Chapter 1

Programming with Visual C++ 2005

The .NET Framework

- The .NET Framework is a central concept in Visual C++ 2005
  - 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++ 2005
  - 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++ 2005:
  - 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

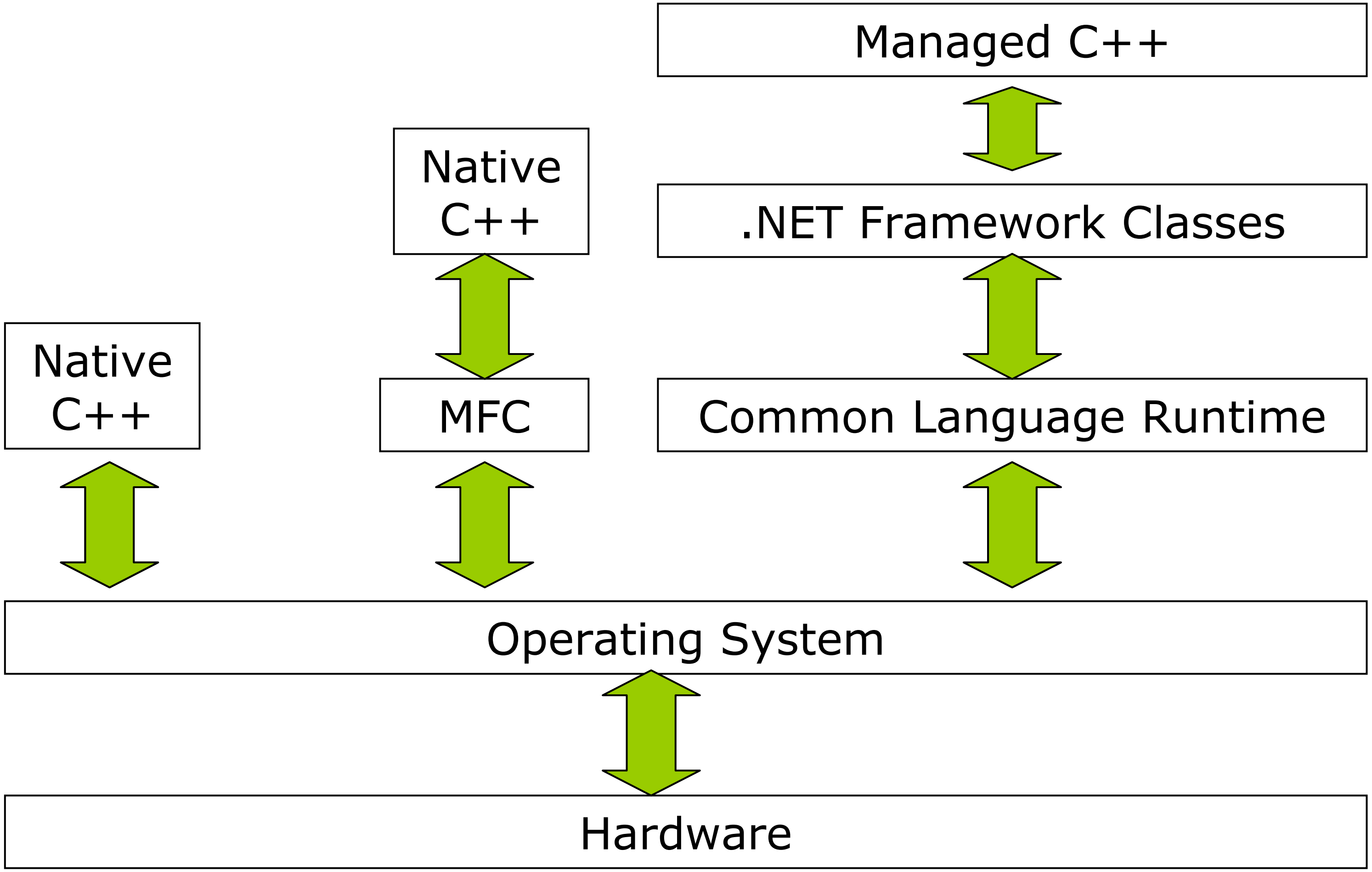
- Common Language Infrastructure (CLI)
  - A standard **intermediate** language for the virtual machine to which the high-level language source code is compiled
  - ECMA-355 (European Computer Manufacturers) standard
  - Equivalent ISO/IEC 23271 standard
- CLI also defines a common set of data types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


Managed C++
Native C++
.NET Framework Classes
Native C++
MFC
Common Language Runtime
Operating System
Hardware

