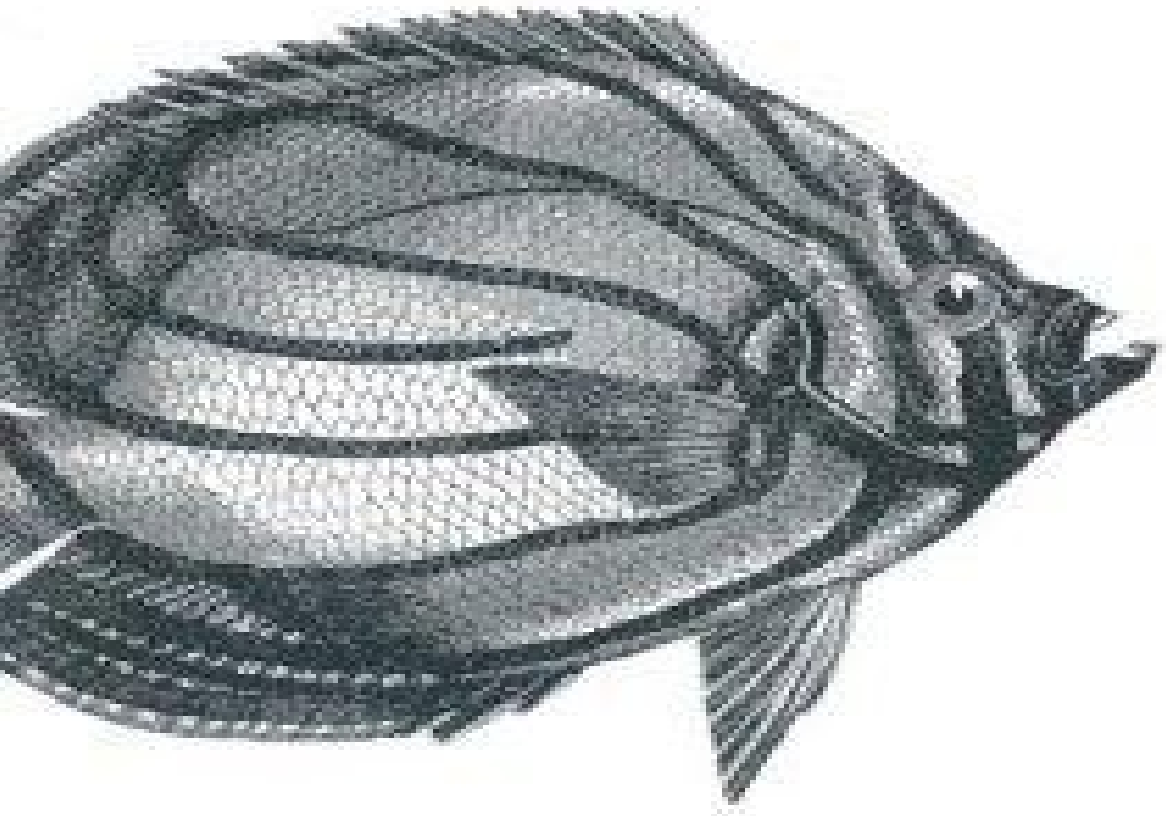
Using the Full-Featured IDE

Eclipse IDE

Pocket Guide



O'REILLY®

Ed Burnette

Eclipse IDE Pocket Guide
Table of Contents
Chapter 1. Introduction
1.1. What Is Eclipse?
1.2. Conventions Used in This Book
1.3. System Requirements
1.4. Downloading Eclipse
1.5. Installing Eclipse
1.6. Exploring Eclipse
1.7. Getting Upgrades
1.8. Moving On
Chapter 2. Workbench 101
2.1. Views
2.2. Editors
2.3. Menus
2.4. Toolbars and Coolbars
2.5. Perspectives
2.6. Rearranging Views and Editors
2.7. Maximizing and Minimizing
Chapter 3. Java Done Quick
3.1. Creating a Project
3.2. Creating a Package
3.3. Creating a Class
3.4. Entering Code
3.5. Running the Program
Chapter 4. Debugging
4.1. Running the Debugger
4.2. Setting Breakpoints
4.3. Single Stepping
4.4. Looking at Variables
4.5. Changing Code on the Fly
Chapter 5. Unit Testing with JUnit
5.1. A Simple Factorial Demo
5.2. Creating Test Cases
5.3. Running Tests
5.4. Test First
Chapter 6. Tips and Tricks
6.1. Code Assist
6.2. Templates

