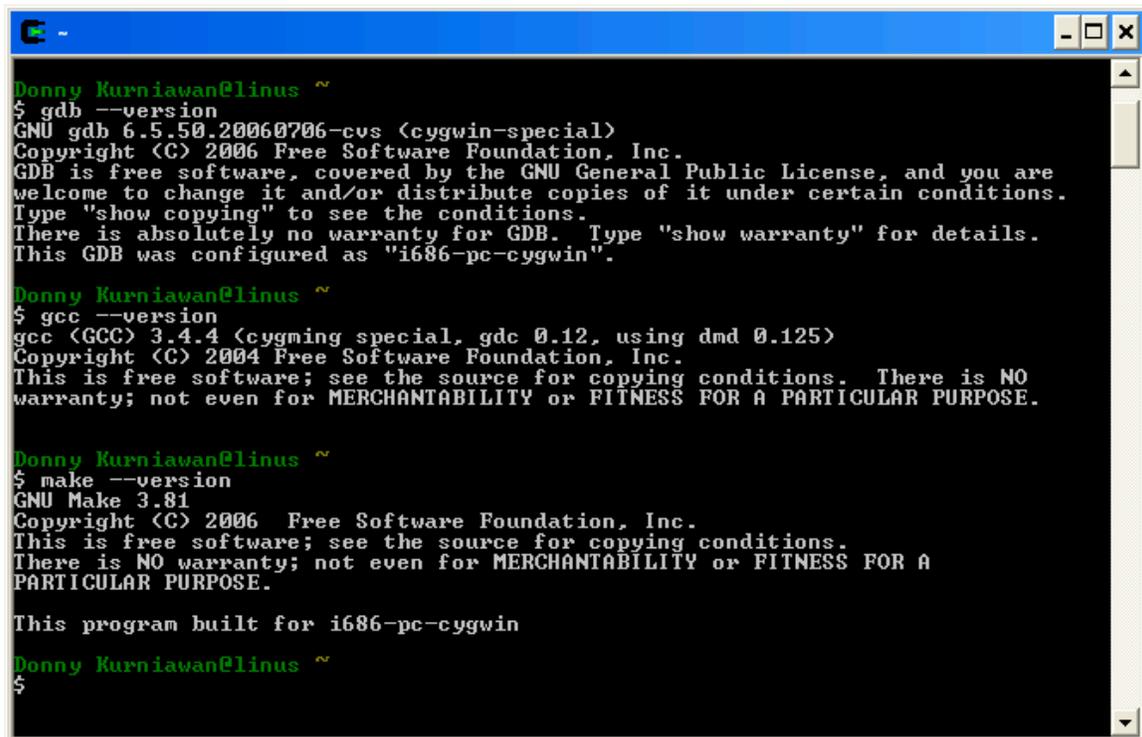# Tutorial 05: Eclipse for C/C++ Development

Contents:
1. Installing Cygwin.
2. Installing CDT.
3. Creating and running a C application.
4. Debugging a C application.

Installing Cygwin

1. Download setup.exe.
2. Double-click the file.
3. Accept the default and recommended options: "Install from Internet", "C:\cygwin", "All Users", "Unix/binary".
4. Select the "http://mirror.aarnet.edu.au" mirror.
5. Install the latest version of "gcc-core", "gdb", and "make" (under All > Devel) and accept the default installation options.
6. Check your installation by running the Cygwin Bash Shell.

Eclipse CDT is really designed to work with GNU tool chain (which, on Windows, is Cygwin/MinGW). CDT might not work correctly with other tool chains (i.e. Visual Studio, Borland, Watcom, etc.).