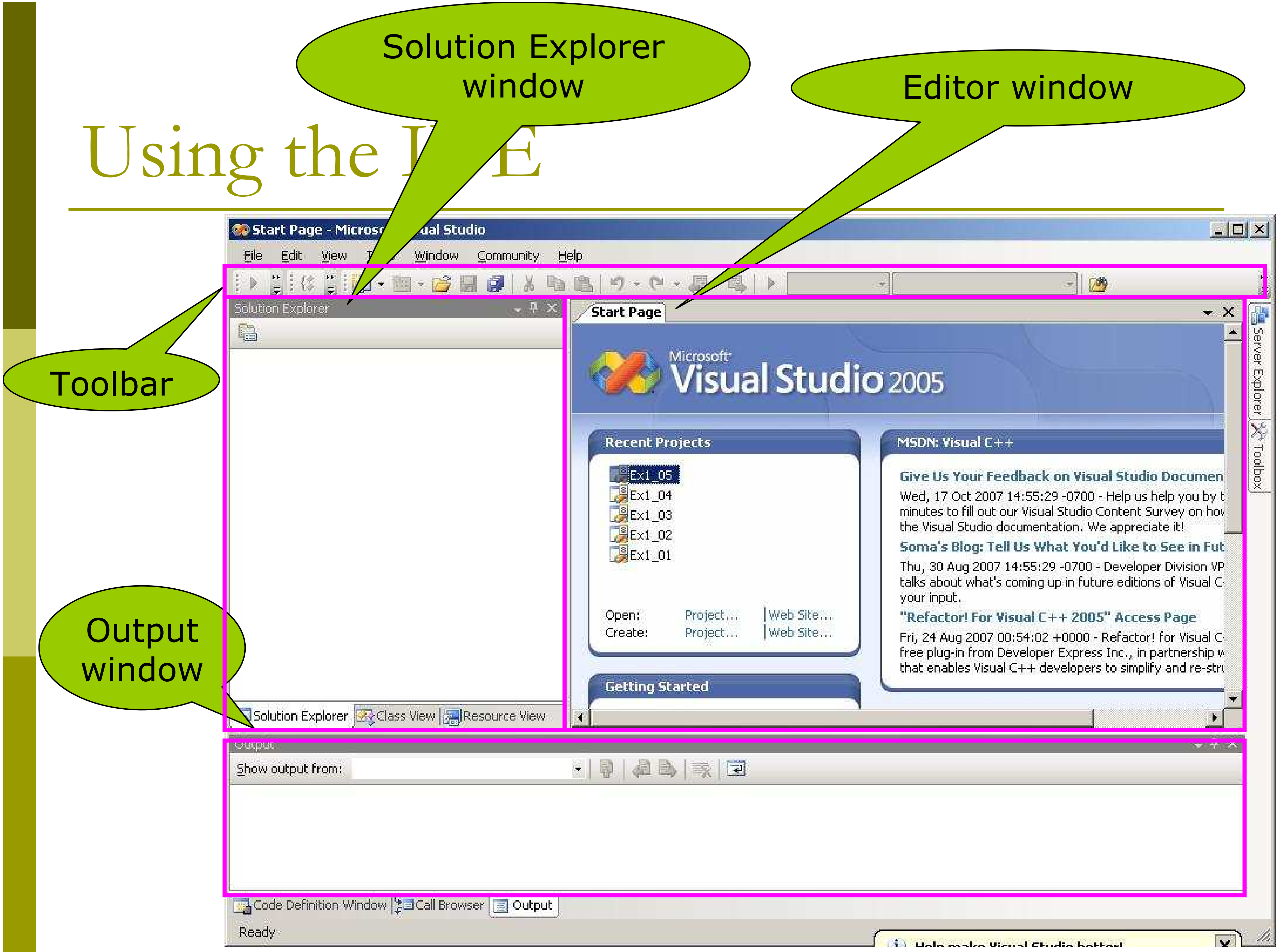
Learning Windows Programming

- 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++ 2005.
  - 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
  - Coverts 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



Give Us Your Feedback on Visual Studio Documen

Wed, 17 Oct 2007 14:55:29 -0700 - Help us help you by t minutes to fill out our Visual Studio Content Survey on how the Visual Studio documentation. We appreciate it!

Soma's Blog: Tell Us What You'd Like to See in Fut

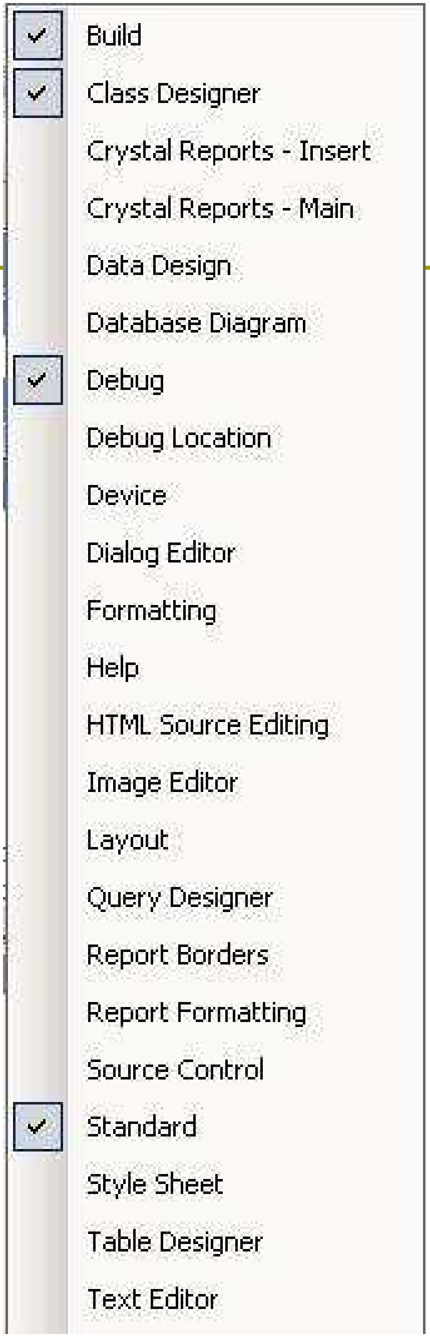
Thu, 30 Aug 2007 14:55:29 -0700 - Developer Division VP talks about what's coming up in future editions of Visual C your input.

"Refactor! For Visual C++ 2005" Access Page

Fri, 24 Aug 2007 00:54:02 +0000 - Refactor! for Visual C free plug-in from Developer Express Inc., in partnership w that enables Visual C++ developers to simplify and re-stru

