

Chapter 1



Programming with Visual C++ 2008

The .NET Framework

- The .NET Framework is a central concept in Visual C++ 2008
 - As well as in other .NET development environments
- The .NET Framework consists of two elements
 - Common Language Runtime (CLR)
 - .NET Framework class library
- Strictly speaking, the .NET Framework is not part of Visual C++ 2008
 - It is a component of the Windows operating system that makes it easier to build software applications and Web services.

C++ Applications

- You can develop two kinds of applications with Visual C++ 2008:
 - Native C++ program
 - Defined by the ISO/IEC 14882 language standard
 - Also known as ANSI C++
 - C++/CLI program
 - Defined by the ISO/IEC 23271 standard

Common Language Runtime

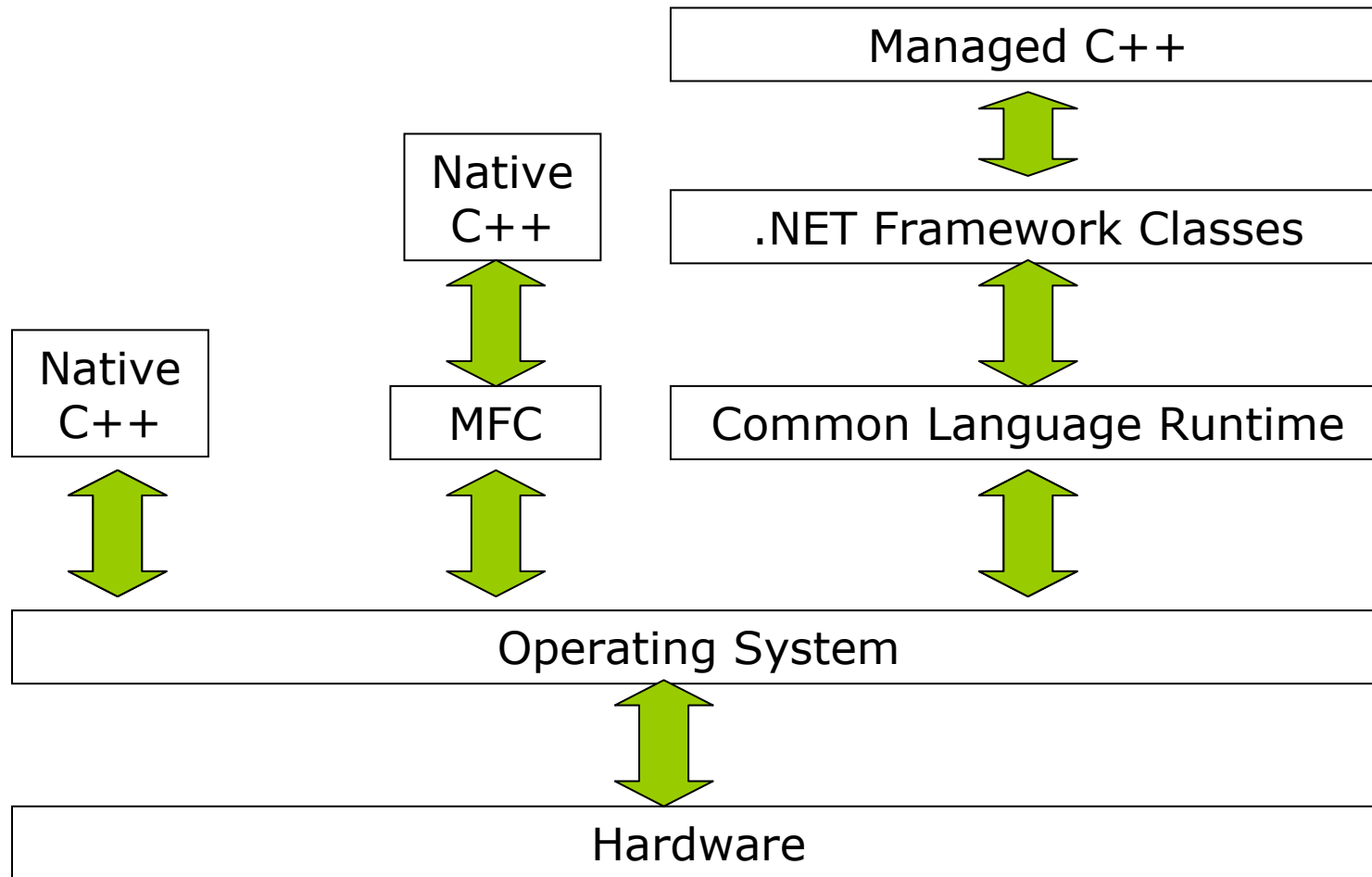
- Common Language Infrastructure (CLI)
 - A standard **intermediate** language for the virtual machine to which the high-level language source code is compiled
 - ECMA-355 (European Computer Manufacturers) standard
 - Equivalent ISO/IEC 23271 standard
- CLI also defines a common set of data types called the **Common Type System (CTS)**
- CLR is Microsoft's implementation of this standard (CLI)

Writing C++ Applications

- ❑ You may write code that executes with the CLR.
 - Managed C++: data and code is managed by the CLR.
 - ❑ Especially, dynamic memory allocation and release
 - For window-based applications, use Windows Forms as the base for the graphical user interface (GUI) provided by the .NET Framework libraries.

- ❑ You may also write code that compiles directly to machine code and thus executes natively.
 - Native C++ / Unmanaged C++
 - Use the Microsoft Foundation Classes (MFC) for programming the GUI.

Figure 1-1



Learning Windows Programming

- ❑ User interacts with GUI.
- ❑ You write code to process these interactions.
- ❑ All the basic code to create the GUI can be generated automatically by Visual C++ 2008.
 - However, you need a comprehensive understanding of C++ to extend and modify the code.
- ❑ You will first learn C++ without getting involved in Windows programming considerations.
 - The tutorial on the C++ language uses examples that are [console programs](#) with simple command-line input and output.

Integrated Development Environment (IDE)

- Editor
 - An interactive environment to create and edit C++ source code.
 - Cut & paste
 - Color pattern
- Compiler
 - Converts your source code into object code
 - Detects and report errors in the compilation process
- Linker
 - Combines various modules generated from the source code files, adds required code modules from libraries, generates an executable file.
- Libraries
 - A collection of pre-written routines
 - Square roots, trigonometric functions
 - Characters and strings comparison

