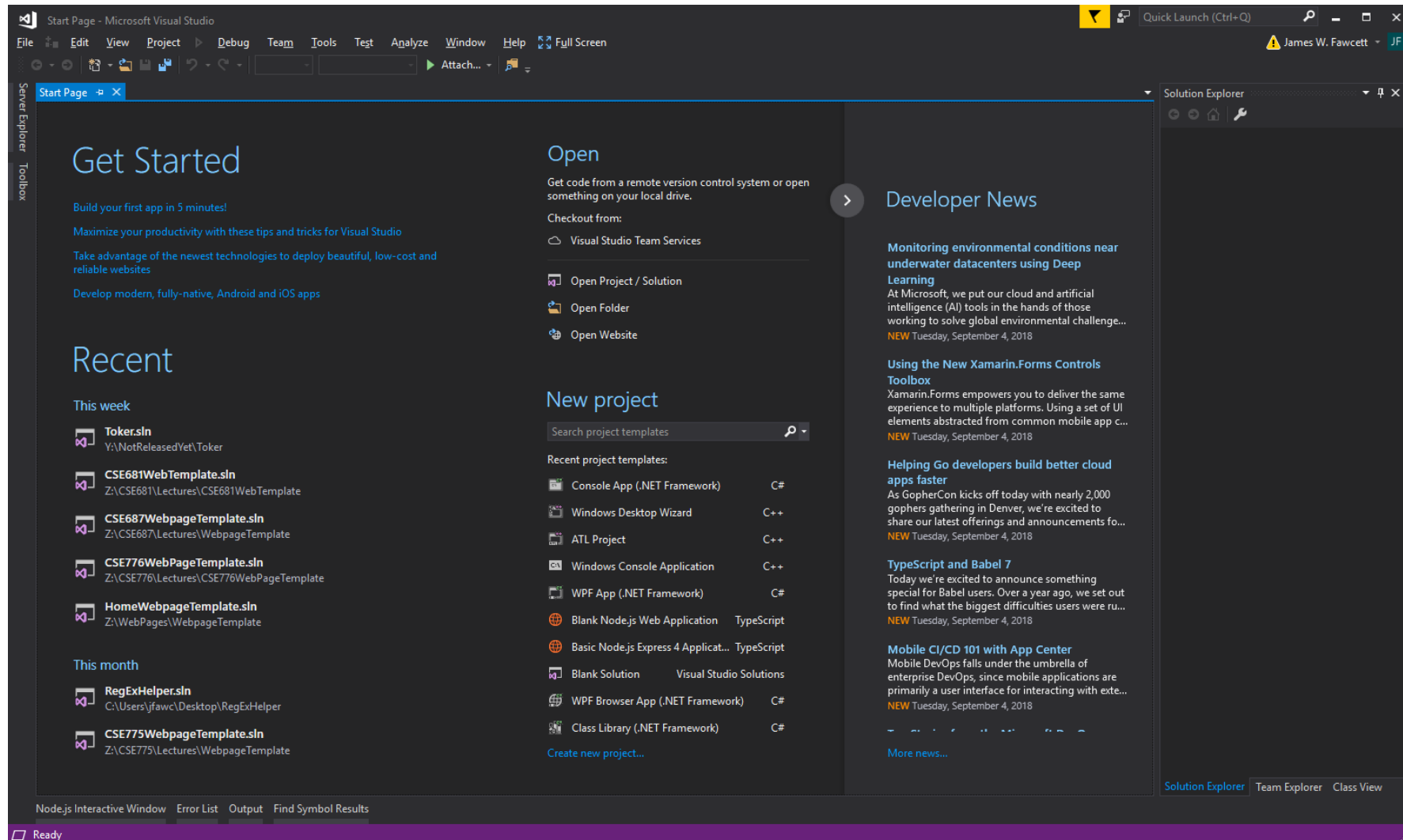


Visual Studio 2017 Help Screenshots

Jim Fawcett

Fall 2018

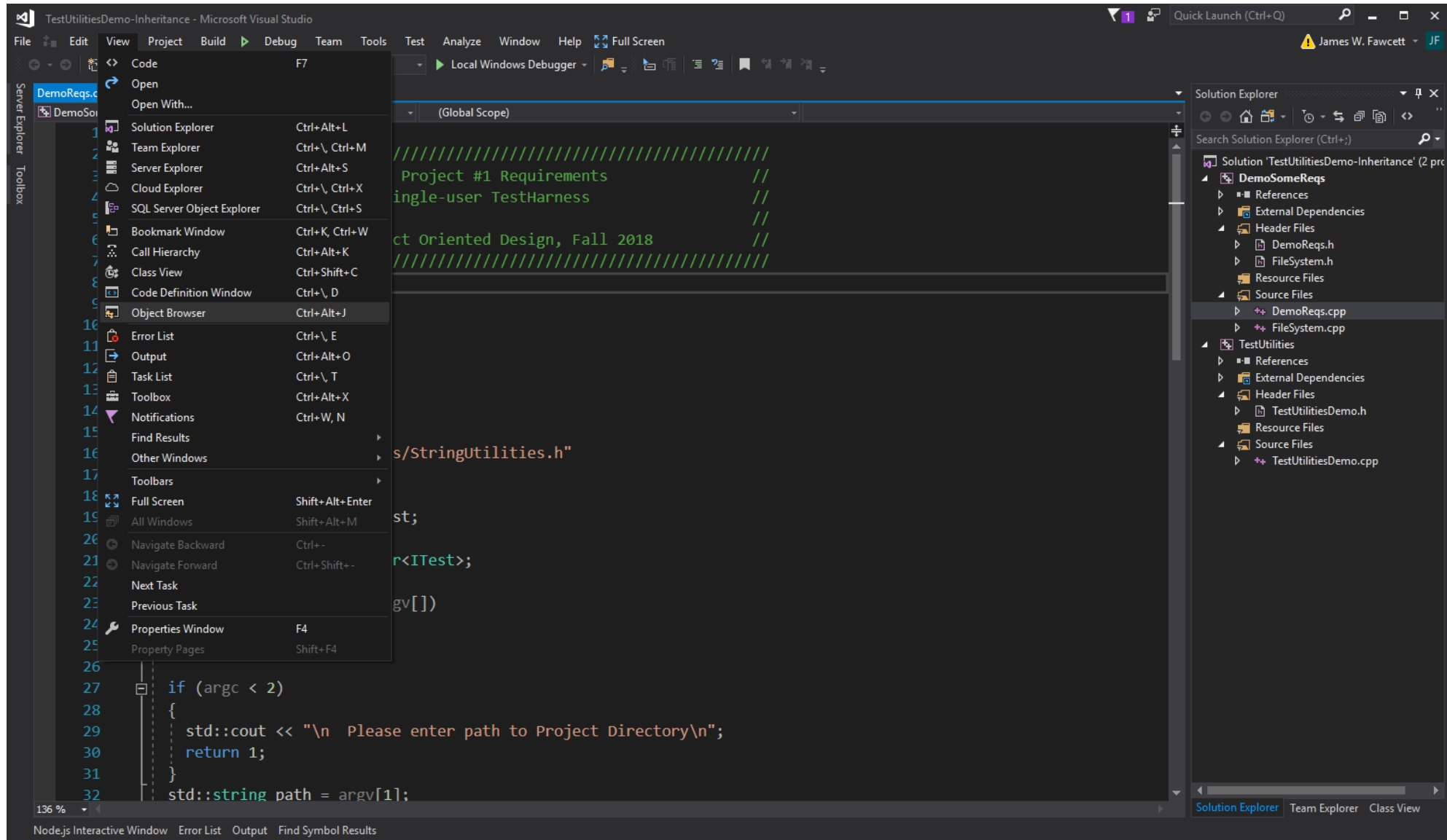
Opening Screen



View Existing Projects

Perhaps code you've downloaded from College Server

View



View > Object Browser

The screenshot shows the Microsoft Visual Studio interface with the Object Browser open for the file DemoReqs.cpp. The Object Browser displays the following structure:

- Global Functions and Variables
- Global Using Aliases and Typedefs
- Macros and Constants
- FileSystem
 - Block
 - Directory
 - File
 - FileInfo
 - Path
 - FileSystemSearch
- UtilitiesTest
- TestUtilities

The methods listed for the Path class are:

- fileSpec(const std::string & path, const std::string & name)
- getExt(const std::string & fileSpec)
- getFullFileSpec(const std::string & fileSpec)
- getName(const std::string & fileSpec, bool withExt = true)
- getPath(const std::string & fileSpec)
- toLower(const std::string & src)
- toUpper(const std::string & src)

The Solution Explorer on the right shows the project structure for 'TestUtilitiesDemo-Inheritance' (2 projects):

