Exercise 0 – Getting started

Objective

The objective of this exercise is to get Java and Eclipse installed on our machines.

(Please note: Links were correct as of the 18th April 2016. They may be subject to change over time!)

Overview

Step one: Install Java

Before we are able to use java it needs installing. We need to get the Java development kit (JDK) from the Oracle website. Either use a search engine to find the JDK or go to:

http://www.oracle.com/technetwork/java/javase/downloads/index.html

Then select the Java Platform (JDK) from the two options. On the page there should be the option to download the JDK.

We need to accept the license agreement and then choose the correct version for our operating system. We're running 64 bit Windows 7, so chose the "Windows x64" file from the list.

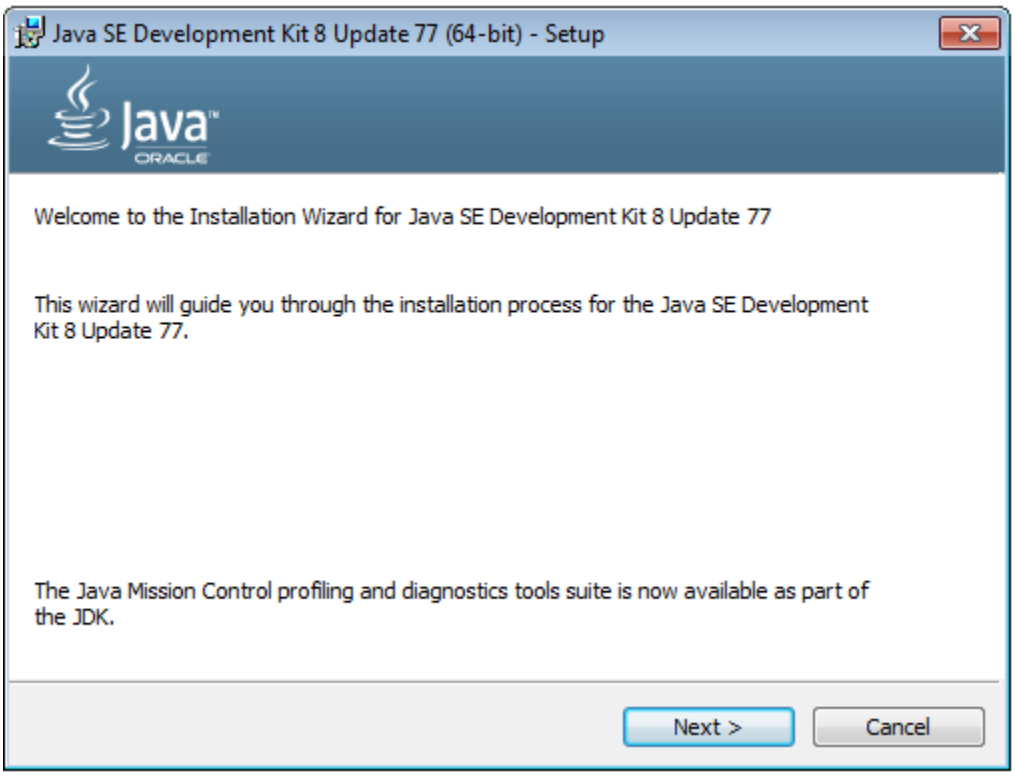
Java SE Development Kit 8u77

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Accept License Agreement Decline License Agreement

Product / File Description	File Size	Download
Linux ARM 32 Soft Float ABI	77.7 MB	jdk-8u77-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Soft Float ABI	74.68 MB	jdk-8u77-linux-arm64-vfp-hflt.tar.gz
Linux x86	154.74 MB	jdk-8u77-linux-i586.rpm
Linux x86	174.92 MB	jdk-8u77-linux-i586.tar.gz
Linux x64	152.76 MB	jdk-8u77-linux-x64.rpm
Linux x64	172.96 MB	jdk-8u77-linux-x64.tar.gz
Mac OS X	227.27 MB	jdk-8u77-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	139.77 MB	jdk-8u77-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	99.06 MB	jdk-8u77-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	140.01 MB	jdk-8u77-solaris-x64.tar.Z
Solaris x64	96.18 MB	jdk-8u77-solaris-x64.tar.gz
Windows x86	182.01 MB	jdk-8u77-windows-i586.exe
Windows x64	187.31 MB	jdk-8u77-windows-x64.exe

