Code::Blocks Manual

Introduction to Computer Programming - Part 1 (CS101.1x)

Prepared by Sandeep Prasad

Edited by Firuza Aibara

Guided by Nagesh Karmali



Department of Computer Science and Engineering Indian Institute of Technology - Bombay Mumbai - 400076.

August 5, 2014

Contents

1	Introduction to Code::Blocks	1
2	Code::Block IDE	1
3	Working with Code::Block on Windows	6
	3.1 Installation of Code::Block	6
	3.2 Writing a new C/C++ Program \ldots	12
	3.2.1 Non-Graphic Project	12
	3.2.2 Graphic Project using graphics.h	20
	3.3 Building the Project	25
	3.4 Opening existing program/project	26
4	Working with Code::Block on Ubuntu	27
	4.1 Installation of Code::Blocks	27
	4.2 Installation of packages for graphics.h header file	28
	4.3 Writing a new $c/c++$ program \ldots	29
	4.4 Building the Project	
	4.4.1 Non-Graphics Project	32
	4.4.2 Graphics Project using graphics.h	
	4.5 Opening Existing Program/Project	37

List of Figures

1	Code::Block IDE	1
2	Menu Bar	1
3	Main Toolbar	2
4	Debugger Toolbar	2
5	Compiler Toolbar	3
6	Manager	3
7	Editor, Start/Home Page	4
8	Shortcut to Create New Project or Open Existing Project	4
9	Shortcut to History of Projects Opened Using Code::Blocks	5
10	Logs	5
11	File association window	6
12	Click on "Run" \ldots	6
13	Welcome to the CodeBlocks-EP Setup Wizard screen	7
14	License Agreement	7
15	Information window	8
16		8
17		9
18		9
19		10
20		10
21^{-5}		11
$22^{$		11
$23^{}$		12
$\frac{-0}{24}$		12
25		13
$\frac{20}{26}$		13
$\frac{20}{27}$		14
28		14
29		15
$\frac{20}{30}$		$15 \\ 15$
31		16
32		16
33		17
$\frac{35}{34}$		17
35 - 35		18
36		18
$\frac{30}{37}$	Project node with ++' sign indicating it can be expanded	
37 38	Project node expanded	
$\frac{38}{39}$	· -	20
39 40	1	20 20
40 41		20 21
$41 \\ 42$		21
$42 \\ 43$		21
43 44	· · ·	22 22
		22 23
45 46		23 23
46		
47		24
48	1 1 0	25
49 50		25 26
50 51	0 1	26 26
51 52		26
52 52		27
53	Installing Code::Block using command line	27

54	Starting a new project
55	Selecting the language for project
56	Title for Project
57	Selecting Compiler to Compile the Program
58	Project Node when Expanded
59	Project Node when Expanded for helloworld.c (with code shown in editor)
60	Project node when expanded for line.c (with code shown in editor)
61	Output for helloworld.c
62	Project build options
63	Linker settings (Add Libraries)
64	Interface for adding libraries
65	Files/libraries to be added for graphic projects
66	Relative or absolute path for files/libraries
67	libraries selected
68	Libraries added to project
69	Output for line.c
70	Select file with .cbp extension to open an existing project

1 Introduction to Code::Blocks

"Code::Blocks is a free C++ IDE built to meet the most demanding needs of its users." [1]. Developed by 'The Code::Blocks Team', Code::Block is a free, open-source [2] and cross-platform IDE, which supports various free compilers. It is built around plugin framework, which allows functionality of Code::Block to be extended by installing appropriate plugins. Plugins required for compiling and debugging are already provided by default. This manual is prepared after installing and testing Code::Block on Ubuntu 12.04^{1} and Windows $7.^{2}$

2 Code::Block IDE

Code::Block IDE is shown in figure 1 (Ubuntu 12.04). The main parts of Code::Block along with figures are discussed below

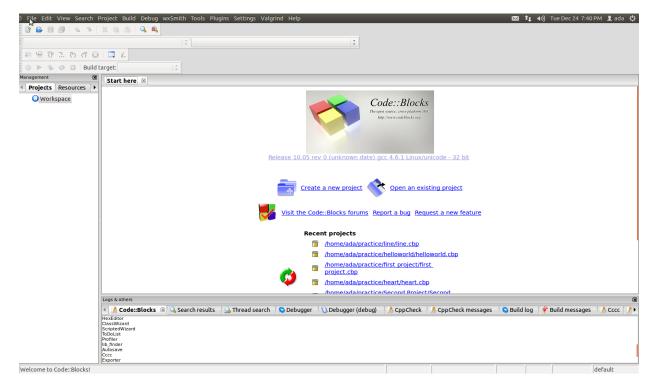


Figure 1: Code::Block IDE

1. Menu bar:

Menu bar is shown in figure 2. Menu bar can be toggled using F10. Few important link in menu bar are described below (described from left to right):

🗢 📼 File Edit View Search Project Build Debug wxSmith Tools Plugins Settings Valgrind Help

Figure 2: Menu Bar

(a) <u>File:</u> File menu link contains options to create a new project, open an already existing project, save file, save project, save workspace and save everything. It also contains options for closing

 $^{^1\}rm Ubuntu$ 12.04 with intel @Core $^{\rm TM}i3\textsc{-}2120$ CPU @ 3.30GHzx4 processor 4 GB RAM and 32-bit architecture and 64 bit architecture.

 $^{^{2}}$ Windows 7 with intel (Core TM i3-2120 CPU (3.30GHzx4 processor 4 GB RAM and 32-bit architecture.

a single file, closing a project or closing entire workspace. Other options in File are to print, export and quit the Code::Block

- (b) <u>Edit</u>: All the editing options required for editor are provided in Edit.
- (c) <u>View</u>: This menu link contains link for various perspectives and toolbars along with manager, logs, script console, status bar, full screen.
- (d) <u>Project</u>: Options related to the project is provided in this link which includes configuring build options along with options for adding files, removing files and autoversioning of project.
- (e) <u>Build</u>: Options for building the project, compiling a single file, running the project, building and running the project, rebuilding the project and cleaning the project is provided in build. Options for Building, rebuilding and cleaning the entire workspace is also provided along with options to select target (debug/release) and analysing error one by one.
- (f) Debug: Various Debugging options are provided in this link.
- (g) <u>Plugins</u>: Various plugins can be executed using this link. The link to manage the plugins is also provided here.
- (h) <u>Settings</u>: This contains link for various settings, setting related to *Environment...*, *Editor...*, *Compiler and debugger...*, *Global Variables...* and *Scripting...*. Script to be executed during Code::Block start-up can also be edited here.
- (i) <u>Help</u>: It contains information about Code::Block version, tips which can be toggled to be displayed at start-up and information about various plugins.

2. Main tool bar:

Main tool bar is shown in figure 3. The buttons in Main toolbar are (from left to right):



Figure 3: Main Toolbar

- (a) <u>New File</u>: For creating a new project.
- (b) Open: For opening an already created project.
- (c) <u>Save</u>: To save the file open in active editor (active editor means the editor tab in focus).
- (d) <u>Save all files</u>: To save all the files for the current/selected project.
- (e) <u>Undo:</u> To undo the executed action.
- (f) <u>Redo:</u> To redo the undone action.
- (g) <u>Cut:</u> To cut the selected/highlighted part in editor.
- (h) Copy: To copy the selected/highlighted part in editor.
- (i) <u>Paste:</u> To paste the cut/copy message in editor.
- (j) <u>Find:</u> To find required text in the file in active editor.
- (k) Replace: To replace required text in the file in active editor by some alternate text.