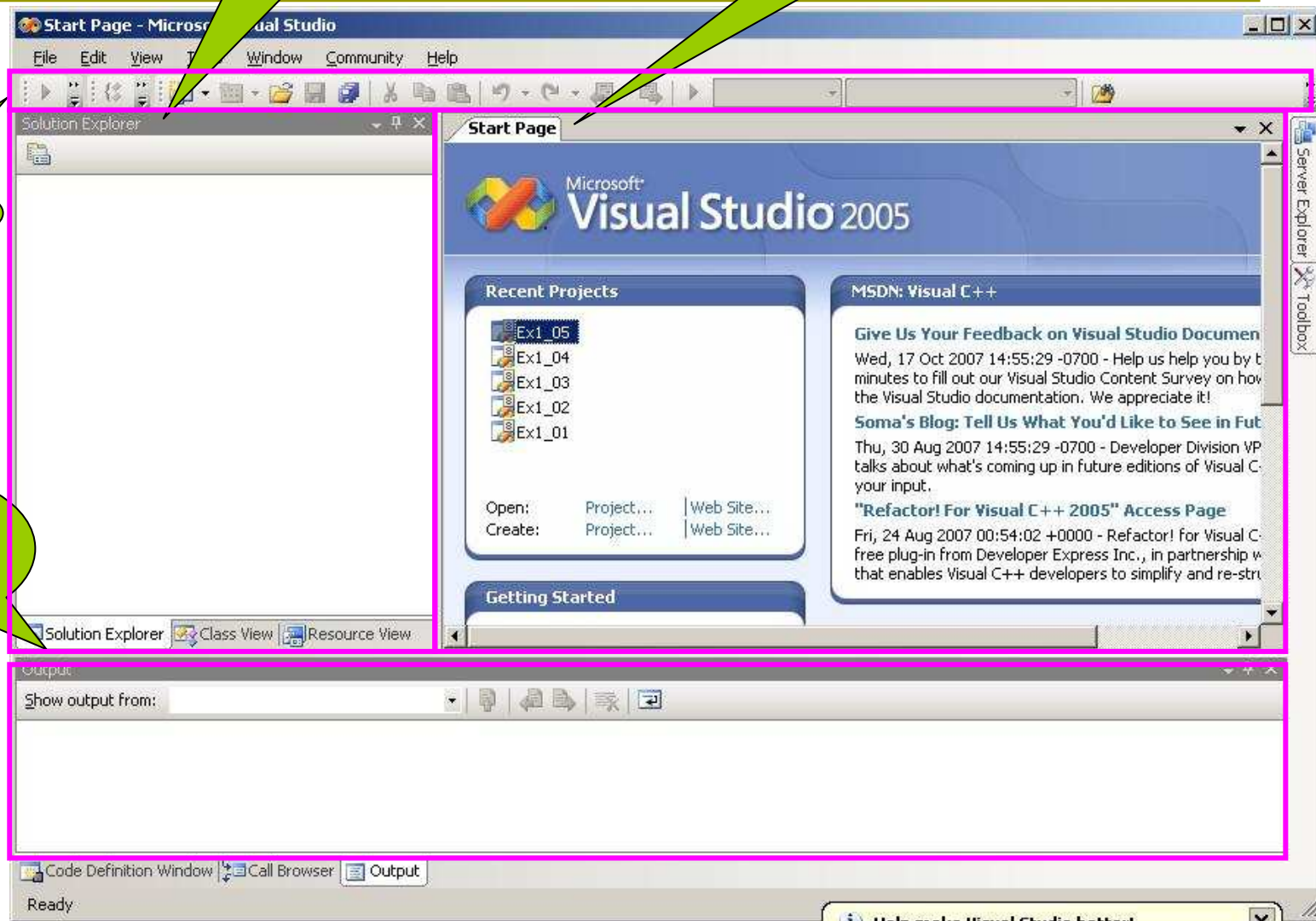
Using the IDE

Solution Explorer window

Editor window

Toolbar

Output window



Toolbar Options

- ❑ Right-clicking in the toolbar area, a pop-up menu will appear, showing a list of toolbars
 - Currently displayed toolbars have check marks alongside