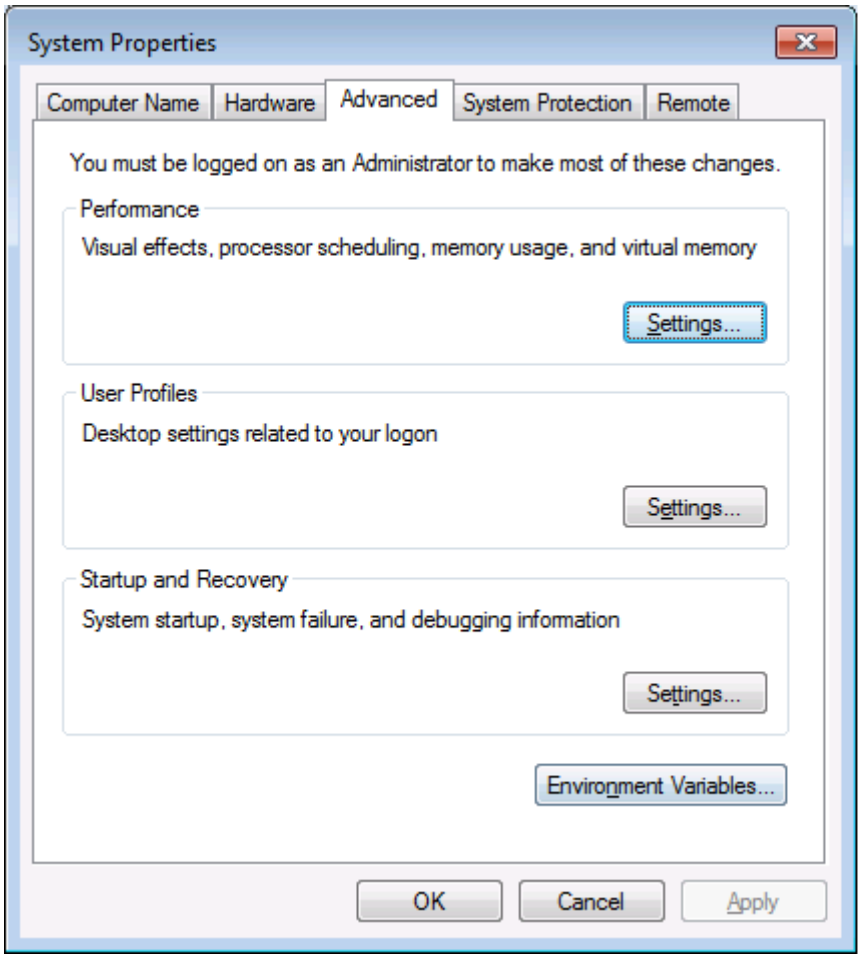
When it has downloaded, open the file and follow the instructions. Just allow everything to install using the defaults.



To check if java has installed correctly we need to use the command line terminal. Press the start button and type "`cmd`" followed by enter to open a terminal window. To check if the main java virtual machine has been setup type `"java -version"` (without the quotes). This should output the current java version.

Then type `"javac -version"` to see if the compiler has successfully been added to the command line. If it hasn't then we will need to add the path for the javac program to the environment variables.

To set these go back to the start button and type "Environment Variables". Select the second option, "Edit System Environment Variables".

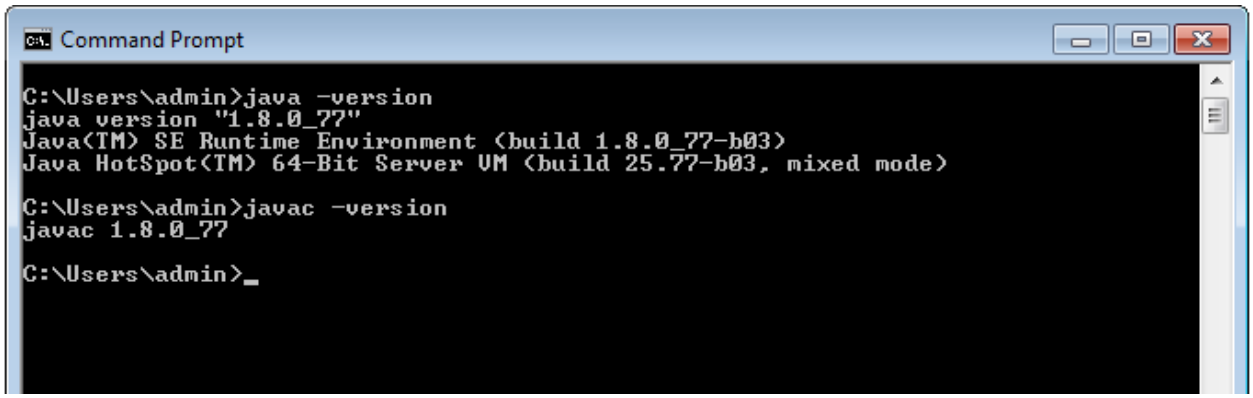


On this screen select the "Environment Variables" button at the bottom. In the lower scroll box, system variables, scroll down until you find path. Click edit.

