Compiling C modules

Table of Contents

modules in C

C programming projects in industry have more than one source file because it usually takes more than one programmer to complete the job. However, only one of those .c source files in an application contains `int main() { }`. Other .c source files are known conceptually as modules. A module's source code contains functions() which work independently or together with other modules. A "main" program calls those functions.

Source files making up an application are grouped together in a Visual Studio IDE Project or in the same folder/workspace when using Visual Studio Code or other development tools including command line compilation.

A typical C application has .h header files, .c module files, and a single main.c source file which calls functions in the modules.

TL;DR for macOS

- Visual Studio Code or Xcode are good choices for C development.

TL;DR for Windows

- Visual Studio IDE is the usual choice for C development on Windows.
- Visual Studio Code, with a lighter weight and smaller memory footprint, is generally up to the task as of 2022.

TL;DR for command line compilation

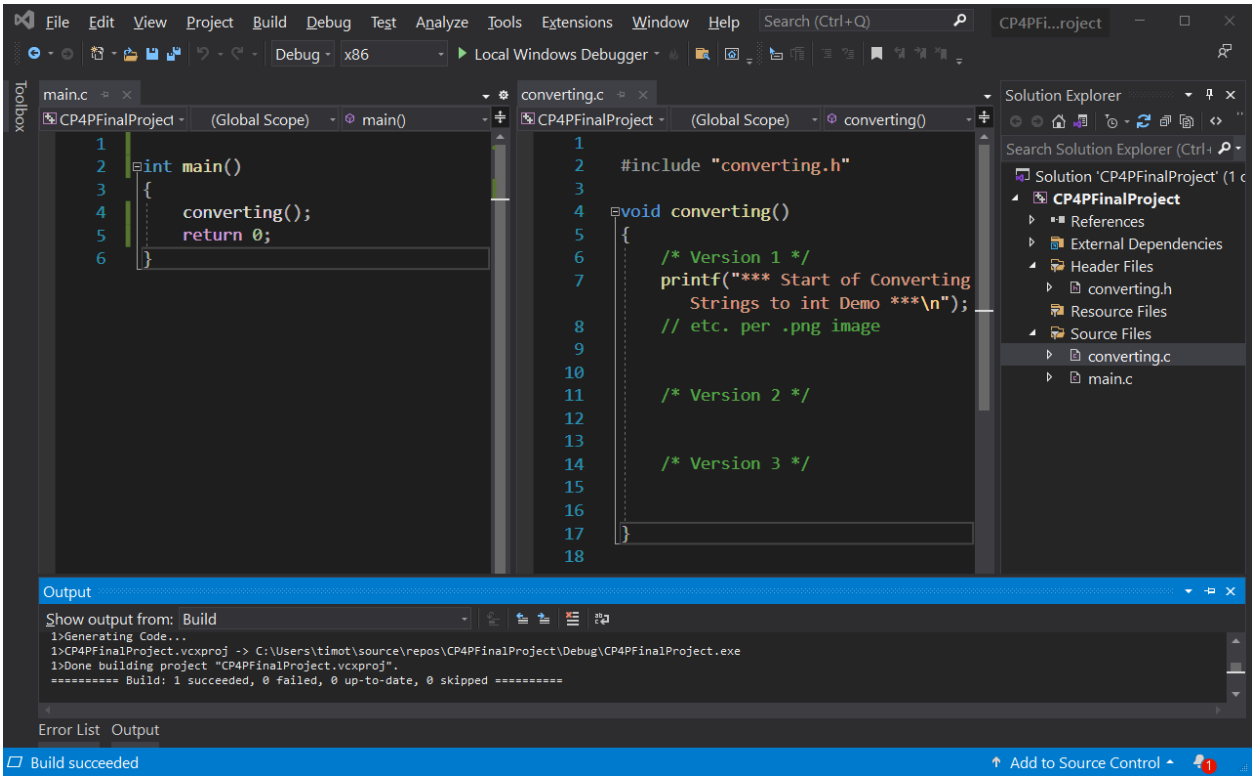
- Microsoft Windows cl compiler: https://youtu.be/rqLbyj0TnIg (see notes below)
- macOS gcc compiler: with Visual Studio Code or Xcode, a gcc command line compiler is included. gcc compiler on Mac: https://youtu.be/we2Oc4WQ7FM
- The gcc compiler is native to the Unix / Linux world. If you are going all hardcore in Windows 10+, do it in the Windows Subsystem for Linux where gcc is very happy.