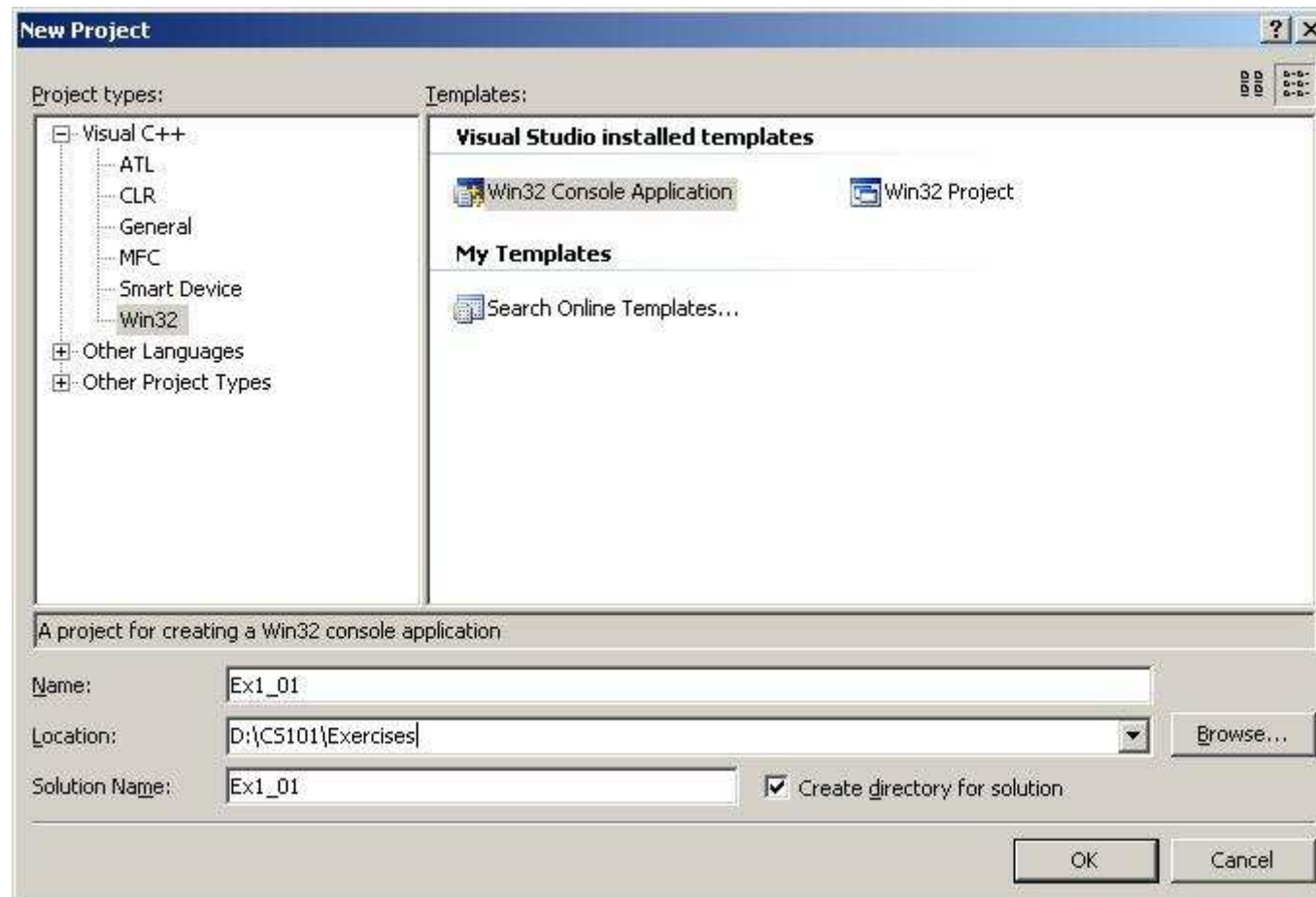


Projects & Solutions

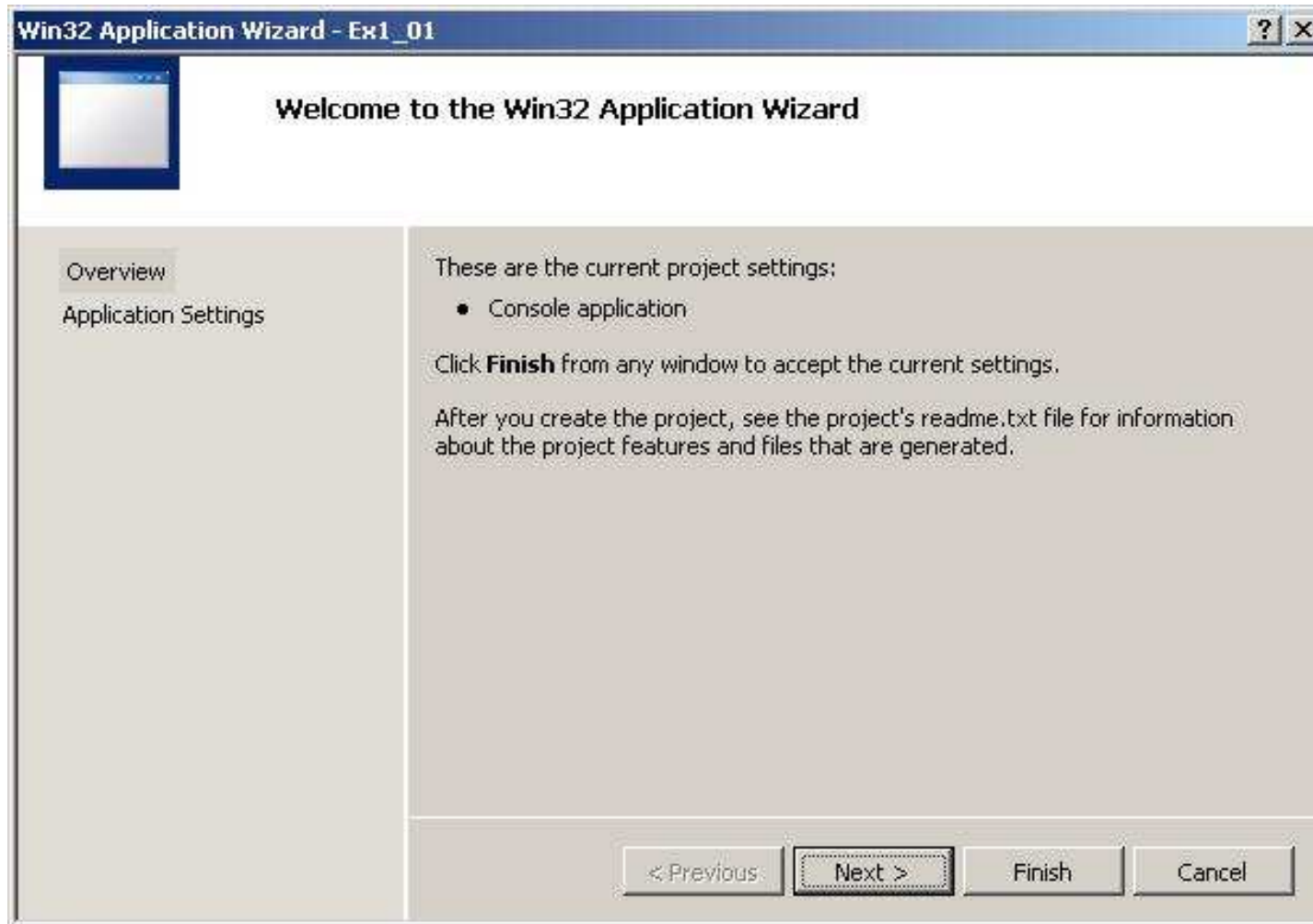
- A **project** is a container for all the things that make up a program
 - It usually consists of one or more source files containing your code
 - Possibly other files containing auxiliary data
 - All the files for a project are stored in **the project folder**
 - Detailed information about the project is stored in an **XML** file with the extension `.vcproj`
- A **solution** is a container for one or more projects that form a solution to a specific information-processing problem.
 - A solution is a folder in which all the information relating to one or more projects is stored.
 - When you create a project, a new solution is created automatically unless you choose to add the project to an existing solution.