3. Debugger tool bar:

Debugger tool bar is shown in figure 4. Debugger tool bar is used to debug the current/selected project. The buttons in debugger toolbar are (from left to right) *Debug/Continue*, *Run to cursor*, *Next line,Next instruction*, *Step into*, *Step out*, *Stop debugger*, *Debugging Windows* and *Various info*. You will be able to understand the use of this buttons by rigorous practise of debugging various projects.



Figure 4: Debugger Toolbar

4. Compiler tool bar:

Compiler tool bar is shown in figure 5 and is used in building/compiling/running the current/selected project. The buttons in Compiler toolbar are (from left to right):



Figure 5: Compiler Toolbar

- (a) <u>Build:</u> For building the current/selected project.
- (b) <u>Run:</u> For running the current/selected project.
- (c) <u>Build and run:</u> For building and running the current/selected project.
- (d) <u>Rebuild:</u> For rebuilding the current/selected project.
- (e) <u>Abort</u>: For aborting the build process for the current/selected project.
- (f) <u>Build target:</u> For defining the type of build target for current/selected project, either debug or release.

5. Manager:

Manager is shown in figure 6. It is labelled as Management. This window provides the list of all the open projects and files for easy access to any required file of any project.

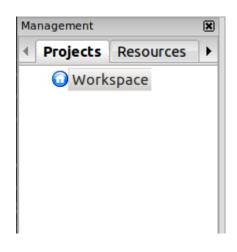


Figure 6: Manager

6. **Editor:**

Editor is shown in figure 7. Here, all the coding work will take place. It is provided in tabbed fashion to work with many files at once. When no project is open, the start page or home page is displayed in editor. The links given in start page is divided into two parts and explained below



Figure 7: Editor, Start/Home Page

7. Figure 8 is short-cut on Starting page of IDE for creating a new project and opening an already created project. It also contains link for Code::Block forum where many useful resources can be found along with other useful discussions. The link points to url http://forums.codeblocks.org/. The second and third link points to BerLiOS Developer Site aims at enriching the Open Source community by providing a centralized place for Open Source Developers to control and manage Open Source Software Development.



Figure 8: Shortcut to Create New Project or Open Existing Project

8. Figure 9 is short-cut to list of projects and files already opened in the IDE. It is link to few projects and files from history of IDE.



Figure 9: Shortcut to History of Projects Opened Using Code::Blocks

9. Logs:

Log window is shown in figure 6. It is labelled as 'Logs & others'. This window acts as log for various actions performed in IDE. All logs related to various activities can be checked at appropriate windows.

Logs & others									×
🖣 🚺 Code::Blocks 🐵 🕓 Search results	🔒 Thread search	😒 Debugger	🔌 Debugger (debug)	📝 CppCheck	CppCheck messages	🛇 Build log	📌 Build messages	🚺 Cccc	1
HexEditor									
ClassWizard									
ScriptedWizard									
ToDoList									
Profiler									
lib_finder									
Autosave									
Cccc									
Exporter									

Figure 10: Logs

3 Working with Code::Block on Windows

In this section we discuss writing and building of two projects. First project (hello world.cpp) is a simple program which displays *hello world* on output. The second project (line.c) uses graphics.h header file and displays a line. For Windows (Windows 7) we will be using 'CodeBlocks-EP'. When CodeBlocks-EP is run for first time a 'file association window' is displayed as shown in figure 11. Select 3rd option "Yes, associate Code::Blocks with c/c++ file types" or 4th option "Yes, associate Code::Blocks with every supported type(including project files from other IDEs)" and click 'OK'.

File associations	×
Code::Blocks is currently not the default application for C/C++ source files. Do you want to set it as default?	
You can always change associations from the environment settings later.	
No, leave everything as it is	
No, leave everything as it is (but ask me again next time)	
Yes, associate Code::Blocks with C/C++ file types	
\bigcirc Yes, associate Code::Blocks with every supported type (including project files fi	rom other IDEs)
	ок

Figure 11: File association window

3.1 Installation of Code::Block

Code::Block used for windows is CodeBlocks-EP (stands for Code::Blocks - EDU Portable). WinBGIm (Borland Graphics Interface) required to run programs with graphics.h header is already integrated in CodeBlocks-EP. Download CodeBlocks-EP from http://codeblocks.codecutter.org/ The installation steps are as given below.

1. Download Code::Block-EP installer from the link given above and browse to the appropriate directory where the installer is downloaded. Click the .exe file downloaded and window as shown in figure 12 will pop up. Click 'Run'.



Figure 12: Click on "<u>R</u>un"

2. A new window appears as shown in figure 13. Click 'Next'.



Figure 13: Welcome to the CodeBlocks-EP Setup Wizard screen

3. The third windows that appears is of license agreement as shown in figure 14. Select "I accept the agreement" and click 'Next'.

😼 Setup - CodeBlocks-EP			
License Agreement Please read the following important information before continuing.			
Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.			
Licensing of Code::Blocks			
Code::Blocks and Code::Blocks EDU-Portable are distributed under the GPL v3.0 license which means they can be used freely by anyone!			
The only exception is the SDK which is allowed to be linked by dosed-source plugins. All pre-installed libraries are Open Source.			
Full license text of the GPL version 3.0: http://www.codeblocks.org/license/3			
accept the agreement			
\bigcirc I <u>d</u> o not accept the agreement			
< Back Next > Cancel			

Figure 14: License Agreement

4. The next window displays some important information (shown in figure 15) regarding Code::Block-EP. Kindly go through information and click on "Next".

😼 Setup - CodeBlocks-EP	x
Information Please read the following important information before continuing.	
When you are ready to continue with Setup, click Next.	
graphs of results - 2D/3D graphics (GLUT) - GUI Libraries (wxwidgets, FLTK) - conio and conio2 libraries - C/C++ style formatter (AStyle) - automatic code indentation - Automatic documentation generator (Doxygen & DoxyBlocks) - makes proper code documentation a pleasant task. Why Portable? - Being portable allows using portable Code::Blocks without interference on PCs	
where where the day the owner of the contract of the contract of the second secon	-
< Back Next > Ca	ncel

Figure 15: Information window

5. Next window asks for location where the CodeBlock-EP will be installed. The default location will be C:\Program Files\CodeBlocks-EP. Figure 16 shows the location provided for CodeBlocks-EP installation. Provide appropriate location and click on "Next".

😼 Setup - CodeBlocks-EP	X
Select Destination Location Where should CodeBlocks-EP be installed?	
Setup will install CodeBlocks-EP into the following folder.	
To continue, dick Next. If you would like to select a different folder, dick	Browse.
C:\Program Files\CodeBlocks-EP	Browse
At least 626.7 MB of free disk space is required.	
< <u>B</u> ack Next >	Cancel

Figure 16: Providing location for CodeBlocks-EP installation

Figure 17 shows the location provided for CodeBlocks-EP shortcuts in Start Menu Bar. Click on "Next".

😼 Setup - CodeBlocks-EP	
Select Start Menu Folder Where should Setup place the program's shortcuts?	
Setup will create the program's shortcuts in the following Start M	ienu folder.
To continue, click Next. If you would like to select a different folder, click	Browse.
CodeBlocks-EP	Browse
Don't create a Start Menu folder	
< <u>B</u> ack Next >	Cancel

Figure 17: Providing folder for CodeBLocks-EP shortcuts

Select the checkbox 'Create a desktop icon' as shown in Figure 18. Click 'Next'.

