

C++ Development using eclipse IDE– Starters guide

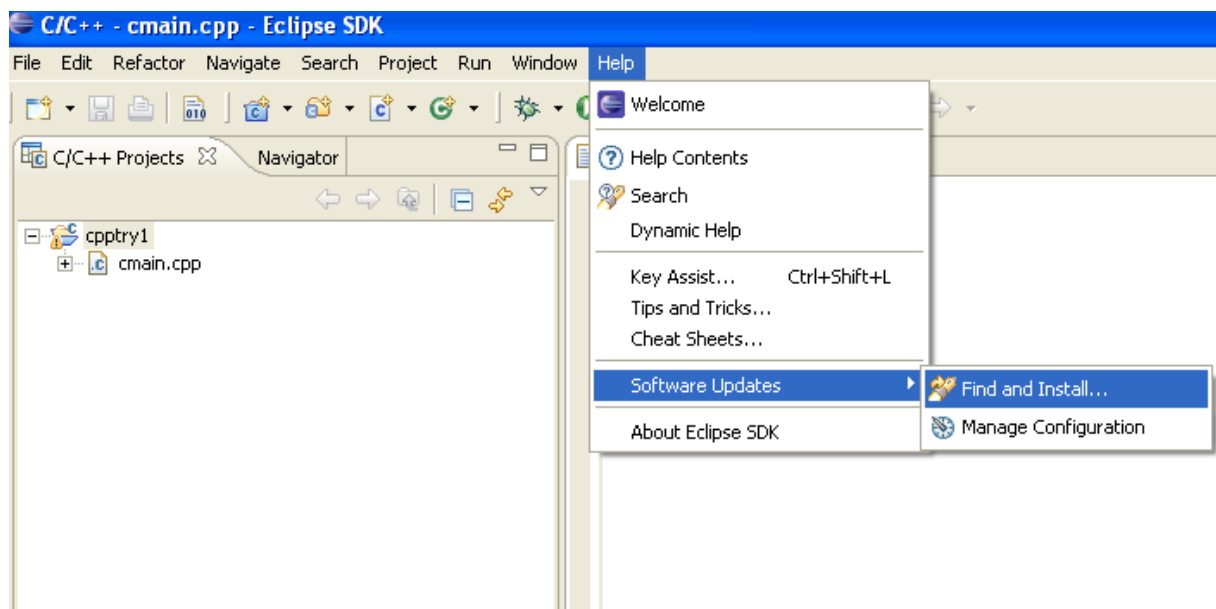
Introduction

I was a fulltime user of windows operating systems and Microsoft visual studio products for c/c++ application development. Couple of month's back I started using Linux (Ubuntu breezy). I searched for a good IDE for C++ developments in Linux and found eclipse and its plug in architecture very powerful. I decided to start using Eclipse in Ubuntu and everything worked very well. Then I decided to use the same IDE for windows as well, which will make myself more comfortable with eclipse environment. But I faced few problems while setting up a C++ development environment with eclipse and so I decided to document the same so that others also can get benefit from what I have found out.

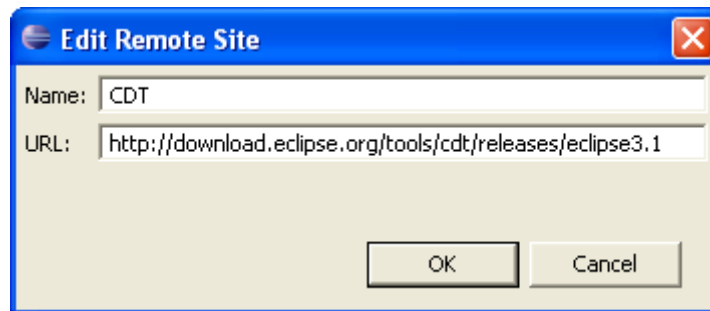
Setting up Eclipse CDT C++ IDE

Steps

1. Download and install latest java runtime environment. (You might be having one. If not install it). You can find it in www.eclipse.org.
2. Download and extract eclipse to a suitable directory. You can find this also in eclipse website.
3. Launch eclipse (It will launch without any problems if you have Java runtime environment installed) and add the CDT link in help -> Software updates -> find and install.

