**Eclipse open source tool**

**Introduction**:

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.

**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 PHP eclipse 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.

## 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.

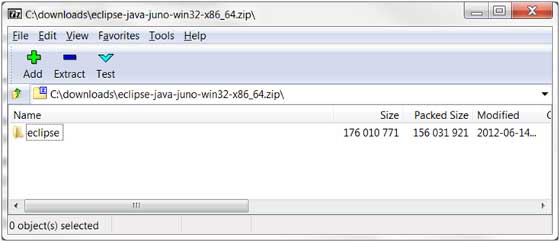
The drop down box in the right corner of the download page allows you to set the operating system on which eclipse is to be installed. You can choose between Windows, Linux and Mac. Eclipse is packaged as a zip file.

## 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](http://www.7-zip.org/)
* [PeaZip](http://sourceforge.net/projects/peazip/)
* [IZArc](http://www.izarc.org/)

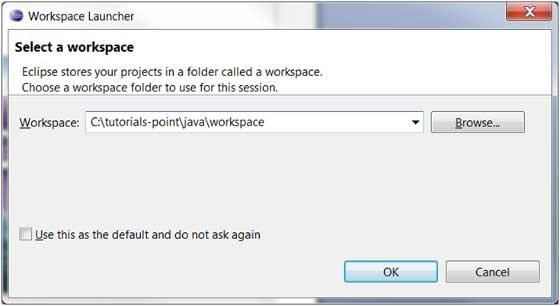
Using any one of these tools, extract the contents of the eclipse zip file to any folder of your choice.



## 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.



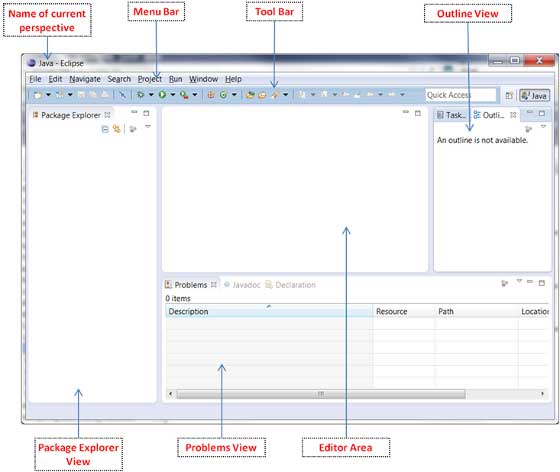
If you select "Use this as the default and do not ask again", this dialog box will not come up again. You can change this preference using the Workspaces Preference Page. See the [Preference tutorial](https://www.tutorialspoint.com/eclipse/eclipse_preferences.htm) page for more details.

**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**

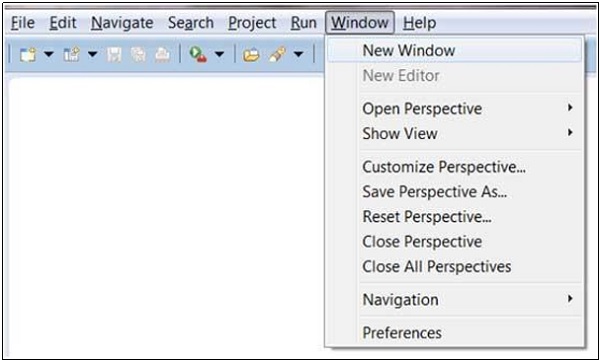
Multiple Eclipse Windows can be open at the same time. To open a new window, click on the Windows menu and select the New Window menu item.

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.

**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**

|  |  |
| --- | --- |
| **Sr.No** | **Menu Name & Description** |
| 1 | **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. |
| 2 | **Edit**  The Edit menu presents items like copy & paste. |
| 3 | **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. |
| 4 | **Navigate**  The Navigate menu allows you to quickly locate resources and navigate to them. |
| 5 | **Search**  The Search menu presents items that allow you to search the workspace for files that contain specific data. |
| 6 | **Project**  The menu items related to building a project can be found on the Project menu. |
| 7 | **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. |
| 8 | **Window**  The Window menu allows you to open and close views and perspectives. It also allows you to bring up the Preferences dialog. |
| 9 | **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. |

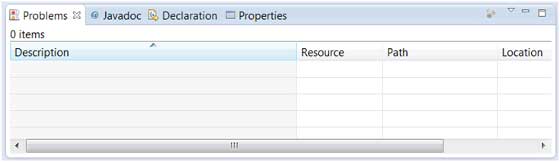
**About Views**

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.

