



click2try™

click2try™ Tutorial

---

# Eclipse

Copyright © 2008-2009 Presage Technologies, LLC.

You may freely distribute or publish this content provided you publish the content in its entirety and include all existing copyright notices, attributions, links, and acknowledgements to [click2try](#)<sup>™</sup>, Presage Technologies, LLC. and to organizations mentioned explicitly or implicitly herein.

## Contents

<b>Introduction .....</b>	<b>5</b>
<b>Signing In and Running Eclipse.....</b>	<b>6</b>
<b>Building the HelloWorld Project .....</b>	<b>6</b>
<b>Defining an Eclipse Project .....</b>	<b>8</b>
<b>Creating a New Class .....</b>	<b>9</b>
<b>Adding a Print Statement.....</b>	<b>10</b>
<b>Compiling in Eclipse .....</b>	<b>11</b>
<b>Debugging in Eclipse.....</b>	<b>12</b>
<b>Resources .....</b>	<b>13</b>



## Introduction

Eclipse is a state-of-the-art framework aimed at helping software developers synthesize best-of-breed language compilers, components, debuggers, profilers, and other tools into the best possible development toolset on the market. Originally developed by IBM, Eclipse was spun off into an Open Source project and is currently maintained by the Eclipse Foundation, which is a not-for-profit, member-supported corporation. In short, Eclipse is a massive community effort to create a universal development workbench.

This tutorial will barely scratch the surface of Eclipse functionality, but should give you plenty to work with and think about, as you're evaluating Eclipse for your own use. Because the Eclipse universe is so large and well-populated, we encourage you to take a look at the Resources section for a list of links to some great tutorials and reference materials.

This tutorial focuses on the integrated development environment, or IDE. Although we'll be looking at Eclipse running on the CentOS platform, Eclipse runs on virtually all popular platforms.

In this tutorial, you'll learn about:

- Signing in and Running Eclipse
- Defining an Eclipse Project
- Writing a Sample Program
- Compiling in Eclipse
- Debugging in Eclipse

## Signing In and Running Eclipse

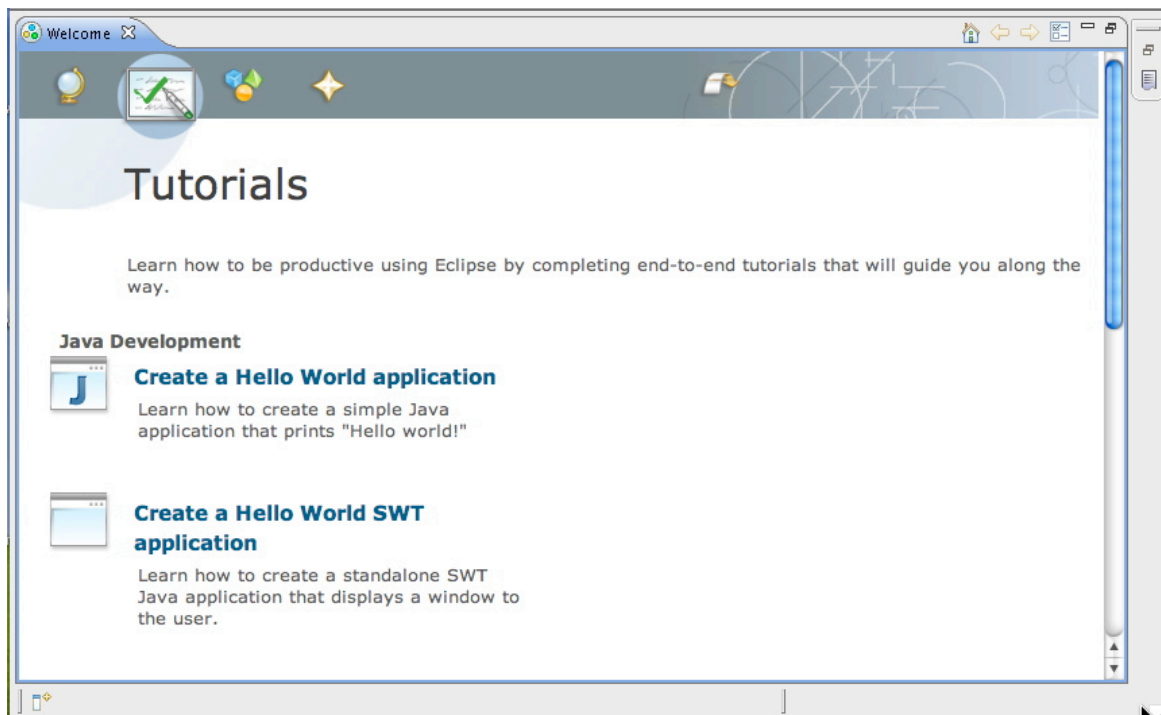
Once you sign in to the click2try™ Eclipse appliance, you'll be able to start using the Eclipse workbench.