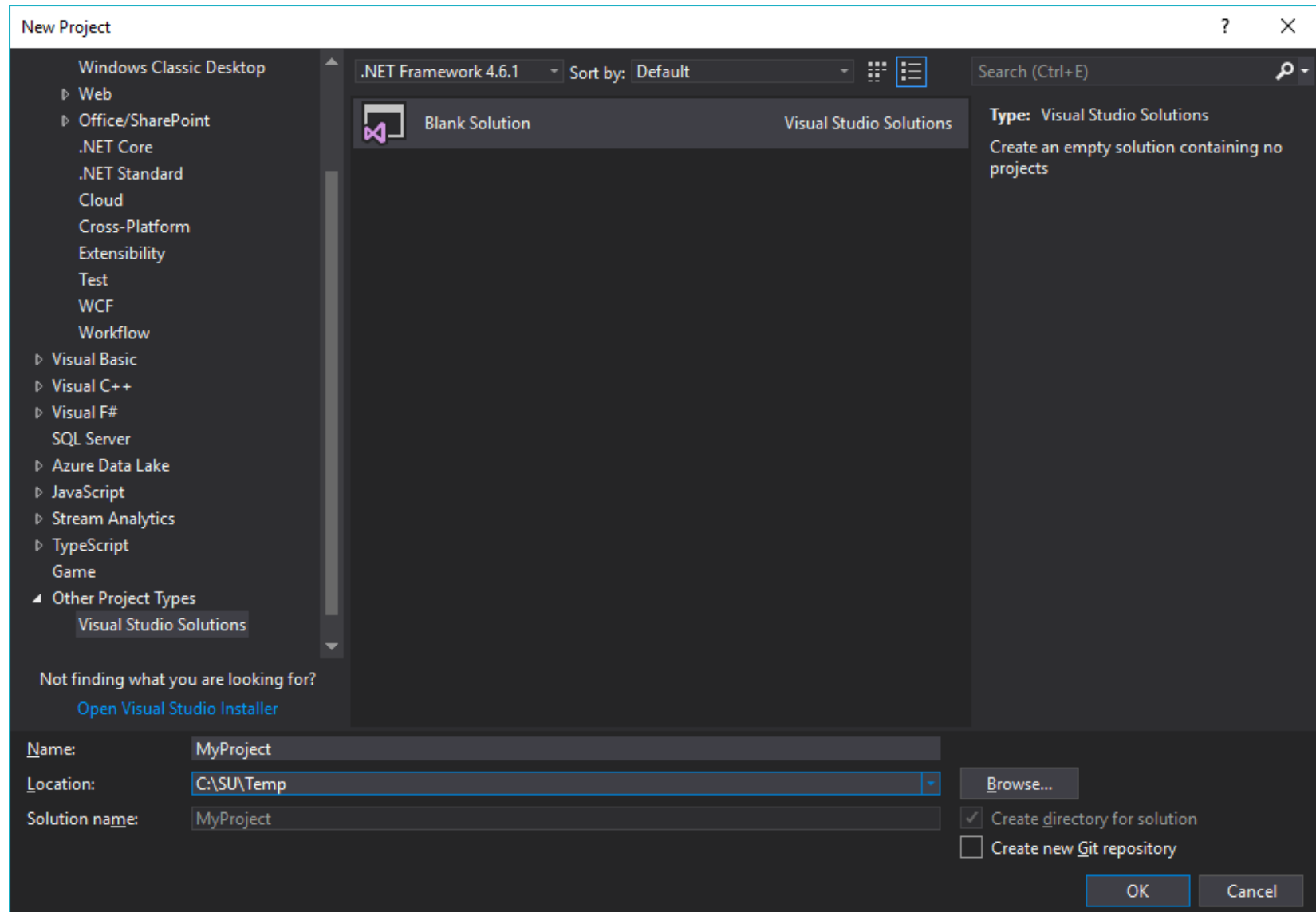
- DemoSomeReqs
 - External Dependencies
 - Header Files
 - DemoReqs.h
 - FileSystem.h
 - Resource Files
 - Source Files
 - DemoReqs.cpp
 - FileSystem.cpp
- TestUtilities
 - References
 - External Dependencies
 - Header Files
 - TestUtilitiesDemo.h
 - Resource Files
 - Source Files
 - TestUtilitiesDemo.cpp

The class name `class FileSystem::Path` is visible at the bottom of the Object Browser.

