Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Information and Communication Systems Research Group

Distributed Computing Group

Installing Eclipse

In some exercises of the Computer Networks (Vernetzte Systeme) course, you will need to write or extend applications written in *Java* or *C*. There are numerous development tools available and you can even write your code in standard editors like *emacs, vim* or *Notepad*. However, we recommend using *Eclipse*, a very powerful open source *IDE* (integrated development environment).

This short tutorial guides you through the installation of *Eclipse 3.2.2*. In the following lecture, we will give you an introduction on how to use *Eclipse* and how to create *Java* and *C* applications. We strongly suggest that you install *Eclipse* before the next lecture because it has a size of about 120 MB. Concurrent download by multiple students would probably overstrain the wireless network during the lecture…

Step 1: Installing Java

Since *Eclipse* itself is written in *Java*, you need to install *Java* first. If you already have the *Java JDK* (1.4, 5, or 6) installed (the JDK, not just the JRE runtime environment!), you can skip this step.

Go to http://java.sun.com and download the *Java SE*. You find it under *Popular Downloads* → *Java SE*. Go to *JDK 6u1,* accept the license agreement and download the version that corresponds to your platform: jdk-6u1-windows-i586-p.exe (56 MB) for Windows, jdk-6u1-linux-i586-rpm.bin for Fedora Core/RedHat, jdk-6u1-linux-i586.bin for Linux versions with other than RPM package management. For Windows, you can take the online version of the file, too, if bandwidth is a problem.

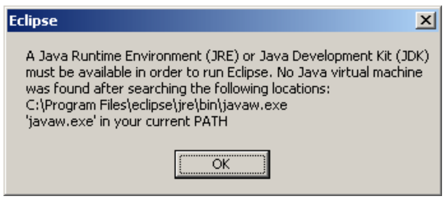
After downloading, execute the file and install *Java SE*.
On **Windows**, the wizard is pretty self-explanatory and you don't need to make any changes to the default settings.
On **Linux**, all that you have to do is to read the license text (or press 'q' ☺), type in *YES* and wait that the installation finishes. You most likely have to add `export JAVA_HOME=/usr/java/current` (or wherever you have installed java) to `~/.bashrc` if you are using bash as a shell. Some Linux distributions ship with the gcj wrapper, which often causes problems with Eclipse. Add the Java executable to your path to ensure that Eclipse is using your Sun JDK and not the *gcj* wrapper (add `export PATH=$PATH:$JAVA_HOME/bin` to `~/.bashrc`). (If this does not help, the brute force approach of simply deleting */usr/java* always helps ☺)

Step 2: Installing Eclipse

Note: If you have an older version of Eclipse installed (like 3.0), you can update it with exactly the same procedure as described below. Eclipse does not support easy updates between major releases (like 3.0 to 3.1), so you need to reinstall the whole application.

Go to http://www.eclipse.org and download *Eclipse 3.2.2* (Eclipse SDK 3.2.2, 121 MB **Windows**, 119 MB **Linux**). After downloading, extract the archive to your favored destination (For Windows, typically in the *Program Files* directory). There is no installer, simply extracting the files from the zip or tar.gz archive suffices.

Now you are ready to start *Eclipse* by double-clicking *eclipse.exe* (**Windows**) or typing `> eclipse &` (**Linux**). If everything works as expected, you should be asked to specify your workspace directory. You can leave this on the default value or choose any other location. After this dialog, you should see the *Welcome to Eclipse* screen. If you reach this point, you are ready for the tutorial lecture :)


Eclipse

A Java Runtime Environment (JRE) or Java Development Kit (JDK) must be available in order to run Eclipse. No Java virtual machine was found after searching the following locations:
C:\Program Files\eclipse\jre\bin\javaw.exe
'javaw.exe' in your current PATH

If you see this message on startup, *Eclipse* cannot find your *Java* installation. In that case, you have to manually add the location of *Java* to your PATH variable.
In **Windows**, this is done by right-clicking *My Computer,* clicking on *Properties,* selecting the *Advanced* tab and clicking on the *Environment Variables* button on the bottom of the window. Create a new System variable, call it JAVA_HOME and assign it

