Setting Up and Customizing Code::Blocks

(CS 1428 Guide)
Table of Contents

Setting up Code::Blocks – Getting Started.................................................................................................................................................. 2
Using a single source file without a project.................................................................................................................................................. 2
  Creating the source file................................................................................................................................................................................. 2
  Compiling the source file ............................................................................................................................................................................. 4
Using a project ........................................................................................................................................................................................................ 5
  Making a new project .................................................................................................................................................................................. 5
  Renaming the source file .......................................................................................................................................................................... 7
  Compiling the source file ........................................................................................................................................................................... 7
Creating Input Files with the Editor .............................................................................................................................................................. 8
Customizing the Editor .................................................................................................................................................................................................. 8
  Putting in a code wrap line.......................................................................................................................................................................... 8
  Setting the Tab Size and/or Converting Tabs to Spaces ......................................................................................................................... 9
    About Tabs ..................................................................................................................................................................................................... 9
    How to configure tab settings.................................................................................................................................................................. 10
Changing the Color Scheme for Readability .................................................................................................................................................. 11
Setting Default Code ................................................................................................................................................................................................ 12
Printing from Code::Blocks – Changing the default paper size .................................................................................................................. 14
Setting up Code::Blocks – Getting Started

When you first open up Code::Blocks, you should be prompted with this screen:

![Compiler auto-detection screen](image)

**GNU GCC Compiler** should be the first one. Make sure to click on it, and then click **Set as default**.

The next setting Code::Blocks will ask about is whether you want to associate C and C++ files with Code::Blocks. This means that whenever you double click on a file ending in .c or .cpp, it will open in Code::Blocks. Choose whichever option you like.

![File associations](image)

Unlike Eclipse, Code::Blocks will not force you to make a project for a single source file. This gives us the option of either making just source files, or making projects. **This is the preferred method for classes taught by Trish Sumbera.**

**Using a single source file without a project**

**Creating the source file**

In the top left corner, click on **File**

Click on **New**, then click on **File...**
Click on **C/C++ source** then click **Go**

Click on **Next**

Select **C++** then click **Next**

Click on the ‘…’ to start searching for where to save your file.
Navigate to where you would like to save your file, then enter the name of your source file. You do not need to end it with `.cpp`.

Click **Save**, then **Finish**.

Your file will immediately open up in a tab and is ready to be edited.

Once you have something ready to be compiled, the process is the same as Eclipse.

**Compiling the source file**

We’re going to

**Save**

**Build**

and **Run**
Using a project

Making a new project

A window will pop up with different categories of projects available. We want to choose **Console application** in the top right corner.

**Project Title** is what you’ll call the folder that your actual source file(s) will be saved in. As for **Folder to create project in**, I suggest making a workspace folder somewhere for you to store all your projects in.

Code::Blocks will fill in the **Project filename** and **Resulting filename**. Do not worry about either of these 2.
As long as you set **GNU GCC Compiler** to be your default compiler, you shouldn’t have to do anything on this page. Click finish.

You are now in the project you just made! Code::Blocks has also conveniently made a hello world project for you, so you don’t have to start from scratch. To see this file, click on the + to the left of the **Sources** tab. It’ll be called **main.cpp**

Double click on **main.cpp** to open it in a tab.
Renaming the source file

If you wish to rename your source file, you must first close it if it is open in a tab. Then you can right click on the file in the sidebar, and select **Rename file**.

Make sure your new file name still ends with `.cpp`

Now double click to reopen the file in a tab.

Compiling the source file

We’re going to

**Save**

**Build**

and **Run**
Creating Input Files with the Editor

Many of your programs may require the reading of one or more input files. Because different editors may treat end of line and end of file indicators differently you may want to just use Code::Blocks to create the input files you will be using to test your programs.

To create an input file while in Code::Blocks:

  Click on File in upper left-hand corner.
  Click on New and select ‘Empty file’
  Type in the input file data as desired
  Click on ‘File’ and ‘Save File as’
  Navigate to the same folder/directory in which your source code file is stored
  In the ‘File Name’ field, type in the desired name of the file (be sure to include the .txt file extension)
  Click ‘Save’

Customizing the Editor

Putting in a code wrap line

First click on Settings in the top bar.

A drop down menu will appear. Click on editor.
Once in the editor menu, click on **Margins and caret** on the left side bar.

