



Queensland University of Technology

INB370 Software Environment Handbook

How to configure your Windows computer for INB370

Andrew Craik
23-Feb-09

Introduction

Welcome to INB370 – Software Development. In this course you will gain experience using professional development tools as part of the real-world development process with the exception of a couple of the more advanced processes. All of the software required to undertake the work in this course will be made available on the lab computers. Past experience has shown us that many students use their own personal computers together with/instead of the lab software. This guide will help you to setup your Windows computer for the world in this course.

Please remember that the software environment on the lab computers is the official environment and all assignments will be tested using this configuration so it is in your best interests to ensure your submission works in this environment prior to submission.

The tutorial work in this course will be undertaken in Java using the Eclipse IDE. Additional tools will be introduced for testing, source control, and building. For Semester 1 2009 we plan to use the following pieces of software (this list is subject to revision during the semester, please check on blackboard for any revisions):

- Sun Java SE JDK 1.6.0.
- Eclipse Europa Java SE Edition
- JUnit 4.5
- Ant 1.7.1
- Tortoise SVN 1.5.7

The remainder of this document covers the installation and configuration of these different applications.

Java SE 1.6.0

Unlike the .NET platform which you may have seen previously, there are a number of competing Java implementations on the Windows platform. The most recent version of Java available on the Windows platform is Java 1.6. Java 1.5 will also work, but the official supported configuration will be Java 1.6.

The Java platform comes in a number of editions:

Edition	Abbreviation	Useage
Standard Edition	SE	Standalone software & browser applets
Enterprise Edition	EE	Enterprise applications (superset of SE)
Micro Edition	ME	Mobile phones and other resource limited platforms (subset of SE)

In addition to these different editions, each edition is supplied in two variants: the Java Runtime Environment (JRE) and the Java Development Kit (JDK). The JRE includes only the Java Virtual Machine used to run programs. The JDK includes the runtime environment (JRE) as well as the Java compiler used to compile programs as well as other development and documentation tools.

Obtaining Sun Java SE 1.6.0

1. The Sun Java JDK 1.6.0 is available from <http://java.sun.com/javase/downloads/index.jsp>.
2. Locate the link for the JDK 1.6.0 Update 12 and click the download button
3. Choose your Platform (Windows if you are running a 32bit version of Windows or Windows x64 if you are running a 64bit version of Windows)
4. Read and agree to the JDK 6.0 License Agreement and click continue
5. Choose either the online or offline installation and download the appropriate installer (the online installation has a smaller overall download, but the installer will need the computer to be connected to the internet during installation so that JDK can be downloaded).

Installing Sun Java SE 1.6.0

1. Execute the installer you downloaded.
2. During installation you will be prompted to choose the location where you would like the JDK installed, you should change the path to C:\jdk6. The use of the default path or a path with spaces can confuse some build tools; choosing a short directory name off the root of the drive prevents these complications

3. Once the JDK has installed, you will be prompted to choose a location to install the JRE to, you can accept the default for this.
4. You will also be asked if you want to register the JRE with your browser, you should do this so you are able to run Java Applets.

Configuring your Environment

Once you have installed Java 1.6.0 on your system you need to make sure your system is properly configured to use the new version of Java. Java reads its configuration from environment variables and we will now setup these variables:

1. Open your Control Panel and select Classic View on the left hand side if it isn't your default view.
2. Double click System.
 - a. *Windows XP and earlier:* Choose the Advanced Tab and click on the Environment Variables button
 - b. *Windows Vista:* Click the Advanced system settings link on the left of the window and then click the Environment Variables button
3. In the window that opens there are two lists of variables, we are interested in the System variables list.
4. Scroll down the list to see if there is a variable called JAVA_HOME, if there is make sure that its value is set to C:\jdk6 (or the location where you chose to install Java).
 - a. *I can't find JAVA_HOME* click on the New... button in the System variables area, enter the variable name JAVA_HOME, and enter the variable value C:\jdk5 (or the location where you chose to install Java).
 - b. *I have a JAVA_HOME variable, but it isn't set right:* Double-click the entry and fix the value
5. Next we will set the CLASSPATH variable which is a ; delimited list of places where Java should look if it can't find a class. The CLASSPATH should contain . and %JAVA_HOME%\lib as a minimum. Other entries in addition to these are ok. Double-click to edit the CLASSPATH or add a new CLASSPATH as was done with the JAVA_HOME variable
6. Finally, scroll down to the PATH variable and double-click to modify the entry and append ;%JAVA_HOME%\bin to the value.
7. Click OK to close all the open dialog boxes and close the Control Panel
8. To test if you have gotten things setup you can open a Command Prompt (usually under Accessories) and at the prompt enter 'javac' without the quotes. You should see a version and usage instructions. If you see 'javac' is not recognized as an internal or external command, operable program or batch file you made a mistake.

