

Eclipse Project CDT (C/C++) Plugin Tutorial ^{1,2}

By Brian Lee

Department of Computer Science, University of Manitoba, Winnipeg, Manitoba, Canada

Last revised: February 20, 2004

Overview:

This tutorial describes the installation of the Eclipse Project CDT (C/C++ Development Tools) plugin on the Windows platform (running under Linux is mentioned at the end of this tutorial). It describes the entire installation process in detail and is intended for developers who are familiar with the Eclipse environment and would like to develop C/C++ applications in this environment. As a result, it has been assumed that Eclipse has already been installed and that the developer has a basic understanding of how to use Eclipse. To learn more about the Eclipse Project, please visit:

http://www.eclipse.org

In addition, this tutorial is especially useful for students who are uncomfortable writing C/C++ applications in a UNIX or Linux console environment. By using the Eclipse CDT plugin, students can develop C/C++ applications within the Eclipse IDE which provides an intuitive GUI for development. As well, applications developed this way require minimal effort in porting to a UNIX or Linux environment.

Requirements:

This tutorial has been written in a Windows XP Professional environment, satisfying the following requirements:

- Java 2 SDK v1.4.1
- Eclipse v2.1
- MinGW v3.0.0-1
- CDT v1.1.0 GA

¹ This work was funded by an IBM Eclipse Innovation Grant.

² © Brian Lee and David Scuse

For instructions on how to install Eclipse, please refer to the following document:

http://www.cs.umanitoba.ca/~eclipse/1-Install.pdf

Links to downloading MinGW and the Eclipse CDT plugin will be mentioned later in this document.

Installing and Setting Up MinGW:

In order to be able to compile C and C++ source code using Eclipse, it is necessary to install a C/C++ compiler for Windows. There are several C/C++ compilers available for Windows but MinGW has been selected for this tutorial due to its easy installation and configuration. MinGW is based on GNU toolsets and provides the header files and libraries needed for C/C++ development. At the time of writing this tutorial, the latest version of MinGW that has been tested to work properly with the Eclipse CDT plugin is version 3.0.0-1. To download MinGW, please visit the following site:

http://sourceforge.net/projects/mingw/

SourceForge.net: Project In	nfo - MinGW - M	linimalist GN	U for Windows - N	licrosoft Inte	ernet Explore	r	
<u>F</u> ile <u>E</u> dit ⊻iew F <u>a</u> vorites <u>T</u> oo	ols <u>H</u> elp						1
🌀 Back 🔹 💮 🕤 🗾 💋	🏠 🔎 Sea	rch 🤺 Favo	rites 😢 Media 🍕	3 😒 🖗	2		
Address 🕘 http://sourceforge.net/p	orojects/mingw					💌 🔁 Go) Links »
6 phpMyAdmin 7 phpBB 8 ffdshow	Candidate	binutils	August 11, 2003	III - 12	Download		
9 eMule Plus 10 TightYNC	Current	Windows API	September 11, 2003	U - D	Download		
More Statistics>>	Previous	make	May 15, 2003	Ø - Ø	Download		
Sponsored Content	Proposed	pdcurses	July 20, 2003	Ø - Ø	Download		
 Try a Microsoft server for 6 months 	Snapshot	libiconv	May 11, 2003	J - D	Download		
 Visit webservices.devchani 		[Vie	w ALL Project Fil	es]			
 Get 100s of Microsoft protocols easier 	Public Areas	;	Latest	News		_	
	Project Hom	e Page	Release	ed: w32api	-2.4.tar.gz		
	Tracker		earni [Re	e - 2003-09 ad More/Co)- <i>12 08:32</i> mment]		
	- Bugs (70 total) Bug Tracking	open / 427 System	Release 1,0xe earni [Re	ed: MinGW-	-3.0.0- -05 08:03		
<	- Support Re	equests (4		22 110/0/00			>
http://sourceforge.net/projects/vir	tualdub/		/			🥑 Internet	.:

Click on the **Released: MinGW-3.0.0-1.exe** link.



Click on the link to download the file to your system.

Once the download is complete, locate the file on your system and double-click on the file to begin the installation of MinGW.

1.) Click **Yes** to continue with the installation of MinGW.



Click Next to continue.



2.) Click **Yes** to accept the License Agreement.

Note: Unlike other ports of GCC to Windows, the runtime libraries are not distributed using GNU's General Public License (GPL). You, therefore, do not have to distribute your source code with your programs unless, of course, you use a GPL library in your programs.³



3.) Click **Next** to continue.

🕏 Setup - Minimalist GNU for Win32
Information Please read the following important information before continuing.
When you are ready to continue with Setup, click Next.
MinGW - Minimalist GNU for Win32 Version 3.0.0 http://www.mingw.org
Welcome to the world of MinGW. MinGW provides a native Win32 version of GCC, binutils, gdb and a four them of www.fournits.foll.ites[s.gdb]
packages used to create this MinGW-3.0.0 installation package see
/minow/doc/MinGW//MinGW/ PACK AGES rtf
< Back Next> Cancel

4.) Install MinGW to the following directory: C:\MinGW\

Note: If you install to another directory, remember the directory you install to as it will be used later in this tutorial to complete the installation of the plugin.

³ Taken from: http://www.mingw.org/mingwfaq.shtml

🕏 Setup - Minimalist GNU for Win32
Select Destination Directory Where should Minimalist GNU for Win32 be installed?
Select the folder where you would like Minimalist GNU for Win32 to be installed, then click Next.
D:\MinGW
CA development Documents and Settings eclipse-SDK-2.1-win32 MinGW Program Files WINDDW/S VINDDW/S VIN
🧇 C: 💌
The program requires at least 43.8 MB of disk space.
Cancel Cancel

5.) Click **Install** to continue.

Ready to Install Setup is now ready to begin installing	Minimalist GNU for	Win32 on your comp	outer.
Click Install to continue with the instal	ation, or click Back	. if you want to revie	w or
Destination directory: C:\MinGW			
			~
GW - Minimalist Gnu for Windows''			

The following screen shows the progress of the installation. MinGW installs a number of GNU GCC and binutils projects, including GDB which is very useful in debugging your C/C++ applications.

😼 Setup - Minimalist GNU for Win32	
Installing Please wait while Setup installs Minimalist GNU for Win32 on your computer.	
Extracting files C:\MinGW\bin\gdb.exe	
15 %	
""MinläW - Minimalist länu for Windows" —	Cancel

6.) Click **Finish** to exit the setup of MinGW.