Creating a Project for a Win32 Console Application

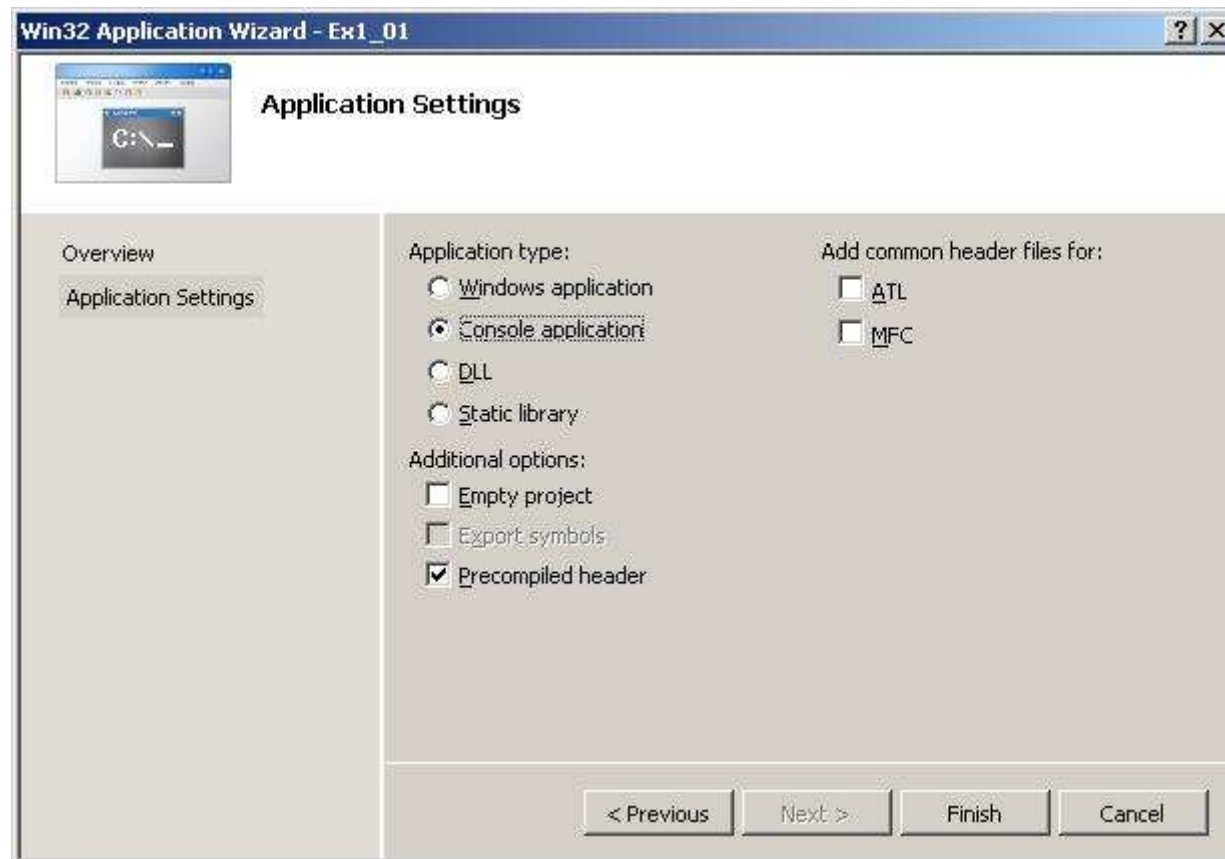
□ File > New > Project



Win32 Application Wizard dialog box

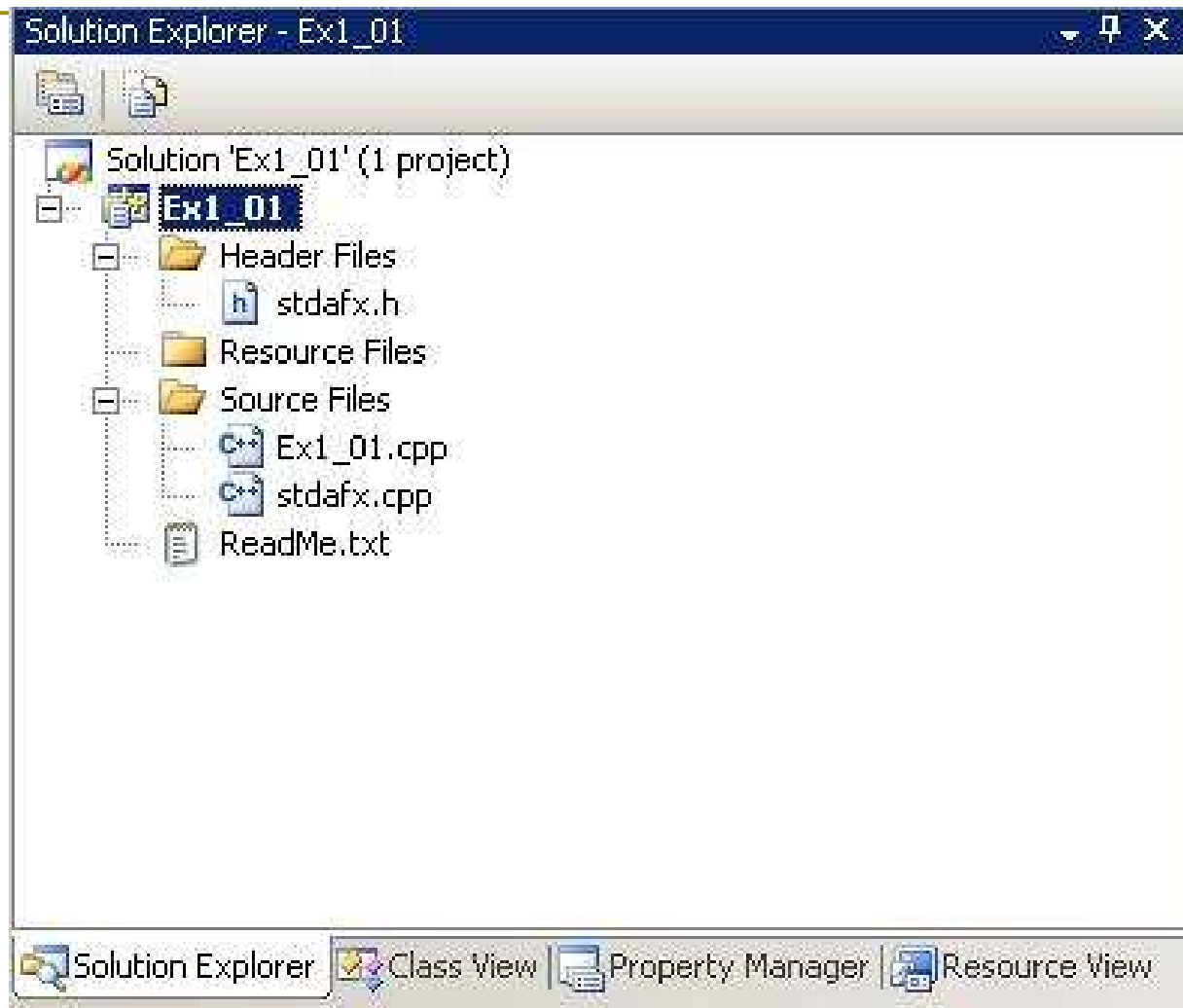


Applications Settings



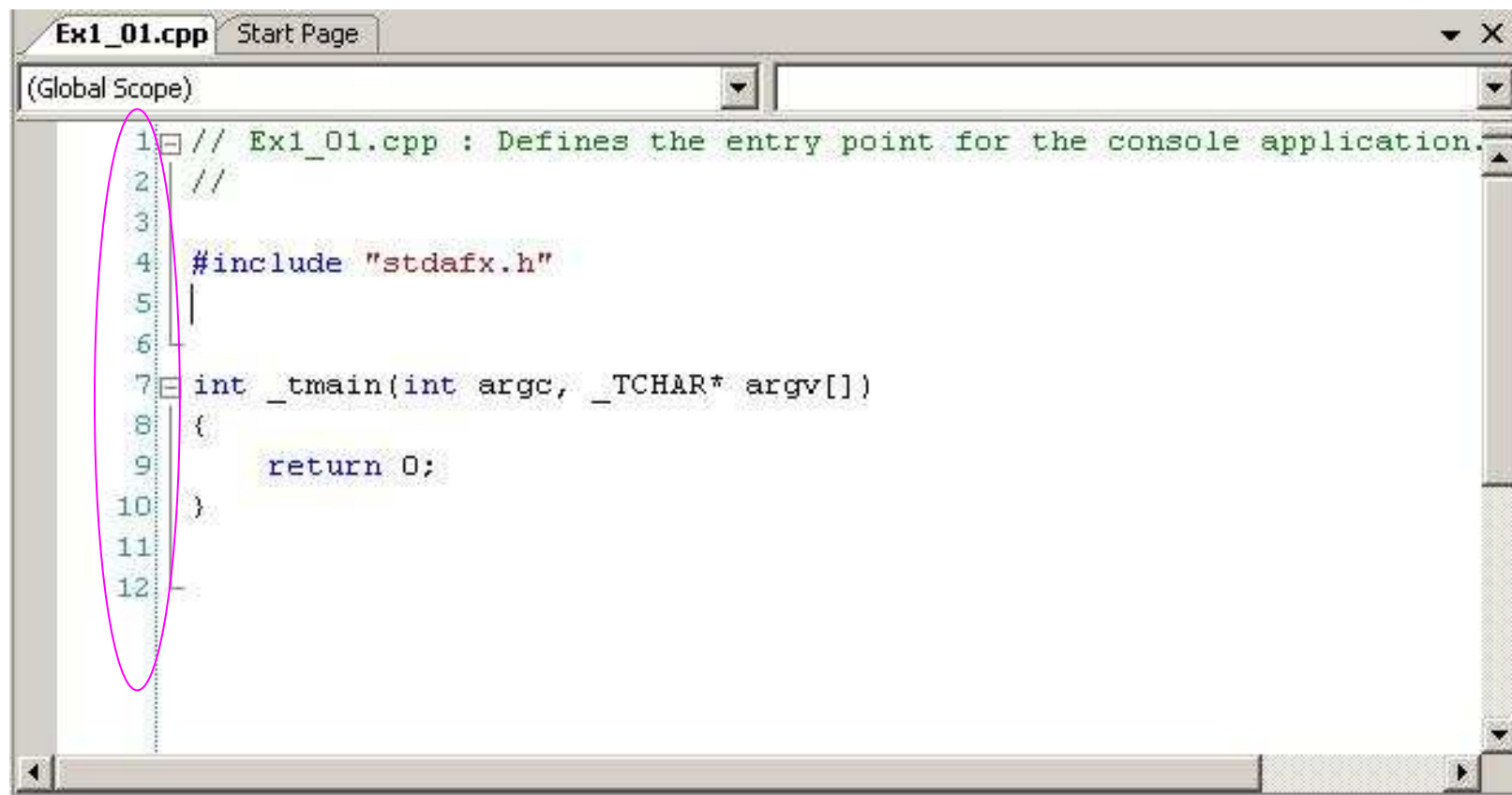
- Just click the `Finish` button to accept all the default settings.

Solution Explorer



Modifying the Source Code

- ❑ Double-click `Ex1_01.cpp` in the Solution Explorer pane.



```
Ex1_01.cpp Start Page
(Global Scope)
1 // Ex1_01.cpp : Defines the entry point for the console application.
2 //
3
4 #include "stdafx.h"
5
6
7 int _tmain(int argc, _TCHAR* argv[])
8 {
9     return 0;
10 }
11
12
```


Insert Two New Lines

```
// Ex1_01.cpp : Defines the entry point //
```

```
#include "stdafx.h"
```

```
#include <iostream>
```

```
int _tmain(int argc, _TCHAR* argv[])
```

```
{
```

```
    std::cout << "Hello world!\n";
```

```
    return 0;
```

```
}
```

Executing the Program

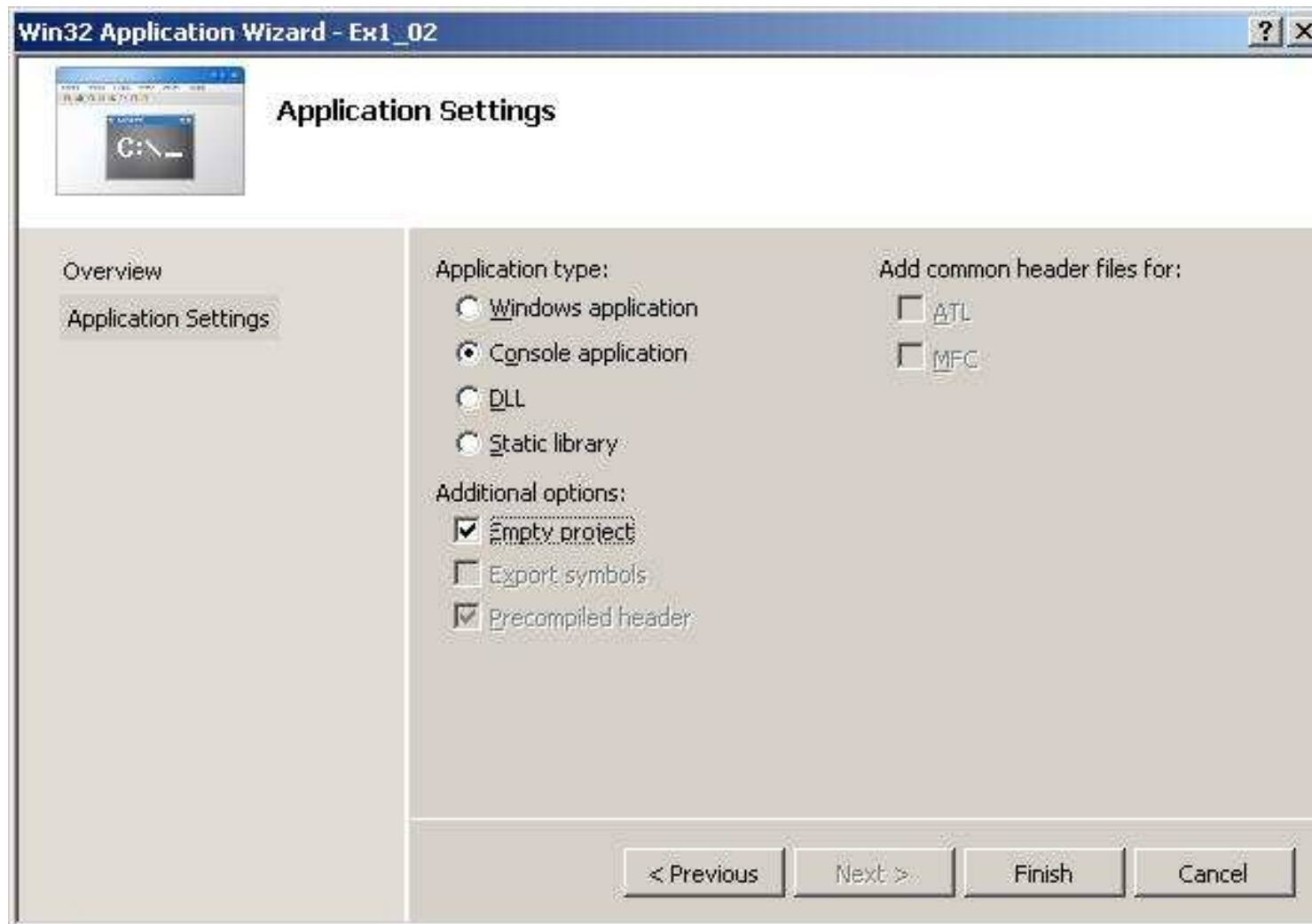
- Compile the solution
 - Select the Build > Build Solution menu item or press F7.