Start a new solution

File > New > Project > Other Project Types

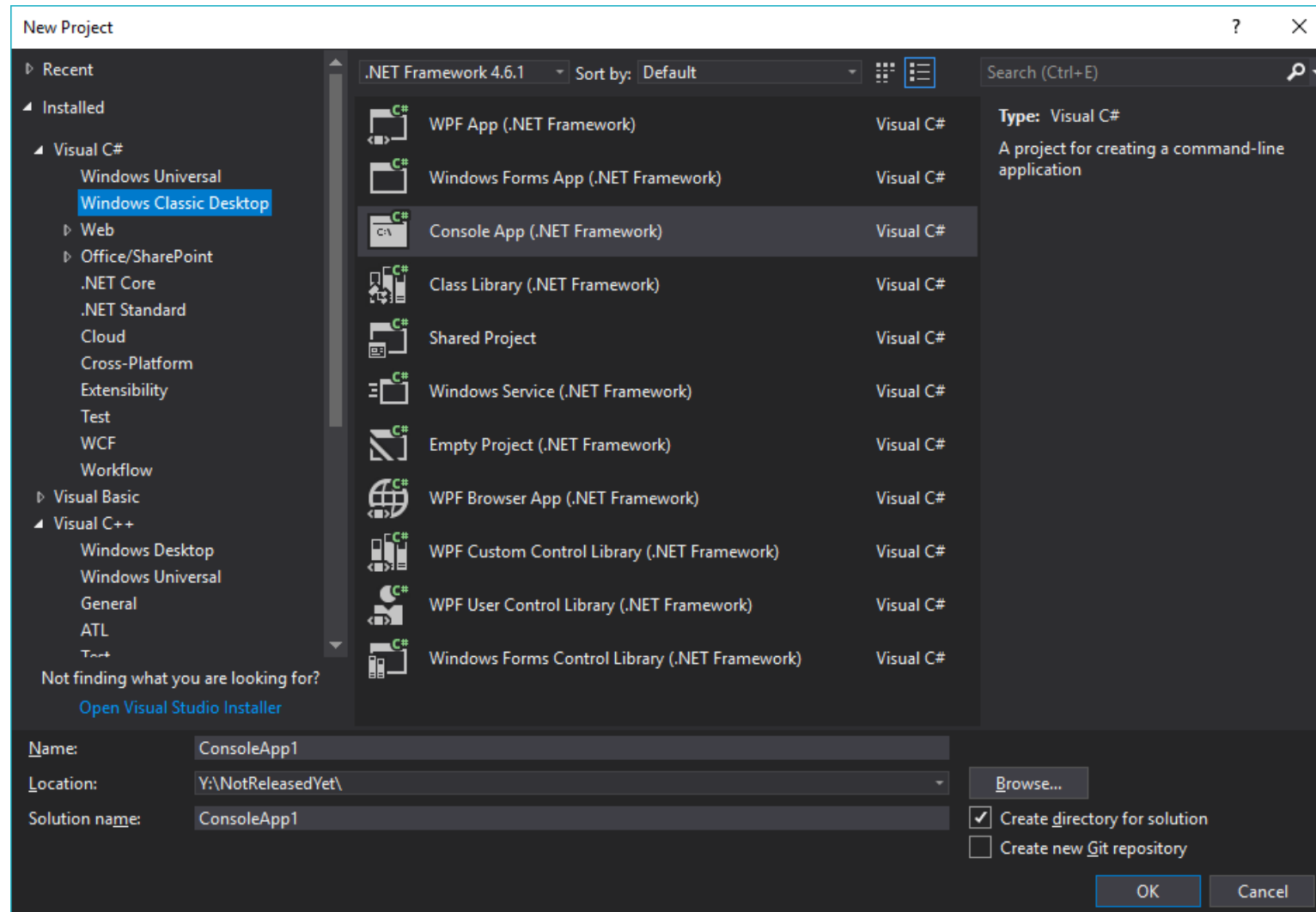
Start a new solution



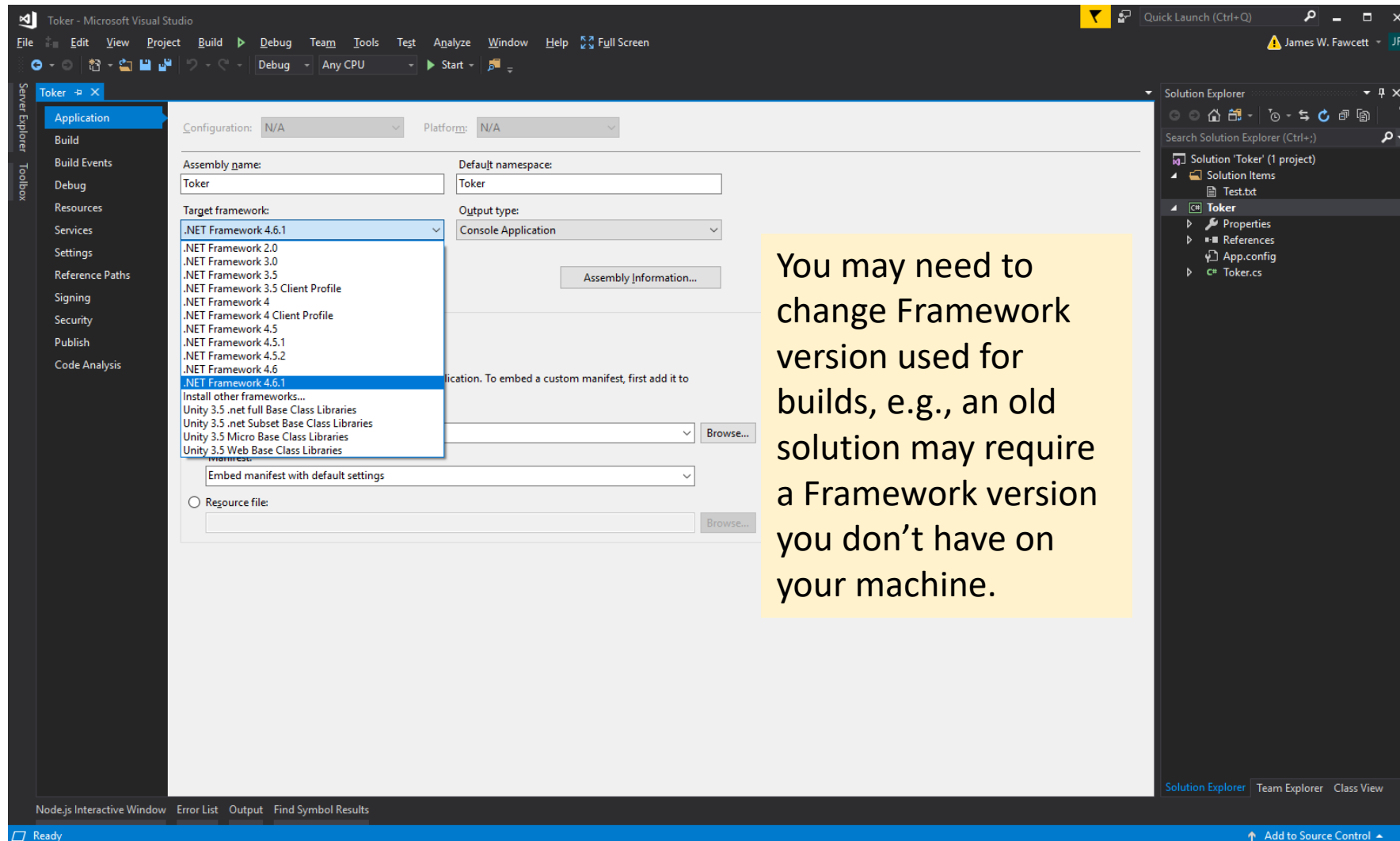
Starting and Building Projects

Perhaps to add a project to an existing solution

File > New > Project > C#



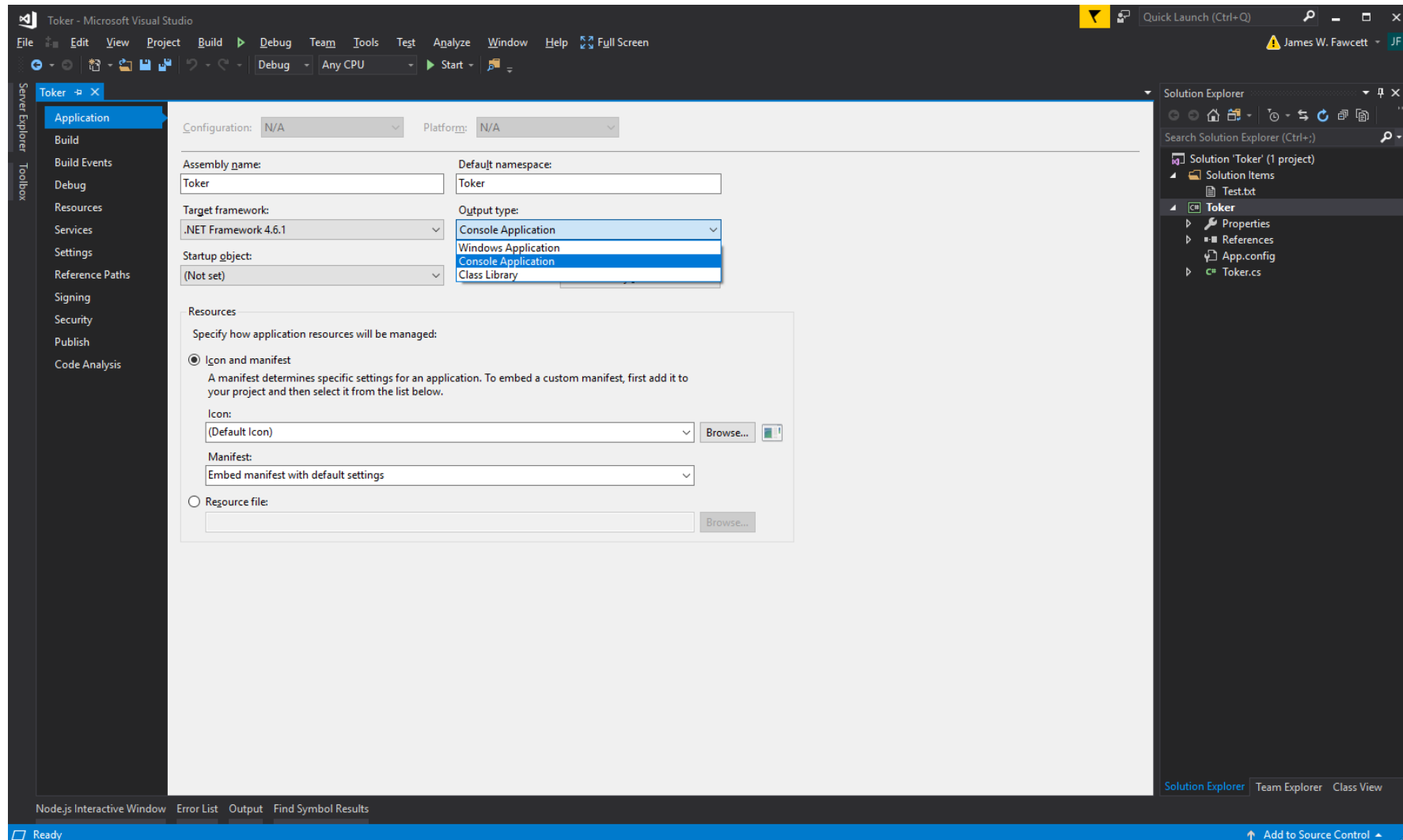
C# Project > Properties > Application



The screenshot shows the Visual Studio interface with the 'Application' tab selected in the Properties window. The 'Target framework' dropdown menu is open, showing a list of .NET Framework versions from 2.0 to 4.6.1. The 'Output type' is set to 'Console Application'. The 'Assembly name' and 'Default namespace' are both set to 'Toker'. The 'Solution Explorer' on the right shows the project structure for 'Toker', including 'Properties', 'References', 'App.config', and 'Toker.cs'.

You may need to change Framework version used for builds, e.g., an old solution may require a Framework version you don't have on your machine.