To run Eclipse:

1. Start the click2try™ Eclipse appliance.
2. Click **OK** to accept the default workspace.

The first thing you'll see is the Eclipse Welcome page. On that page are several icons that will lead you to learning resources, documentation, and the workbench itself.

For this tutorial, we'll work through a combination of defining projects and the environment from scratch, and also walking through the Eclipse HelloWorld Tutorial. You can access this tutorial by clicking the Tutorials icon in the Welcome page.



## Building the HelloWorld Project

Let's take a quick walk through the HelloWorld project. Doing so will give you a solid grasp of Eclipse and the basic capabilities of the workbench.

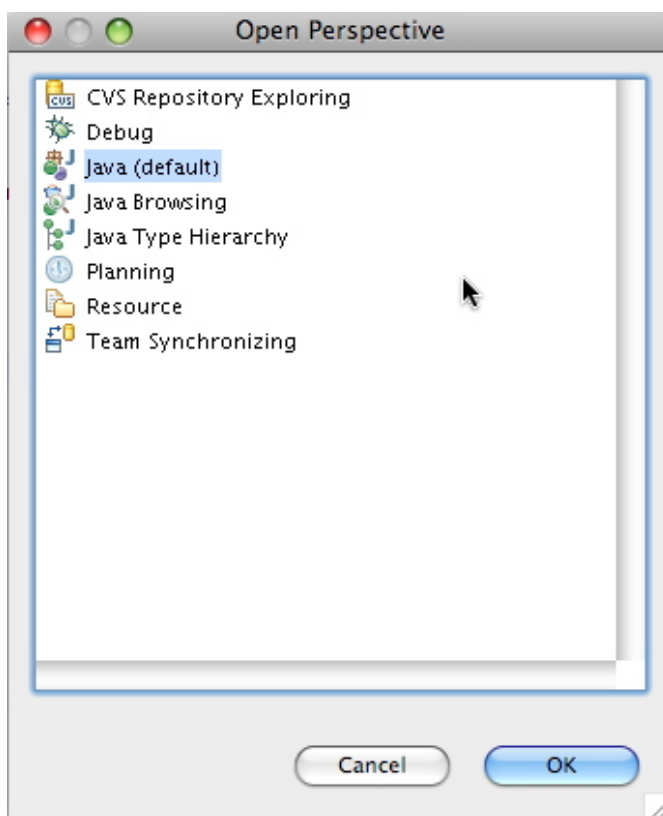
To start the project:

1. On the **Tutorials** page, click the **Create a Hello World application** link. Eclipse loads the necessary project files and opens the workbench.
2. Expand the **HelloWorld** project in the **Project Explorer** on the left side of the screen. You can see the many files that are included just to create a simple program that displays a single line of text.
3. Note the instructions on the right side of the screen. These are Tutorial steps you can follow. They will help you get a better feel for how to use Eclipse.

If you plan on using Eclipse to write code in different languages, you'll need to change the perspective each time to switch to another language. Perspective is a sort of mode. The perspective indicates to Eclipse that it should display certain language-specific features.

To change the perspective:

1. Select **Window > Open Perspective > Other**.
2. If it's not already selected, choose **Java (default)**.
3. Click **OK**.