IMPORTANT: DO NOT DELETE ANYTHING IN THIS BOX!

Add on the end of the list `";C:\Program Files\java\jdk1.8.0_77\bin"`. The semi colon adds this entry to the end of the list in the path already. The exact path to put in there will be dependent on which version of java 8 we have downloaded. In this case I installed it to the default location. The \bin folder is where all the binary executable files are held, this is where we have the javac program we use to compile java.

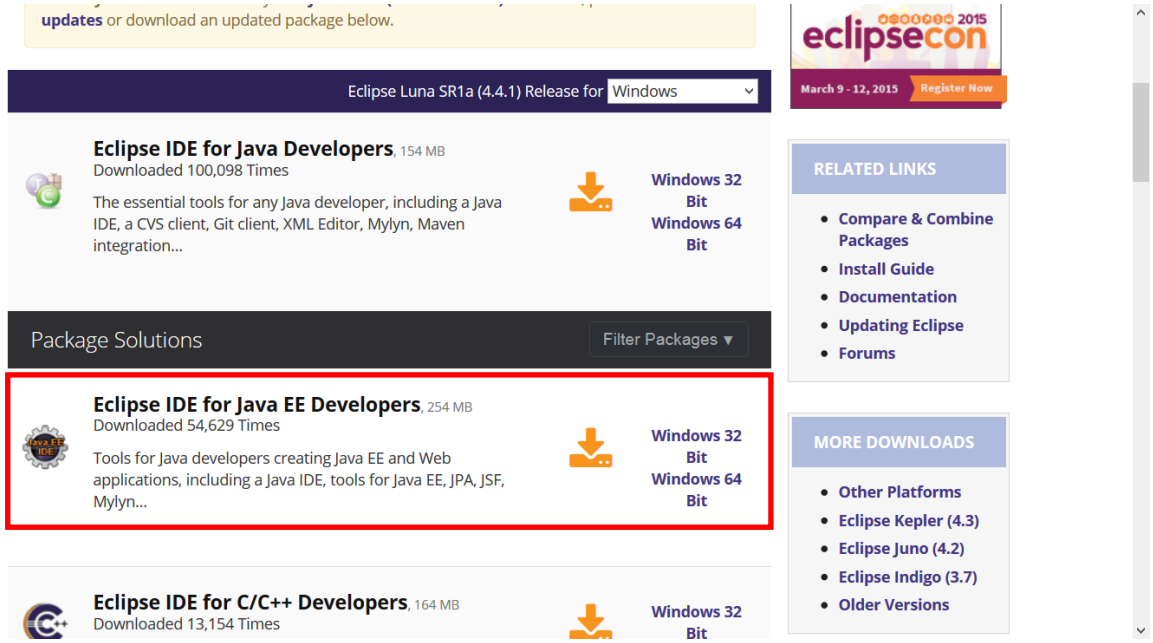
When you've added the new path, click okay and close the system environment variables dialog. Close the current terminal and open a new one. This will reload the environment variables (you can check what your current value for path is by typing `"echo %path%"`). In the new terminal type `"java -version"` and `"javac -version"` to check if it has set everything up correctly. The output should look something like this:



Step two: Install Eclipse

Eclipse is much easier to install than Java. Eclipse is the IDE of choice for this course. We can find it at https://eclipse.org

Click the download button and select the version of the IDE that you want in either 32 or 64 bit. We are only going to play around with the basic java tools and some of the web projects, so we want the second option, "Eclipse IDE for Java EE Developers". You can install other plugins to make eclipse handle other languages from within the program later. Select a mirror to download from, any will do.



Eclipse doesn't 'install' like other things. It just runs without needing to add anything to your operating system. This means we can run it from a USB stick if required!