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



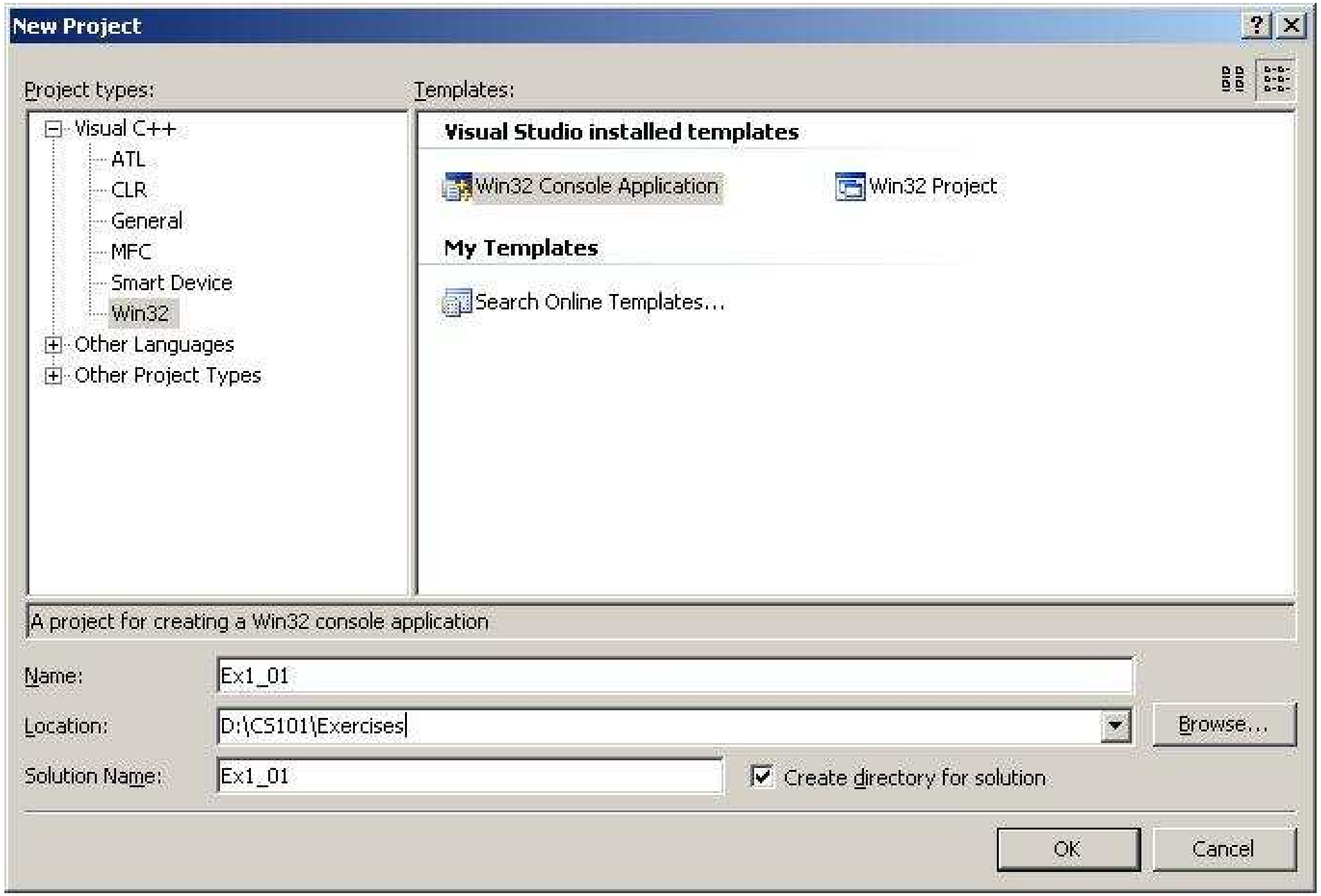
Projects & Solutions

- 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



A project for creating a Win32 console application

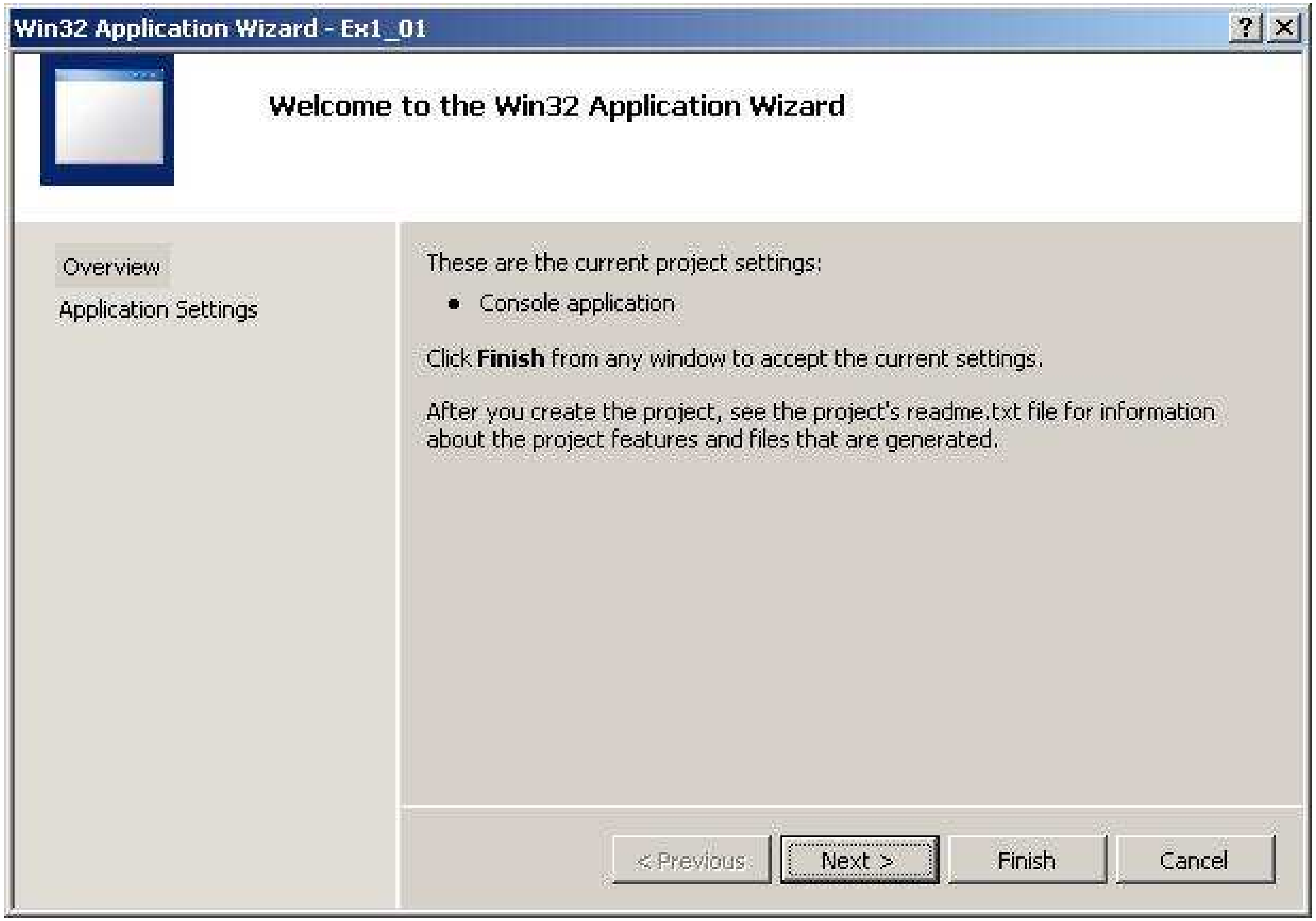
Name: Ex1_01

Location: D:\CS101\Exercises

Solution Name: Ex1_01

☑ Create directory for solution

Win32 Application Wizard dialog box



Win32 Application Wizard - Ex1_01
?
Welcome to the Win32 Application Wizard
Overview
Application Settings

These are the current project settings:

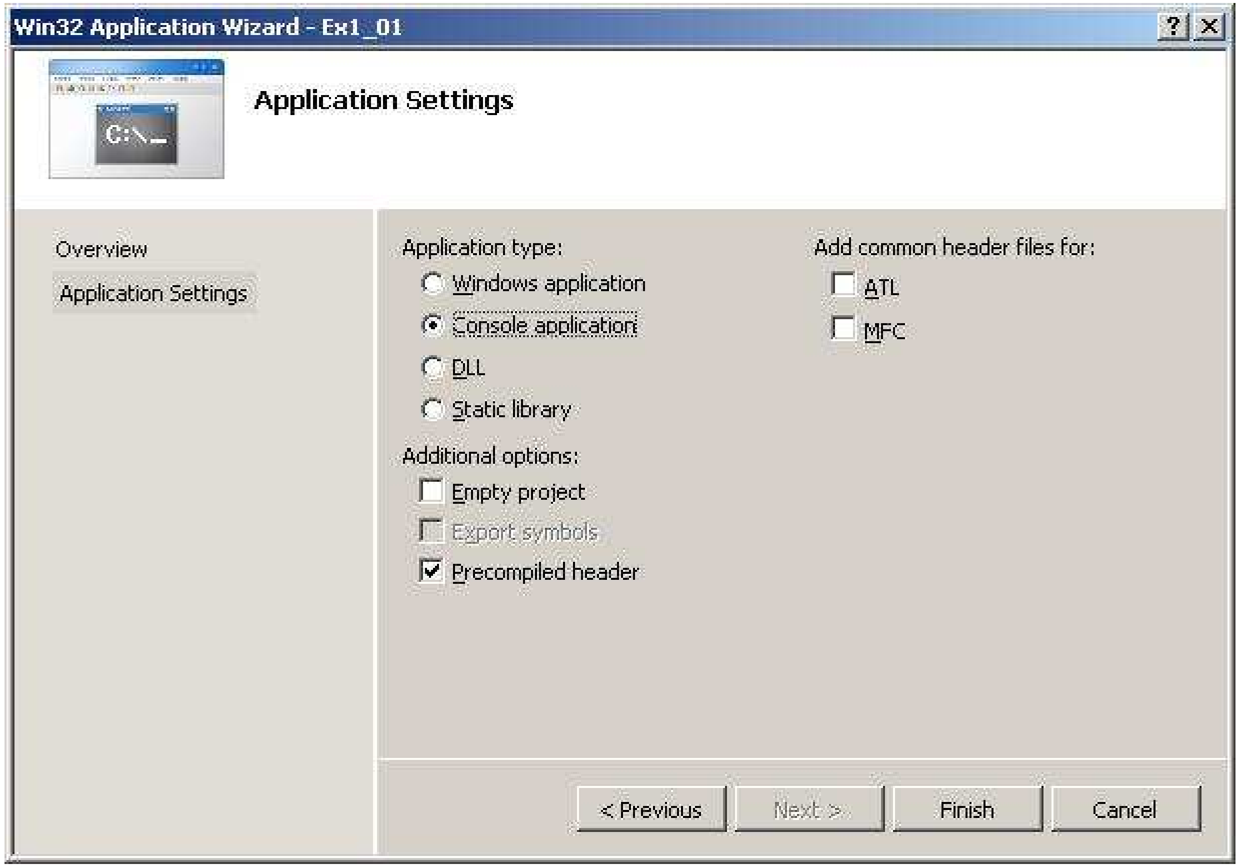
- Console application

Click **Finish** from any window to accept the current settings.

After you create the project, see the project's readme.txt file for information about the project features and files that are generated.

< Previous
Next >
Finish
Cancel

Applications Settings



Win32 Application Wizard - Ex1_01

Application Settings

Overview

Application Settings

Application type:

○ Windows application

● Console application

○ DLL

○ Static library

Additional options:

□ Empty project

□ Export symbols

☑ Precompiled header

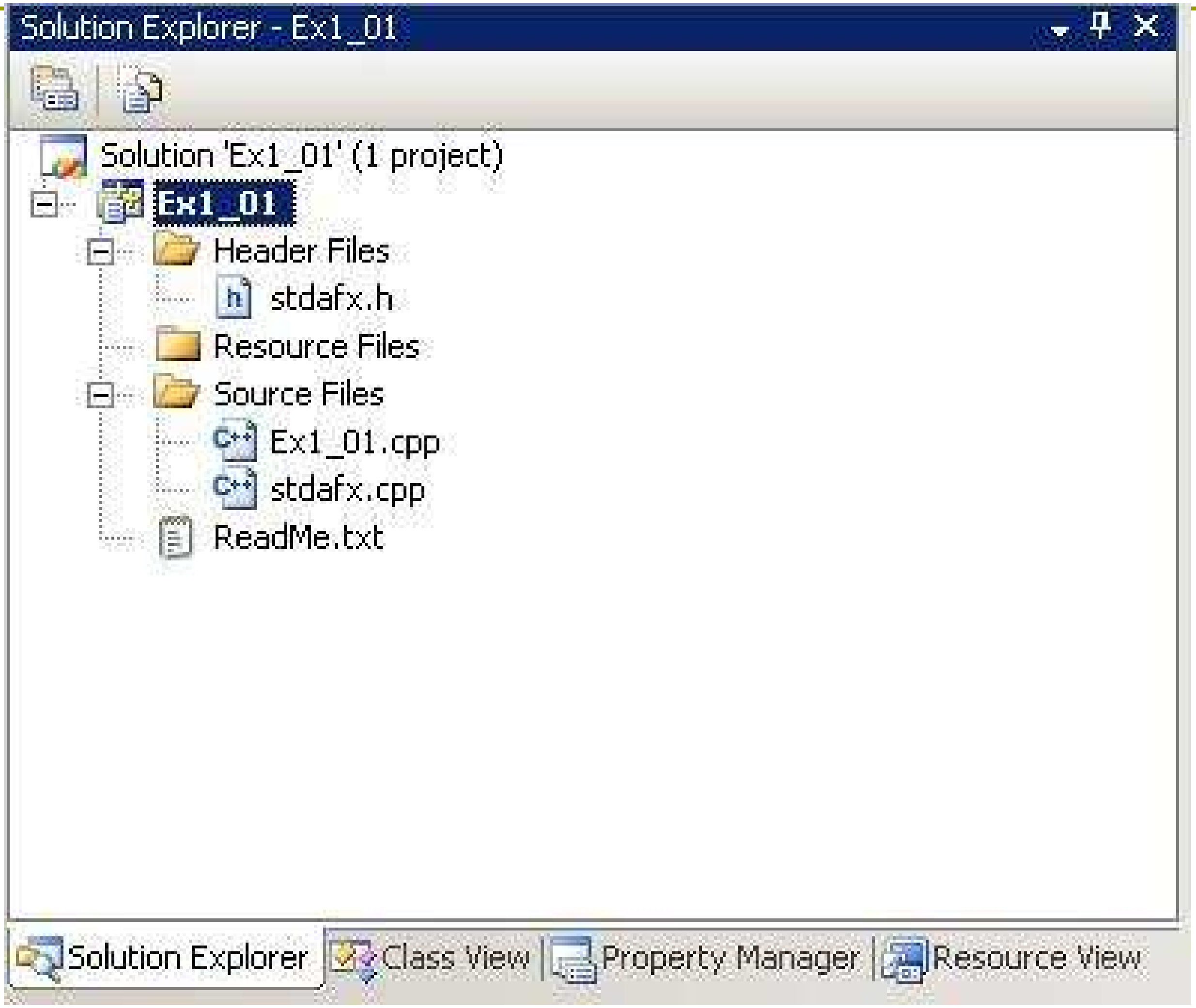
Add common header files for:

□ ATL

□ MFC

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

Solution Explorer

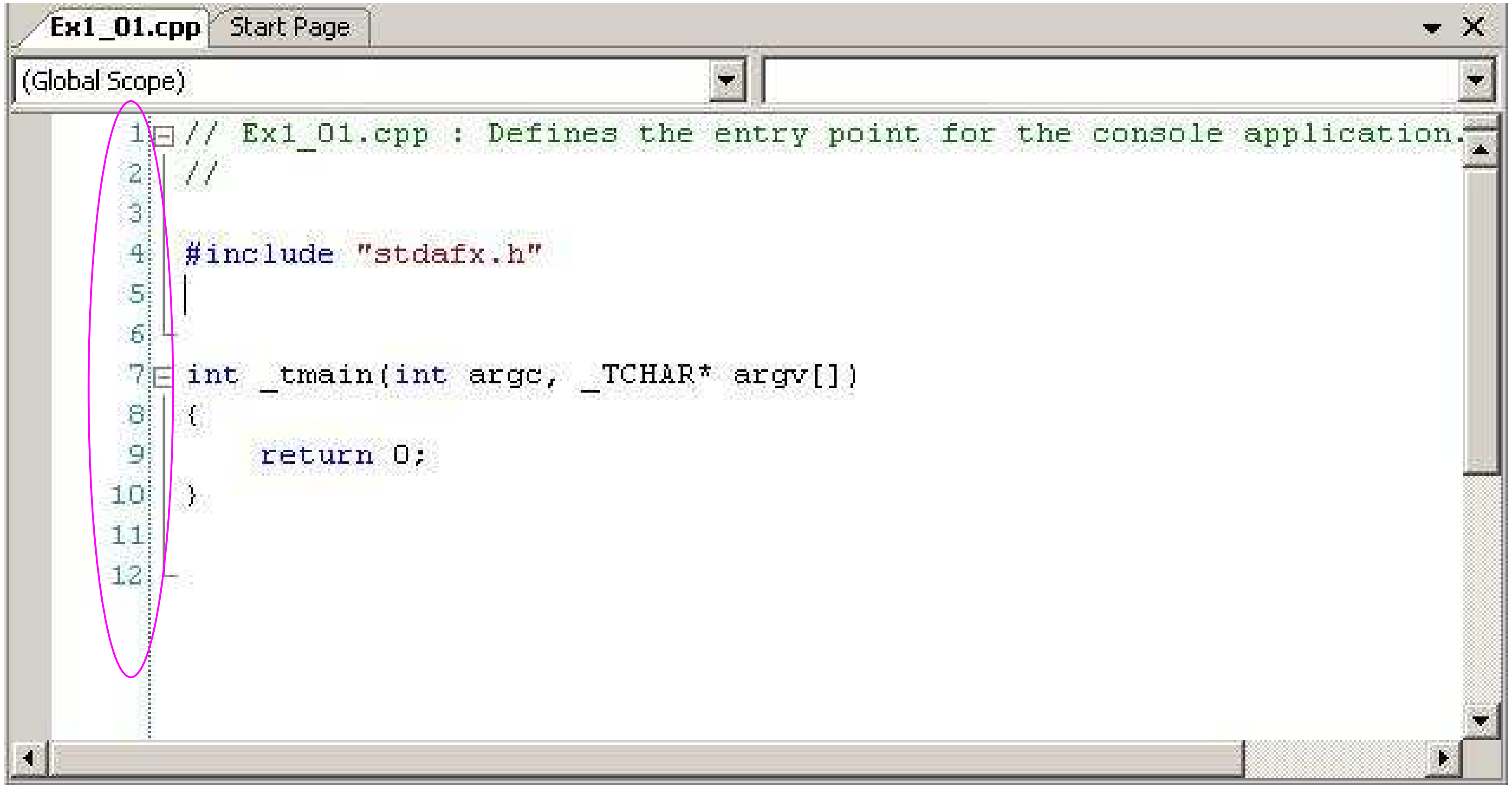


Solution Explorer - Ex1_01

- Solution 'Ex1_01' (1 project)
  - Ex1_01
    - Header Files
      - stdafx.h
    - Resource Files
    - Source Files
      - Ex1_01.cpp
      - stdafx.cpp
    - ReadMe.txt

