

Eclipse and Java: Using the Debugger

Companion Tutorial Document

Version 1.1 (February 17, 2008)

By Mark Dexter

Table of Contents

| | |
|---|---|
| Introduction & Setup..... | 1 |
| Playing the Lessons..... | 2 |
| Closed Captions and Lesson Table of Contents..... | 2 |
| Tutorial Target Audience..... | 2 |
| Tutorial Objectives..... | 2 |
| Tutorial Approach..... | 3 |
| Getting The Most From This Tutorial..... | 3 |
| Sample Java Application – Personal Lending Library..... | 3 |
| Downloading and Installing Eclipse..... | 4 |
| Lesson Outlines..... | 5 |
| Lesson 1 | 5 |
| Lesson 2 | 5 |
| Lesson 3..... | 5 |
| Lesson 4..... | 5 |
| Lesson 5..... | 6 |
| Lesson 6..... | 6 |
| Lesson 7..... | 6 |
| General Resources..... | 6 |
| Eclipse Websites..... | 6 |
| Java Websites..... | 6 |
| Eclipse Books..... | 7 |
| Java Books..... | 7 |
| Alphabetical Index by Lesson..... | 8 |

Introduction & Setup

This document is designed to accompany the “Eclipse And Java: Using the Debugger” video tutorial, which is available at <http://eclipsetutorial.sourceforge.net/>. Before starting this tutorial, please download the following files from the website:

- Debugger-Tutorial-Companion-Document.PDF – this document.
- debugger_tutorial_project.zip – zip file for importing the Eclipse “DebuggerTutorial” project. This is done as part of Lesson 1.
- debugger-lesson01.zip through debugger-lesson07.zip – zip files for the 7 lessons in the tutorial

Playing the Lessons

1. Unzip each lesson's zip file into a directory on your system.
2. Find the file called "lessonxx.html", where xx is the lesson number 01-07.
3. Open this file with your browser (e.g., Internet Explorer or Firefox) and press the large play button on the video thumbnail. The lesson should play inside the browser. Note that you need to have the Adobe Flash player installed on your system. This can be downloaded from Adobe at <http://www.adobe.com/products/flashplayer/>.

Alternatively, you can just open the file "lessonxx.swf" with your browser. In Windows Internet Explorer you might get a security warning and need to click and select "Allow blocked content". If you open the lessonxx.swf file directly (as opposed to lessonxx.html), you can resize the video to any desired size, making it larger or smaller. Note that playing the video from the lessonxx.html file will provide the clearest rendition of the video but does not allow resizing.

Closed Captions and Lesson Table of Contents



The audio track of each lesson can be accompanied by closed captions (subtitles), which display in the lower portion of the video. Press the "CC" button in the lower right corner to toggle them on and off. Each lesson also contains a table of contents that allows you to jump to a specified point inside the video. To access the table of contents, click on the Table of Contents icon in the lower right corner of the video. Next to the Table of Contents control is the Volume control. The controls are shown above.

Tutorial Target Audience

This tutorial is targeted for people who have complete the tutorial "Eclipse and Java for Total Beginners" or who have some familiarity with Eclipse and Java. Completing the "Introducing Persistence" tutorial is not necessary but might be helpful to beginning Eclipse and Java programmers. No prior experience with the Eclipse Java Debugger is needed for this tutorial.

Tutorial Objectives

The objectives of this tutorial are to:

- Demonstrate how to use the Eclipse Java debugger to step through a running Java program and examine the contents of the program's variables

Eclipse and Java: Using the Debugger Tutorial Companion Document

- Learn some cool tips and tricks about the Eclipse debugger
- Use the debugger to look at object reference issues and recursive methods in Java

Tutorial Approach

The tutorial is organized around the following activities. We will start with the MyLibrary project which was created in the “Total Beginners” and “Introducing Persistence” tutorials.

- Import the MyLibrary project
- Learn basic debugger skills: stepping through the code and examining variables and expressions
- Use the debugger to fix two programming errors in the MyLibrary class
- Use the debugger to look at some Java programming concepts

Concepts are introduced as needed during the debugging process.