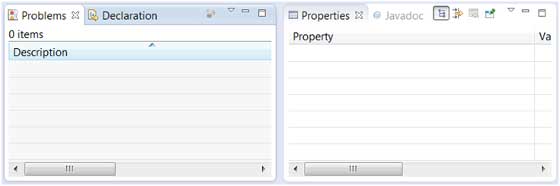
An eclipse perspective can show any number of views and editors. All editor instances appear in a single editor area, whereas views are placed inside view folders. A workbench window can display any number of view folders. Each view folder can display one or more views.

**Organizing Views**

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

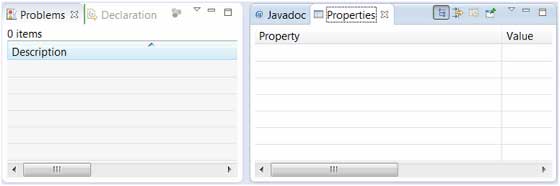


The picture given below shows the same four views arranged in two view folders.



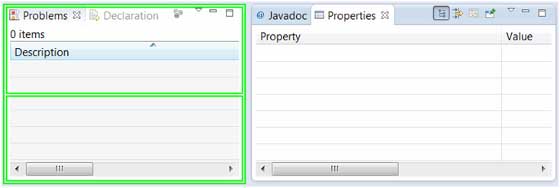
**Moving Views**

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**

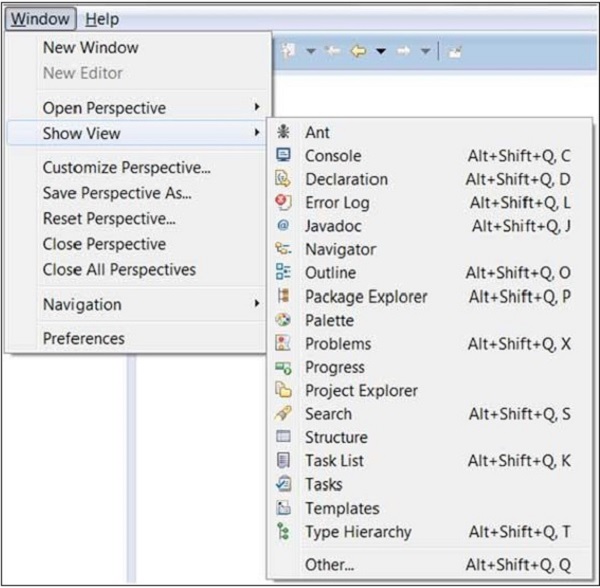
View folders can be dynamically created by dragging the title bar of a view to anywhere outside the editor area and title bar of another view folder. As you drag the title bar around, green lines will indicate where exactly the new view folder will be created.



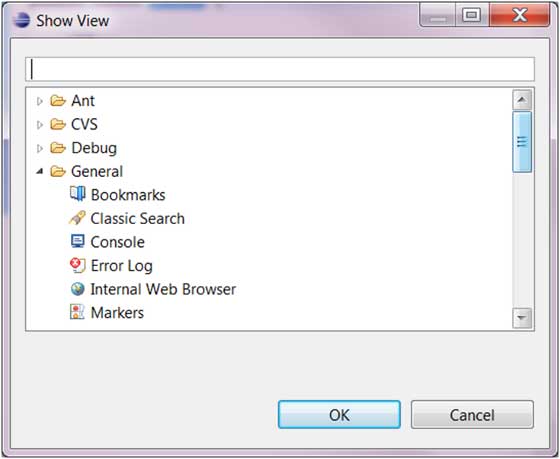
Moving the drag icon to the bottom of a window allows you to create a view folder that spans the entire width of the window. Moving the drag icon to the left or right edge of window allows you to create a view folder that spans the entire height of the window.

**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.

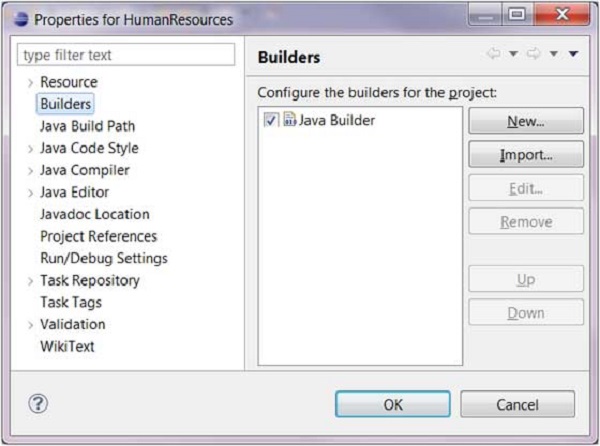


The views are organized by category. To quickly locate a view just type the name of a view into the filter text box. To open a view, select it and click on the OK button. The subsequent pages of this tutorial introduce you to a number of useful views.