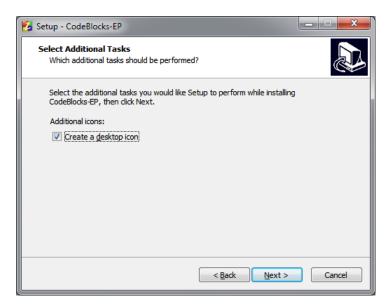


Figure 18: Creating desktop shortcut for CodeBlocks-EP

6. A new window displays stating that it is ready to install, which is shown in Figure 19. Click 'Install' to proceed with installation.

😼 Setup - CodeBlocks-EP	x
Ready to Install Setup is now ready to begin installing CodeBlocks-EP on your computer.	
Click Install to continue with the installation, or click Back if you want to review or change any settings.	
Destination location: C:\Program Files\CodeBlocks-EP Start Menu folder: CodeBlocks-EP Additional tasks: Additional tasks: Create a desktop icon	*
4 F	
< Back Instal C	Cancel

Figure 19: Installing CodeBlocks-EP

Figure 20 shows installation progress with progress bar.

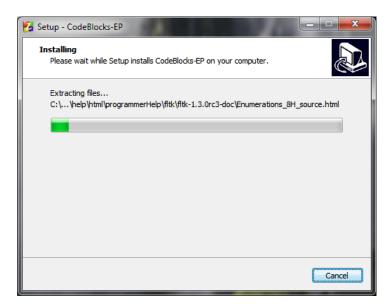


Figure 20: Progress bar to show progress of installation

7. When the installation in complete a window is displayed shown in figure 21. If you want to launch the CodeBlock select "Launch CodeBlocks-EP" and click 'Finish'.

😼 Setup - CodeBlocks-EP	
	Completing the CodeBlocks-EP Setup Wizard
	Setup has finished installing CodeBlocks-EP on your computer. The application may be launched by selecting the installed icons.
	Click Finish to exit Setup.
	✓ Launch CodeBlocks-EP
	Einish

Figure 21: Completing the installation process

8. CodeBlocks IDE opens as shown in figure 22. Alternatively CodeBlocks-EP can be launched by double clicking the desktop icon created or clicking on it's shortcut icon in Start Menu Bar.

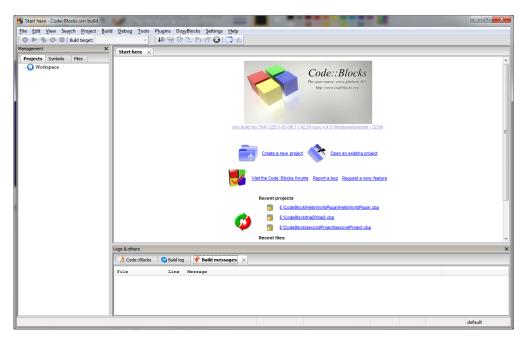


Figure 22: Code::Block IDE in Windows 7

3.2 Writing a new C/C++ Program

3.2.1 Non-Graphic Project

1. Click 'File' \rightarrow 'New' \rightarrow 'Project'. A new window opens as shown in figure 23. Click 'Console application' \rightarrow Click Go.

New from templa	te	×			
Projects Build targets Files Custom User templates	Category: <ali categories=""> Console application Console application Win32 GUI project Dynamic Link Library Empty project Empty project GLUT GLUT project GRX GRX project Koolplot project </ali>	Go Cancel View as Carge icons O Large icons O List			
TIP: Try right-did	ing an item				
 Select a wizard type first on the left Select a specific wizard from the main window (filter by categories if needed) Press Go 					

Figure 23: New form template

2. When Go button is clicked, a new window opens as shown in figure 24. Select checkbox 'Skip this page next time' so that the page is not displayed again. Click 'Next'.



Figure 24: New console application wizard

3. Next window enables user to select the language to be used for project as shown in Figure 25. Click 'C++' \rightarrow 'Next'.

Console application		x
Console	Please select the language you want to use. Please make a selection C C++	
	< Back Next > Cancel	

Figure 25: Selecting language for the project

4. Next windows enables user to provide title for the project and the folder where user wishes to create the project in. This is Shown in figure 26. After filling in the details click on *Next*.

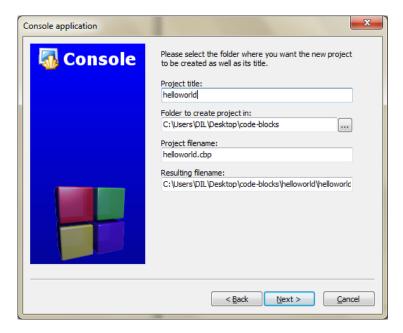


Figure 26: Providing title and folder for the project

5. Next window is used to select the compiler as shown in figure 27. By default 'GNU GCC Compiler' is selected. Click on *Finish*.

Console application	×
🖏 Console	Please select the compiler to use and which configurations you want enabled in your project. Compiler:
	Create "Debug" configuration: Debug Debug" options Output dir.: bin\Debug\ Objects output dir.: obj\Debug\
	✓ Create "Release" configuration: Release "Release" options Output dir.: bin\Release \ Objects output dir.: obj\Release \
	< Back Finish Cancel

Figure 27: Selecting compiler for the project

6. The project node opens in manager window as shown in figure 28. The project node is empty and we have to add files to the project.

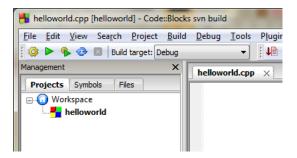


Figure 28: Project node with no files

7. To add files to the project select project node. Click 'File' \rightarrow 'New' \rightarrow 'File' as shown in figure 29.

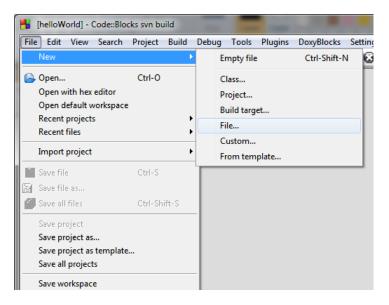


Figure 29: Adding file to the project

8. New from template opens as shown in figure 30. Click 'C/C++ source' \rightarrow 'Go'.

N	ew from templa	te		×
	Projects Build targets Files Custom User templates	C++ c/c	<all categories=""> ++ header ++ source ty file </all>	<u>Go</u> <u>C</u> ancel
				View as Carge icons List
	TIP: Try right-dicking an item 1. Select a wizard type first on the left 2. Select a specific wizard from the main window (filter by categories if needed) 3. Press Go			

Figure 30: Selecting type of file to be added in the project

9. A new window is displayed as shown in figure 31. Select the checkbox 'Skip this page next time' so that it is not displayed again.

