**

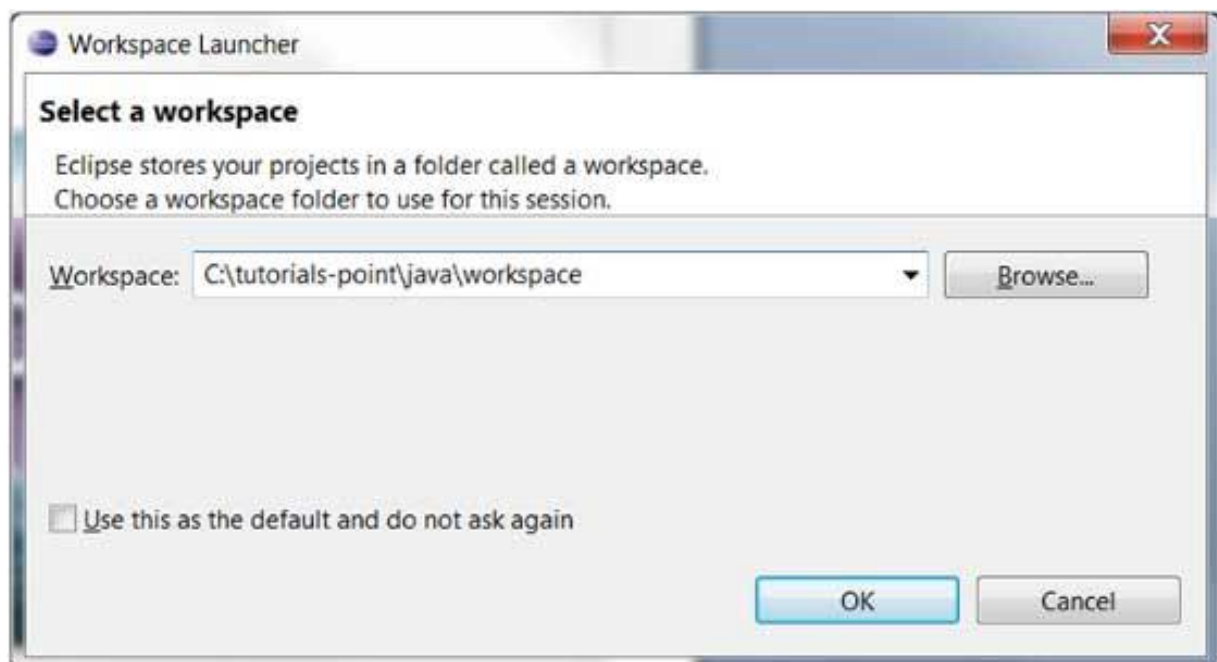
Using any one of these tools, extract the contents of the eclipse zip file to any folder of your choice.



Launching Eclipse

On the windows platform, if you extracted the contents of the zip file to c:\, then you can start eclipse by using c:\eclipse\eclipse.exe

When eclipse starts up for the first time it prompts you for the location of the workspace folder. All your data will be stored in the workspace folder. You can accept the default or choose a new location.



If you select "Use this as the default and do not ask again", this dialog box will not come up again. You can change this preference using the Workspaces Preference Page. See the [Preference tutorial](#) page for more details.

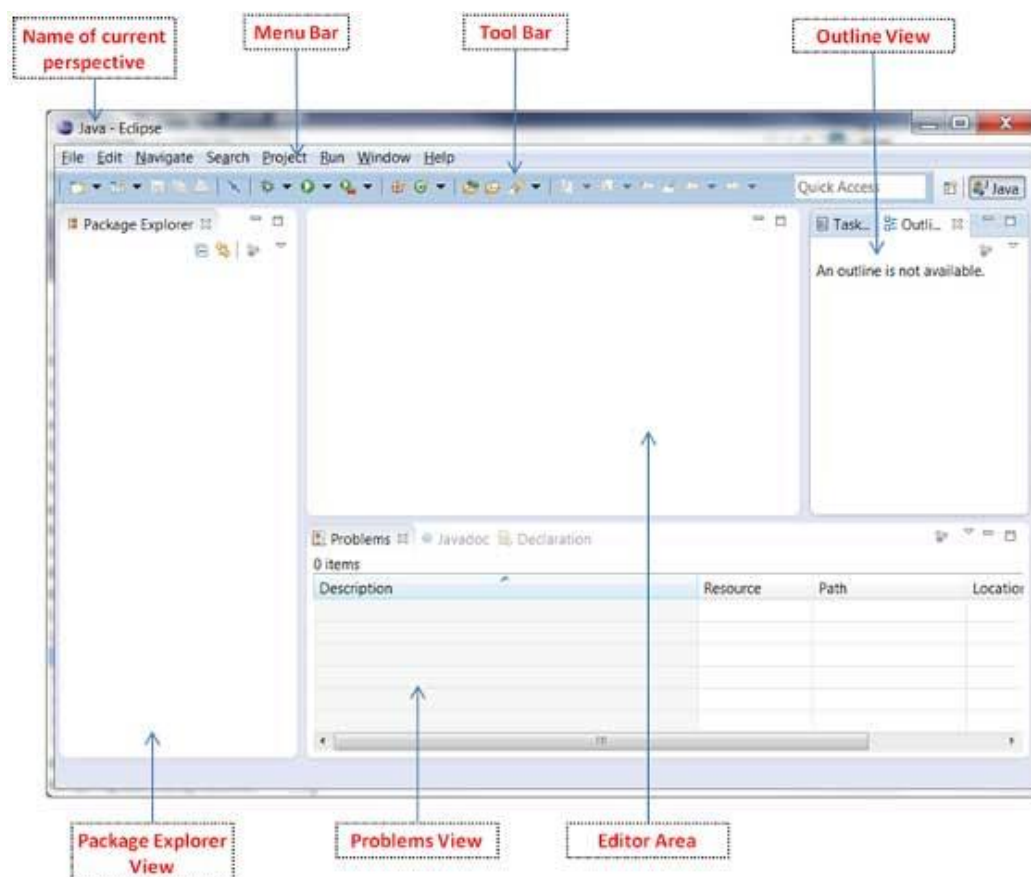
3. EXPLORE WINDOWS

Parts of an Eclipse Window

The major visible parts of an eclipse window are:

- Views
- Editors (all appear in one editor area)
- Menu Bar
- Toolbar

An eclipse perspective is the name given to an initial collection and arrangement of views and an editor area. The default perspective is called java. An eclipse window can have multiple perspectives open in it but only one perspective can be active at any point of time. A user can switch between open perspectives or open a new perspective. A perspective controls what appears in some menus and tool bars.



A perspective has only one editor area in which multiple editors can be open. The editor area is usually surrounded by multiple views. In general, editors are used to edit the

project data and views are used to view the project metadata. For example, the package explorer shows the java files in the project and the java editor is used to edit a java file.

The eclipse window can contain multiple editors and views but only one of them is active at any given point of time. The title bar of the active editor or view looks different from all the others.

The UI elements on the menu bar and tool bar represent commands that can be triggered by an end user.

Using Multiple Windows

Multiple Eclipse Windows can be open at the same time. To open a new window, click on the Windows menu and select the New Window menu item.

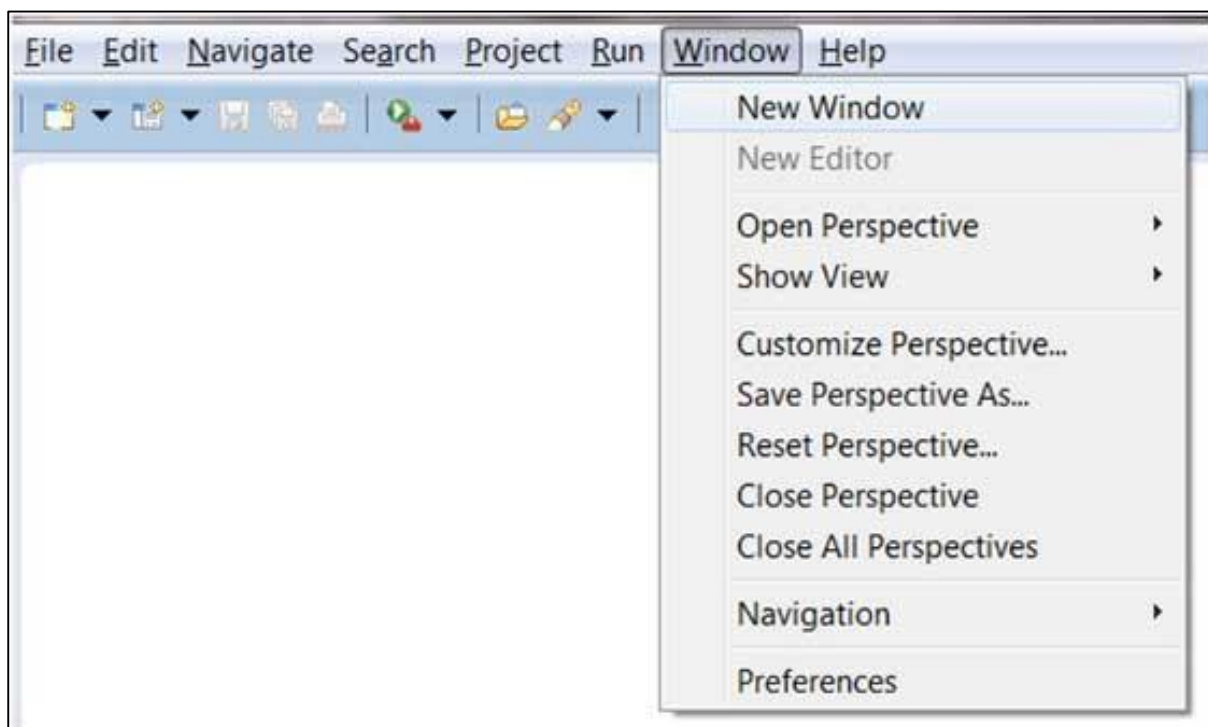
Each window can have a different perspective open in them. For example you could open two Eclipse windows one in the Java perspective and the other in the Debug perspective. The window showing the Java perspective can be used for editing the java code and the window showing the debug perspective can be used for debugging the application being developed.

4. EXPLORE MENUS

Typical Eclipse Menus

The typical menus available on the menu bar of an Eclipse window are:

- File menu
- Edit menu
- Navigate menu
- Search menu
- Project menu
- Run menu
- Window menu
- Help menu



Plug-ins can add new menus and menu items. For example when the java editor is open, you will see the Source menu and when the XML editor is open, you will see the **Design** menu.

Brief Description of Menus

Menu Name	Description
File	The File menu allows you to open files for editing, close editors, save editor content and rename files. Among the other things, it also allows you to import and export workspace content and shutdown Eclipse.
Edit	The Edit menu presents items like copy & paste.
Source	The Source menu is visible only when a java editor is open. It presents a number of useful menu items related to editing java source code.
Navigate	The Navigate menu allows you to quickly locate resources and navigate to them.
Search	The Search menu presents items that allow you to search the workspace for files that contain specific data.
Project	The menu items related to building a project can be found on the Project menu.
Run	The menu items on the Run menu allow you to start a program in the run mode or debug mode. It also presents menu items that allow you to debug the code.
Window	The Window menu allows you to open and close views and perspectives. It also allows you to bring up the Preferences dialog.
Help	The Help menu can be used to bring up the Help window, Eclipse Marketplace view or Install new plug-ins. The about Eclipse menu item gives you version information.

Customizing Menus

The visible menu items on a menu depend on the installed plug-ins and customization done using the [Customize Perspective](#) dialog box.

5. EXPLORE VIEWS

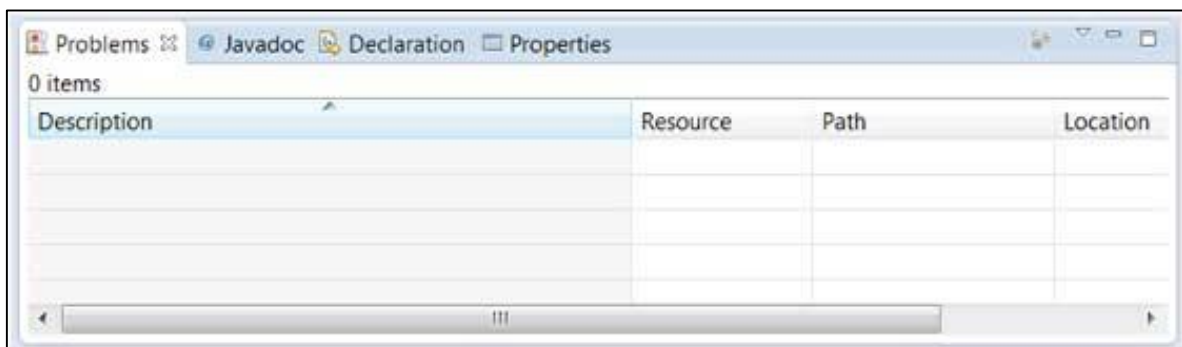
About Views

Eclipse views allow users to see a graphical representation of project metadata. For example the project navigator view presents a graphical representation of the folders and files associated with a project and properties view presents a graphical representation of an element selected in another view or editor.

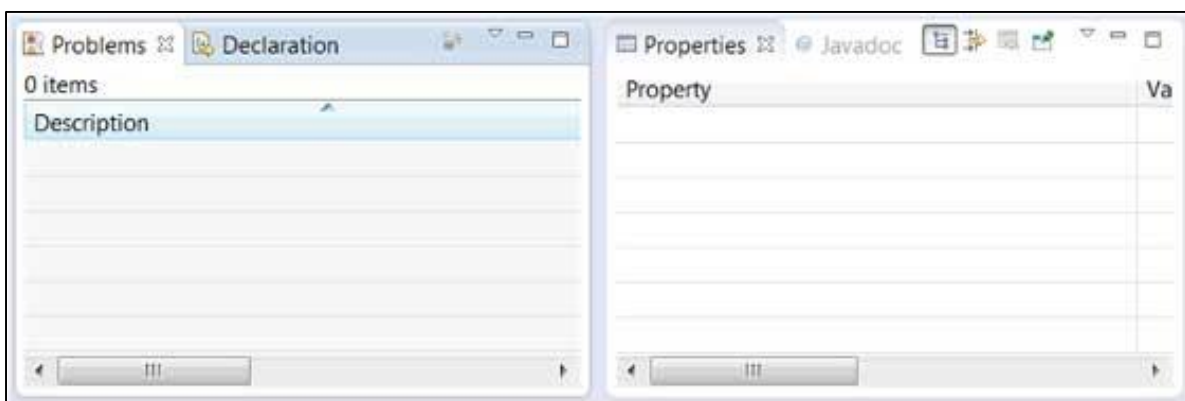
An eclipse perspective can show any number of views and editors. All editor instances appear in a single editor area, whereas views are placed inside view folders. A workbench window can display any number of view folders. Each view folder can display one or more views.

Organizing Views

The following picture shows four views arranged in a view folder.



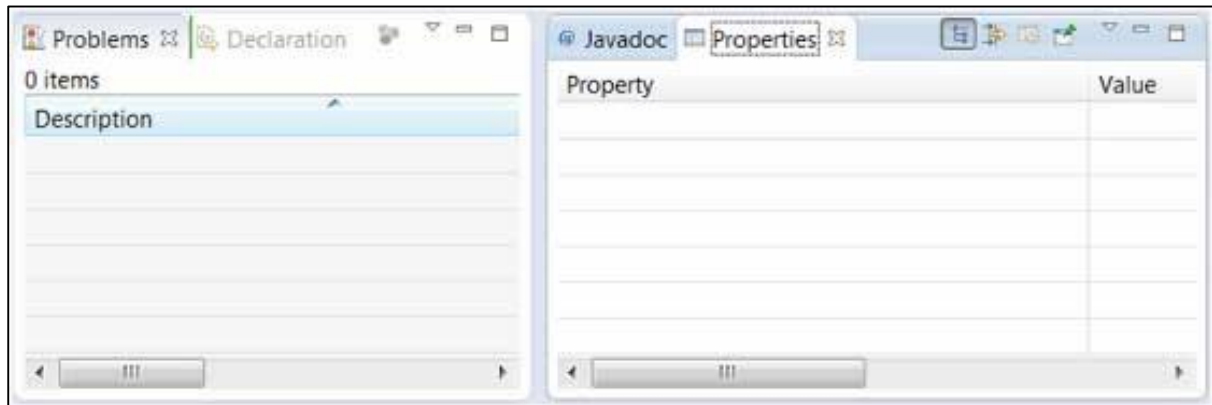
The picture given below shows the same four views arranged in two view folders.



Moving Views

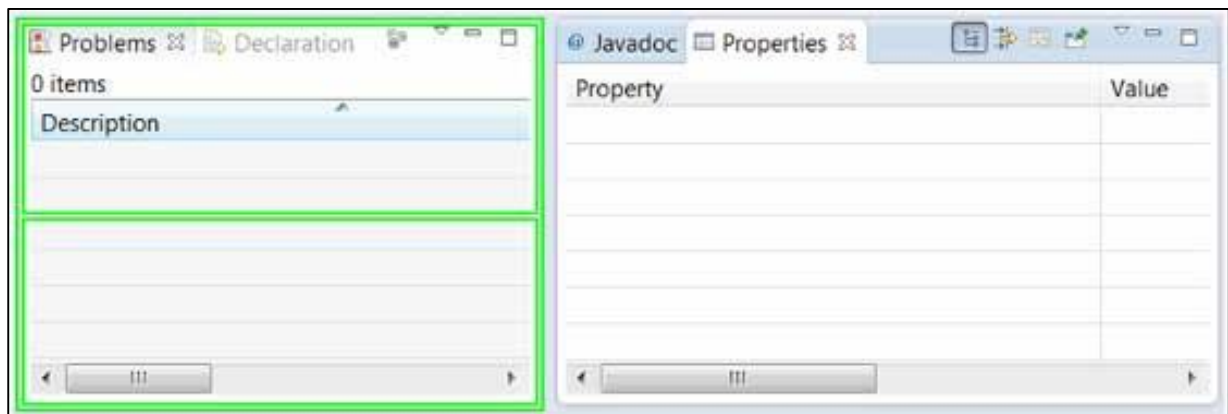
To move a view from one view folder to another, just click on the view title and drag to the title bar area of another view folder. The green line shown below is a result of dragging the title bar of the Properties view from one view folder to the title bar area of another

view folder. The Properties view can be moved to where the green line is by releasing the mouse button and sending out a drop event.



Creating View Folders

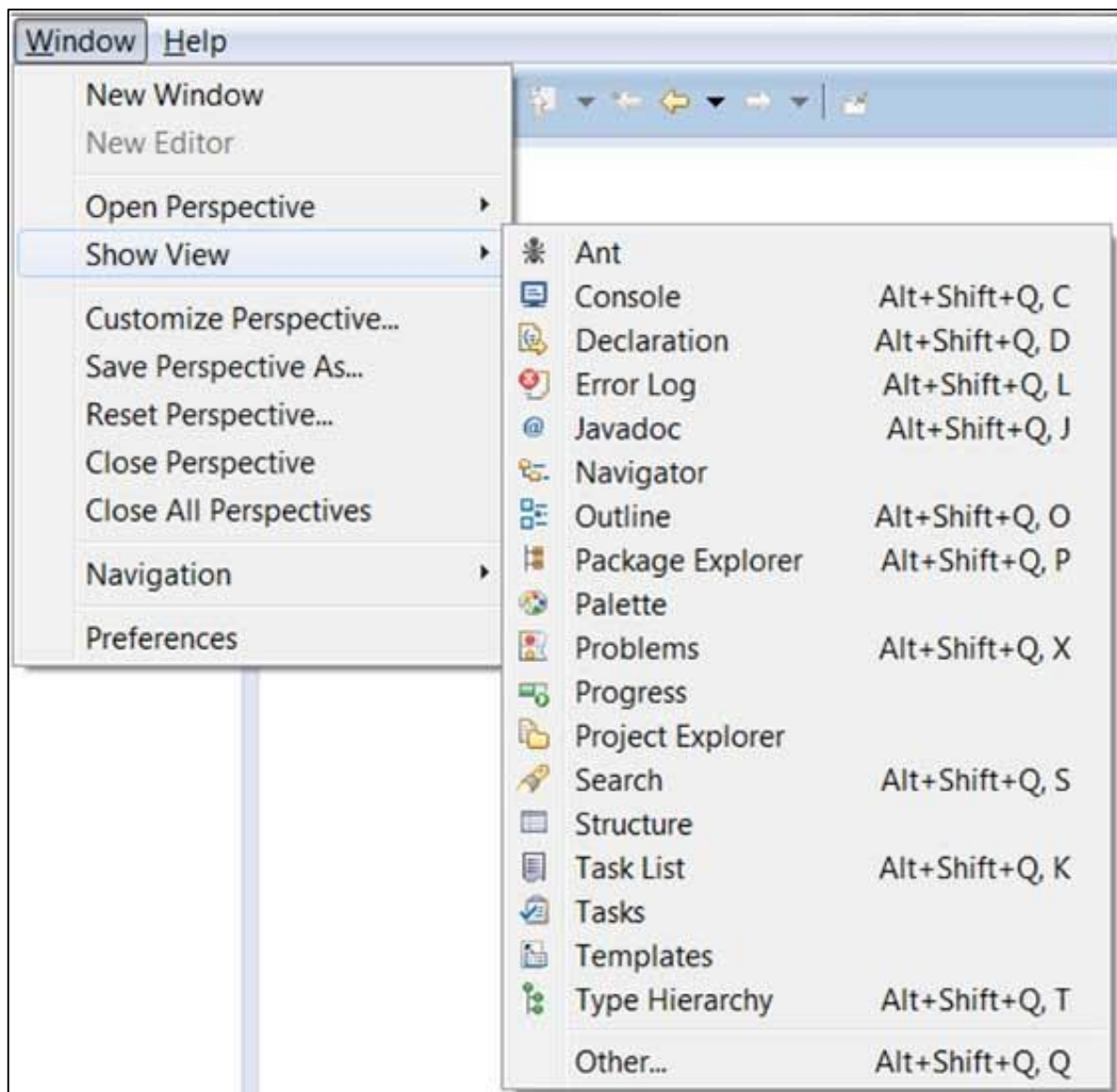
View folders can be dynamically created by dragging the title bar of a view to anywhere outside the editor area and title bar of another view folder. As you drag the title bar around, green lines will indicate where exactly the new view folder will be created.



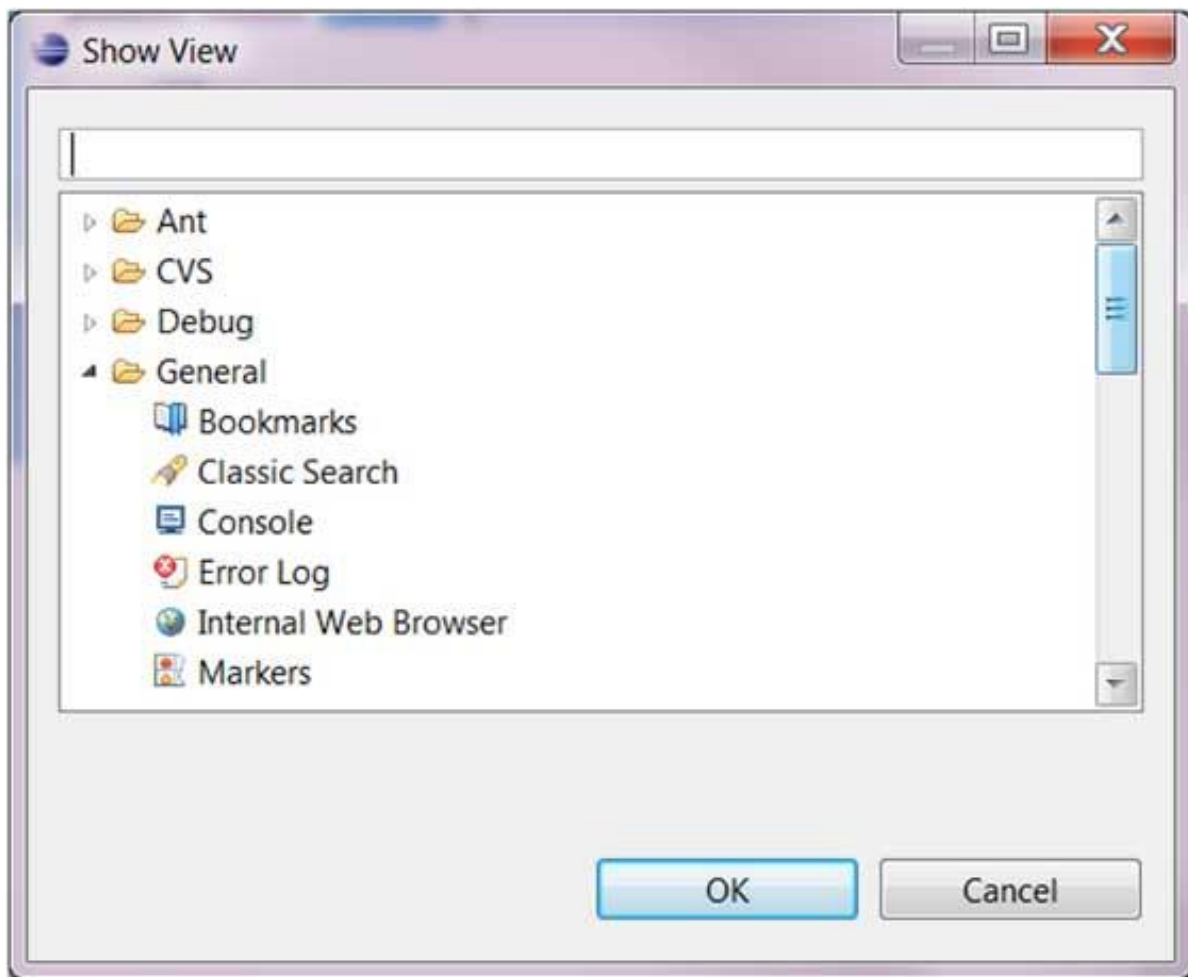
Moving the drag icon to the bottom of a window allows you to create a view folder that spans the entire width of the window. Moving the drag icon to the left or right edge of window allows you to create a view folder that spans the entire height of the window.

Opening a view

To open a view, click on the **Window** menu and select the **Show View** menu item.



Clicking on the **Other** menu item brings up the Show View dialog box that allows you to locate and activate a view.



The views are organized by category. To quickly locate a view, just type the name of a view into the filter text box. To open a view, select it and click on the OK button. The subsequent pages of this tutorial introduce you to a number of useful views.

6. PERSPECTIVES


What is a Perspective?

An eclipse perspective is the name given to an initial collection and arrangement of views and an editor area. The default perspective is called java. An eclipse window can have multiple perspectives open in it but only one perspective is active at any point of time. A user can switch between open perspectives or open a new perspective. The active perspective controls what appears in some menus and tool bars.

Opening a Perspective

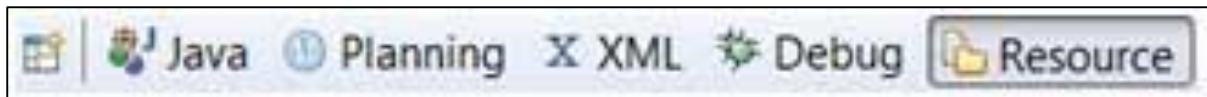
To open a new perspective, click on the Windows menu and select Open Perspective → Other



The Open Perspective dialog box shows all the available perspectives. The same dialog can be brought up by clicking on the Open Perspective button () on the tool bar.

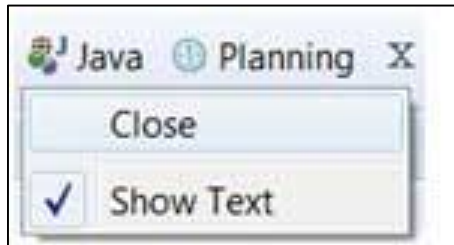
Switching between Perspectives

The most commonly used perspectives for java development are the Java perspective and Debug perspective. Users can switch between open perspectives by clicking on the Perspective name on the toolbar.



Closing a Perspective

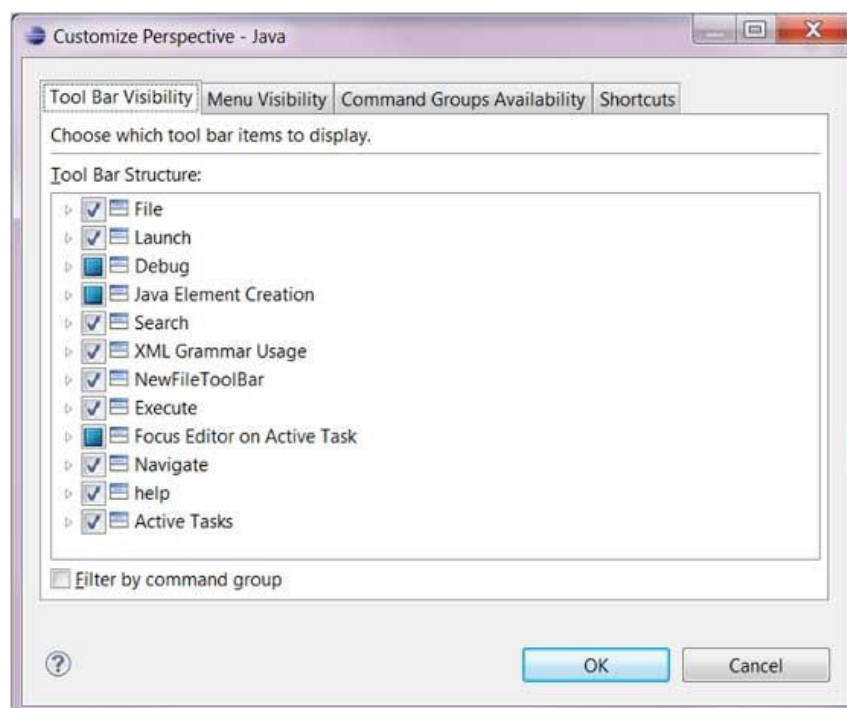
To close a perspective, right click on the perspective name in toolbar and select the Close menu item.