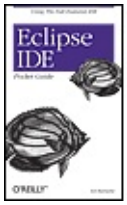
6.3. Automatic Typing
6.4. Refactoring
6.5. Hover Help
6.6. Hyperlinks
6.7. Quick Fixes
6.8. Searching
6.9. Scrapbook Pages
6.10. Java Build Path
6.11. Launch Configurations
Chapter 7. Views
7.1. Breakpoints View
7.2. Console View
7.3. Debug View
7.4. Declaration View
7.5. Display View
7.6. Error Log View
7.7. Expressions View
7.8. Hierarchy View
7.9. Javadoc View
7.10. JUnit View
7.11. Navigator View
7.12. Outline View
7.13. Package Explorer View
7.14. Problems View
7.15. Search View
7.16. Tasks View
7.17. Variables View
Chapter 8. Short Takes
8.1. CVS
8.2. Ant
8.3. Web Tools Platform
8.4. Testing and Performance
8.5. Visual Editor
8.6. C/C++ Development
8.7. AspectJ
8.8. Plug-in Development
8.9. Rich Client Platform
8.10. Standard Widget Toolkit
Chapter 9. Help and Community
9.1. Online Help

9.2. Eclipse Web Site
9.3. Community Web Sites
9.4. Reporting Bugs
9.5. Newsgroups
9.6. Mailing Lists
9.7. Conclusion
Appendix A. Commands
A.1. Edit Commands
A.2. File Commands
A.3. Help Commands
A.4. Navigate Commands
A.5. Perspective Commands
A.6. Project Commands
A.7. Refactor Commands
A.8. Run/Debug Commands
A.9. Search Commands
A.10. Source Commands
A.11. Text-Editing Commands
A.12. View Commands
A.13. Window Commands
Index
index_A
index_B
index_C
index_D
index_E
index_F
index_G
index_H
index_I
index_J
index_K
index_L
index_M
index_N
index_O
index_P
index_Q
index_R
index_S

index_T
index_U
index_V
index_W

Eclipse IDE Pocket Guide

By Ed Burnette



Publisher: **O'Reilly**
Pub Date: **August 2005**
ISBN: **0-596-10065-5**
Pages: **127**

Table of Contents | Index | Errata

Overview

Eclipse is the world's most popular IDE for Java development. And although there are plenty of large tomes that cover all the nooks and crannies of Eclipse, what you really need is a quick, handy guide to the features that are used over and over again in Java programming. You need answers to basic questions such as: Where was that menu? What does that command do again? And how can I set my classpath on a per-project basis?

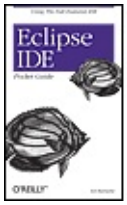
This practical pocket guide gets you up to speed quickly with Eclipse. It covers basic concepts, including Views and editors, as well as features that are not commonly understood, such as Perspectives and Launch Configurations. You'll learn how to write and debug your Java code--and how to integrate that code with tools such as Ant and JUnit. You'll also get a toolbox full of tips and tricks to handle common--and sometimes unexpected--tasks that you'll run across in your Java development cycle.

Additionally, the *Eclipse IDE Pocket Guide* has a thorough appendix detailing all of Eclipse's important views, menus, and commands.

The *Eclipse IDE Pocket Guide* is just the resource you need for using Eclipse, whether it's on a daily, weekly, or monthly basis. Put it in your back pocket, or just throw it in your backpack. With this guide in hand, you're ready to tackle the Eclipse programming environment.