C/C++ source	
C/C++ FILE	Welcome to the new C/C++ source file wizard! This wizard will guide you to create a new C/C++ source file. When you 're ready to proceed, please click "Next" Skip this page next time
	<back next=""> Cancel</back>

Figure 31: Select checkbox to skip this window next time

10. Click 'C++' as shown in figure 32.

C/C++ source	
	Please select the language for the file.
	< <u>Back Next > Cancel</u>

Figure 32: Select the language of the file added

11. Click on '...' besides the textbox of 'Filename with full path'. This is shown in Figure 33

C/C++ source	×
C/C++ FILE	Please enter the file's location and name and whether to add it to the active project. Filename with full path:
	Add file to active project In build target(s): Release All None
	< Back Einish Cancel

Figure 33: Details of file to be added

12. A window opens as shown in figure 34. Enter the file name (.cpp). and click 'Save'.

H Select filename				×
Colored Code-blocks + helloworld	- - i i j	Search hello	world	Q
Organize 🔻 New folder				0
★ Favorites	Dat	e modified	Туре	
Desktop No items m	atch your s	earch.		
Elibraries Documents				
d Music				
Videos				
🖳 Computer				
🕌 Local Disk (C:) 🔻 < 🔤				Þ
File name: helloworld.cpp				-
Save as type: C++ files (*.cpp;*.cx;*.cc)				•
) Hide Folders		<u>S</u> ave	Can	icel

Figure 34: Selecting the location and file name to be added

13. Select the checkbox Debug and Release. Click 'Finish' as shown in figure 35.

C/C++ source	×
CA C/C++ FILE	Please enter the file's location and name and whether to add it to the active project. Filename with full path: C: \Users\DIL\Desktop\code-blocks\helloworl
	 ✓ Add file to active project In build target(s): ✓ Debug ✓ Release All None
	< Back Einish Cancel

Figure 35: Finalize details of file to be added

14. Management window now shows project node which can be expanded. Double click 'Helloworld' and then double click 'Sources'. This is shown in figure 36 and figure 37 respectively.

helloworld.cpp [helloworld] - Code::Block	s svn build		
<u>File Edit View Search Project Build</u>	<u>D</u> ebug	<u>T</u> ools	P <u>l</u> ugins
🕴 🧼 🕨 😵 🔟 🛛 Build target: Debug		•	i 📲 🖷
Management X	hellowo	orld.cop	×
Projects Symbols Files			
🖃 😡 Workspace			
🗄 🕂 helloworld			

Figure 36: Project node with '+' sign indicating it can be expanded

helloworld.cpp [helloworld] - Code::Bl	ocks svn build
<u>File Edit View Search Project Bu</u>	ild <u>D</u> ebug <u>T</u> ools P <u>l</u> ugins
🕴 🧔 🕨 🍫 🌚 🛛 Build target: Debu	, 🚽 🕴 🧏
Management >	helloworld.cpp
Projects Symbols Files	
🖃 🕢 Workspace	
helloworld	
🗄 🗁 Sources	

Figure 37: Project node with '+' sign indicating it can be expanded

15. Double click on 'helloworld.cpp' to open the file in editor. This is shown in figure 38

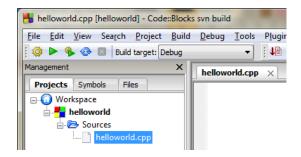


Figure 38: Project node expanded

16. Now, start writing your code in the editor.

3.2.2 Graphic Project using graphics.h

1. Click on *New file* button. The 'New form template' window as shown in figure 23 opens. For graphics projects, select 'WinBGIm project'. *Go* button gets highlighted (top right corner). Click on *Go*.

New from templa	ate	×
Projects Build targets Files Custom User templates	Category: <all categories=""> Code::Blocks plugin Static library Console application >Win32 GUI project Dynamic Link Library >WinBGIm project Empty project >WinBGIm project FLTK FLTK project >WinBGLUT project GRX GRX project Koolplot project</all>	Go Cancel View as Carge icons O Large icons
TIP: Try right-dicking an item		
	l type first on the left ic wizard from the main window (filter by categories if needed)	

Figure 39: New form template

2. When Go button is clicked a new window opens as shown in Figure 40. Select checkbox "Skip this page next time" so every time new project is created this window should not come. Click on <u>Next</u>. If this step has been performed earlier, this window will not be displayed.



Figure 40: New console application wizard

3. Next window enables user to select the language to be used for project as shown in figure 41. For

the example hello world used in this manual select ' C and click on $\underline{Next}.$

Console application	
Console	Please select the language you want to use. Please make a selection C C++
	< <u>Back Next > Cancel</u>

Figure 41: Selecting language for the project

4. Next window asks the user to select type of project. The options are 'Add Console' and 'Graphics only' as shown in figure 42. Select 'Graphics only' and click on *Next*.

WinBGIm project	
WinBGIm Graphics	Please select type of project. Please make a selection Add Console Graphics only
	< Back Next > Cancel

Figure 42: WinBGIm project

5. Next windows enables user to provide title for the project and the folder where user wishes to create the project. This is Shown in figure 43. After filling in the details click on <u>Next</u>.

WinBGIm project	X
WinBGIm Graphics	Please select the folder where you want the new project to be created as well as its title. Project title: Ine Folder to create project in: E:\CodeBlock\ Project filename: Ine.cbp Resulting filename: E:\CodeBlock\jine\jine.cbp
	< Back Next > Cancel

Figure 43: Providing title and folder for the project

6. Next window is used to select the compiler as shown in figure 44. By default 'GNU GCC Compiler' is selected. Click on \underline{Finish} .

WinB	GIm project	×
	Win <mark>BGIm</mark> Graphics	Please select the compiler to use and which configurations you want enabled in your project. Compiler: SNU GCC Compiler Create "Debug" configuration: Debug Debug" options Output dir.: bin\Debug Objects output dir.: obj\Debug
	Win	
		< <u>B</u> ack <u>Finish</u> <u>Cancel</u>

Figure 44: Selecting compiler for the project

7. The project node opens in manager window. The project node is empty and we have to add files to the project. To add files to the project select project node and click on File in menu bar, then click on 'File...' in options in 'New'. New from template opens as shown in figure 30. For our example select 'C/C++ source' and click on Go. A new window pops out which have a checkbox 'Skip this page next time'. Select the checkbox so this window should not open every time a new file is added to the project. Select the preferred language. For our example select 'C'. A new window opens which allows user to add the files to the project. Click on '...' beside 'Filename with full path'. This is shown in Figure 33. A window as shown in figure 34 opens. Select the folder of the project and enter file name to be added. Click on 'Save' (see steps 6-11 of section 5.2.1.1). Select Debug and Release. Shown in figure 45. Click on Finish.

C/C++ source	
C C/C++ FILE	Please enter the file's location and name and whether to add it to the active project. Filename with full path: E:\CodeBlock\line\line.c
	 ✓ Add file to active project In build target(s): ✓ Debug ✓ Release All None
	< Back Finish Cancel

Figure 45: Finalize details of file to be added (for line project)

8. Management window now shows project node which can be expanded (figure 46).

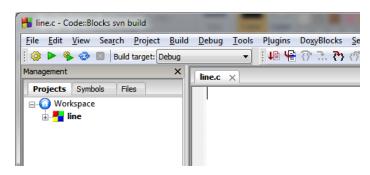


Figure 46: Project node with '+' sign indicating it can be expanded

9. Click on project node and double click on 'line.c' to open the file in editor. When the line.c file opens in editor, user can start coding. Code is shown in figure 47.

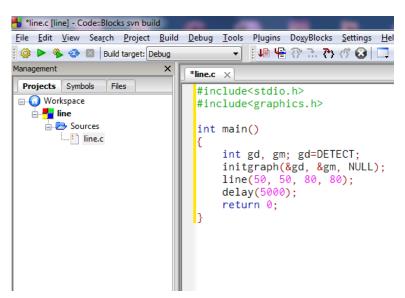


Figure 47: Writing program in editor (line project)

3.3 Building the Project

The process to build the graphics and non-graphics project is same, just Click on 'Build' and then 'Run' (or directly on 'Build and run'). The output for the program used is shown in figure 48 for hello world project, figure 49 for line project.

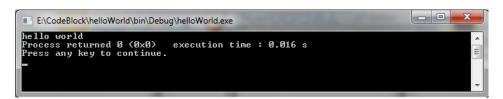


Figure 48: Output of hello world project



Figure 49: Output of line project

3.4 Opening existing program/project

Click on Open button 50. Browse to desired directory and open the file with .cbp extension as shown in figure 51.