Getting The Most From This Tutorial

This tutorial can be used as an in-depth demonstration of the Eclipse Java Debugger. However, if you want to learn how to use the debugger, the following approach is recommended:

- Have Eclipse installed and ready to go.
- Work side-by-side with the lessons, pausing and rewinding as needed.
- Use this guide as needed.
- Consult other resources as needed to understand the topics covered in greater depth.
- Keep a positive attitude!

Sample Java Application – Personal Lending Library

This is a small sample application to track our personal library of books. It includes a list of people to whom we loan our books. The application has four Java classes:

1. Person has fields for the person's name and a number which will be the maximum number of books this person can borrow at one time.
2. Book has fields for title, author, and the person who is currently borrowing the book.
3. MyLibrary contains a list of all of our books and a list of all of the people who might borrow them. It includes methods for checking books out to people and checking them back in. It also includes a main() method that simulates a user session. In the main() method, a new MyLibrary is created, books and people are created and added to it, and some books are checked out and check in. At various points, information about the MyLibrary object is printed to the system console.
4. MyUtilities contains methods to save and retrieve text files to and from disk as well as methods to save and retrieve a MyLibrary object to and from a file in serialized format. These methods

were developed in the “Introducing Persistence” tutorial. There are also two new methods added for this tutorial.

Note that this application is very similar to, but not the same, as the application we wrote in the Persistence tutorial. Here we do not use the XStream XML library that we used in the Persistence tutorial, and we have removed all of the methods that depended on this library. Also, this project by design contains some bugs that we will fix during the course of this tutorial.

Downloading and Installing Eclipse

Before Installing Eclipse, you need to have the Java JDK (Java development kit) installed on your computer. These are available at <http://java.sun.com/javase/downloads/index.jsp>. IMPORTANT NOTE: You need to install Java version 1.5 or later to run the tutorial. Version 1.6 (Java 6) is recommended, since this supports all of the latest Eclipse Java Debugger features.

Java versions prior to 1.5 do not support generics (used in the MyLibrary project) or the JUnit 4.0 unit test framework. Also, although you can run Eclipse with the JRE, we need the JDK for this tutorial, since we'll be using the debugger-enabled version of the JRE that comes with the JDK.

Installing the JDK is reasonably simple. Detailed, step-by-step instructions, if needed, are available in the PDF Eclipse Tutorial at the <https://www.arctechsoftware.com/tutorial/welcomePage.do>. (Follow the link to “Beginning Eclipse”.)

This tutorial is based on Eclipse 3.3. Here are the steps to install Eclipse 3.3 from www.eclipse.org:

- Navigate to www.eclipse.org/downloads
- Select “Eclipse IDE for Java Developers”. If your platform is Linux or Mac OSX, be sure to select the link to the right. Note that you can use “Eclipse IDE for Java EE Developers”, “Eclipse for RCP/Plug-in Developers”, or “Eclipse Classic” as well. All of these include the Java development portions of Eclipse used in this tutorial.
- On the www.eclipse.org/downloads page, follow the link “Find out more”. Scroll your browser to display the far right-hand side of the screen to the column “Tutorials and Help”. The first tutorial is a video tutorial that steps you through downloading and installing Eclipse on Windows.

The Eclipse installation is very straightforward. There is no installation program. Instead, you just create the top-level folder and then unzip the file inside this folder. In Windows XP, for example, just copy the zip file to your root directory (e.g., “C:\”) and then unzip the downloaded zip file.

(Note: You need to use an unzip program. Do not use the Windows File Explorer unzip.)

When you unzip the downloaded file, this will create a folder called “C:\eclipse”. The Eclipse programs will be created in several sub-folders (configuration, features, plugins, readme). The procedure for Linux is similar, except you unzip the “.tar.gz” file.