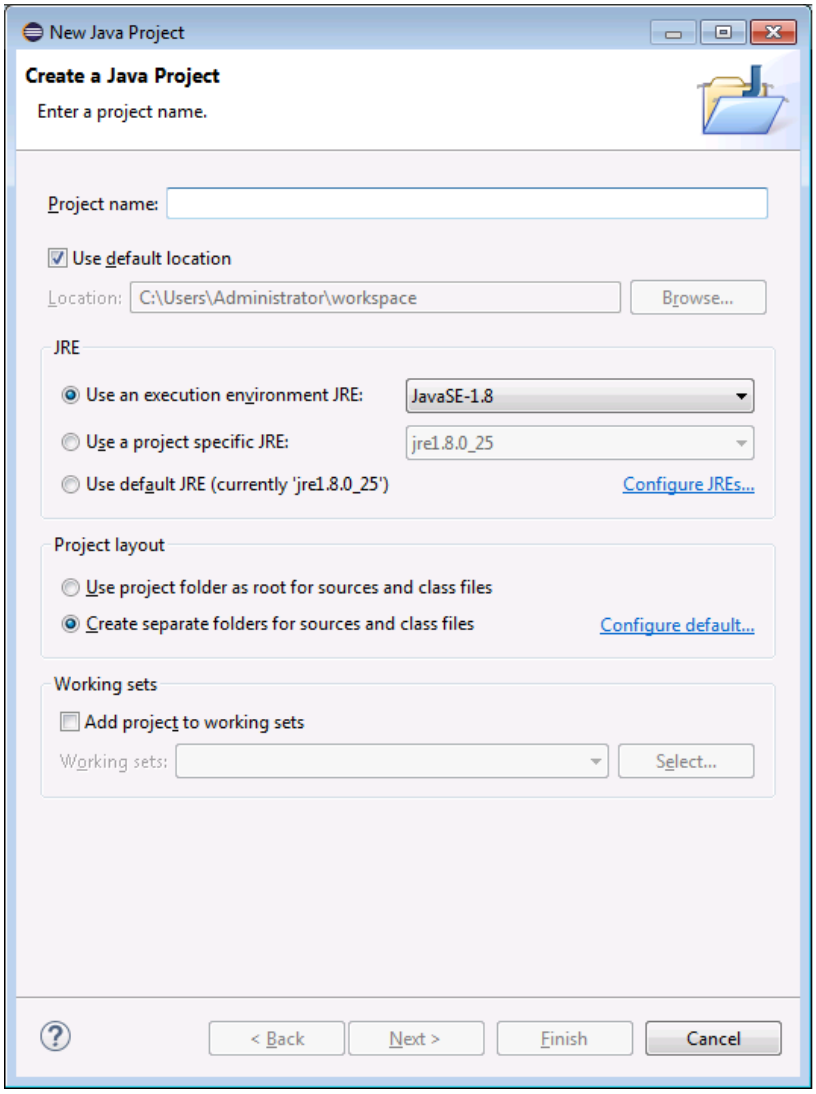
Extract the zip file to somewhere you want to keep eclipse. It's not recommended that you use Program Files as it will have write permission issues. I tend to extract it to either a folder called programs in my documents, or to c:\eclipse

Finally, we want to create a shortcut to make life easier. Look in the Eclipse directory for a file: `eclipse.exe` Right click this, go to "send to" and select desktop. If everything has worked then we can now click on the shortcut on the desktop to open the Eclipse tool suite.

Step three: Hello World!

When you open Eclipse for the first time it will ask you about which workspace you want to use. A workspace is where Eclipse saves all the project files that tells it how to open each project you will create. You don't need to have your java project files in the same place as your workspace, but it does make life a little easier and neater. The default will be fine for this week so click okay.

On the Eclipse main screen there are loads of options. We want to create a new project though rather than any of these so click on the file menu, and go to new and select Java Project.



Give the project a name - "HelloWorld"

Note how it uses the default workspace location if you want it to. You can change this to anything you like.

Next we have the JRE boxes. This refers to the java runtime environment you want to use. We're going to be using java 8 in this course as it's the most recent one, and allows us access to some of the functional constructs we are after for later in the week.

The next box is Project Layout, leave this as a default. You can have all the source and class files mixed together, but that is not usually recommended.

We can also add this project to a working set. We don't have any working sets at the moment, so let’s create one.

A working set is a folder inside the workspace that you can use to organise projects. For example: separating examples, exercises and just playing around with Java. You can organise them into specific projects you're working on at work if you want, or anything you like. Let's create a working set called "Exercises"

Select the tick box for "Add project to working sets" then click the select button on the right. This brings up another interface where we can create a new working set.

Select New, then Java and click next. Give it the name exercises, and click finish

We can now select our new working set for this project. Click the tick box to the left of the name and select okay. With all that done we can now click on the "Finish" button to create the class.

When you click finish it looks like nothing has happened at first, but if you click the X on the welcome tab then the project will be visible.