Eclipse IDE Pocket Guide

By Ed Burnette



Publisher: **O'Reilly**
Pub Date: **August 2005**
ISBN: **0-596-10065-5**
Pages: **127**

Table of Contents | Index | Errata

Chapter 1. Introduction
- Section 1.1. What Is Eclipse?
- Section 1.2. Conventions Used in This Book
- Section 1.3. System Requirements
- Section 1.4. Downloading Eclipse
- Section 1.5. Installing Eclipse
- Section 1.6. Exploring Eclipse
- Section 1.7. Getting Upgrades
- Section 1.8. Moving On

Chapter 2. Workbench 101
- Section 2.1. Views
- Section 2.2. Editors
- Section 2.3. Menus
- Section 2.4. Toolbars and Coolbars
- Section 2.5. Perspectives
- Section 2.6. Rearranging Views and Editors
- Section 2.7. Maximizing and Minimizing

Chapter 3. Java Done Quick
- Section 3.1. Creating a Project
- Section 3.2. Creating a Package
- Section 3.3. Creating a Class
- Section 3.4. Entering Code
- Section 3.5. Running the Program

Chapter 4. Debugging
- Section 4.1. Running the Debugger
- Section 4.2. Setting Breakpoints
- Section 4.3. Single Stepping
- Section 4.4. Looking at Variables
- Section 4.5. Changing Code on the Fly

Chapter 5. Unit Testing with JUnit
- Section 5.1. A Simple Factorial Demo
- Section 5.2. Creating Test Cases
- Section 5.3. Running Tests
- Section 5.4. Test First

Chapter 6. Tips and Tricks
- Section 6.1. Code Assist
- Section 6.2. Templates
- Section 6.3. Automatic Typing

Section 6.4. Refactoring
Section 6.5. Hover Help
Section 6.6. Hyperlinks
Section 6.7. Quick Fixes
Section 6.8. Searching
Section 6.9. Scrapbook Pages
Section 6.10. Java Build Path
Section 6.11. Launch Configurations
Chapter 7. Views
Section 7.1. Breakpoints View
Section 7.2. Console View
Section 7.3. Debug View
Section 7.4. Declaration View
Section 7.5. Display View
Section 7.6. Error Log View
Section 7.7. Expressions View
Section 7.8. Hierarchy View
Section 7.9. Javadoc View
Section 7.10. JUnit View
Section 7.11. Navigator View
Section 7.12. Outline View
Section 7.13. Package Explorer View
Section 7.14. Problems View
Section 7.15. Search View
Section 7.16. Tasks View
Section 7.17. Variables View
Chapter 8. Short Takes
Section 8.1. CVS
Section 8.2. Ant
Section 8.3. Web Tools Platform
Section 8.4. Testing and Performance
Section 8.5. Visual Editor
Section 8.6. C/C++ Development
Section 8.7. AspectJ
Section 8.8. Plug-in Development
Section 8.9. Rich Client Platform
Section 8.10. Standard Widget Toolkit
Chapter 9. Help and Community
Section 9.1. Online Help
Section 9.2. Eclipse Web Site
Section 9.3. Community Web Sites
Section 9.4. Reporting Bugs
Section 9.5. Newsgroups
Section 9.6. Mailing Lists
Section 9.7. Conclusion
Appendix A. Commands
Section A.1. Edit Commands
Section A.2. File Commands
Section A.3. Help Commands
Section A.4. Navigate Commands
Section A.5. Perspective Commands
Section A.6. Project Commands
Section A.7. Refactor Commands
Section A.8. Run/Debug Commands

Section A.9. Search Commands
Section A.10. Source Commands
Section A.11. Text-Editing Commands
Section A.12. View Commands
Section A.13. Window Commands
Index

Chapter 1. Introduction

Welcome to the pocket guide for the Eclipse Integrated Development Environment. This book is the ultimate "no fluff" user's manual for the Eclipse IDE, in particular, its Java Development Toolkit (JDT). This book is designed to get you up and running quickly in the environment even if you've never used Eclipse before. Some Java™ programming knowledge will be helpful when reading this guide, but even if you're new to Java, you can still find a good deal of useful information within these pages. Let's begin with an overview of what Eclipse is and how to download and install it. If you're already using Eclipse, you can skip this section and jump to Part II.

1.1. What Is Eclipse?

Eclipse is an IDE for "anything, and nothing at all," meaning that it can be used to develop software i any language, not just Java. It started as a proprietary replacement for Visual Age for Java from IBM but was open sourced in November 2001. Eclipse is now controlled by an independent nonprofit organization called the Eclipse Foundation. Since 2001, it has been downloaded over 50 million time it is now being used by thousands of developers worldwide. It also has a sizable following in the university community, where it is used in classes on programming and object-oriented design.