C# Project > Properties > Application

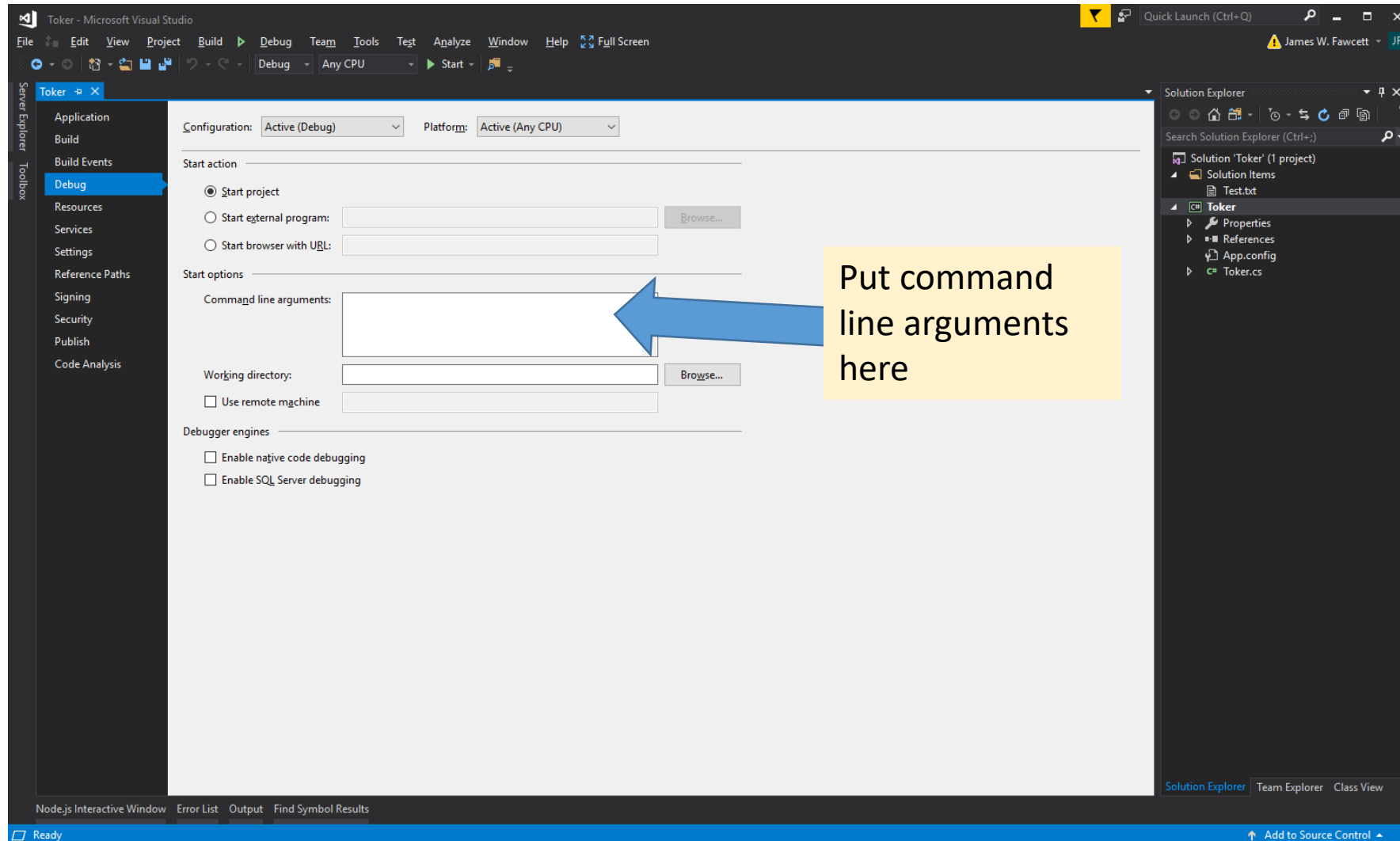


C# Project > Properties > Build

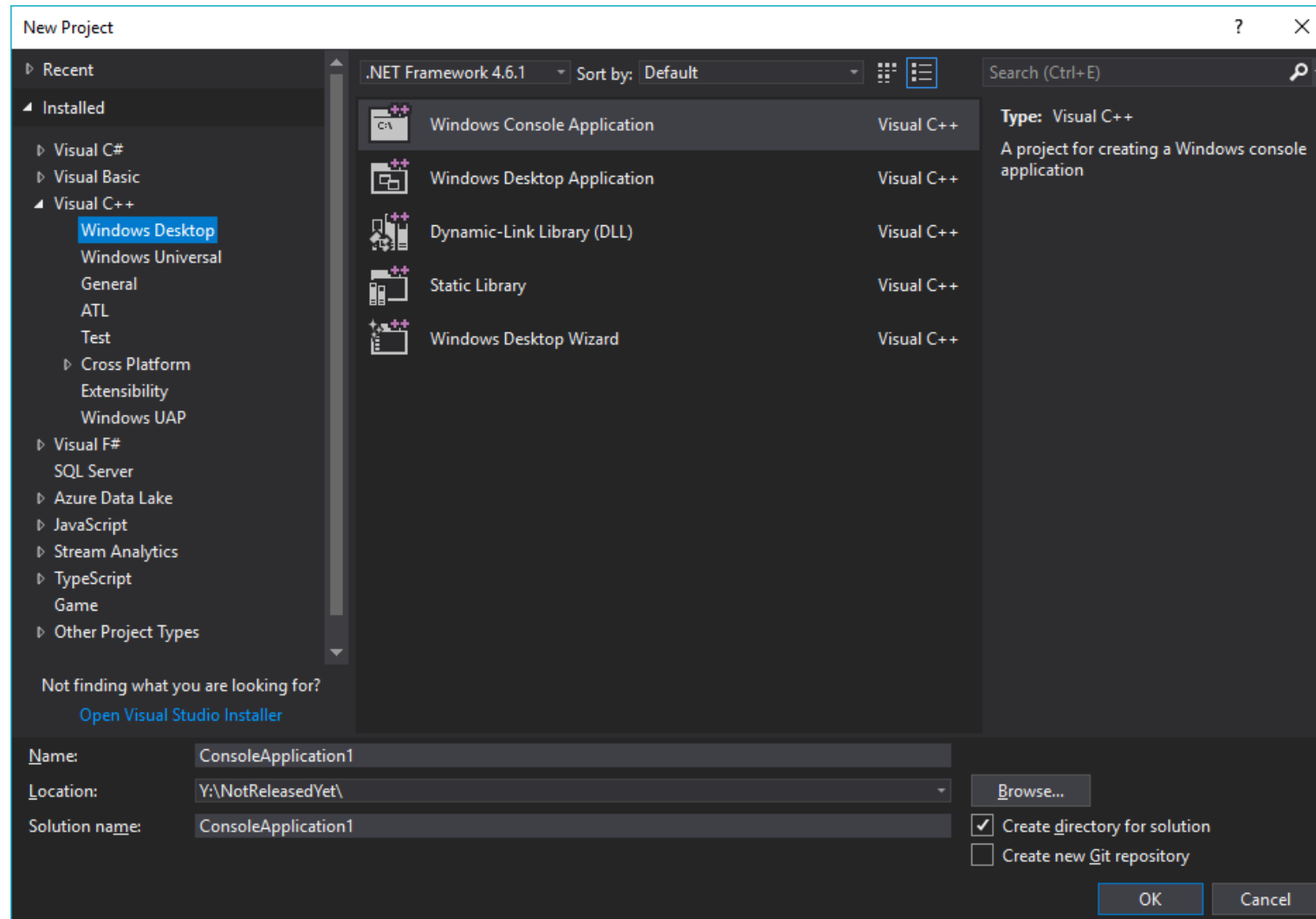
The screenshot shows the Visual Studio interface with the 'Build' properties dialog open for a project named 'Toker'. The 'General' tab is selected, and the 'Conditional compilation symbols' field is set to 'TEST_TOKER'. A blue arrow points from a yellow callout box to this field. The callout box contains the text: 'Define to enable compilation of TestStub'. Other settings visible include 'Platform target' set to 'Any CPU', 'Warning level' set to '4', and 'Output path' set to 'bin\Debug\'. The 'Solution Explorer' on the right shows the project structure, including 'Toker.cs'. The status bar at the bottom displays an error message: 'Error HRESULT E_FAIL has been returned from a call to a COM component. The tab has been closed.'