Customizing a Perspective

The customize perspective dialog can be used to customize a perspective. Customizing a perspective means:

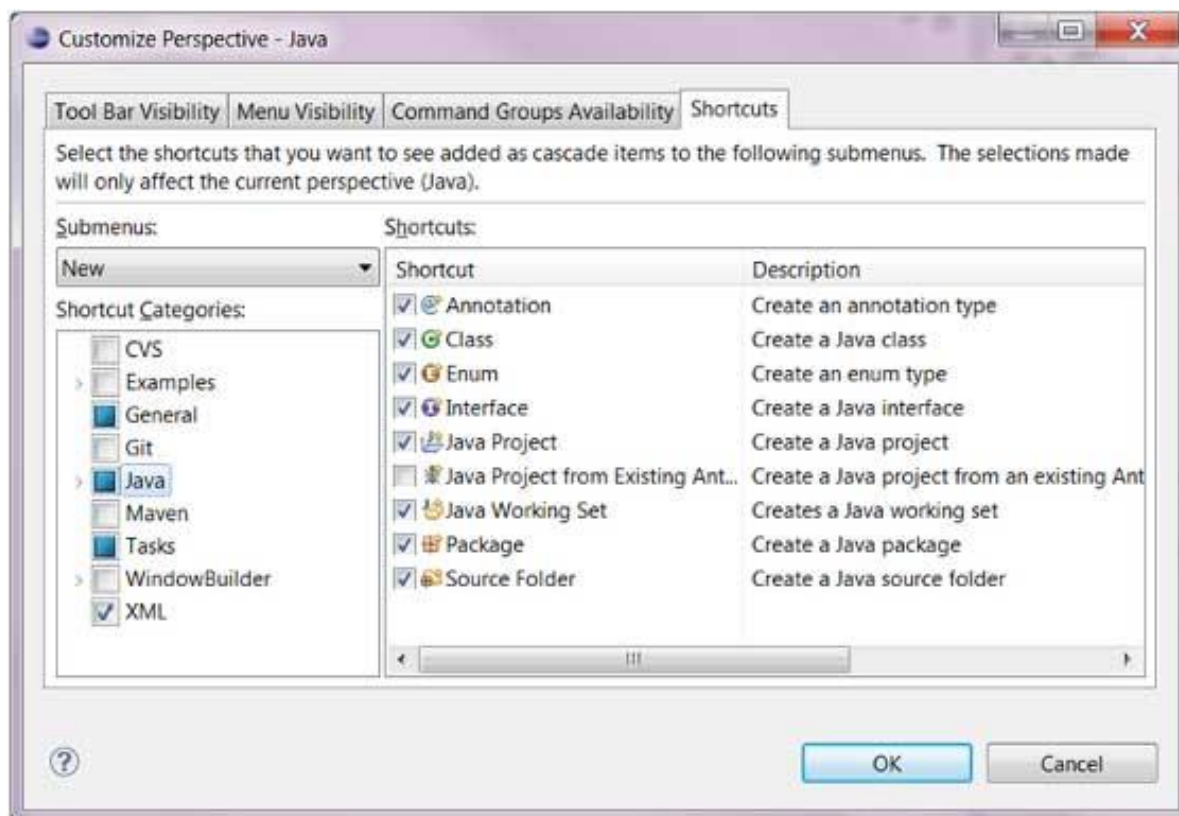
- Determining the icons visible on the toolbar when a perspective is active
- Determining the menu items visible when a perspective is active
- Determine the menu items in New submenu, Show View submenu and Open Perspective submenu



The tab descriptions of customize perspective are given below:

- The **Tool Bar Visibility** tab can be used to determine which icons are visible on the toolbar when a perspective is open.
- The **Menu Visibility** tab can be used to determine which menu items are visible when a perspective is active.
- The **Command Groups Availability** tab can be used to control the visibility of toolbar icons and menu items.
- The **Shortcuts** tab can be used to determine the menu items in New submenu, Show View submenu, and Open Perspective submenu.

Based on the Shortcuts selection in the picture given below, to bring up the "New Java Project from Existing Ant Build File" wizard users have to bring up the New submenu (File → New), click on Other, expand the Java category and then select "Java Project from Existing Ant Build File". On the other hand, to start the New Class wizard they can bring up the New submenu (File → New) and select the Class menu item because its selected in the picture given below. By selecting the "Java Project from Existing Ant Build File" check box this item will also appear under the New menu.



7. WORKSPACES

About Eclipse Workspace

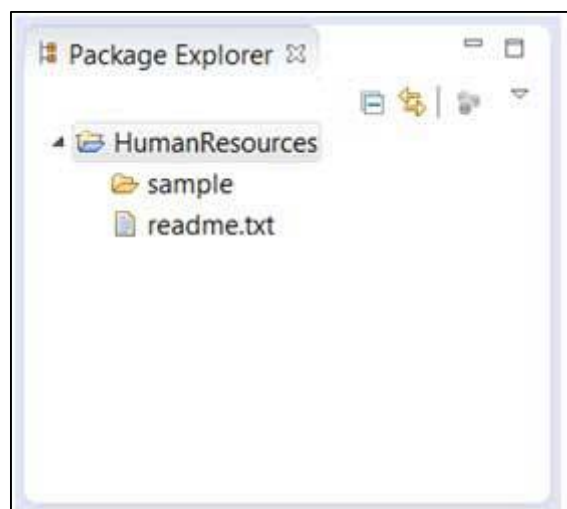
The eclipse workspace contains resources such as:

- Projects
- Files
- Folders

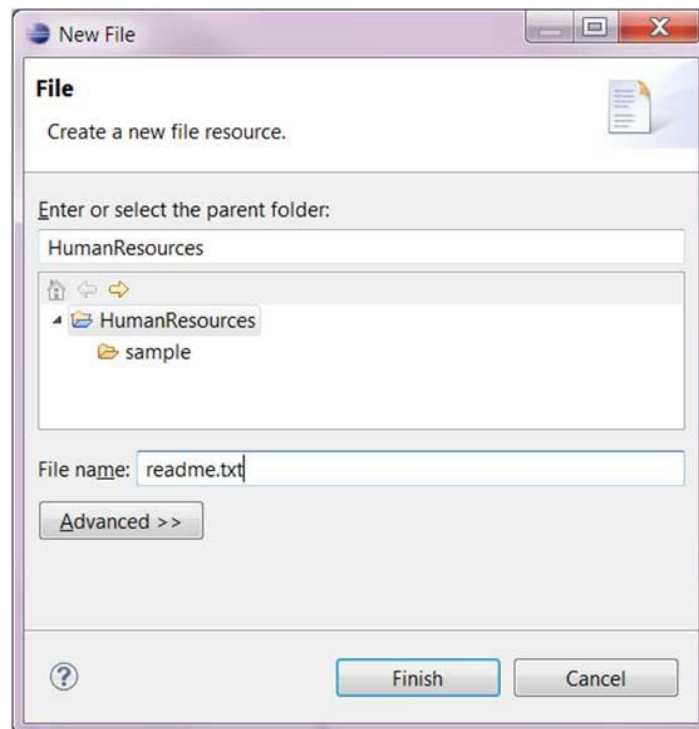
The workspace has a hierarchical structure. Projects are at the top level of the hierarchy and inside them you can have files and folders. Plug-ins use an API provided by the resources plug-in to manage the resources in the workspace.

UI Elements for Managing the Workspace

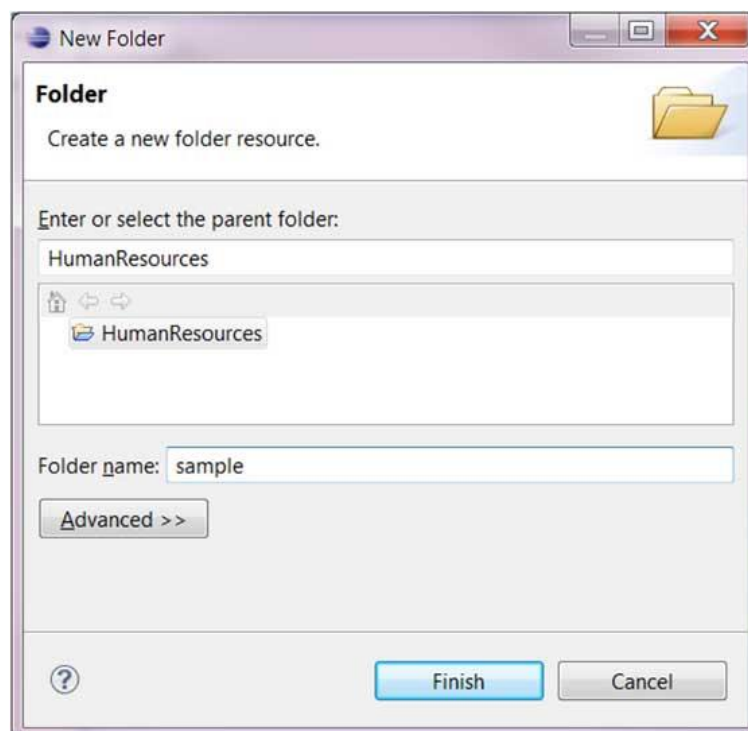
Users use the functionality provided by views, editors, and wizard to create and manage resources in the workspace. One among the many views that show the content of the workspace is the Project Explorer view.



The File Wizard (File → New → File) can be used to create a new file.




The Folder Wizard (File → New → Folder) can be used to create a new folder.



8. CREATE JAVA PROJECT

Opening the New Java Project wizard

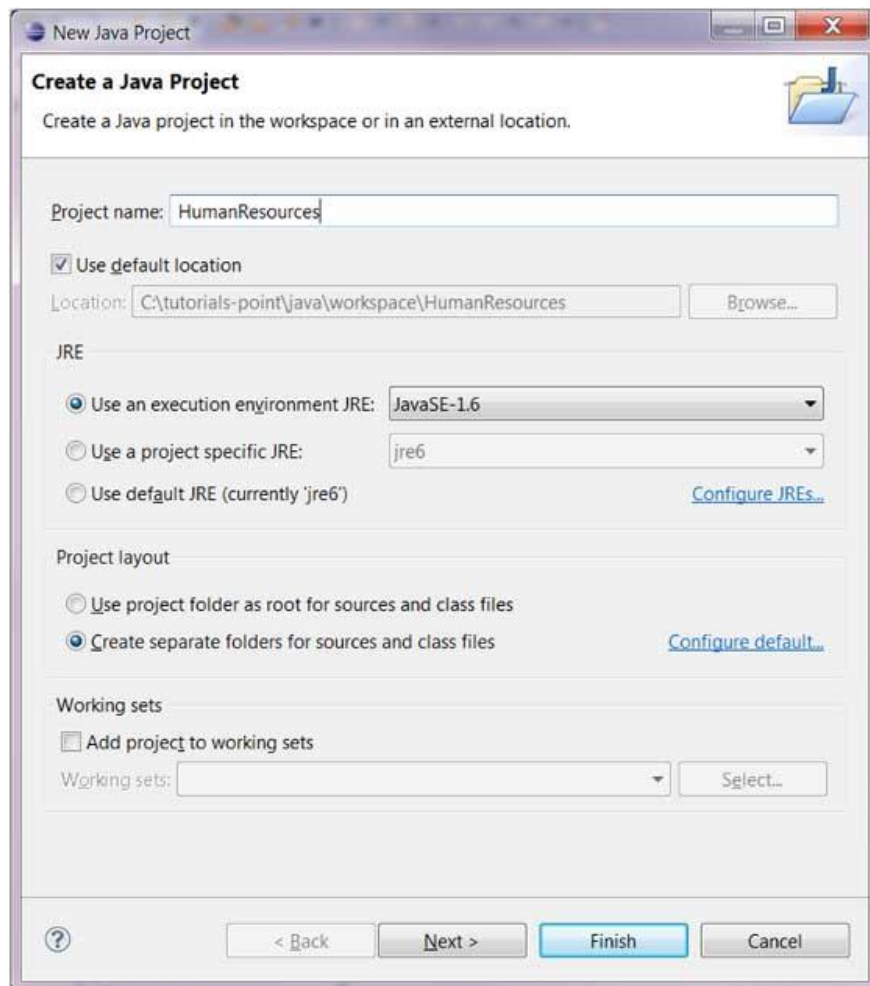
The New Java Project wizard can be used to create a new java project. There are many ways to open this wizard:

- By clicking on the File menu and choosing New → Java Project
- By right clicking anywhere in the Project Explorer and selecting New → Java Project
- By clicking on the New button () in the Tool bar and selecting Java Project

Using the New Java Project wizard

The New Java Project Wizard has two pages. On the first page:

- Enter the Project Name
- Select the Java Runtime Environment (JRE) or leave it at the default
- Select the Project Layout which determines whether there would be a separate folder for the source codes and class files. The recommended option is to create separate folders for sources and class files.

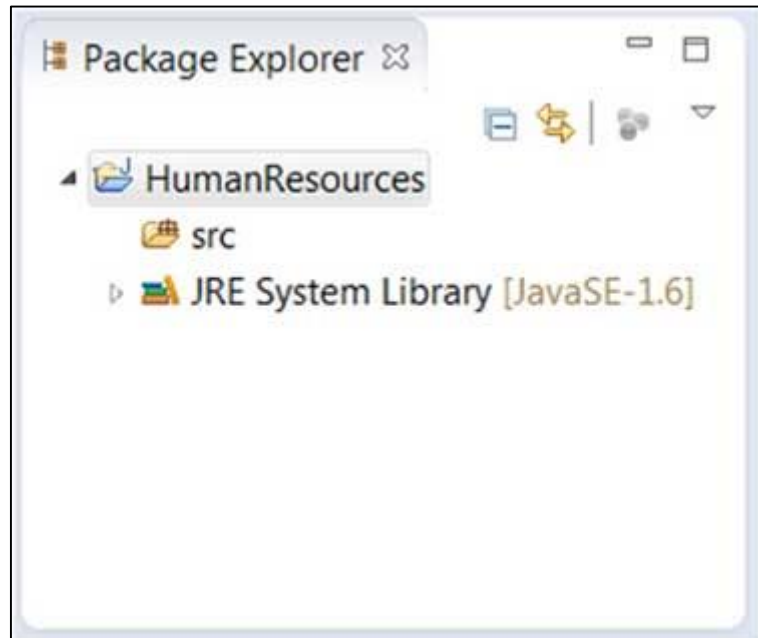


You can click on the Finish button to create the project or click on the Next button to change the java build settings.

On the second page you can change the [Java Build Settings](#) like setting the Project dependency (if there are multiple projects) and adding additional jar files to the build path.

Viewing the Newly Created Project


The package explorer shows the newly created Java project. The icon that represents a Project is decorated with a "J" to show that it is a Java Project. The folder icon is decorated to show that it is a java source folder.



9. CREATE JAVA PACKAGE

Opening the New Java Package wizard

You can use the New Java Package wizard to create a Java package. The Java Package wizard can be opened in different ways:

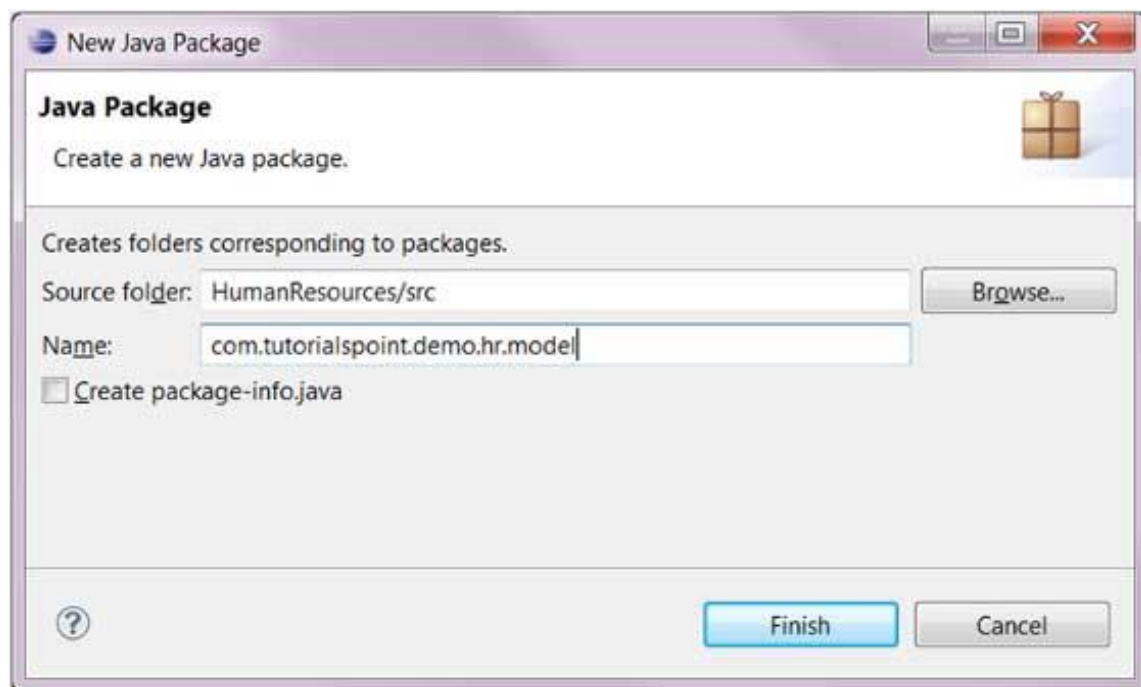
- By clicking on the File menu and selecting New → Package
- By right click in the package explorer and selecting New → Package
- By clicking on the package icon which is in the tool bar()

If you are creating a sub package, before opening the Java Package wizard select the parent package so that name field can have a default value in it.

Using the New Java Package Wizard

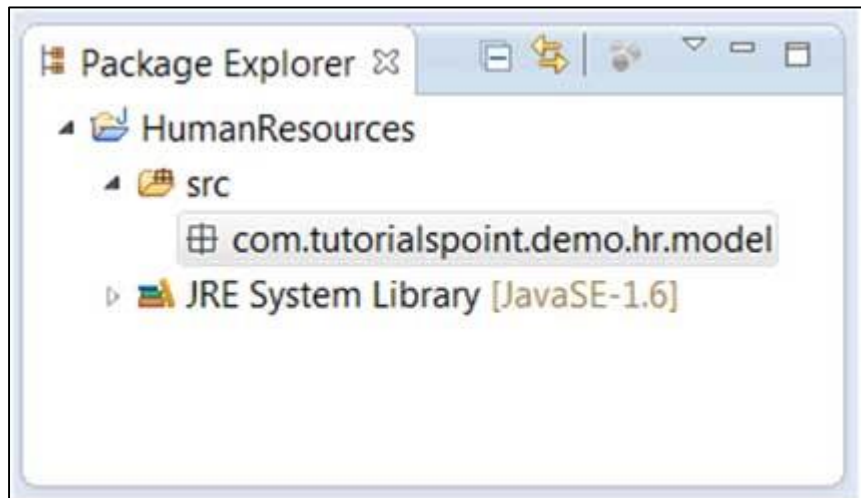
Once the Java Package wizard comes up:

- Enter/confirm the source folder name
- Enter the package name
- Click on the Finish button



Viewing the Newly Created Package



The package explorer will show the newly created package under the source folder.



10. CREATE JAVA CLASS

Opening the New Java Class Wizard

You can use the New Java Class wizard to create a Java class. The Java Class wizard can be invoked in different ways:

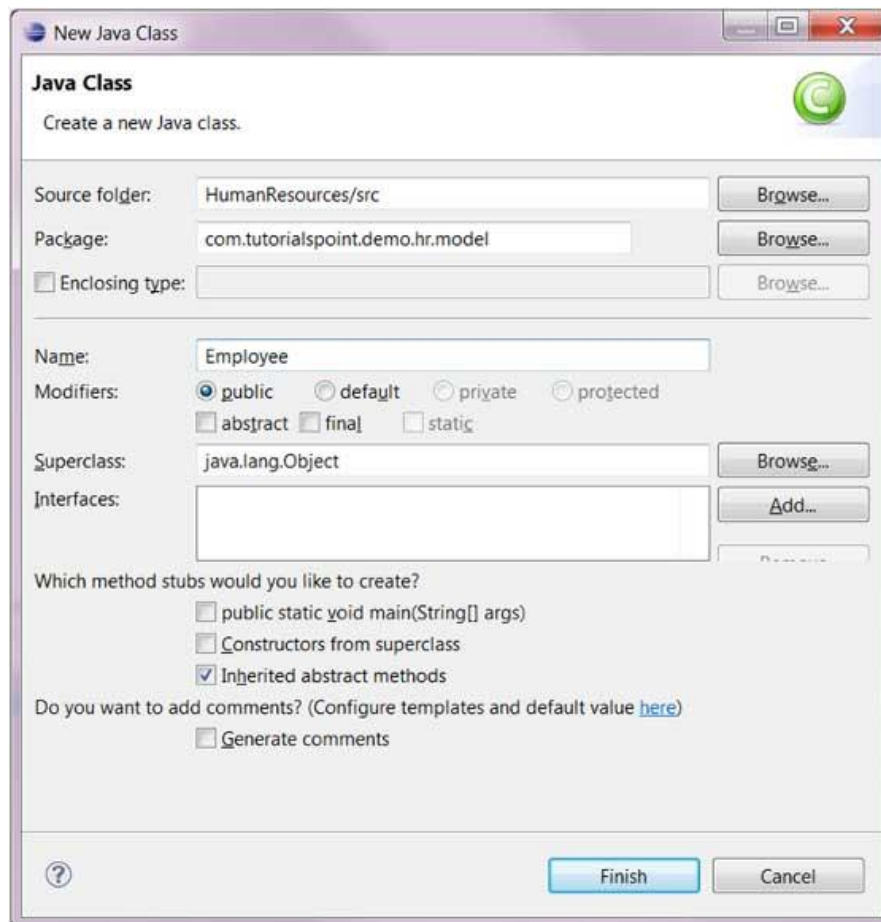
- By clicking on the File menu and selecting New → Class
- By right clicking in the package explorer and selecting New → Class
- By clicking on the class drop down button () and selecting class ()

Before bringing up the New Java Class wizard, if possible, select the package in which the class is to be created so that the wizard can automatically fill in the package name for you.

Using the New Java Class Wizard

Once the java class wizard comes up:

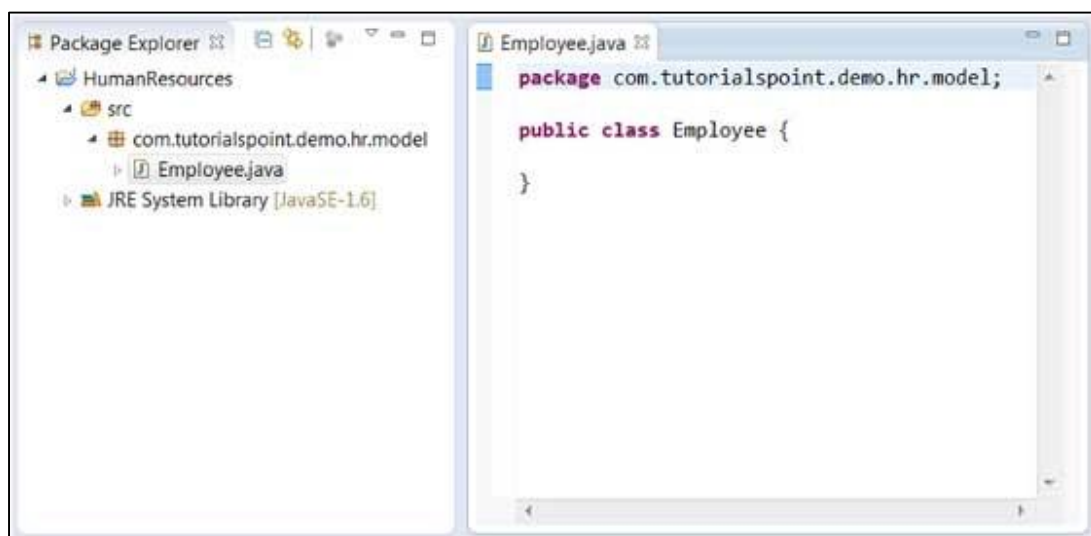
- Ensure the source folder and package are correct
- Enter the class name
- Select the appropriate class modifier
- Enter the super class name or click on the Browse button to search for an existing class
- Click on the Add button to select the interfaces implemented by this class
- Examine and modify the check boxes related to method stubs and comments



- Click the Finish button

Viewing the Newly Created Java class



The newly created class should appear in the Package Explorer view and a java editor instance that allows you to modify the new class. It should appear in the editor area.



11. CREATE JAVA INTERFACE

Opening the New Java Interface Wizard

The New Java Interface wizard can be used to create a new java interface. There are many ways of opening this wizard:

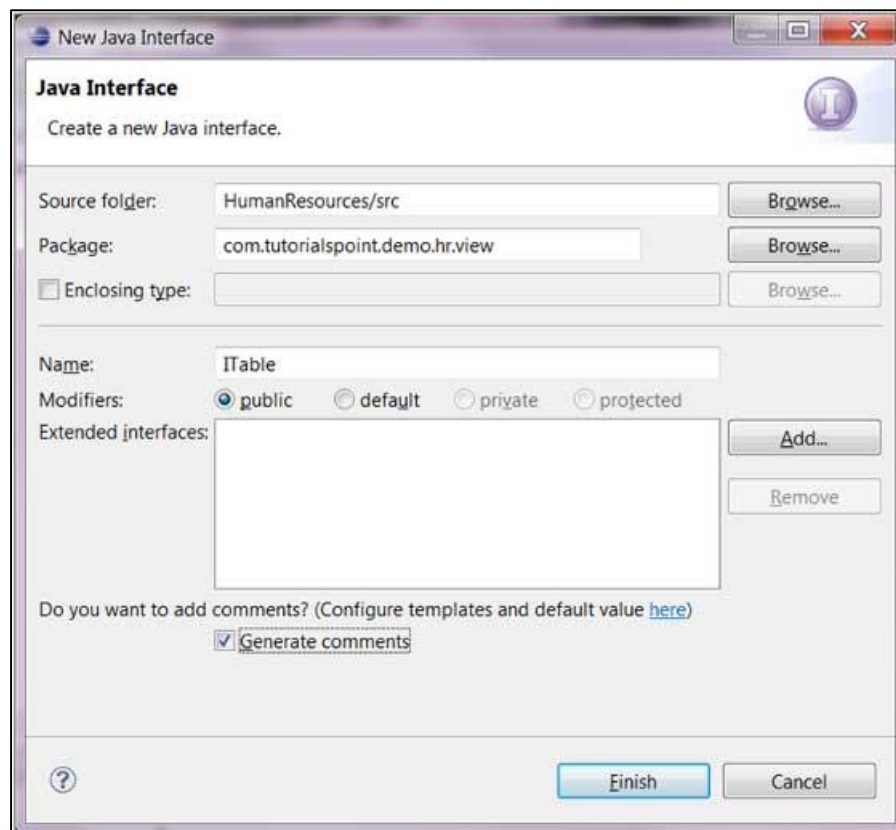
- Clicking on the File menu and selecting New → Interface
- Right clicking in the package explorer and selecting New → Interface
- Clicking on the class drop down button () in the tool bar and selecting Interface ()

Before bringing up the New Java Interface wizard, if possible, select the package in which the interface is to be created so that the wizard can automatically fill in the package name for you.

Using the New Java Interface Wizard

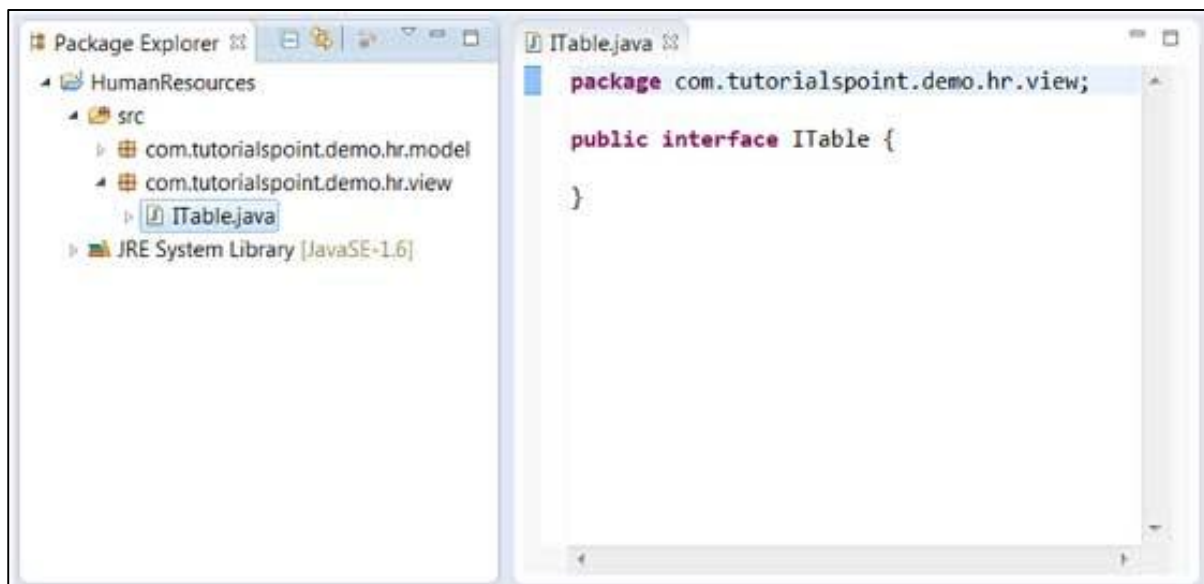
Once the java interface wizard comes up:

- Ensure the source folder and package are correct
- Enter the interface name
- Click on the **Add** button to select the extended interfaces
- Select the Generate comments check box if you like comments to be generated
- Click on the Finish button



Viewing the Newly Created Java Interface


The newly created interface should appear in the Package Explorer view and a java editor instance that allows you to modify the new interface should appear in the editor area.