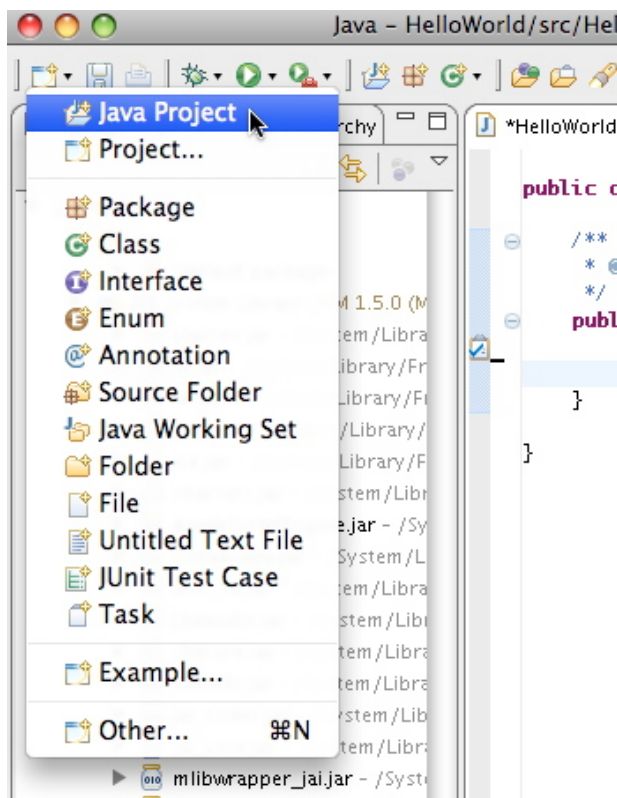


## Defining an Eclipse Project

Every application you write must be part of a project. A project is how Eclipse keeps track of all of the various class, resource, and configuration files that your application will use.

To define a project:

1. Click the down arrow next to the **New Project** icon on the main toolbar.
2. Select Java Project.



3. Enter a project name.
4. Click **Finish**.

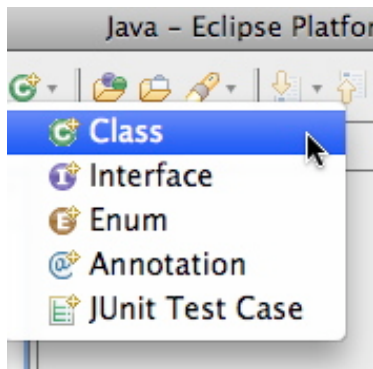
The project appears in the **Project Explorer** on the left side of the workbench.

## Creating a New Class

You'll need to create a new class. In Eclipse, this is incredibly easy.

To create a new class:

1. Select the **Class** button from the main menu. It's a green circle with a "C" in the middle.
2. Select **Class**.



3. Enter a name.
4. Select the **public static void main(String[ ] args)** option.
5. Click **Finish**.

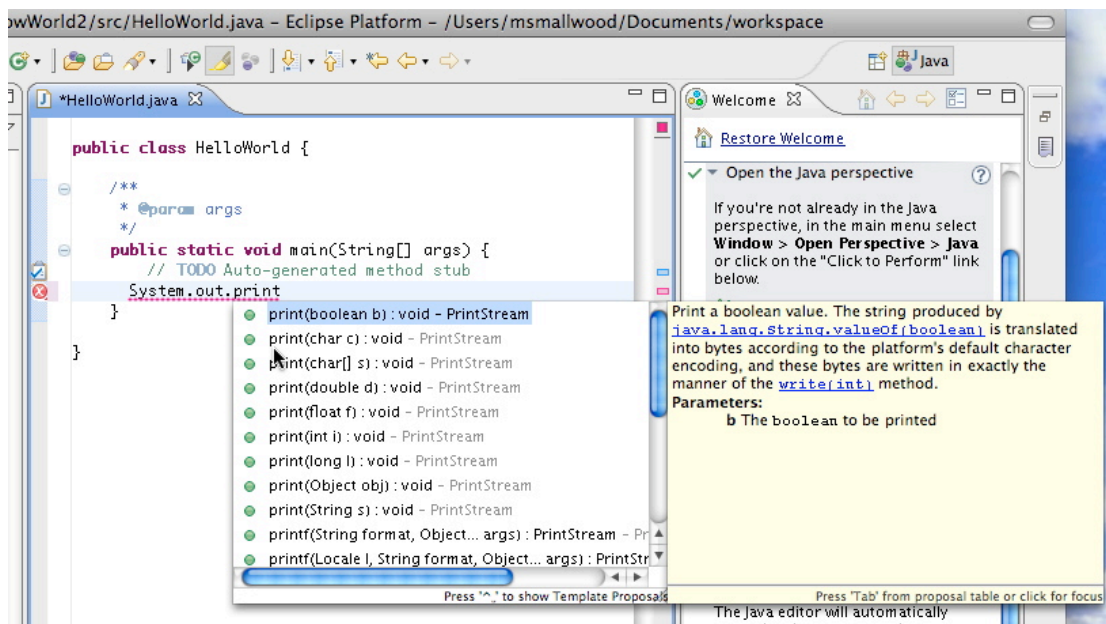
Eclipse builds the class in your workbench window, and adds the **main()** method for you.

