

在 Eclipse 中寫 C 與 C++ 程式

作者：陳建村，徐天送

Email：ctchen@pl.csie.ntut.edu.tw or s1669021@ntut.edu.tw, s2598003@ntut.edu.tw

網址：<http://pl.csie.ntut.edu.tw/~ctchen>, <http://www.ntut.edu.tw/~s2598003/>

日期：2005/9/15 (0.04 版)

修改：陳孟哲(s3419010@ntut.edu.tw)

前言

本文假設使用者已經知道如何安裝Eclipse，並且已經尚未安裝Eclipse的使用者，可至<http://www.eclipse.org> 下載。在本文撰寫時，Eclipse的最新版本為 3.1。Eclipse可在Windows與Linux環境下執行，下載檔案時請自行選擇所使用的作業系統平台檔案。這裡有一篇關於如何安裝Eclipse的中文說明，包含Windows XP及以Red Hat Linux 9 為範例的 Linux 平台，有需要的人可自行參考（PDF 格式）：
http://pl.csie.ntut.edu.tw/~sgslide/eclipse/InstallEclipseHowTo_0.05.pdf。

安裝C/C++ Development Toolkit (CDT)

Eclipse安裝好後，已經內建支援Java的開發環境（Java Development Tools, JDT），使用者只要安裝好JDK(1.3.x 版以後的Sun JDK¹)並設定好JAVA_HOME與CLASSPATH等環境變數之後，就可以直接開發Java程式。如果我們要在Eclipse上面開發C/C++的程式，則需要另外安裝plug-ins，在此我們使用的是CDT這個plug-ins。CDT可從<http://www.eclipse.org/cdt/>中下載，目前最新版本為CDT 3.0.0，並提供Runtime及SDK兩個版本，請下載適合你使用平台的CDT及版本。下圖1為下載畫面。表一為Windows平台與Linux (GTK)平台的CDT檔案名稱。

CDT Runtime Feature

includes editor, search, builders, launch, debug, gnu toolchain integrations for build/debug, user documentation

AIX/ppc	org.eclipse.cdt-3.0.0-aix_ppc.tar.gz
Linux/x86	org.eclipse.cdt-3.0.0-linux_x86.tar.gz
Linux/x86_64	org.eclipse.cdt-3.0.0-linux_x86_64.tar.gz
Win32/x86	org.eclipse.cdt-3.0.0-win32.x86.zip

CDT SDK Feature

superset of runtime that adds source and extension point schemas

AIX/ppc	org.eclipse.cdt.sdk-3.0.0-aix_ppc.tar.gz
Linux/x86	org.eclipse.cdt.sdk-3.0.0-linux_x86.tar.gz
Linux/x86_64	org.eclipse.cdt.sdk-3.0.0-linux_x86_64.tar.gz
Win32/x86	org.eclipse.cdt.sdk-3.0.0-win32.x86.zip

圖1：CDT 下載畫面

¹ Eclipse似乎也可搭配IBM或其他公司的JDK，不過我並沒有實際測試過。

平台	Type	檔名
Windows	Runtime	org.eclipse.cdt-3.0.0-win32.x86.zip
	SDK	org.eclipse.cdt.sdk-3.0.0-win32.x86.zip
Linux	Runtime	org.eclipse.cdt-3.0.0-linux.x86.tar.gz
	SDK	org.eclipse.cdt.sdk-3.0.0-linux.x86.tar.gz

表一：Windows 與Linux平台之CDT 3.0.0 檔名

下載完CDT之後，就可以開始安裝。安裝的方式非常簡單（Windows 與Linux平台安裝方式都一樣），在此以Windows系統下的SDK的版本為例，將下載的檔案解壓縮後，會發現features與plugins這兩個目錄，請參考圖2。分別將這兩個目錄中的所有檔案，直接使用檔案總管複製到當初安裝Eclipse時所在的目錄中相對應的目錄即可。例如，假設Eclipse安裝在C:\eclipse目錄中，剛剛的檔案解壓縮在C:\CDT目錄中，則將C:\CDT\features 目錄中所有的檔案複製到C:\eclipse\features 目錄中，將C:\CDT\plugins目錄中所有的檔案複製到C:\eclipse\plugins中就完成了安裝²。

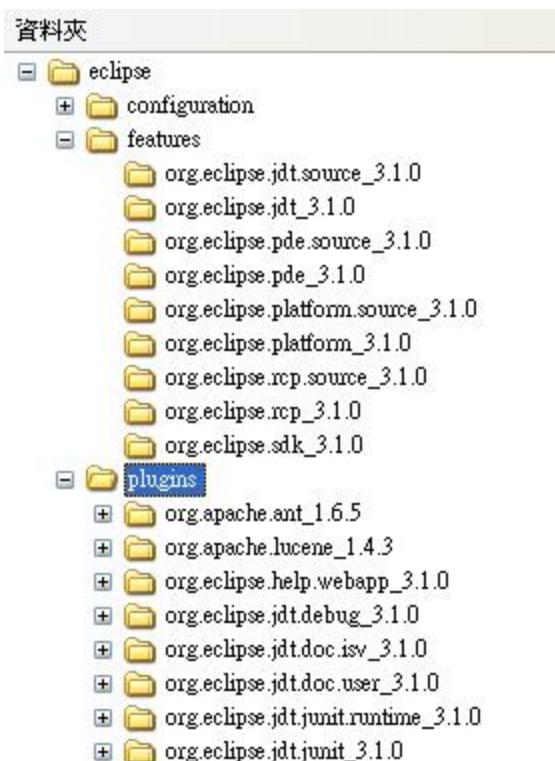


圖2：解壓縮後的CDT plug-in目錄結構

² Eclipse的plug-in安裝方法幾乎都一樣，把檔案下載回來之後，直接解到安裝eclipse所在目錄的features或plugins目錄中即可。如果您是在Linux系統下安裝，請記得以root身分執行上述複製檔案的動作。

使用Eclipse下的Software Updates安裝C/C++ Development Toolkit (CDT)

我們也可以使用Eclipse所提供的軟體更新下載的功能來安裝CDT，在選單【Help】中的【Software Updates】選擇【Find and Install】，請參考圖3的畫面，接下來Eclipse將會開啟一個Install/Update精靈，協助我們安裝或更新plug-ins。我們即將開始安裝CDT，接下來的步驟將詳細說明安裝的過程(相關資訊可參考<http://www.eclipse.org/cdt/>)。



圖3 選擇Eclipse的軟體更新

首先Install/Update精靈會詢問你要更新已經安裝過的features(features是一群相關的plug-ins所共同組成的完整應用)，或是安裝全新的features，以我們而言選擇後者，如圖4所示，我們選擇【Search for new features to install】，然後按下【Next】。

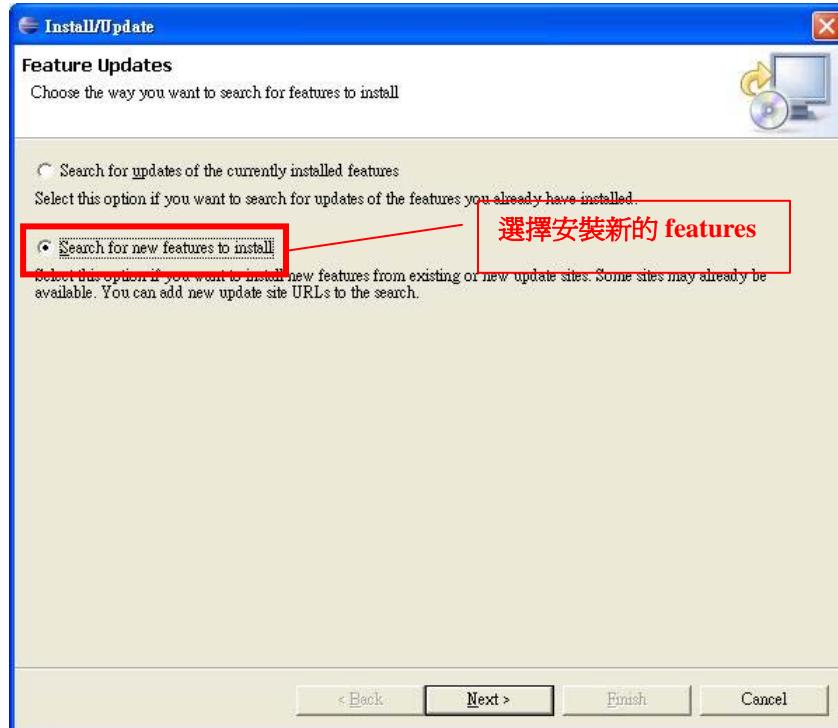


圖 4 Install/Update 精靈畫面(選擇安裝新的 features)