- Execute your program
 - Press Ctrl + F5



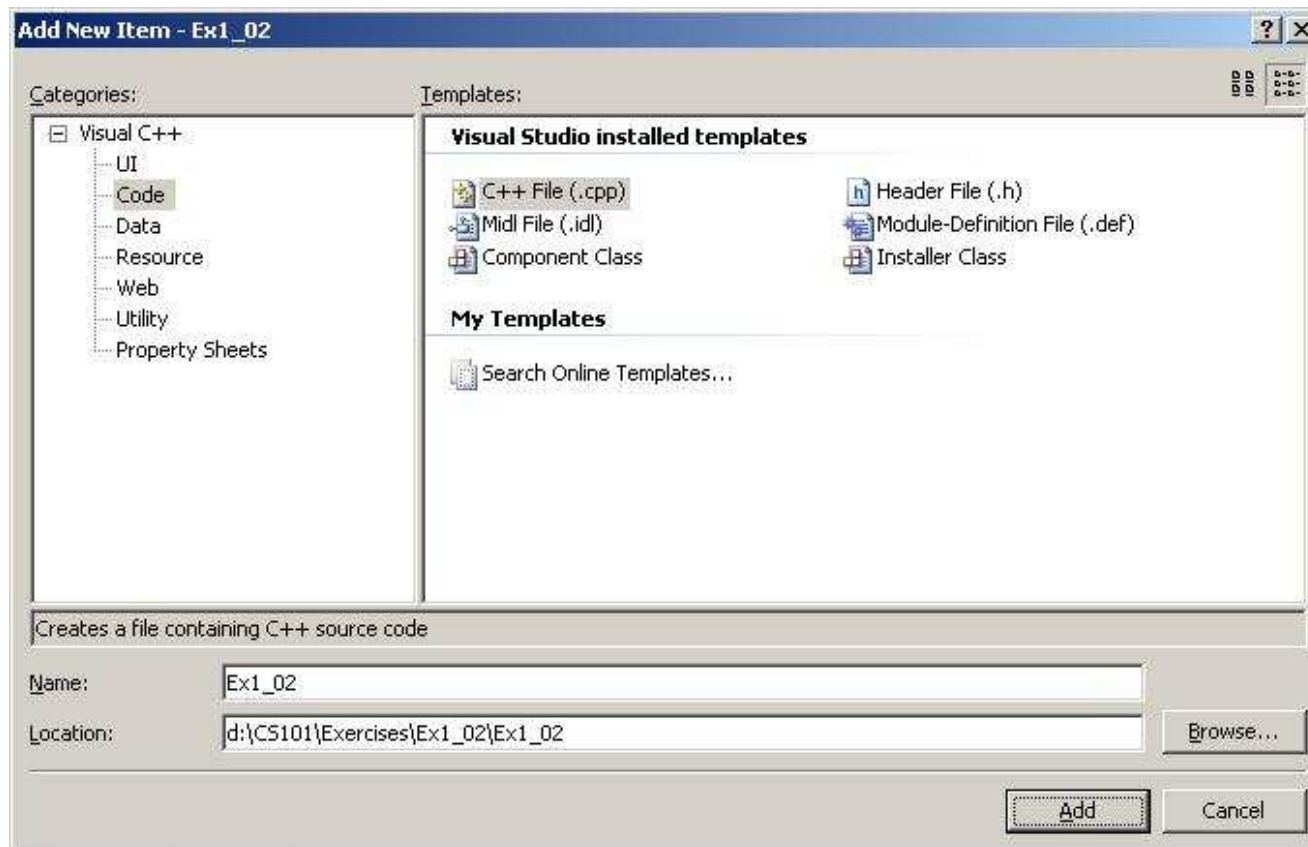
```
cmd C:\WINDOWS\system32\cmd.exe
Hello world!
Press any key to continue . . .
```

