

Eclipse IDE Installation Instructions

Table of contents

1 Audience.....	2
2 Purpose.....	2
3 Prerequisites.....	2
4 Installing Eclipse on a Windows Box.....	3
4.1 Obtain Eclipse.....	3
4.2 Install Eclipse.....	3
4.3 Familiarize yourself with the IDE.....	5
4.4 Additional configuration steps for all Eclipse users.....	5
4.5 Set up CVS or Subversion repositories.....	6
4.6 Additional configuration steps for Streek Java/J2EE developers.....	6

1. Audience

Individuals who wish to install the [Eclipse](#) IDE on their Windows workstations.

2. Purpose

To use the [Eclipse](#) IDE (Integrated Development Environment) to develop software.

Eclipse (www.eclipse.org) bills itself as a "universal tool platform." Complete with built-in CVS client, data access tools, webapp testing tools, and a rich Java development environment, coupled with a wealth of plug-ins for development in languages from C++ to Perl to Ruby and Ruby on Rails, it is easy to use, and the price can't be beat. The IDE is free and wholly extendable ... but be warned that it's also a memory-sink. Trying to run Eclipse on a modern Windows operating system with less than 1GB of RAM may keep your workstation very very busy writing memory pages to disk. At least 2GB of RAM are recommended to run this IDE, though 1GB may do depending on other concurrent usage of the developer's machine.

3. Prerequisites

These are the basics; if your machine isn't running these, please install or upgrade as appropriate:

- **Hardware:** a modern, well-equipped workstation; 1GB RAM is minimal; 2GB RAM is recommended where possible
- **Operating System:** Upgrade to latest **stable**, high-encryption releases of [Windows](#)
- **Web Browser:** It will be useful to have a current version of [FireFox](#). If you also use other browsers, they should also be upgraded to recent, stable versions.
- **Java:** Install J2SE (as of December 2006, Streek projects are developed using 1.5.0); if you're not sure how to install Java, you can refer to these [local instructions](#); or to [Sun's instructions](#). *[Note: Eclipse is a Java program, and requires a modern JRE (Java Runtime Environment) in which to run. Generally speaking, your best bet is to run a current version of Eclipse in a recently released JRE. In the instructions that follow, developers are recommended to install the Java Development Kit (JDK), which includes the Java Runtime Environment. If you are certain you will not be doing Java development, and have some other reason to install the JRE only, that is also an option; however, installation and usage HowTo documents on this site will assume that the JDK is installed on a development machine.]*
- **Public Key Login on repository host:** Exchange public keys with the CVS and/or Subversion server you will use from within Eclipse, to allow access to the repository (or repositories) unencumbered by authentication dialogs. To do so (or to test whether this is

already done), follow these [local instructions](#), applying them to your CVS or SVN server (e.g., `canvas.berkeley.edu` or `svn.berkeley.edu`)

4. Installing Eclipse on a Windows Box

Following these instructions in the order presented should get you started using the Eclipse IDE in short order.

4.1. Obtain Eclipse

Note that these instructions are written for installation of version 3.3.x (Europa), but may be generalizable to other versions of the IDE.

These alternatives may be available to obtain a copy of Eclipse for installation:

- download a current stable build of Eclipse from the [Eclipse web site](#); note that the installation file is large (over 120 MB)
- a colleague may have a copy of the download on CD already
- For the recommended package to download and install, click the link **Eclipse IDE for Java EE Developers** on the [Eclipse Packages](#) page. For the reason why this is the recommendation, see the following bullet (and sub-bullets).
- There are [a number of downloadable packages](#) available, each a different "flavor" of Eclipse, which can be compared and contrasted from the [Compare Eclipse Packages](#) page. The **recommended Eclipse package** is the [Eclipse IDE for Java EE Developers](#). This recommendation is made for those who develop in other languages / on other platforms as well, for the following reasons:
 - The "slimmer" [Eclipse IDE for Java Developers](#) lacks data tools, testing support, and parts of the Web Standard Toolkit useful to all Java web application developers
 - The [Eclipse Classic](#) seems like it ought to be "slimmer" but in fact it is a larger download than the JEE package. Downloading and installing this package, then picking and choosing among additional packages described on the [Eclipse Classic](#) page is a viable alternative, but requires each developer to spend time researching the contents and utility of the multiple options.
 - Other choices are appropriate for C/C++ or Eclipse Plug-in developers, but work of this kind is not currently in the IST-AS queue.

4.2. Install Eclipse

Note:

These instructions are written for Eclipse v. 3.3.0 but are likely to be usable for any 3.3.x version of the IDE.