接下來我們必須選擇一個Update Site，我們有三個來源可供安裝或更新plug-ins，第一個是Remote Site，為遠端提供plug-ins下載更新的網站；第二個是Local Site，為已經解壓縮的plug-in目錄；最後一個則是Archived Site，為下載下來的plug-in壓縮檔。若我們要安裝CDT，我們選擇New Remote Site，則會出現一個對話框，輸入Name為CDT，URL為<http://download.eclipse.org/tools/cdt/releases/eclipse3.1>（參考<http://www.eclipse.org/cdt/>的安裝說明）。畫面如圖 5 所示。

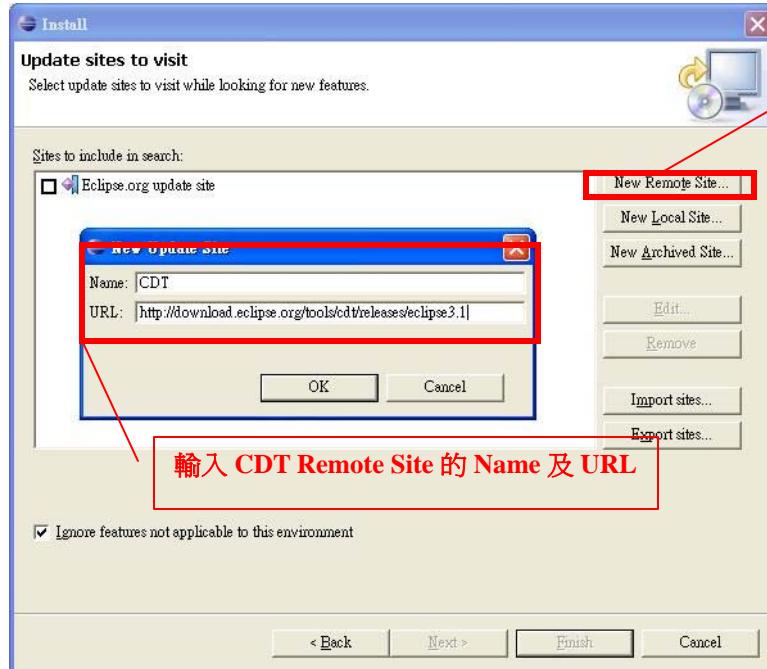


圖 5 Install/Update 精靈畫面(輸入 Update Site 的 Name 及 URL)

新增完一個 Remote Site 後，按下 Finish 按鈕則會有 Updates 視窗列出 CDT 的節點及版本資訊，我們選擇 CDT 3.0.0 版安裝，如圖 6 所示，再按下【Next】進行下一個步驟。

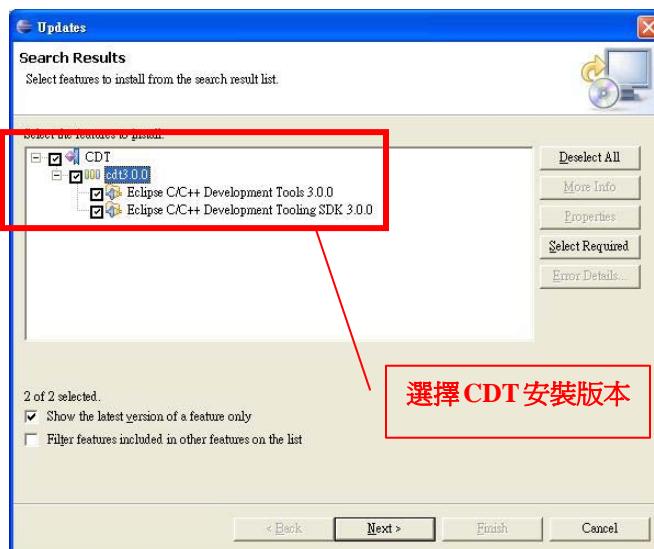


圖 6 Install/Update 精靈畫面(選擇安裝版本)

圖 8 為版權宣告說明，說明 CDT 是 EPL 的 license，若同意的話就選擇【Accept】後，再按下【Next】進行下一步。

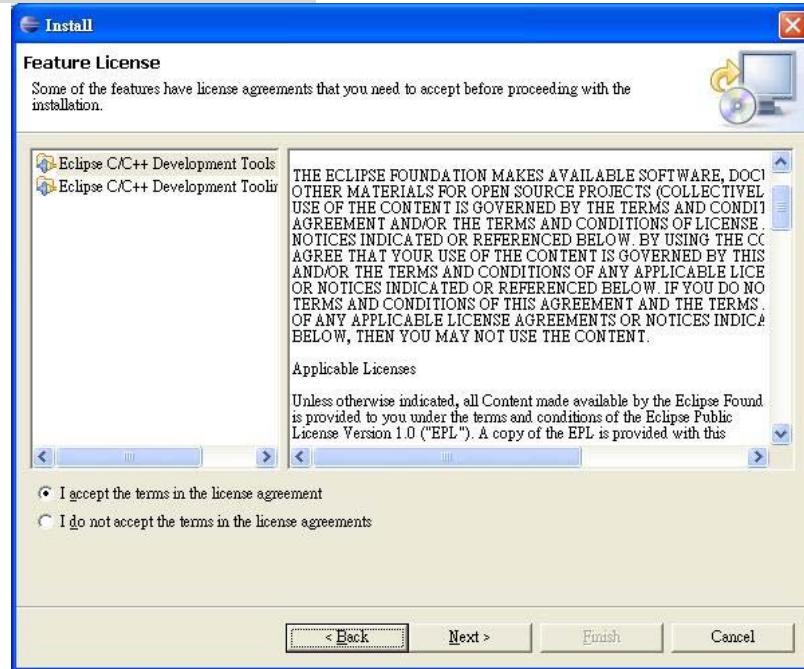


圖 7 Install/Update 精靈畫面(License Agreements)

接下來，為選擇即將安裝的 Eclipse 目錄，以我們的例子而言，我們預設為 C:\Documents and Settings\Administrator\桌面\eclipse，若要安裝至系統其他的 Eclipse(你可以同時安裝不同版本的 Eclipse)，可以選擇 Change Location，然後指定其他 Eclipse 安裝目錄即可。在這個步驟，我們直接按下【Next】，如圖 8 所示。

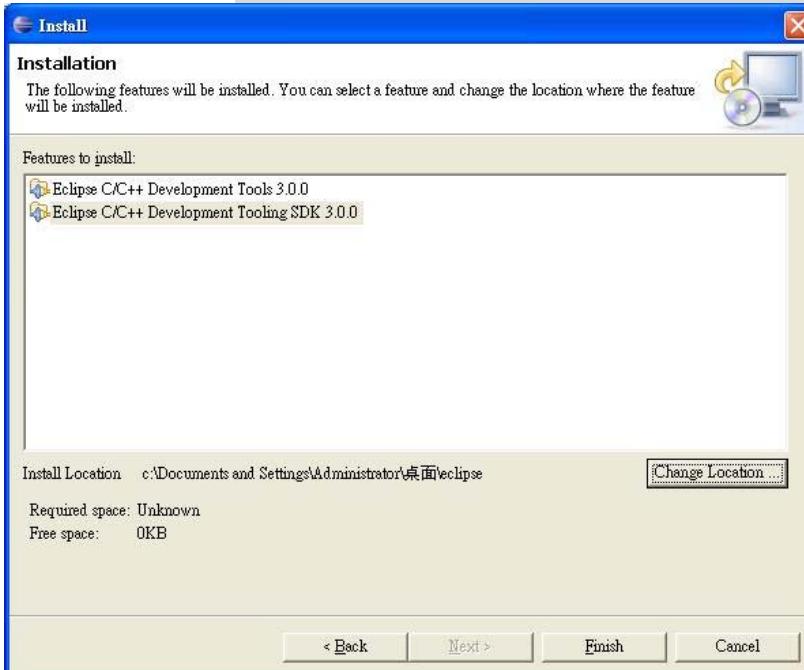


圖 8 Install/Update 精靈畫面(選擇要安裝的 Eclipse 目錄)