## Adding a Print Statement

The Java *println* method tells the computer to send output to either the screen, or to a printer or other output device.

To add a print statement:

1. Click in the edit window to set the insertion pointer.
2. Type "System.out.println("Hello world!");" and don't forget the trailing semi-colon.
3. Notice how Eclipse provides statement completion help as you type:



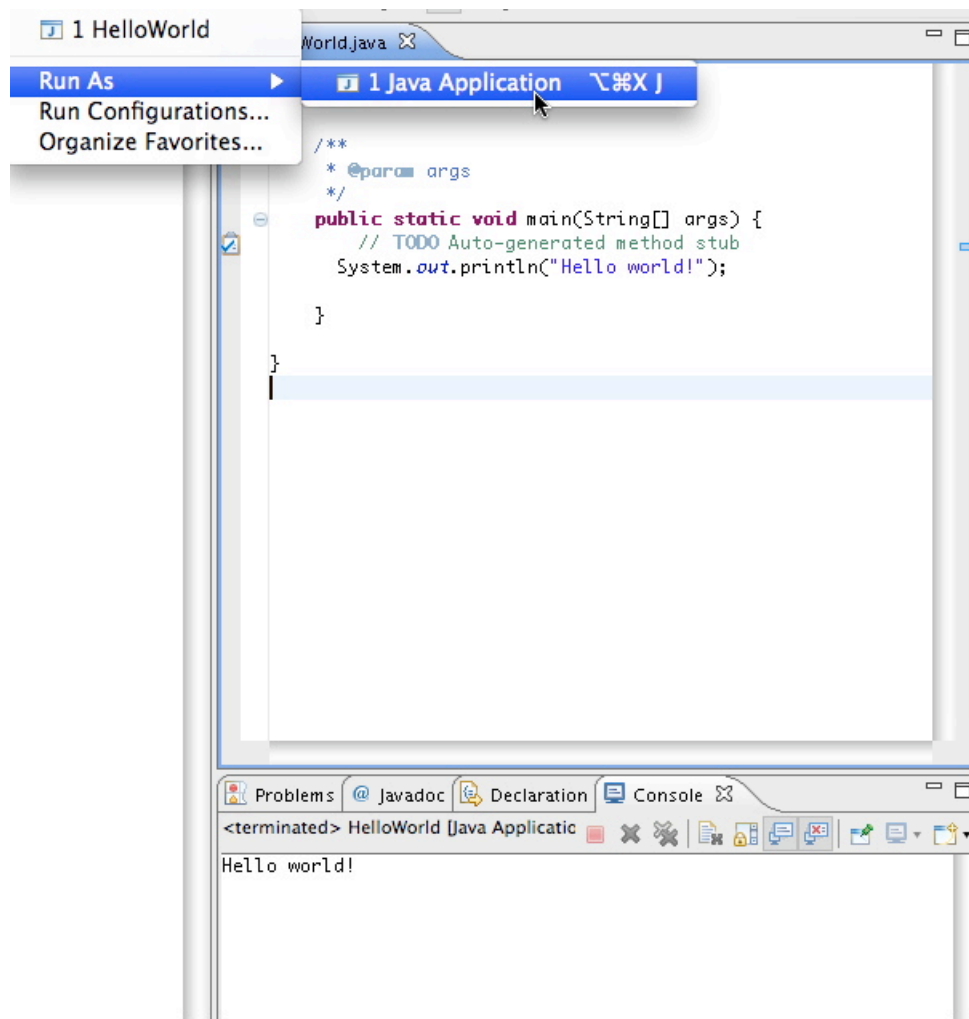
4. Select **File > Save**.

## Compiling in Eclipse

Compiling and running programs in Eclipse is a simple task. If you're writing Java apps, you can run the application in the Java Perspective, without specifying a separate compile step.

To compile:

1. Select the **Run** button, which is a green circle with a white arrow.
2. Select **Run As > Java Application**. If your application name is listed in the Run menu, you can just select it, instead.