There is no installer (executable program) used to install Eclipse. The process described below involves unarchiving (unzipping) a directory tree of files and placing it in an appropriate location on your hard disk. It is **very strongly** recommended that you locate the `eclipse\` directory at the root of your computer's hard drive; or, minimally, on a directory path with no spaces in its name (e.g., `C:\mystuff\eclipse\`). It is worth noting that Eclipse does not write entries to the Windows registry; therefore, you can simply delete (or move) the installed files, shortcuts, and/or the workspace: there is no executable uninstaller either.

- Unzip (or copy/unjar/check-out) the software (e.g., the downloaded file `eclipse-jee-europa-win32.zip`) into an appropriate location on your hard disk (e.g., `C:\eclipse`).
- These instructions are written assuming that you are running eclipse from `C:\eclipse`; if you are using a different path, adjust accordingly.
- Once the unzipped (copied/unjarred/checked-out) files are located on your filesystem, get started using Eclipse:
 - Run Eclipse by running `C:\eclipse\eclipse.exe`
 - The first time you run Eclipse, you will be prompted to identify a location for your Eclipse workspace. This is where local copies of your projects (files you check in and/or out of code repositories) will live on your file system. *Do not create the workspace in a directory path that has spaces in it - i.e., **not** in the default `C:\Documents and Settings\...` directory presented by default on the first startup of Eclipse. Instead, it is recommended that your workspace be located at the root of your machine's hard disk, e.g., `C:\workspace`.*
 - It is advisable to pass JVM arguments to Eclipse at startup to reserve a larger memory space for the IDE than the default. To, specify recommended JVM arguments, create a shortcut (probably on your desktop) with the following target (modified if you're using different directories):

```
C:\eclipse\eclipse.exe -jvmargs -Xms128m -Xmx512m
-XX:MaxPermSize=128m
```

[Alternately, you can specify these arguments in `eclipse.ini`, located in the same directory as the Eclipse executable (e.g., `C:\eclipse\eclipse.ini`). The contents of `eclipse.ini` would be:

```
-jvmargs
-Xms128m
-Xmx512m
-XX:MaxPermSize=128m
```

]

Warning: If you have multiple installations of Java on your machine, and if the JVM that occurs first in your PATH environment variable is not one in which Eclipse can run, you will see an error on attempted startup. The error will look something like this:

```
Incompatible JVM - Version [###] of the JVM is not suitable
for this product ...
```

Should this occur, the optimal solution is to change your PATH so that Eclipse finds a usable JVM. If that is not possible, you can specify which JVM Eclipse uses with the `-vm` runtime parameter. Building on the example above, the shortcut target used to start eclipse might look, in that case, like this (but all on one line):

```
C:\eclipse\eclipse.exe -vm C:\jdk1.5.0_12\jre\bin\java.exe
-jvmargs -Xms128m -Xmx512m -XX:MaxPermSize=128m
```

However, this will cause an extra command-window to open on Eclipse startup, and is not an elegant solution.

4.3. Familiarize yourself with the IDE

Note:

The Eclipse User Guides are strongly recommended!

The material/tutorials in the Help system are strongly recommended! Starting at `Help:Contents`, go through the `Workbench User Guide`. If you are going to be developing Java platform, you'll also want to go through the `Java Development User Guide`.

If you are going to be using a Subversion repository rather than CVS, and have not worked with CVS or a similar versioning system before, it may nonetheless be worthwhile to review the "Team programming with CVS" material from the `Workbench : Concepts` section of the User Guide. Because CVS support is integrated into the Eclipse IDE, the excellent material for users new to working with code repositories is oriented toward CVS. [The Subclipse (Subversion plug-in for Eclipse) documentation is task-oriented, and is suitable for developers already familiar with code repositories, who need to learn only how to use the Subclipse plug-in.]

4.4. Additional configuration steps for all Eclipse users

Additional configuration steps include setting critical environment variables; adding to the

Ant runtime classpath; and installing useful plugins. The configuration pane is accessed from Eclipse's Window menu: Window : Preferences