最後，在安裝前會跳出一個列表顯示我們即將安裝的 features，如圖 9 所示，確認後直接按下【Install All】進行安裝。

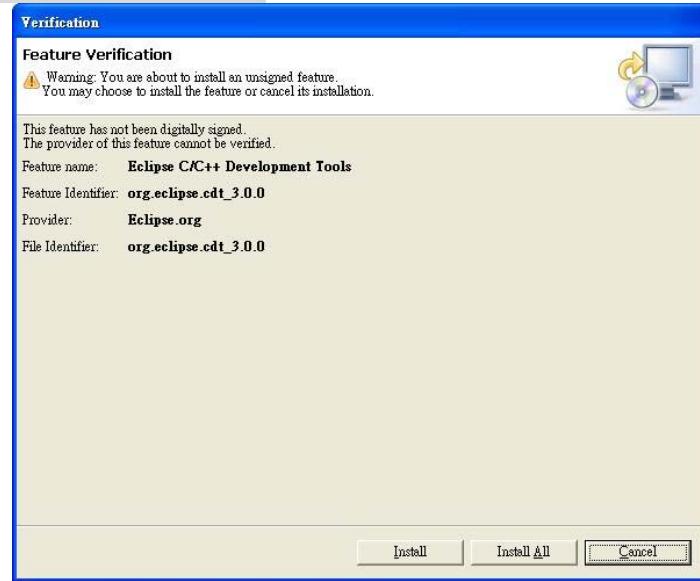


圖 9 Install/Update 精靈畫面(安裝前最後確認)

安裝完畢後，Eclipse 會詢問是否要重新啓動 Eclipse，我們選擇【Yes】後，重新啓動 Eclipse，如圖 10 所示。

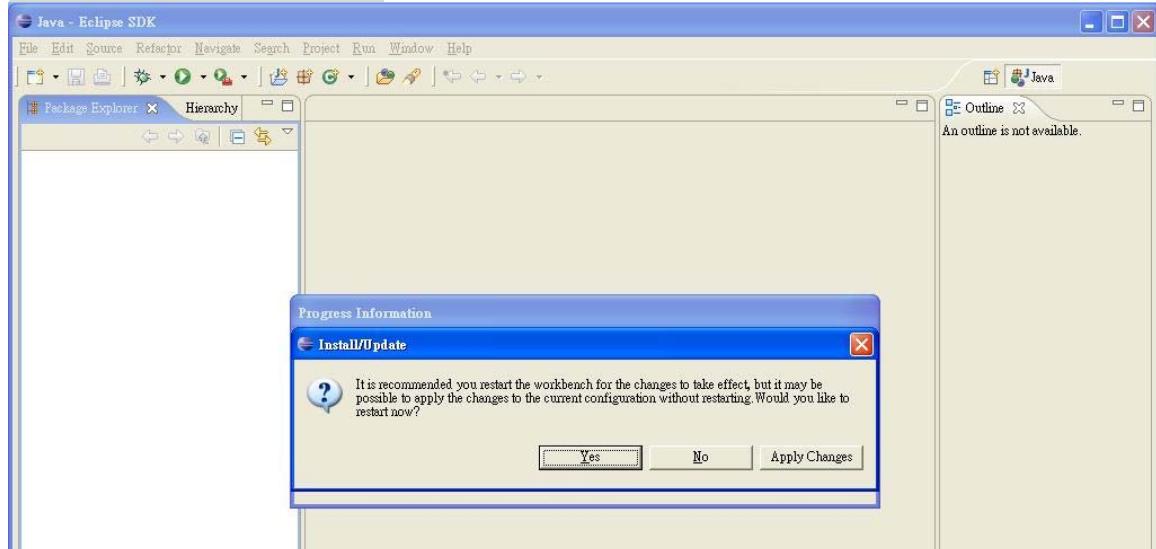


圖 10 安裝完畢，Eclipse 詢問是否重新啓動

待 Eclipse 重新啓動後，我們可以檢查我們剛剛所安裝的 plug-ins，如圖 11、12、13 所示。我們可以選擇選單【Help】下的【About Eclipse SDK】後出現一個關於 Eclipse 版本的對話框，由於我們剛剛安裝的是屬於 features，我們選擇【feature Details】，按一下後會跳出另一個視窗，顯示目前 Eclipse 所有安裝的 features 及其版本等資訊，當我們找到關於 CDT 的部份，即表示安裝成功。未來若是有安裝任何其他新的 plug-ins，都可在這裡作確認的動作。

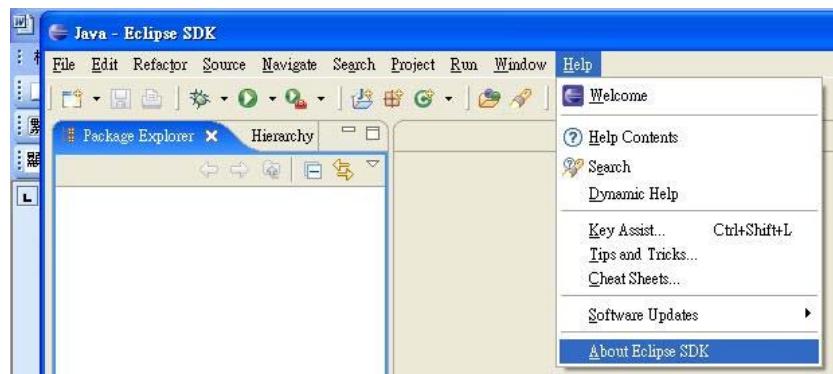


圖 12 檢查安裝的 plug-ins(步驟 1)



圖 13 Eclipse 版本說明(步驟 2)

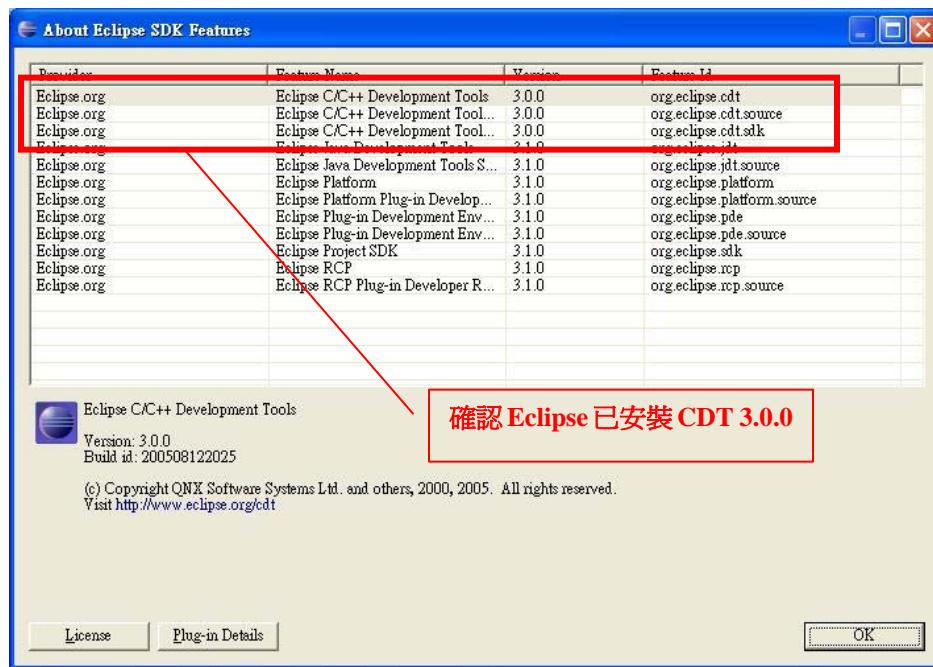


圖 14 Eclipse 目前已安裝的 features 列表(步驟 3)

在Windows系統上寫C/C++ 版本的HelloWorld

CDT原本的目的，是提供一個在Linux系統中，結合Eclipse的C/C++開發環境。CDT需要與gcc、make、gdb等GUN工具配合使用，它本身並沒有提供C/C++ compiler。所以，安裝好CDT之後，只是讓我們有一個整合的環境可以在Eclipse開發（編輯、除錯與執行）C/C++程式，但是真正要讓CDT動起來，還是需要gcc等GUN工具。然而，Windows系統中並沒有gcc 可使用，所幸已有人將gcc移植到Windows系統中。根據CDT網站上的資料³，有兩個工具可以讓我們在Windows上使用GUN的軟體：

- [Cygwin](#)
- [MinGW](#)