Define to enable compilation of TestStub

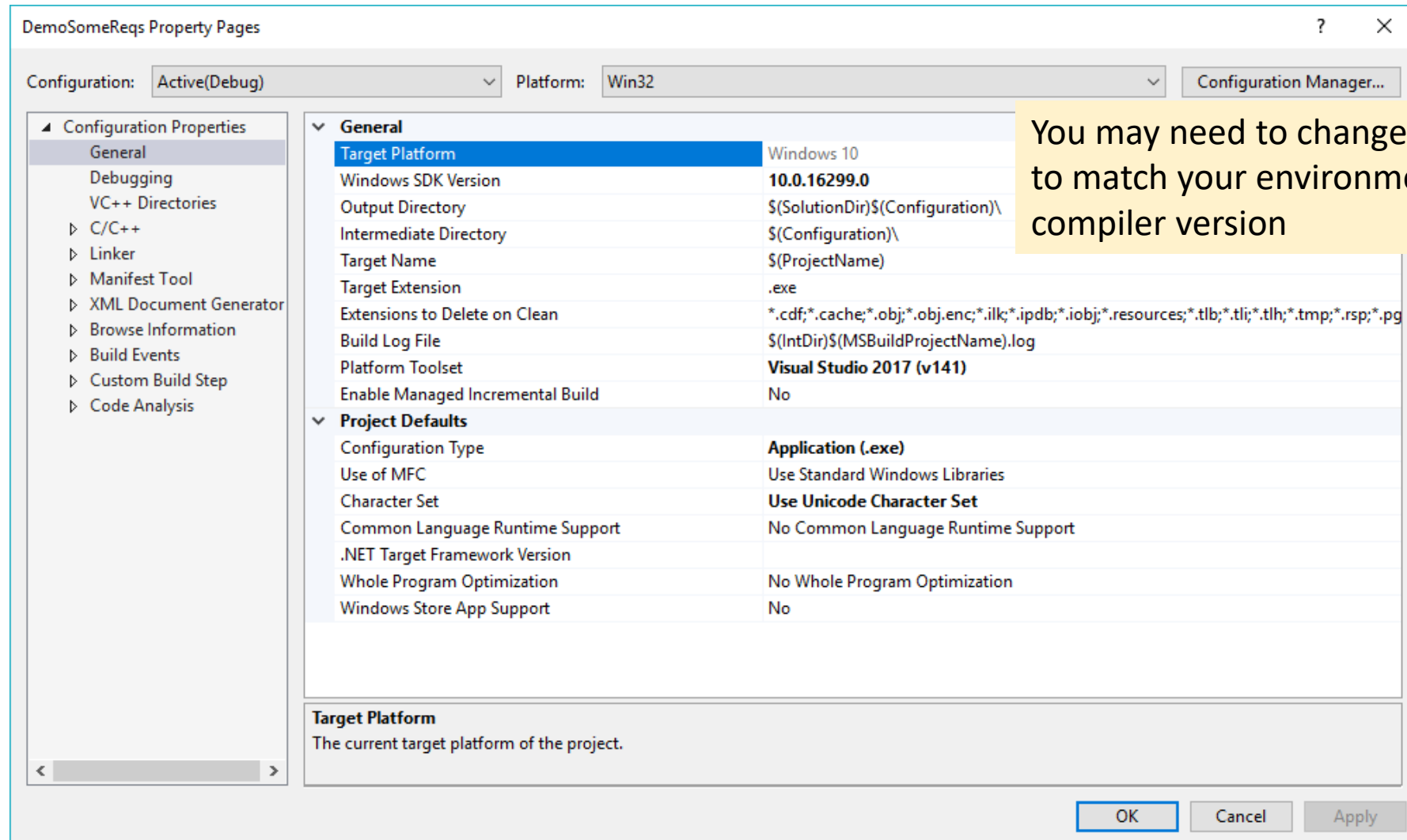
C# Project > Properties > Debug



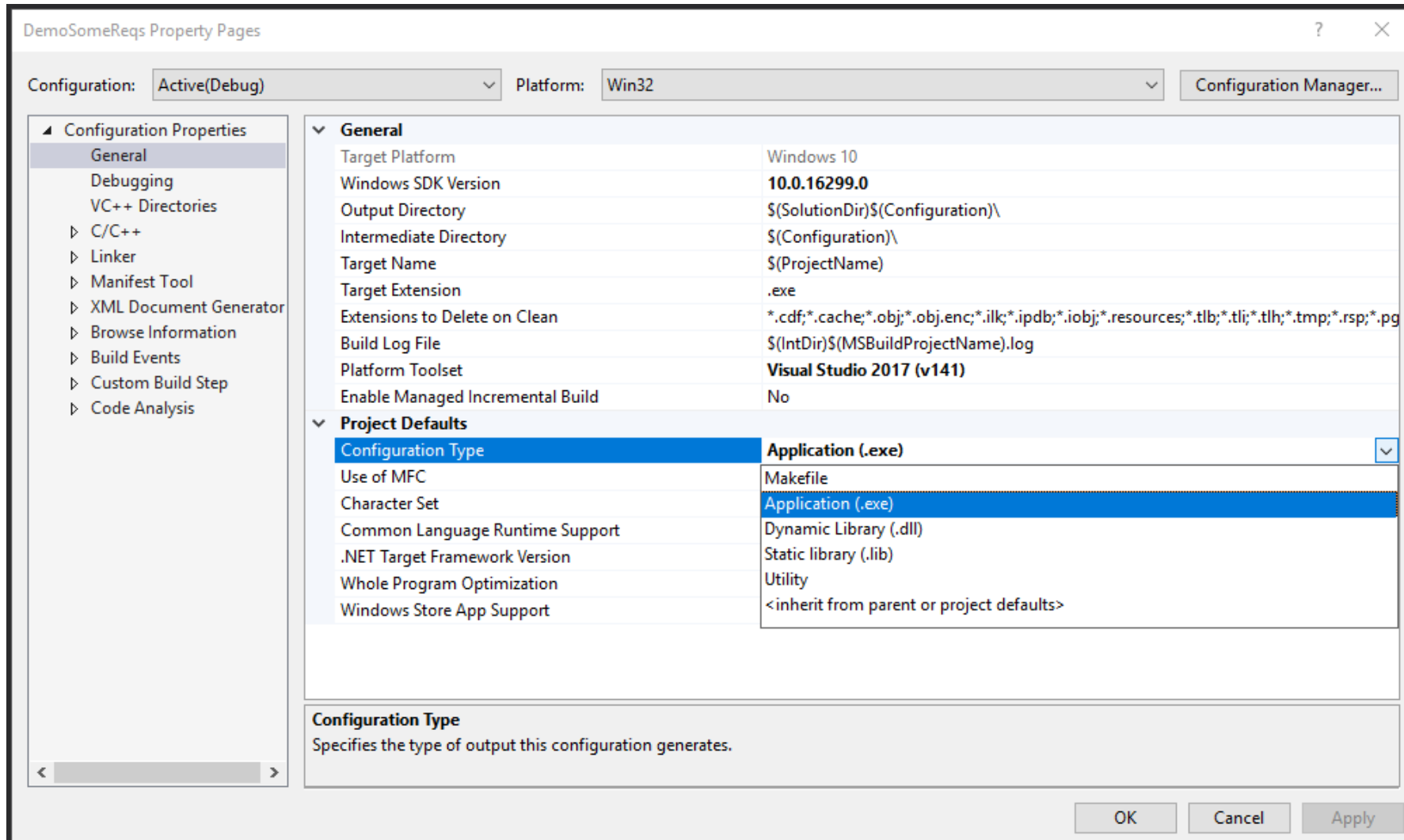
File > New > Project > C++



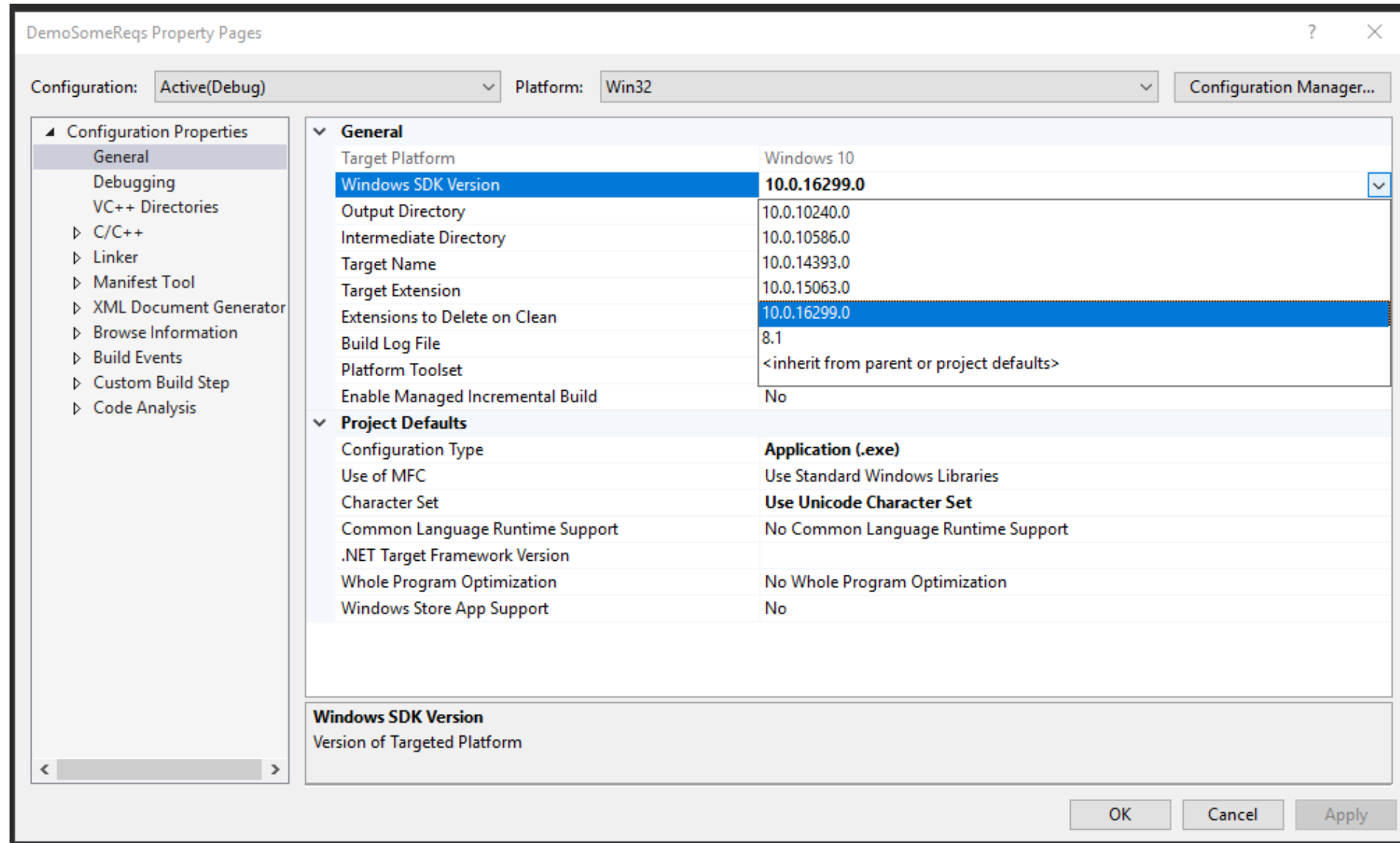
C++ Project > Properties > General



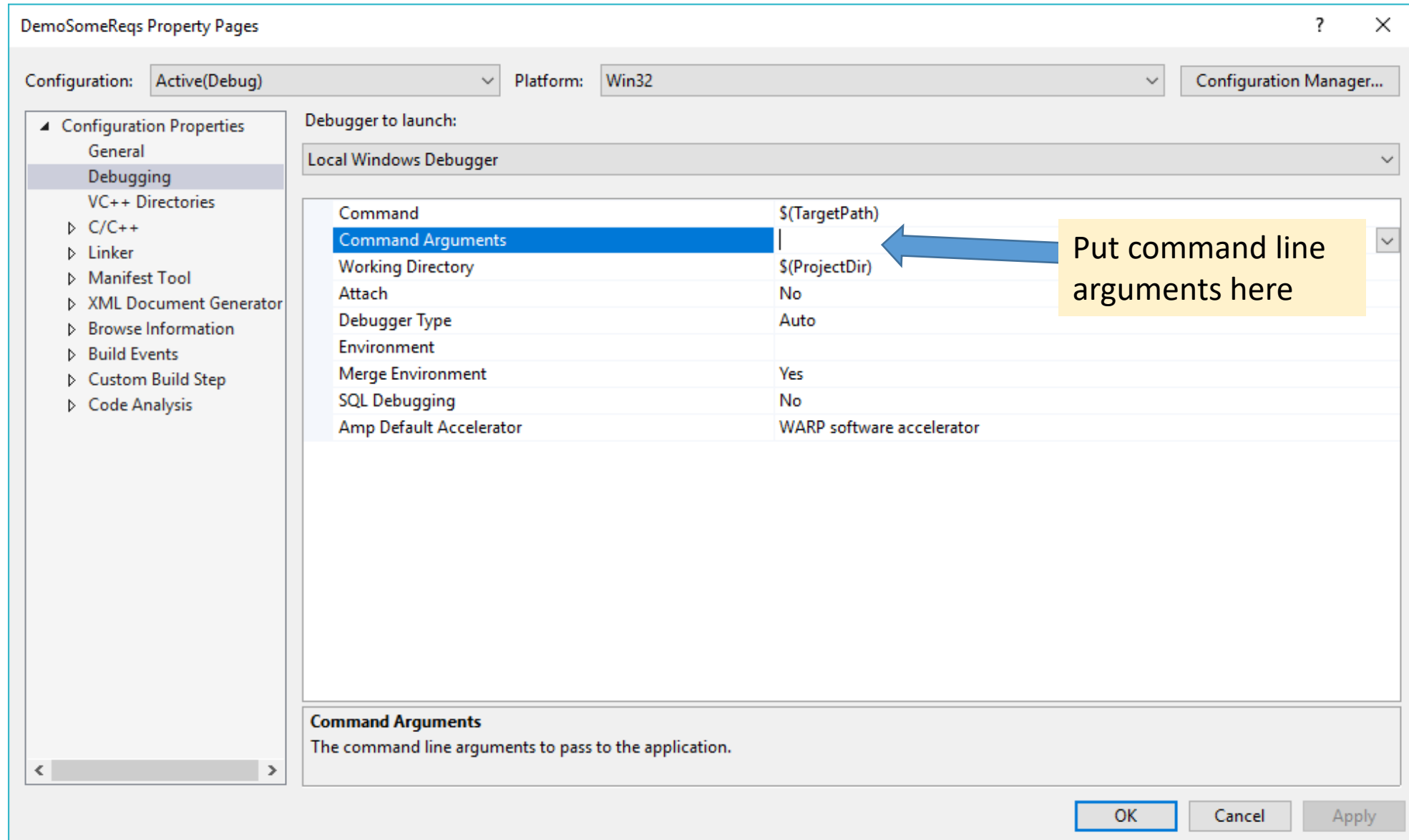
C++ Project > Properties > Configuration Type