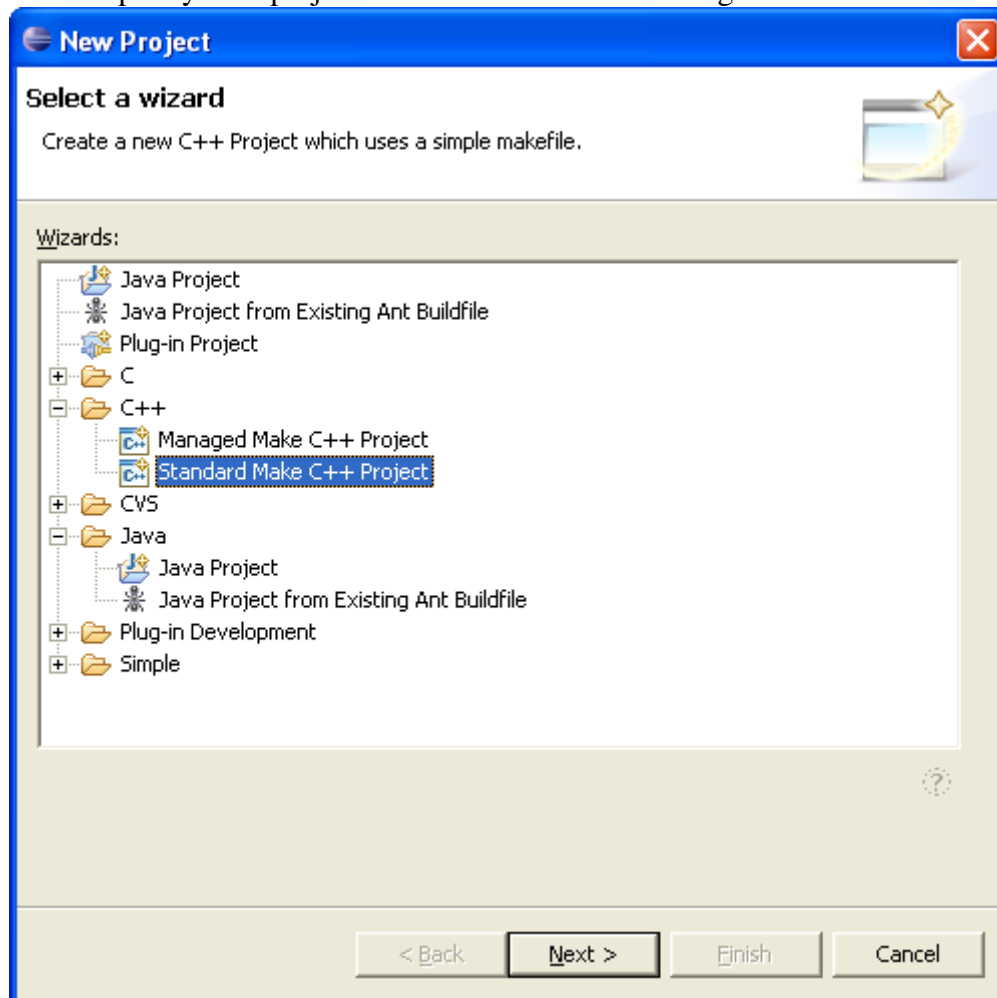


You will get a dialog where you can specify the eclipse CDT plug in path for installation.



Now eclipse will connect to the Internet location specified and install the required components for eclipse CDT. You need to be connected to Internet for this installation. Please read the document in the eclipse CDT location for offline installation. <http://www.eclipse.org/cdt/>

4. Now Launch File -> New -> Project and you will get a dialog like this where you can specify C++ project and continue with remaining stuffs.



5. Looks very easy right. Create a new project as usual write a hello world program. And compile it. It wont compile because eclipse expect you to supply the make file required for compilation. Write a simple make file as given below and keep it in the same directory where your c++ source files are kept.

```
#Make file written by boby Thomas Pazheparampil
#21-5-2006
INCLUDES = -I ./include
```

```
CC = g++ -g -Wno-deprecated
CFLAGS = -c $(INCLUDES)
all : cmain
```

```
cmain : cmain.o
        $(CC) cmain.o -o cmain
```

```
cmain.o : cmain.cpp
        $(CC) $(CFLAGS) cmain.cpp
```

```
clean :
        rm -f *.o
```

6. Now compile again, you will get a message like this “**error launching external scanner info generator /GCC -E -P -v -dD eclipse CDT**”. The reason for this error is, the GCC compiler required for compiling the source files are not present in the windows path location. You confirm this by typing “**gcc --version**” in command prompt. If you get a message saying command not found, you can be sure that GCC compiler is not available.