12. CREATE XML FILE

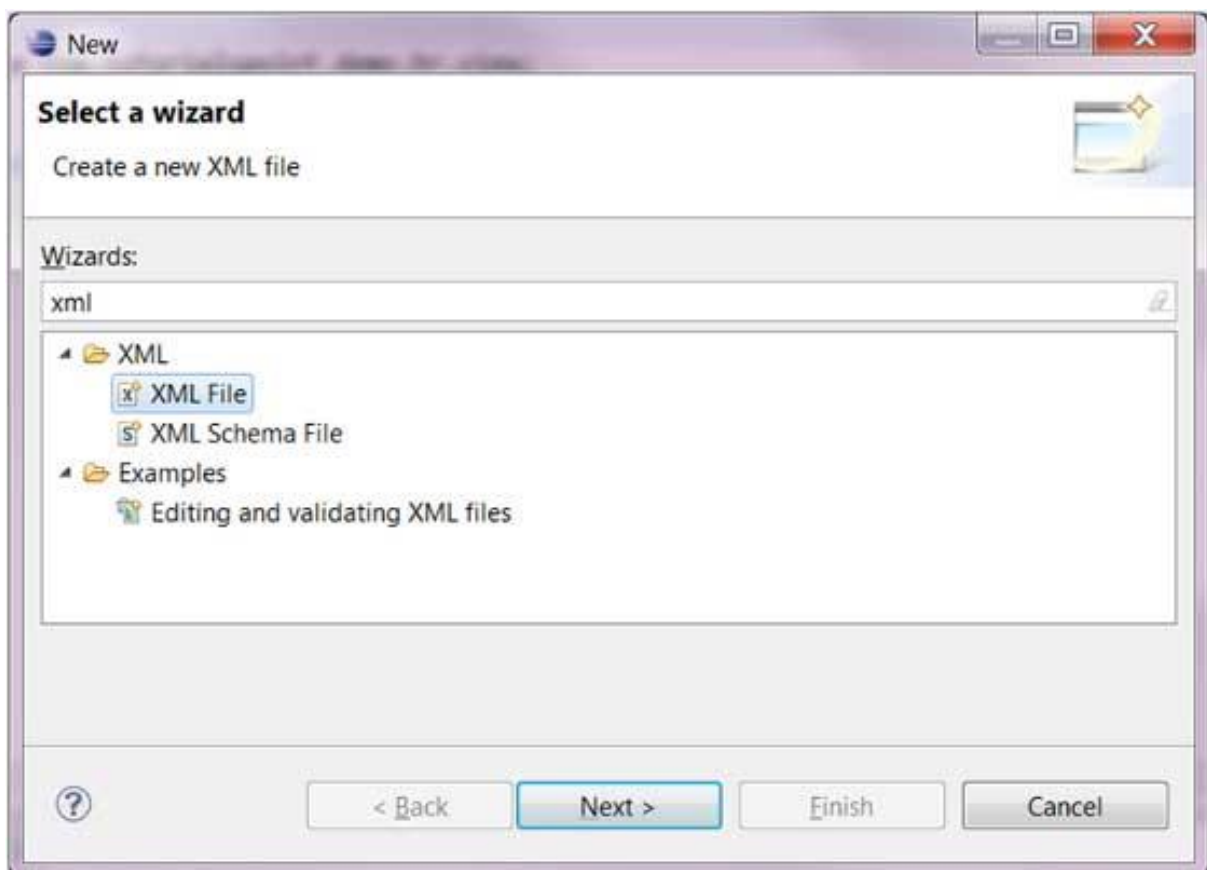
Opening the New XML File wizard

You can use the New XML file wizard to create an XML file. To invoke this wizard:

- Click on the File menu and select New → Other or
- Click on the new drop down box () select other or
- Press ctrl + N

Once the Select a wizard dialog box comes up:


- In the filter text box enter xml, this should show only the wizards related to xml
- Expand XML category and select XML File



- Click on Next to bring up the New XML File wizard

Note

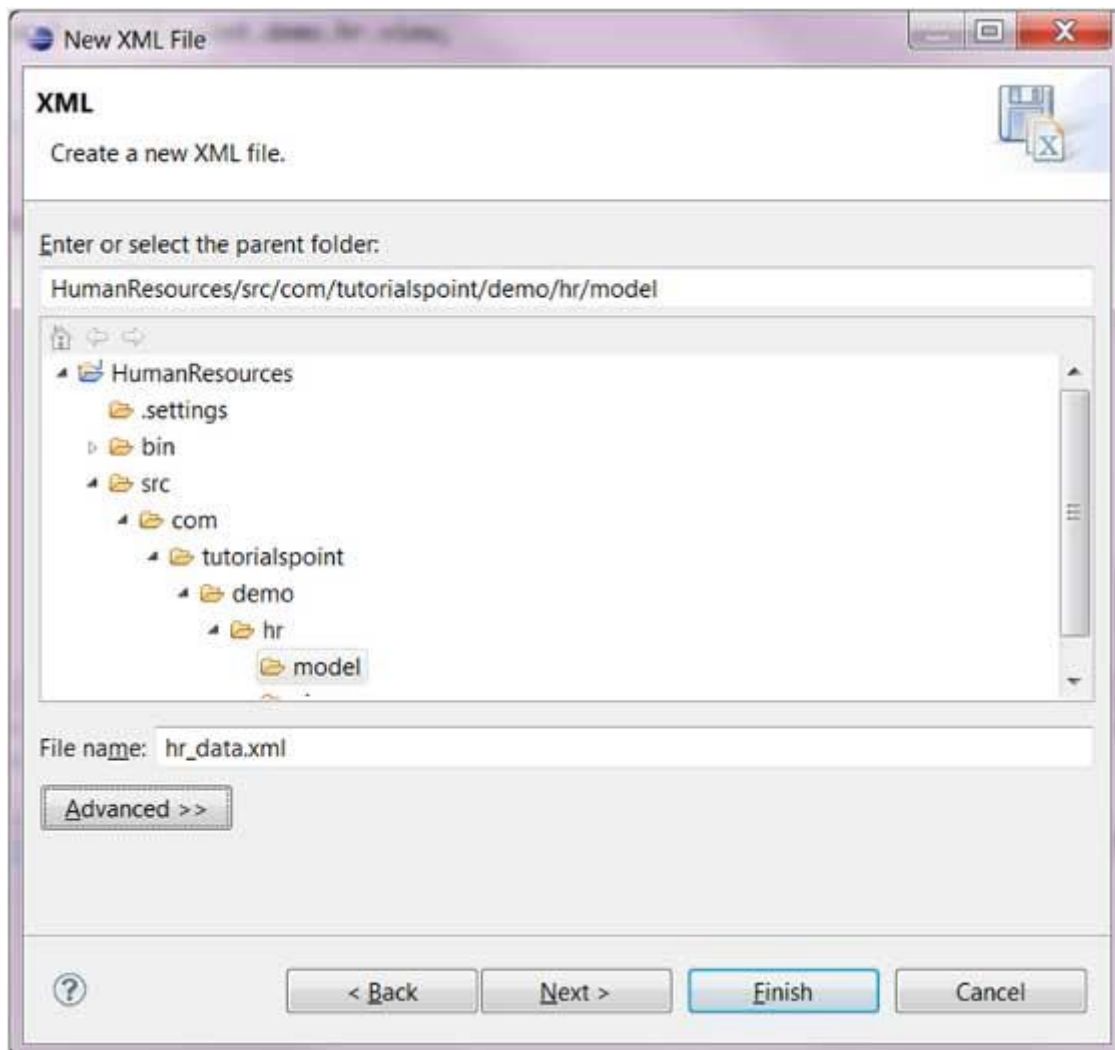
If the XML perspective is open the New XML wizard can be invoked by:

- Clicking on the File menu and selecting New → XML File or
- Clicking on the XML File button ( XML File...) on the tool bar

Using the New XML File wizard

Once the New XML File wizard comes up:

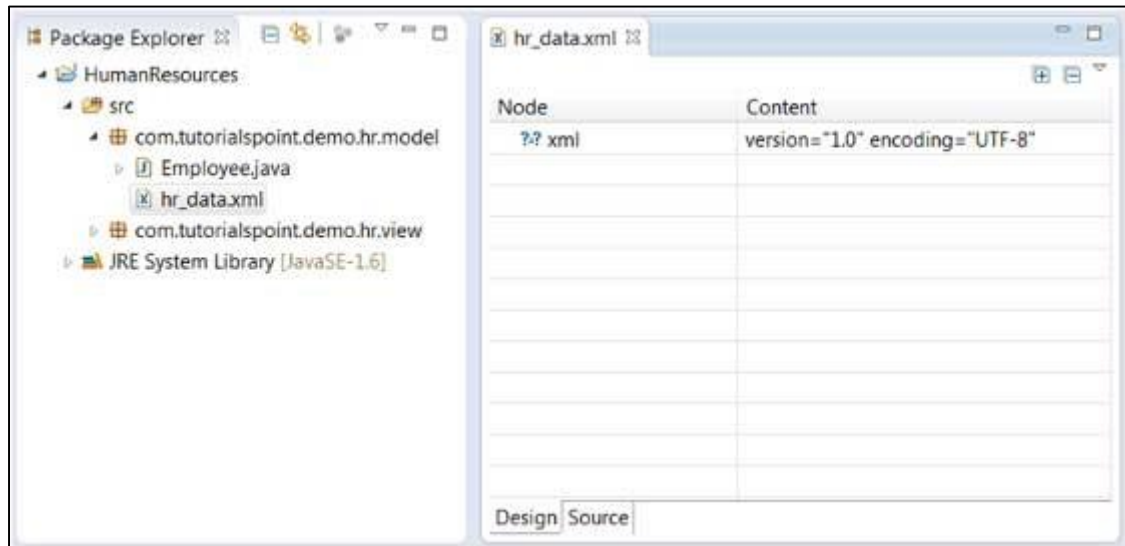
- Enter or select the parent folder
- Enter the name of the xml file



- Click on the **Next** button to base the xml file on DTD, XML Schema or XML template else click on **Finish**

Viewing the Newly Created XML File

The newly created XML file should appear in the Package Explorer view and an XML editor instance that allows you to modify the newly created XML file should appear in the editor area.



The XML editor allows you to edit an XML file using either the Design view or Source view.

13. JAVA BUILD PATH

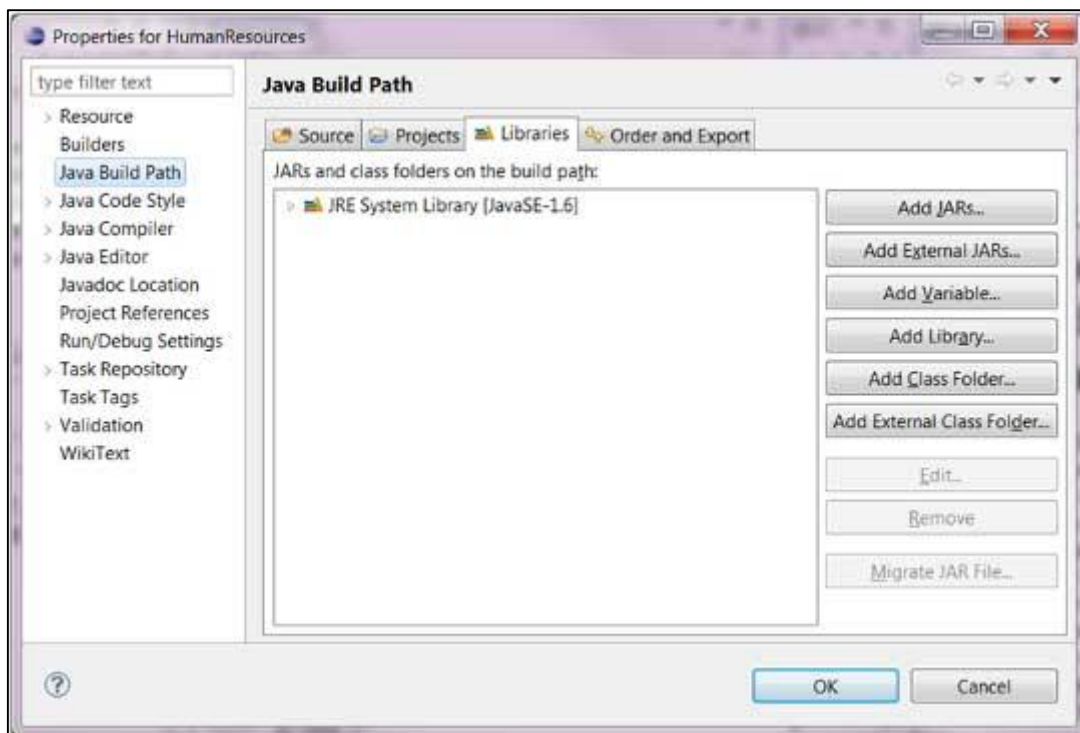
Setting the Java Build Path

The Java build path is used while compiling a Java project to discover dependent classes. It is made up of the following items:

- Code in the source folders
- Jars and classes folder associated with the project
- Classes and libraries exported by projects referenced by this project

The java build path can be seen and modified by using the Java Build Path page of the Java Project properties dialog.

To bring up the Java Project properties dialog box, right click on a Java Project in the Package Explorer view and select the Properties menu item. On the left hand side tree select **Java Build Path**.

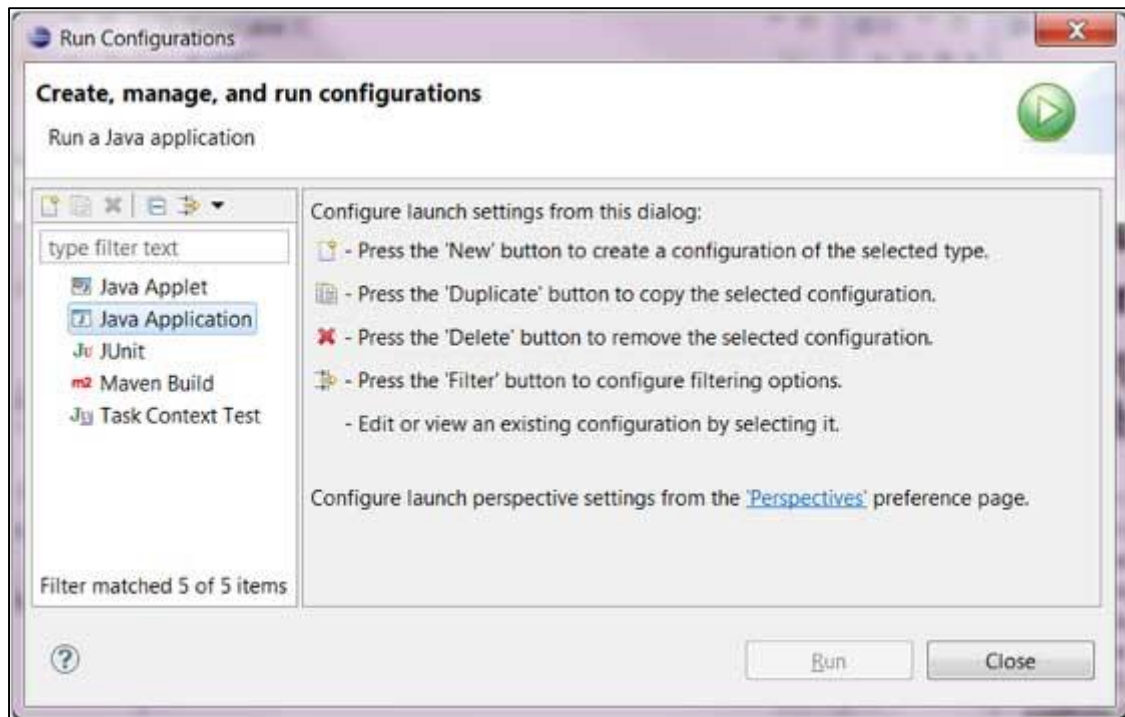


A common requirement seen while developing java applications is to add existing jars to the java build path. This can be accomplished using the Libraries tab. In the Libraries tab, just click on **Add JARs** if the jar is already in the Eclipse workspace or click on Add External JARs if the jar is elsewhere in the file system.

14. RUN CONFIGURATION

Creating and Using a Run Configuration

The Run Configurations dialog allows you create multiple run configurations. Each run configuration can start an application. The Run Configuration dialog can be invoked by selecting the Run Configurations menu item from the Run menu.

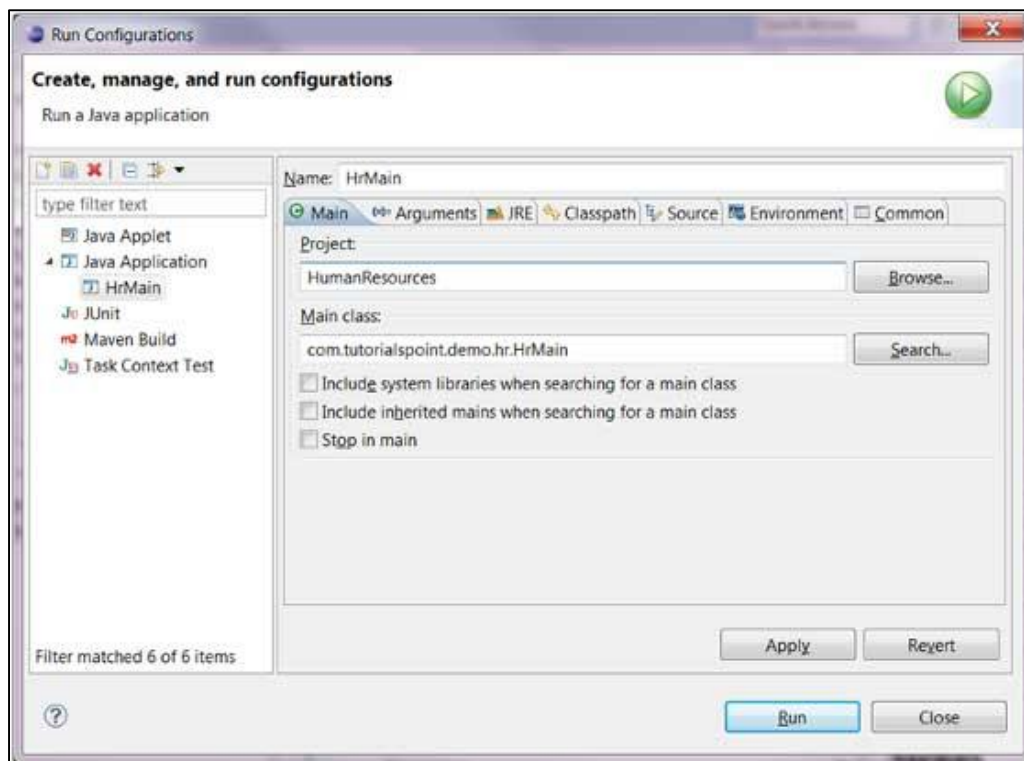


To create a run configuration for a Java application select "Java Application" from the list on the left hand side and click on the New button. In the dialog box that comes up in the main tab specify:

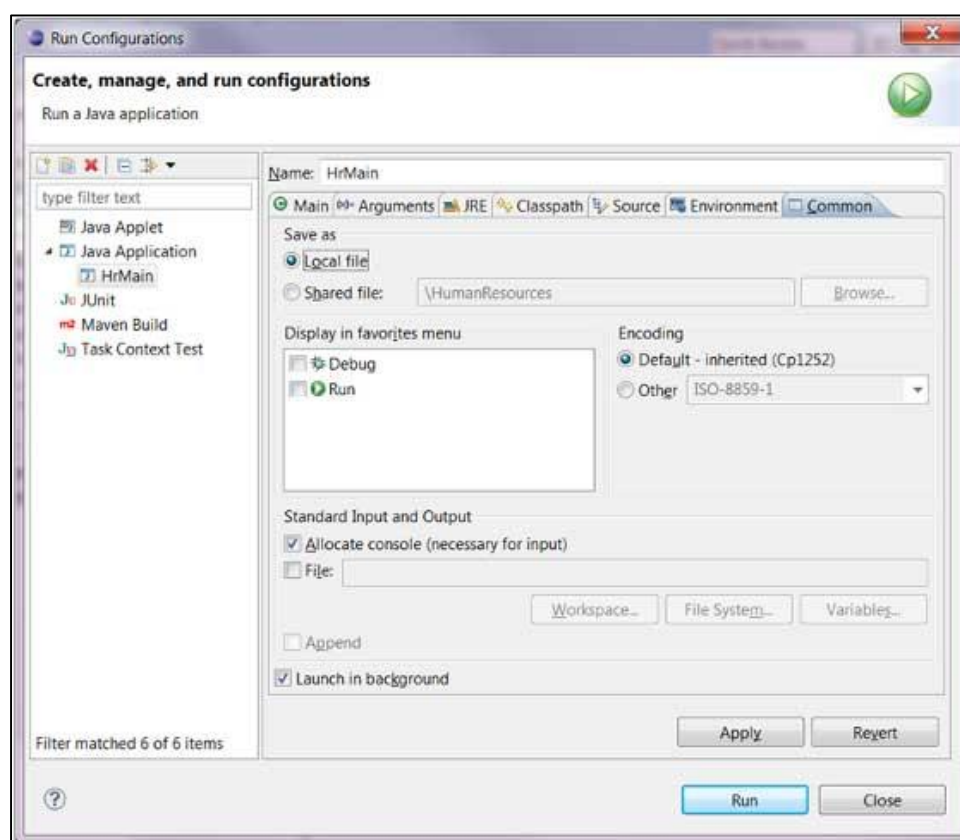
- A name for the run configuration
- The name of a Project
- The name of the main class

In the arguments tab specify:

- Zero or more program arguments
- Zero or more Virtual Machine arguments



The Commons tab provides common options such as the ability to allocate a console for standard input and output.



To save the run configuration click on the Apply button and to launch the application click on the Run button.

15. RUNNING A PROGRAM

Running a Java Program

The quickest way to run a Java program is by using the Package Explorer view.

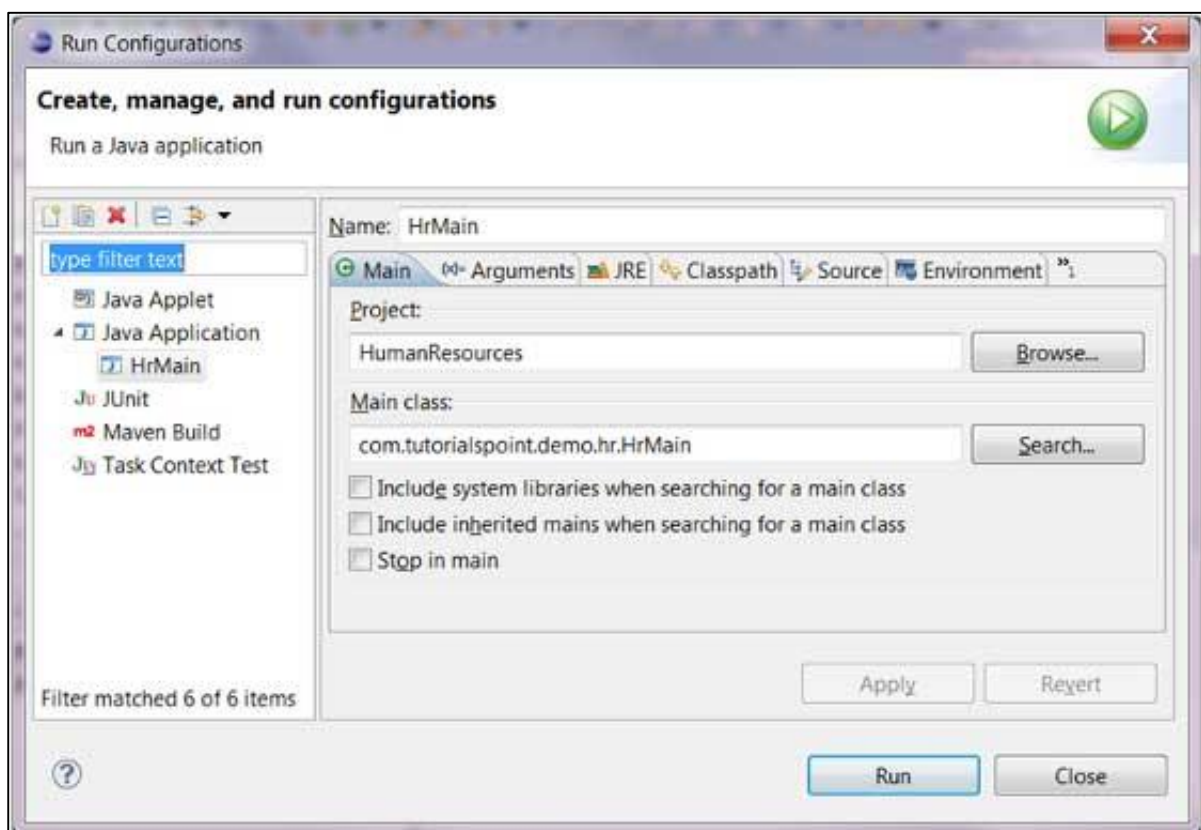
In the Package Explorer view:

- Right click on the java class that contains the main method
- Select Run As → Java Application

The same action can be performed using the Package Explorer view by selecting the class that contains the main method and clicking Alt + Shift + X, J

Either actions mentioned above create a new [Run Configuration](#) and use it to start the Java application.

If a Run configuration has already been created you can use it to start the Java application by selecting Run Configurations from the Run menu, clicking on the name of the run configuration and then clicking on the Run button.



The **Run** item on the **Run** menu can be used to restart the java application that was previously started.



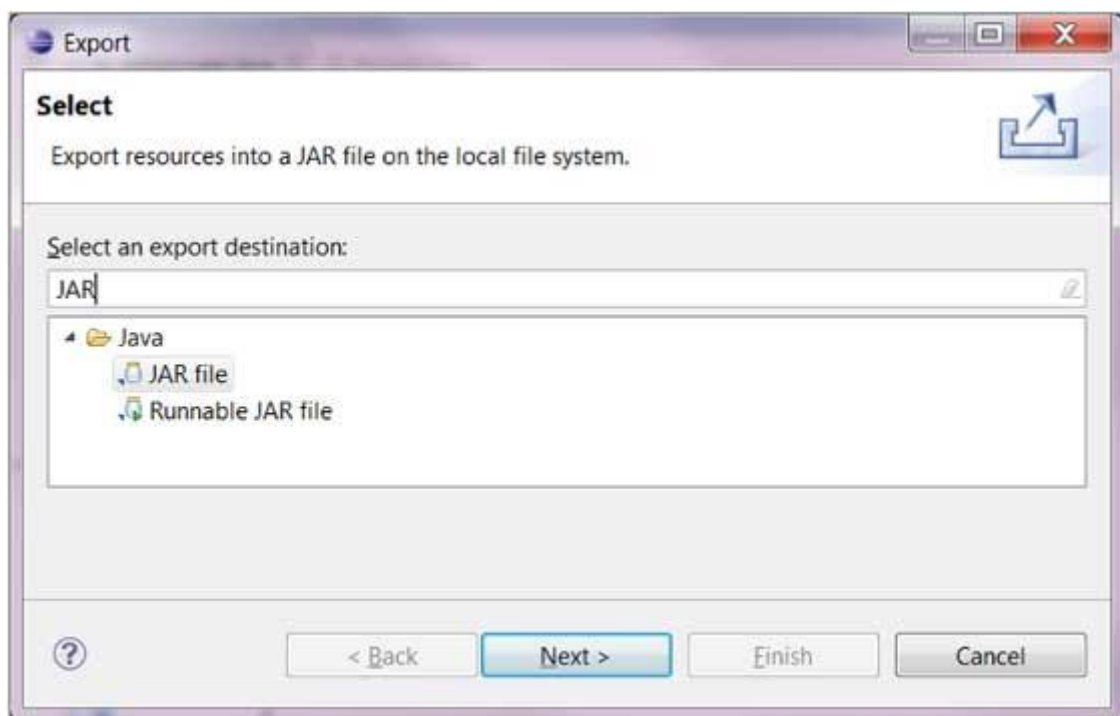
The shortcut key to launch the previously launched Java application is Ctrl + F11.

16. CREATE JAR FILES

Opening the Jar File wizard

The Jar File wizard can be used to export the content of a project into a jar file. To bring up the Jar File wizard:

- In the Package Explorer select the items that you want to export. If you want to export all the classes and resources in the project just select the project.
- Click on the **File** menu and select **Export**.
- In the filter text box of the first page of the export wizard type in "JAR".