- 7.) In order for your system to recognize the MinGW executables from any directory without having to type the full path of the command, the PATH variable needs to be modified:
 - a. Click on Start \rightarrow Control Panel.
 - b. Double-click on System.
 - c. Click on the **Advanced** tab and then **Environment Variables**.

System Proper	ties		? 🛛
System Re	store Automa	atic Updates	Remote
General	Computer Name	Hardware	Advanced
You must be k	ogged on as an Administra	itor to make most ol	f these changes.
Visual effects	, processor scheduling, m	emory usage, and t	virtual memory
		(<u>S</u> ettings
User Profiles			
Desktop sett	ings related to your logon		
		(S <u>e</u> ttings
Startup and F	Recovery		
System startu	ıp, system failure, and deb	ugging information	
		(Settings
	Enviro <u>n</u> ment Var	iables E <u>r</u> ro	r Reporting
	04	Cance	Apply

Variable	Value
TEMP	C:\Documents and Settings\Administrat
TMP	C:\Documents and Settings\Administrat
	Naur Edit Dalaka
/stem variables	Value
/stem variables Variable ComSpec	Value C:\WINDOWS\system32\cmd.exe
vstem variables Variable ComSpec NUMBER_OF_P	Value C:\WINDOWS\system32\cmd.exe
vstem variables Variable ComSpec NUMBER_OF_P OS	Value C:\WINDOWS\system32\cmd.exe 1 Windows_NT
ystem variables Variable ComSpec NUMBER_OF_P OS Path	Value C:\WINDOWS\system32\cmd.exe 1 Windows_NT C:\WINDOWS\system32;C:\WINDOWS;

d. Select **Path** under the System Variables section and click on **Edit**.

Edit System Varia	ble 🛛 🖓 🔀
Variable name:	Path
Variable value:	remRoot%)System32)Whem:C:)MinGW)hin
Tanabio <u>T</u> alaol	

e. Assuming that MinGW has been installed to C:\MinGW, **add** the following to the **end** of the Variable value:

;C:\MinGW\bin

Note: The semi-colon is used to separate different directories for the PATH variable and **must** precede the pathname C:\MinGW\bin. **Also**, **please make sure that there are no spaces between the semi-colon and the path name!**

- f. Click OK, then OK, and OK again to apply the changes.
- 8.) To verify that the installation of MinGW is successful and that the PATH variable has been changed correctly, perform the following steps:
 - a. Click on Start \rightarrow Run ...
 - b. Type **cmd** into the dialog box and press OK.
 - c. Type the following into the command prompt and then press ENTER:

gcc –v

d. If the output of the above command is similar to the screen shown below, then you have successfully installed MinGW and changed the PATH variable:



e. If the output of the above command is similar to the screen shown below, this means that the PATH variable has not been changed successfully. Follow the instructions carefully on the previous pages to change the PATH variable:



9.) After verifying that MinGW has been installed and that the PATH variable has been changed successfully, you are ready to install the Eclipse CDT plugin.

Installing the Eclipse CDT Plugin:

Now that your system is ready to compile C/C++ applications, download the Eclipse CDT plugin at the following site: <u>http://www.eclipse.org/cdt/</u>

1.) Click on the **Download Site** link:



2.) Scroll down and click on the **1.1.0 GA – Windows** link:

🕘 Eclipse Projects -	Microsoft Internet Explorer			
<u>File E</u> dit <u>V</u> iew F <u>a</u> v	vorites <u>T</u> ools <u>H</u> elp			A*
🚱 Back 🝷 🛞 -	🖹 🗟 🏠 🔎 Search 🌱	👆 Favorites 🛛 😵 Media	🚱 🔗 - 🍓 🖻	
Address 🙆 http://www.	.eclipse.org/cdt/			💙 🄁 Go 🛛 Links 🌺
eclipse				eclipse project universal tool platform
home	Latest Releases			<u>_</u>
consortium	Build Name	Use With Eclipse	Package\Files	
projects	2002/05/47	version DD 1	110.04 Mindaus	
downloads	2003/05/17	<u>R2.1</u> B2.4	1.1.0 GA - Windows	
articles	2003/05/17	<u>R2.1</u>	1.1.0 GA - Linux (Motif)	
newsgroups	2003/05/17	<u>R2.1</u>	1.1.0 GA- Linux (GTK)	
mailing lists	2003/05/17	<u>R2.1</u>	1.1.0 GA - Solaris (Motif)	
community	2003/05/17	<u>R2.1</u>	1.1.0 GA - QNX Neutrino (Ph	<u>oton)</u>
search				
bugs	Previous Release			
eclipse tools	T TEVIOUS ITEIEUSE	Ilse With Eclinse		
Downloads	Build Name	Version	Package\Files	
CDT	2003/01/08	<u>R2.0.2</u>	1.0.1 GA - Windows	
GEF	2003/01/08	<u>R2.0.2</u>	1.0.1 GA - Linux (Motif)	
COBOL	2003/01/08	<u>R2.0.2</u>	1.0.1 GA- Linux (GTK)	
Hyades	2003/01/08	<u>R2.0.2</u>	1.0.1 GA - Solaris (Motif)	
EMF	2003/01/08	<u>R2.0.2</u>	1.0.1 GA - QNX Neutrino (Ph	oton)
	2002/04/00	00.0 C	10101	· · · · · · · · · · · · · · · · · · ·
e				🥥 Internet

- 3.) Save the file to your system.
- 4.) Once the download is complete, unzip the contents of the file:

org.eclipse.cdt-win32_1.1.0.bin.dist.zip

to the directory where Eclipse.exe is located. For example:

C:\eclipse-SDK-2.1-win32\eclipse

Extraction Wizard	
Select a Destination Files inside the ZIP a choose.	archive will be extracted to the location you
	Select a folder to extract files to. Files will be extracted to this girectory: D:\eclipse-SDK-2.1-win32\eclipse Browse Password Extracting
	< Back Next > Cancel

Note: After unzipping the file into the Eclipse directory, the contents of the plugin directory should have the following **org.eclipse.cdt.*** folders:

🔄 plugins					
<u> File Edit View Favorites Iools H</u>	elp				.
🚱 Back 👻 🕥 – 🏂 🔎 Sear	ch 😥 Folders 🛄 🔹				
Address 🛅 C:\eclipse-SDK-2.1-win32\eclips	e\plugins			>	Go
Folders ×	Name 🔺	Size	Туре	Date Modified	^
🕝 Desktop 🔨	Corg.apache.ant_1.5.2		File Folder	8/23/2003 4:48 PM	
🗉 📋 My Documents	Constant of the constant of th		File Folder	8/23/2003 4:48 PM	=
🖃 👿 My Computer	Corg.apache.xerces_4.0.7		File Folder	8/23/2003 4:48 PM	
H 4 316 Floppy (A:)	ant.core_2.1.0		File Folder	8/23/2003 4:48 PM	-
Local Disk (C:)	Construction of the second sec		File Folder	9/8/2003 8:40 PM	
E Course development	Corg.eclipse.cdt.core_1.1.0		File Folder	9/8/2003 8:40 PM	
Documents and Settings	Construction of the second sec		File Folder	9/8/2003 8:40 PM	
clining	Corg.eclipse.cdt.debug.mi.core_1.1.0		File Folder	9/8/2003 8:40 PM	
	Corg.eclipse.cdt.debug.mi.ui_1.1.0		File Folder	9/8/2003 8:40 PM	
E Confectures	Corg.eclipse.cdt.debug.ui_1.1.0		File Folder	9/8/2003 8:40 PM	
	Construction of the constr		File Folder	9/8/2003 8:40 PM	
anache ai	Corg.eclipse.cdt.launch_1.1.0		File Folder	9/8/2003 8:40 PM	
Corg apache lu	arg.eclipse.cdt.ui_1.1.0		File Folder	9/8/2003 8:40 PM	
	avg.eclipse.cdt.win32_1.1.0		File Folder	9/8/2003 8:40 PM	
T C org. epiece ar	Corg.eclipse.cdt_1.1.0		File Folder	9/8/2003 8:40 PM	
E C org.eclipse.cl	Corg.eclipse.compare_2.1.0		File Folder	8/23/2003 4:48 PM	
	arg.eclipse.core.boot_2.1.0		File Folder	8/23/2003 4:48 PM	
< >	Tora.eclipse.core.resources.win32 2.1.0		File Folder	8/23/2003 4:49 PM	*

- 5.) To verify that the Eclipse CDT plugin has been installed successfully, run Eclipse.exe from the install directory as you normally would. Then perform the following steps:
 - a. Click on Help \rightarrow About Eclipse Platform.
 - b. Click on the Plug-In Details button.
 - c. If your screen looks similar to the following then the Eclipse CDT plugin has been successfully installed:

Provider	Plug-in Name	Version	Plug-in Id
Eclipse.org	Ant Build Tool Core	2.1.0	org.eclipse.a
Eclipse.org	Apache Ant	1.5.2	org.apache.ant
Eclipse.org	Apache Lucene	1.2.1	org.apache.l
Eclipse.org	C/C++ Development Tools Core	1.1.0	org.eclipse.c
Eclipse.org	C/C++ Development Tools Debugger UI	1.1.0	org.eclipse.c
Eclipse.org	C/C++ Development Tools Debug Model	1.1.0	org.eclipse.c
Eclipse.org	C/C++ Development Tools GDB/MI CDI D	1.1.0	org.eclipse.c
Eclipse.org	C/C++ Development Tools GDB/MI CDI D	1.1.0	org.eclipse.c
Eclipse.org	C/C++ Development Tools Launching Sup	1.1.0	org.eclipse.c
Eclipse.org	C/C++ Development Tools UI	1.1.0	org.eclipse.c
Eclipse.org	Compare Support	2.1.0	org.eclipse.c
Eclipse.org	Core Boot	2.1.0	org.eclipse.c
Eclipse.org	Core Resource Management	2.1.0	org.eclipse.c
Eclipse.org	Core Runtime	2.1.0	org.eclipse.c
Eclipse.org	CVS SSH Core	2.1.0	org.eclipse.t
Eclipse.org	CVS Team Provider Core	2.1.0	org.eclipse.t
Eclipse.org	CVS Team Provider UI	2.1.0	org.eclipse.t
Eclipse.org	Debug Core	2.1.0	org.eclipse.d
Eclipse.org	Debug UI	2.1.0	org.eclipse.d
Eclipse.org	Default Text Editor	2.1.0	org.eclipse.ui
Eclipse.org	Eclipse C/C++ Development Tools	1.1.0	org.eclipse.cdt
Eclipse.org	Eclipse C/C++ Development Tools (Windo	1.1.0	org.eclipse.c
Eclipse.org	Eclipse C/C++ Development User Guide	1.1.0	org.eclipse.c

d. If your screen is missing the C/C++ Development Tools Plug-In components, then verify that the file org.eclipse.cdt-win32_1.1.0.bin.dist.zip has been unzipped to the eclipse\plugins folder as noted on the previous page.

Creating a C++ Project using the Eclipse CDT Plugin:

This section describes how to create a C/C++ project in Eclipse using the CDT plugin. It has been assumed at this point, that the system is ready to compile C/C++ applications using tools from MinGW and that the Eclipse CDT plugin has been successfully installed.

Run Eclipse from the install directory as you normally would. Once Eclipse is open, create a new C/C++ project by going to File → New → Project.

🔙 New Project			
Select Select a wizard			
C++ Java Plug-in Development Simple		Standard Make C Project	
	< <u>B</u> ack	Next > Einish	Cancel

2.) For this tutorial, you will be creating a C++ Project. When the dialog box pops up, click on C++ and select **Standard Make C++ Project**. Click **Next** to continue.

🚝 New Project	
Select Create a new C++ project	
C+ C++ - Java - Plug-in Development Simple	Standard Make C++ Project
< <u>B</u> ack	Next > Enish Cancel

 Enter HelloWorld for the name of the project and either use the default workspace or select an alternate location. In the example below, the project is stored in C:\development\HelloWorld. Click Next to continue.

🗲 New Project	
Standard Make C++ Project	
Create a new Standard Make C++ Project.	E ST
Name: HelloWorld	
Use Default Location	
Location: C:\development\HelloWorld	Browse
< <u>Back</u> <u>N</u> ext > Einish	Cancel

- 4.) Click on the **Build Settings** tab and under the **Build Command** section:
 - a. **Uncheck** the Use Default checkbox.
 - b. Enter the following in the Build Command field (note that mingw32-make is one string, with no embedded blanks; also, the complete path the the mingw32-make command may be specified):

mingw32-make -f makefile

c. Click **Finish** to continue.