Another option is to install the EasyEclipse distribution of Eclipse (<http://www.easyeclipse.org>). This is a free third-party distribution that includes a Windows installation program that installs Java and Eclipse all in one step. You should install version 1.3 (based on Eclipse 3.3) or later of the Expert Java or Desktop Java distributions.

Lesson Outlines

Lesson 1

- Discuss Tutorial Target Audience
- Introduce Debugger concepts
- Create Java project in Eclipse using Import from archive file
- Run MyLibrary main method in debug mode
- Learn how to use Step Over, Step Into, and Step Return commands to step through program

Lesson 2

- Apply Step Filters
- Learn more about Step commands
- Use multiple breakpoints and Resume command
- Use Run to Line command
- Explore Stack Frames and Variables view
- Inspecting expressions

Lesson 3

- Use the Expressions view and Watch Expressions
- Use the Display view to create Watch Expressions
- Debug with JUnit tests
- Start debugging the MyLibrary class
- Select Expressions using short-cut keys

Lesson 4

- Use Exception Breakpoints
- Set Eclipse Debugger preferences
- Correct bugs with hot code replacement
- Use Drop To Frame to rewind a debug session
- Change the value of variables inside a debug session

Lesson 5

- Implement breakpoints options: conditional breakpoint, hit counts
- Set watchpoints (field breakpoints) and Class Prepare breakpoints
- Use the Stop at main (open debug run dialog)
- Use the Suspend command
- Use the Step into selection command

Lesson 6

- Debug into Java system classes (e.g., ArrayList)
- Use the debugger to illustrate Java object references and the String .equals() method

Lesson 7

- Use the debugger to understand a recursive method
- Use the Drop to Frame command with stack frames to rewind the debug session

General Resources

There are many general resources available for learning more about Eclipse and Java. These are just a few that I've found helpful.

Eclipse Websites

- www.eclipse.org/resources. This lists a number of articles, books, presentations, demonstrations and other resources on a variety of topics related to Eclipse.
- eclipse.newcomer newsgroup. This is a friendly, active newsgroup where newcomers to Eclipse can ask questions. The search feature of this and other newsgroups can be especially valuable, since there is a good chance that your question has already been asked and answered.
- Beginning Eclipse Tutorial on ArcTech Software LLC website. Written tutorial to help you install Eclipse and Java. Login required to download. It has a very good section on downloading and installing the Java JDK. Link to tutorial is <https://www.arctechsoftware.com/tutorial/tutorial.do?subcatId=1>. Link to home page is <https://www.arctechsoftware.com/tutorial/welcomePage.do>.

Java Websites

- The Java Tutorials from Sun (<http://java.sun.com/docs/books/tutorial/java/index.html>). This is the gold standard for learning Java, and it's free.
- JavaRanch Big Moose Saloon web site (<http://saloon.javaranch.com/cgi->

Eclipse and Java: Using the Debugger Tutorial Companion Document

bin/ubb/ultimatebb.cgi?category=1). This has a variety of forums, including Java in General (beginner), Java in General (intermediate), and many other Java topics. Very active and friendly, with knowledgeable moderators.

- The Java Developers Almanac 1.4 (<http://www.exampledepot.com/>). Contains Java code samples for many topics.

Eclipse Books

- The Java Developer's Guide to Eclipse, Second Edition, by Jim D'Anjou, Scott Fairbrother, Dan Kehn, John Kellerman, and Pat McCarthy. Clear, thorough reference for using Eclipse to write Java code. It covers Eclipse Version 3.0, which is still mostly applicable. It has a very good, concise section on the Eclipse debugger.

Java Books

- Head First Java, by Kathy Sierra & Bert Bates. Excellent, fun, creative book for Java and OOP beginners.
- Thinking In Java, by Bruce Eckel. Excellent, thorough reference for Java. For all levels of programmer.
- Effective Java, by Joshua Bloch. Concise book documents specific recommendations for Java best practices. For intermediate to advanced programmers.