-	Start her	re - Cod	e::Blocks	svn build		
File	Edit	View	Search	Project	Build	Debug
	New)	
ß	Open			Ctrl-O		5tart h
	Open w	vith hex	editor			
			orkspace/			
		project:	5)	
	Recent	files				_
	Import	project			•	
	Save fil	e		Ctrl-S		
2	Save fil	e as,,,				
Ø	Save all	files		Ctrl-Sh	ift-S	
	Save pr	oject				
	Save pr	oject as				
		-	template			
	Save all	project	3			
	Save w	orkspac	e			
	Save w	orkspac	e as,			

Figure 50: Selecting Open under file in Menu Bar

Organize 🔻 New	folder						(
🧮 Desktop	*	Name	Date modified	Туре	Size		
Downloads		퉲 bin	12/20/2013 4:09 PM	File folder			
Recent Places		鷆 obj	12/20/2013 4:09 PM	File folder			
-		i main.c	12/23/2013 4:48 PM	C source file		1 KB	
Libraries		secondProject.cbp	12/23/2013 4:00 PM	project file		2 KB	
Documents Music	Ξ	secondProject.depend	12/23/2013 3:50 PM	DEPEND File		1 KB	
Pictures	-	secondProject.layout	2/6/2014 12:33 PM	LAYOUT File		1 KB	
Videos							
- Hacos							
🖳 Computer							
Local Disk (C:)							
👝 New Volume (E:)						

Figure 51: Select file with .cbp extension to open an existing project

4 Working with Code::Block on Ubuntu

In this section we discuss writing and building of two projects. First project (hello world.cpp) is a simple program which displays *hello world* on output. The second project (line.c) uses *graphics.h* header file and displays a line. *graphics.h* is not supported by *gcc*, which is the default C/C++ compiler on Ubuntu. We have to install some packages, include few libgraph libraries during building the project with graphics.h header file.

4.1 Installation of Code::Blocks

Pre-requisite for installing Code::Block is 'libwxgtk' which is available in ubuntu software center. This package will be already installed in your system³. It is also advised to install 'build-essential' package and update repository list. In case the libwxgtk is not installed, it can be installed from command line using command given in listing 1 [3]. Code::Block is available in Ubuntu's repository. It can be installed using Ubuntu Software Center or it can also be installed using command line as given in listing 2.

```
1 $ sudo apt-get install libwxgtk2.8-0
2 $ sudo apt-get install build-essential
3 $ sudo apt-get update
```

Listing 1: Installing libwxgtk2.8-0 using command line

```
1 $ sudo apt-get install codeblocks
```





Figure 52: Code::Block in Ubuntu Software Center

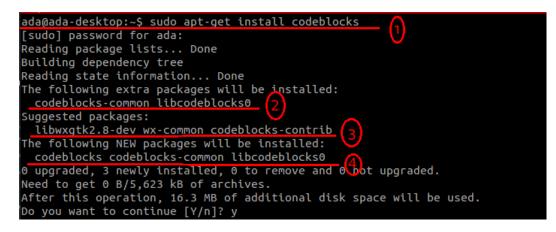


Figure 53: Installing Code::Block using command line

 $^{^{3}\}mathrm{Check}$ for the libwxgtk version available for your Ubuntu, minimum required version for Code::Block to run is 2.0, version available on Ubuntu 12.04 is 2.8

Figure 52 shows Code::Block along with it's logo in Ubuntu Software Center, while figure 53 shows installation of Code::Block using command line. In figure 53 four underlined lines are:

line 1: Command to install the Code::Block.

line 2: Packages installed along with Code::Block.

- line 3: Packages suggested along with Code::Block installation. *libwxgtk2.8*⁴ package as mentioned above is already installed in your system. *libwxgtk2.8-dev* package is not required.
- line 4: Final list of packages that will be installed in your system

When command line prompts for [Y/n] enter 'y' and press enter. When Code::Block is run for first time, It asks for default compiler. Select the appropriate compiler to proceed further. We have used 'GNU GCC Compiler'.

4.2 Installation of packages for graphics.h header file

GCC compiler does not support graphics.h, conio.h, windows.h and few other header files that works on Turbo C or Borland C. graphics.h header files enables programmer to write simple c/c++ graphics programs. In Ubuntu, gcc is default c/c++ compiler, thus, we have to make some settings, for gcc to support graphics[4]. We will start with installing some packages from command line as given in listing 3

\$ sudo apt-get install libsdl-image1.2 libsdl-image1.2-dev guile -1.8 guile -1.8-dev libsdl1.2 debian libart -2.0-dev libaudiofile-dev libesd0-dev libdirectfb-dev libdirectfb-extra libfreetype6-dev libxext-dev x11proto-xext-dev libfreetype6 libaa1 libaa1-dev libslang2-dev libasound2 libasound2-dev

Listing 3: Installing required packages to support graphics.h

After the above mentioned packages are installed, download the **libgraph** package (download link given in footnote⁵) and untar it in home directory. For this untarring tool must be installed on system. Open the command line and follow the instructions given in listing 4.

```
1 $ cd libgraph -1.0.2
2 $ ./configure
3 $ sudo make
4 $ sudo make install
5 $ sudo cp /usr/local/lib/libgraph.* /usr/lib
```

Listing 4: Installing libgraph package using command line

⁴Details about above mentioned packages can be found at http://packages.ubuntu.com/precise/allpackages. ⁵http://download.savannah.gnu.org/releases/libgraph/libgraph-1.0.2.tar.gz

4.3 Writing a new c/c++ program

1. Click on New file button. The 'New form template' window opens as shown in figure 54. Click 'Console application' \rightarrow 'Go'.

😣 New from t	emplate			
Projects Build targets	Category: <a>All cate	gories>	:	Go
Files	S.	5	GLUT	Cancel
User templates	ARM Project	Dapplication	GLUT project	C
	an		F	
	AVR Project	Empty project	GTK+ project	¢
	1 👘 👘	FLT	RUCHT	
	Code::Blocks plugin	FLTK project	Irrlicht project	F
	S	GLFW OpenGL		View as
	Console application	GLFW project	Lightfeather project	
				O List
TIP: Try right-click	king an item		r -	
	l type first on the left ic wizard from the main	n window (filter l	by categories if needed	d)

Figure 54: Starting a new project

2. A new window opens as shown in figure 55. Click 'C++' \rightarrow Next.

😣 🗈 Console applicat	ion
🐻 Console	Please select the language you want to use. Please make a selection
	C++
	< Back Next > Cancel

Figure 55: Selecting the language for project

3. The subsequent windows enable the user to provide title for the project and the folder where user wishes to create the project in. This is shown in figure 56. After filling in the details click 'Next'.

⊗ 🗈 Console applic	ation
🐻 Console	Please select the folder where you want the new project to be created as well as its title.
	Project title helloworld
	Folder to create project in: /home/ada/practice/
	Project filename: helloworld.cbp
	Resulting filename: /home/ada/practice/helloworld/helloworld.cbp
	\$
	< Back Next > Cancel

Figure 56: Title for Project

4. Next window is to select the compiler. By default 'GNU GCC Compiler' is selected. Click 'Finish'.

😵 🗈 Console applic	ation
🚮 Console	Please select the compiler to use and which configurations you want enabled in your project.
	Compiler: GNU GCC Compiler
	Create "Debug" configuration: Debug
	"Debug" options Output dir.: bin/Debug/
	Objects output dir.: obj/Debug/
	Create "Release" configuration: Release "Release" options Output dir.: bin/Release/
	Objects output dir.: obj/Release/
	< Back Fitish Cancel

Figure 57: Selecting Compiler to Compile the Program

5. Now, the project node opens in manager window as shown in figure 58. Project node can be expanded to see the main.c file.