![Configure editor](image)

Setting the Tab Size and/or Converting Tabs to Spaces

**About Tabs**

**Tab Size**

The default tab size in Code::Blocks is 4. That is, every time you hit the tab key the cursor is moved over 4 spaces. If you use the tab key for indentation your indentation amount automatically then becomes 4 spaces. If you prefer to use a different number of spaces for code indentation then it will be extremely useful to reset the tab size. Too small a tab size results in indentation that is difficult to discern. Too large a tab size results in code being shifted too far to the right so that it often does not fit in a reasonably sized screen window and you find yourself constantly scrolling to the right to read the rest of a line of code.

**Tabs vs Spaces**

Using tabs to create indentation really speeds up the process of creating source code. However, the size (number of spaces) that a ‘tab’ stroke moves the cursor (and thus the immediately following text) is defined by the environment (editor) in which it is displayed. You can create a source code file in the Code::Blocks editor where the tab size is 4 and then view the same file in Notepad (or some other editor) where the tab size is defined differently and readability of your code may be compromised. Of more note to students, the version printed from this other editor may not display the formatting characteristics required by the instructor and your grade could suffer. Tabs vs. Spaces is an ongoing argument in the programming community.
One way to avoid the display issue with tabs is to configure the editor in your IDE to expand tabs to spaces. Instead of inserting a ‘tab character’ in the file, the editor will insert the defined number of spaces. Doing this insures that the document displays the same in all editors. The down side is that removal of tab indentation which normally required one keystroke would then require ‘tab size’ keystroke as each space has to be removed.

**How to configure tab settings**

First click on **Settings** in the top bar.

A drop down menu will appear. Click on editor.

Once in the editor menu, click on **Source formatter** on the left side bar. You will have to scroll the page down to see it.

You should see the following window.

To set the tab size, click on the Indentation tab and set the number of spaces you want the tab size to be.

Click OK.
To tell the compiler editor to Convert TABs to space, click on the Formatting tab and check the box by **Convert TABs to spaces.**

Click OK.

**Changing the Color Scheme for Readability**

First click on **Settings** in the top bar.

A drop down menu will appear. Click on editor.

Once in the editor menu, click on **Syntax highlighting** on the left side bar.

You can create your own preferred color scheme; however, the changes here that are most important to your CS 1428 instructor (or anyone else needing to read printouts generated from Code::Blocks) is to darken the color used for comments so that they are readable when printed.
In the top half of the Syntax highlighting window you should see a list of syntax categories for which the display color can be configured. For each of the categories that start with Comment perform the following steps:

Use the mouse to highlight a category.

To the right click on box under Foreground.

A window with color choices will pop up. Select a dark color choice such as maroon or brown and click OK.

![Syntax Highlighting Window]

You will probably want to avoid selecting a color that is used for other frequently used categories such as dark blue for keywords or the default black. The results of your choice of color can be seen in the code sample window.

**Setting Default Code**

Default code is code that you will want to include in almost every source code file that you create. For example, for code that you generate for your CS 1428 (or other) class you might want each new source code file to start off with a file header comment that includes the filename of the project, your full name, your NetID, your course section number and possibly other items required by your instructor. For my classes the following might be a good choice for default code. Things inside <> will be filled in for each source code file.

```cpp
/*
 * <filename>.cpp
 *
 * John Smith (or whatever your name is)
 * js123 (or whatever your NetID is)
 * Section # <999> (or whatever your course section is)
 *
 * <Program description>
 * 
 *
 */

using namespace std;
#include <iostream>

int main()
{
    return 0;
}
```
To create this default code (or template) follow the step below.

First click on **Settings** in the top bar.

A drop down menu will appear. Click on editor.

Once in the editor menu, click on **Default code** on the left side bar.

The following screen will appear.

Type the code you want included in every source file in the blank window and click OK.
Printing from Code::Blocks – Changing the default paper size

Some of the computers in the campus and CS labs may have an incorrect paper size configured as the default for Code::Blocks. The incorrect size is A4 which is a European paper size that is slightly longer than letter. If you print a document without correcting the paper size, the print may run off the end of the paper and some of the lines of code may get skipped.

To print out of Code::Blocks select File on the top menu bar and then Print. You will get a small pop-up window that allows you to make color selections, etc. Select Print. You should get a window similar to the one below.

Click on Preferences to get another window with two tabs at the top. Click on the Paper/Quality tab to get the window on the right below.

If A4 appears in Page Size selection, click on the drop down arrow and select Letter. You will probably need to do this every time you want to print a document.