目前在這個部份只需要MinGW來替我們編譯程式碼，而Cygwin則是在Windows下提供類似Unix環境，目前暫不需要。所以，接下的說明以MinGW為例子。

安裝MinGW

本文所撰寫時，最新的MinGW版本為MinGW 3.1.0-1，檔名為MinGW-3.1.0-1.exe，下載後直接安裝即可，請參考圖15。我們把WinGW安裝在C:\MinGW目錄中。



圖15：安裝WinGW畫面

安裝好後，需要把C:\cygwin\bin 加入環境變數中的PATH，以便讓Eclipse可以呼叫到裡面的GUN Tools。設定環境變數的方法，在此以Windows XP系統為例說明（在Windows 98/ME 中應該是設定在autoexec.bat檔案中）。請參考圖16，用滑鼠右鍵點選桌面上的「我的電腦」圖示，然後選取【內容(R)】，之後會出現如圖17之畫面，請點選【環境變數(E)...】按鈕，之後出現圖18。在圖16中的「系統變數(S)」中找到Path這個項目，按下【編輯(I)...】按鈕出現圖19。將C:\cygwin\bin加到原有設定資料的最後面即可。請注意，每個設定之間要用;（分號）隔開。

³ CDT User FAQ，然而這份文件目前並不完整，上面的說明也不是很清楚：

http://dev.eclipse.org/viewcvs/index.cgi/~checkout~/cdt-home/user/faq.html?cvsroot=Tools_Project

然而，為了讓Eclipse能自動執行make⁴，我們必須提供一個make程式，否則Eclipse因無法找到make而無法編譯。最直接解決問題的方式是我們提供一個名為make的程式即可解決，而MinGW也有提供make程式，但其檔名卻為mingw32-make.exe，因此我們只須將「C:\MinGW\bin\mingw32-make.exe」更名為「make.exe」即可，如圖20所示。

設定好之後，要如何確定剛剛的設定是正確的呢？我們可以在DOS視窗下輸入gcc，如果出現「gcc: no input files」，就表示剛剛的路徑設定是正確的並且已經生效。這時再輸入make，如果出現「**make: *** No targets specified and no makefile found. Stop.**」，就表示make指令也沒問題。如果都沒有，剛剛輸入的資料也正確，那就試看看重新開機。(請注意，設定完之後Eclipse必須重新開啟一次，以確保之前的環境設定能套用至Eclipse上)

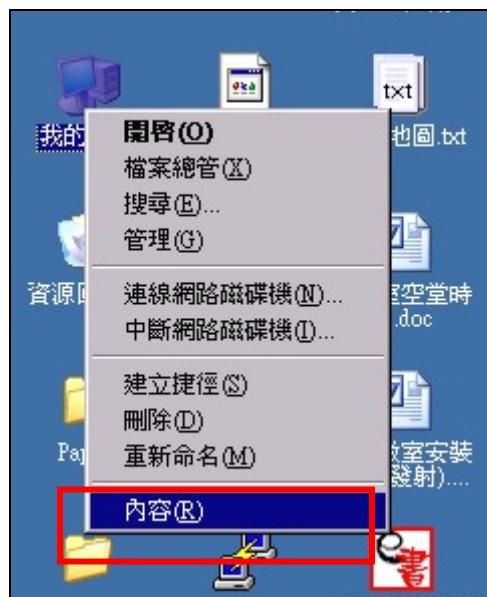


圖16：用滑鼠右鍵點選「我的電腦」

⁴ 若是以寫Make檔案來編譯，預設是以系統的Make程式來執行，以Windows的VC而言，是一個名為nmake的程式。但是，Eclipse必須直接指定執行的程式，而Eclipse預設為make。

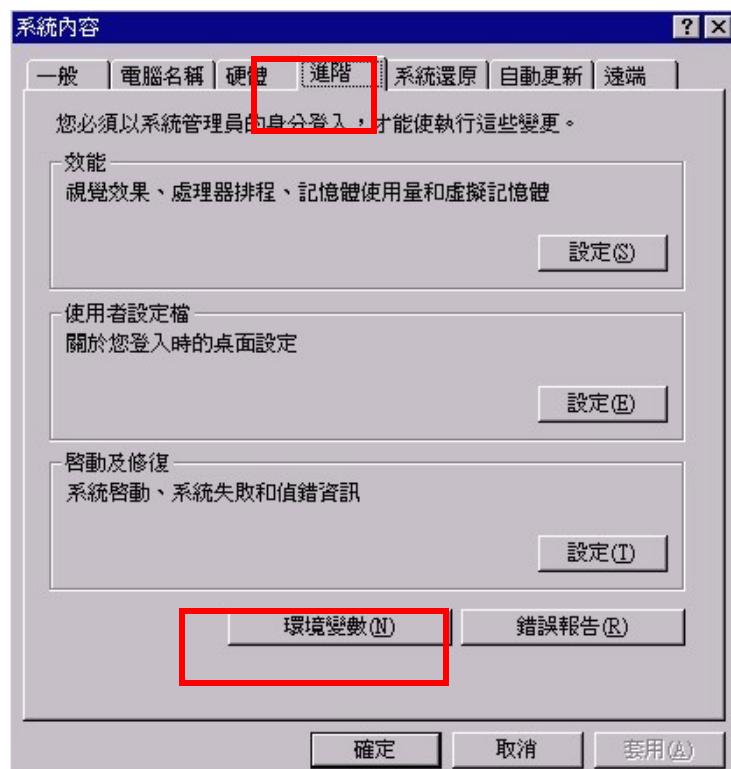


圖17：按下【環境變數(E)...】按鈕

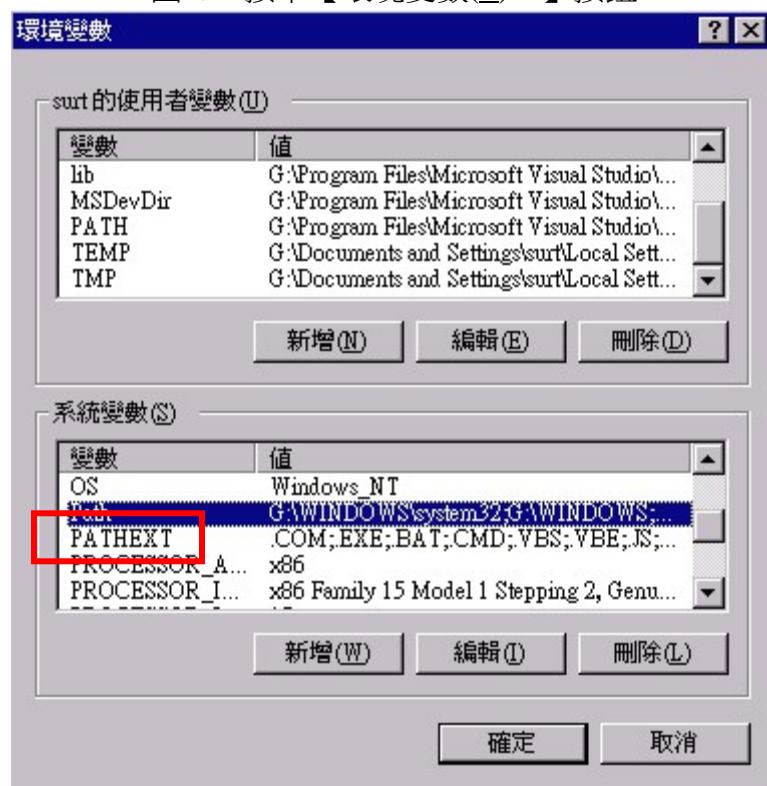


圖18：選取Path變數



圖19：將C:\MinGW\bin加入Path變數中

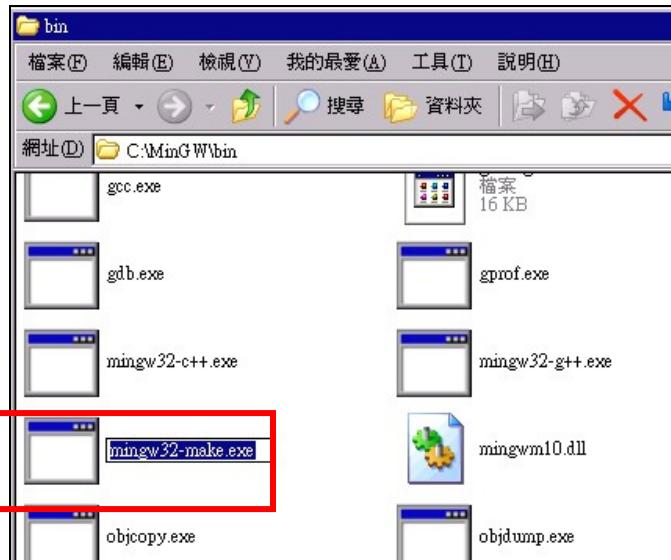


圖20：將C:\MinGW\bin目錄下的mingw32-make.exe改名為make.exe

在Eclipse中新增C/C++ Project(Managed Make C++ Project)