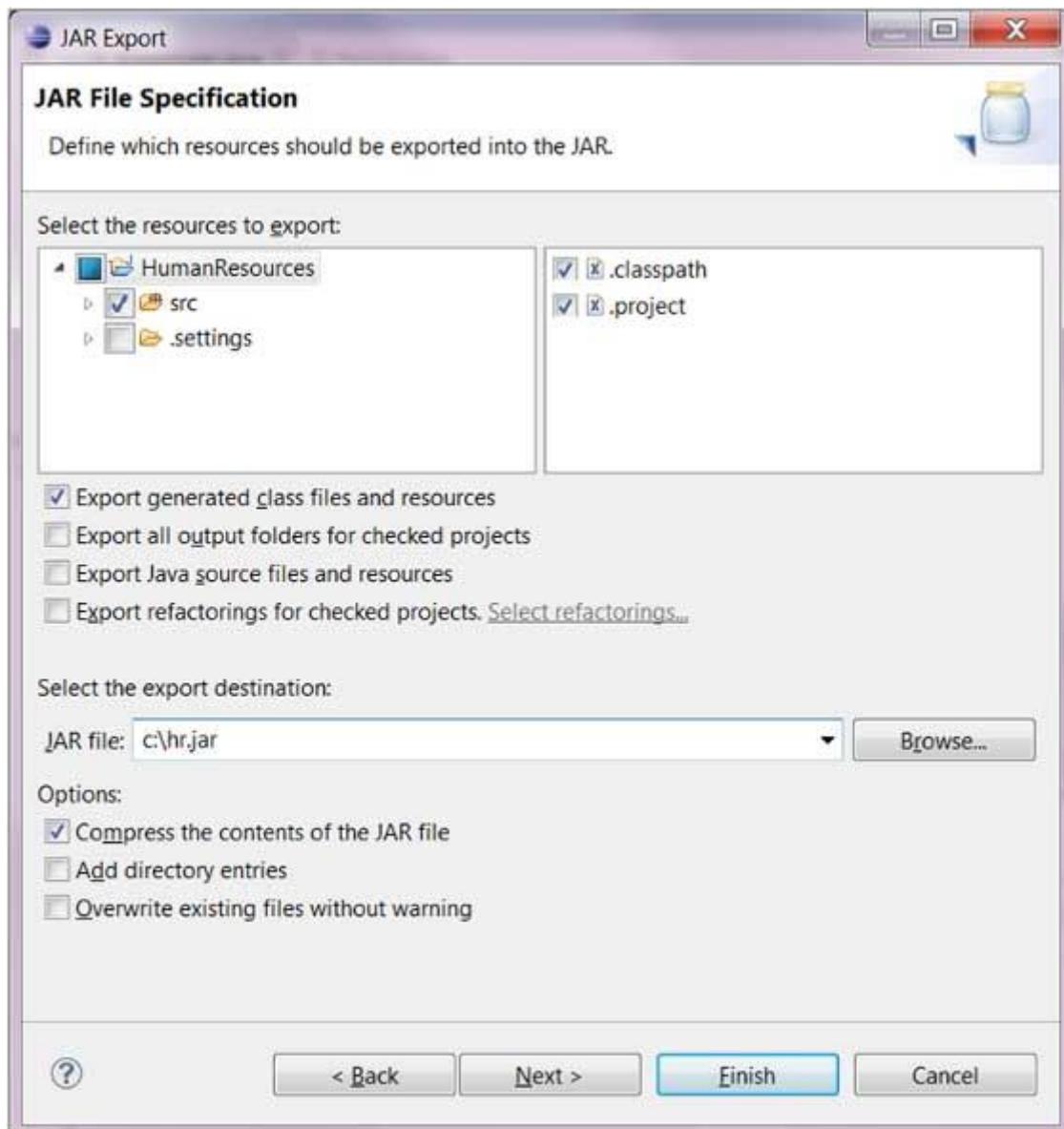


- Under the Java category select **JAR file**
- Click on **Next**

Using the Jar File wizard

In the JAR File Specification page:

- Enter the JAR file name and folder
- The default is to export only the classes. To export the source code also, click on the "Export Java source files and resources" check box.



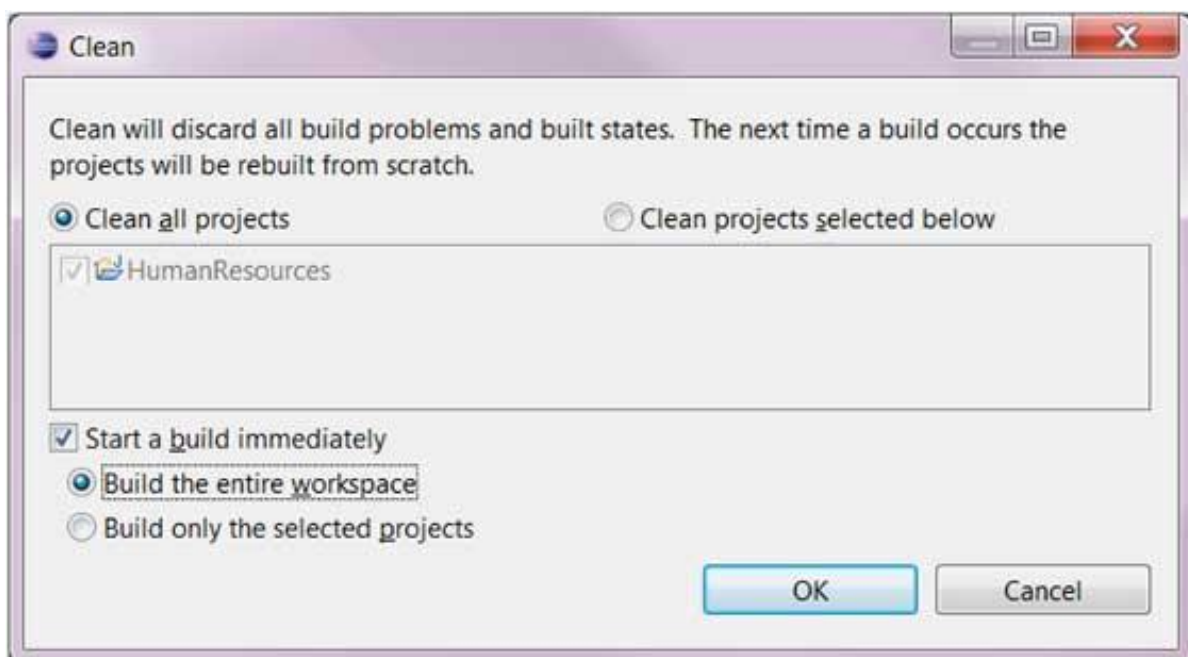
- Click on **Next** to change the JAR packaging options
- Click on **Next** to change the JAR Manifest specification
- Click on **Finish**

17. CLOSE PROJECT

Why Close a Project?

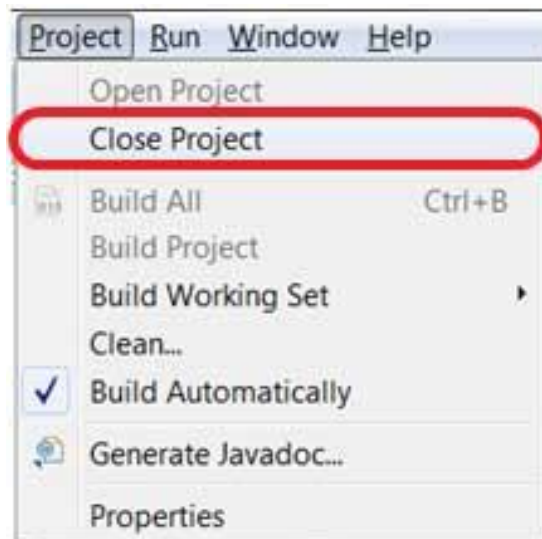
An eclipse workspace can contain any number of projects. A project can be either in the open state or closed state. Open projects:

- Consume memory
- Take up build time especially when the Clean All Projects (Project → Clean all projects) with the **Start a build immediately** option is used.



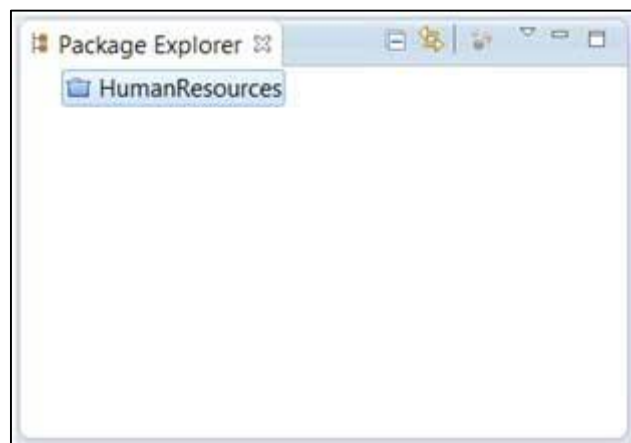
How to Close a Project?

If a project is not under active development it can be closed. To close a project, from the Project select the Close Project menu item.



Closed Project in Package Explorer

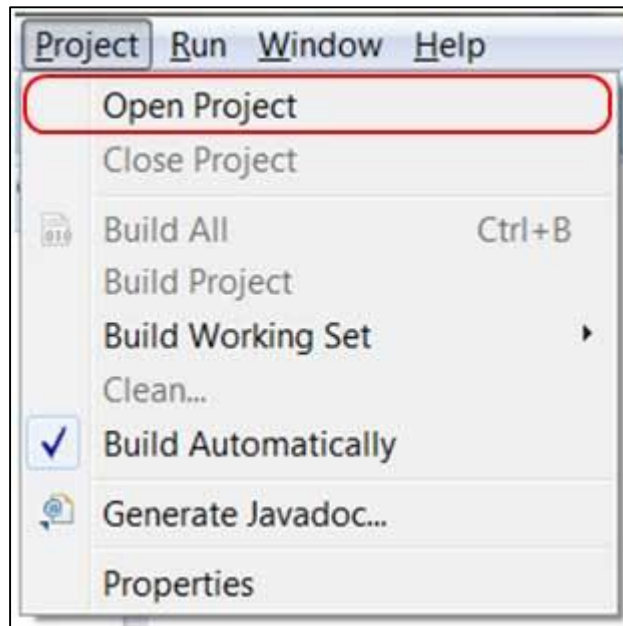
A closed project is visible in the Package Explorer view but its contents cannot be edited using the Eclipse user interface. Also, an open project cannot have dependency on a closed project. The Package Explorer view uses a different icon to represent a closed project.



18. REOPEN PROJECT

Reopening a Closed Project

To reopen a closed project, in the Package Explorer view, select the closed project and click on the Project menu and select Open Project.



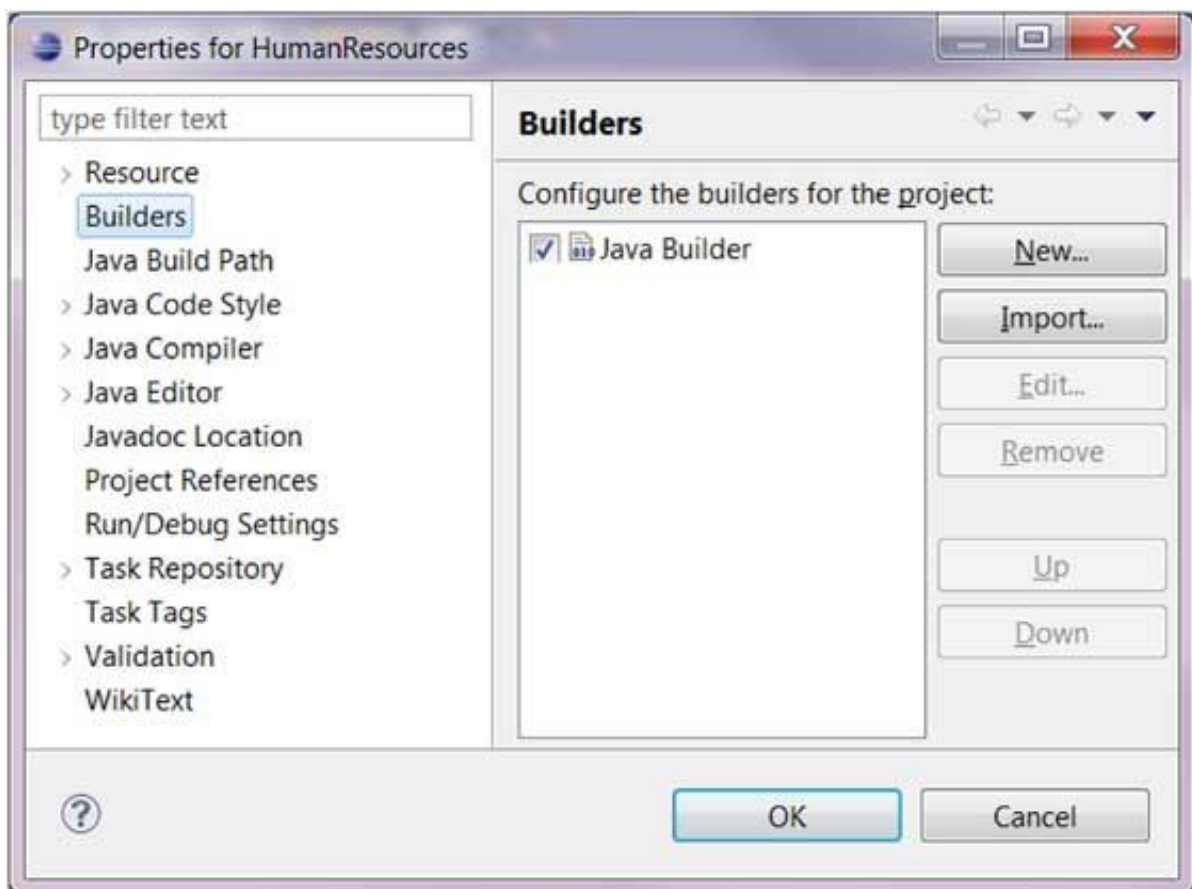
Once the project is open its content can be edited using the Eclipse user interface.

19. BUILD PROJECT

Building a Java Project

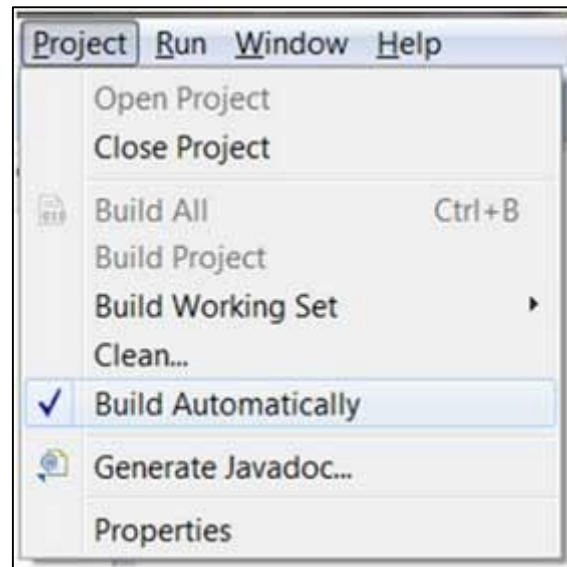
A project can have zero or more builders associated with it. A java project is associated with a java builder. To see the builders associated with a project:

- In the Package Explorer view right click on the project and select Properties
- In the left hand side tree click Builders



It's the java builder that distinguishes a Java project from other types of projects. By clicking on the New button you can associate the Ant builder with a java project. The java builder is responsible for compiling the java source code and generating classes.

The java builder is notified of changes to the resources in a workspace and can automatically compile java code. To disable automatic compilation, deselect the Build Automatically option from the Project menu.



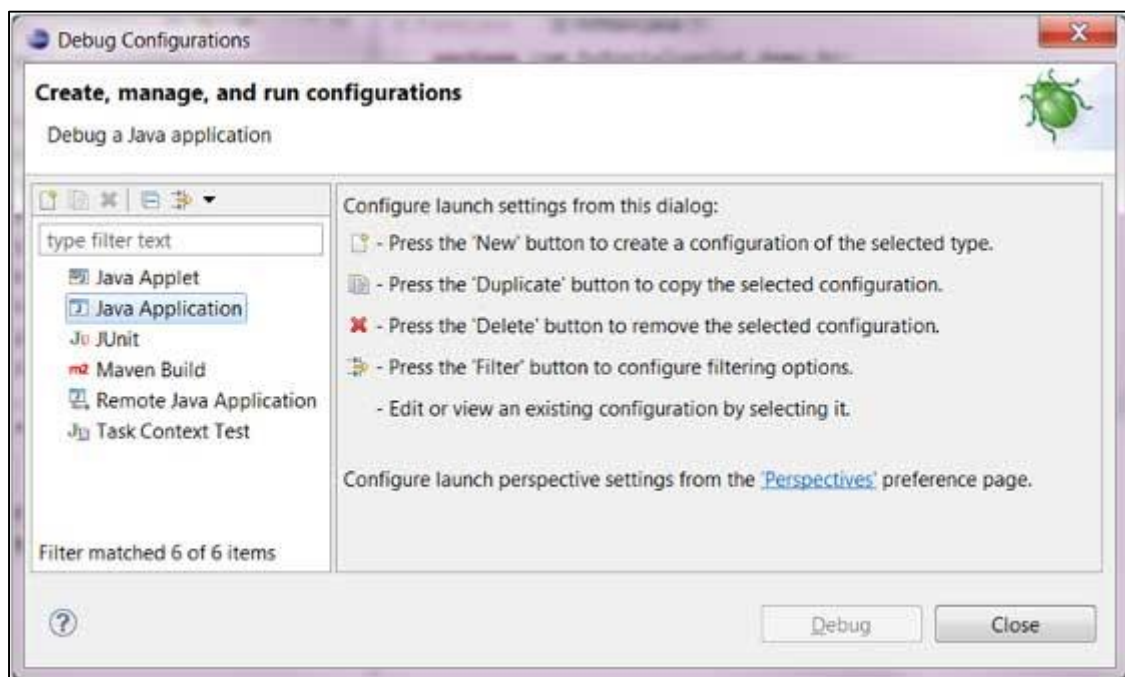
If automatic compilation is disabled then you can explicitly build a project by selecting the Build Project menu item on the Project menu. The Build Project menu item is disabled if the Build Automatically menu item is selected.

20. DEBUG CONFIGURATION

Creating and Using a Debug Configuration

An eclipse debug configuration is similar to a run configuration but it used to start an application in the debug mode. Because the application is started in the debug mode the users are prompted to switch to the debug perspective. The debug perspective offers a number of views that are suitable for debugging applications.

The Debug Configuration dialog can be invoked by selecting the Debug Configurations menu item from the Run menu.

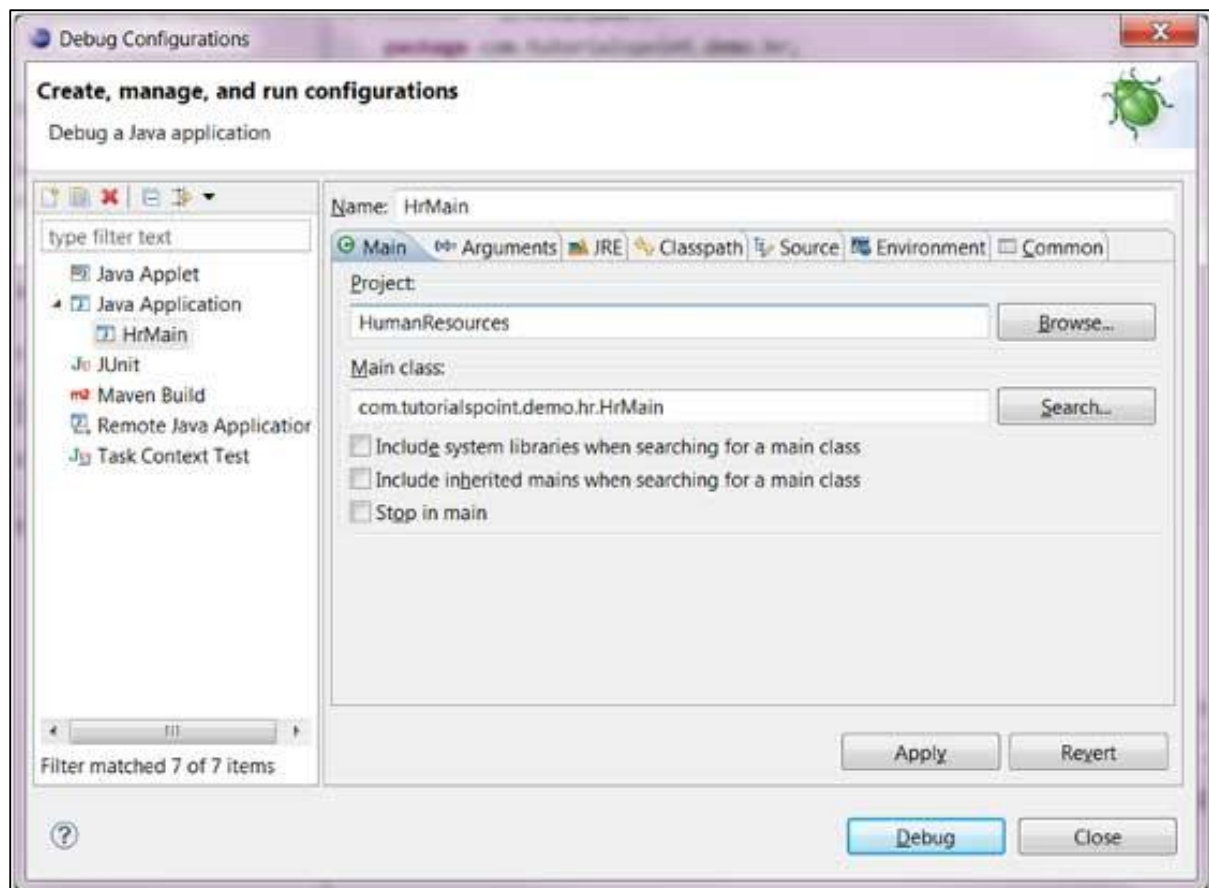


To create a debug configuration for a Java application, select "Java Application" from the list on the left hand side and click on the New button. In the dialog box that comes up in the main tab specify:

- A name for the debug configuration
- The name of a Project
- The name of a main class

In the arguments tab, specify:

- Zero or more program arguments
- Zero or more Virtual Machine arguments



To save the run configuration, click on the Apply button and to launch the application in the debug mode click on the Debug button.

21. DEBUGGING A PROGRAM

Debugging a Java Program

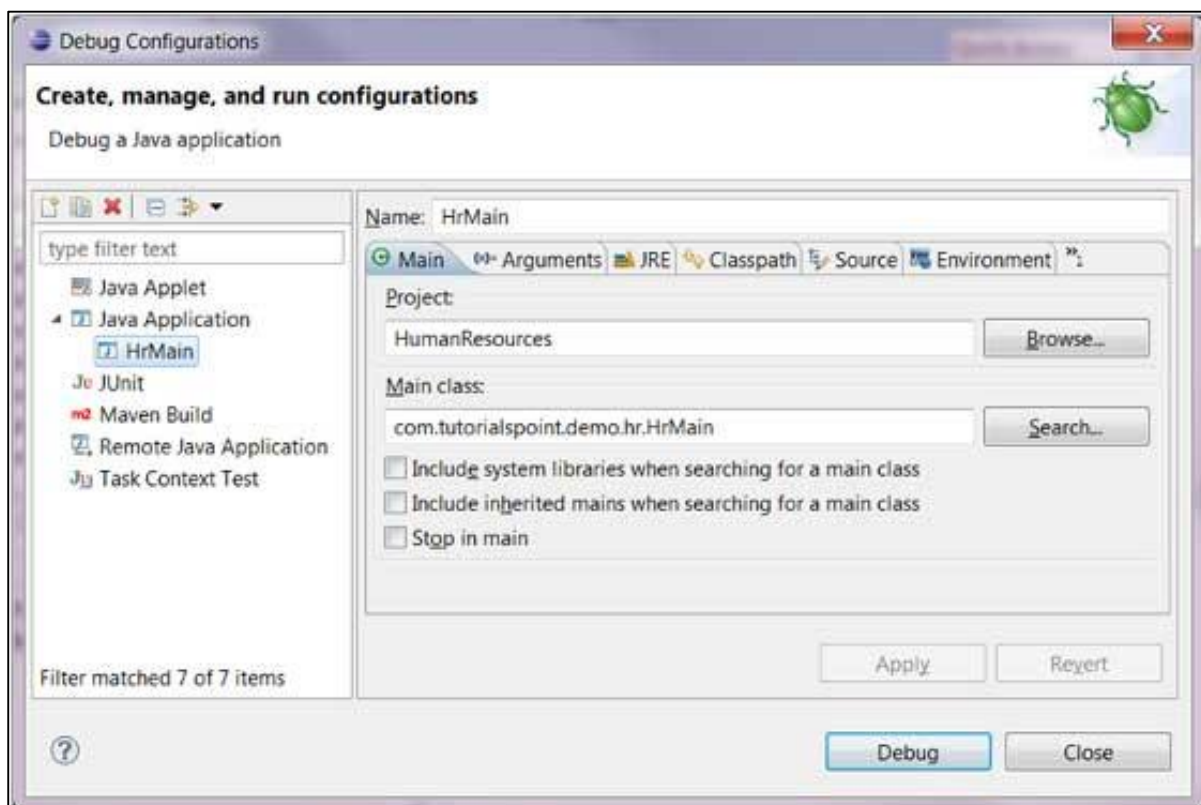
The quickest way to debug a Java program is to using the Package Explorer view. In the Package Explorer view:

- Right click on the java class that contains the main method
- Select Debug As → Java Application

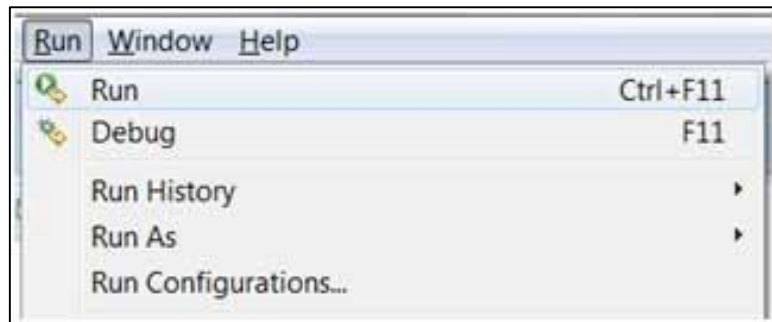
The same action can be performed using the Package Explorer by selecting the class that contains the main method and clicking Alt + Shift + D, J

Either actions mentioned above create a new [Debug Configuration](#) and use it to start the Java application.

If a Debug configuration has already been created you can use it to start the Java application by selecting Debug Configurations from the Run menu, clicking on the name of the debug configuration and then clicking on the Debug button.

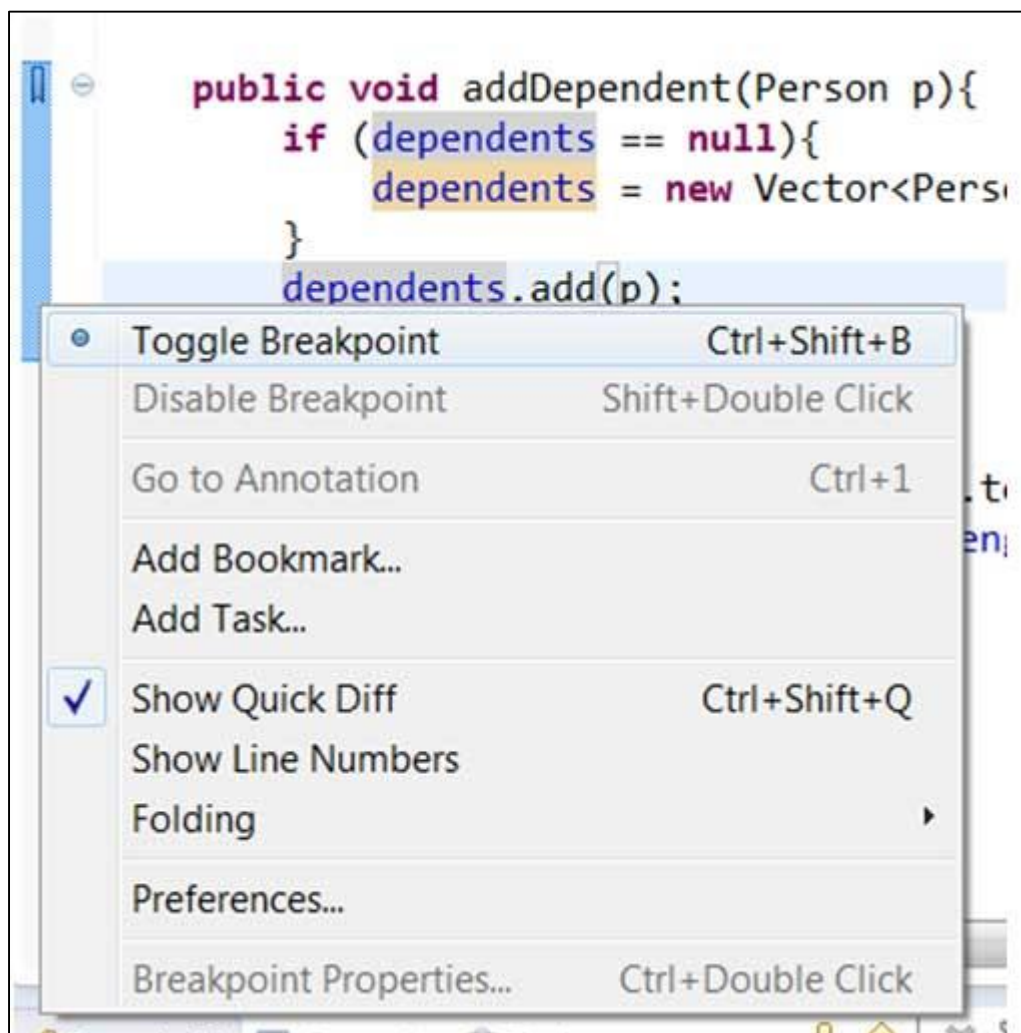


The Debug menu item on the Run menu can be used to restart the java application that was previously started in the debug mode.