Additional tools required for CDT

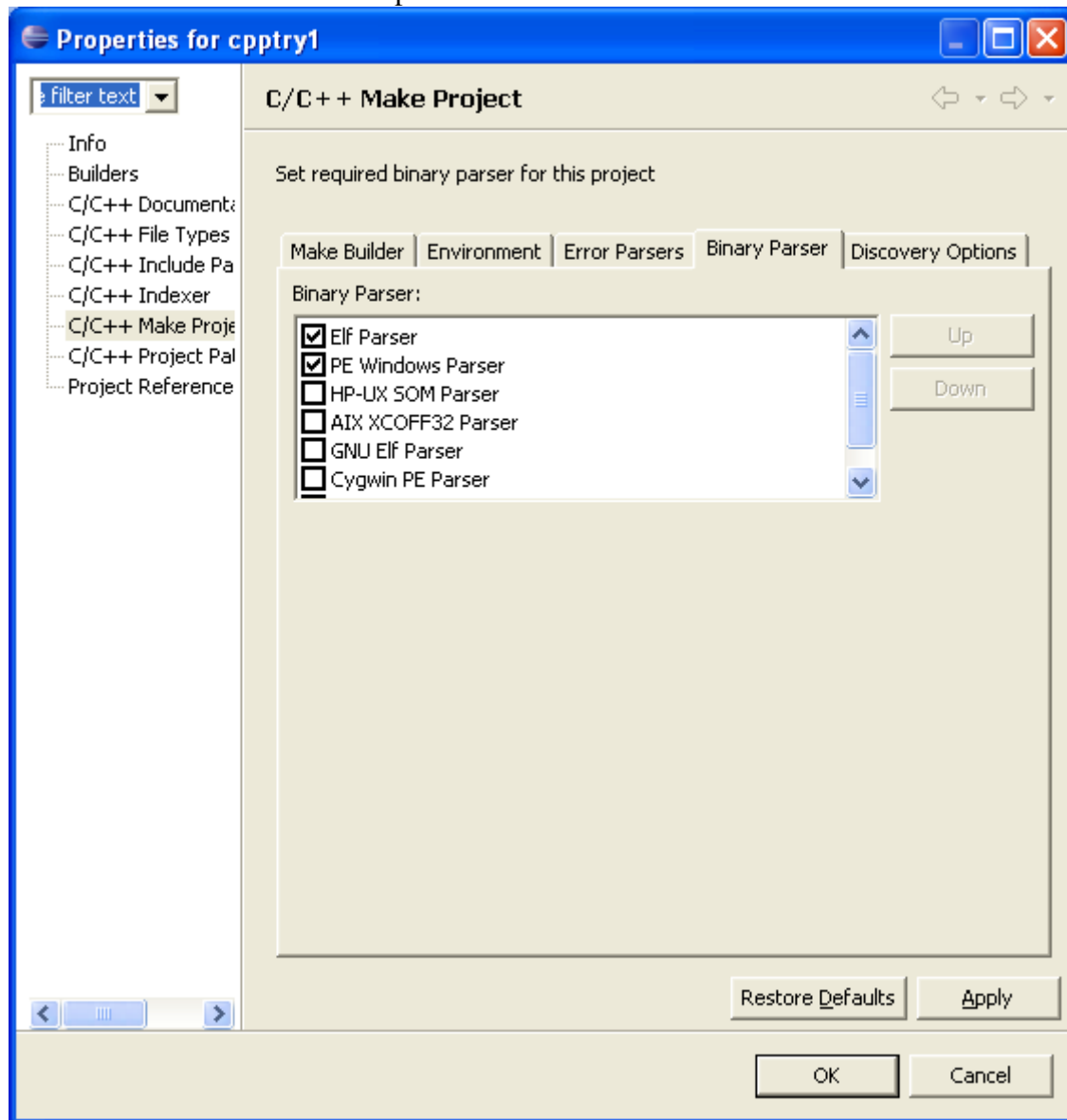
For running CDT from eclipse, you need to have the following tools.

- Make tools for make process.
- GNU compiler collection (GCC) for compilations.
- GDB for debugging.