Modifying the Source Code

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



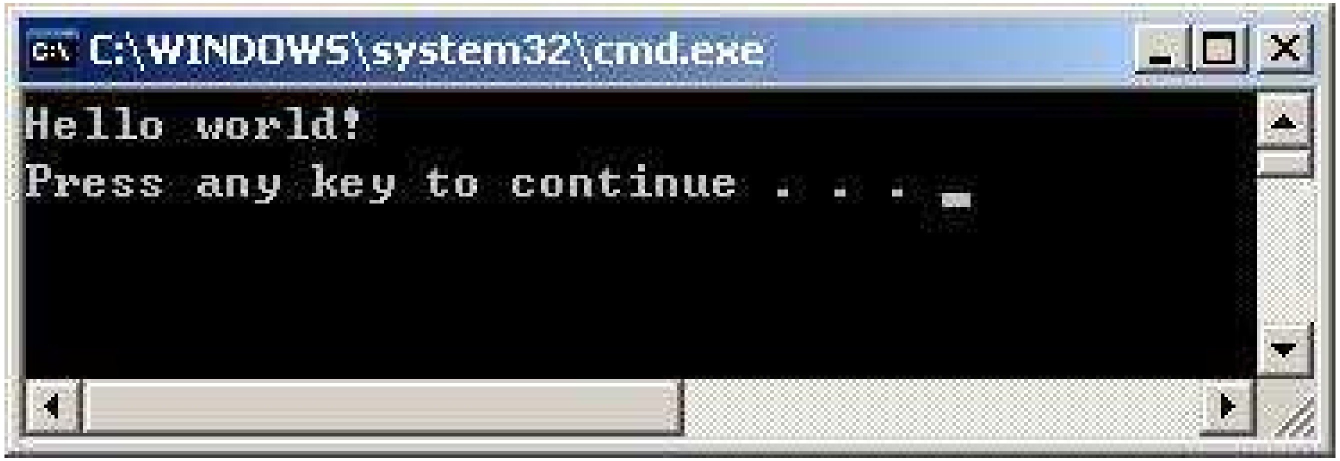
```
// Ex1_01.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"


int _tmain(int argc, _TCHAR* argv[])
{
	return 0;
}

```

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;
}
```

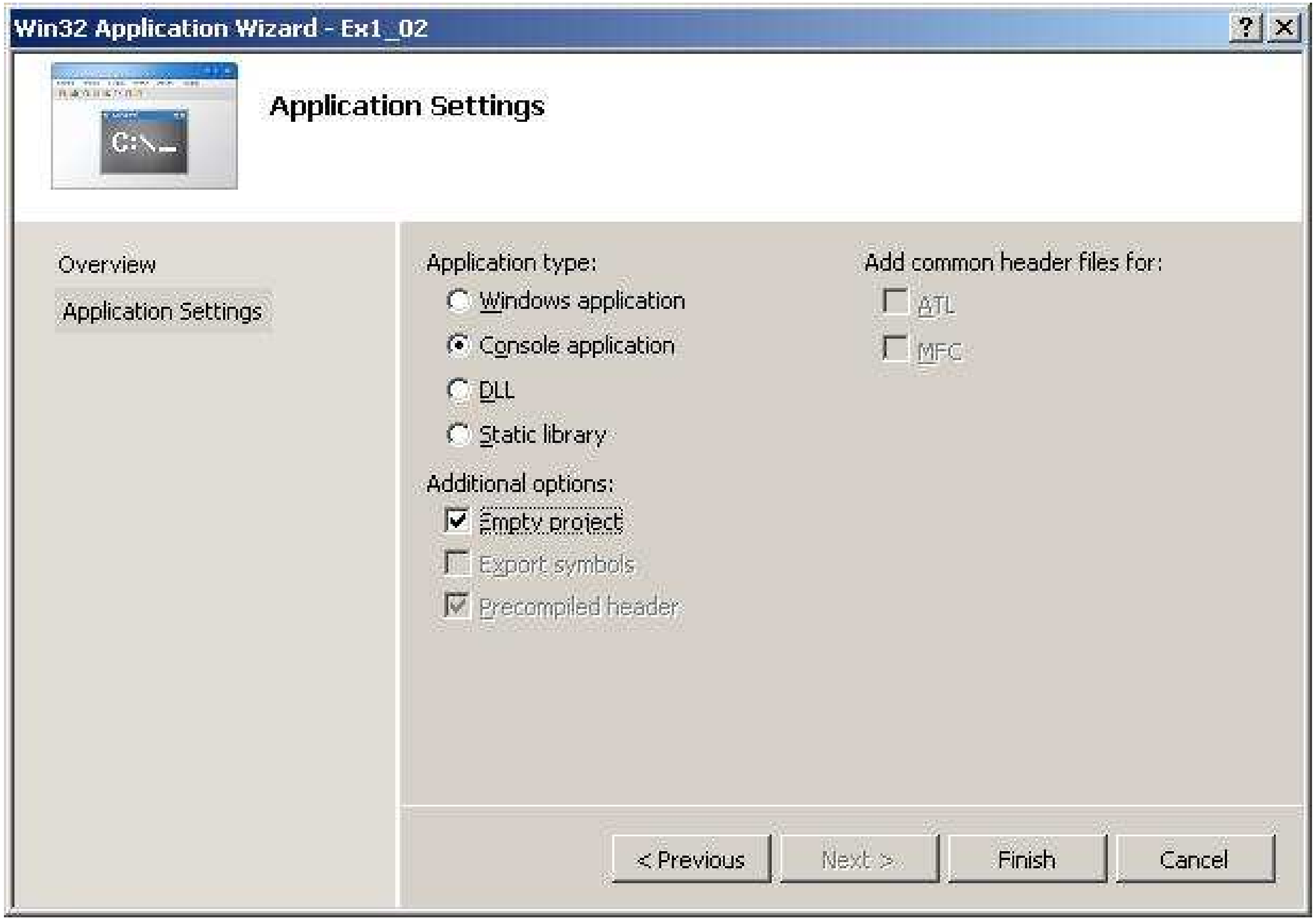
Executing the Program

- Compile the solution
  - Select the `Build > Build Solution` menu item or press `F7`.
- Execute your program
  - Press `Ctrl + F5`