7. Update the Windows Path. Go to the Control Panel and open up the System Properties.
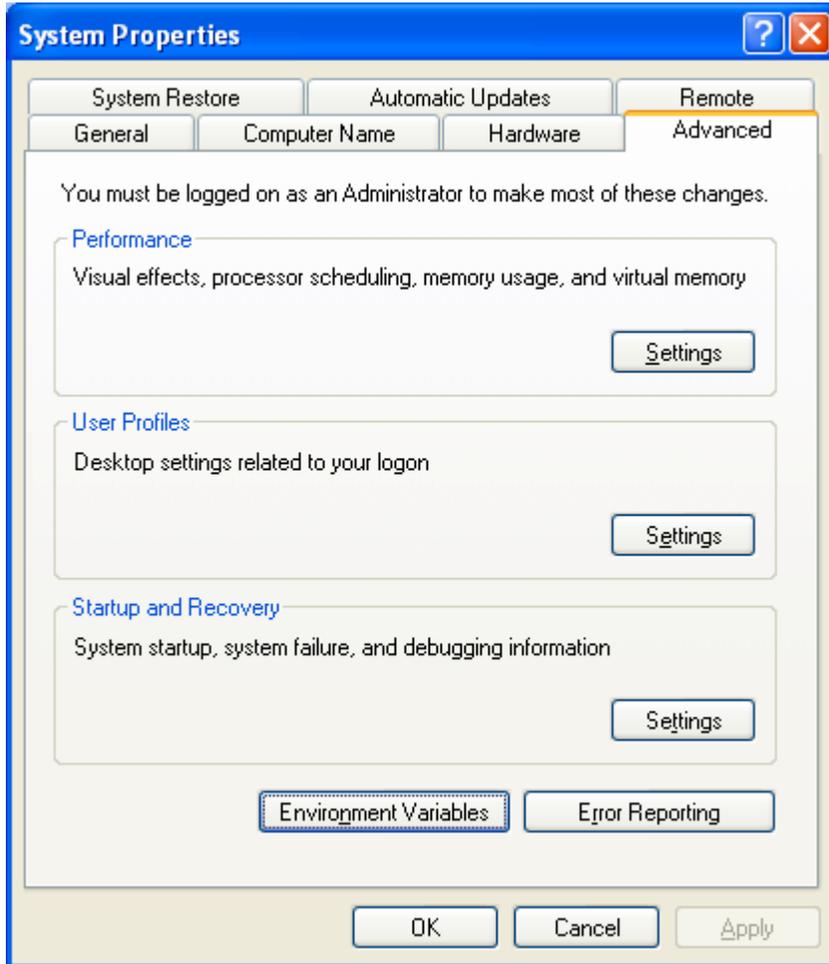
8. Click Environment Variables, edit the System Variable Path, and append ";c:\cywin\bin".
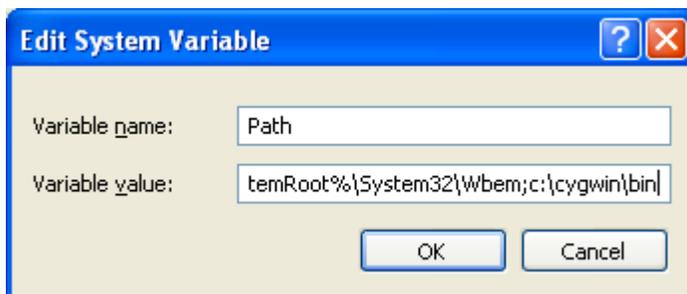
Installing CDT