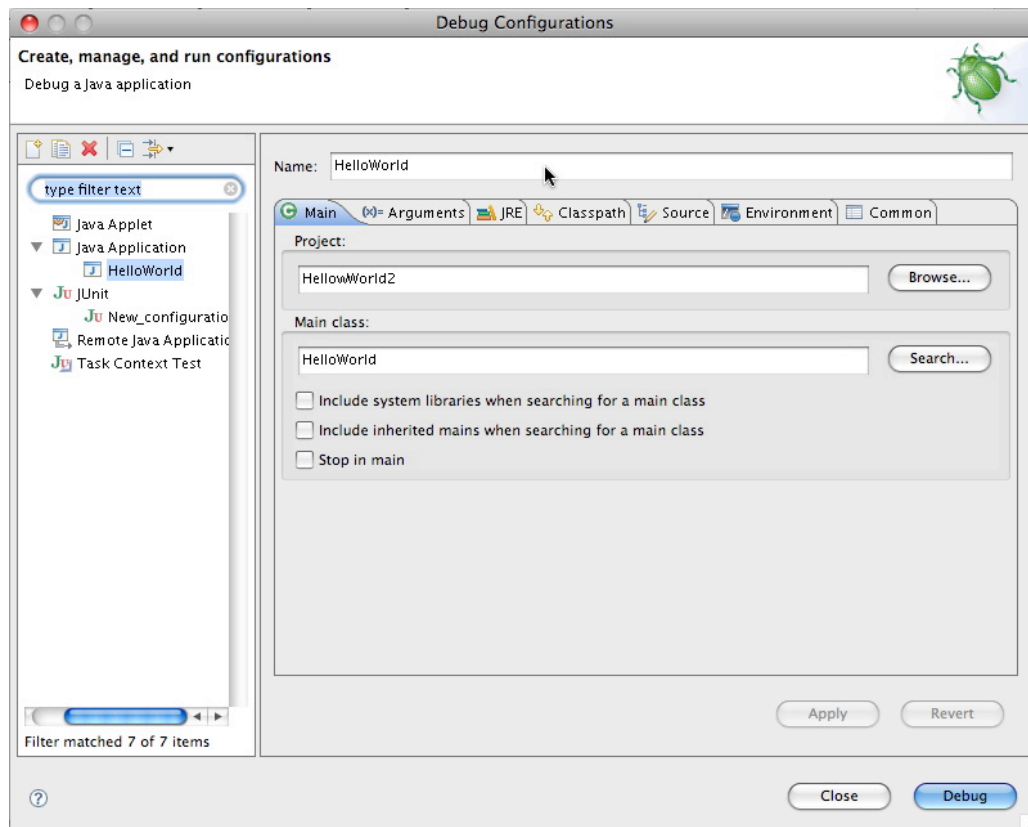
In the sample HelloWorld application, the output appears in the console window at the bottom of the page.

## Debugging in Eclipse

Debugging is a huge topic and is often performed with different techniques based on personal style or by accepted conventions within your own company. Eclipse provides built-in support for debugging. In the Java Perspective, you can use JUnit, a popular debugging framework that helps you build and run repeatable test cases.

To use the debugging tools:

1. Select the bug icon.
2. Select **Debug Configurations**. This selection displays the Debug Configuration dialog.
3. Create new configurations, set arguments, classpath, and other debug parameters.
4. Add breakpoints to your code.
5. Run your application by selecting the bug icon, and then **Debug As > Java Application**.



You can also include references to external programs in Eclipse. You can, for example, include an external debugger.

## Resources

Because the Eclipse project is so large, and its community so varied and active, you will find just about everything you need in the Eclipse community sites on the Web. Here are a few valuable links:

**Eclipse.org** <http://www.eclipse.org> - The main Eclipse portal.

### **Eclipse Tutorials**

<http://www.eclipse.org/resources/?sort=date&category=Tutorial> - List of detailed printed and video tutorials on a variety of important Eclipse topics.

**onEclipse** <http://www.oneclipse.com/products/essential-eclipse/> - A great site with an incredibly good video tutorial called *Essential Eclipse*.

**Eclipse Resources** <http://www.eclipse.org/resources/> - A large list of articles, books, and videos covering hundreds of Eclipse topics.

**Eclipse Portals** <http://www.eclipse.org/community/portals.php> - Extensive list of Eclipse sites, including platform-specific Eclipse sites, Eclipse plug-in sites, and bookmarks sites, just to name a few.

**Eclipse Live** <http://live.eclipse.org/> - Tons of great Eclipse videos, webinars, podcasts, and demos.