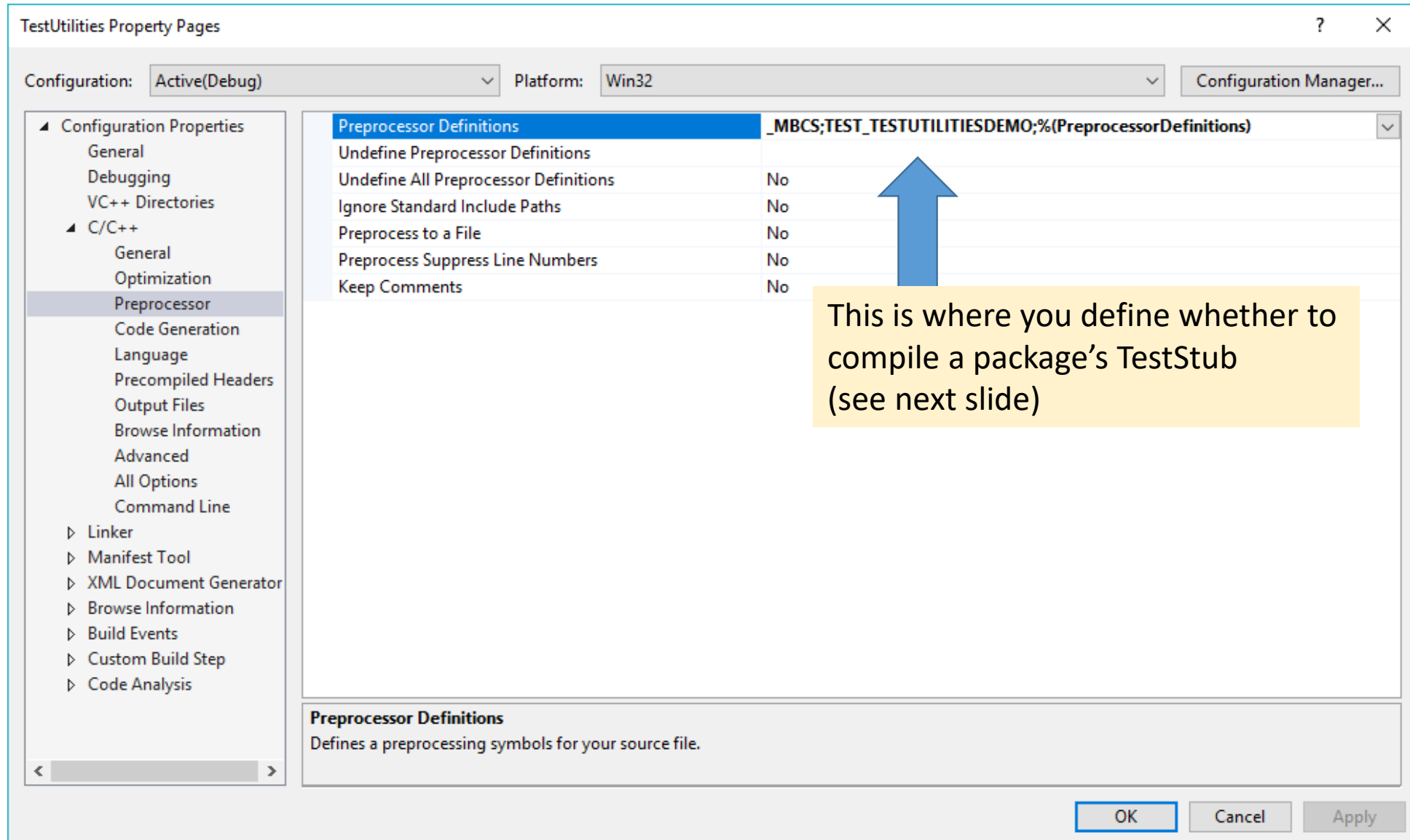
C++ Project > Properties > Configuration Type



C++ Project > Properties > Debugging



C++ Project > Properties > C/C++



The screenshot shows the 'TestUtilities Property Pages' dialog box with the 'C/C++' section expanded to the 'Preprocessor' tab. The 'Preprocessor Definitions' field contains the text: `_MBCS;TEST_TESTUTILITIESDEMO;%(PreprocessorDefinitions)`. A blue arrow points from a yellow callout box to this field. The callout box contains the text: 'This is where you define whether to compile a package's TestStub (see next slide)'. The dialog box also shows the 'Configuration' set to 'Active(Debug)' and 'Platform' set to 'Win32'. The 'Preprocessor Definitions' table is as follows:

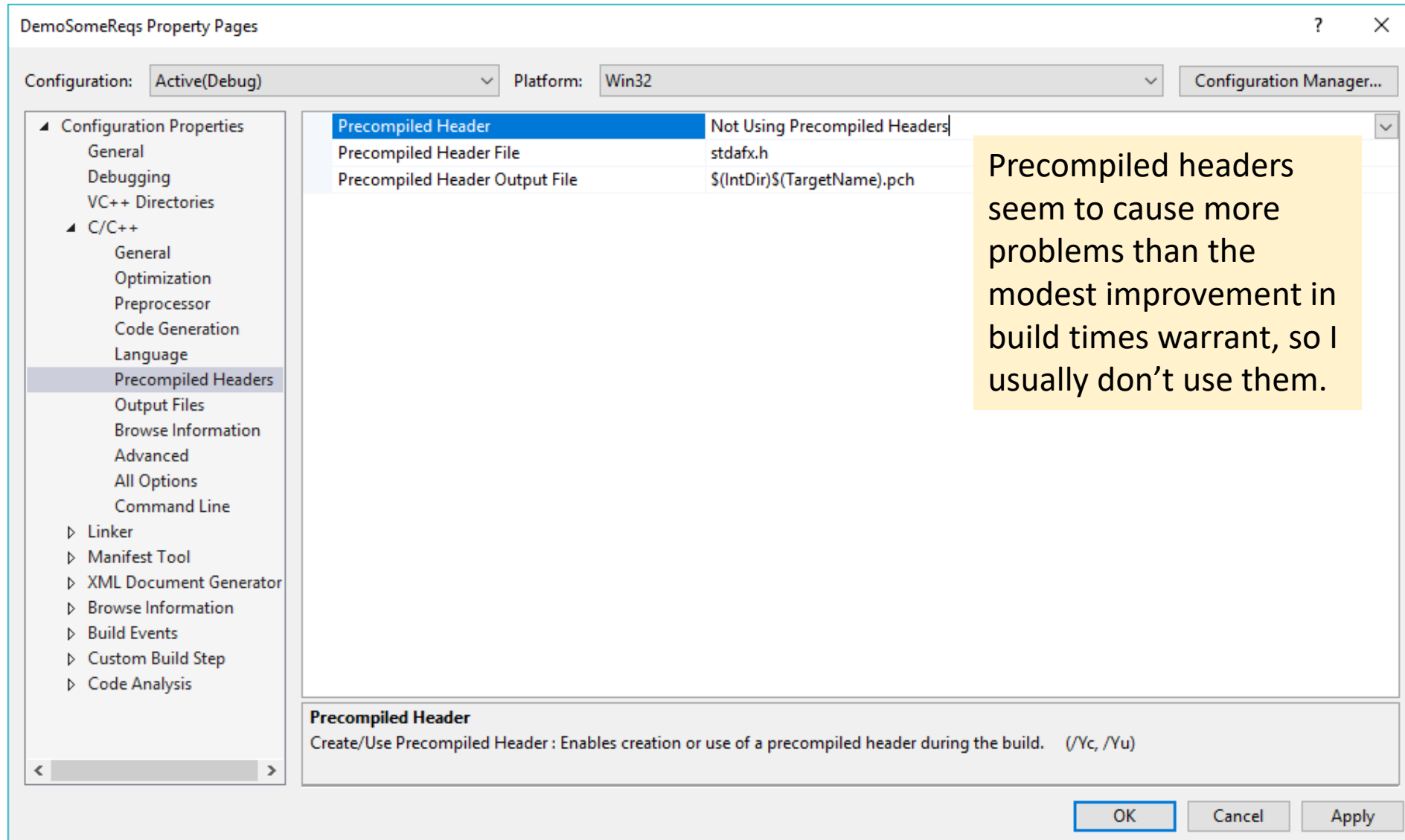
Property	Value
Preprocessor Definitions	<code>_MBCS;TEST_TESTUTILITIESDEMO;%(PreprocessorDefinitions)</code>
Undefine Preprocessor Definitions	
Undefine All Preprocessor Definitions	No
Ignore Standard Include Paths	No
Preprocess to a File	No
Preprocess Suppress Line Numbers	No
Keep Comments	No

At the bottom of the dialog, there is a description for 'Preprocessor Definitions': 'Defines a preprocessing symbols for your source file.' The 'OK', 'Cancel', and 'Apply' buttons are visible at the bottom right.

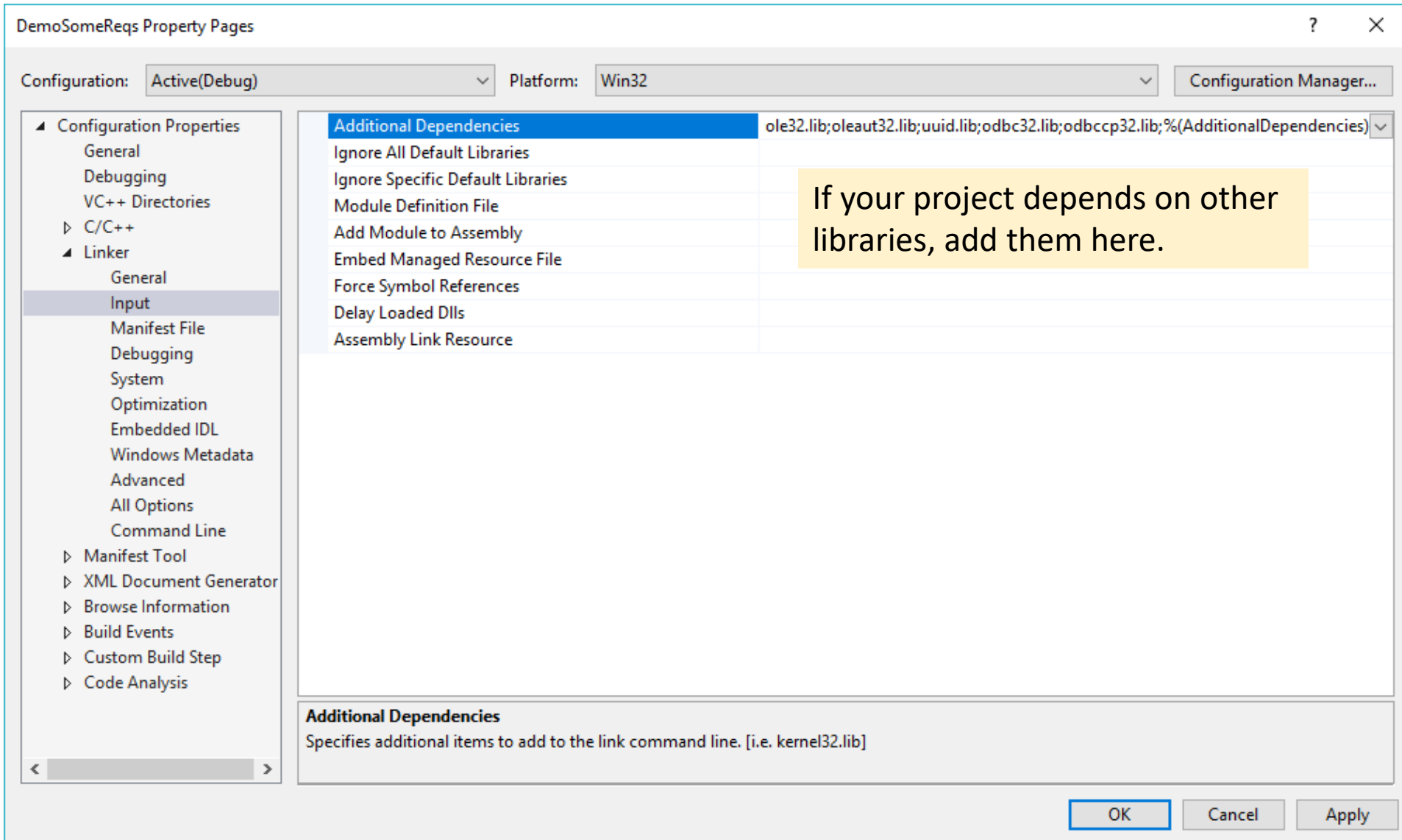
Option to compile package's main function

```
9
10     #include <cctype>
11     #include <iostream>
12     #include <memory>
13     #include "TestUtilitiesDemo.h"
14
15     std::ostream& out = std::cout;
16
17     #ifdef TEST_TESTUTILITIESDEMO
18     |
19     # namespace Utilities
20     | {
21     # ///////////////////////////////////////////////////////////////////
22     // define demo tests - testing the tester
23     // - destructors in each test simply demonstrate that
```