1.2. Conventions Used in This Book

Italic

Used for filenames, directory names, URLs, and tools from Unix such as *vi*. Also used for emphasis and to introduce new terms.

`Constant width`

Used for names of Java packages, methods, etc.; commands; variables; and code excerpts.

`Constant width bold`

Used for keywords within code examples and for text that the user should type literally.

1.3. System Requirements

Eclipse runs on today's most popular operating systems, including Windows XP, Linux, and Mac OS X. It requires Java to run, so if you don't already have Java installed on your machine, you must first install a recent version. You can download Java for Windows and Linux from http://java.sun.com; look for the J2SE SDK (Software Development Kit) package without a NetBeans™ bundle. Mac OS has Java preinstalled. See Table 1 for the minimum and recommended system requirements.

Table 1-1. System requirements for Eclipse

Requirement	Minimum	Recommended
Java version	1.4.0	5.0 or greater
Memory	512 MB	1 GB or more
Free disk space	300 MB	1 GB or more
Processor speed	800 Mhz	1.5 Ghz or faster

In order to unpack Eclipse's download package, you will need a standard archive program. Some versions of Windows have one built in; for other versions, you can use a program such as WinZip (http://www.winzip.com). The other platforms come with an archive program preinstalled.

Tip: In the interests of space and simplicity, the rest of this book will focus on the Windows version of Eclipse. Other platforms will be very similar, although you may notice slight platform-specific differences.

1.4. Downloading Eclipse

To download the Eclipse IDE, go to http://www.eclipse.org. Click on "downloads" and then select the most recent stable or release version of the Eclipse SDK for your platform. If prompted for a mirror site, pick the one located closest to you. If that one is slow or unavailable, simply return to the download page and try a different mirror, or try the main site.

Tip: You may see other download packages such as Runtime, JDT, and RCP on the download page. You don't need those. Ju get the one package called Eclipse SDK.

1.5. Installing Eclipse

First, install Java if you haven't already. Then download the Eclipse SDK to a temporary directory. Use your archive program to unpack Eclipse into a permanent directory. There are no setup programs and no registry values to deal with.

After you have unpacked the SDK, you should have a subdirectory called *eclipse*, which in turn has directories in it such as *plugins* and *features*. If you don't see these, check the settings on your archiv program. A common mistake is to unpack Eclipse in such a way that its directory structure is not preserved. Eclipse won't run unless you unpack it with the exact directory paths that exist in the archive.

1.5.1. 3, 2, 1, Launch!

You are now ready to launch Eclipse. Inside the *eclipse* directory, you'll find a launcher program for the IDE called, strangely enough, *eclipse* (or *eclipse.exe*). Invoke that program to bring up the IDE.

Tip: On Windows, you may find it convenient to create a desktop shortcut to launch Eclipse.

1.5.2. Specify a Workspace

The first time you start Eclipse, you will be prompted for the location of your workspace. The workspace is the location where your source code and other files and settings will be stored on your workstation. Specify a permanent locationsomewhere *not* in your install directorypreferably a locatic that will be backed up regularly.

Putting the workspace in a different place from where you installed Eclipse makes upgrades easier. See the "Getting Upgrades" section, later in Part I, for more information.

1.6. Exploring Eclipse

When Eclipse starts up, you will be greeted with the Welcome screen (see Figure 1). This screen provides an introduction for new users who don't have the benefit of a pocket guide to Eclipse; for now you can skip over it by closing the Welcome view (click on the close iconthe x next to the word "Welcome"). You can always come back to the Welcome screen later by selecting Welcome from the Help menu.

Figure 1-1. The Welcome screen allows you to explore introductory material, includin examples and tutorials.

1.7. Getting Upgrades

Eclipse includes an automatic update facility that can handle *point releases* (i.e., bug-fix versions) without any work on your part. For example, Eclipse would install an upgrade from 3.1.0 to 3.1.1 automatically. However, for anything more substantial, the best practice is to do a manual clean install.

Tip: A clean install is especially important if you want to use beta versions of Eclipse (called *Stable* or *Milestone builds* on the download page). Milestone builds are sometimes buggy, so you may need to temporarily go back and run your previous versio

---

For example, let's say you have been running Version 3.1 for a while and now Version 3.2 has come out. You want to upgrade right away because each new release contains a number of important bug fixes and useful new features. Also, if you have a problem with an older release and report it to the developers, they will simply ask you to upgrade (see "Reporting Bugs" in Part IX). So, you should upgrade, but what's the best way to do it?