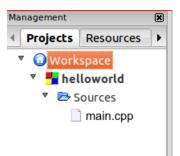


Figure 58: Project Node when Expanded

When main.c file is clicked, it opens in editor as shown in figure 59 for hello world project.

ment Imain.cpp cts Symbols 1 #include <iostream> 2 using namespace std; 4 int main()</iostream>		2 main() : int
main.cpp main.cpp int main.cpp main.cpp int main()	🖹 🖓 🗂 🏷 🌾 😂	
Imain.cpp Imain.cpp icts Symbols 1 #include <iostream> 2 using namespace std; 4 int main()</iostream>	🕨 锋 🐵 🛛 🛛 Build t	arget: Release
Workspace 3 using namespace std; helloworld 5 int main()		
		2 3 using namespace std;
	helloworld	4 5 int main() 6 □ {
Sources 7 main.cpp 8 9 cout << "Hello world!" << endl; return 0;		<pre>7 cout << "Hello world!" << endl; 8 return 0;</pre>

Figure 59: Project Node when Expanded for helloworld.c (with code shown in editor)

6. Code used for *graphics* program (line) is shown in figure 60.

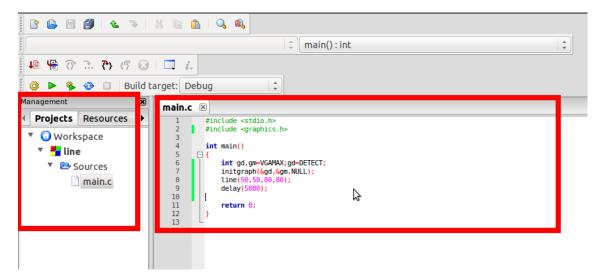


Figure 60: Project node when expanded for line.c (with code shown in editor)

7. While using Code::Blocks for the first time, some extra windows will be displayed. In this manual only the important windows are shown.

4.4 Building the Project

4.4.1 Non-Graphics Project

After the code is written, project needs to be built. Click 'Build' \rightarrow 'Build and Run'. Output is as shown in figure 61.

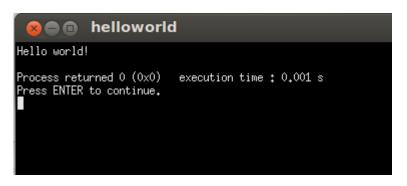


Figure 61: Output for helloworld.c

4.4.2 Graphics Project using graphics.h

The steps to link the libraries and build the project is given below.

1. Right click on project node in Manager box and select *build options*... A new window 'Project build options' as shown in figure 62 pops up. Option to change the compiler selected for the project is also available in this window.

	Selected compiler
Debug	GNU GCC Compiler
elease	Compiler setting Linker settings earch directories Pre/post build steps
	Policy: Append target options to project options 🗘
	Compiler Flags Other options #defines
	Categories:
	<all categories=""> ‡</all>
	Produce debugging symbols [-g]
	 Profile code when executed [-pg]
	In C mode, support all ISO C90 programs. In C++ mode, remove GNU extensions
	Enable all compiler warnings (overrides many other settings) [-Wall]
	Enable extra compiler warnings [-Wextra]
	Enable extra compiler warnings [-Wextra] Stop compiling after first error [-Wfatal-errors]
	Stop compiling after first error [-Wfatal-errors]
	 Stop compiling after first error [-Wfatal-errors] Inhibit all warning messages [-w]
	 Stop compiling after first error [-Wfatal-errors] Inhibit all warning messages [-w] Have g++ follow the 1998 ISO C++ language standard [-std=c++98]
	Stop compiling after first error [-Wfatal-errors] Inhibit all warning messages [-w] Have g++ follow the 1998 ISO C++ language standard [-std=c++98] Have g++ follow the coming C++0x ISO C++ language standard [-std=c++0x]
	Stop compiling after first error [-Wfatal-errors] Inhibit all warning messages [-w] Have g++ follow the 1998 ISO C++ language standard [-std=c++98] Have g++ follow the coming C++0x ISO C++ language standard [-std=c++0x] Enable warnings demanded by strict ISO C and ISO C++ [-pedantic]

Figure 62: Project build options

2. In 'Project build options' window, click on *linker settings* tab. This tab is shown in figure 63.

😣 🗈 Project	build options
line Debug Release	Selected compiler GNU GCC Compiler \$
Release	Compiler setting Linker settings search directories Pre/post build steps
	Policy: Append target options to project options 💲
	Link libraries: Other linker options:
\$	
	Add dit Delete Clear Copy selected to
	Cancel 🗸 OK

Figure 63: Linker settings (Add Libraries)

3. In linker settings tab click on Add button under Link libraries box. When Add button is clicked a small window titled Add library opens as shown in figure 64.

🗴 🗊 🛛 Project	build options
line Debug Release	Selected compiler GNU GCC Compiler
	Compiler settings Linker settings Search directories Pre/post build steps
	Policy: Append target options to project options 💲
	Link libraries: Other linker options:
	Add library
	File:
	😮 Cancel 🛷 OK
	ß
	Add Edit Delete Clear Copy selected to
	Scancel

Figure 64: Interface for adding libraries

4. Click on dotted button to right of box. A new window opens as shown in figure 65. This window enables user to browse to appropriate folder and to select required library. Browse to the */usr/lib* directory. All the required library files are not visible.

Select all files in dropdown placed at the bottom, to enable visibility of all types of files.

Select all the libgraph .* files, except libgraph.la file. Files to be selected are shown in the figure 65.