Documentation

The Java programming language comes with a large collection of libraries that you will use extensively. These libraries are extensively document in the Java API documentation. This documentation can be read online at <http://java.sun.com/javase/6/docs/api/>.

To download a copy of the documentation so that you can refer to it even when your computer is not connected to the Internet:

1. Go to <http://java.sun.com/javase/downloads/index.jsp> and download the Java SE 6 Documentation entry.
2. Right click on the zip file you just downloaded and choose Extract All...
3. In the wizard enter the destination as C:\jdk6 (or where you chose to install Java).
4. Open Windows Explorer and navigate to where you installed Java.
5. Inside the location you will see a docs folder, open it.
6. Double click on the index.html file and your browser will open with your very own copy of the Java documentation.

Eclipse Europa

There are a number of IDEs available for the Java language. We will be using Eclipse which is an open source Java IDE originally written by IBM, but now developed by a large number of companies.

Eclipse, like Java, comes in many different editions which are customized for use with the different editions of Java discussed in the previous section. We will be focusing mainly on using Java SE so we will be using the Java SE edition of Eclipse.

The most current version of Eclipse is called Ganymede. Unfortunately, the lab computers are configured with Eclipse Europa, the previous edition. While the differences between the two are not major for the things we will be doing, to avoid any compatibility problems you are encouraged to use the same version as the lab computers.

Obtaining Eclipse

1. Open a browser and navigate to <http://www.eclipse.org/downloads/packages/release/europa/winter>.
2. Click on the link for the Eclipse IDE for Java Developers
3. Click on the link next to Download from: to start your download

Installing Eclipse

1. Right-click on the zip file you just downloaded from eclipse.org and choose Extract All...
2. Choose a location on your system; C:\ is usually a good place. *Windows Vista Users:* Do **NOT** extract the zip into C:\Program Files; Eclipse needs to write its configuration files in its directory and Windows Vista by default prohibits this in protected areas like the Program Files folder; life will be much easier if you put it somewhere else like C:\
3. Open a browser and navigate to the location where you extracted the zip file to. You will see an eclipse folder.
4. Open the eclipse folder and double click on the file eclipse.exe.
5. After the splash screen as displayed, you will be prompted to choose a workspace, the default is fine for now so choose OK (for those familiar with Visual Studio a workspace is like a solution).
6. Once Eclipse is open your installation is complete. You may wish to add shortcuts to eclipse to your desktop or start menu for easy access.

Ant

Building a large application is not as simple as hitting the run button in the IDE. There may be dependencies between different chunks of code which require them to be compiled in a specific order. Further, there are almost always tools other than the compiler which must be run to produce the final product. These can include extracting the source code from the repository, generating documentation using tools like javadoc, building the installation package using tools like InstallShield, as well as all housekeeping tasks to clean up the build environment and place the build results in an accessible location. To orchestrate the running of all these different tasks large software project make use of build tools. In this course, we will be using one popular open source option called Ant.

Obtaining Ant

1. The binary version of the latest Ant version can be downloaded from <http://ant.apache.org/bindownload.cgi>.
2. Download the .zip archive

Installing Ant

1. To Install Ant, extract the zip file. The zip file will expand to create an apache-ant-1.7.1 directory with all of the documentation and jar files arranged under it. You should avoid expanding the zip file to a path which contains spaces so extracting it to c:\ is a good option. *Windows Vista Users:* You should avoid using the Program Files folder due to the User Account Controls.
2. Open your Control Panel and select Classic View on the left hand side if it isn't your default view.
3. Double click System.
 - a. *Windows XP and earlier:* Choose the Advanced Tab and click on the Environment Variables button
 - b. *Windows Vista:* Click the Advanced system settings link on the left of the window and then click the Environment Variables button
4. In the window that opens there are two lists of variables, we are interested in the System variables list.
5. Click the New... button in the System variables area. Set the variable name to ANT_HOME and the value as the location of the ant folder you just created. For example, if you extracted to c:\ the value would be c:\apache-ant-1.7.1. Click OK.
6. Double-click the PATH variable and append ;%ANT_HOME%\bin to it.
7. Click OK to close all the dialog boxes

```
C:\>ant
Buildfile: build.xml does not exist!
Build failed
```

JUnit

Unit testing is an important part of major programming projects; without it bugs can be easily introduced into applications and required functionality may not be added to the application. There are a number of tools available to facilitate the writing and running of unit tests. One popular open source option for the Java platform is JUnit which you will be using in this course.

JUnit can be integrated with Eclipse, but to give you exposure to different interfaces and different ways of working we will also be using it as a standalone application.

Obtaining JUnit

1. JUnit can be obtained from <http://www.junit.org>.
2. Go to the download section and download the junit4.5.zip file. This file contains the JUnit jar as well as all the JUnit documentation for your reference.