🚝 New Project				X
Standard Make C++ Settings Define the Standard Make C++ b	uild settings.			
Projects Build Settings Bi Build Settings C Stop On Error C Keep Going On Error Build Command	nary Parser			
Use Default Build Command; mingw32-mak	e -f makefile			
	< <u>B</u> ack	Next >	Einish	Cancel

5.) When the following dialog box pops up, click **Yes** to continue.

Confirm Perspective Switch	×
This kind of project is associated with the C/C++ Development Perspective Do you want to switch to this perspective now?	э.
☑ Do not show this message again	
<u>Yes</u> <u>N</u> o	

- 6.) You should now be in the C/C++ Development Perspective, where you can see the HelloWorld project you just created. Now that your project has been created:
 - a. Click on File \rightarrow New \rightarrow File...
 - b. Enter **main.cpp** as the name of the file.
 - c. Click **Finish** to continue.

A New File	
File Create a new file resource.	
Enter or select the parent folder:	
HelloWorld	
≧ ⇔ ⇔ - 🕞 Helloworld	
File name: main.cpp	
Advanced >>	
	Einish Cancel

Tutorial 7 – Page 13

7.) Once the file is created, copy and paste the following code into **main.cpp**:



Press **Ctrl** + **S** to save the file. Note the following error messages in the C-Build window below; these error messages will eventually disappear:



- 8.) To create the makefile:
 - a. Click on File \rightarrow New \rightarrow File...
 - b. Enter **makefile** as the name of the file.
 - c. Click **Finish** to continue.

🖉 New File	
File Create a new file resource.	
Enter or select the parent folder:	
HelloWorld	
in the low orld in the low of th	
File <u>n</u> ame: makefile	
_Advanced >>	
	Einish Cancel

9.) Once the file is created, copy and paste the following into the **makefile**:

```
main : main.o

g++ -o main main.o -L C:/MinGW/lib/gcc-lib/mingw32/3.2.3/ -lstdc++

main.o : main.cpp

g++ -ggdb -c main.cpp

all :

${MAKE} main

clean :

-del main.o
```

The above makefile is simply a generic makefile that could easily be expanded to include a number of different source files. To learn more about the structure of makefiles and how to write your own, check out the many resources available on the Web.

After typing the above into the makefile, press Ctrl + S to save the file. Now, some different error messages are displayed in the **Tasks** window below:

e Edit Navigate Search Project Ri	un <u>Wi</u> ndow Help	
š• 🔜 🖳 👜 🛛 🏘 • 🖈 • 🔧	· · · · · · · · · · · · · · · · · · ·	
Package Explorer 🛛 💌 🗙	Main.cpp X	
g ← ← È X ↓ → ₩ HelloworldCpp → ₩ HelloworldCpp → ₩ makefile	<pre>#include <lostream> int main(int argc, char *argv[]) (</lostream></pre>	
	Tasks (2 Rems)	× 4 ≈ %
	✓ Tasks (2 items) ✓ I Description Resource	Son
		e InFolder HelloWorldCpp
	Zasks (2 items) ✓ 1 Description Cout' undeclared (first use this function) main.cpp `end' undeclared (first use this function) main.cpp	

- 10.) If you look in the C/C++ Project window, you will notice a red 'X' beside main.cpp. The error messages indicate that 'cout' and 'endl' are both undeclared. But the source code in main.cpp looks fine! This is a very common problem and developers tend to get very frustrated in their attempts to solve this problem. The fix is actually quite simple:
 - a. In main.cpp, add the following line right #include <iostream>:

using namespace std;

b. Press **Ctrl** + **S** and your application should compile successfully as shown in the screen below. Note that **main.exe** has now been created.



- 11.) We are almost done! Now to run your application:
 - a. Click on **Run** \rightarrow **Run**...
 - b. In the Configurations window, click on C/C++ Local, and then the New button.
 - c. In the Name field, type HelloWorldCpp.
 - d. In the C/C++ Application field, type **main.exe**.
 - e. The settings on the other tabs will be sufficient for the purpose of this tutorial.
 - f. Then click on the **Apply** button. Your screen should look like the following:

🔙 Run		X
Create, manage, and run con	ìgurations	*
Configurations: Conf	Name: Helloworld Main Arguments Environment Debugger Project:	Source
Ne <u>w</u> Delețe	Apply	Reyert
	Run	Close

g. Now click the **Run** button to run your application. You should get the following output in the console window:



h. Congratulations! You have just created your first C++ application in Eclipse using the Eclipse CDT plugin.

Creating a C Project:

Creating a C project is almost identical to creating a C++ project. First, create a new C project called HelloWorldC. Set the Build Command property to:

mingw32-make -f makefile (the same as for the C++ project).

Create main.c with the following contents:



Create the makefile with the following contents:

Compile the project causing main.exe to be generated.



Create the associated Run command.

🖉 Run		
Create, manage, and run confi	igurations	Å
Configurations:	Main Arguments Environment Debugger Source 第 Common Project:	Browse Search
Ne <u>w</u> Delețe	Αρρίχ	Reyert
	Ryn	Close

and then Run the project:

ie Edit Navigator X Imain.c Imain.c <	C/C++ Development - main.c - Ec	lipse Platform	. 🕫 🔀
Navigator Navigator Navigator Console Console Console Console Console Console Console Console Console Console Console Console <th><u>File E</u>dit <u>N</u>avigate Se<u>a</u>rch <u>P</u>roject <u>R</u>u</th> <th>n <u>W</u>indow <u>H</u>elp</th> <th></th>	<u>File E</u> dit <u>N</u> avigate Se <u>a</u> rch <u>P</u> roject <u>R</u> u	n <u>W</u> indow <u>H</u> elp	
Image: Second system Image: Second system Image: Second		·★·┭ · タ ₽ ₽ ₽ *	
<pre>int main(int argc, char *argv[]) { printf("Hello world!\n"); return; // end main main.ce main.c main.ce m</pre>	🗄 🗟 Navigator 🔹 🗙	C main.c X makefile	
Imain.c Imain.cxee Imain.o	↓ ↓ ↓ ↓ ✓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ </td <td><pre>int main(int argc, char *argv[]) { printf("Hello world!\n"); return; } // end main</pre></td> <td></td>	<pre>int main(int argc, char *argv[]) { printf("Hello world!\n"); return; } // end main</pre>	
Console [<terminated> C:\Program Files\Eclipse\workspace\HelloWorldC\main.exe (10/28/03/3/53 PM)]</terminated>	← C main.c main.exe main.o main.o main.o makefile		
		Console Exterminated's C/IProgram Elles/Erlinselworksnare/HelloWorldCimain.eve (10/28/03 3/53 PM	
			^
Hello world!		Hello world!	-
		<	>
C/C++ Projects Navigator Tasks C-Build Properties Console	C/C++ Projects Navigator	Tasks C-Build Properties Console	

Once again, Hello world should appear in the console.