- You can install gcc to run under Windows. You can also walk across Canada. In both cases, you need a really good reason. There are numerous issues to deal with: good boots and the MinGW-w64_WinLibs.docx file. (You'll need a good walk afterward.)

helloWorld C source to test a compilation

```
/*
helloWorld : the canonical test of any programming language thanks to K&R.
*/
#include <stdio.h>      // Standard Input/Output
int main(void)          // mainline – only one in an application
{
    // console output as proof of compiler installation and operation
    printf("Hello, World!\nThis is a compiler test.\n");
    return 0;
}
```

project C source template example



Your project or workspace/folder/directory contains three files:

- *moduleName*.h header file
- *moduleName*.c function file
- main.c with `int main() {` which calls the function inside *moduleName*.c }

Command line compilation of Final Project source files

To compile your module for unit testing on

Microsoft cl — see next page

macOS or Linux gcc
`> gcc moduleName.c main.c -o main`
e.g.
`> gcc converting.c main.c -o main`

To compile all modules into a program for Integration testing,
specify all the module names:

`> gcc moduleA.c moduleB.c moduleC.c moduleD.c main.c -o main`

e.g.

`> gcc fundamentals.c manipulating.c converting.c tokenizing.c main.c -o main`

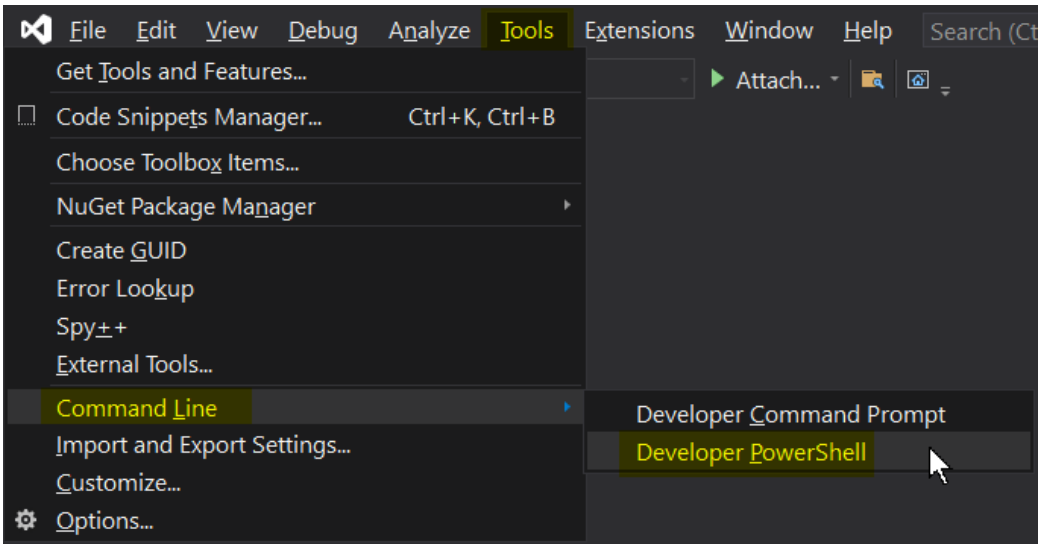
To compile a module only and make it runnable: (works only in gcc)

`> gcc -nostartfiles moduleA.c -o module`

e.g. `gcc -nostartfiles converting.c -o converting`

running the Microsoft cl compiler

You can start the `cl` compiler only from a Visual Studio developer command prompt.




```
**********************************************************************
** Visual Studio 2019 Developer PowerShell v16.11.5
** Copyright (c) 2021 Microsoft Corporation
**********************************************************************
PS C:\Users\me\source\repos\CP4PFinalProject>
   ————————————————————————————————> same as VS Project
   cd "C:\Users\me\Documents\Seneca\CPR101\Final" –> as required

PS C:\Users\me\source\repos\CP4PFinalProject>
   cl .\moduleName.c  .\main.c  /link  /out:main.exe
   cl .\converting.c  .\main.c  /link  /out:main.exe

Microsoft (R) C/C++ Optimizing Compiler Version 19.29.30136 for x86
Copyright (C) Microsoft Corporation.  All rights reserved.

converting.c
main.c
Generating Code...
Microsoft (R) Incremental Linker Version 14.29.30136.0
Copyright (C) Microsoft Corporation.  All rights reserved.

/out:converting.exe
/out:main.exe
converting.obj
main.obj
PS C:\Users\timot\source\repos\CP4PFinalProject> .\main.exe
*** Start of Converting Strings to int Demo ***
...
```