The shortcut key to launch the previously launched Java application in the debug mode is F11. When a java program is started in the debug mode, users are prompted to switch to the debug perspective. The debug perspective offers additional views that can be used to troubleshoot an application.

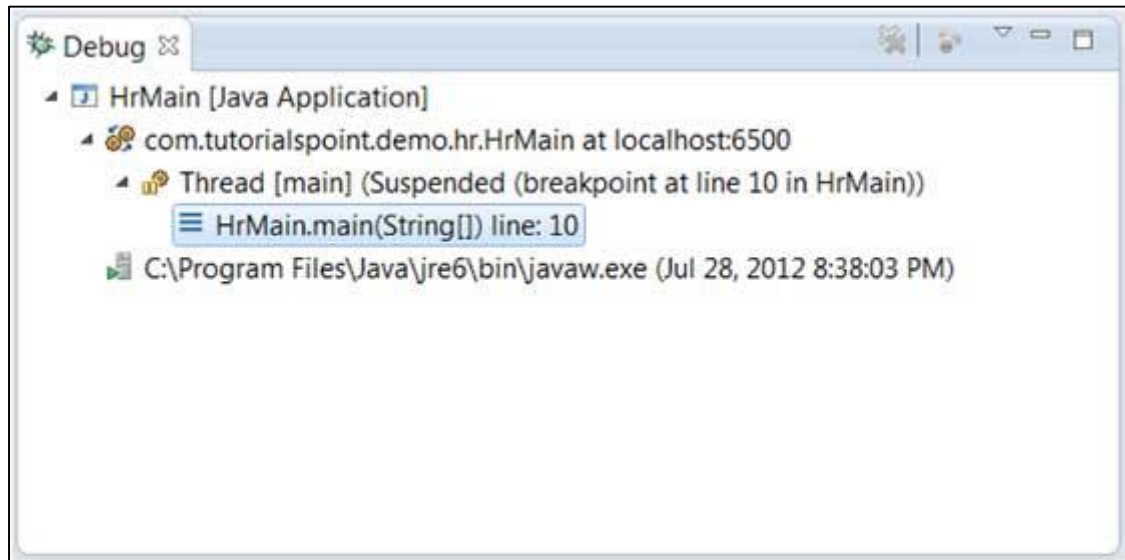
The java editor allows users to place break points in the java code. To set a break point, in the editor area right click on the marker bar and select Toggle Breakpoint.



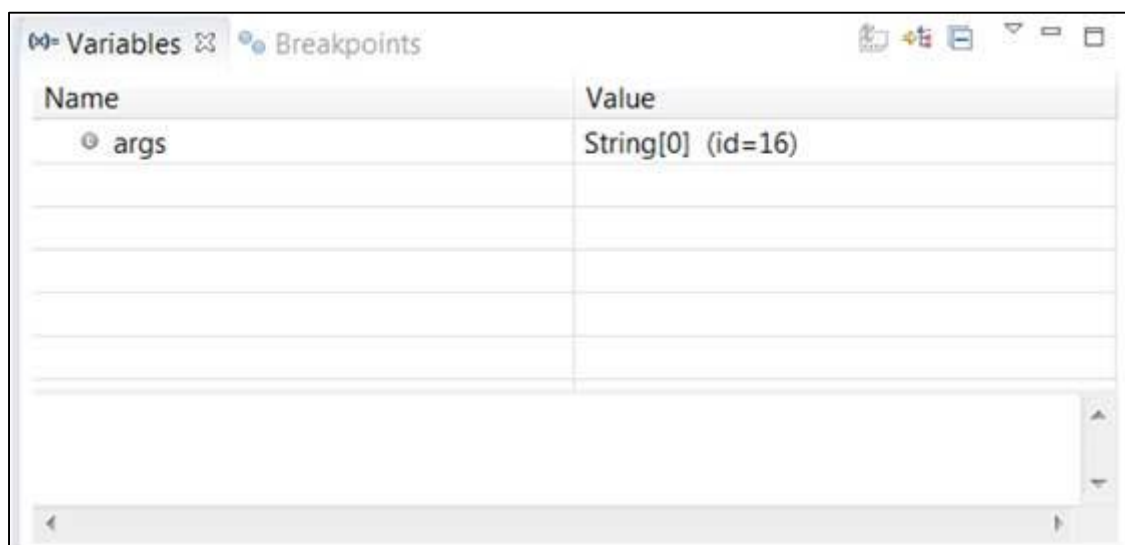
Breakpoints are shown on the marker bar. They are also visible in the Breakpoints View.

When the execution of code reaches a break point the JVM suspends the execution of the program. At this point, you can inspect the memory and execute the program in a controlled manner.

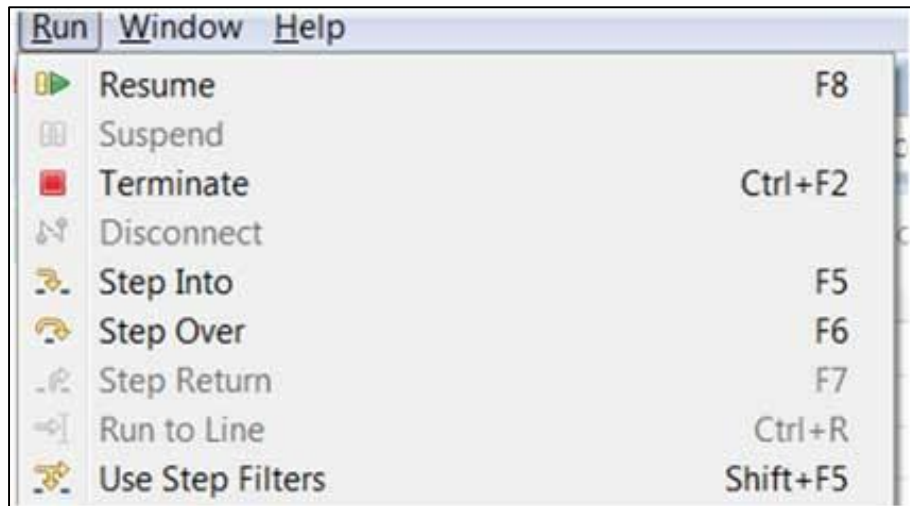
When the execution of a program is paused, the debug view can be used to inspect the call stack.



The variables view can be used to inspect the value of variables.



The run menu has menu items that allow you to Resume execution, step over a line of code, step into a function etc.



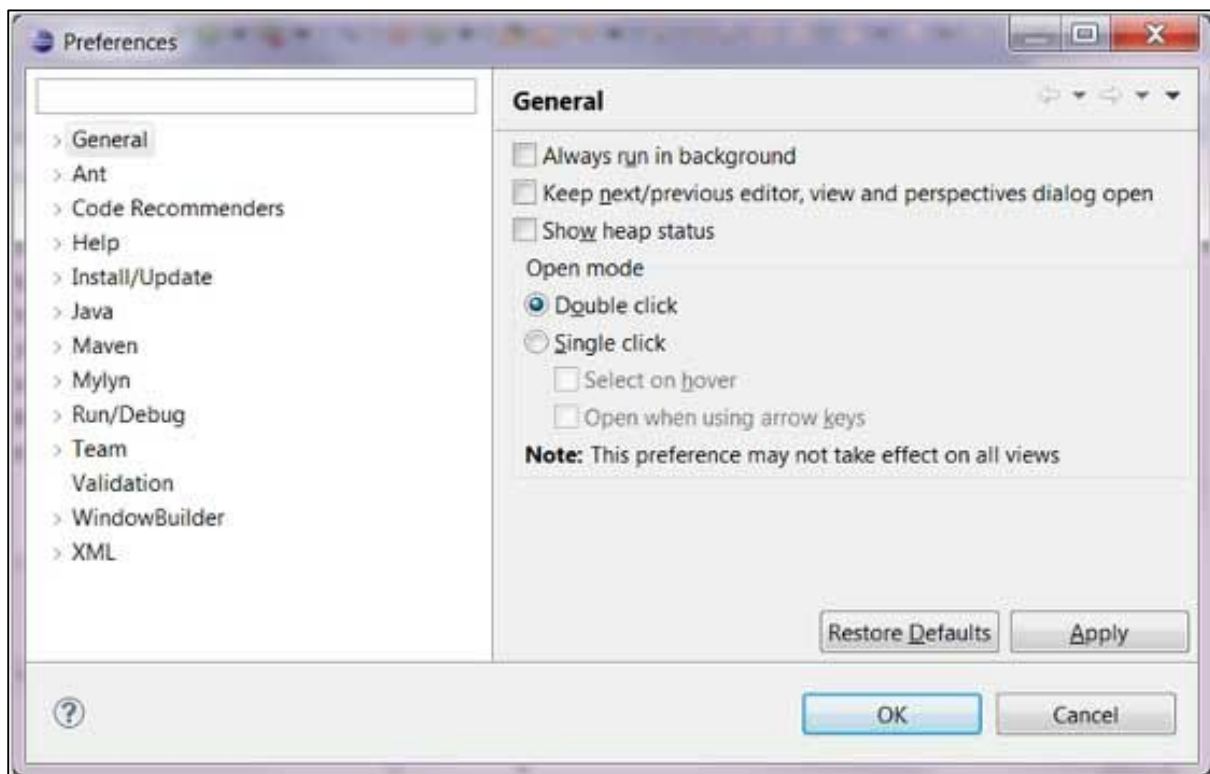
The picture given above shows the shortcut keys associated with actions such as Resume, Step Into and Step Over, etc.

22. PREFERENCES

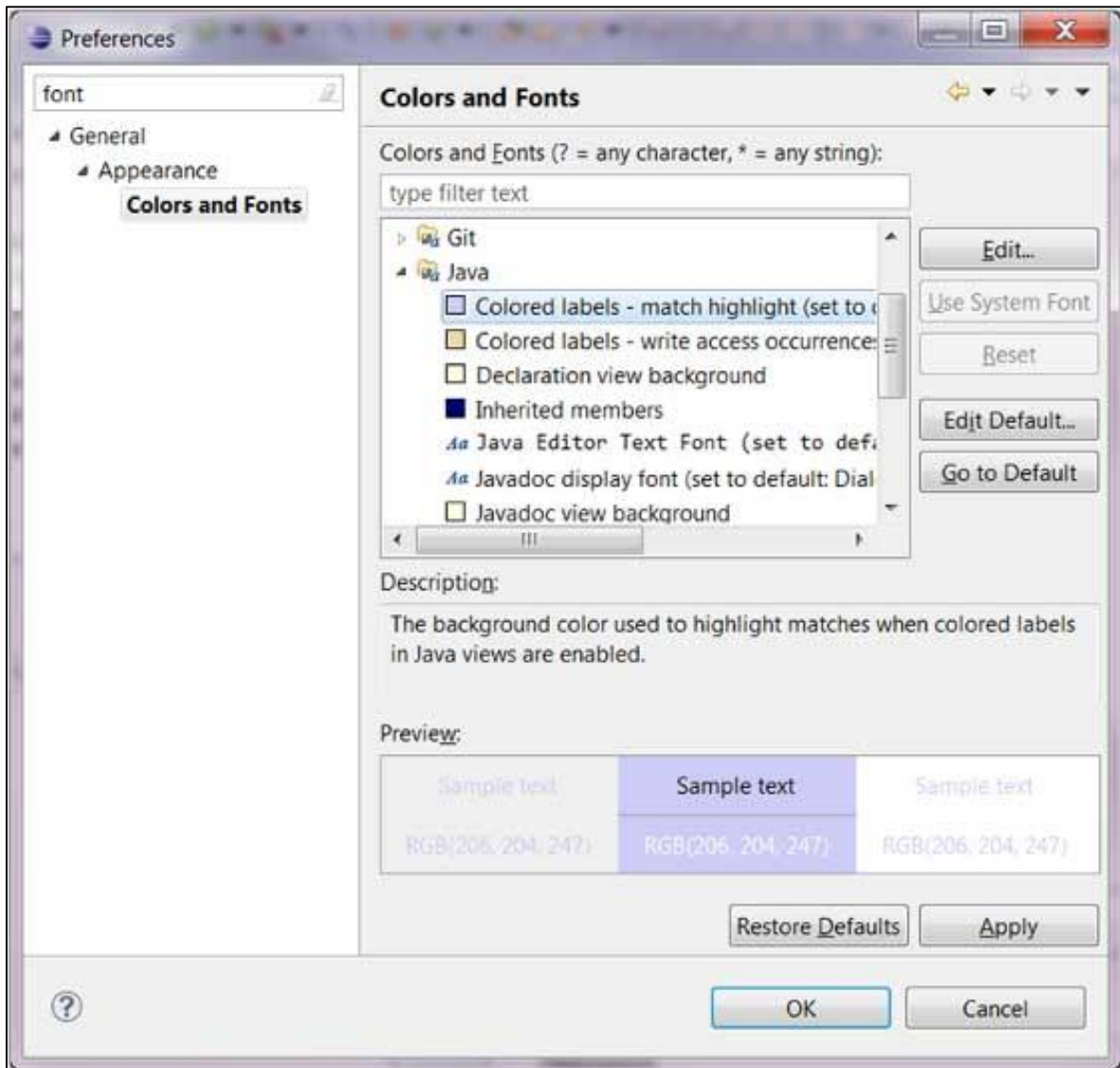
Setting Preferences

The Preferences dialog allows Eclipse users to manage their preferences. This dialog box is managed by the framework but any plug-in can add multiple pages to the dialog box.

To invoke this dialog, click on the Window menu and select the Preferences menu item. The preference pages are organized by category. You can locate a preference page by expanding the nodes of the tree in the left hand side of the dialog box.



The filter text box on the left hand side top corner allows you to quickly locate a preference page. As you type characters into the filter text box the items shown in the left hand side tree are filtered. For e.g. type in font to quickly get to the Font preferences page.



After you make changes to the appropriate preferences click on the **OK** button to save the changes and close the dialog box or click on the **Cancel** button to discard the changes and close the dialog box.

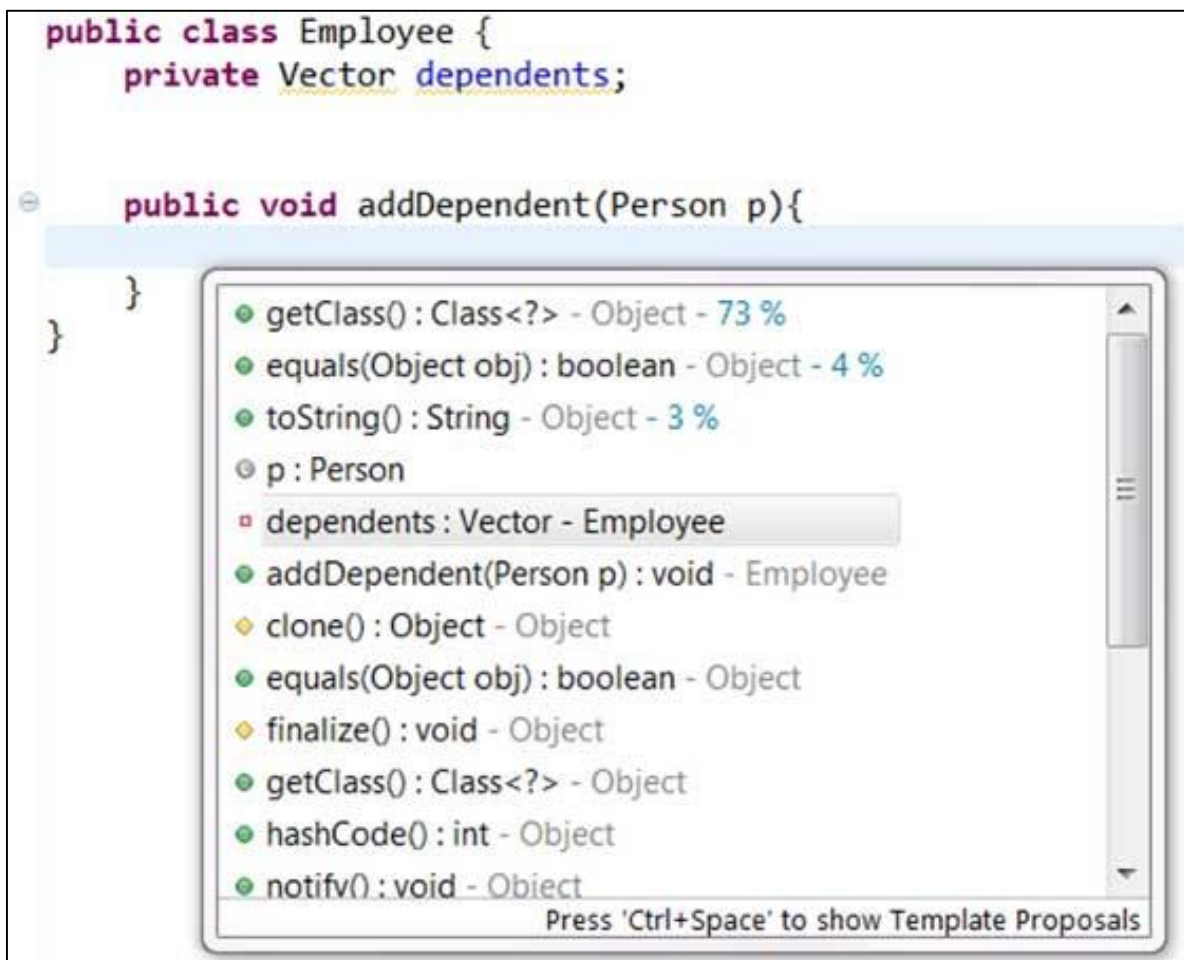
23. CONTENT ASSIST

Using Content Assist

Within an editor, content assist helps reduce the characters typed by providing a context sensitive list of possible completions to the characters already typed. The context assist can be invoked by clicking Ctrl + Space.

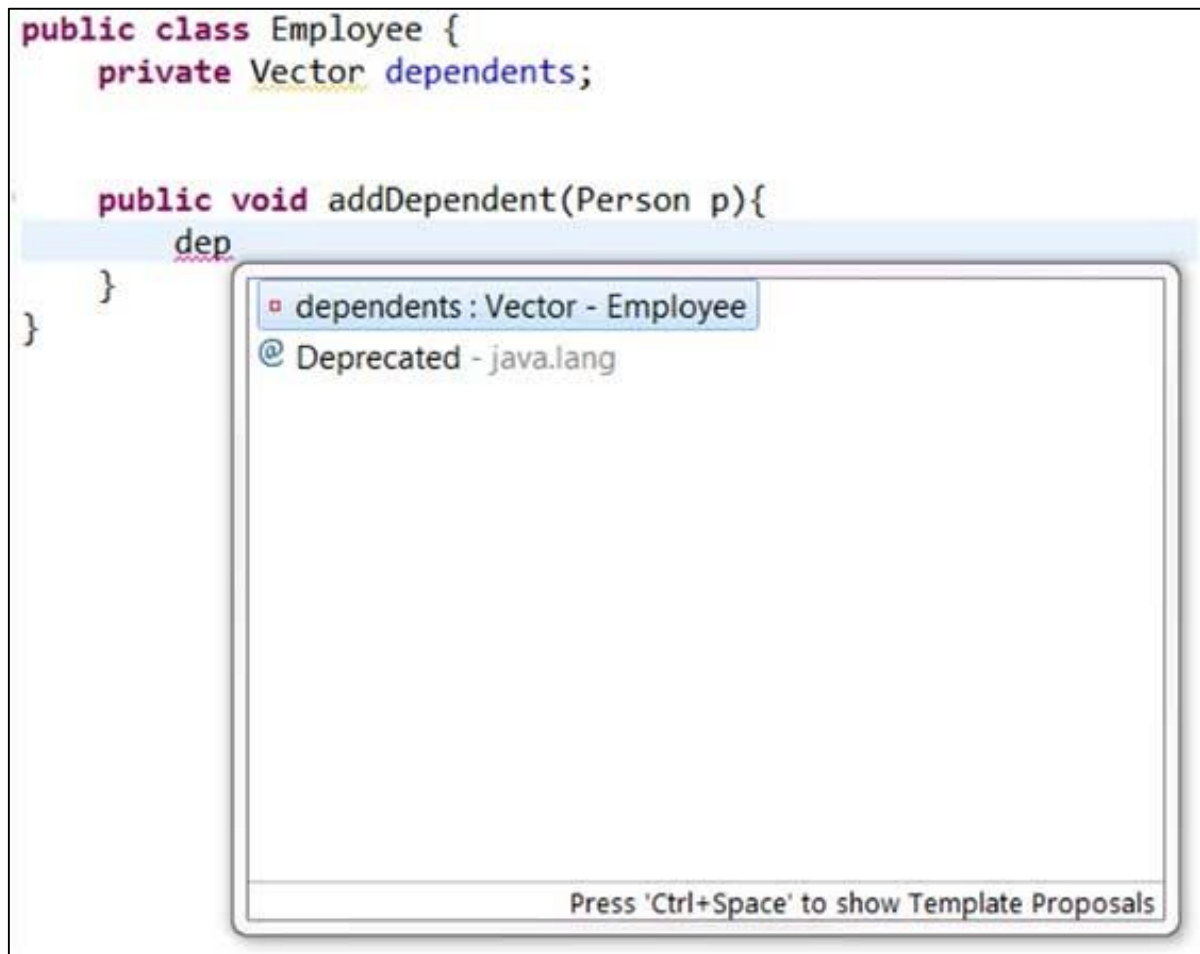
If you are editing the body of a method in the java editor and press Ctrl + Space on a new line it will bring up a dialog showing:

- The class variables
- The class methods
- Super class methods
- Other relevant classes



To select an item from the list just highlight the item using the up or down arrow keys and press enter.

If you type some characters and then press Ctrl + space the dialog box will show only items that start with the characters typed in.



24. QUICK FIX

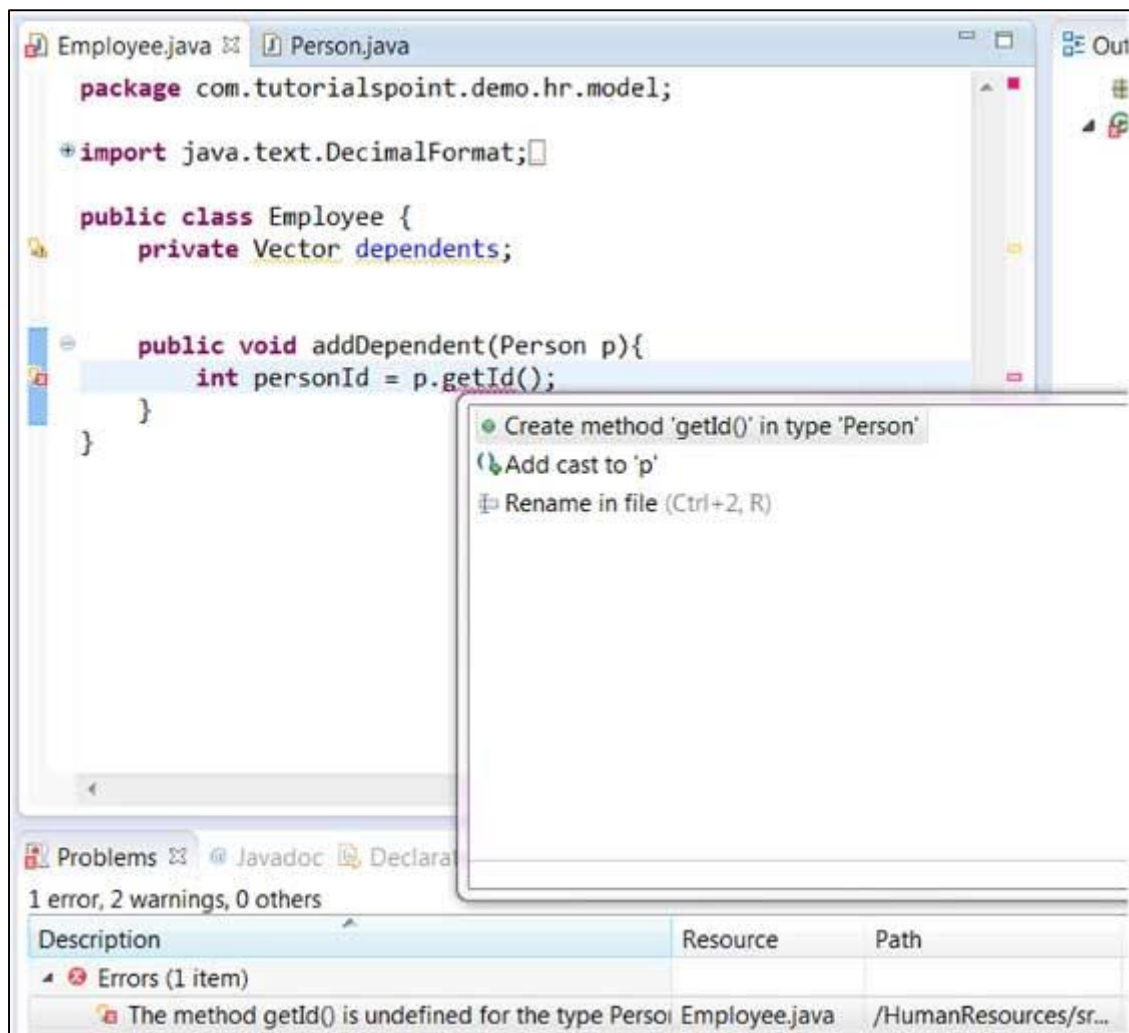
Using Quix Fix

As you type characters into an eclipse editor it analyzes the document content for potential error and warnings. The java editor uses the java syntax to detect errors in the code. When it finds error or warning, it:

- Highlights errors using red squiggly lines
- Highlights warnings using yellow squiggly lines
- Displays errors and warnings in the **Problem** view
- Adds a light bulb with a warning sign or error sign to the vertical ruler

The quick fix dialog provides a list of possible corrections. The quick fix dialog can be invoked by:

- Placing the mouse pointer on a squiggly line
- Clicking on the light bulb
- Placing the cursor in the highlighted text and selecting Quick fix from the Edit menu or by clicking shortcut "Ctrl + 1"



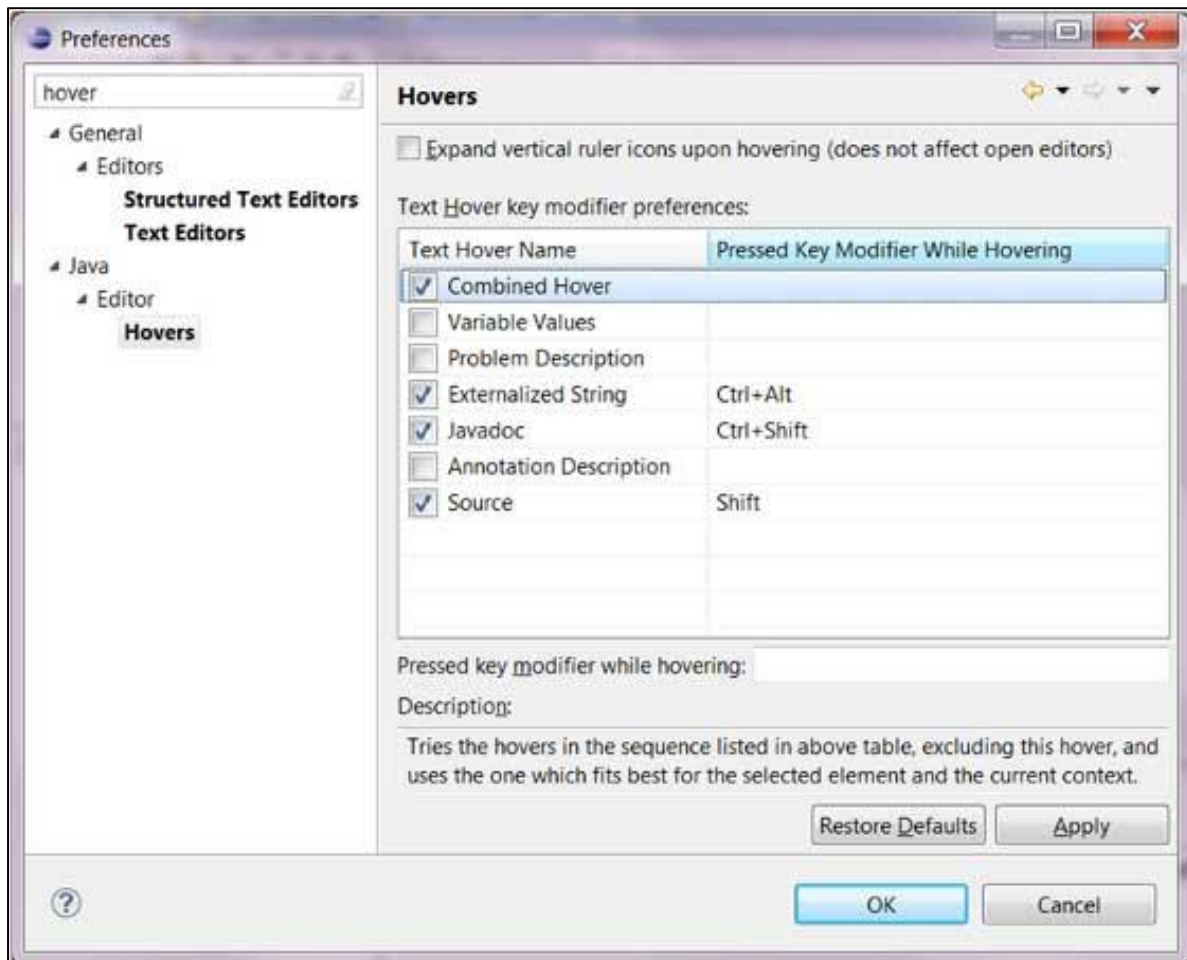
In the above diagram, `getId` is highlighted because the `Person` class does not have a method called `getId()`. Selecting "Create method 'getId()' in type 'Person'" would cause the `getId()` method to be added to the `Person` class.