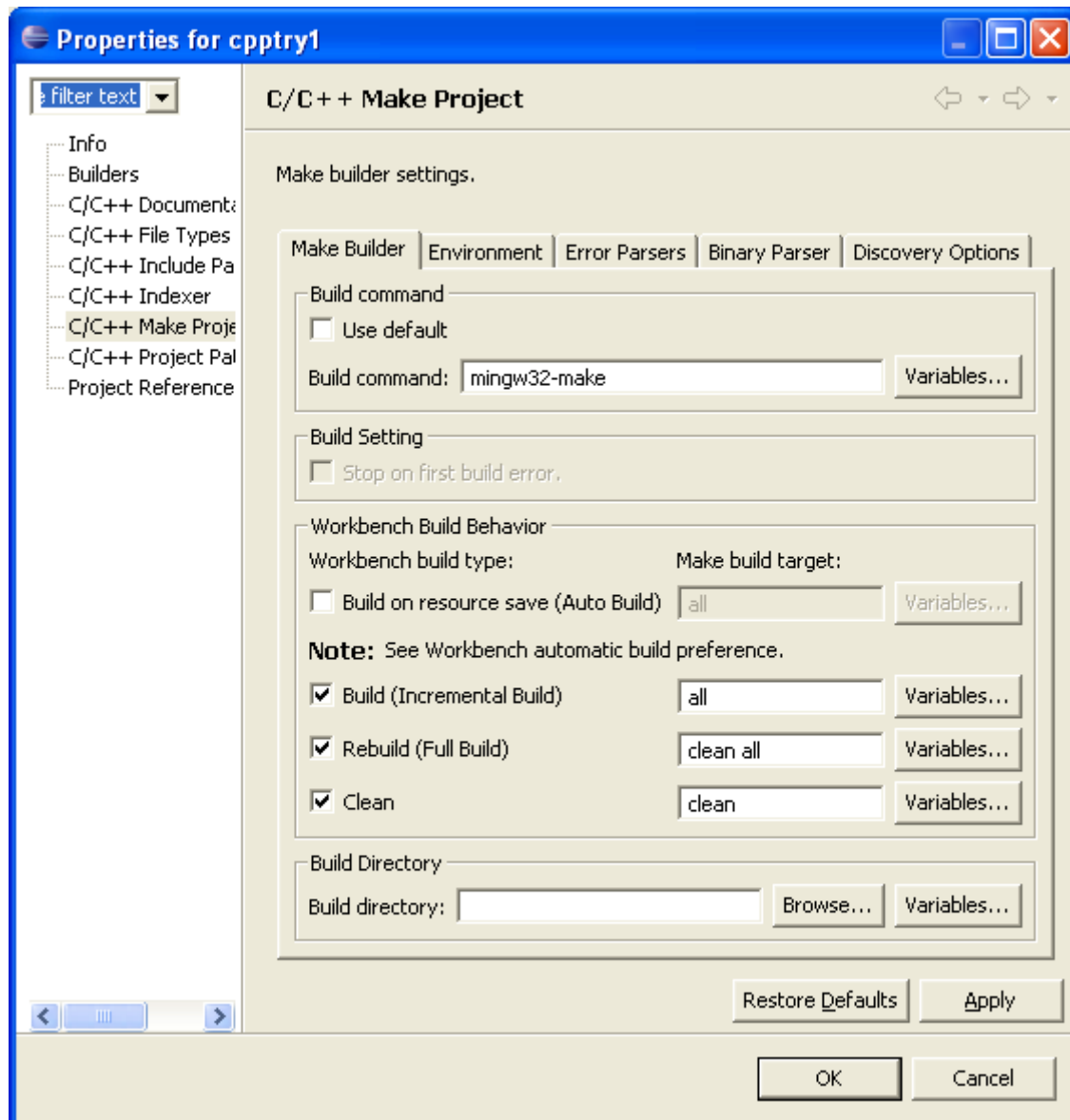
You can download all these tools from www.MinGW.com. You have to download and install build tools (make), compiler (GCC) and debugger (GDB). You can find all these tools in the location www.cygwin.com also. But I recommend the usage of mingw.

Once you have all the supporting tools available, change few settings in eclipse.