Creating an Empty Console Project



Add a New Source File

- Right-click the Solution Explorer pane
- Add > New Item



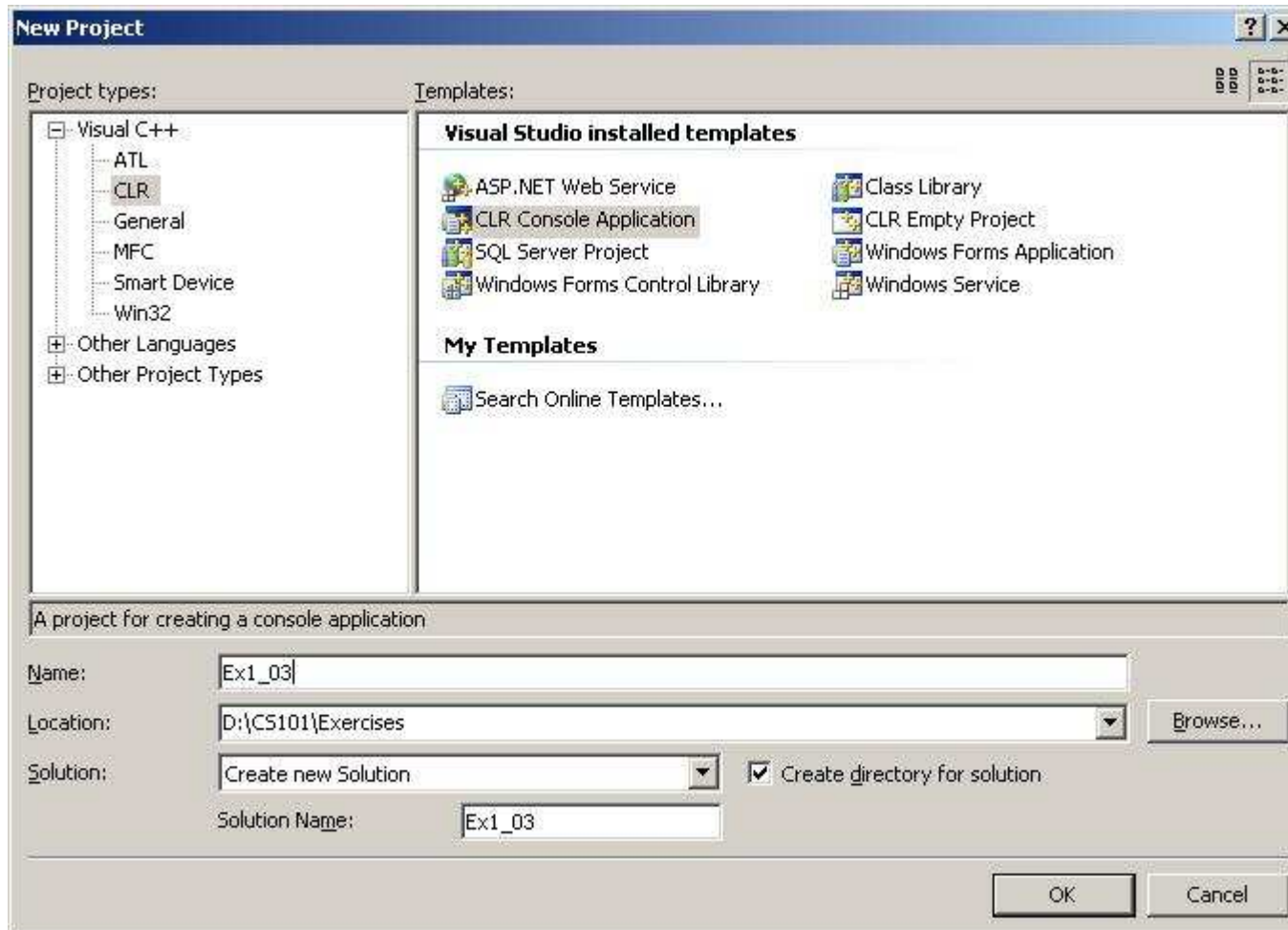
Ex1_02.cpp

- ❑ C++ uses indenting to make programs more readable
- ❑ You can also see the syntax color highlighting in action as you type.

```
// Ex1_02.cpp A simple console program
#include <iostream> // Basic input and output library

int main()
{
    std::cout << "This is a simple program that outputs some text." << std::endl;
    std::cout << "You can output more lines of 中文text" << "\n";
    std::cout << "just by repeating the output statement like this." << std::endl;
    return 0; // Return to the operating system
}
```

Creating a CLR Console Project

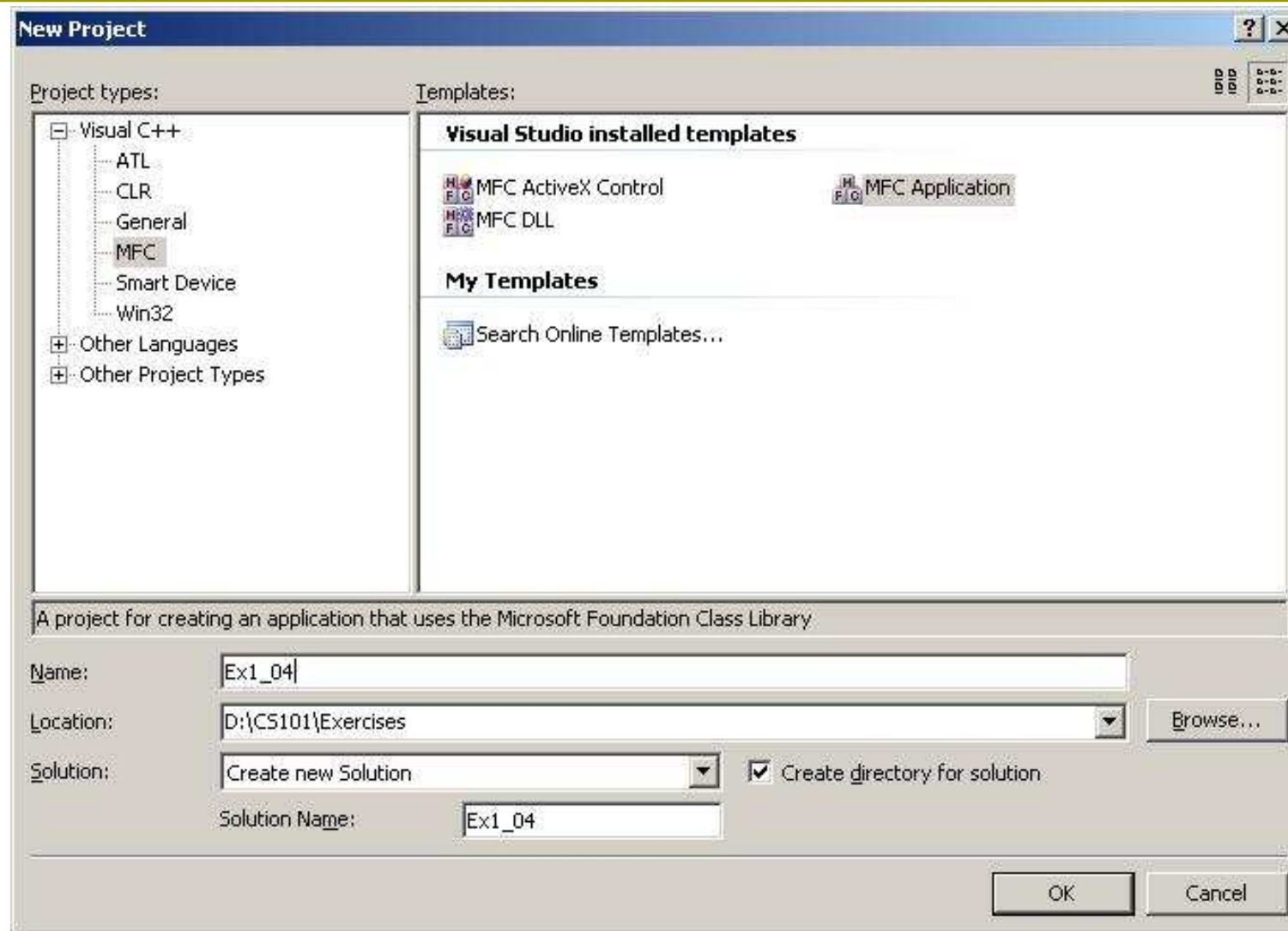


Ex1_03.cpp

```
// Ex1_03.cpp : main project file.  
  
#include "stdafx.h"  
  
using namespace System;  
  
int main(array<System::String ^> ^args)  
{  
    Console::WriteLine(L"Hello World");  
    return 0;  
}
```



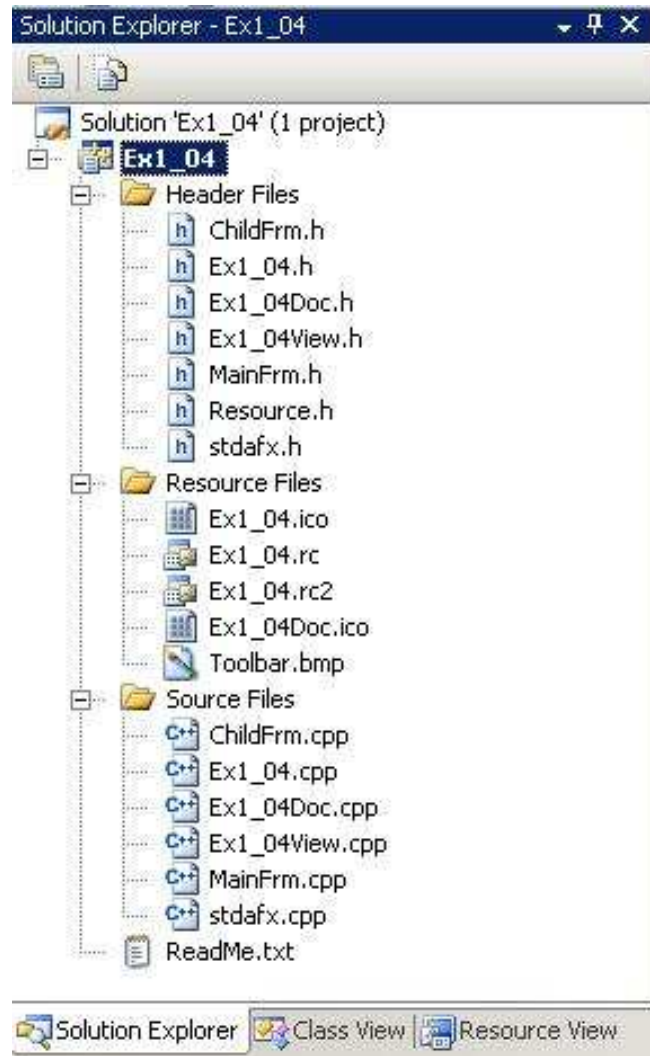
Creating a Windows Application using MFC