The Quick Fix dialog can also be displayed by right clicking on the error item in the Problems view and selecting the Quick Fix menu item.

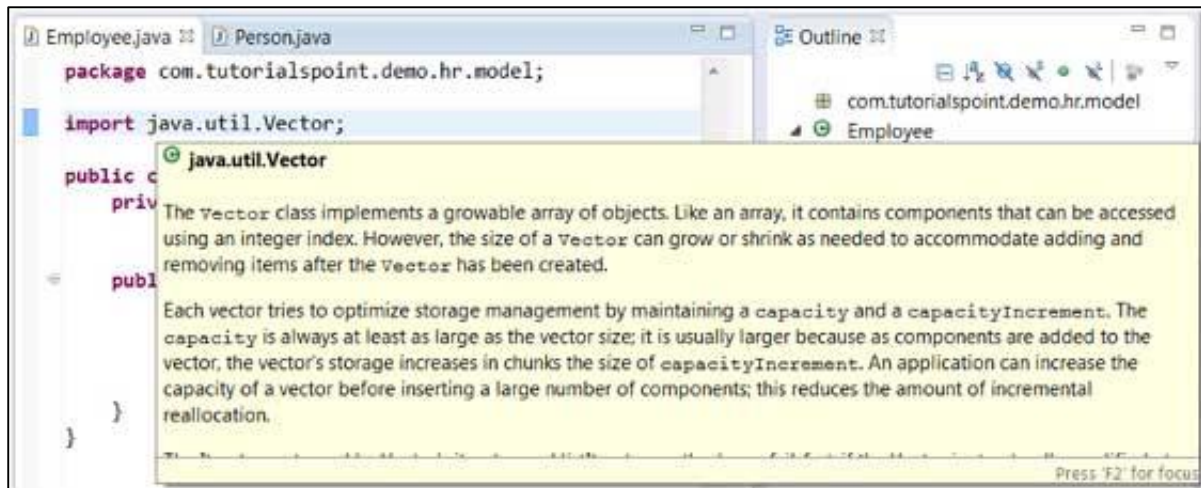
25. HOVER HELP

Using Hover Help

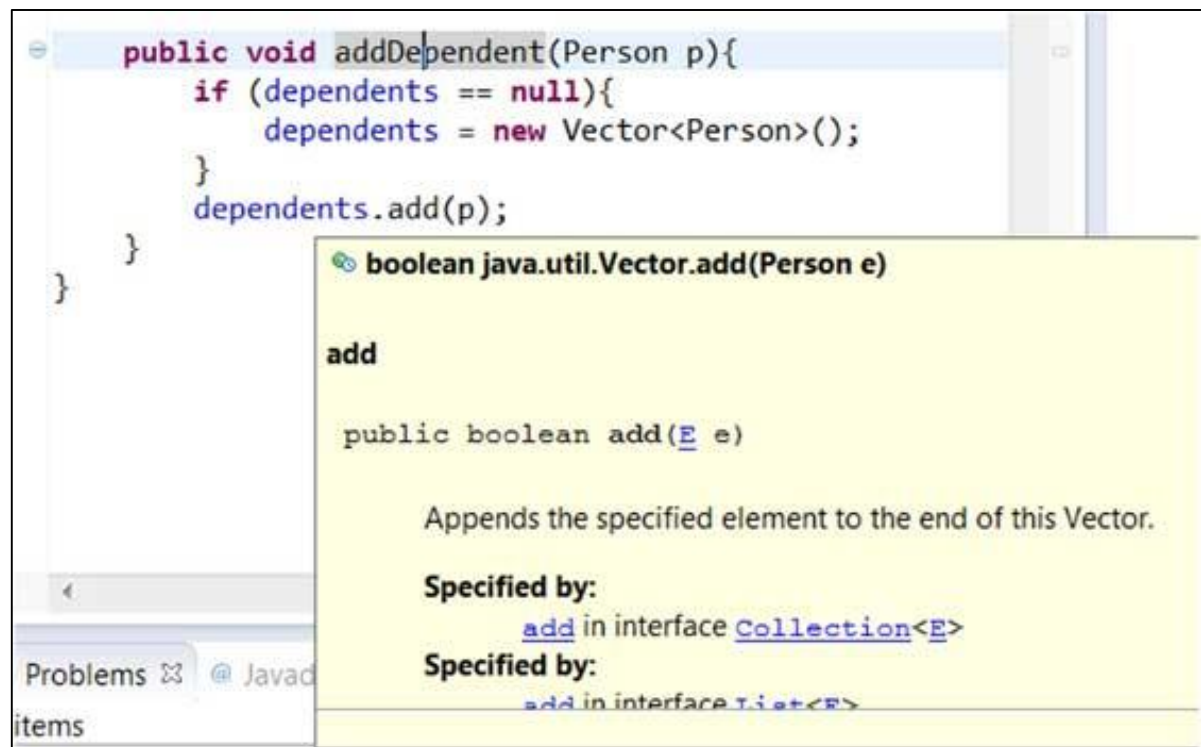
The java editor is associated with different type of hovers. A hover provides additional information about the element under the mouse pointer. All the hovers associated with a java editor can be seen in the Java editor Hovers preference page.



In the java editor, placing the mouse pointer over the class being imported displays the java doc associated with that class.



Placing a mouse pointer on a method shows the java documentation for that method.



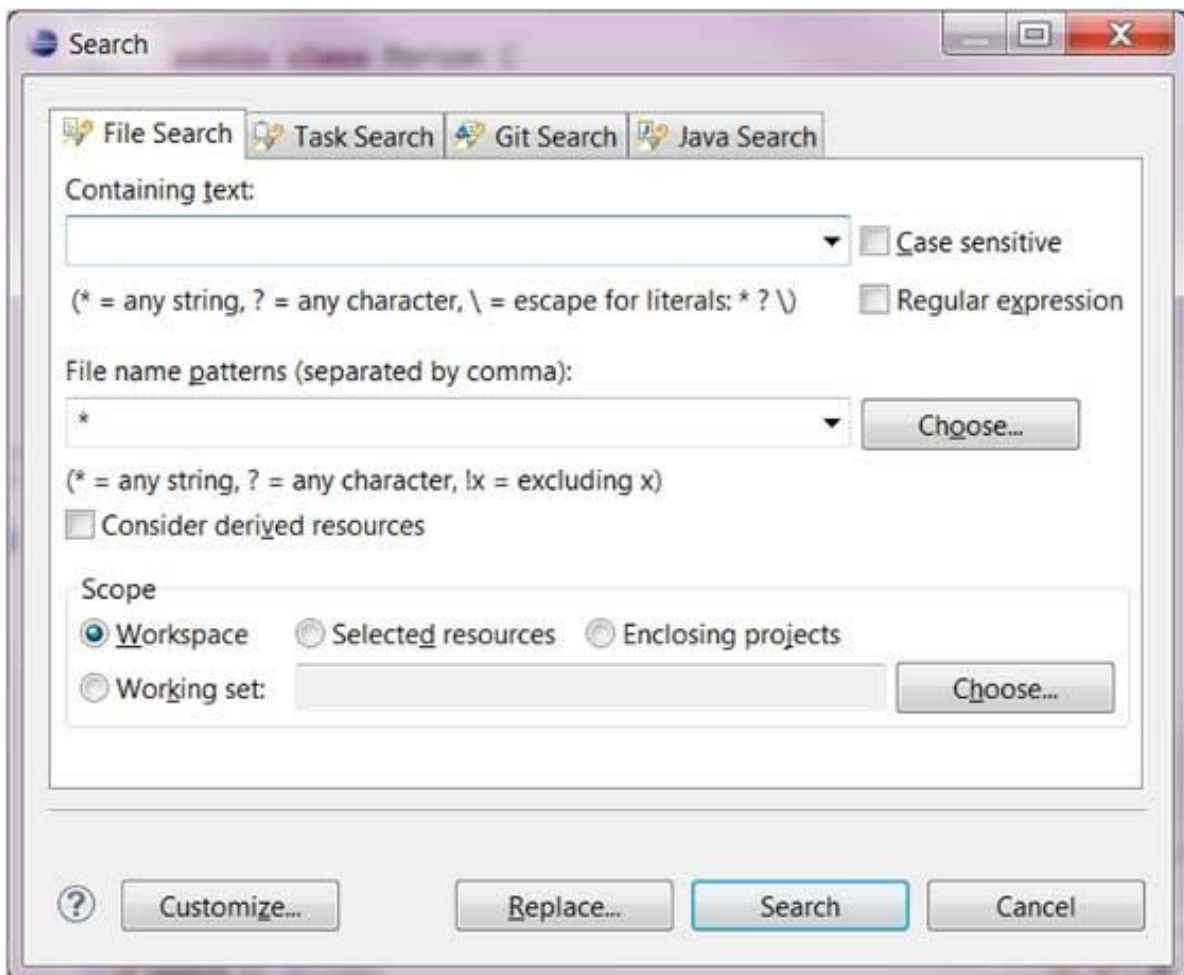
26. SEARCH MENU

Searching the Workspace

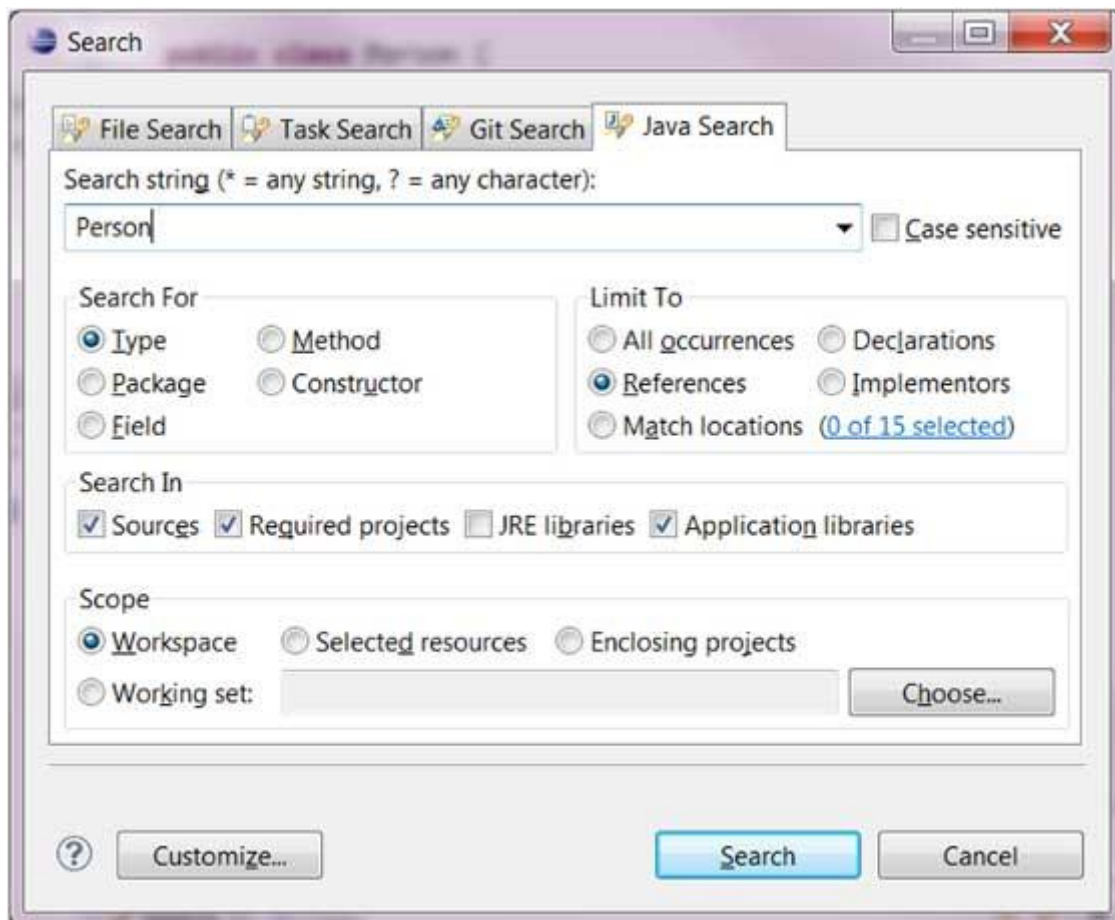
The Eclipse search dialog box allows you to search for files that contain a literal or a character pattern in the entire workspace, a set of projects, a specific project or folders selects in the package explorer view.

The search dialog box can be invoked by:

- Clicking on the Search menu and selecting Search or File or Java
- Clicking Ctrl + H



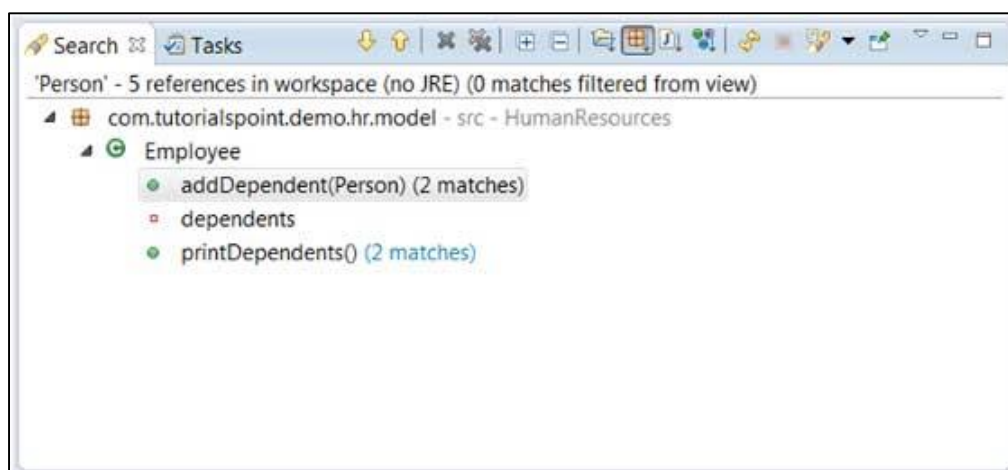
The File Search page allows you to search any type of file but the Java Search page focuses on Java files only.



For example, to search for usage of the Person type, in the Java Search page:

- Enter Person into the **search box**
- Set the **Search For** radio button to **Type**
- Set the **Limit To** radio button to **References**
- Click **Search**

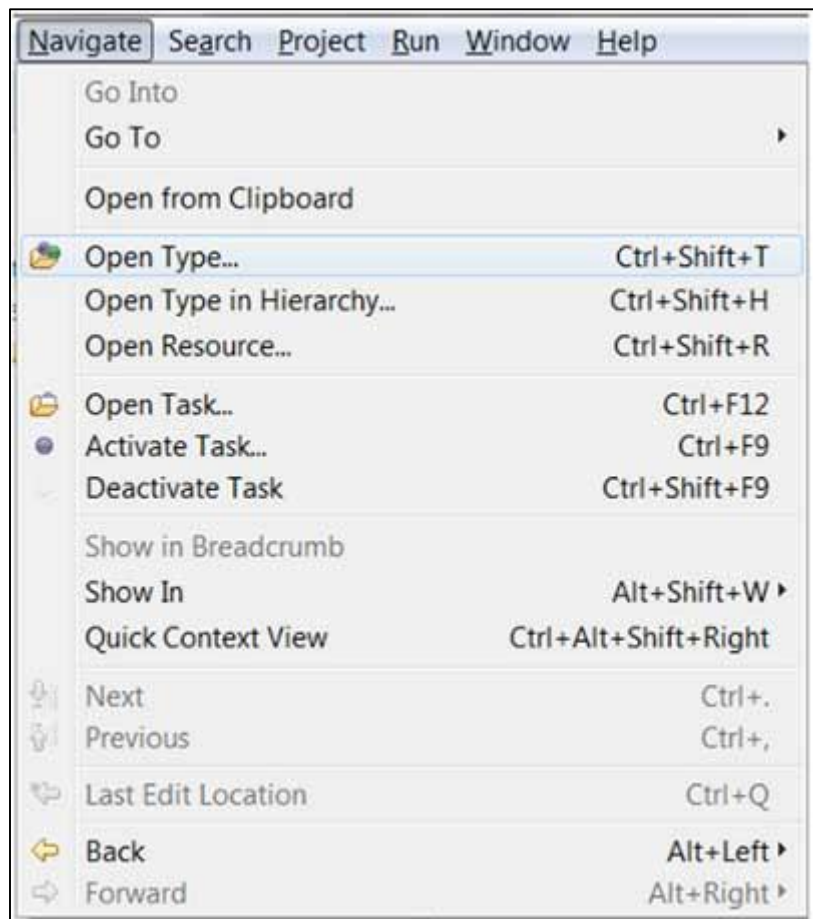
The search result appears in the Search View.



27. NAVIGATION

Navigating the Eclipse Workspace

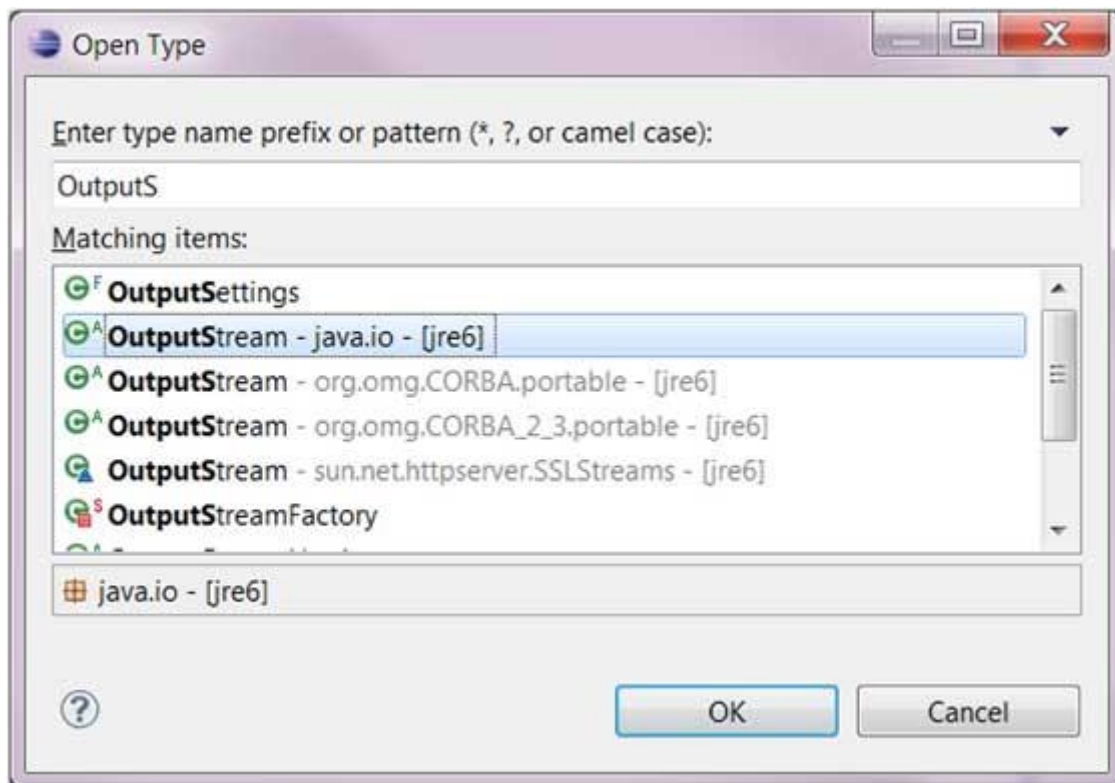
The navigate menu provides a number of menu items that allow you to quickly locate and navigate to a resource.



Among them, the Open Type, Open Type in Hierarchy and Open Resource menu items tend to be very useful.

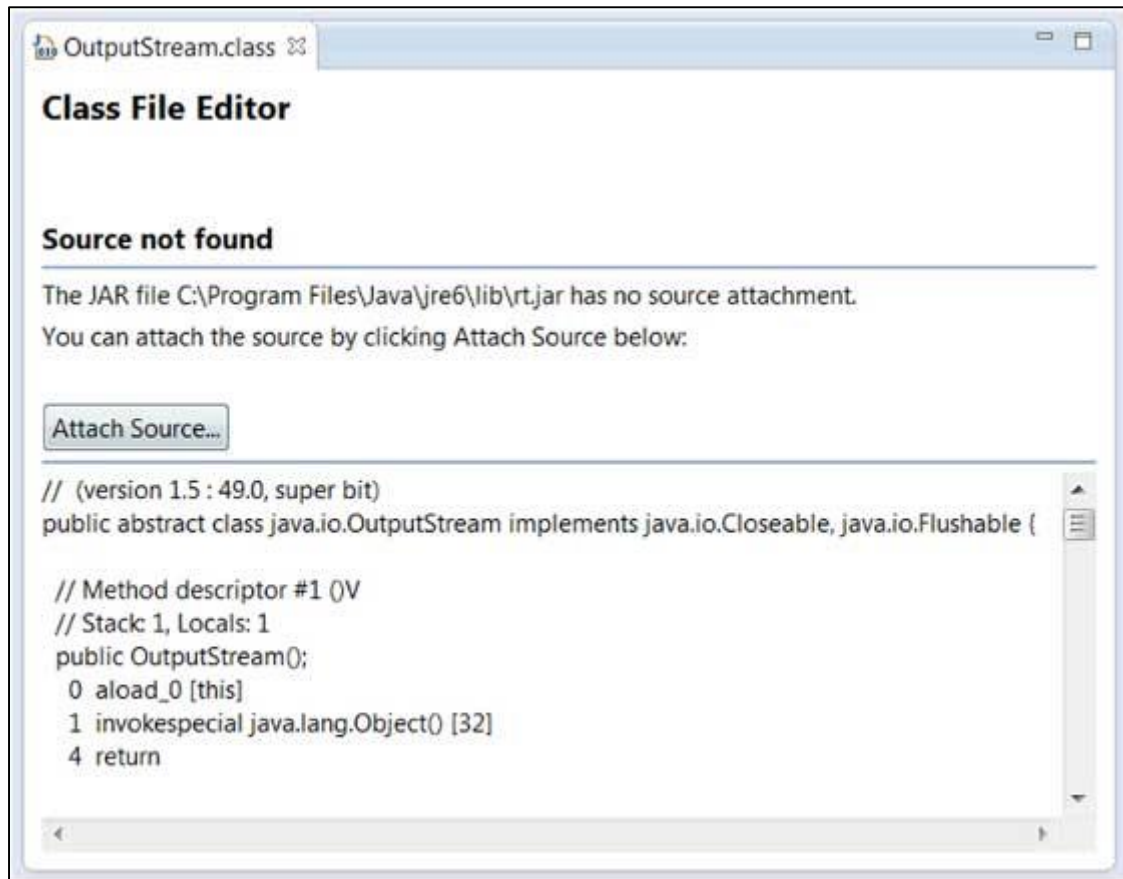
Open Type

The Open Type menu item brings up a dialog box that allows you to locate a Java type. In the filter text box enter in either the fully qualified name or name of a class. The '*' character which stands for 0 or more characters and '?' which stands for a single character can be used to specify patterns. The dialog box will show all the names that match the given pattern.



Select the type you are interested in and click **OK**.

Eclipse will open up an editor showing the selected type. If source code is not available for the selected type it will use the Class File editor to show the byte code of the selected type.

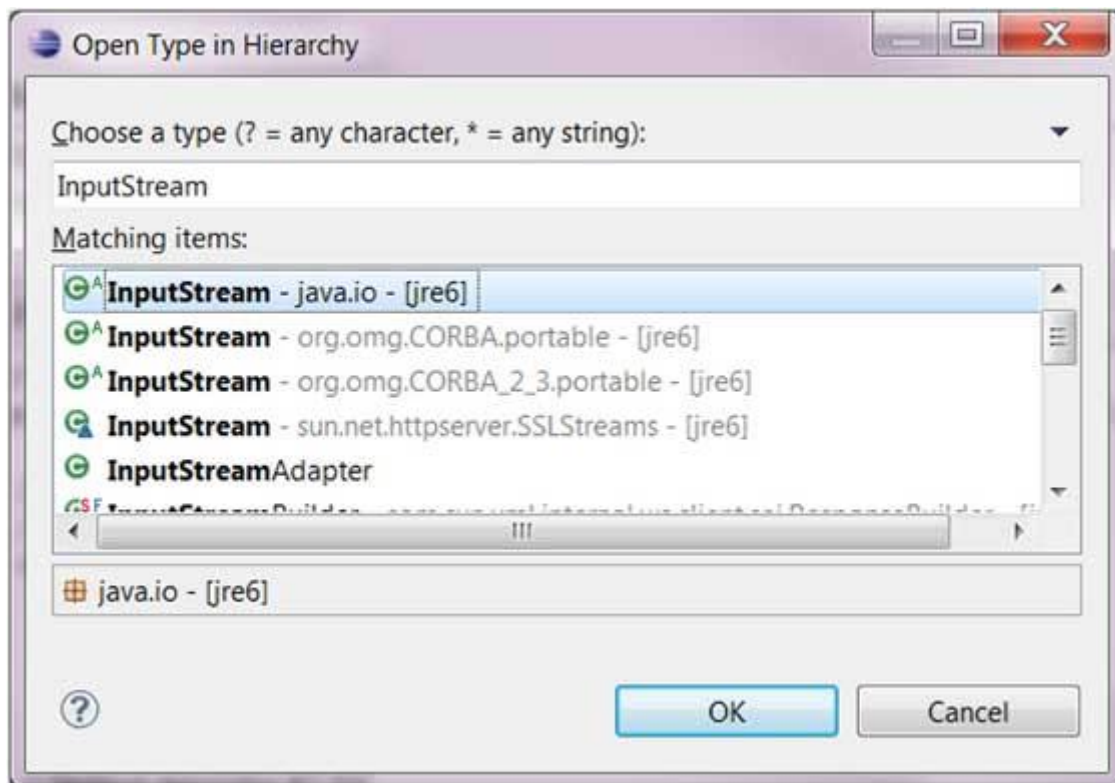


You can use the Attach Source button to specify where the source code for the class is located.

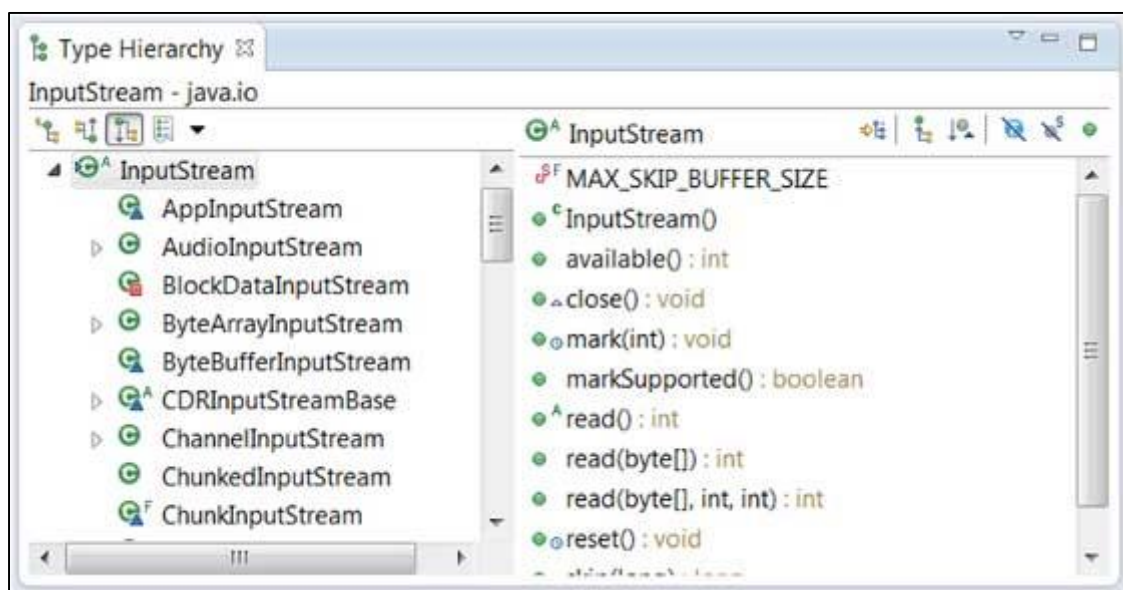
The source code for the types that comes with the java distribution is in src.zip which is located in the java home folder.

Open Type in Hierarchy

The Open Type in Hierarchy menu items allows you to open a type in the Type Hierarchy view. The Open Type in Hierarchy dialog box can be used to locate any Java type in the build path.