Recently Used I libgraph.a 451.1 KB Monday 17 October 2011 a da I libgraph.a 140.4 KB Tuesday Desktop I libgraph.la 859 bytes Tuesday File System I libgraph.so.1 161.1 KB Tuesday 102 GB Filesyst I libgraph.so.1.0.2 116.1 KB Tuesday System Reserved I libgraph.so.4.0.0 50.6 KB Monday 02 April 2012 Documents I libgrip.so.0.305.0 29.7 KB Friday 31 August 2012 Music I libgs.o.9.05 8.4 MB Saturday 08 September 2012 Videos I libgs.o.0 2.1 MB Friday 08 Arrch 2012	Places	Name	~	Size	Modified
Recently Used140.4 KBTuesdaya ada116,1 KBTuesdayDesktop116,1 KBTuesdayFile System116,1 KBTuesday102 GB Filesystem116,1 KBTuesdayNew Volume116,1 KBTuesdayS2 GB Filesystem116,1 KBTuesday102 GB Filesystem116,1 KBTuesday102 GB Filesystem116,1 KBTuesdayS2 GB Filesystem116,1 KBTuesdaySystem Reserved116,1 KBTuesday116,1 KB116,1 KBTuesday116,1 KB116,1 KBTuesdaySystem Reserved116,1 KBTuesday116,1 KB116,1 KB<	🔍 Search			451.1 KB	
a ddaEilbgraph.la859 bytesTuesdayDesktop16.1 KBTuesdayFile System105 GF filesyst105 GF filesyst116.1 KBTuesdayNew Volume1162 GF filesyst116.1 KBTuesdayS 2 GB Filesystem116 Jytes50.6 KBMonday 02 April 2012System Reserved116 Jytes50.6 KBMonday 02 April 2012Documents116 Jytes50.6 KBFriday 31 August 2012Documents116 Jytes116 Jytes50.6 KBVideos116 Jytes50.0 S29.7 KBFriday 31 August 2012Videos116 Jytes.o.029.7 KBFriday 03 August 2012Ubgs.so.9.058.4 MBSaturday 08 September 2012116 Jytes.o.02.1 MBFriday 02 March 2012116 Jytes.o.0.02.1 MBFriday 02 March 2012116 Jytes.o.0.02.1 MBFriday 02 March 2012116 Jytes.o.0.0241.3 KBFriday 02 March 2012116 Jytes.o.1.0.1249.5 KBThursday 05 November 2009	Recently Used				
Desktopii libgraph.la859 bytesTuesdayFile System1 bgraph.so.116.1 KBTuesday102 GB Filesystem1 bgraph.so.1.0.216.1 KBTuesdaySystem Reservel1 bgraph.so.450.6 KBMonday 02 April 2012Documents1 bigsraph.so.4.0.050.6 KBMonday 02 April 2012Documents1 bigsraph.so.3.0.029.7 KBFriday 31 August 2012Discurse1 bigs.so.98.4 MBSaturday 08 September 2012Downloads1 bigs.so.0.0.02.1 MBFriday 02 March 2012Disg.so.0.10.21 bigs.so.0.02.1 MBFriday 02 March 2012Disg.so.0.10.1249.5 KBFursday 05 November 2009	👼 ada			140.4 KB	Tuesday
File System'ibgraph.so.1116.1 KBTuesday102 GB Filesystem1163 KBTuesdayNew Volume116.1 KBTuesday25 CB Filesystem116.1 KBTuesdaySystem Reservel116.1 KBTuesday100 Cournents116.1 KBMonday 02 April 2012100 Cournents116.1 KBSo.6 KBMonday 02 April 2012100 Cournents116.1 KBFriday 31 August 201210 Downloads116.1 KBSaturday 08 September 201211 Uibgr.so.0.305.029.7 KBFriday 31 August 201211 Uibgr.so.0.305.029.7 KBFriday 03 August 201211 Uibgr.so.0.305.029.7 KBFriday 04 September 201211 Uibgr.so.0.305.029.7 KBFriday 02 March 201211 Uibgr.so.0.305.021.1 KBFriday 02 March 201211 Uibgr.so.0.16.021.1 KBFriday 02 March 201211 Uibgr.so.0.10241.3 KBFriday 02 March 201211 Uibgr.so.1.0.1249.5 KBThursday 05 November 2009				859 bytes	Tuesday
I 102 GB Filesyst I libgraph.so.1 116.1 KB Tuesday New Volume 52 GB Filesystem I libgraph.so.4 50.6 KB Monday 02 April 2012 System Reserved I libgraph.so.4 50.6 KB Monday 02 April 2012 I bgraph.so.4 50.6 KB Monday 02 April 2012 I bgraph.so.4 50.6 KB Monday 02 April 2012 I bgraph.so.4 50.6 KB Monday 02 April 2012 I bgraph.so.0 29.7 KB Friday 31 August 2012 I bgraph.so.0.305.0 29.7 KB Friday 31 August 2012 I bigs.so.9 8.4 MB Saturday 08 September 2012 I bigs.so.9.05 8.4 MB Saturday 08 September 2012 I bigs.so.0 2.1 MB Friday 02 March 2012 I bigs.so.0.6.0 2.1 MB Friday 02 March 2012 I bigs.so.0.6.0 2.1 MB Friday 02 March 2012 I bigs.so.0.6.0 2.1 MB Friday 02 March 2012 I bigs.so.0.16.0 2.1 MB Friday 02 March 2012 I bigs.so.0.10.0 241.3 KB Friday 02 March 2012 I bigs.so.1.0.12 49.5 KB Thursday 05 November 2009 I bigs.so.1.0.12 49.5 KB				116.1 KB	Tuesday
New Volume 116.1 KB Tuesday S2 GB Filesystem 116.1 KB Tuesday S2 GB Filesystem 116.1 KB Monday 02 April 2012 System Reserved 116.1 KB Monday 02 April 2012 Documents 116.1 KB Friday 31 August 2012 Documents 116.1 KB Friday 31 August 2012 Wusic 116.1 KB Friday 31 August 2012 Videos 116.1 KB Friday 31 August 2012 Videos 116.1 KB Saturday 08 September 2012 116.1 KB 116.1 KB KB Videos 116.1 KB Friday 02 March 2012 116.1 KB 116.1 KB KB 116.1 KB 116.1 KB Saturday 08 September 2012 116.1 KB		🗋 libgraph.so.1		116.1 KB	Tuesday
S2 GB Filesystem S0.6 KB Monday 02 April 2012 System Reserved S0.6 KB Monday 02 April 2012 Ibgraph.so.4.0 S0.6 KB Monday 02 April 2012 Ibgraph.so.4.0 S0.6 KB Monday 02 April 2012 Ibgraph.so.0 S0.6 KB Monday 02 April 2012 Ibgraph.so.0 S0.6 KB Monday 02 April 2012 Ibgraph.so.0 S0.6 KB Monday 02 April 2012 Ibgraph.so.0.305.0 29.7 KB Friday 31 August 2012 Ibgraph.so.0.305.0 S1.6 KB Saturday 08 September 2012 Ibgsl.so.9 8.4 MB Saturday 08 September 2012 Ibgsl.so.0 2.1 MB Friday 02 March 2012 Ibgsl.so.0.6.0 2.1 MB Friday 02 March 2012 Ibgsl.blas.so.0.0 241.3 KB Friday 02 March 2012 Ibgsl.blas.so.0.0.0 241.3 KB Friday 05 November 2009 Ibgsl.so.1.0.12 49.5 KB Thursday 05 November 2009				116.1 KB	Tuesday
System Reserved Signaph.so.4.0.0 So.4.8 Monday 02 April 2012 Ibgrip.so.0 29.7 KB Friday 31 August 2012 Ibgrip.so.0.305.0 29.7 KB Friday 31 August 2012 Ibgrip.so.9 8.4 MB Saturday 08 September 2012 Ibgsl.so.9.05 8.4 MB Saturday 08 September 2012 Ibgsl.so.0 2.1 MB Friday 02 March 2012 Ibgsl.so.0.6.0 2.1 MB Friday 02 March 2012 Ibgsl.blas.so.0.0.0 241.3 KB Friday 02 March 2012 Ibgsl.blas.so.1.0.12 49.5 KB Thursday 05 November 2009		🗋 libgraph.so.4 😓		50.6 KB	Monday 02 April 2012
iibgrip.so.0 29.7 KB Friday 31 August 2012 iibgrip.so.0.305.0 8.4 MB Saturday 08 September 2012 iibgs.so.905 8.4 MB Saturday 08 September 2012 iibgsl.so.0 2.1 MB Friday 02 March 2012 iibgsl.so.0.6.0 2.1 MB Friday 02 March 2012 iibgsl.sba.so.0.00 241.3 KB Friday 02 March 2012 iibgsl.sblas.so.0.0.0 241.3 KB Friday 05 November 2009 iibgsl.so.1.0.12 49.5 KB Thursday 05 November 2009		🗋 libgraph.so.4.0.0		50.6 KB	Monday 02 April 2012
Music 10 Bigh, So. 0.305.0 29.7 kB Friday 31 August 2012 Music 1 bigh, So. 0.305.0 8.4 MB Saturday 08 September 2012 I Videos 1 bigs, So. 0.16.0 2.1 MB Friday 02 March 2012 I bigs, So. 0.16.0 2.1 MB Friday 02 March 2012 I bigs, So. 0.16.0 2.1 MB Friday 02 March 2012 I bigs, So. 0.16.0 241.3 KB Friday 02 March 2012 I bigs, So. 0.00 241.3 KB Friday 02 March 2012 I bigs, So. 1.0.12 49.5 KB Thursday 05 November 2009	_ ,	🗋 libgrip.so.0		29.7 KB	Friday 31 August 2012
Impletures Impletures 8.4 MB Saturday 08 September 2012 Impletures Impletures 1mpletures 8.4 MB Saturday 08 September 2012 Impletures Impletures 1mpletures Saturday 08 September 2012 Impletures Impletures 1mpletures Saturday 08 September 2012 Impletures Impletures Saturday 08 September 2012		libgrip.so.0.305.0		29.7 KB	Friday 31 August 2012
Wideos 8.4 MB Sacurday 08 September 2012 Ibgsl.so.0 2.1 MB Friday 02 March 2012 Ibgsl.so.0.16.0 2.1 MB Friday 02 March 2012 Ibgsl.so.0.16.0 2.1 MB Friday 02 March 2012 Ibgsl.blas.so.0 241.3 KB Friday 02 March 2012 Ibgsl.blas.so.0.0 241.3 KB Friday 02 March 2012 Ibgsl.blas.so.1.0.12 49.5 KB Thursday 05 November 2009		🗋 libgs.so.9		8.4 MB	Saturday 08 September 2012
Ibgsl.so.0 2.1 MB Friday 02 March 2012 Ibgsl.so.0.6.0 2.1 MB Friday 02 March 2012 Ibgsl.blas.so.0 241.3 KB Friday 02 March 2012 Ibgsl.blas.so.0.0 241.3 KB Friday 02 March 2012 Ibgsl.blas.so.1.0 241.3 KB Friday 02 March 2012 Ibgsl.blas.so.1.0 241.3 KB Friday 02 March 2012 Ibgsl.blas.so.1.0.12 49.5 KB Thursday 05 November 2009		🗋 libgs.so.9.05		8.4 MB	Saturday 08 September 2012
LibgsLso.0.16.0 2.1 MB Friday 02 March 2012 LibgsLcblas.so.0 241.3 KB Friday 02 March 2012 LibgsLcblas.so.0.0 241.3 KB Friday 02 March 2012 Libgsm.so.1 49.5 KB Thursday 05 November 2009 Libgsm.so.1.0.12 49.5 KB Thursday 05 November 2009		🗋 libgsl.so.0		2.1 MB	Friday 02 March 2012
Ibipslcblas.so.0.00 241.3 KB Friday 02 March 2012 Ibipsm.so.1 49.5 KB Thursday 05 November 2009 Ibipsm.so.1.0.12 49.5 KB Thursday 05 November 2009	🔛 Downloads	libgsl.so.0.16.0		2.1 MB	Friday 02 March 2012
Libgsm.so.1 49.5 KB Thursday 05 November 2009 Libgsm.so.1.0.12 49.5 KB Thursday 05 November 2009		libgslcblas.so.0		241.3 KB	Friday 02 March 2012
Libgsm.so.1.0.12 49.5 KB Thursday 05 November 2009		libgslcblas.so.0.0.0		241.3 KB	Friday 02 March 2012
		🗋 libgsm.so.1		49.5 KB	Thursday 05 November 2009
🗋 libgssdp-1.0.so.3 49.8 KB Tuesday 03 January 2012		libgsm.so.1.0.12		49.5 KB	Thursday 05 November 2009
		🗋 libgssdp-1.0.so.3		49.8 KB	Tuesday 03 January 2012