```
Hello world!
Press any key to continue . . .
```

Creating an Empty Console Project



Win32 Application Wizard - Ex1_02

Application Settings

Application type:

- ○ Windows application
- ● Console application
- ○ DLL
- ○ Static library

Additional options:

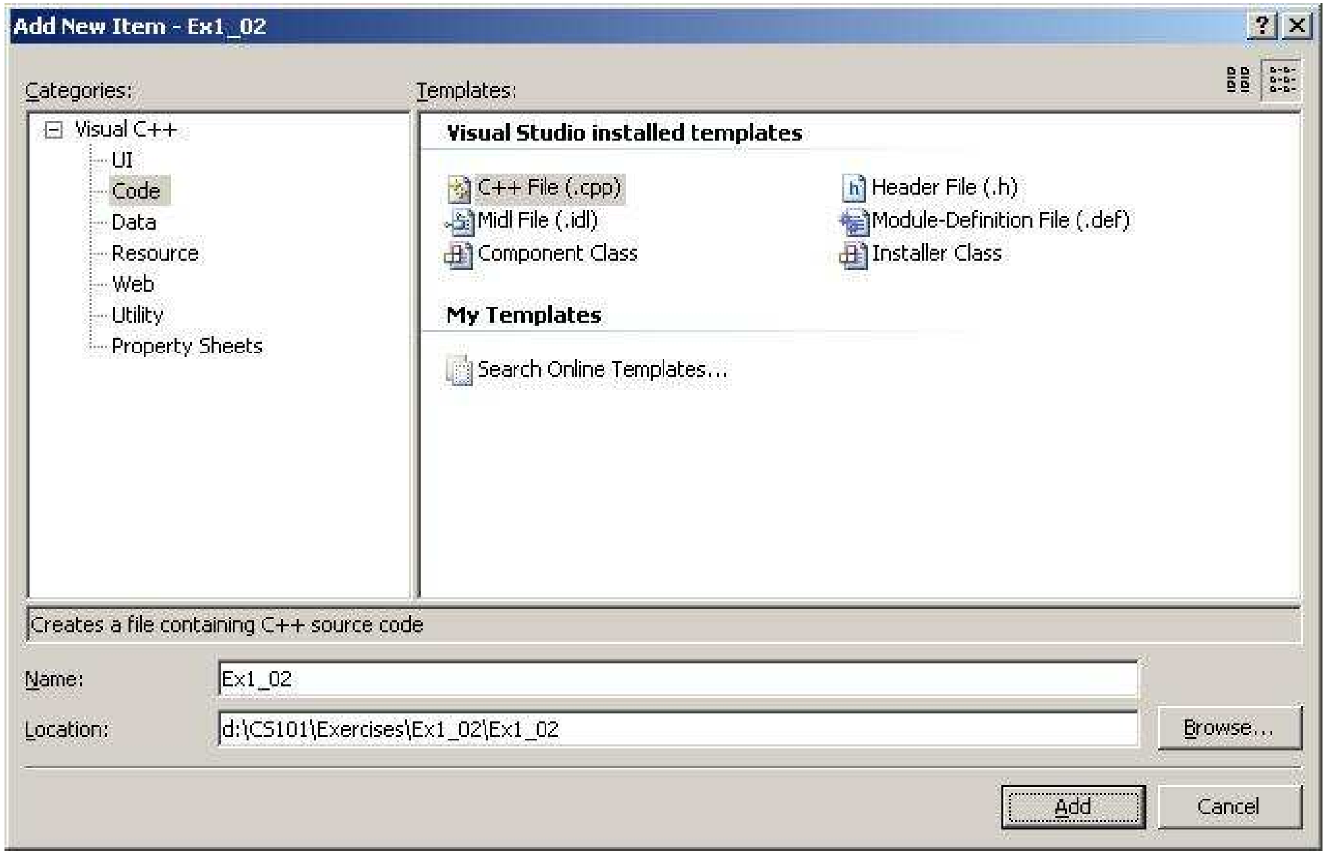
- ☑ Empty project
- □ Export symbols
- ☑ Precompiled header

Add common header files for:

- □ ATL
- □ MFC

Add a New Source File

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



Creates a file containing C++ source code

Name: Ex1_02

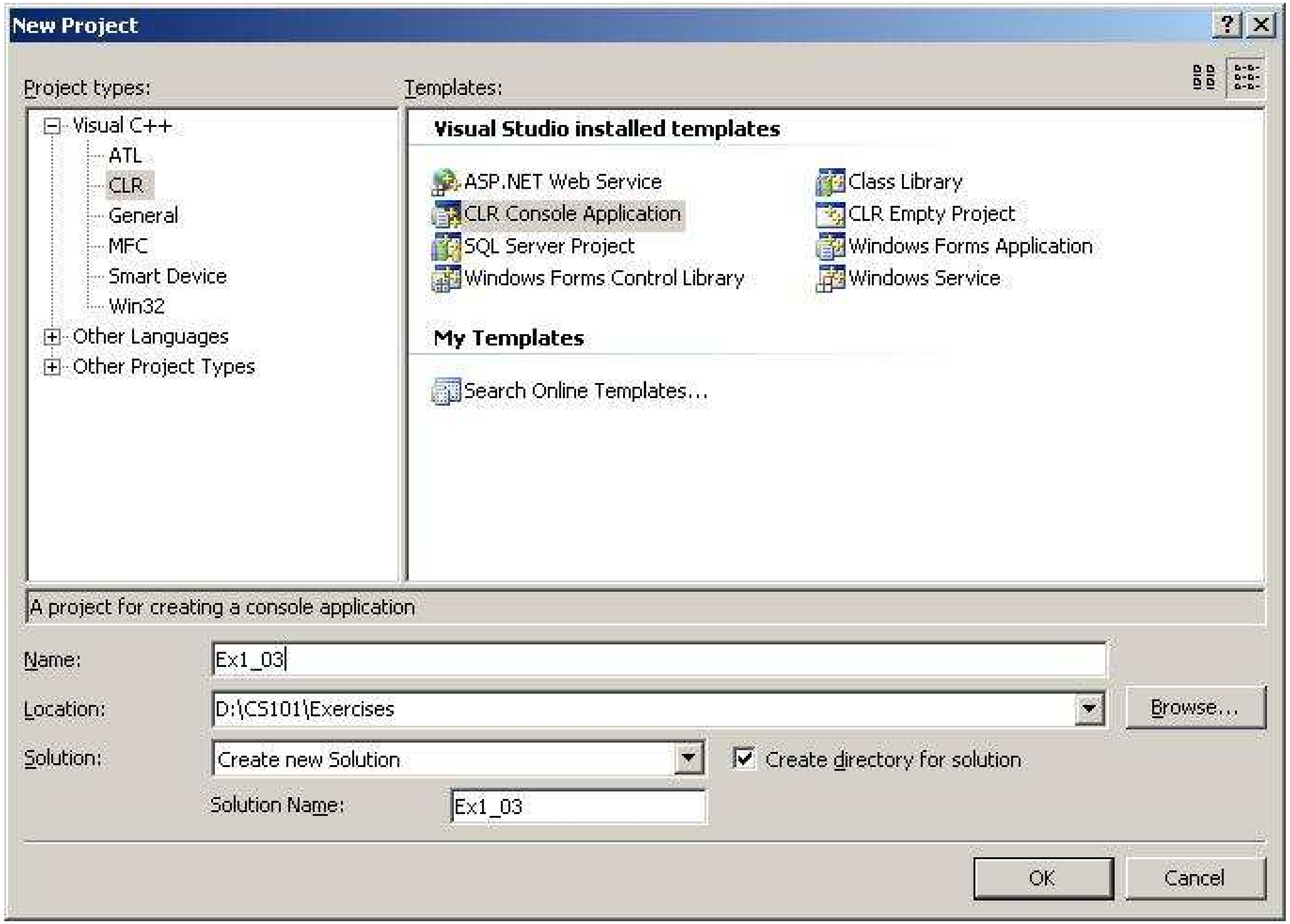
Location: d:\CS101\Exercises\Ex1_02\Ex1_02