First, rename your *eclipse* directory to something else, like *eclipse3.1*. Then download the new SDK package and install it normally, as if you had never installed Eclipse before. This is called a *clean install* because you are not attempting to mix new and old code together. Note that your workspace doesn't need to change at all, but you should back it up before running the new version just in case. Now do you see why I recommended you don't keep your workspace in the install directory?

Tip: Any additional plug-ins you have installed for Eclipse will need to be reinstalled at this point unless you keep them in an *extension location* separate from the Eclipse SDK.

1.8. Moving On

Congratulationsyou've successfully downloaded, installed, and started exploring Eclipse. In Part II, you'll learn what all the windows and buttons are for and how to set up the environment just the way you like it. If you want to skip ahead and start writing a Java program, jump to Part III.

Chapter 2. Workbench 101

Eclipse's main window, called the *workbench*, is built with a few common user interface elements (se
Figure 2). Learn how to use them and you can get the most out of the IDE. The two most important
elements are views and editors. If you're already familiar with the Eclipse workbench, you can skim
this section or skip to Part III to start programming.

Figure 2-1. The Eclipse workbench is made up of views, editors, and other elements.

2.1. Views

A *view* is a window that lets you examine something, such as a list of files in your project. Eclipse comes with dozens of different views; see Table 2 for a partial list. These views are covered in more detail in Part VII.

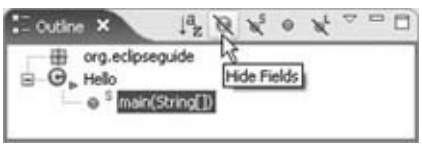
Table 2-1. Commonly used Eclipse views

View name	Description
Package Explorer	Shows all your projects, Java packages, and files.
Hierarchy	Displays the class and interface relationships for the selected object.
Outline	Displays the structure of the currently open file.
Problems	Shows compiler errors and warnings in your code.
Console	Displays the output of your program.
Javadoc	Shows the description (from comments) of the selected object.
Declaration	Shows the source code where the selected object is declared.

To open a view, select Window ⟶ Show View. The most commonly used views are listed in that menu. To see the full list, select Other….

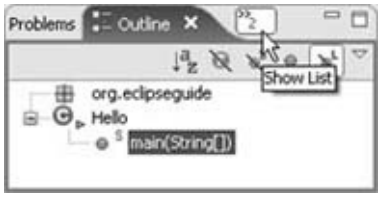
Most views have a titlebar that includes the icon and name for the view, a close icon, a toolbar, and a area for the content (see Figure 3 for an example showing the Outline view). Note that if the view is too narrow, the toolbar will be pushed to the next line. To discover what all the buttons do, move you mouse over a button, and a little window called a *tool tip* will appear that describes the item.

Figure 2-2. Views usually have titles, toolbars, and a content area. Let the mouse pointer hover over an item to bring up a description.



Multiple views can be stacked together in the same rectangular area. The titlebar will show a tab for each view, but only one view can be active at a time. Click on a tab to bring its view to the front. If th window is too narrow to show all the titles, a *chevron menu* will appear (see Figure 4; the number below the >> shows how many views are hidden). Click on the chevron menu to list the hidden views

Figure 2-3. Views can be stacked on top of one another. If space is short, some may b hidden in a chevron menu.

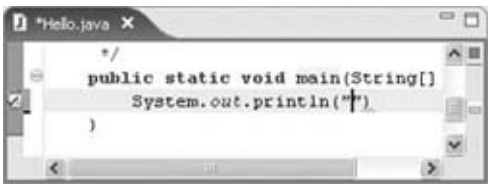


2.2. Editors

An *editor* in Eclipse is just like any other editorit lets you modify and save files. What sets editors in Eclipse apart is their built-in language-specific knowledge. In particular, the Java editor completely understands Java syntax; as you type, the editor can provide assistance such as underlining syntax errors and suggesting valid method and variable names (see Figure 5). Most of your time will be spe in the Java editor, but there are also editors for text, properties, and other types of files.