由於CDT 2.0多了一項新的功能，允許我們建立一個Managed Make C++ Project，也就是說，讓Eclipse替我們管理Make檔案，這功能在之前的CDT 1.x版是不支援的。假設Eclipse與CDT和MinGW都已經正確安裝完成⁵，現在我們可以寫第一個C++的程式: Hello World。在Eclipse的選單中選擇【File】→【New】→【Project】，會出現類似圖21之畫面，此時我們選擇「Managed Make C++ Project」⁶，按下【Next】按鈕，出現圖22之畫面。在「Name」中輸入Hello C++(此設定值為這個project的名稱)。按下【Next】按鈕，出現圖23之畫面，在這個畫面我們可以選擇最後產生出來的程式是Executable、Shared Library或是Static Library的。也可選擇要以Debug、Release或是兩者都產生，在這兒我們以預設值即可，即直接按下【Next】按鈕。之後便出現圖24之畫面，此時按下【Finish】按鈕即可產生一個新的(空的)專案。

在專案名稱上按下滑鼠右鍵，選【New】→【File】以便新增一個C++程式，請參考圖25。選取新增檔案之後，出現如圖26的畫面，在「File name」中輸入hello.cpp之後按下【Finish】按鈕。

⁵ Eclipse本身也是用Java寫的，所以要執行Eclipse，還需要安裝Sun JDK 1.3.x以後的版本。

⁶ 你也可以選擇「Standard Make C++ Project」，但這必須自己撰寫Make檔案。

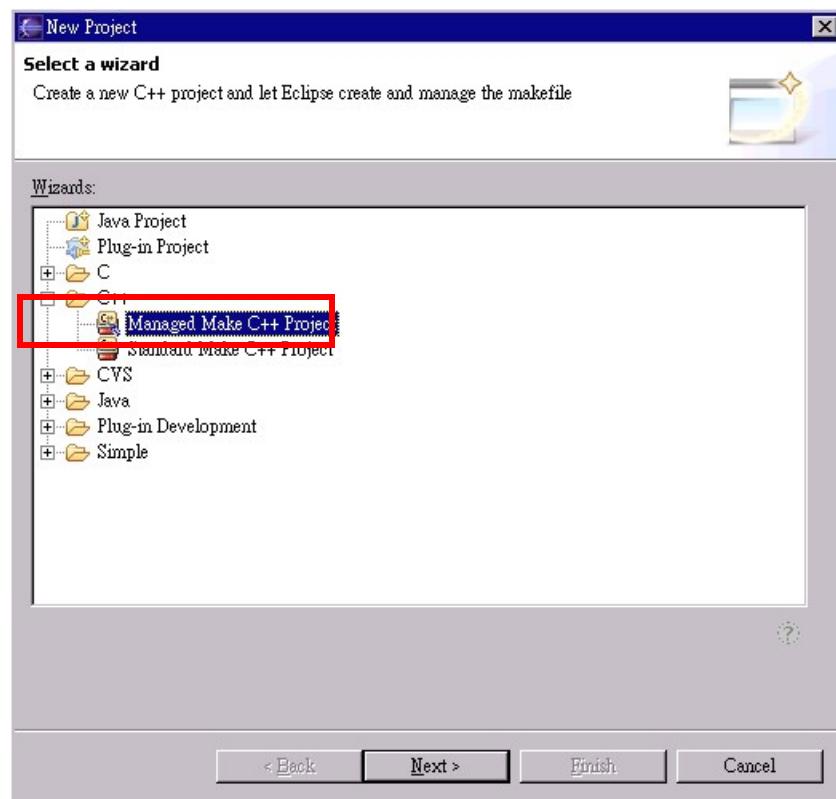


圖 21：新增專案畫面(Managed Make C++ Project)