Running CDT under Linux:

We did one simple test of CDT under Linux using the following configuration:

- Debian Linux with kernel 2.4
- Java 2 SDK v1.4.2
- Eclipse v2.1.1
- MinGW v3.3.2
- CDT v1.1.0

Once CDT was unzipped into the Eclipse directory, the default settings for "Standard Make C++ project" worked correctly since GCC was already installed. If you can compile a C++ program from a console, the default settings should be sufficient for compiling within Eclipse.

Basic Console Input/Output:

It is quite common that C/C++ programs require user input via the console. The following example shows how to enter input into the console of a program written using the CDT plugin:

- 1.) Create a new C++ project called **HelloWorld** which contains the following files:
 - main.cpp
 - makefile

2.) Copy and paste the following code into **main.cpp** and save the file:

```
#include <iostream>
#include <iostream>
#include <string>
using namespace std;
int main(int argc, char *argv[])
{
    string name;
    cout << "Please enter your name: " << endl;
    cin >> name;
    cout << "Hello " + name << endl;
    return 0;
}// end main</pre>
```

3.) Copy and paste the following code into **makefile** and save the file:

4.) There should be no errors in the C-Build window and the project should compile itself successfully, generating an executable file called **main.exe**:



If you get a **"Build Error"** message in the C-Build window, verify that your build settings are setup correctly.



a. **Right-click** on the HelloWorld project and click properties.

E	C/C++ Deve	elopment - main.cpp	Eclipse Platform				
File	Edit Naviç	gate Search Project R	n Window Help				
ÌÈ	- 🛛 🖷	8 S S S	*****	-> - │ 🖻 🖁 🗍			
Ē	€ <mark>8</mark> C/C++	Projects 💌 🗙	🖻 main.cpp 🗙 📋 makefile			🗄 Outline	↓ ^a _z ×
2 B		New Go Into GoTo Open in New Window Dauld Clean Make Add Make Targets Refresh Close Project Deserve	<pre>#include <lostream> #include <lostream> #include <string> # namespace std; # wain(int argc, cha string name; cout << "Please en cin >> name; # cout << "Hello " + ceturn 0; end main</string></lostream></lostream></pre>	r *argv[]) ter your name: " <- name << endl;	< endl;	e m	stream ring ain(int, char*[])
		Delete Team Compare With	}		×	Cutine Ma	ke Targets
		Replace With	ild [HelloWorld]				<i>.</i> ×
		Restore from Local His Properties	error:Launching fa	iled)			~
	CIC++ Droi	incte Navigator	Tacks C-Ruild Properties				~
	(HelloWorld	incos internation	rasis Crouid Properties		Writable	Insert	17:1

b. Select C/C++ **Project** on the left hand side.

Properties for HelloWorld		
Info External Tools Dudders Project References	E/C++ Project Build Settings Indexer Binary Parser -Build Settings - Stag On Error - ' Stag On Error - '' Keep Gaing On Error - Build Command - - Use Default - Build Command:	Restore Dafaults 6504
		OK Cancel

c. Uncheck the **Use Default** checkbox under the Build Command section and type the following into the Build Command box:

Properties for HelloWorld		
Info External Tools Bullders External Tools Bullders Project References	E/C++ Project Build Settings Indexer Binary Parser Build Settings C Stop On Error Build Command Build Command: Build Command: Mingw32-make -f makefile Restore D	efaults Apply
		OK Cancel

mingw32-make -f makefile

- d. Click OK.
- e. Right-click on the HelloWorld project and click on **Rebuild**. The project should compile itself successfully as shown in the screen below. Note that **main.exe** has now been created.



- 5.) To run the application:
 - a. Click on **Run** \rightarrow **Run**...
 - b. In the Configurations window, click on C/C++ Local, and then the New button.
 - c. In the Name field, type **HelloWorld**.
 - d. In the C/C++ Application field, type **main.exe**.
 - e. The settings on the other tabs will be sufficient for the purpose of this example.
 - f. Then click on the **Apply** button. Your screen should look like the following:

🔚 Run		
Create, manage, and run con	igurations	ď
		~1
Configurations:	Name: HelloWorld	
E-FC C/C++ Local FC HelloWorld	Main Arauments Environment Debugger	Source ☆ ◀ ▶
Java Apple. 	Project:	
Run-time Workbench	Helloworld	Browse
	C/C++ Application:	
	main.exe	Search
Ne <u>w</u> Dele <u>t</u> e	Apply	Revert
	Run	Close

g. Now click the **Run** button to run your application. You should get the following output in the console window:

🚰 C/C++ Development - main.cpp - Eclipse Platform					
Eile Edit Navigate Search Project	Elle Edit Navigate Search Project Run Window Help				
Ă·⊟風A ĂŇ \$¢	 ★ - 美 - 美 - ● ダ ジ - ジ - ○ 目 省 県 				
🖆 📸 C/C++ Projects 🛛 🔻 🗙	C main.cpp X i makefile	🗄 Outline 🛛 🖡 🗴			
Construction C	<pre>#include <iostream> #include <iostream> #include <string> using namespace std; int main(int argc, char *argv[]) (string name; cout << "Please enter your name: " << endl; cin >> name; cout << "Hello " + name << endl;</string></iostream></iostream></pre>	 isstream string main(int, char*[]) 			
	return 0; // end main Console [C:\development/new]HelloWorld]man.exe (12/20/03 6:06 PM)]	Outline Make Targets			
	Please enter your name:	×			
C/C++ Projects Navigator	Tasks C-Build Properties Console				

h. Type your name and then press Enter. The next line should read "Hello [name]" and then the program terminates.