Alphabetical Index by Lesson

| Topic | Lesson |
|--|--------|
| Add Watch Expression..... | 3 |
| Alt+Shift+Up to select expressions | 3 |
| alternate JRE in Eclipse, adding..... | 6 |
| breakpoint conditions..... | 5 |
| breakpoints – setting and toggling..... | 1 |
| Breakpoints View..... | 4 |
| caught and uncaught exceptions, suspending on..... | 4 |
| changing code in a running debug session..... | 4 |
| changing variable value during debug session..... | 4 |
| Class Prepare breakpoints..... | 5 |
| companion PDF document..... | 1 |
| conditional breakpoints..... | 5 |
| Convert To Watch Expression..... | 3 |
| CTRL-R – Run to Line..... | 3 |
| Ctrl+Alt+Left Click: Step Into Selection with hyperlink..... | 5 |
| Ctrl+F5: Step Into Selection keyboard shortcut..... | 5 |
| Ctrl+Shift+I = Move to Expressions View..... | 3 |
| Debug Perspective..... | 1 |
| Debug Preferences, Suspend on uncaught exceptions..... | 4 |
| Debug View..... | 1 |
| debugger – defined..... | 1 |
| debugger_tutorial_project.zip..... | 1 |
| Display View and code assist..... | 3 |
| Drop to Frame command with stack frames..... | 7 |
| Drop to Frame command, rewinding debug session with..... | 4 |
| Edit Step Filters..... | 2 |
| Edit View..... | 1 |
| exception breakpoints..... | 4 |
| Execute command: in Display View..... | 3 |
| Expression View..... | 3 |
| factorial() method..... | 7 |
| hit counts: using with breakpoints..... | 5 |
| hot code replace: changing code in a running debug session..... | 4 |
| Inspect command (Ctrl+Shift+I)..... | 2 |
| Java Exception Breakpoints..... | 4 |
| Java Runtime Engine (JRE) (also known as Java Virtual Machine or JVM)..... | 1 |
| Java system classes: step into..... | 6 |
| Java Virtual Machine (JVM) (also known as Java Runtime Engine or JRE)..... | 1 |
| JRE compiled with debug information, using..... | 6 |
| JUnit tests..... | 3 |

| Topic | Lesson |
|---|--------|
| keyboard shortcut: Ctrl+F5 = Step Into Selection command..... | 5 |
| keyboard shortcuts for Step Commands..... | 2 |
| keyboard shortcuts: Ctrl+R (Run to Line), Ctrl+Shift+I (Inspect)..... | 2 |
| line numbers – seeing in Java editor..... | 1 |
| MyLibrary classes..... | 1 |
| NullPointerException..... | 3 |
| object references..... | 6 |
| Open Declaration: with Ctrl+Left Click hyperlink or with F3..... | 5 |
| Pause command..... | 1 |
| recursive method, defined..... | 7 |
| Resume command..... | 1 |
| rewinding debug session with Drop to Frame command..... | 4 |
| Run / Debug as command..... | 1 |
| Run to Line command (Ctrl+R)..... | 2 |
| selecting expressions with Alt+Shift+Up..... | 3 |
| stack frames..... | 1, 2 |
| stack frames and Variables View..... | 2 |
| Step commands, keyboard shortcuts for | 2 |
| Step Filters..... | 2 |
| step into Java system classes..... | 6 |
| Step Into Selection command: with hyperlink / hover..... | 5 |
| Step Over, Step Into, Step Return commands..... | 1 |
| stop in main option..... | 5 |
| String equals() method, compared with “==” operator..... | 6 |
| Suspend command..... | 5 |
| Suspend on uncaught exceptions: in Debug Preferences..... | 4 |
| Terminate command..... | 1 |
| test-driven development (TDD)..... | 3 |
| uncaught exceptions and JUnit..... | 4 |
| Use Step Filters toggle button..... | 2 |
| Variables View..... | 1 |
| Variables View menus..... | 2 |
| Watch Expressions, adding with code assist..... | 3 |
| watching fields: field watchpoints..... | 5 |
| watchpoints..... | 5 |