1. Click Help > Software Updates > Find and Install…

2. Select "Search for new features to install" and click Next.
3. Select "Callisto Discovery Site" and click Finish.
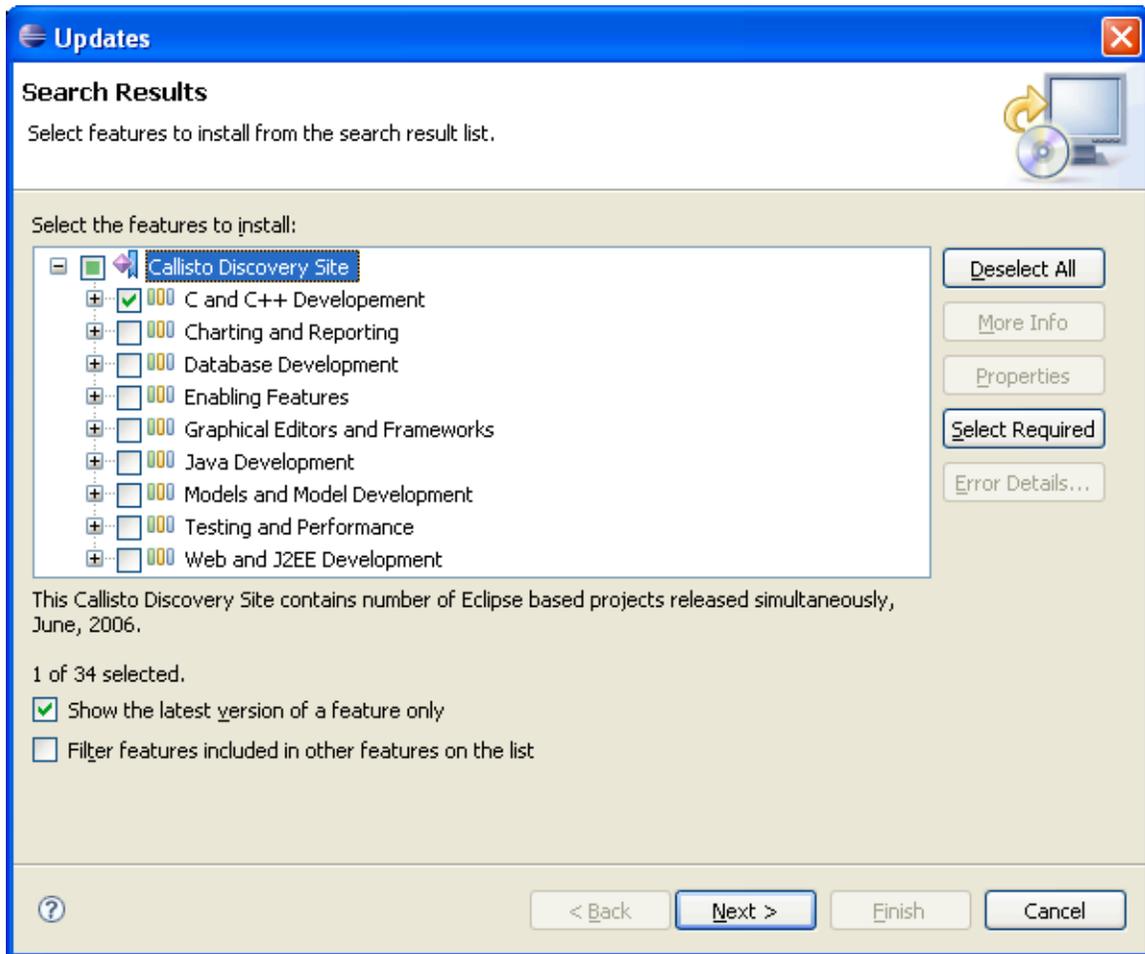4. Select one of the mirrors in Australia and click OK.
5. Select "C and C++ Development" and click Next.

6. Accept the agreement and the default install location and click Finish.
7. When prompted, restart Eclipse.

Creating and running a C application

1. Switch to the C/C++ perspective.
2. Click File > New > Managed Make C Project.

Use Standard Make if you have your own makefile. Use Managed Make if you do not have a makefile (Eclipse manages the makefile for you).

3. Name the project HelloC, accept the default options, and click Finish.
4. Select and right-click HelloC in the C/C++ Projects view, and click New > Source File.
5. Name the file main.c, and type the following code.

```c
#include<stdio.h>

int main() {
    int a = 10;
    int b = 5;

    printf("Hello C 1\n");
    printf("A = %d, B = %d\n", a, b);
    printf("Hello C 2\n");
    printf("Hello C 3\n");
    printf("Hello C 4\n");

    return 0;
}
```

6. Click Run > Run As > Run Local C/C++ Application. Select Cygwin gdb Debugger.


Debugging a C application

1. Open the main.c file.
2. Double-click the marker bar to the left of the line "printf("A=%d, B=%d\n", a, b);" to place a breakpoint. A breakpoint marker appears next to the appropriate source code line.
3. Click the Debug toolbar button.
4. There will be an error because we haven't set up the appropriate path mapping. Click the terminate button and switch back to the C/C++ perspective.

Eclipse is speaking native Windows file system names, starting with C: and such, and gdb is speaking cygwin file names, in which C: is /cygdrive/c .