**Building a Java Project**

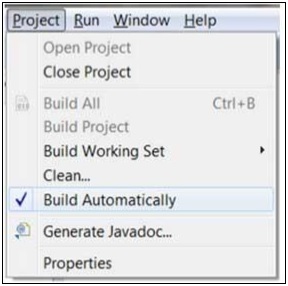
A project can have zero or more builders associated with it. A java project is associated with a java builder. To see the builders associated with a project −

* 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 click 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.

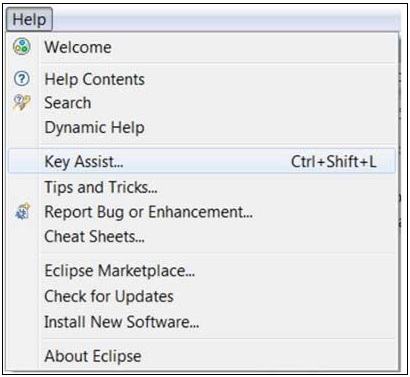


If automatic compilation is disabled then you can explicitly build a project by selecting the Build Project menu item on the Project menu. The Build Project menu item is disabled if the Build Automatically menu item is selected.

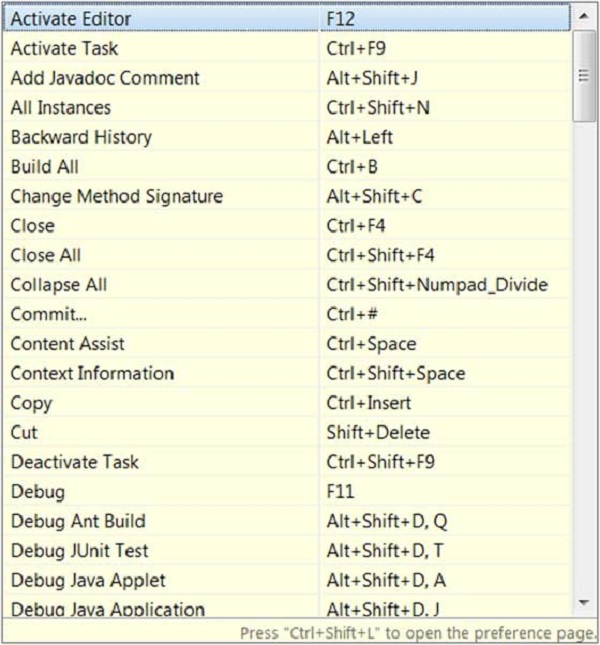
**About Shortcuts**

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.

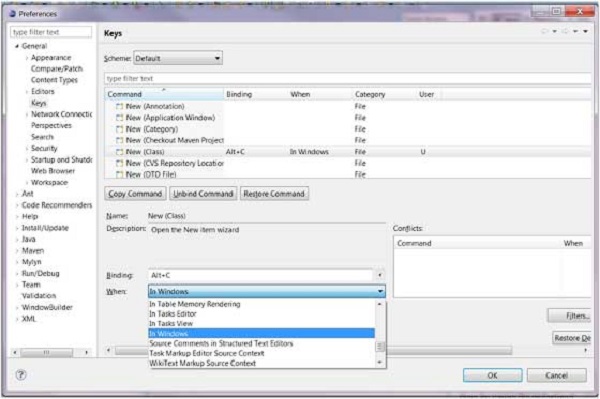


The Key Assist action which can be invoked by clicking Ctrl + Shift + L shows us all the accelerators or shortcut keys available in Eclipse.



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.

## Running a Java Program

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

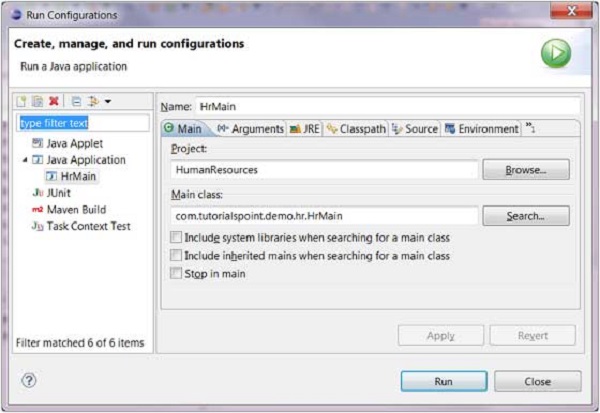
In the Package Explorer view −

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

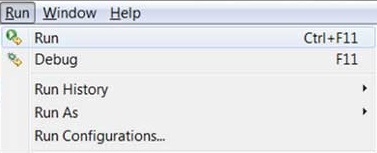
The same action can be performed using the Package Explorer view by selecting the class that contains the main method and clicking Alt + Shift + X, J.

Either actions mentioned above create a new [Run Configuration](https://www.tutorialspoint.com/eclipse/eclipse_run_configuration.htm) 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.