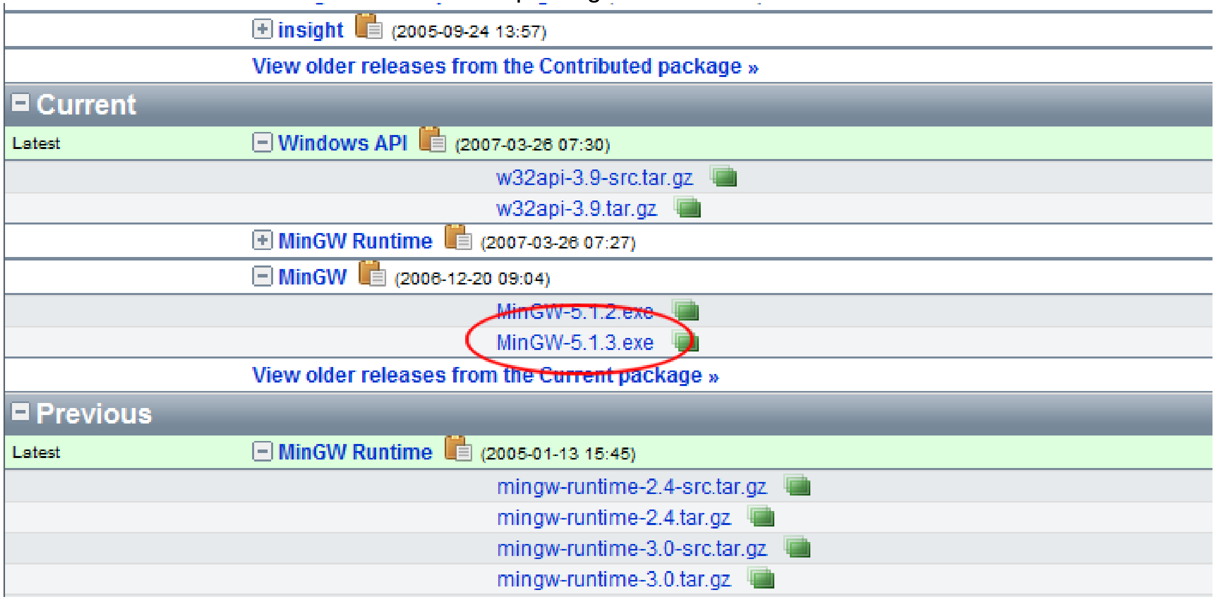
the value C:\Program Files\Java\jdk1.5.0_06 (or wherever you installed Java).
If *Eclipse* still cannot find *Java*, try to add C:\Program Files\Java\jdk1.5.0_06\bin to your Path variable (separated by a semicolon). If this does not help, too, try uninstalling and reinstalling *Java*.
In Linux, ensure that the java in your path is actually the Sun JDK by typing `which java.`

Step 3: Installing the CDT

CDT is the C Development Toolkit for Eclipse. You can use it for the C exercises. CDT is an optional Plugin that you have to install on Eclipse. Open Eclipse, click on the *Help* menu item and choose *Software Updates → Find and Install*. Select the Search for new features to install radio button and click Next >. Select the *Callisto Discovery Site* and click *Finish*. (You are most likely asked to choose a mirror site. Select one that is close, preferably the *SWITCHmirror*). Expand the list of features of the *Callisto Discovery Site*. *Select C and C++ Development* and click *Next* >. Accept the license agreement, click *Next* > and then *Finish*. Once the update manager has finished the job, you are asked to restart the workbench.

Step 4: Configuring the CDT

In order to do something useful with the CDT, you have to have a C compiler on your machine.
On **Linux**, just make sure that you have gcc and gdb installed. If not, install the two packets and you are done.
On **Windows**, you have to install the basic GNU tools.
If you already have Cygwin with gcc, g++, make, and gdb installed, just add your Cygwin bin directory to the path so that the GNU tools are visible to the Windows system.
If you don't have Cygwin, the best way is to use the CDT with MinGW, the minimalist GNU environment for Windows. Download and install it from http://sourceforge.net/projects/mingw/. You have to choose MinGW-5.3.1 from the current MinGW package.



	⊞ insight (2005-09-24 13:57)	
	View older releases from the Contributed package »	
⊟ Current		
Latest	⊟ Windows API (2007-03-26 07:30)	
		w32api-3.9-src.tar.gz
		w32api-3.9.tar.gz
	⊞ MinGW Runtime (2007-03-26 07:27)	
	⊟ MinGW (2006-12-20 09:04)	
		MinGW-5.1.2.exe
		[✓ MinGW-5.1.3.exe]
	View older releases from the Current package »	
⊟ Previous		
Latest	⊟ MinGW Runtime (2005-01-13 15:45)	
		mingw-runtime-2.4-src.tar.gz
		mingw-runtime-2.4.tar.gz
		mingw-runtime-3.0-src.tar.gz
		mingw-runtime-3.0.tar.gz

Start the installation. Select *Download and install*. Choose a custom installation with the *MinGW base tools, g++* and *MinGW make*. Once the installer has finished right-click on *My Computer,* select the *Advanced* tab and click on the *Environment Variables* button. Assuming that you have installed MinGW in C:\MinGW, add *;C:\MinGW\bin* to the PATH environment variable (or create it when it does not exist so far). If you want to debug your applications, you might also want to install the GBU debugger (*gdb*), available under *View older releases from the "Current" package*, *gdb, gdb-5.2.1-1.exe*

Close the window, open a command shell (cmd.exe) and type *gcc --version*. If you get a correct output, you are almost done. Go to *C:\MinGW\bin* and copy *mingw32-make.exe* to *make.exe*.