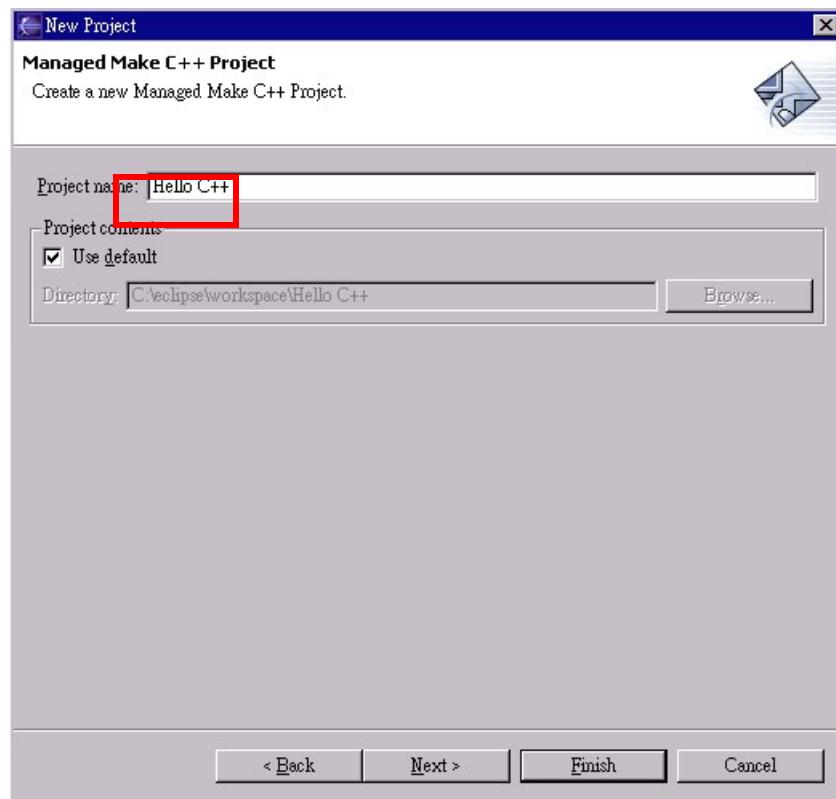


圖 22：輸入專案名稱為 Hello C++

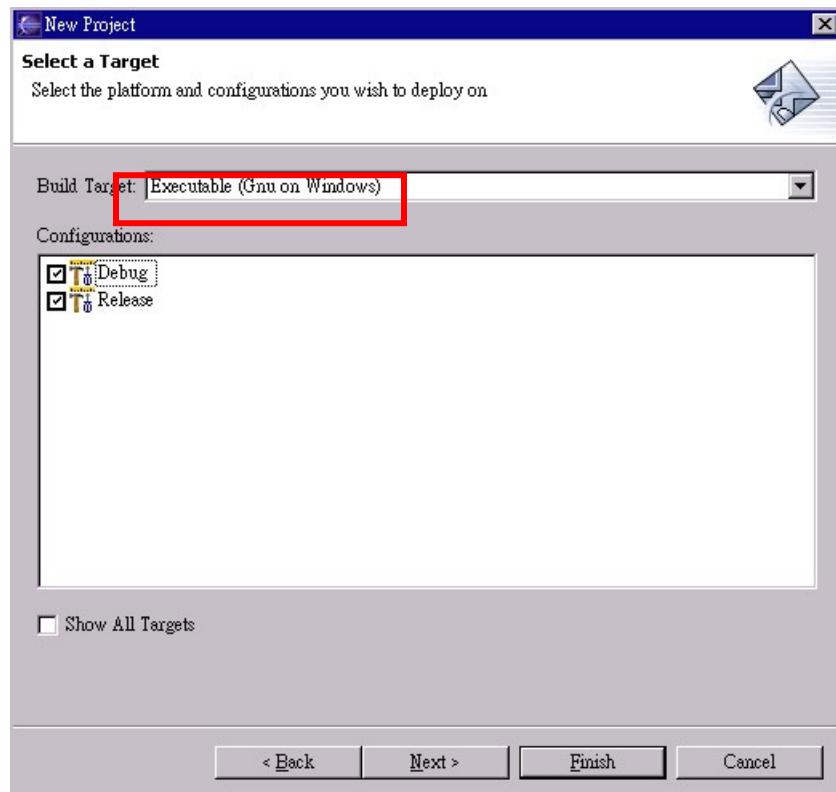


圖23：選擇產生程式之Link模式及Debug、Release的設定

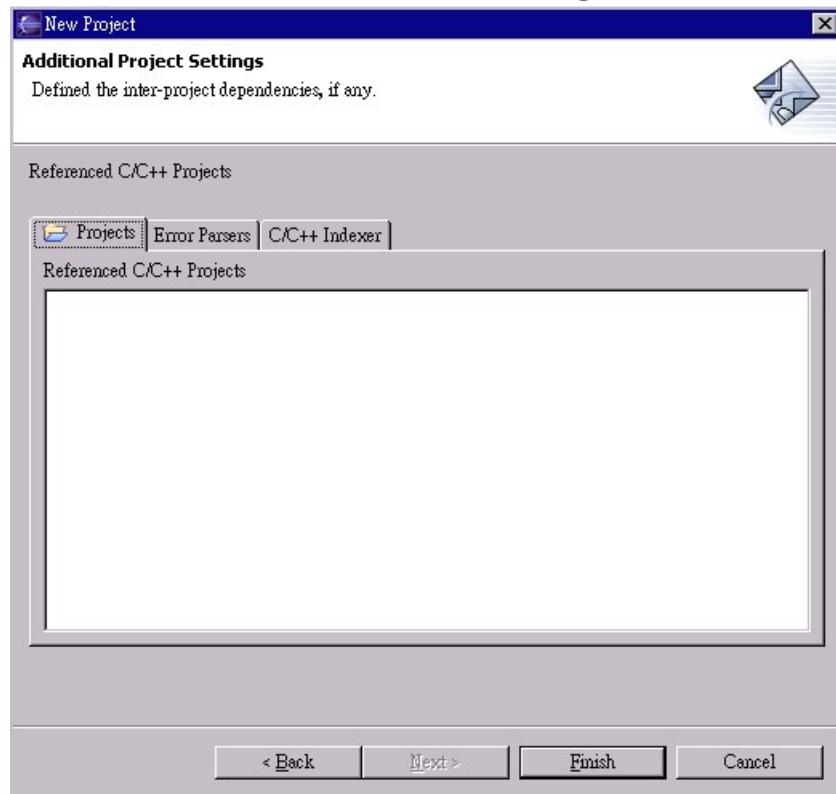


圖 24：按下「Finish」按鈕以完成新增專案

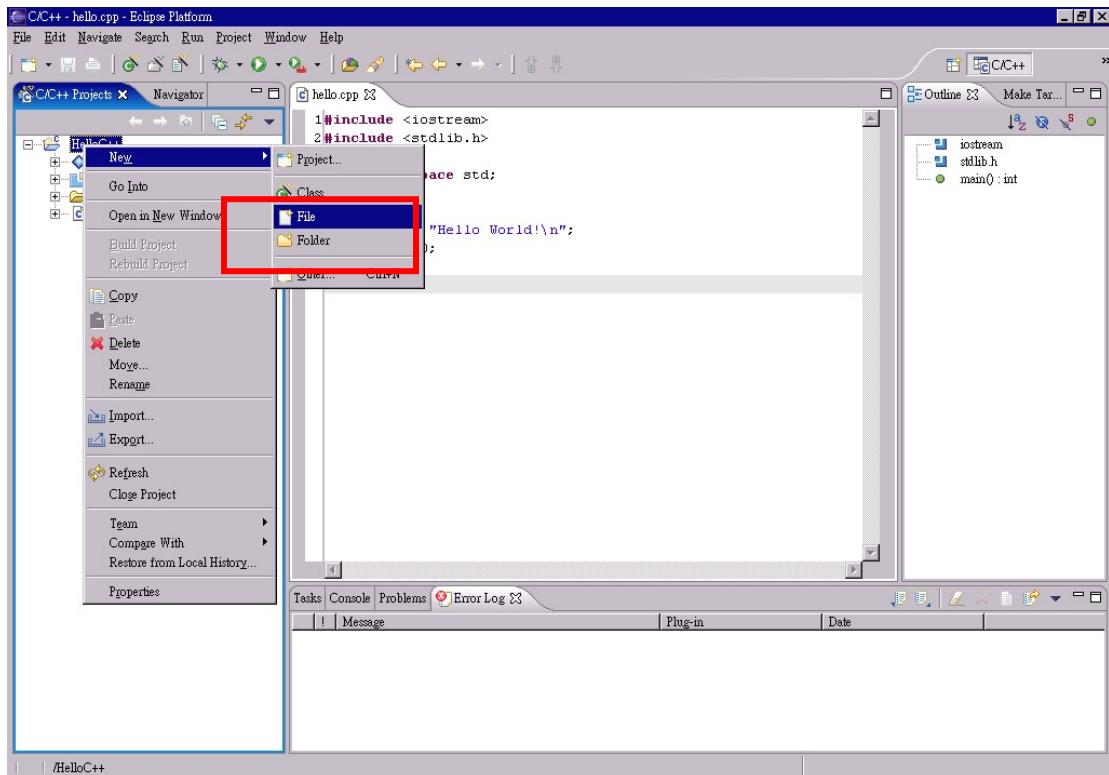


圖 25：新增一個檔案

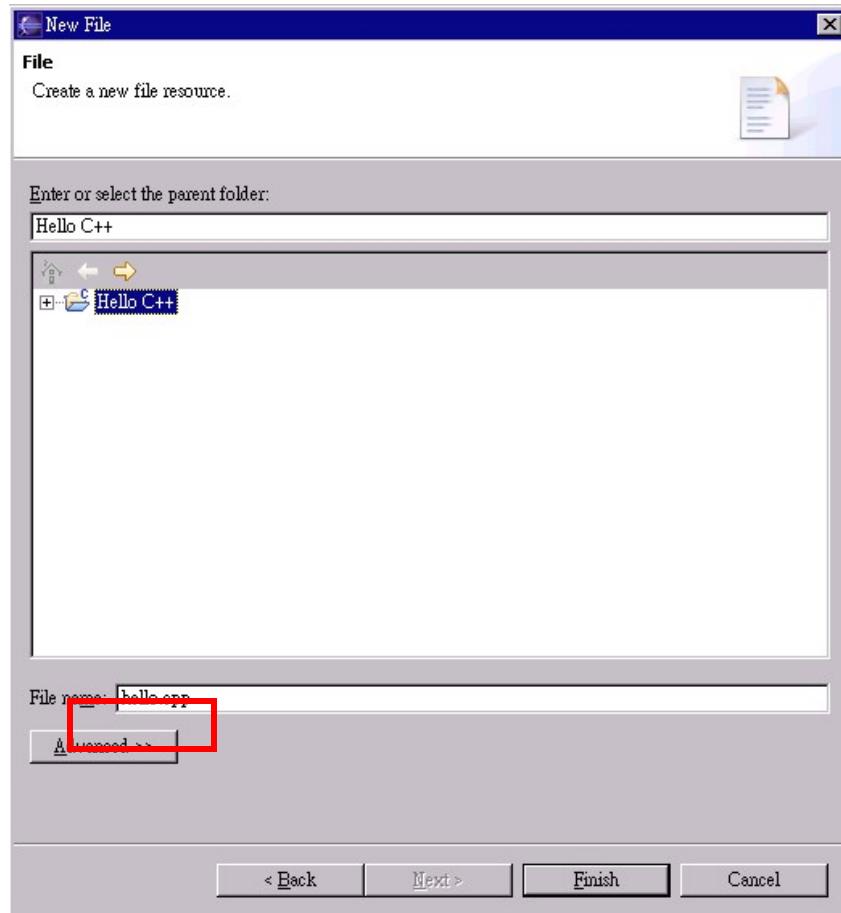


圖 26：輸入檔案名稱

把下列程式輸入到hello.cpp 中：

```
#include <iostream>
#include <stdlib.h> ▶

using namespace std;

int main() {
    cout << "Hello World!\n";
    return 0;
}
```

注意，是iostream，不是iostream.h
如果使用iostream.h在編譯程式時
會看到一堆警告訊息

寫好之後，將檔案存檔。Eclipse會在我們存檔時自動幫我們編譯程式，如果一切都正常，會看到類似圖27的畫面。接下來，我們要執行該程式，請從【Run】選單中選擇【Run...】（【Run】→【Run...】），此時會出現如圖28之畫面。請參考該畫面中的資料，新增一個C/C++ Local Configurations。新增完後，按下【Run】按鈕就可以執行該程式。圖29為程式執行結果，經過一番奮鬥之後，終於完成了最簡單的Hello World。