- Set file encoding to UTF-8 and line-endings for new files to Unix, so that text files are saved in a format that is not specific to the Windows OS and most easily shared across heterogeneous developer desktops:
 - Navigate to the Workspace preferences (General:Workspace)
 - Change the Text File Encoding to UTF-8
 - Change the New Text File Line Delimiter to Other and choose Unix from the pick-list
 - *Note: to convert the line endings of an existing file, open the file in Eclipse and choose File : Convert Line Delimiters to : Unix*
- On the same preferences page (General:Workspace), *clear* the checkbox "Build automatically."; this will stop Eclipse from performing long build processes when you don't need them performed
- Environment variables necessary to perform Eclipse builds of projects that conform to IST-AS standards for Java development are listed below. (This site contains a brief discussion of [how environment variables are set in Windows](#).)
 - CVSHOME should be set to your Eclipse workspace, e.g., C:\workspace
 - MAVEN_HOME should be set to the place you store files checked out from the maven repository project in CVS, e.g., %CVSHOME%\maven-repo-j5
- Set up repository access (e.g., to CVS or Subversion repositories); see the the [Set up CVS or Subversion repositories](#) section of this HowTo for more detailed information.
- Refer to [Recommended Eclipse Plugins](#) to identify, download, and install the plug-ins you'll need for Streek development.

4.5. Set up CVS or Subversion repositories

Separate HowTo documents on this site describe steps necessary to set up access to a CVS or Subversion repository. If you will be using both flavors of repository, you should visit both HowTo pages referenced below:

- If you'll be using Subversion, follow the [Subclipse Setup](#) HowTo on this site to install and configure the plug-in through which you'll access the repository.
- If you are using CVS, repository access is integrated into the core IDE; follow the [CVS Repository Setup in Eclipse](#) HowTo to see how access to a repository is set up.

4.6. Additional configuration steps for Streek Java/J2EE developers

Warning:

Omission of these steps may result in inability to run Ant builds for Streek projects.

Additional configuration steps include setting critical environment variables; adding to the Ant runtime classpath; and installing useful plugins. The configuration pane is accessed from Eclipse's Window menu: Window : Preferences

- (This site contains a brief discussion of [how environment variables are set in Windows.](#)) Environment variables necessary for building Java/J2EE projects in Eclipse are listed below:
 - JAVA_HOME should be set to the root directory of your Java installation, e.g., C:\jdk1.5.0_12
 - CVSHOME should be set to your Eclipse workspace, e.g., C:\workspace
 - IST_JXDE_HOME should be set to the place you store files checked out from the ist-jxde project in CVS, e.g., C:\workspace\ist-jxde
 - MAVEN_HOME should be set to the place you store files checked out from the maven repository project in CVS, e.g., C:\workspace\maven-repo-j5
 - JBOSS_HOME and TOMCAT_HOME should be set once and once Jboss-Tomcat is checked out from the third-party/jboss repository
 - FORREST_HOME should be set if (and after) [Forrest](#) is installed on your system (it's best to check the Forrest software out from CVS repository third-party/apache rather than downloading it from the project site, as described in the [Prerequisites section](#) of this page).
 - ANT_HOME may be set. This environment variable is useful if [Ant](#) will be used from the command line. A modern version of Ant should be used (e.g., 1.7.0 as of June 2007).
 - CVSROOT may be set. This environment variable is useful if CVS will be used from the command line (e.g., from [Cygwin](#) on a Windows machine). A typical value for CVSROOT is:

```
:ext:mylogin@canvas.berkeley.edu:/usr/local/cvsrep/org/ist
```

- Check in Window:Preferences:Java:Installed JREs to be sure that the correct JRE(s) - the one or ones in which you'll be developing - are installed. Add additional JREs as necessary, and select the one you'll use most regularly (especially for new projects) as the default.
- Refer to [Recommended Eclipse Plugins](#) to identify, download, and install the plug-ins you'll need for Streek development.
- Refer to [Configuring Oxygen...](#) once Oxygen and Forrest are installed to configure validation based on the Forrest DTDs referred from its catalog files.
- Ant tweaks for Eclipse 3.3.x:
 - Navigate to the Ant Runtime preferences (Window:Preferences:

-)
- Choose the Classpath pane
 - You need to add these JAR files to the Global Entries of Ant's Runtime Classpath from the indicated directory (mostly `IST_JXDE_HOME/lib` (the `ist-jxde` project must be downloaded from the repository into your workspace before this step can be performed). Note that you use the `Add JARs` button for additions from within your workspace; and the `Add External JARs` button otherwise.
 - `xercesImpl.jar` from your `IST_JXDE_HOME/lib` directory
 - `xml-apis.jar` from your `IST_JXDE_HOME/lib` directory
 - `ant-contrib-0.6.jar` from your `IST_JXDE_HOME/ant/lib` directory
 - `junit-3.8.1.jar` from your `MAVEN_HOME/junit/jars` directory
 - `xml-commons-resolver-1.1.jar` from your `IST_JXDE_HOME/ant/lib` directory or your `FORREST_HOME/../../lib/core/` directory (cf. [Building Forrest within Eclipse](#) for more info)