Editors share many characteristics with views. But unlike views, editors don't have toolbars, and you will usually have more than one of the same type of editor open (for example, several Java editors). Also, you can save or revert an editor's contents, but not a view's. An asterisk in the editor's titlebar indicates that the editor has unsaved data. Select File ⟶ Save or press Ctrl+S to write your changes disk.

Figure 2-4. The Java editor provides typing assistance and immediate error detection

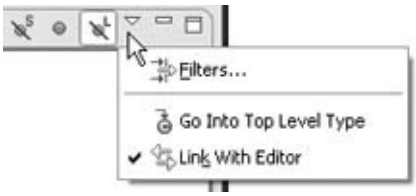


2.3. Menus

Eclipse is filled with menus, yet it's not always obvious how to access them. So, let's take a quick tou
The most prominent one is the *main menu* across the top of the Eclipse window. Click on a menu iten
to activate it or press Alt and the shortcut key for the menu (for example Alt+F for the File menu).

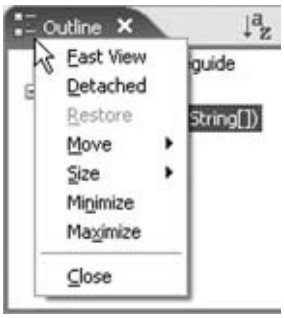
Some views have *view menus* that open when you click on the downward-pointing triangle icon near
the upper right of the view (see Figure 6 for an example).

Figure 2-5. If you see a triangle in the toolbar, click on it for more options.



Another menu is hidden in the titlebar under the icon to the left of the title. Right-click on the icon to
access the *system menu*; this allows you to close the view or editor, move it around, and so forth. The
system menu is shown in Figure 7.

Figure 2-6. Right-click on the icon to the left of the title to get the system menu.

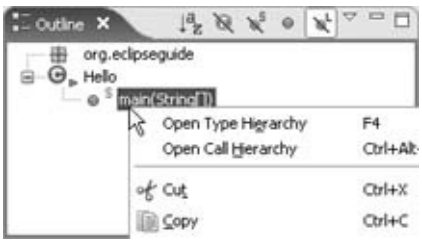


Tip: Most commands in Eclipse can be performed in several different ways. For example, to close a view you can either use th
system menu or click on the close icon. Use whichever way is most convenient for you.

Finally, you can right-click on any item in the content area to bring up the *context menu* (see Figure
8). Notice the keyboard shortcuts listed to the right of the menu description. These shortcuts can be
used instead of the menu to execute a particular command. For example, instead of right-clicking on
`main` and selecting Open Type Hierarchy, you can just select `main` and press the F4 key.

Tip: Starting in Eclipse 3.1, you can press Ctrl+Shift+L to see a list of the current key definitions. To change them, go to Windo ⟶ Preferences ⟶ General ⟶ Keys. By using key definitions and shortcuts, you can work in Eclipse without touching the mouse at all.

Figure 2-7. Right-click in the content area for the context menu.



sample content of Eclipse IDE Pocket Guide

- click Divine (House of Oak, Book 2) pdf
- download MaddAddam (MaddAddam Trilogy, Book 3) online
- read Brazil's Dance with the Devil: The World Cup, The Olympics, and the Fight for Democracy
- click Profane Culture for free
- Best of: Egon Schiele book
- read online Writer to Writer: From Think to Ink here

- http://ramazotti.ru/library/Mini-Farming-Guide-to-Vegetable-Gardening--Self-Sufficiency-from-Asparagus-to-Zucchini.pdf
- http://www.netc-bd.com/ebooks/MaddAddam--MaddAddam-Trilogy--Book-3-.pdf
- http://www.netc-bd.com/ebooks/Functional-Programming-Patterns-in-Scala-and-Clojure--Write-Lean-Programs-for-the-JVM.pdf
- http://test.markblaustein.com/library/The-Farm-as-Ecosystem--Tapping-Nature-s-Reservoir-----Biology--Geology--Diversity.pdf
- http://weddingcellist.com/lib/Dream-House.pdf
- http://monkeybubblemedia.com/lib/The-AMA-Handbook-of-Business-Writing--The-Ultimate-Guide-to-Style--Grammar--Punctuation--Usage--Construction--and-F