IE 170 Laboratory 0: The Eclipse Integrated Development Environment

Dr. T.K. Ralphs

Due January 16, 2006

1 Laboratory Description and Procedures

1.1 Learning Objectives

1. Understand the purpose and uses of an integrated development environment (IDE).
2. Understand how to install and use the `Eclipse IDE` with `MinGW` and `MSys`.
3. Understand how to develop a simple C/C++ application using `Eclipse` from scratch.
4. Understand how to import existing C/C++ code into an `Eclipse` project.
5. Understand how to debug a C/C++ application in `Eclipse`.

1.2 Key Words

1. Source (file)
2. Header (file)
3. Object (file)
4. Binary (file)
5. IDE
6. Eclipse
7. Debug

1.3 Installing Eclipse, MinGW, MSys, and GDB

The purpose of this lab is to install the integrated development environment (IDE) and compiler we will be using in the course this semester. An IDE is an application for developing, building, and debugging software applications. You have probably used IDE such as `Borland C++ Builder` or `Microsoft Visual C++`. This semester, we will be using a free, open source IDE called `Eclipse` that was originally developed by IBM and now by the non-profit `Eclipse Foundation`. The compiler, builder, and debugger we will use are all also free, open source tools developed originally by the `Free Software Foundation`. Our first step will be to install Eclipse, by following the steps here.

1. Install `Eclipse 3.1` from the campus network.

2. Install CDT as follows. First, open Eclipse from the Start menu. In the **Help** menu select **Software Updates** and then **Find and Install**. Select **Search for new features to install** and click **Next**. Click **New Remote Site** to add an update site called `C/C++ IDE` with the URL

   `http://download.eclipse.org/tools/cdt/releases/eclipse3.1`

   Check off the site to include it in the search and click **Finish**. In the next window that pops up, select **C/C++ IDE** and click **OK**. Finally, in the **Search Results** window, check off **C/C++ IDE** and click **Finish**. The software should install automatically. Leave `Eclipse` open after installation.

3. Install `MinGW` by running the following installer

   `http://prdownloads.sf.net/mingw/MinGW-5.0.0.exe?download`

   When asked what components to install, check off **g++ compiler** and **MinGW Make**. Choose either `C:\MinGW` or `H:\MinGW` as the install location. If you choose `H:\MinGW`, you will not have to reinstall to use the software at another public site, but it requires 70MB of storage.

4. Install `MSys` by running the following installer

   `http://prdownloads.sf.net/mingw/MSYS-1.0.9.exe?download`

   Choose either `C:\MSys` or `H:\MSys` as the install location, depending on what you did above. Answer the questions posed by the post-install script.

5. Install the debugger `gdb` by running the following installer:

   `http://prdownloads.sf.net/mingw/gdb-5.2.1-1.exe?download`

   Provide the proper location of your `MinGW` installation.

6. In `Eclipse`, under the **Windows** menu, choose **Preferences**. Under **C/C++**, click on **Managed Build** and under the **Environment** tab, click on **New** and choose **Path** on the **Name** pulldown menu. Add `C:\MinGW\bin;C:\MSys\bin` to the front of the path and click **OK**.

An important component of the overall system we will be using is the debugger, which allows you to find errors in your code by examining its state as it is running. The debugger we will use is called `GDB` and should now be installed in the `MinGW` installation directory. Since `GDB` will not be in your execution path, when you set up your program for debugging, you will have to specify the path to it before you will be able to debug your project. To do so, Choose **Debug...** from the use the **Run** menu in Eclipse. On the **Debugger** tab of the window that pops up, Choose `GDB Debugger` on the **Debugger** pulldown menu. Finally, under **Debugger Options**, click on the **Browse** button next to the **GDB Debugger** dialog and locate `gdb.exe` in the MinGW installation directory in the `bin` subdirectory. The debugger should now work.

2 Laboratory Assignments

1. Install `Eclipse`, the `Eclipse C/C++ IDE`, `MinGW`, `MSys`, and `GDB`.
2. Following the steps outlined at the URL

   `http://met.dnsalias.net:1111/teaching/cdt/ar01s04.jsp`

   develop a simple application that prints "Hello, World" to the console window.
3. Change your program so that it prints out the numbers 1 through 10 to the screen.
4. Test the debugger by following the instructions above and then stepping through your program one line at a time.