Note: If you are having problems entering text into the console, make sure that your cursor is blinking at the end of all text in the console. To position the cursor at the end, simply press Ctrl + End while the cursor is blinking in the console. After performing this step, you should be able to enter text into the console.

Basic File Input/Output and Command Line Arguments:

Occasionally, a question on an assignment may require you to create a program that reads in command line arguments, such as the names of input and output files. The following example shows how to provide command line arguments to applications at the beginning of execution:

- 1.) Create a new C++ project called **HelloWorld** which contains the following files:
 - main.cpp
 - names.txt
 - makefile

(For detailed instructions on how to do this, please refer to Part I of this tutorial under the section "**Creating a Project using the Eclipse CDT Plugin**".

2.) Copy and paste the following code into **main.cpp** and save the file:

```
#include <iostream>
#include <fstream>
#include <string>
using namespace std;
// Constants
#define BUFFER SIZE
                               50
#define ARRAY SIZE
                               20
// Global variables
int numElements = 0;
void sort(char *elements[]) {
      bool swap = true;
      char *temp = NULL;
      while(swap) {
            swap = false;
            for (int i = 0; i < (numElements - 1); i++) {</pre>
                  if (strcmp(elements[i], elements[i + 1]) > 0) {
```

```
temp = elements[i];
                         elements[i] = elements[i + 1];
                         elements[i + 1] = temp;
                         swap = true;
                   }
            }
      }
}
int main(int argc, char *argv[])
{
      char buffer[BUFFER SIZE];
      char *listOfNames[ARRAY_SIZE];
      string inputFileName;
      string outputFileName;
      ifstream inputFile;
      ofstream outputFile;
      if(argc != 3) {
            cout << "Error: Please enter input and output files ";</pre>
            cout << "as command line arguments !" << endl;</pre>
      else {
            inputFileName = argv[1];
            outputFileName = argv[2];
            inputFile.open(inputFileName.c_str());
            outputFile.open(outputFileName.c_str());
            // Read names from input file and store into array
            while(!inputFile.eof() && numElements < (ARRAY_SIZE - 1)) {</pre>
                   inputFile.getline(buffer, BUFFER_SIZE);
                   char *p = new char[strlen(buffer) + 1];
                   strcpy(p, buffer);
                   listOfNames[numElements] = p;
                  numElements++;
            }
            // Sort elements in array
            sort(listOfNames);
            // Print elements in array to output file
            for(int i = 0; i < numElements; i++) {</pre>
                   outputFile << listOfNames[i] << endl;</pre>
            }
            inputFile.close();
            outputFile.close();
      }
      return 0;
 // end main
```

The previous block of code simply reads in two command line arguments – the first argument is the name of the input file and the second argument is the name of the output file. The program will read from the input file, store the names into a vector,

and sort these elements alphabetically. Lastly, it will write the ordered list of names to the output file.

3.) Copy and paste the following names into names.txt and save the file:

Brian			
Michelle			
Jeff			
Lori			
Ainsley			
Jimmy			
Bob			
Lorisa			
John			

4.) Now, to specify command line arguments when running your application:

- a. Click on **Run** \rightarrow **Run**...
- b. In the Configurations window, click on C/C++ Local, and then the New button.
- c. In the Name field, type **HelloWorld**.
- d. In the C/C++ Application field, type **main.exe**.
- e. Click the Arguments tab.
- f. In the C/C++ Program Arguments textbox, enter the following: names.txt output.txt
- g. The settings on the other tabs will be sufficient for the purpose of this example.
- h. Then click on the **Apply** button. Your screen should look like the following:

🚝 Run		
Create, manage, and run con	Afigurations	大
Configurations:	Name: Helloworld Main Arguments Environment Debugger Sour C/C++ Program Arguments: Inames.txt output.txt Working directory: ✓ Use default working directory C Local directory	e N I I I I I I I I I I I I I I I I I I
Ne <u>w</u> Dele <u>t</u> e	Apply	Re <u>v</u> ert
	Ryn	Close

i. Now click the **Run** button to run your application. You should see a new file called **output.txt** in the directory view of your project:



j. Double-click on **output.txt** to open the file and you should see the list of names that was in **input.txt**, sorted alphabetically in ascending order:



Debugging Code:

Debugging is an essential skill that every programmer needs to master, whether to fix a problem in an application or even to step through the application slowly to see how it works. Fortunately debugging C/C++ applications on the Eclipse platform using the CDT plugin is very straightforward. Continuing from the example described in the previous section, let's take a look at how to set breakpoints and debug the application:

- 1.) Switch to the Debug perspective:
 - a. Click on Window → Open Perspective → Other...
 - b. Select the **Debug** perspective and click OK.



c. Your perspective should resemble something like the following:



- 2.) In order to have access to the basic C/C++ debugging functionality, perform the following steps:
 - a. Click on **Window** → **Customize Perspective...**
 - b. Expand the **Other** branch, check the C/C++ Debug box, and click OK.

Customize Perspective	
Select the items to be displayed in the current perspective (Debug) Available Items:	Details: Add Address Breakpoint (C/C++) Add Address Breakpoint Add Expression (C/C++) Add Expression (C/C++) Restart IP Resume At Line (C/C++) P Resume Without Signal ⇔I Run To Line (C/C++)
	OK Cancel

- 3.) Let's set our first breakpoint:
 - a. Open **main.cpp** and scroll down to just before the vector of names is sorted:





b. Right-click in the margin and then click Add Breakpoint.

- 4.) Now that we have a breakpoint at that spot, we can being to debug our application:
 - a. Click on **Run** \rightarrow **Debug...**
 - b. In the Configurations window, click on **HelloWorld** (this configuration should already exist after completing the example from the previous section).