MFC Applications Wizard

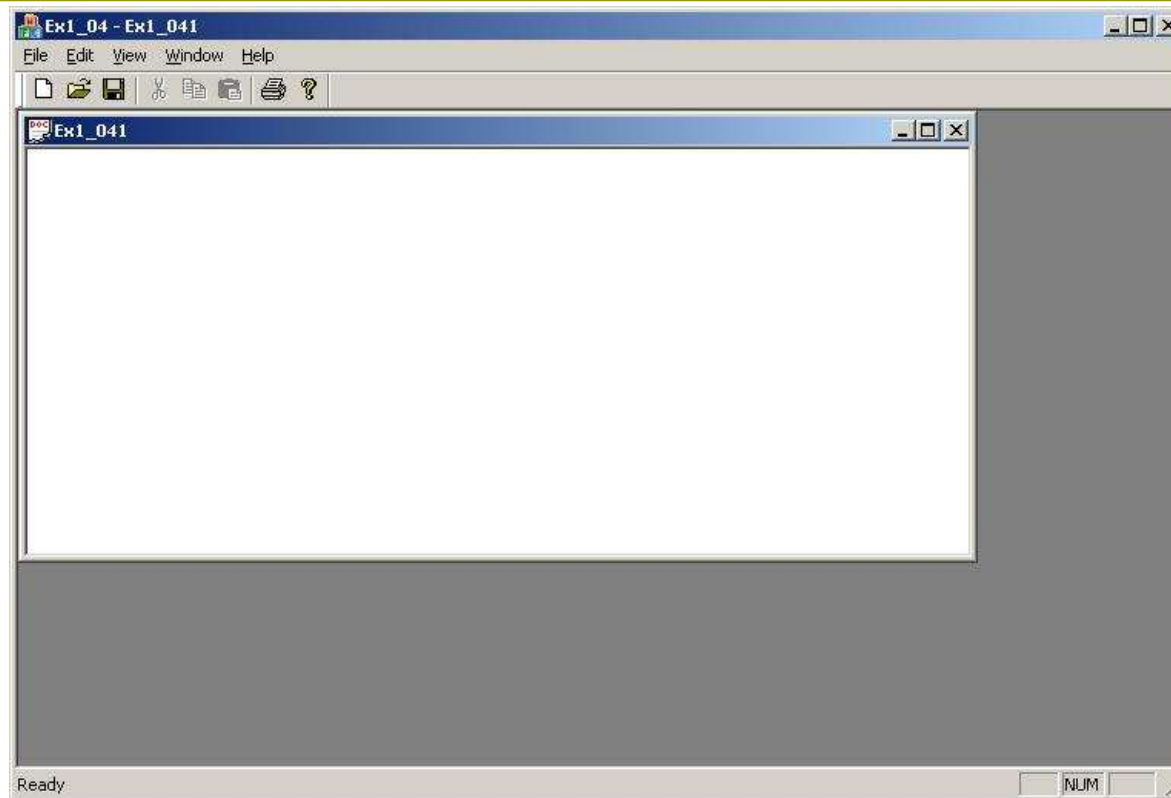


24 Files Are Generated Automatically



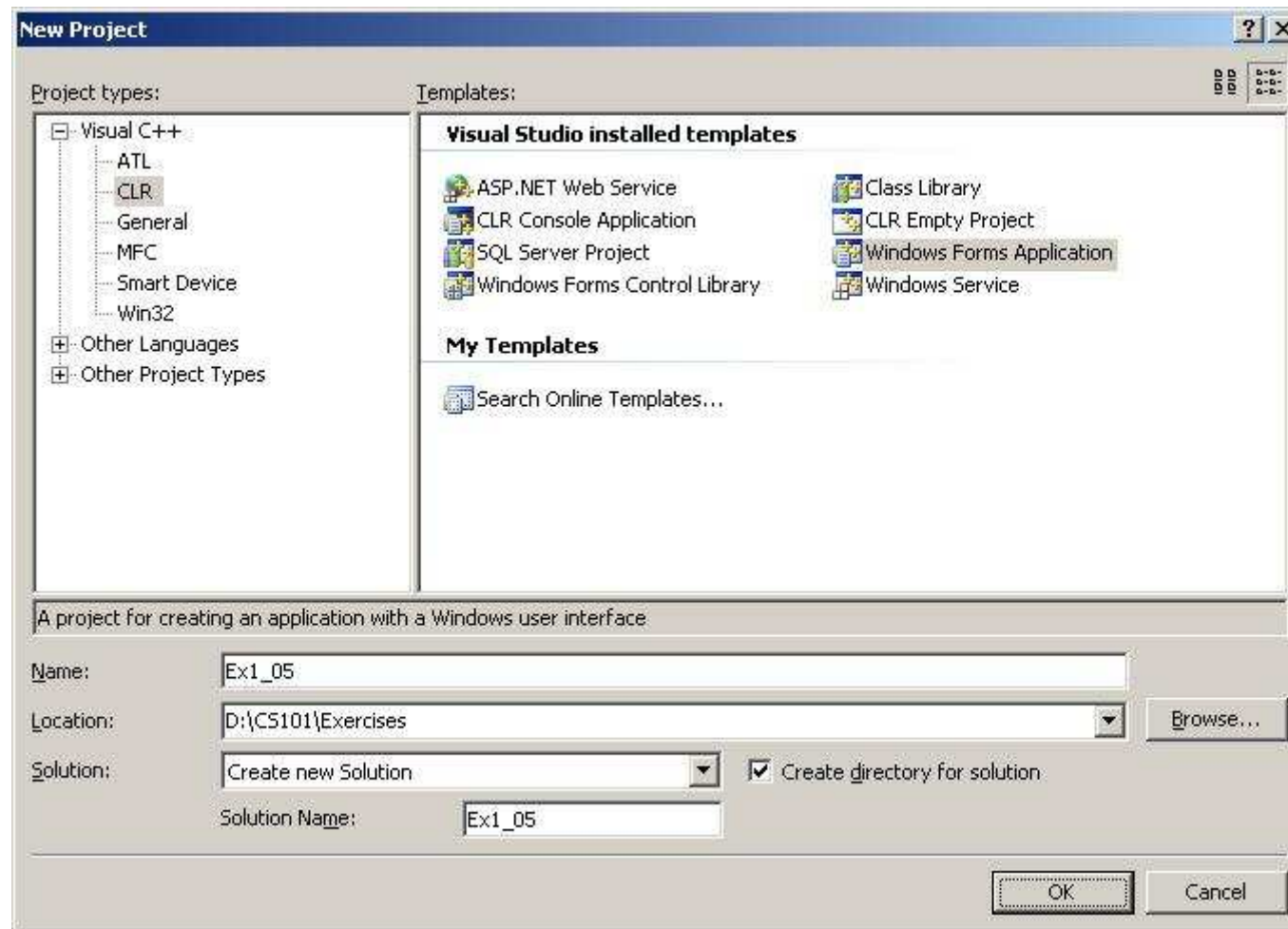
- 3 are in the solution folder
- 17 are in the project folder
- Four are in a subfolder res to the project folder