Once you select a type, its hierarchy is shown in the Type Hierarchy view.

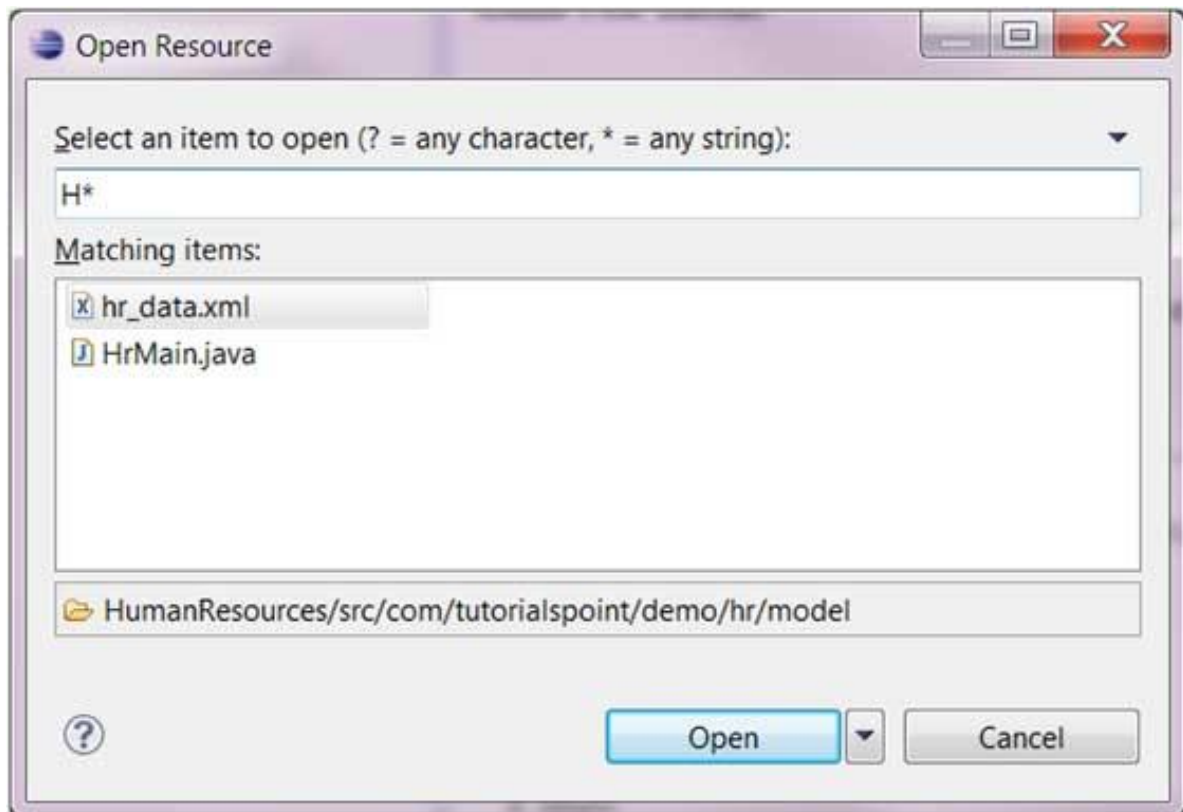


The Type Hierarchy is an excellent view for learning about the hierarchy of a type. On the left hand side pane: type, its super type and sub type are available. On the right hand side pane: attributes and methods of a selected type are shown.

Open Resource

The open resource menu item can be used to locate a file in the workspace. The '*' character which stands for 0 or more characters and '?' which stands for a single character

can be used to specify patterns. The dialog box will show all the names that match the given pattern.



Select the file that you want to open in an editor and click on the OK button.

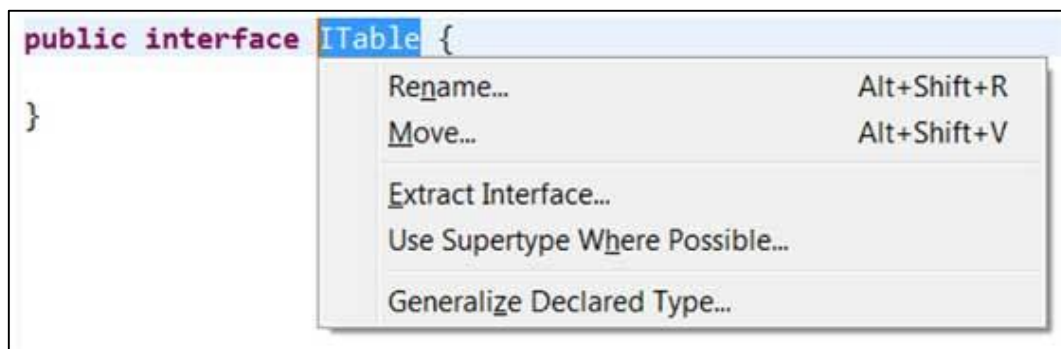
28. REFACTORING

Refactoring using Eclipse

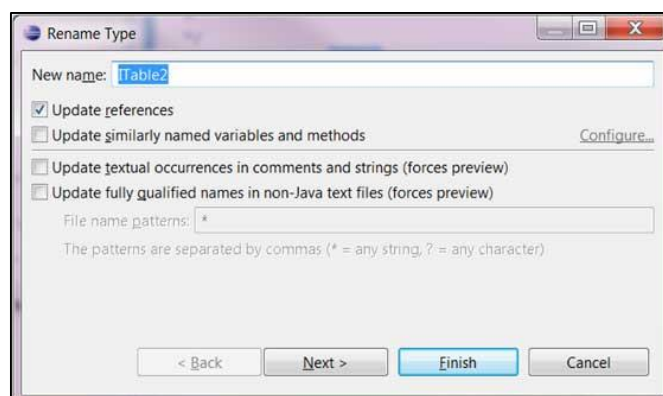
Renaming a class that is used widely throughout a project could take a lot of time but the Eclipse refactoring wizard makes the job easier by automatically detecting all dependencies on the class being renamed and modifying them also.

The refactor menu can be opened by:

- Right clicking on a Java element in the Package Explorer view and selecting Refactor menu item
- Right clicking on a Java element in the Java editor and selecting Refactor menu item
- Selecting a Java element in either the Package Explorer view or Java Editor and clicking Shift + Alt + T



The refactor menu shows all the possible changes that are supported on the selected Java element. To rename a class, select the Rename menu item.



Click on the Next button to see a preview of the changes (if available). Then click on Finish to get the wizard to do its work.

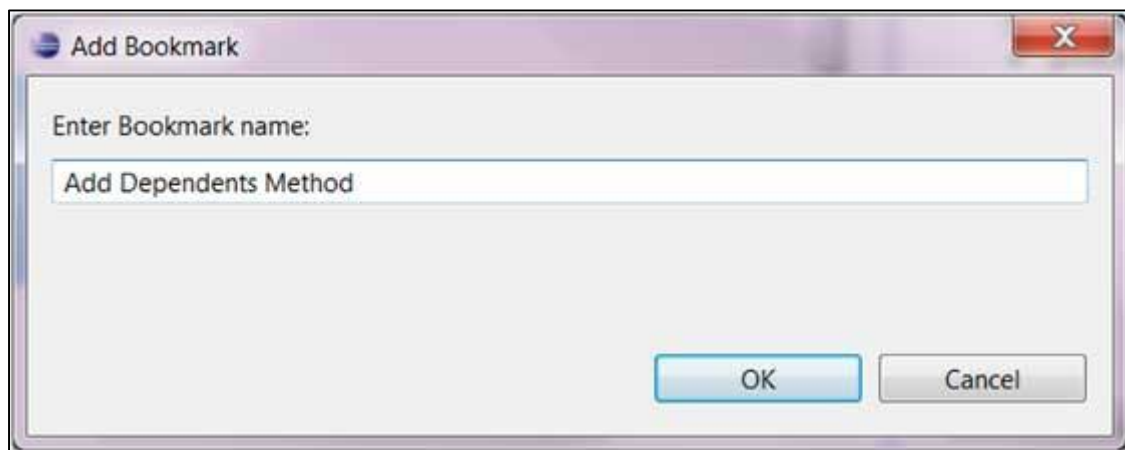
29. ADD BOOKMARKS

About Bookmarks

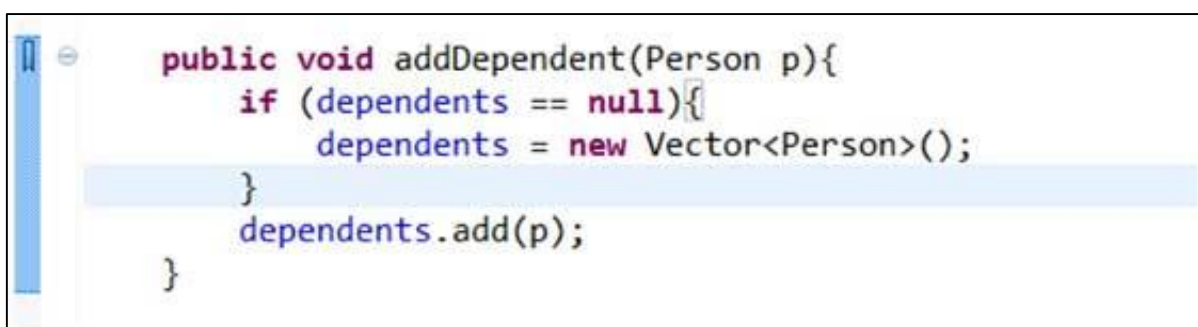
Eclipse editors allow you to place a book mark on any line within the editor. You can use bookmarks to serve as reminders and quickly navigate to any line in a file.

Adding a Bookmark

To set a bookmark, right-click on the vertical ruler near the line on which you want the bookmark and select "Add Bookmark". In the Add Bookmark dialog box enter a name for the Bookmark



A bookmark icon appears in the vertical ruler. Also, a row is added to the table in the Bookmarks view.

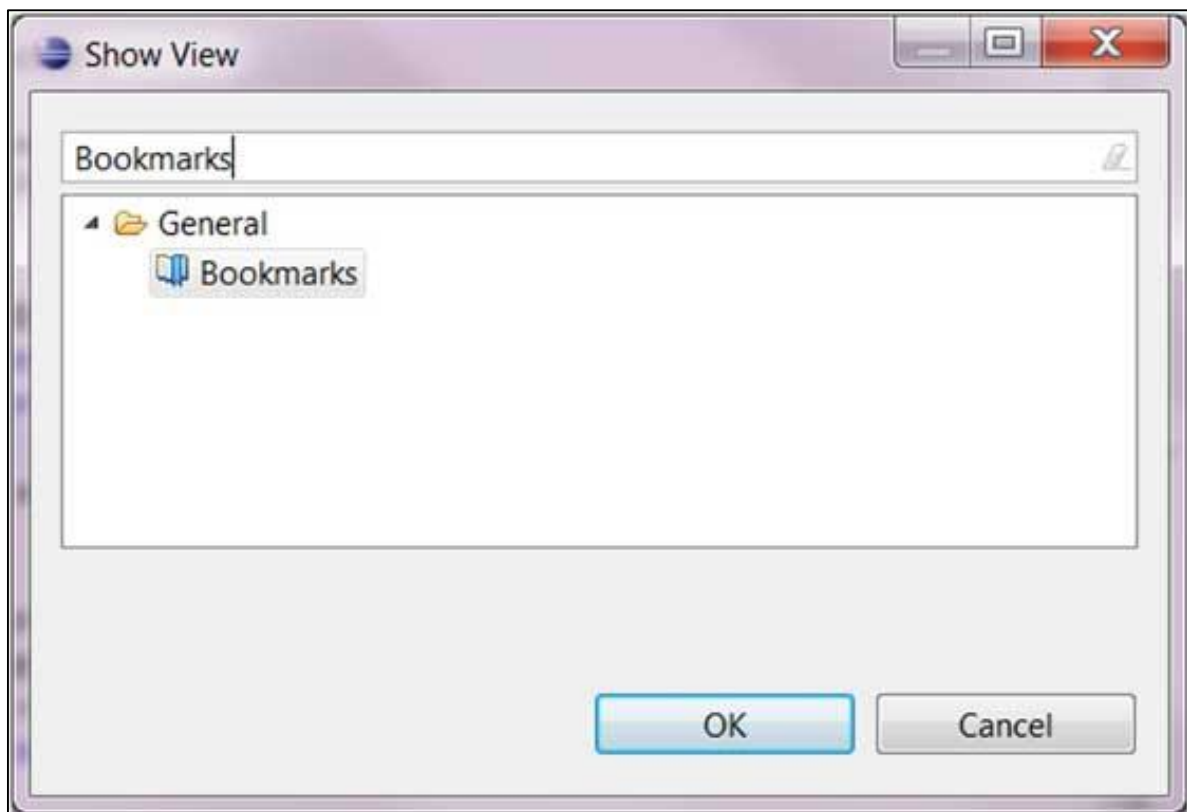


Opening the Bookmarks View

To open the Bookmarks view:

- Click on the Window menu and select Show View → Other
- In the filter text box enter Bookmark

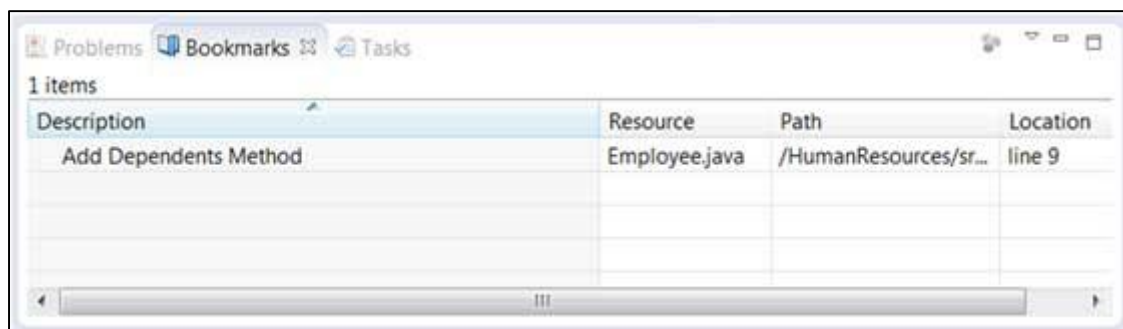
- Under **General**, select Bookmarks



- Click on OK

Using the Bookmarks View

The Bookmarks view shows bookmarks placed in resources across the workspace.

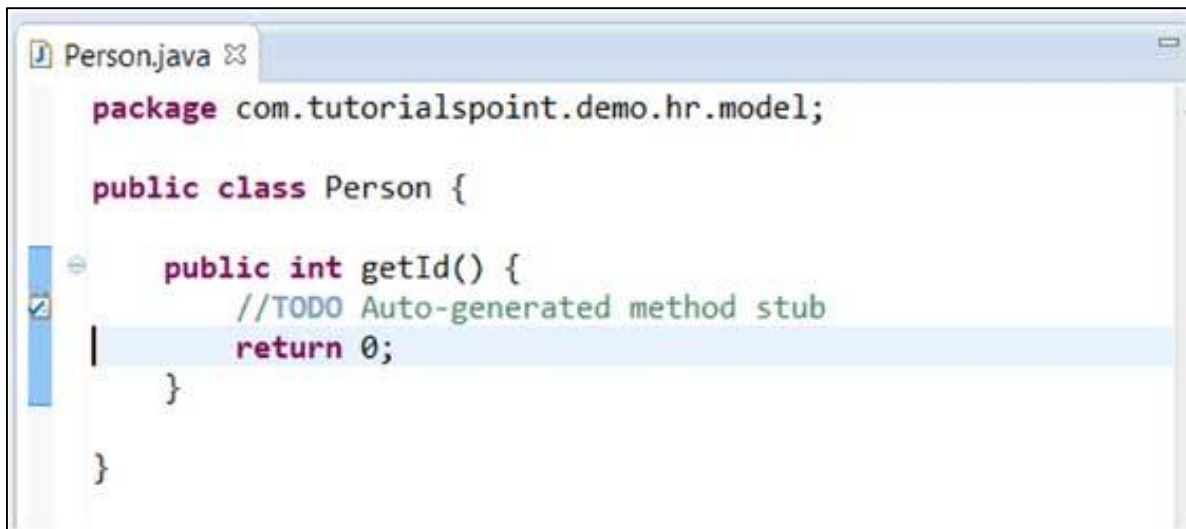


Using the Bookmarks view you can navigate to a bookmark by double clicking on a bookmark or right clicking on a bookmark and selecting "Go to" menu item.

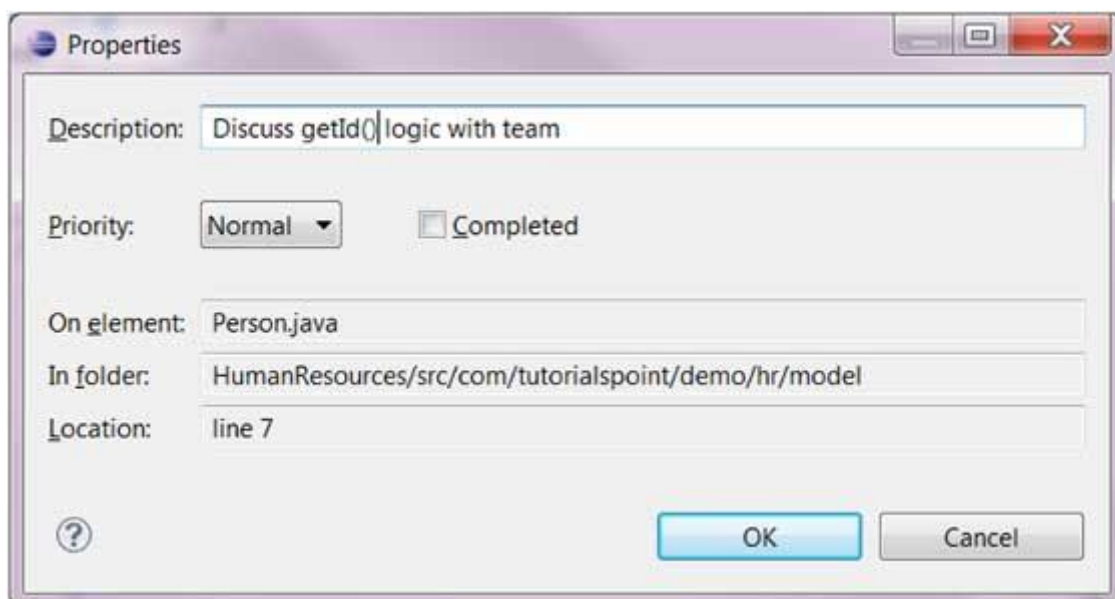
30. TASK MANAGEMENT

Managing Tasks

Programmers like to place TODO markers in their code which serves as a reminder for tasks that need to be completed. Comments in the Java code that has the word TODO in them are recognized as task and shown on the marker bar and in the Tasks view.



An eclipse editor can be used to associate tasks with the file being edited by right clicking on the marker bar and selecting Add Task. In the dialog box that comes up enter the task description and select a priority from the priority drop down list and then click on the OK button.

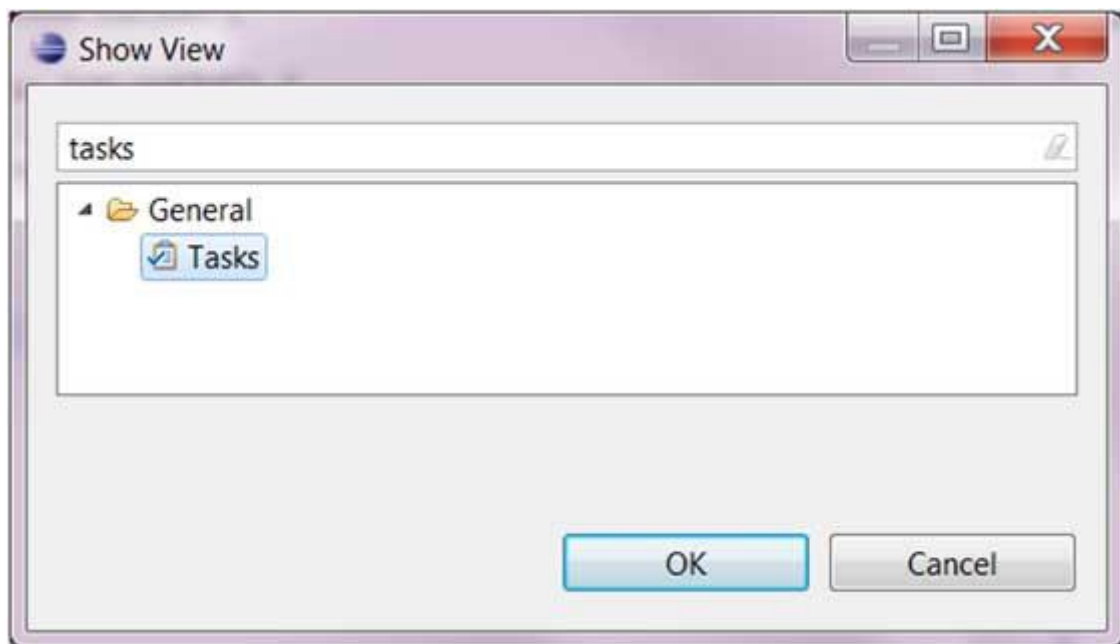


To remove a task using an Eclipse editor, right-click on the task icon in the marker bar and select the Remove Task menu item.

Opening the Tasks View

To open the Tasks view:

- Click on the Window menu and select Show View → Other
- In the filter text box enter Tasks
- Under **General**, select Tasks



- Click **OK**

Using the Tasks View

The Tasks view can be used to view all the tasks and add tasks not associated with any resource.



It can also be used to:

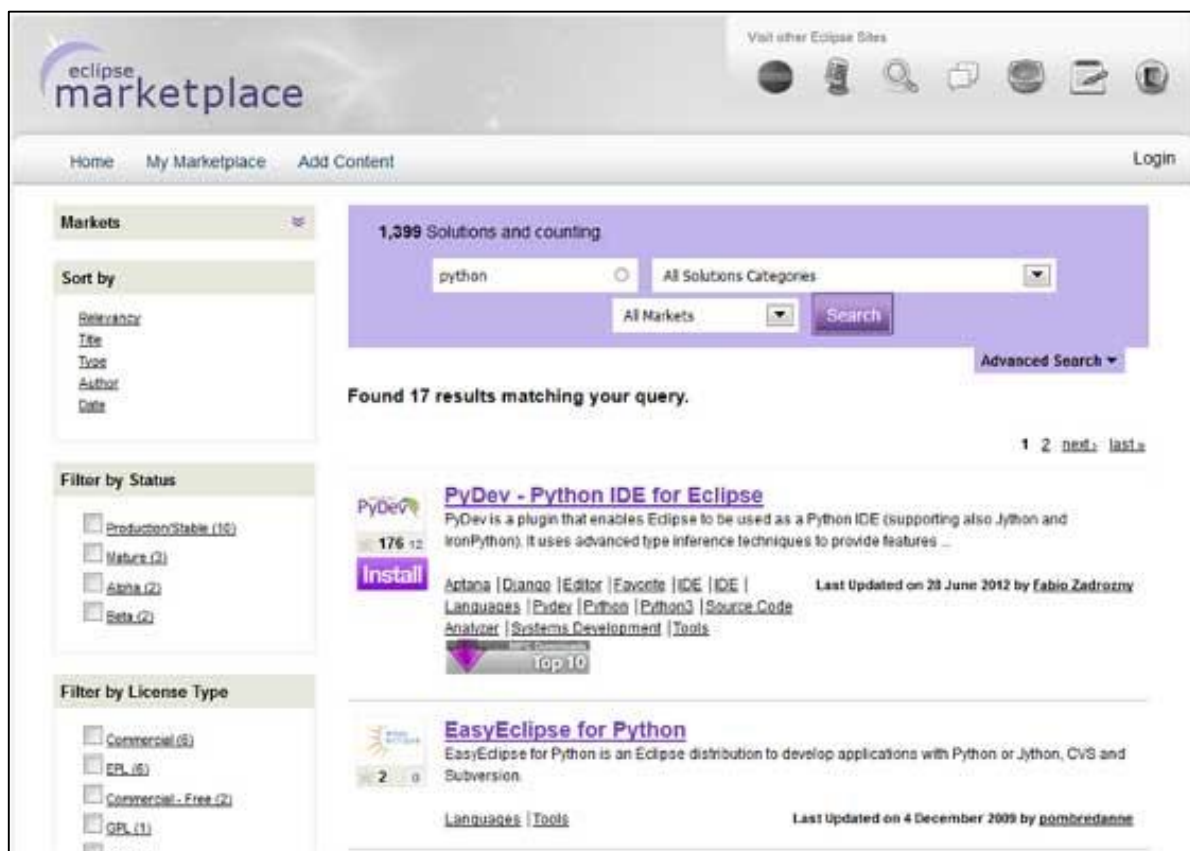
- Change the priority associated with a task
- Mark a task as completed
- Remove a task or all completed tasks

31. INSTALL PLUGINS

Locating and Installing Plug-ins

The Eclipse platform which provides the foundation for the Eclipse IDE is composed of plug-ins and is designed to be extensible using additional plug-ins.

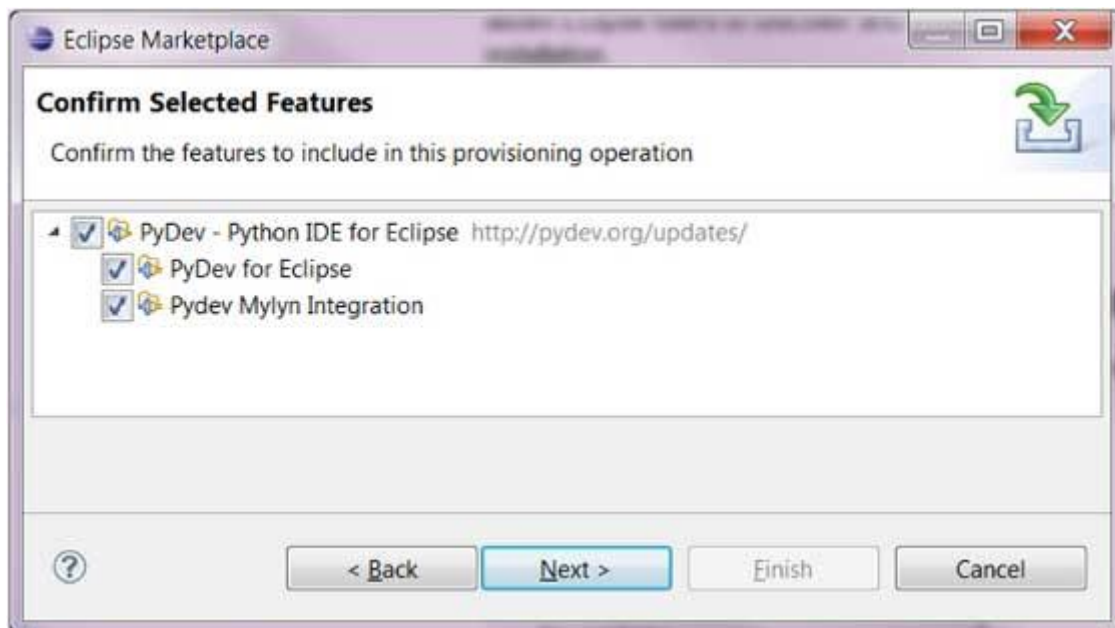
Several hundreds of plug-ins are available. Each plug-in adds more functionality to Eclipse. You can locate a plug-in that provides certain functionality by searching the Eclipse Marketplace (<http://marketplace.eclipse.org/>). For example you can locate plug-ins that allows eclipse to be used as a Python IDE.



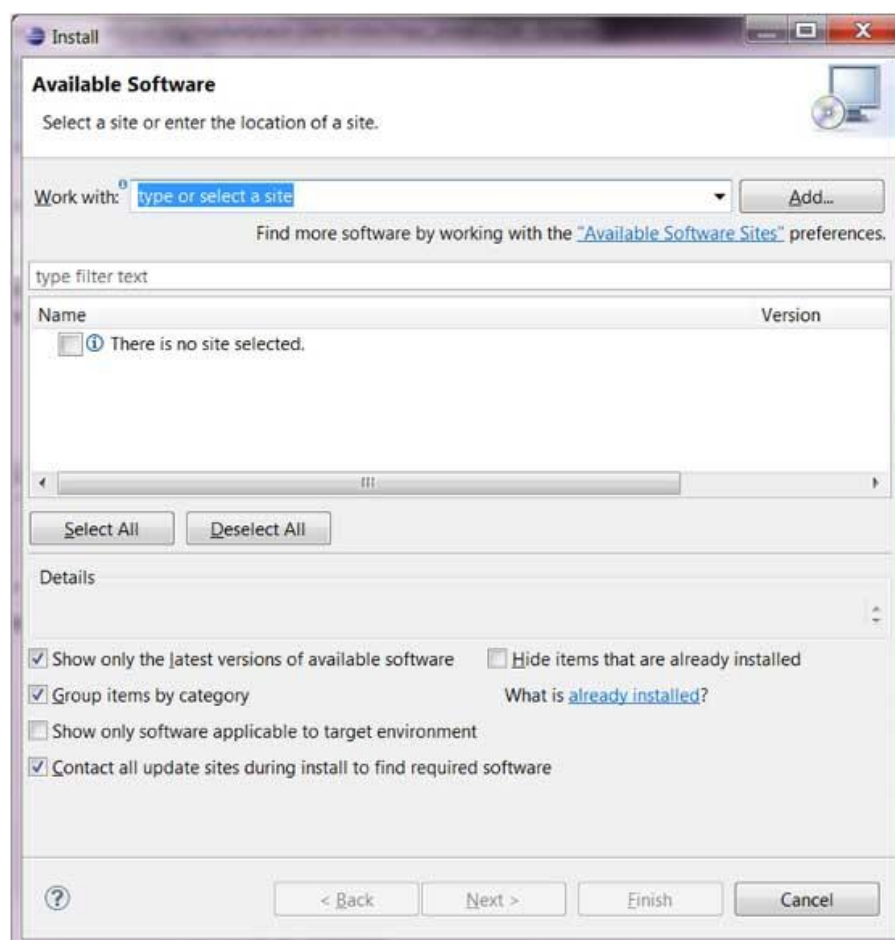
From within the Eclipse IDE, you can search the market place by using the Eclipse Marketplace dialog which can be invoked by clicking on the Help menu and selecting Eclipse Marketplace.