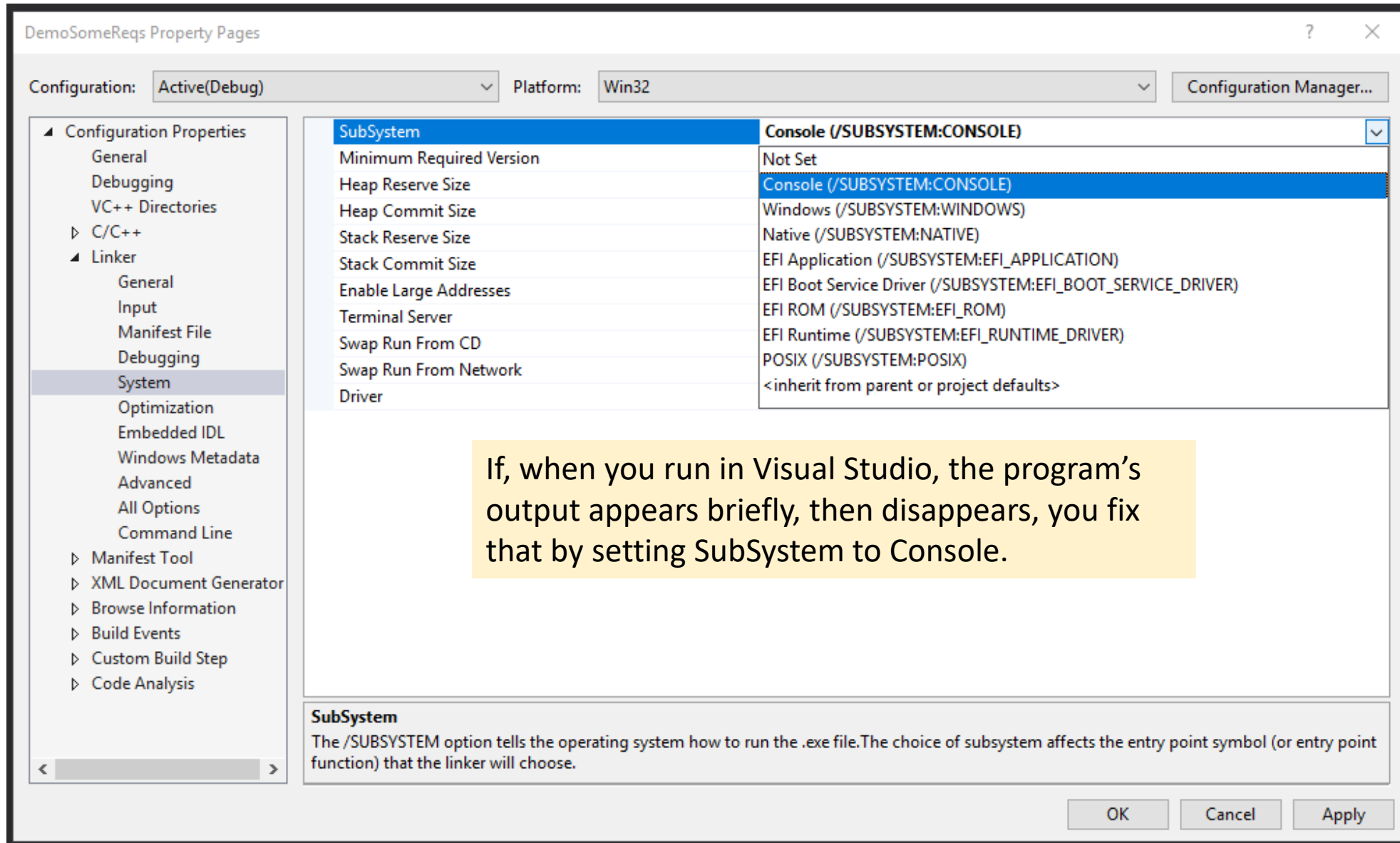
C++ Project > Properties > C/C++



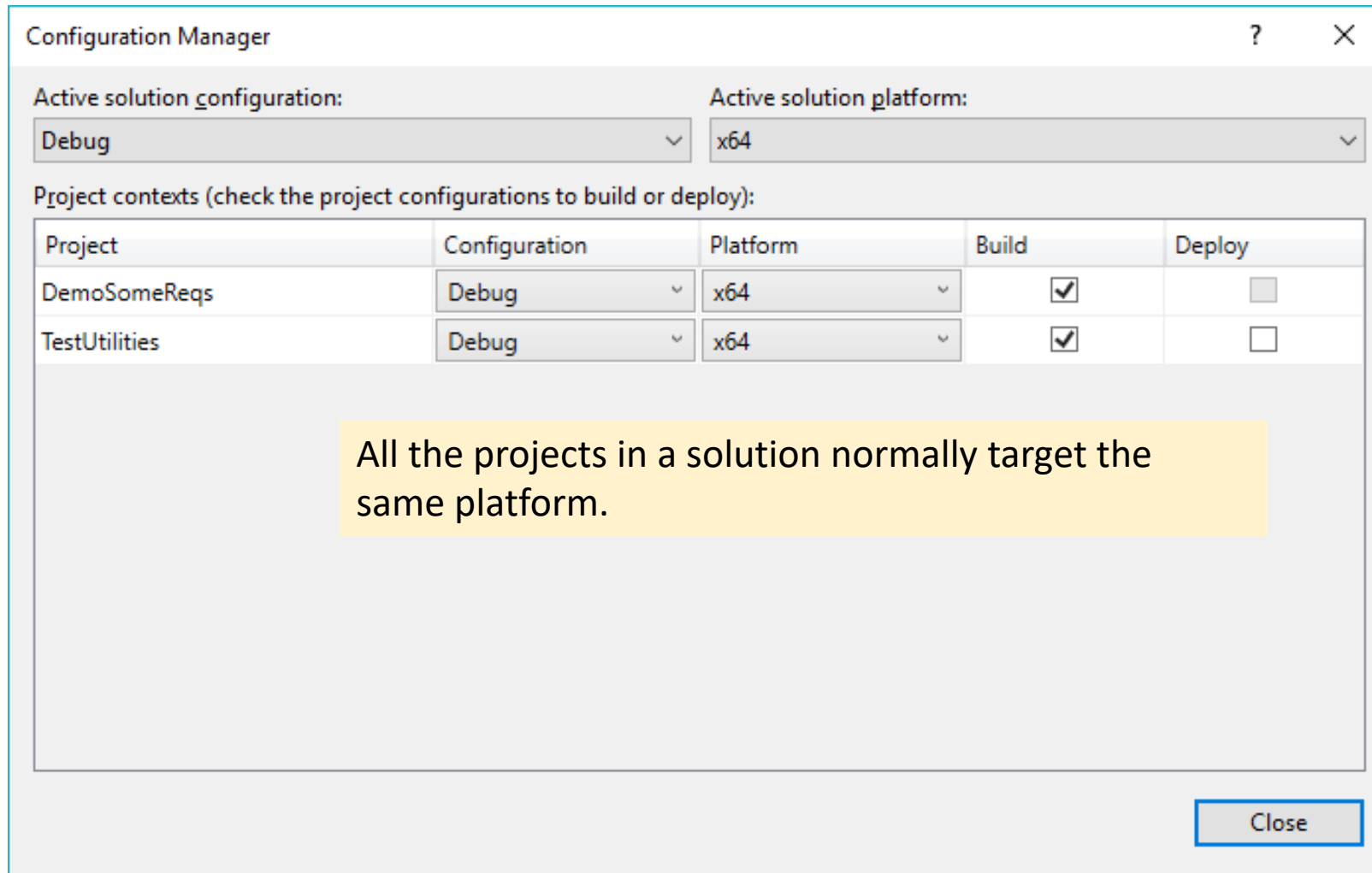
C++ Project > Properties > Linker > Input