Building & Executing the MFC Applications

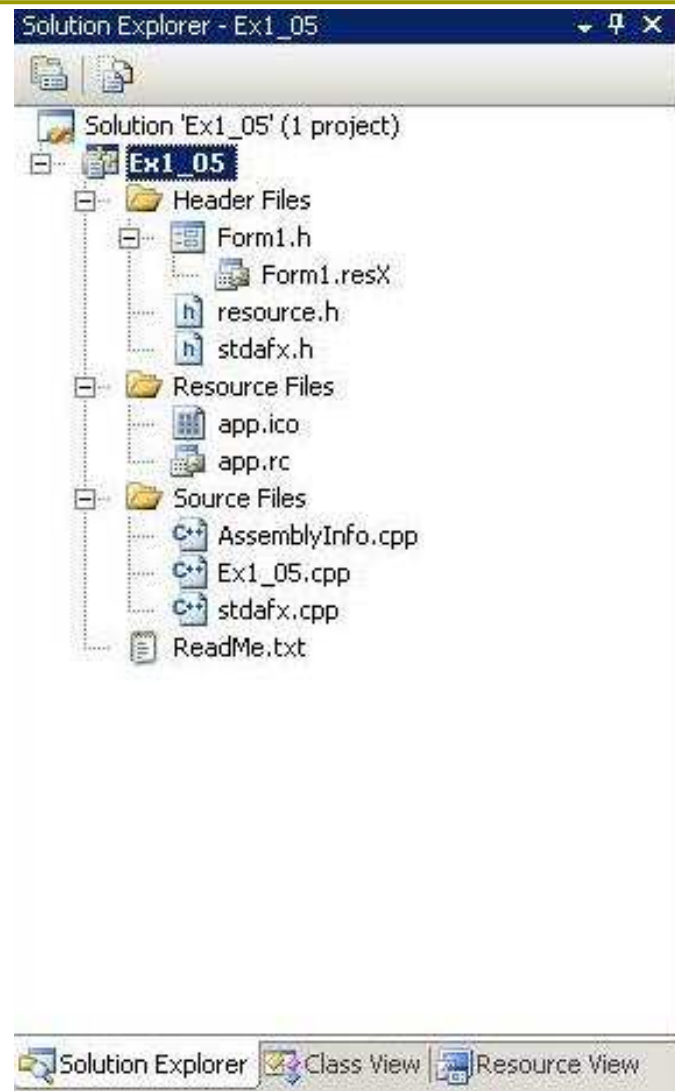


- ❑ Ctrl + F5
- ❑ You can even create further windows by selecting New from the File menu.

Creating a Windows Forms Applications



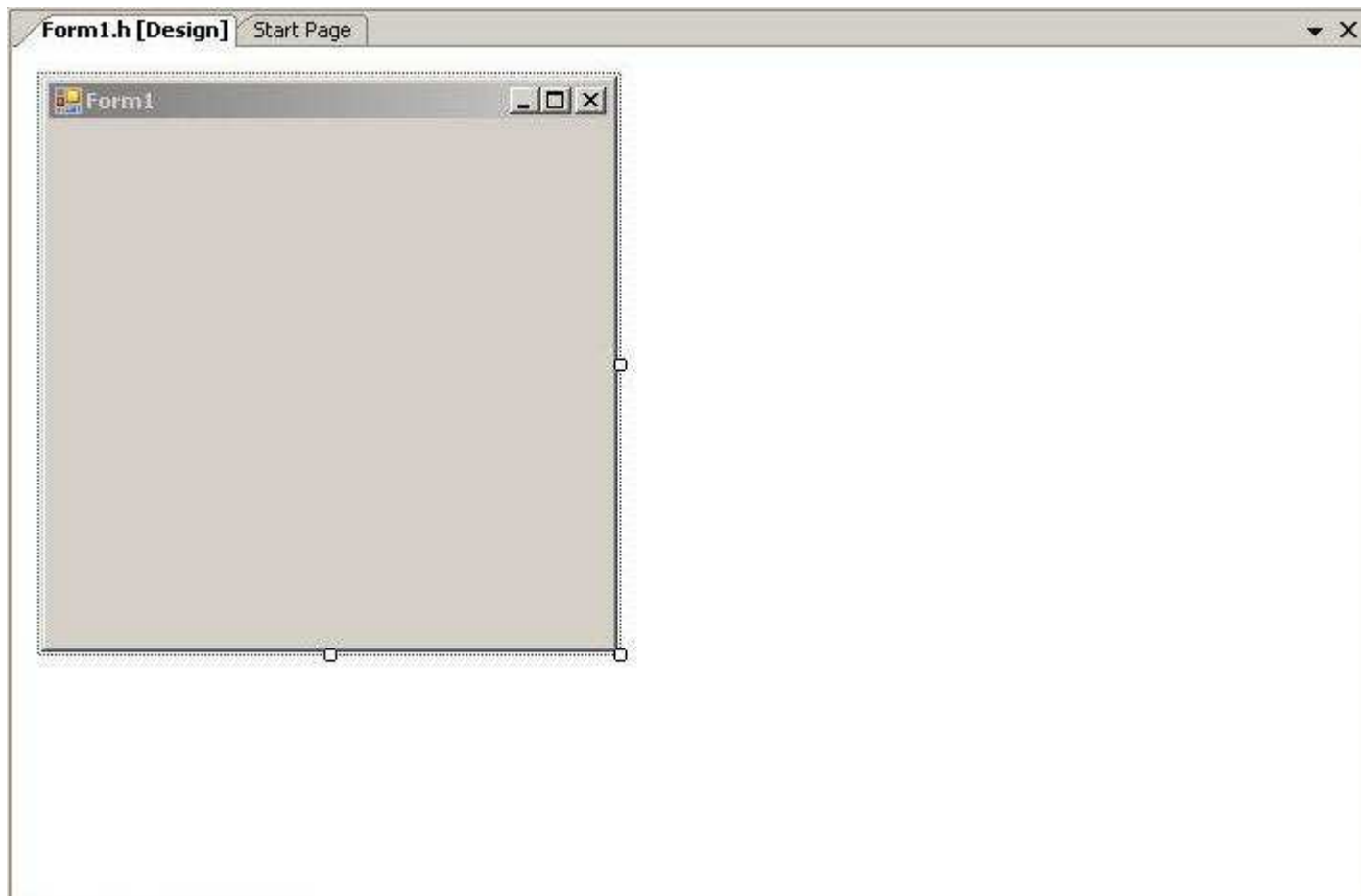
Total of 15 Files



- The Windows Forms application does not include menus or toolbars by default.

Editor Window

- ❑ The Editor window shows an image of the application window rather than code.



Add GUI Components

- View > Toolbox
or Ctrl+Alt+X
- Add two buttons

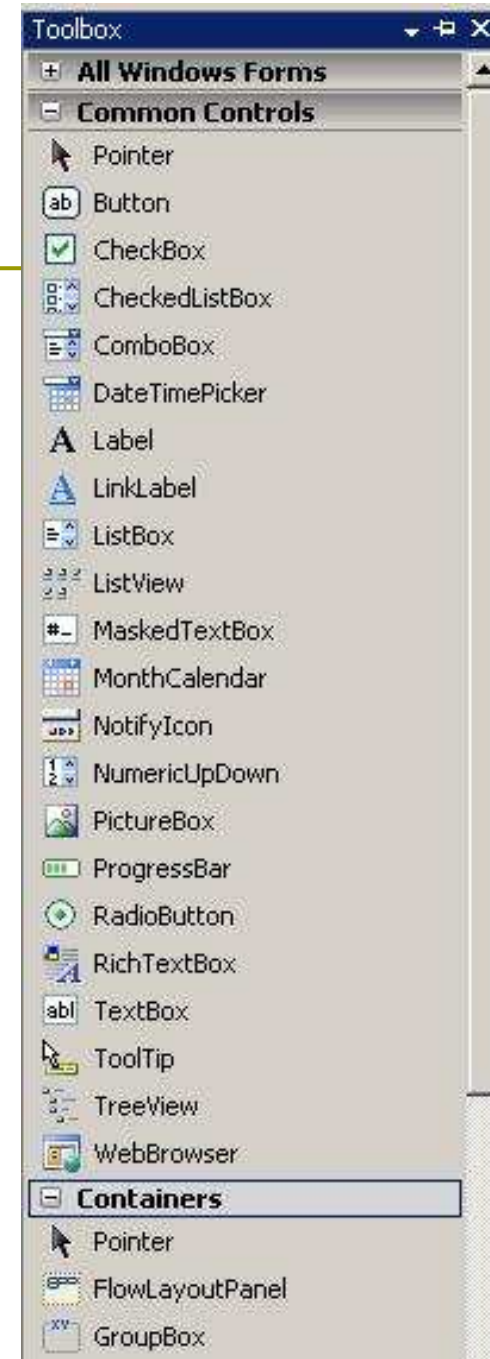


Figure 1-29

- ❑ Creating a Windows application cannot be easier!