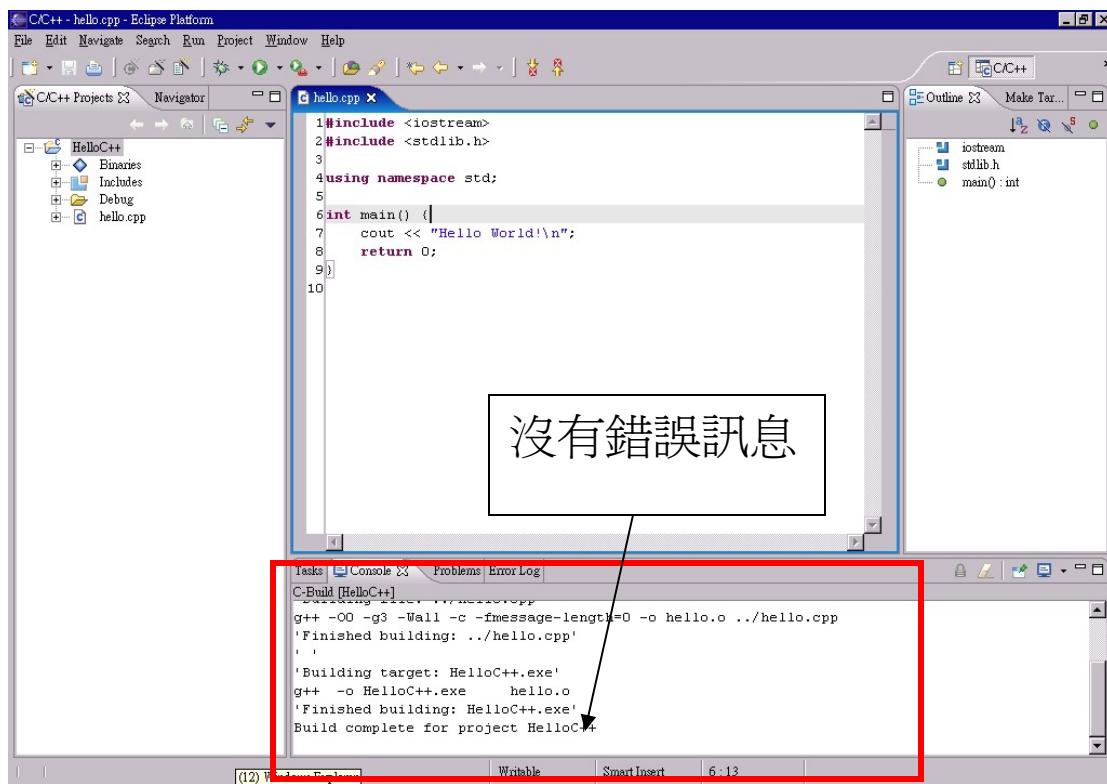


圖 27：程式編譯正常

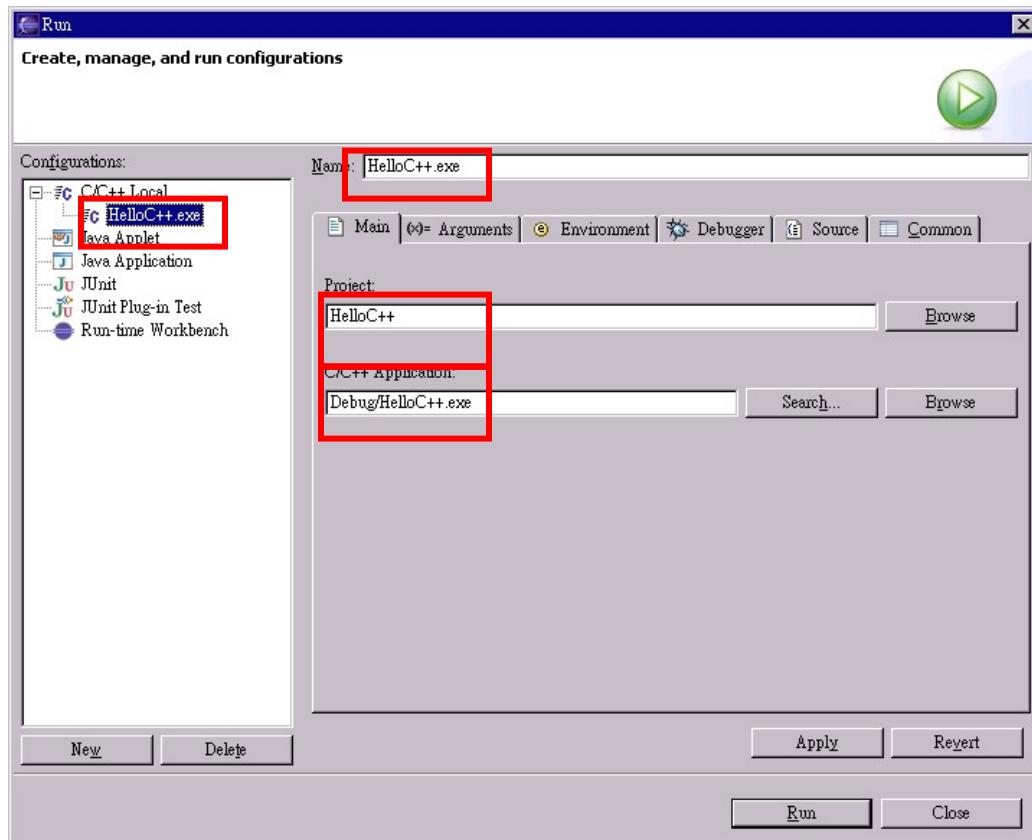


圖 28：新增一個 Run Configuration

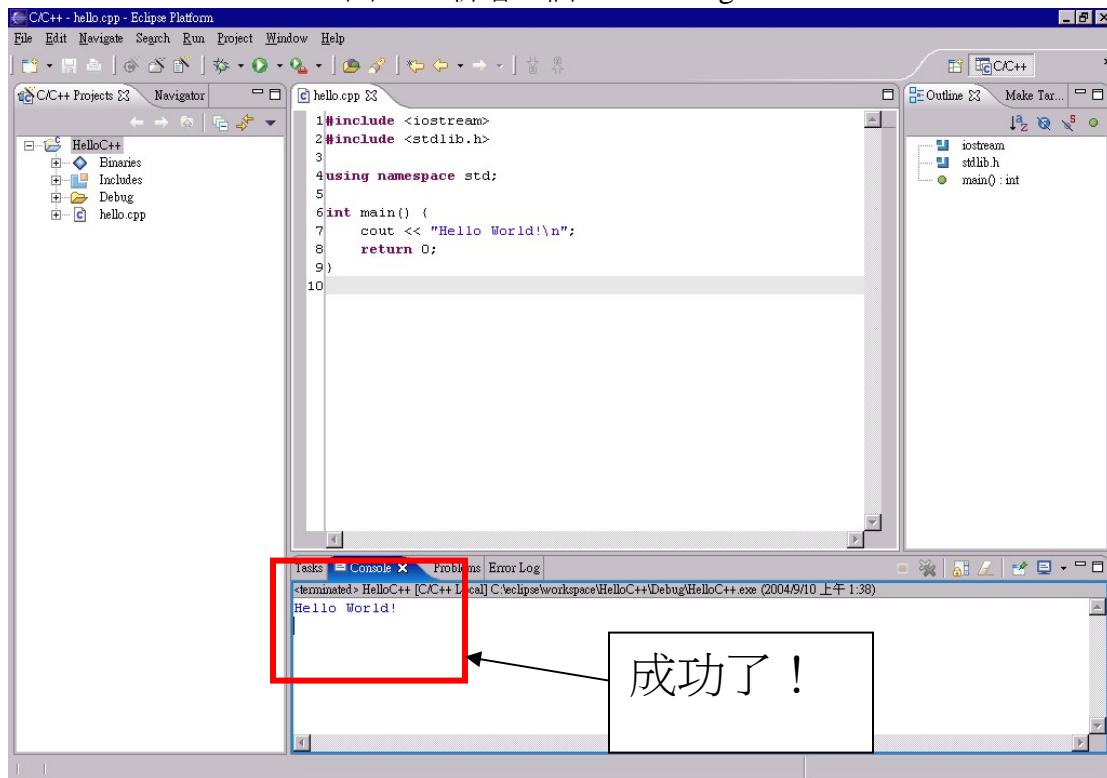


圖 29：Hello.exe 執行結果

在Eclipse中新增C/C++ Project(Standard Make C++ Project)