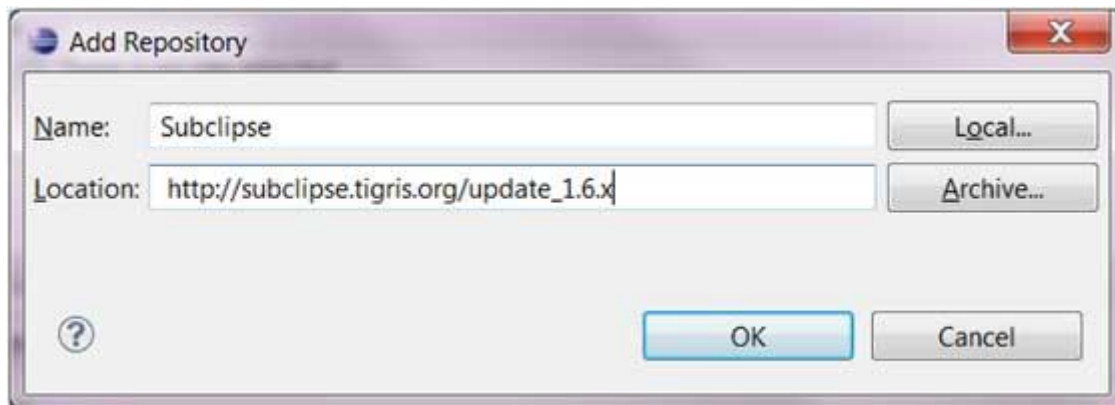
In the screenshot above, to install "PyDev - Python IDE for Eclipse", just click on the Install button. Using the dialog box that comes up you can install the plug-in.



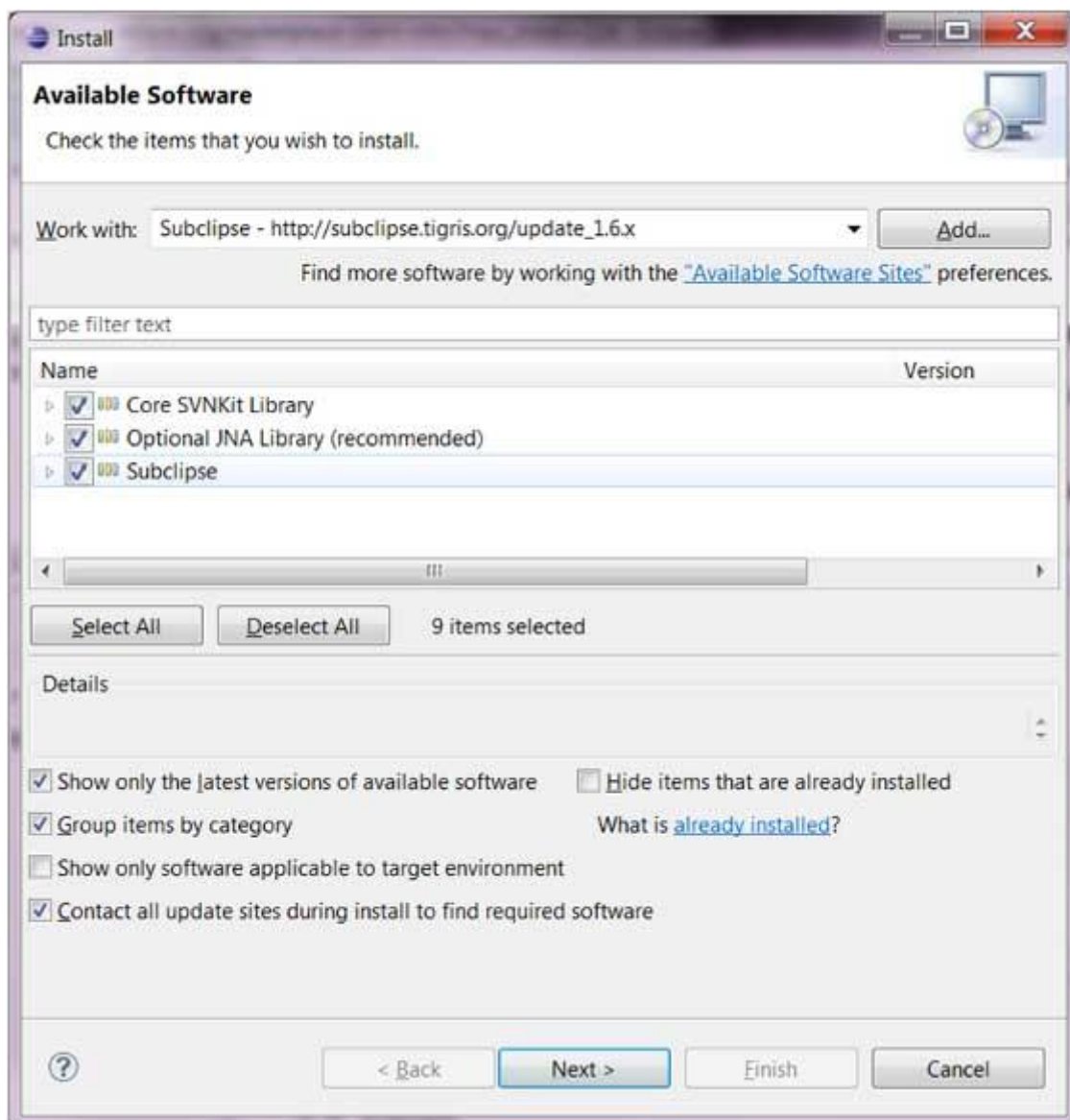
You can also install a plug-in by using the Install New Software menu item accessible from the Help menu (Help → Install New Software)



In this case, you have to know the URL of the remote site from which the plug-in can be downloaded. You can supply this URL by clicking on the Add button.



The **install** dialog box shows the plug-in components available in the remote site and allows you to install them.

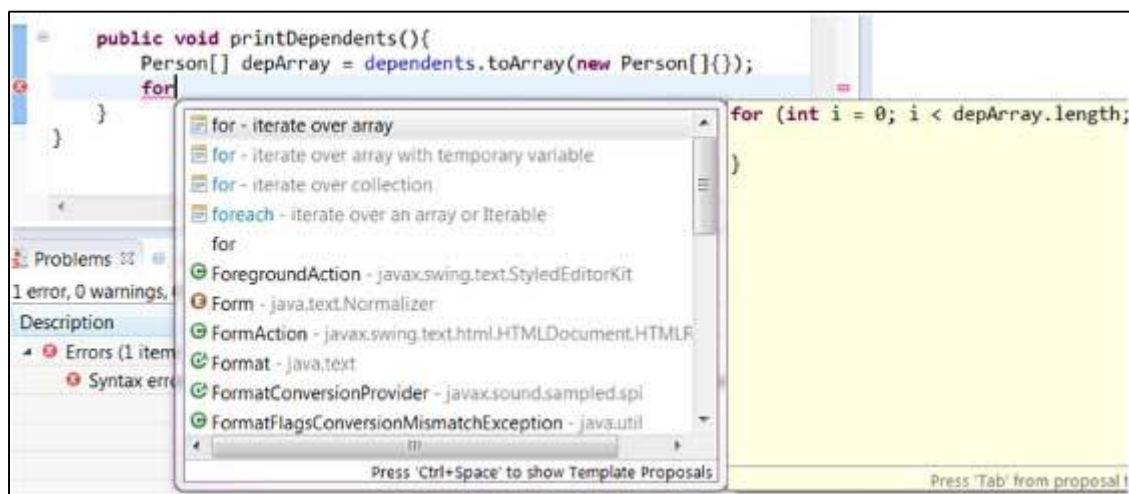


32. CODE TEMPLATES

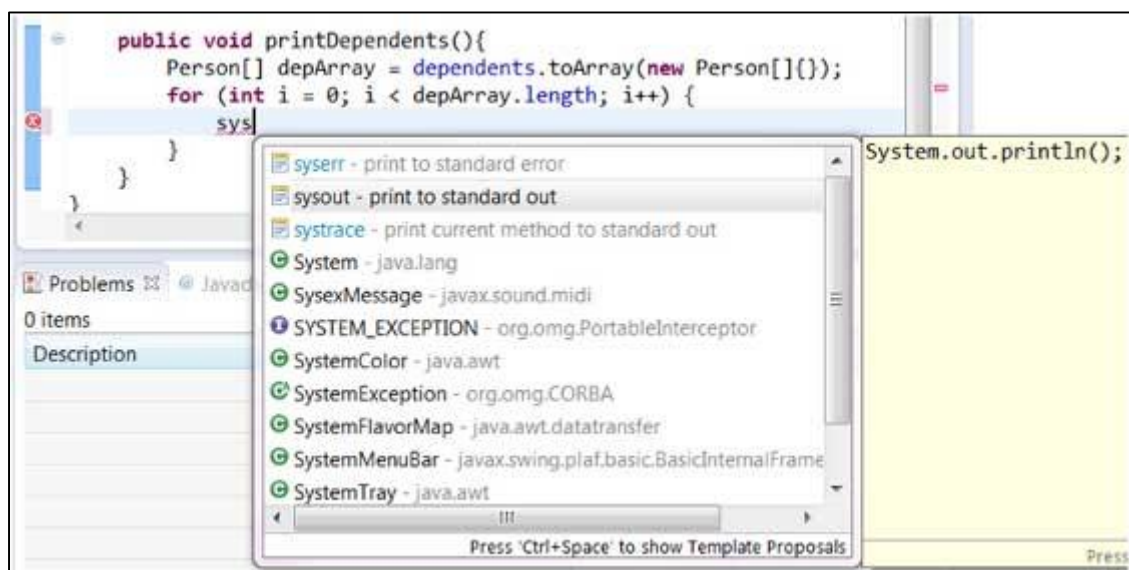
Using Code Templates

Code templates help reduce typing time by inserting code into the editor. Each code template is given a short literal. Typing the literal into the editor window and press Ctrl + Space brings up a dialog box from which a code template associated with that literal can be selected.

Typing "for" within the body of a method in the java editor window and clicking Ctrl + space shows the code templates associated with "for".

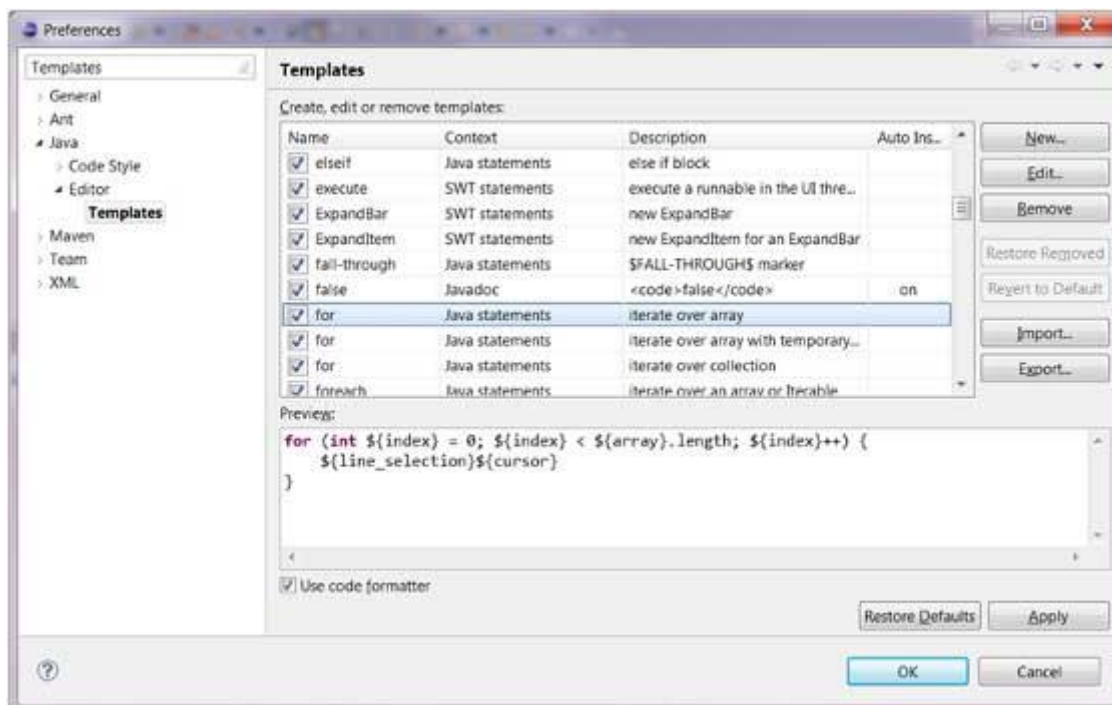


Selecting "for - iterate over array" inserts code for iterating through depArray which was declared earlier in the method body. Another useful code template literal is **sysout** which can be used to insert the System.out.println() method call.

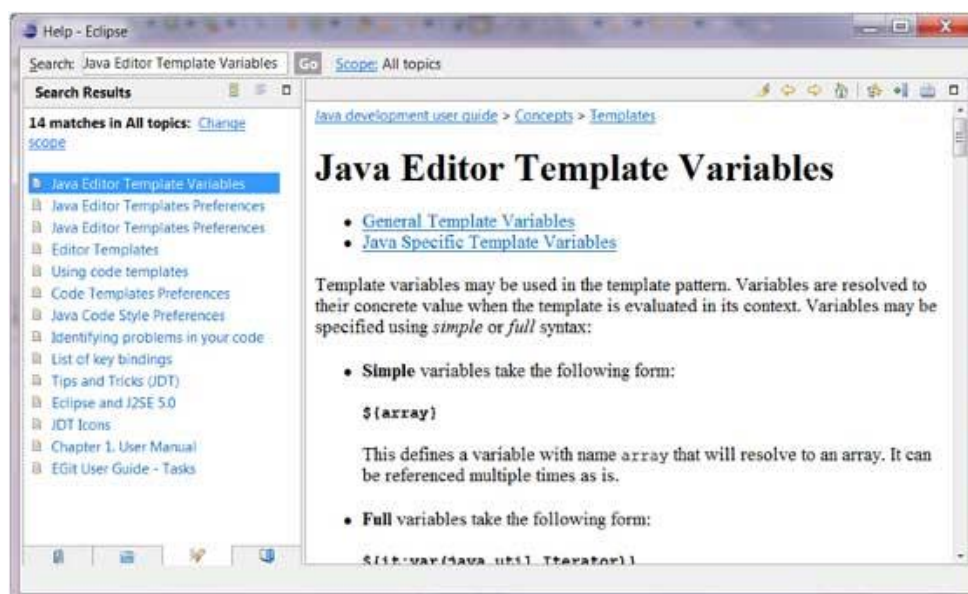


Modifying/Adding code templates

You can add new code templates using the Preferences dialog box. Bring up the Preferences dialog box by clicking on the Windows menu and selecting the Preferences menu item. To locate the Templates page type Templates into the filter text box



You can read more about the variables that can be used to customize the code that is inserted by reading the Eclipse help. To bring up the help window click on the Help menu and select Help Contents. In the search text box enter "Java Editor Template Variables". From the list on the left hand side select "Java Editor Template Variables".

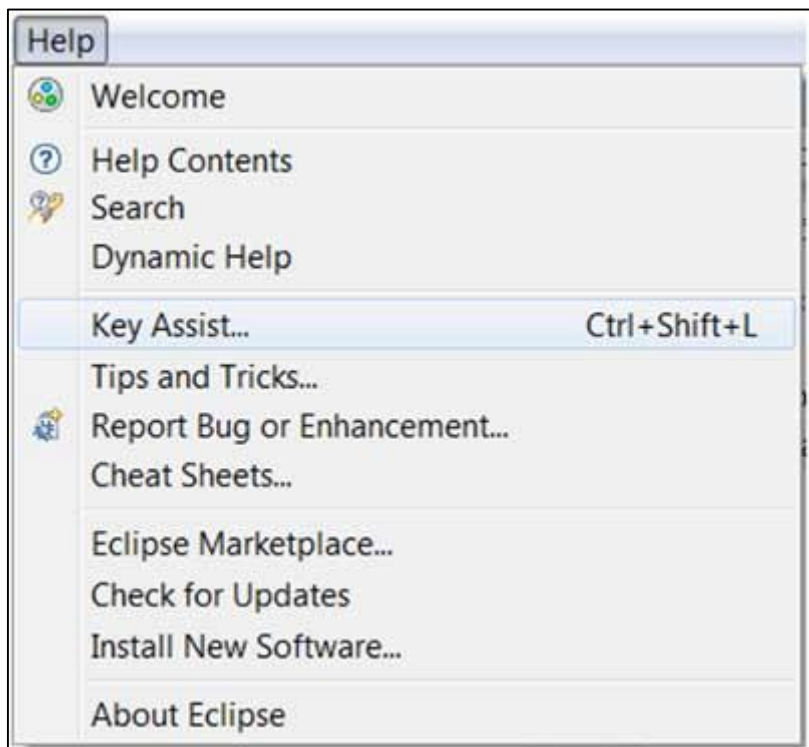


33. SHORTCUTS

About Shortcuts

You can navigate the Eclipse User Interface using only the keyboard by:

- Using mnemonics associated with menus and menu items
- Using mnemonics associated with controls in a dialog box or view or editor
- Using accelerators associated with actions such as menu items and buttons on the toolbar



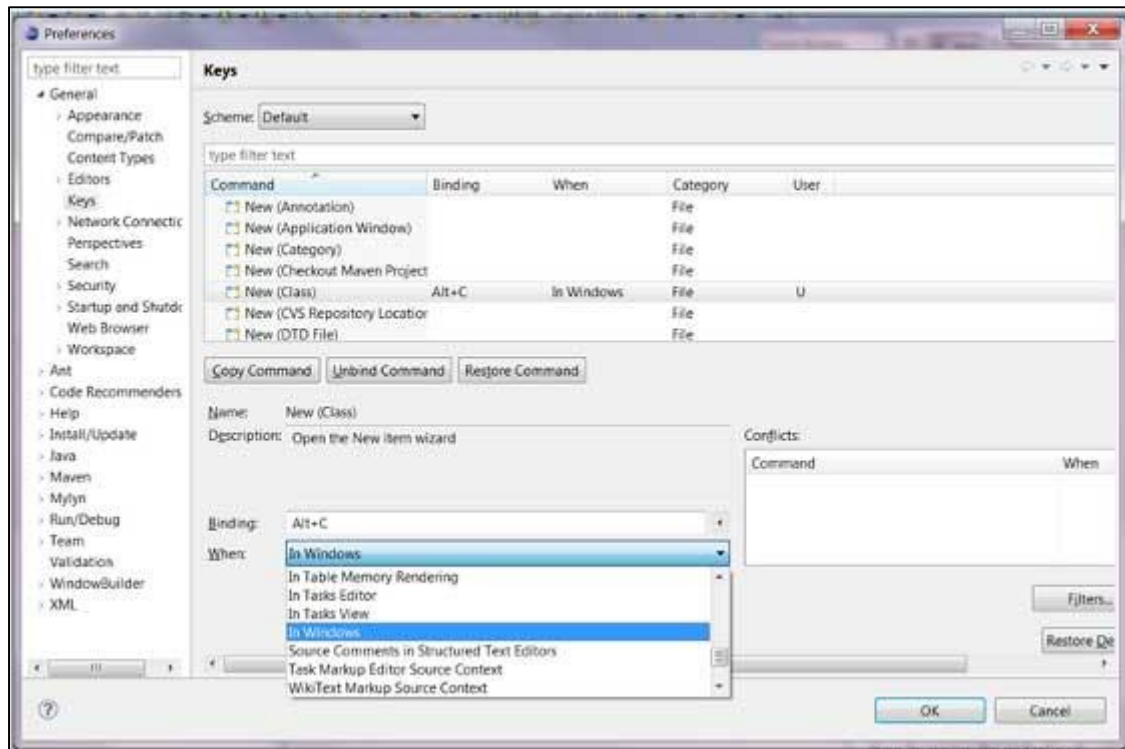
The Key Assist action which can be invoked by clicking Ctrl + Shift + L shows us all the accelerators or shortcut keys available in Eclipse.

Activate Editor	F12
Activate Task	Ctrl+F9
Add Javadoc Comment	Alt+Shift+J
All Instances	Ctrl+Shift+N
Backward History	Alt+Left
Build All	Ctrl+B
Change Method Signature	Alt+Shift+C
Close	Ctrl+F4
Close All	Ctrl+Shift+F4
Collapse All	Ctrl+Shift+Numpad_Divide
Commit...	Ctrl+#
Content Assist	Ctrl+Space
Context Information	Ctrl+Shift+Space
Copy	Ctrl+Insert
Cut	Shift+Delete
Deactivate Task	Ctrl+Shift+F9
Debug	F11
Debug Ant Build	Alt+Shift+D, Q
Debug JUnit Test	Alt+Shift+D, T
Debug Java Applet	Alt+Shift+D, A
Debug Java Application	Alt+Shift+D, J

Press "Ctrl+Shift+L" to open the preference page.

The key combination assigned to an action can be changed using the Keys preference page. There are many commands or actions in Eclipse for which there are no shortcut keys. This preference page also allows you to assign shortcut keys to these actions.

For example, to assign **Alt + C** as the shortcut key to trigger the command that brings up the **New Class** dialog box select **New Class** in the command list, navigate to the **Bindings** text box and then press **Alt + C**. The "when dropdown list" allows you to select a context in which the shortcut key is valid. Ensure that there are no conflicts for the selected key combination and click on the OK button.

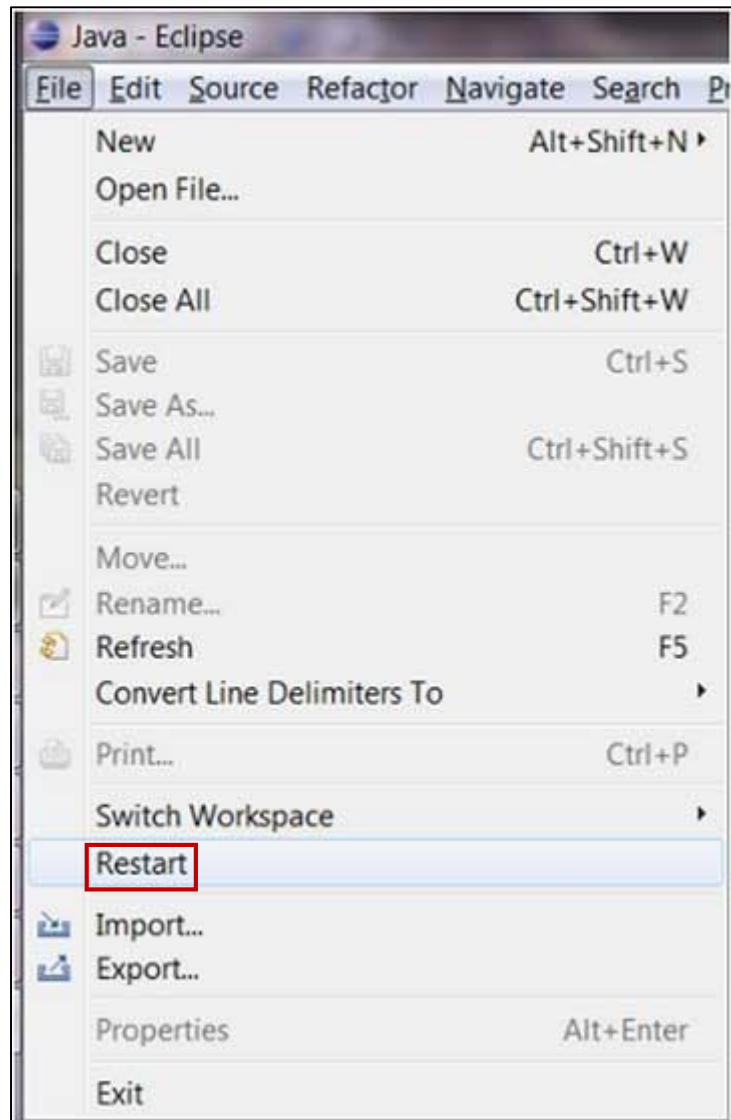


Now you can press **Alt + C** to bring up the New Class wizard.

34. RESTART OPTION

Restarting Eclipse

The restart option allows users to restart eclipse. To restart eclipse, click on the File menu and select the **Restart** menu item.



After a plug-in is installed, users will be prompted to restart eclipse. If they choose not to restart at that point they can restart eclipse later by using the restart option. If eclipse user interface misbehaves, users may want to restart eclipse using this option.

35. TIPS & TRICKS

TIP #1 - DISCOVERING SHORTCUT KEYS

- Press **Ctrl + Shift + L** to open a widget that shows all the shortcut keys.

TIP #2 - CONTENT ASSIST

- In the Java editor press **Ctrl + Space** to see a list of suggested completions. Typing one or more characters before clicking **Ctrl + Space** will shorten the list.

TIP #3 - PARAMETER HINT

- When the cursor is in a method argument, press **Ctrl + Shift + Space** to see a list of parameter hints.

TIP #4 - CAMEL CASE SUPPORT IN CODE COMPLETION

- Code completion supports camel case patterns. Entering **NPE** and clicking **Ctrl + Space** will propose **NullPointerException** and **NoPermissionException**

TIP #5 - CREATING GETTERS AND SETTERS

- Click on **Source** → **Generate Getter and Setter** to open the wizard that allows you to generate getter and setter methods.

TIP #6 - GENERATING HASHCODE() AND EQUALS() METHODS

- Click on **Source** → **Generate hashCode() and equals()** to generate these methods for a Java class

TIP #7 - ADDING CODE AROUND A BLOCK OF CODE

- Select a block of code and press **Alt+Shift+Z** to see a menu of items like **if** statement, **for** loop, **try/catch** etc that can enclose the selected block of code.

TIP #8 - LOCATING A MATCHING BRACKET

- Select an opening or closing bracket and press **Ctrl+Shift+P** to find its matching bracket.

TIP #9 - SMART JAVADOC

- Type **'/**'** and press **Enter** to automatically add a Javadoc comment stub.

TIP #10 - ORGANIZAING IMPORTS

- Press Ctrl+Shift+O to organize all the imports.

TIP #11 - ACTIVATING THE MENU BAR

- Press F10 to activate the Menu bar

TIP #12 - MAKING A VIEW/EDITOR ACTIVE

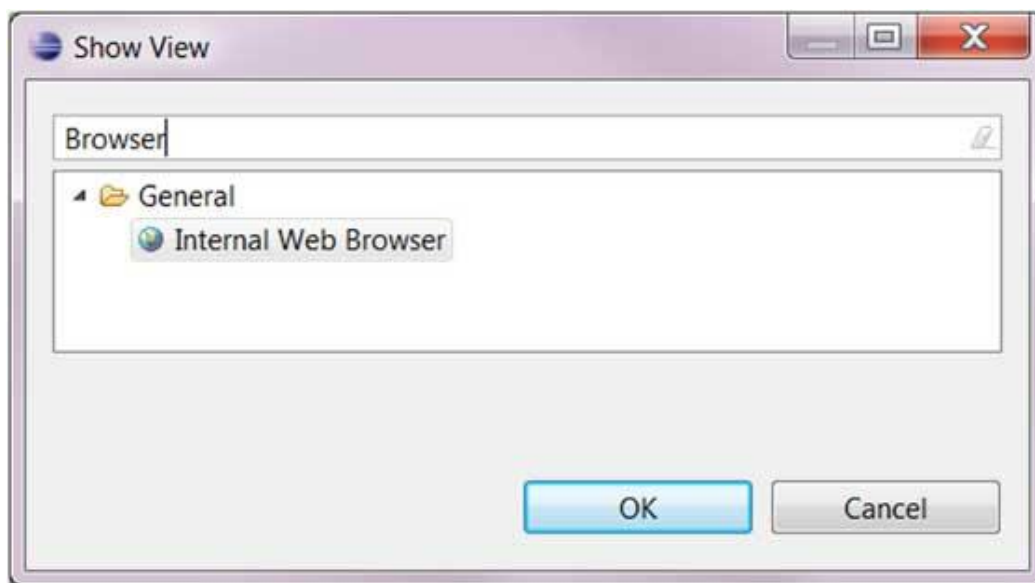
- Press Ctrl+F7 to see a list of open views and editor area and switch to one of them.

36. WEB BROWSERS

Internal Web Browser

The Internal Web Browser view allows you to surf the internet from within the Eclipse Workbench Window. To activate the Internal Web Browser view, click on the Window and select Show View → Other.

In the filter text box of the Show View dialog box enter "browser". From the tree select "Internal Web Browser" and click **OK**.



To navigate to a web page, in the URL text box of the Internal Web Browser, enter a URL of required web page.