🚰 Debug		×
Create, manage, and run confi	gurations	芬
Configurations:	Name: HelloWorld Main Arguments Environment Debugger Project: HelloWorld C/C++ Application: main.exe	Source 🟂 ()
New Delete	Apply	Reyert
	Debug	Close

c. Click the **Debug** button. Just minimize the Windows console window that pops up and your screen should look something like the following:

	🛢 Debug - main.cpp - Eclipse Platform				
Eile	<u>E</u> dit <u>N</u> avigate Search <u>P</u> roject <u>R</u> un <u>W</u> indow <u>H</u> elp				
Ď	·□□▲ ☆·★·♥· @メ ☆々・☆・	🖬 🖞 🖗			
Ē	🕸 Debug 👻 🗙	👀= Variables		粒	
Pa	₩ _ % <i>_</i> % <i>A</i> _ _ _	✓ argc=3 ★ argy=0v3d2538			^
EC	HelloWorld [C/C++ Local] - - Cyawin GDB Debugger (1/11/04 10:29 PM) (Suspended)	± Z buffer[50]			=
棼	- Inread [1] (Suspended)	IstOfNames[20] InputFileName			_
	1 main() at main.cpp:34 C:\development\new\HelloWorld\main.exe (1/11/04 10:29 PM)	🛨 🗾 outputFileName			~
		Variables Breakn Expres	Registers	Memory Shared	LSianz € ▶
					12
	c man.cpp X			Dutline	JªZ ×
	char *listOfNames[ARRAY SIZE];		₽	fstream	
	string inputFileName;			u string	I
	string outputFileName;			# BUFFER_SI	ZE
	ifstream inputFile; ofstream outputFile;		#	# ARRAY_SIZ	t s int
				sort(char*[]) : void
	if(arga = 2) (*	✓	main(int, ch	ar*[]) : int
		1			
	Console [Cygwin GDB Debugger (1/11/04 10:29 PM) (Suspended)]			■) * ·	• 🔒 🖉 🗙
					~
	<				>
	Console Tasks				
			Writable	Insert	34:26

- 5.) Let's further analyze the previous screen before proceeding with the debugging of our application:
 - a. The **Variables** window displays all the local variables currently within the scope of the current block of code. Also note the tabs at the bottom of the Variables window such as **Breakpoints**, **Expressions**, etc.. We will take a look at some of these windows in future tutorials.



b. The **Debug** window displays the current state of the stack frame of your application. This will prove to be very helpful when debugging programs with recursive functions.

🖶 Debug - main.cpp - Eclipse Platform	
<u>Eile Edit N</u> avigate Se <u>a</u> rch <u>P</u> roject <u>R</u> un <u>W</u> indow <u>H</u> elp	
E <u>Debug</u> ✓ X → Variables → □ □ N → Ø <u>Z ~ 0</u> <u>B</u> <u>B</u> <u>B</u> <u>A</u> → ⁷ argc=3 → 0 → 0 → 0 → 0 → 0 → 0 → 0 → 0 → 0 → 0	約 및 ▼ ×
Image: State Sta	
	×
	×
Variables Breakp Expres Reg	jisters Memory Shared Signa 🔹 🕨
C main.cpp X	🗄 Outline 🛛 🖓 🕹
char buffer[BUFFER_SIZE];	iostream
<pre>char *listOfNames[ARRAY_SIZE];</pre>	fstream
string outputFileName;	# pileen cize
	# ARRAY SIZE
ifstream inputFile;	A numElements : int
ofstream outputFile;	🦳 🔍 🛛 sort(char*[]) : void
	main(int, char*[]) : int
if(arm != 3) {	
Console [Cygwin GDB Debugger (1/11/04 10:29 PM) (Suspended)]	■ 3x • A 2 ×
	<u>~</u>
8	
Console Tasks	

c. The buttons circled below provide you with the basic debugging functionality.

C I	ebug - main.cpp - Eclipse Platform		
Eile	<u>E</u> dit <u>N</u> avigate Se <u>a</u> rch <u>P</u> roject <u>R</u> un <u>W</u> indow <u>H</u> elp		
	- □ □ □ □ □ □ □ □ - ☆ - ★ - ♥ - □ ● タ や 수 - ↔ -		
Ē	² ∑ Debug	👀 Variables	2 ↓
₽ <mark>~</mark> 電 参		rgc=3 argv=0x3d2538 buffer[50] four lstofNames[20] inputFileName outputFileName	
		Variables Breakp Expres Regis	sters (Memory (Shared) Sign ()
	<pre>c main.cpp X</pre>		Coutine 1ª × iostream string # BUFFEr_SIZE # ARRAY SIZE
	ifstream inputFile; ofstream outputFile; if(ergg l= 3) (A numElements : int sort(char*[]) : void main(int, char*[]) : int
	Console [Cygwin GDB Debugger (1/11/04 10:29 PM) (Suspended)]		🗉 🕅 🕶 🔒 🖉 🗙
			~
			>
	Console Tasks		

Familiarize yourself with the shortcut keys, as they will greatly speed up the debugging process!

Command	Name	Shortcut Key	Description
	Resume	F8	Resumes a suspended thread.
	Terminate	-	Terminates the selected debug target.
Ъ.	Step Into	F5	Steps into the highlighted statement.
•	Step Over	F6	Steps over the highlighted statement. Execution will continue at the next line either in the same method or (if you are at the end of a method) it will continue in the method from which the current method was called.
	Step Out	F7	Steps out of the current method.

*Note: Above table is a modified version of the material from Eclipse's Help file.

- 6.) Continue on with the debugging process:
 - a. Click on the button or press F8 to allow the application to resume the suspended thread and hit the breakpoint we set earlier.



b. Before we step into the sort method, let's take a look at our variables. If we expand the **listOfNames** array, you will notice that each element in the array corresponds to a name read in from the unsorted input file.

🚝 Debug	g - main.	cpp - Eclipse Platform	
<u>Eile E</u> dit	<u>N</u> avigate	Se <u>a</u> rch <u>P</u> roject <u>R</u> un <u>W</u> indow <u>H</u> elp	
🛎 🗕 🛛] ! . ê)	
¹ № □	Debug		
	} c:\d	Variables Breakp Expres Regist	ters Memory Shared Sign: ()
,+ _)	<pre>>> // Sort elements in array sort(listOfNames); // Print elements in array to output file for(int i = 0; i < numElements; i++) (outputFile << listOfNames[i] << endl;) innutFile close().</pre>	Courine Courine Courine Courine Stream Stream String BUFFER_SIZE AnumElements : int sort(char*[]) : void main(int, char*[]) : int
Consc	Console [Cy sole Tasks	ygwin GDB Debugger (1/11/04 10:39 PM) (Breakpoint hit)]	

c. Now click on the -button or press F5 to step into the sort method. Once inside the sort method, if you will look in the **Debug window**, you will notice an extra frame on top of the stack called:





d. If you click on the frame below called:

1 main() at main.cpp:60

You will notice that you can get view the variables from the main method in the Variables window. Sometimes during debugging, you may need to refer to variables from the calling method and you could easily do this by clicking on the corresponding stack frame. As well, this proves quite handy when debugging through recursive functions.



e. Click on the **2 sort(char**)() at main.cpp:15** frame to get back to the sort method and expand the **elements** variable in the Variables window. It appears that there's only one element in the array!