另一個Build C++程式的方式是自行撰寫Makefile，此時我們必須在新增Project的畫面選擇「Standard Make C++ Project」，如圖30所示，選擇這種方式開發C++程式時，CDT會在一開始允許設定一些進階的選項(在之前的方式則必須在Preferences中設定)，如圖31，在這兒我們不作任何設定，直接按下【Finish】。

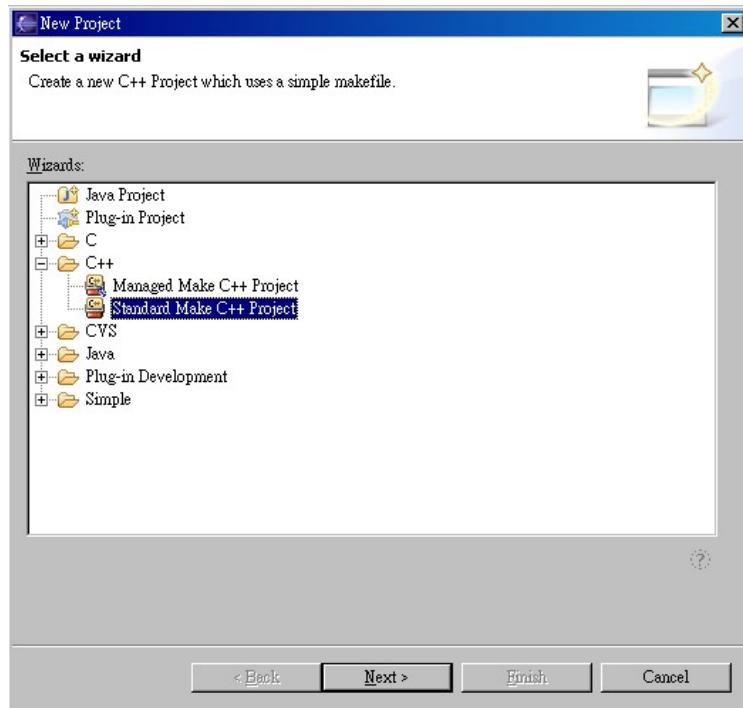


圖 30：新增專案畫面(Standard Make C++ Project)

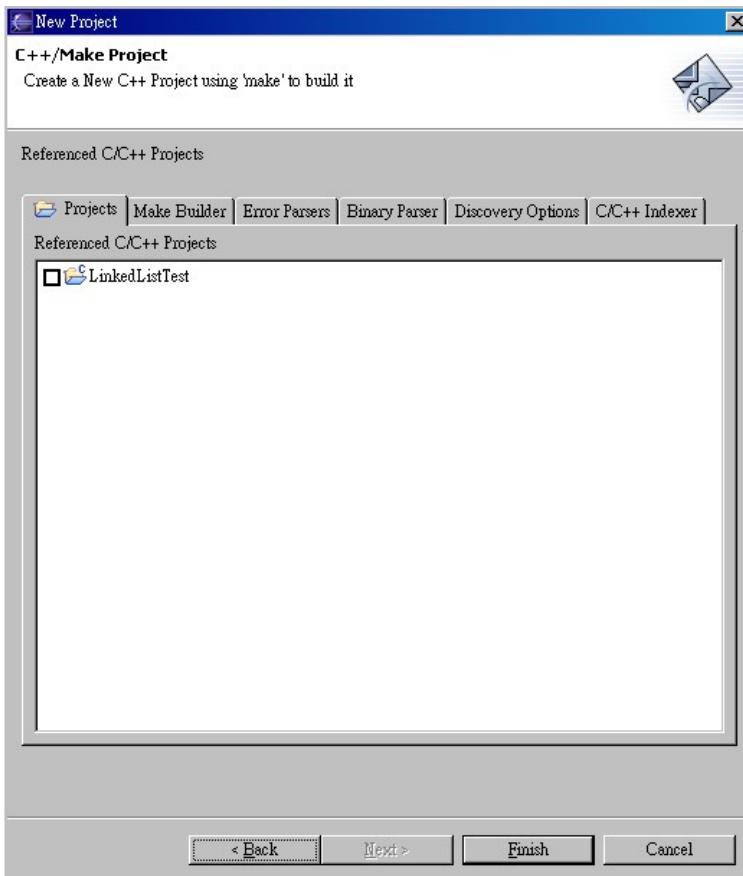


圖31：Project 進階設定畫面

請先如同之前的範例先新增好一個Hello.cpp的程式，接下來，還要寫一個make file，也請仿照之前的步驟，新增一個檔案，名稱為makefile（沒有副檔名），並將下列內容輸入其中：

```
objects = hello.o
hello : $(objects)
    gcc -o hello $(objects)
    -L C:/MinGW/lib/gcc-lib
$(objects) : hello.cpp
    gcc -c hello.cpp
all :
    ${MAKE} hello
.PHONY : clean
clean :
    -del hello.exe $(obje
```

表示接續下一行的字元

這個值會依據你安裝的MinGW版本而不同

寫好之後，如同之前的方式，Eclipse會在我們存檔時自動幫我們編譯程式，然後要執行程式時接下來，請從【Run】選單中選擇【Run...】(【Run】→【Run...】)。

在Linux系統上寫C/C++ 版本的HelloWorld

由於CDT原本就是計劃在Linux上搭配Eclipse執行的一個plug-in，因此，在Linux上面的設定就比在Windows上要簡單的多。在此我們以Red Hat Linux 9為例子。同樣的，使用者必須要先安裝Eclipse 3.1、SUN JDK 1.4.2版以上與CDT 3.0.0。由於Red Hat Linux 9已經內附有gcc、make等GUN tools，所以就不需要像在Windows平台上安裝類似MinGW這樣的軟體。確定環境都設定好之後，參考之前在Windows新增C/C++專案的步驟，所有的步驟都完全一樣。

備註

1. CDT與Eclipse中內附的JDT（Java開發環境）比起來，功能還是差很多。例如，在JDT中相當好用的Refactor(Refactoring)、Organize Imports、Scrapbook Page以及很多code templates，在CDT中都沒有，也許CDT還需要時間來成長。不過，CDT整合了gcc、make與gdb（for debugging）等GUN Tools，讓我們可以在Eclipse中開發與除錯C/C++的程式（Windows與Linux平台皆可），也算是滿方便的；更重要的是，這一切都是免費的。相信假以時日CDT與JDT之間的差異應該會越來越小才是。
2. 新版的3.0.0是搭配Eclipse 3.1的，到了這一版本提供了許多前一點所描述的JDT的功能了，像是這一版提供了少許的Refactor(Refactoring)的功能，而Scrapbook Page及code Templates均以支援，此外也提供了一個Wizard產生Class，但主要的，還是多了個自動管理Makefile的功能，讓使用者省去寫Makefile的麻煩。（當然，還是可以自己寫Makefile來開發程式的。）

參考資料

- [1] CDT home, <http://www.eclipse.org/cdt/>.
- [2] CDT user FAQ,
http://dev.eclipse.org/viewcvs/index.cgi/~checkout~/cdt-home/user/faq.html?cvsroot=Tools_Project
- [3] MinGW home, <http://www.mingw.org/>
- [4] cygwin home, <http://cygwin.com/>
- [5] Getting Started with Eclipse and the SWT, <http://www.cs.umanitoba.ca/~eclipse/>
- [6] IBM developerworks, “C/C++ development with the Eclipse Platform: How to use the C/C++ Development Toolkit (CDT)”, Pawel Leszek (pawel.leszek@ipgate.pl), 22 April 2003(<http://www-106.ibm.com/developerworksopensource/library/os-ecc/>)

修改記錄

日期	版本	主要修改內容	作者
2003/9/18	0.01	本文件第一個公開版本	陳建村
2004/9/10	0.02	1. Eclipse版本為3.0，CDT版本為2.0.1 2. CDT的安裝方式改為以軟體更新方式安裝 3. 以Managed Make C++ Project開發HelloWorld 程式	徐天送
2004/10/3	0.03	1. 更新安裝CDT的方式(修改建村學長寫的) 2. 加入原本以Makefile方式編譯程式的部份	徐天送
2005/9/15	0.04	1. 修正Eclipse版本至3.1，CDT版本至3.0.0 2. 更新安裝Eclipse和CDT的步驟	陳孟哲