5. Click Run > Run… to open up the launch configuration window.
6. Select HelloC.exe and click the Source tab.
7. Click Add… and select Path Mapping and click OK.
8. Select the new Path Mapping and click Edit…
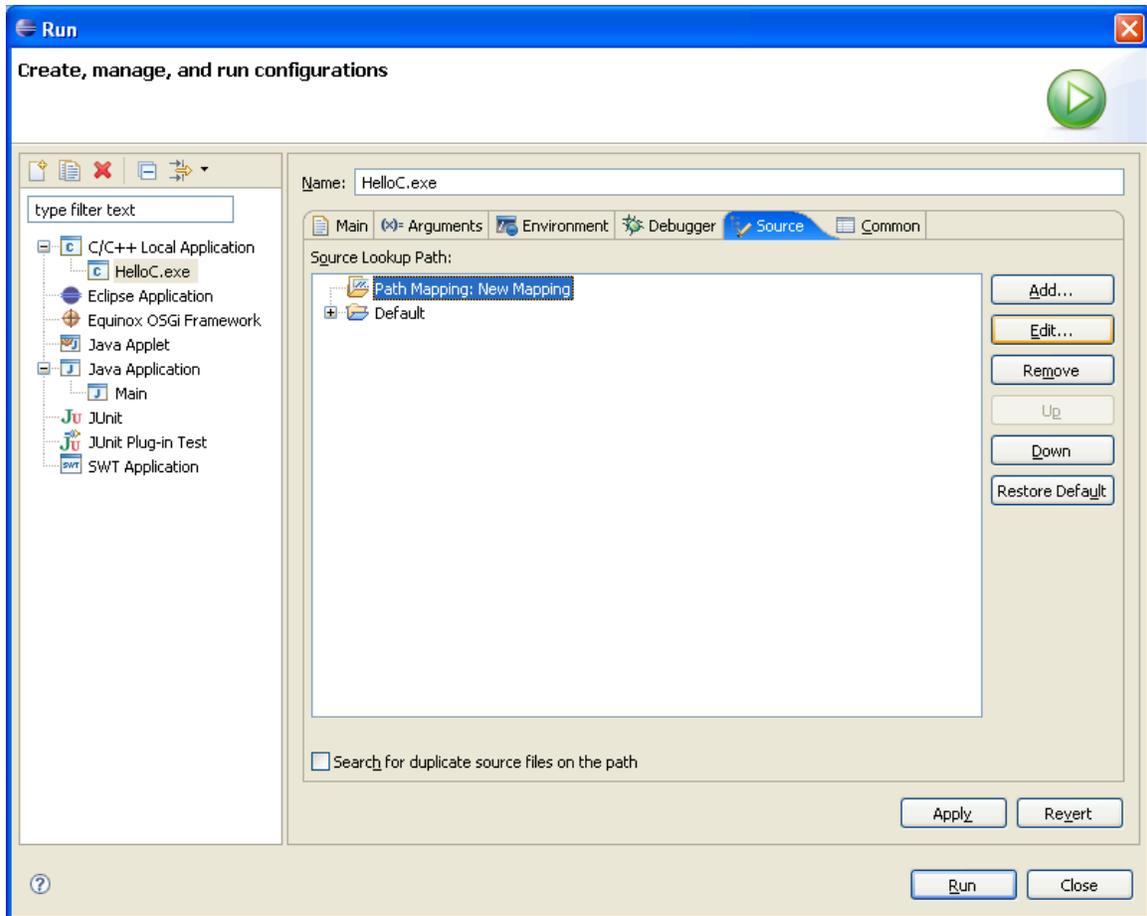
9. Click Add… to add a new path mapping
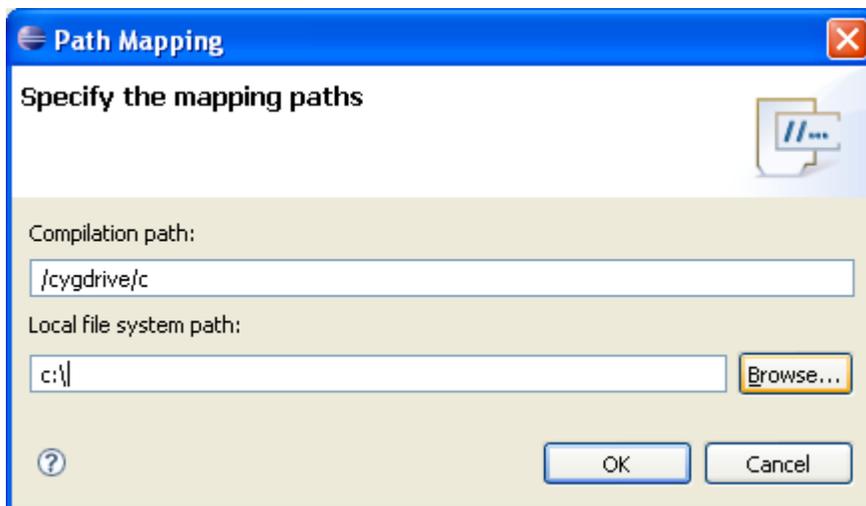
10. Click OK and click Apply to save the configuration.
11. Click the Debug toolbar button.
12. Perform various debug functions. Notice that the output is in another window.

The program input and output on Linux is redirected to the Eclipse console. This doesn't work for Windows because there is no support for GDB TTY.