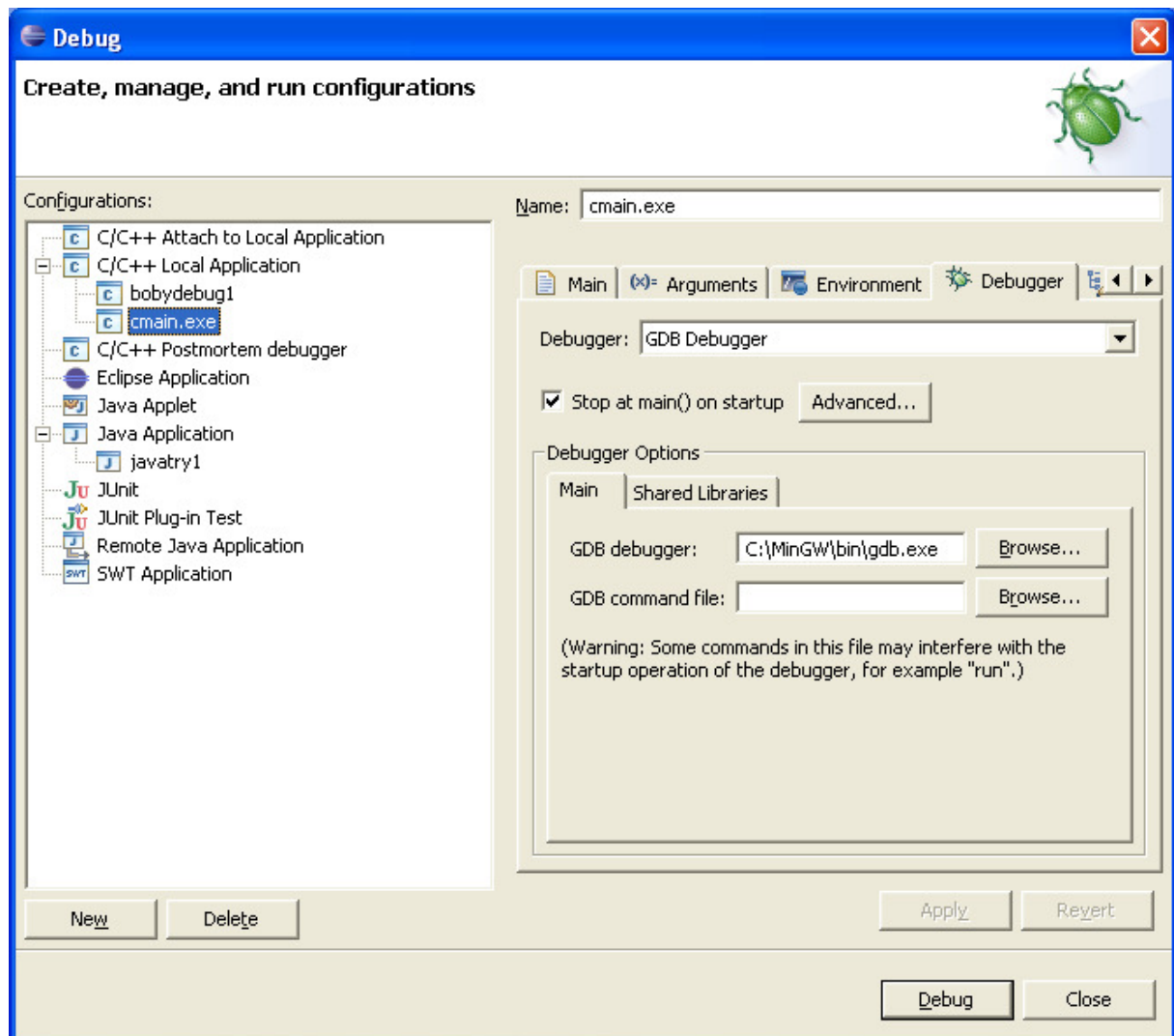
- Select “**Project->Properties->C/C++ Make Project->Binary Parser**” and select Elf Parser and PE windows parser.



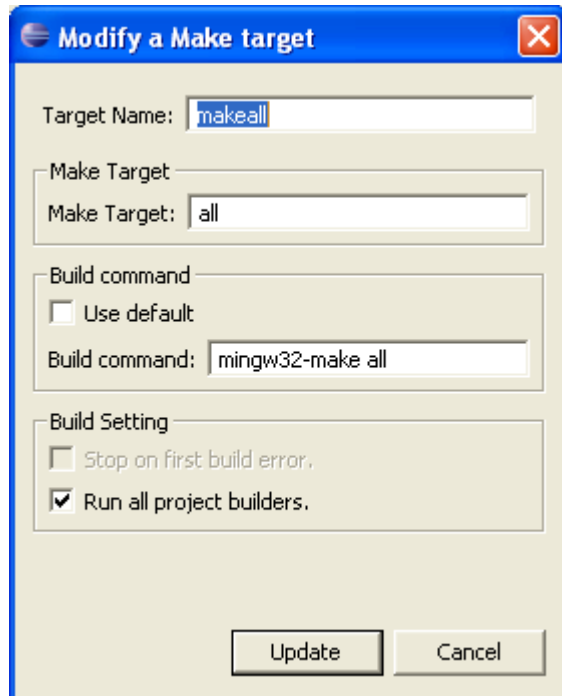
- Select “**Project->Properties->C/C++ Make Project**” and set proper debugger you have installed.



- Select “**Run->Debug->Debugger**” and select “**GDB debugger**” and locate the path to gdb debugger.



- Select “**Run->Create Make Target**” and set a proper make command you need to execute for different make operations.



Now try debugging again. It should work fine. Wish you a happy coding with eclipse.

Conclusion

Eclipse is free software available in Windows as well as Linux. C++ applications developed in eclipse can be easily taken to any other platforms very easily. Also as a developer, if you are familiar with eclipse development environment, switching from Windows to Linux is no big deal.