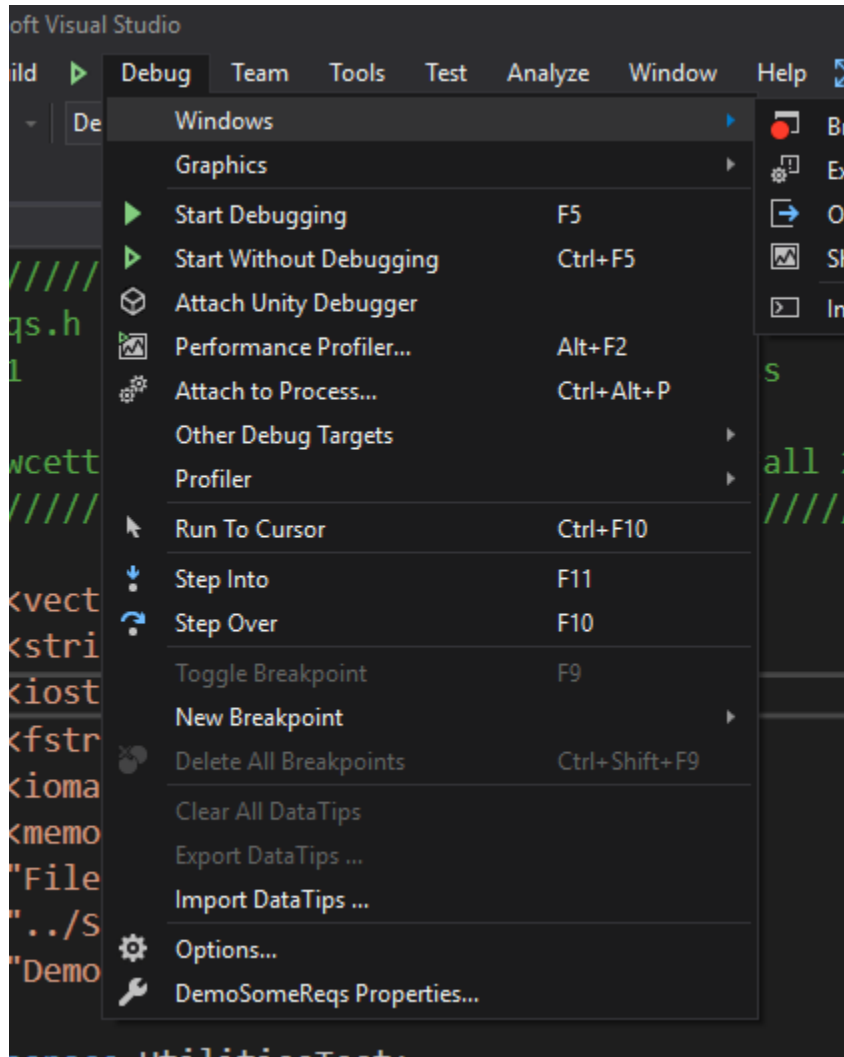
C++ Project > Properties > Linker > System



Build > Configuration Manager



Debug



Debugging Commands

F5: jump to next breakpoint

F10: step next

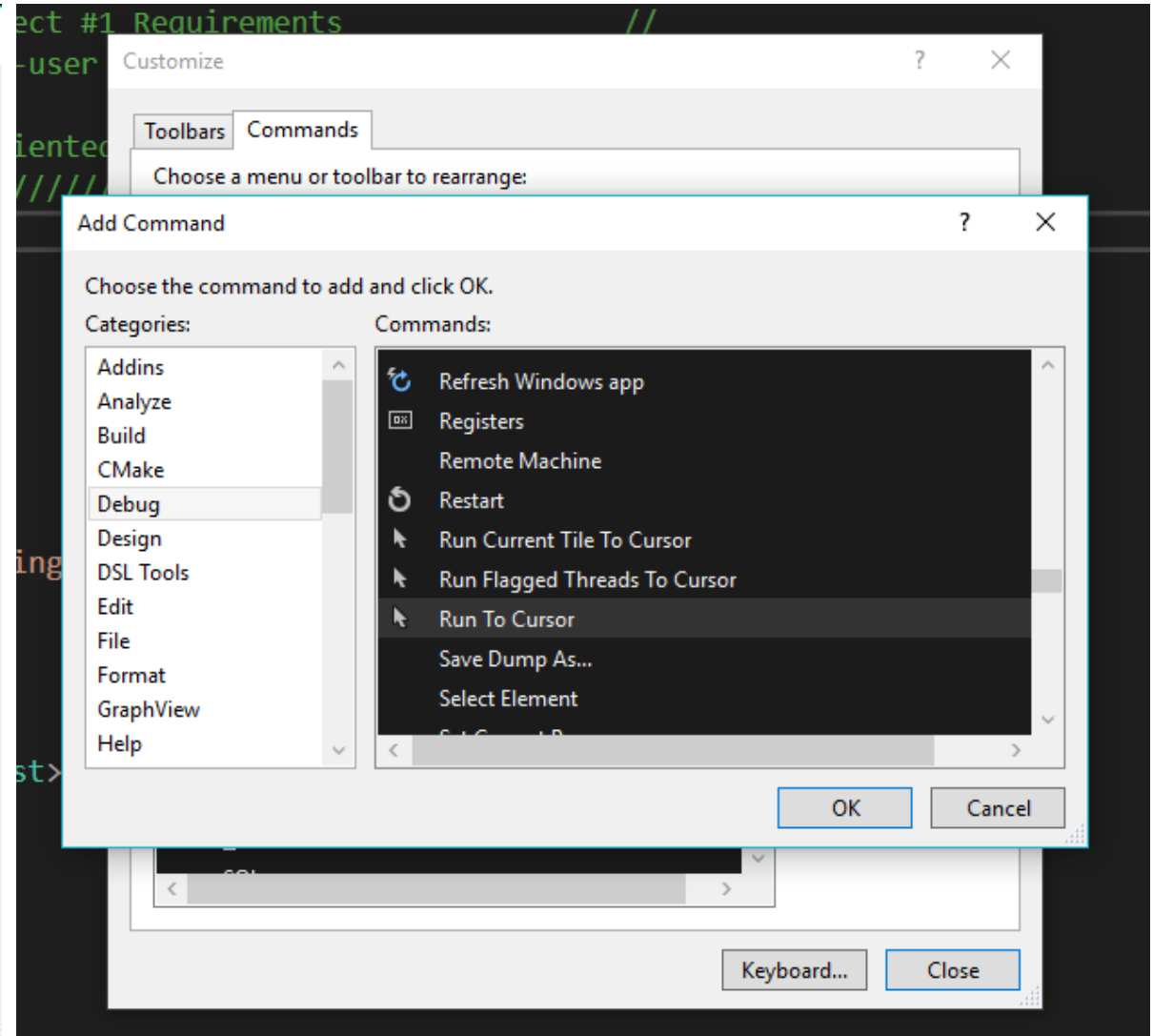
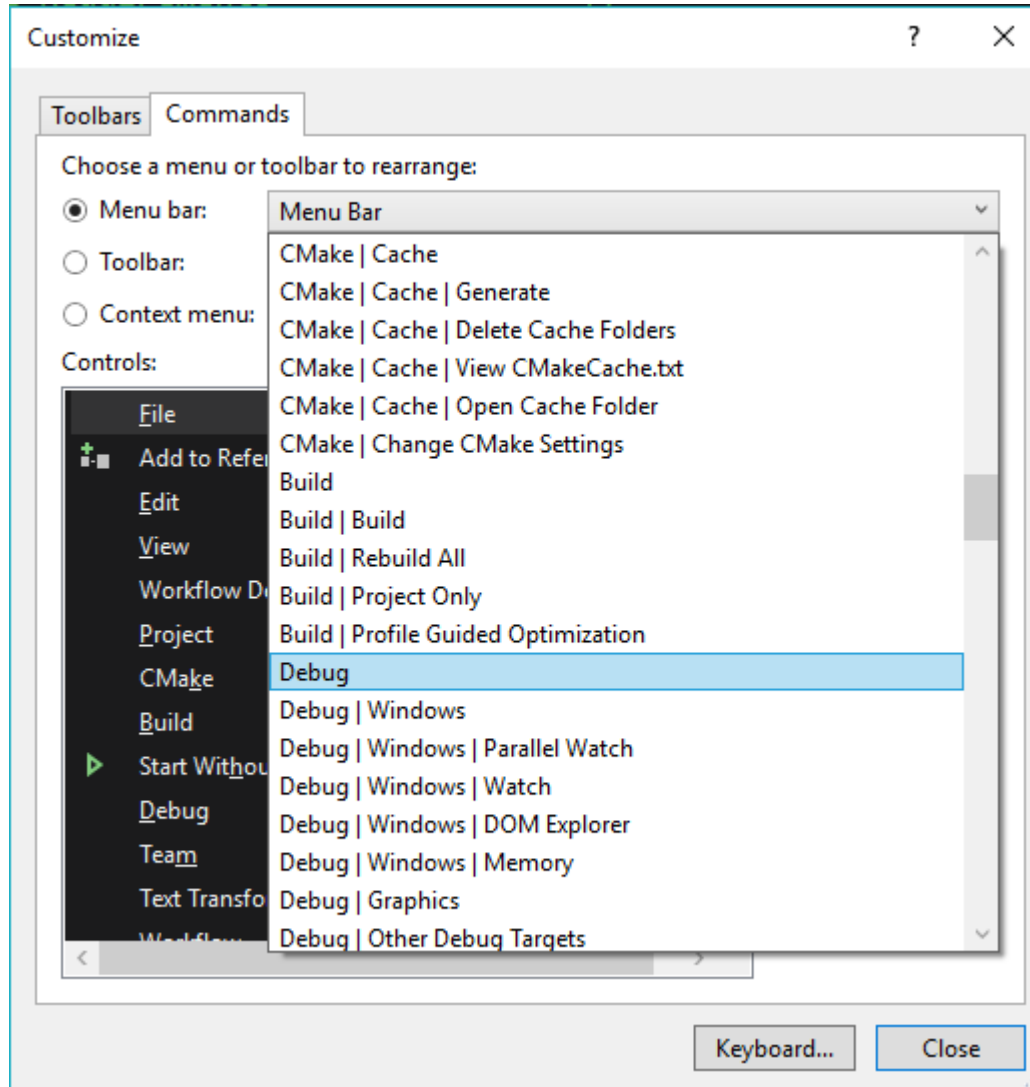
F11: step into function

^F10 : run to cursor

Set or remove breakpoint by clicking on margin

I added Run to Cursor to the Debug menu, as shown in the next slide.

Tools > Customize > Commands



The End