

# JMeterAndEclipse

Michal Kostrzewa (m.kostrzewa@pentacomp.com.pl) Special thanks to (text taken from)

- JMeter Wiki
- JMeter Mailing list
- Email me if your there is your text taken from wiki 😊
- Janka Kostrzewa for correcting this text.

## Revision history

- Mar 09 2004 - Update (Eclipse 3.0, Ant scripts)
- Nov 12 2003 - Update (Eclipse 2.1 + JMeter 1.9.1+)
- Dec 12 2002 - The first release

## Abstract

This document describes briefly how to develop the JMeter project with Eclipse IDE. Eclipse is a very powerful development environment with following features:

- It's fast! Although it's written in JAVA like many other development platforms, it doesn't use SWING, which is big and slow; it uses an especially created SWT widget toolkit instead. This toolkit doesn't emulate widgets in java, but it uses native system widgets (GTK on linux), which improves performance substantially.
- Very functional editor/package browser with syntax highlighting and syntax hints.
- Compiles java classes on the fly (when waiting for user input or saving a file), so one doesn't have to wait when launching an application. Furthermore, one sees errors immediately.
- Very nice debugger.
- Powerful code refactoring: when you want to change the name of a method or change parameters, don't hesitate to do it. Eclipse intelligently changes appropriate things regarding to java syntax (you won't get this result using simple search/replace).
- [JavaDoc](#) integration.
- CVS integration.
- Can automatically create getter and setter methods.
- CPL licence (OSI approved).
- Last but not least: search with regard to java syntax. You can just mark an identifier and click "open super implementation" or "find any reference to.." or "find declaration of...". It helps very much when exploring the internals of an unknown application.
- Many (many) more...

Eclipse is available for download for different platforms at <http://www.eclipse.org>

## Multiple Workspaces

Most Eclipse formatting settings are managed at the workspace level. If you work on other projects in addition to JMeter, you probably don't want the JMeter formatting to apply to your other projects. One way to manage this is to set up multiple Eclipse workspaces – one for JMeter, and one (or more) for other projects. To use multiple workspaces, simply start eclipse with the "-data" parameter:

```
eclipse -data c:\workspace_JMeter
```

You could also create a batch file, shell script, or Windows shortcut to automatically specify the workspace so you don't have to type it each time.

## Setting up JMeter + Eclipse

I assume you downloaded Eclipse IDE and have it working.

## Accessing CVS

1. Open the CVS perspective with and buttons 2. Invoke the popup menu and select "New|" "repository location" 3. Fill appropriate entries as pictured below, entering anoncvs as a password.: 4. You should see the apache cvs tree as pictured below. Unfold HEAD, select jakarta-JMeter and then select "Check Out as ..." from the pop-up menu 5. Select "JAVA project", and give a name for the project in next dialogs. I will assume I it is named jmeter-pserver for the rest of this document. 6. Wait for the cvs checkout to complete

## Setting up paths

After the checkout you'll see something like this:

It means you have the source paths improperly installed. To set them up, do the following:

1. Close this project 2. Copy the jmeter-pserver/eclipse.classpath file to jmeter-pserver/.classpath (you will have to overwrite the jmeter-pserver/.classpath file eclipse created for you 3. Open the project again 4. Now the source folders and jar libraries are mounted properly

NOTE: There are some optional libraries JMeter depends on. This setup defines exclusion filters for all the classes that need optional libraries (you would get compiler errors from Eclipse otherwise). Remove these exclusion filters if you have optional libraries installed. For further details refer to [jmeter-pserver/eclipse.readme](#))