C	🦉 Debug - main.cop - Eclipse Platform			
File	File Edit Navinate Search Project Run Window Help			
Luo Xa I				
Ē	🇉 🏂 Debug 👻 🗙 😔 Variables		犯	🛄 🕶 🗙
PA	🗞 🗈 🗉 🖬 서 🖘 🛷 🕱 🤉 🐢 🕜 📴 팽 👷			
睛	🕞 🖃 🏇 HelloWorld [C/C++ Local]	an		
	→ Swap=119 Swap=119			
7,2	Thread [1] (Suspended)			
	2 sort(char**)() at main.cpp:15			
	I main() at main.cpp:60			1981
	C:(development(new(nellowond(main.exe(1)11)0410:39PM)			
	Uniphies Provide Futures Des	istors Mon	on Charad	. Sign: 4
	Variables breakp Expres Res	iscers men	ory snareu	bight i P
	C main.cpp X	📴 Outlir	e	Ja₂ ×
		- 1	iostream	
	<pre>void sort(char *elements[]) {</pre>	🎴	fstream	
	bool swap = true;		string	
	char *temp = NULL;	#	BUFFER_SIZE	
	while (Swap) (#	ARRAY_SIZE	
	for (int i = 0: i < (numElements - 1): i++) (sort(cbar*[])	: IIIC : void
	<pre>if (strcmp(elements[i], elements[i + 1]) > 0) {</pre>		main(int, chai	*[]) : int
	temn = elements[i]			
	Console [Cygwin GDB Debugger (1/11/04 10:39 PM)]		■ 🕅 🖈 •	₿ <u>/</u> ×
				~
				~
				>
	Console Tasks			

What happened to all the other elements?? Since this method takes in a pointer to a char array, and we know that there are 20 elements in the array, we can easily display this variable as an array instead of a pointer.

f. Right-click on the elements variable and select Display as Array...



g. Change the Length to 20 and click OK.

🚝 Display As Array					
Туре:	char **				
Start index:		0			
Length:		20			
		OK Cancel			

h. You should now be able to see all 20 elements in the array.



i. Now click on the 🗣 button or press F6 to step through the rest of the sort method. You should notice that the array is slowly sorted into ascending order.



j. Once you get tired of stepping through the while/for loops and you understand how the sort works, click on the -* button or press F7 to step out of this sort method and to return back to the main method. Once in the main method, expand the **listOfNames** variable to verify that the array has been sorted.



- k. Click on the ▶ button or press F8 to resume the rest of the application. After the application terminates, you can switch back to C/C++ Development Perspective and open **output.txt** to verify that the input file has been sorted correctly.
- 7.) One of the "quirks" of the C/C++ debugger is that if you have **global variables** in your program (variables defined outside your functions and main method), you will not see them appear in the Variables window. So what if you wanted to inspect those variables during debugging? The solution is actually quite simple:
 - a. Start up the application in debug mode by clicking on the bug in the toolbar.
 - b. Click on the ► button or press F8 to hit the first breakpoint and then click on the -button or press F5 to step into the sort method.
 - c. Now that we're in the sort method, to determine the value of the global variable **numElements**, highlight the variable, right-click and click on **Add Expression...**

Debug - main.cpp - Eclipse Platform								
lebug - main.cpp - Lelipse Platform Edit Navigate Search Project Run Window Hel ·		Undo Revert Cut Copy Paste Shift Right Shift Left Comment Unconnent						
1 main() at main.cpp:60 C:\development\new\HelloWorld\main.exe (1,		Content Assist Add Include Open on Selection	Breakp	Expres Re	gisters	Memory Shared	> Sign: • •	
<pre>G main.cpp X void sort(char *elements[]) (bool swap = true; char *temp = NULL; while(swap) (swap = false; for (int = 0; i < (nume) if (strcmp(elements[i</pre>	⇒[]► ¥¥	Search Run To Line Resume At Line Add Expression Save Constraints [1 + 1])	(> 0) (Utine iostream fstream string BUFFER_SII BUFFER_SII A numElement sort(char*[] main(int, ch	↓ ^a ₂ × 2E s : int) : void ar*[]) : int	
Console [Cygwin GDB Debugger (1/11/04 11:58 PM)] Console Tasks]			Writabl	e	Insert	× A 2 ×	

d. Verify that the Expression to add is **numElements** and then click OK.



e. You should now see that the global variable numElements has been added to the **Expressions** window. Repeat the above steps for any other global variable you would like to verify during debugging.

e	Debug - main.con - Eclinse Platform					
Eile	Edit Navieste Saarde Project Due Window Help					
Elle	Edit Mavidate search Golect Kan winnow Deh					
	S・日 Q 合 参・式・戦・ の タ や や・ウ・					
Ē	The Debug 🗸 🗸 🗸	Expressions	2 = = 4 U - ×			
P_		**** numElements=9				
Ba	- 🏇 HelloWorld [C/C++ Local]					
206	🖃 🔐 Cygwin GDB Debugger (1/11/04 11:58 PM)					
7,30	Interest [1] (Suspended)					
	2 sort(char**)() at main.cpp:15					
	1 main() at main.cpp:60					
	C:\development\new\HelloWorld\main.exe (1/11/04 11:58 PM)					
		<	>			
		Variables Breakp Expres Regi	sters Memory Shared Signa 🔍 🕨			
	C main.cpp ×		E Outline La X			
		~	iostream			
	<pre>void sort(char *elements[]) (</pre>		- I fstream			
	bool swap = true;		- 🛄 string			
	char *temp = NULL;		# BUFFER_SIZE			
	while(swap) {	# ARRAY_SIZE				
	swap = false;		numElements : int			
	for (int i = 0; i < (numElements - 1);	i++) (sort(char*[]) : void			
	<pre>if (strcmp(elements[i], elements[i]</pre>	. + 1]) > 0) {	— 🔍 main(int, char*[]) : int			
	temn = elements[i].					
	Console [Cygwin GDB Debugger (1/11/04 11:58 PM)]		🔲 🐹 • 🗎 🌽 ×			
			<u>~</u>			
			~			
			>			
	Canada Taska					
	Console Lasks					