Figure 65: Files/libraries to be added for graphic projects

- 5. Libraries shown in Ubuntu 12.04, 32-bit OS are *libgraph.a*, *libgraph.so*, *libgraph.so*.1, *libgraph*
- 6. After selecting all the required libraries click on Open. A new window labelled 'Question' will open asking 'Keep this as relative path' as shown in figure 66. Click on No.



Figure 66: Relative or absolute path for files/libraries

7. The libraries will be linked using absolute path as shown in figure 67. Click on 'Ok'.

🛛 🖪 Project	build options
line	Selected compiler
Debug Release	GNU GCC Compiler 🗘
	Compiler settings Linker settings Search directories Pre/post build steps
	Policy: Append target options to project options
	Link libraries: Other linker options:
	Add library File: /usr/lib/libgraph.so;/usr/lib/libgraph.so
	Cancel OK
	Þ
	Add Edit Delete Clear Copy selected to
	Cancel VK

Figure 67: libraries selected

8. We are now re-directed to *linker settings* tab. The added libraries are shown. In *Other linker options* window write '-lgraph'. The final setup is shown in figure 68 with all the required 6 libraries and '-lgraph'. Click on 'Ok' to get back to editor. Now we are ready to build the project with *graphics.h* header file

😣 🗈 Project	build options
line Debug Release	Selected compiler GNU GCC Compiler
	Compiler settings Linker settings Search directories Pre/post build steps
	Policy: Append target options to project options
	Link libraries: Other linker options:
	/usr/lib/libgraph.a /usr/lib/libgraph.so.1 /usr/lib/libgraph.so.1.0.2 /usr/lib/libgraph.so.4 /usr/lib/libgraph.so.4.0.0
	Add Edit Delete Clear Copy selected to
	Cancel VK

Figure 68: Libraries added to project

9. Now click the *build and run* button from compiler bar and output will be displayed as shown in figure 69.

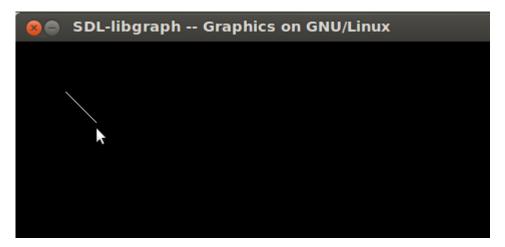


Figure 69: Output for line.c

4.5 Opening Existing Program/Project

Click 'File' \rightarrow 'Open'. Browse to desired directory and select the file with .cbp extension as shown in figure 70 and click Open.

Places	Name	~	Size	Modified
🔍 Search	📄 bin			Wednesday 27 November 201
Recently Used	📄 obj			Wednesday 27 November 201
ada	🖪 line.cbp		1.3 KB	Thursday 28 November 2013
Desktop	🗋 line.depend		106 bytes	Friday 29 November 2013
File System	🙆 line.layout		238 bytes	Yesterday at 17:13
102 GB Filesyst	🔄 line_build_log.html		750 bytes	Friday 29 November 2013
New Volume	📓 main.c		181 bytes	Friday 29 November 2013
52 GB Filesystem				
System Reserved				
Documents				
Music				
Pictures				
Videos				
🔋 Downloads				

Figure 70: Select file with .cbp extension to open an existing project

References

- The Code::Block Team. Code::block homepage http://www.codeblocks.org/.
- [2] The Code::Block Team. Gpl v3.0 license http://www.codeblocks.org/license.
- [3] Installing code::block on ubuntu http://wiki.codeblocks.org/index.php?title=Installing_Code::Blocks_from_source_on_ Linux.
- [4] Eternal thinker: How to use graphics.h in ubuntu? http://blog.eternal-thinker.com/2010/09/how-to-use-graphicsh-in-ubuntu.html.
- [5] Code::block faq's http://wiki.codeblocks.org/index.php?title=FAQ.