Installing JUnit

1. To Install JUnit, extract the zip file. The zip file will expand to create a junit4.5 directory with all of the documentation and jar files arranged under it. You should avoid expanding the zip file to a path which contains spaces so extracting it to c:\ is a good option. *Windows Vista Users:* You should avoid using the Program Files folder due to the User Account Controls.
2. Browse to the location of the junit-4.5.jar file inside the new junit folder.
3. Open your Control Panel and select Classic View on the left hand side if it isn't your default view.
4. Double click System.
 - a. *Windows XP and earlier:* Choose the Advanced Tab and click on the Environment Variables button
 - b. *Windows Vista:* Click the Advanced system settings link on the left of the window and then click the Environment Variables button
5. In the window that opens there are two lists of variables, we are interested in the System variables list.
6. Scroll down and locate the CLASSPATH variable which is a ; delimited list of places where Java should look if it can't find a class. Append the fully qualified location of the junit-4.5.jar file. For example if you expanded the zip in c:\ you would append ;c:\junit4.5\junit-4.5.jar to the CLASSPATH variable.
7. You will now be able to run your JUnit tests from the command prompt. More details of how to this will be shown in the practical sessions.
8. JUnit is one of the most commonly used tools for Java development under Ant. However, Ant needs to be able to find the JUnit jar. The simplest way of handling this is to make a copy of the junit-4.5.jar file, and place it in the lib directory of the Ant installation.

Tortoise SVN

Subversion (SVN) is a popular open-source source control system that was designed to address some of the short comings of the CVS source control system, which was descended from a long line of source control systems going back to the early days of UNIX. Prior to the introduction of Subversion, CVS had a massive following in both the commercial and open source communities and CVS users are slowly migrating to SVN. Like all good open source tools, there are both command-line and GUI clients. On Windows, one of the most popular GUI interface tools is called Tortoise SVN. We will be using this tool to check source into and out of SVN repositories.

Obtaining Tortoise SVN

1. Open a browser and navigate to <http://tortoisesvn.net/downloads>
2. Scroll down the page to the listing of files about half-way down and choose the appropriate MSI for your system (32bit for regular Windows; 64bit for 64bit versions of Windows).

Installation

1. Copy the MSI file to your local system if it isn't already there.
2. Double-click to run the installer and accept the default options.

The school will provide a Subversion server for the use of members of this class. Details of this will be provided once the system is ready for use.

Subclipse

The standalone TortiseSVN client is very versatile and will be used in the practical sessions to demonstrate how to use Subversion. It is often convenient to integrate access to the source control system with the development environment. This reduces the number of different tools the developer must use which increases his or her efficiency. Eclipse by default includes support for the older CVS source control system. There is an open source Eclipse plug-in called Subclipse which adds support for Subversion to Eclipse which you may want to use on your own machine.

Obtaining & Installing Subclipse

1. Eclipse plug-ins are downloaded and installed from within Eclipse. Open Eclipse on your machine (it does not matter which workspace you open).
2. From the Help menu select Software Updates...
3. In the Software Updates dialog choose Add Site...
4. Enter http://subclipse.tigris.org/update_1.4.x as the location and enter a name for the URL like Subclipse.
5. The new name you chose will show up in your list of sites. Ensure it is the only entry checked and then click Finish.
6. Expand the Subclipse entry which shows up in the window that pops up. Place a checkmark next to the Subclipse (required) and the SVNKit Adapter (optional) sub-entries under the URL and click the Install button.
7. Click Next
8. Accept the license and then click Finish. Eclipse will now download and install the Subclipse plug-in from the project's server.
9. Click Next to accept the default optional features.
10. Keep clicking Next until the Finish button is enabled and then click it.
11. You may receive a warning about unsigned features, if you do choose Install All to continue the installation.
12. Once the install is complete you will be asked if you want to restart Eclipse, choose Yes.

Using Subclipse is like using the default Eclipse CVS client. There are a large number of walkthroughs and manuals available online. A Google search for 'how to use subclipse' is a good place to start.

Command Line SVN Client

The GUI clients for Subversion help to simplify accessing and updating the Subversion repository. The official Subversion interface is a command line client. The command line client does not require the computer to support GUI interfaces and provides some more advanced features not accessible through the GUI Subversion clients. The command line client is also the client of choice for automation of Subversion tasks. If you are interested in learning more about the Subversion Command Line Client, you can download a free open source version along with the SVN Book (Subversion's official manual).

Installing the command-line Subversion client

1. Open a browser and go to <http://www.open.collab.net/downloads/subversion/>
2. Scroll down to find the CollabNet Subversion Command-Line Client v1.5.5 (for Windows) and Download it (you will have to register to do so).
3. Run the installer and choose the default options.

The installation will automatically configure your command prompt to recognize the svn command. You can access the SVN Book from your Start Menu.