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



A project for creating a console application

Name: Ex1_03

Location: D:\CS101\Exercises

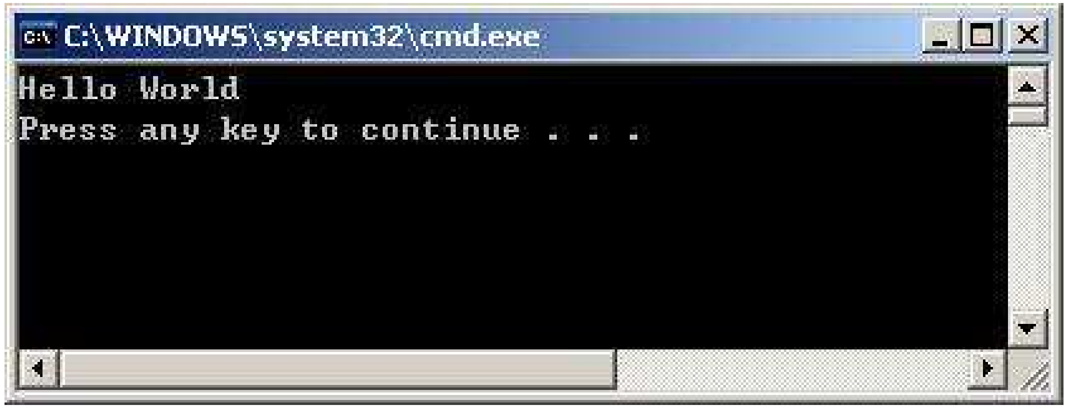
Solution: Create new Solution

☑ Create directory for solution

Solution Name: Ex1_03

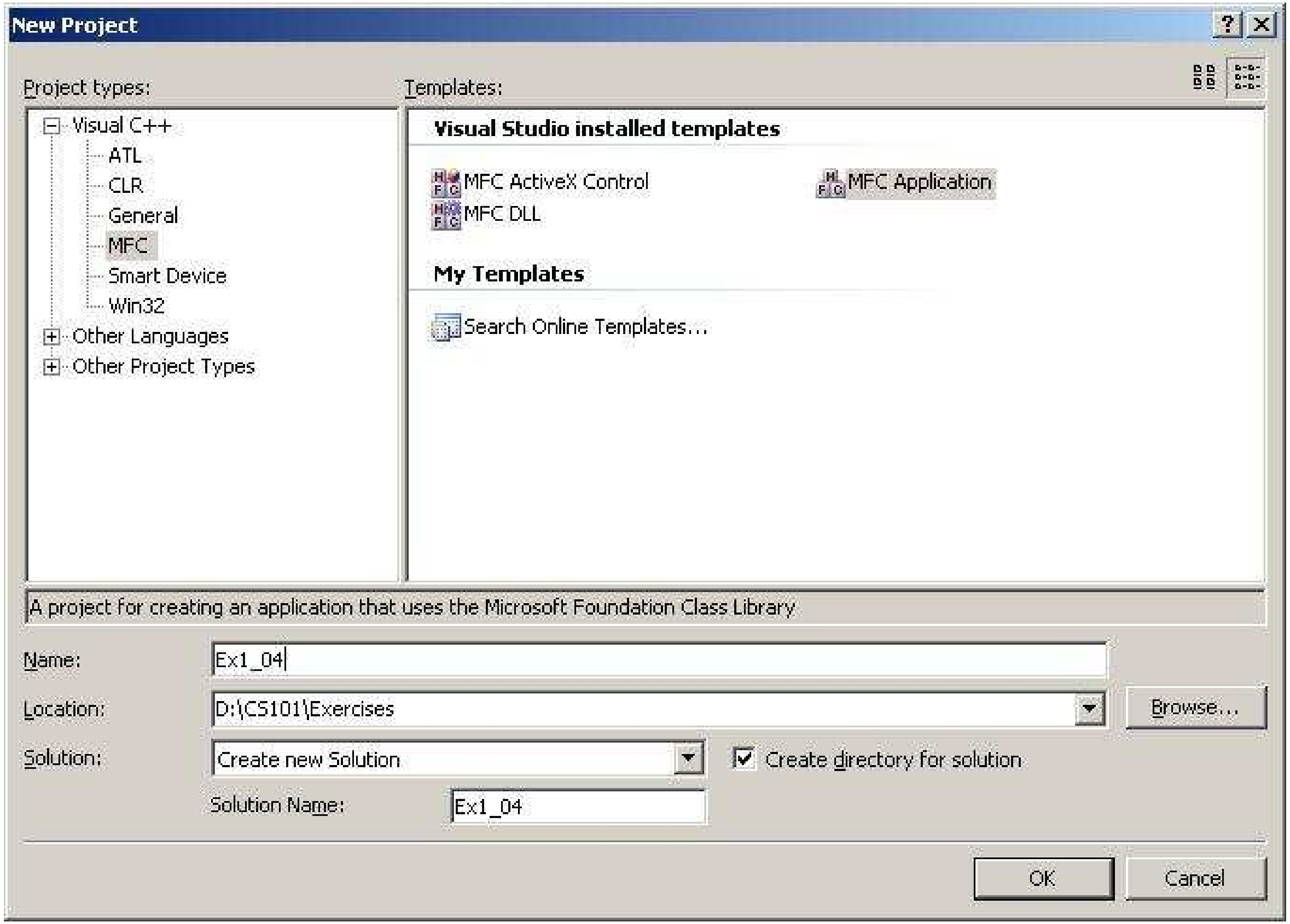
Ex1_03.cpp

```
// 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;
}
```



```
Hello World
Press any key to continue . . .
```

Creating a Windows Application using MFC



A project for creating an application that uses the Microsoft Foundation Class Library

Name: Ex1_04

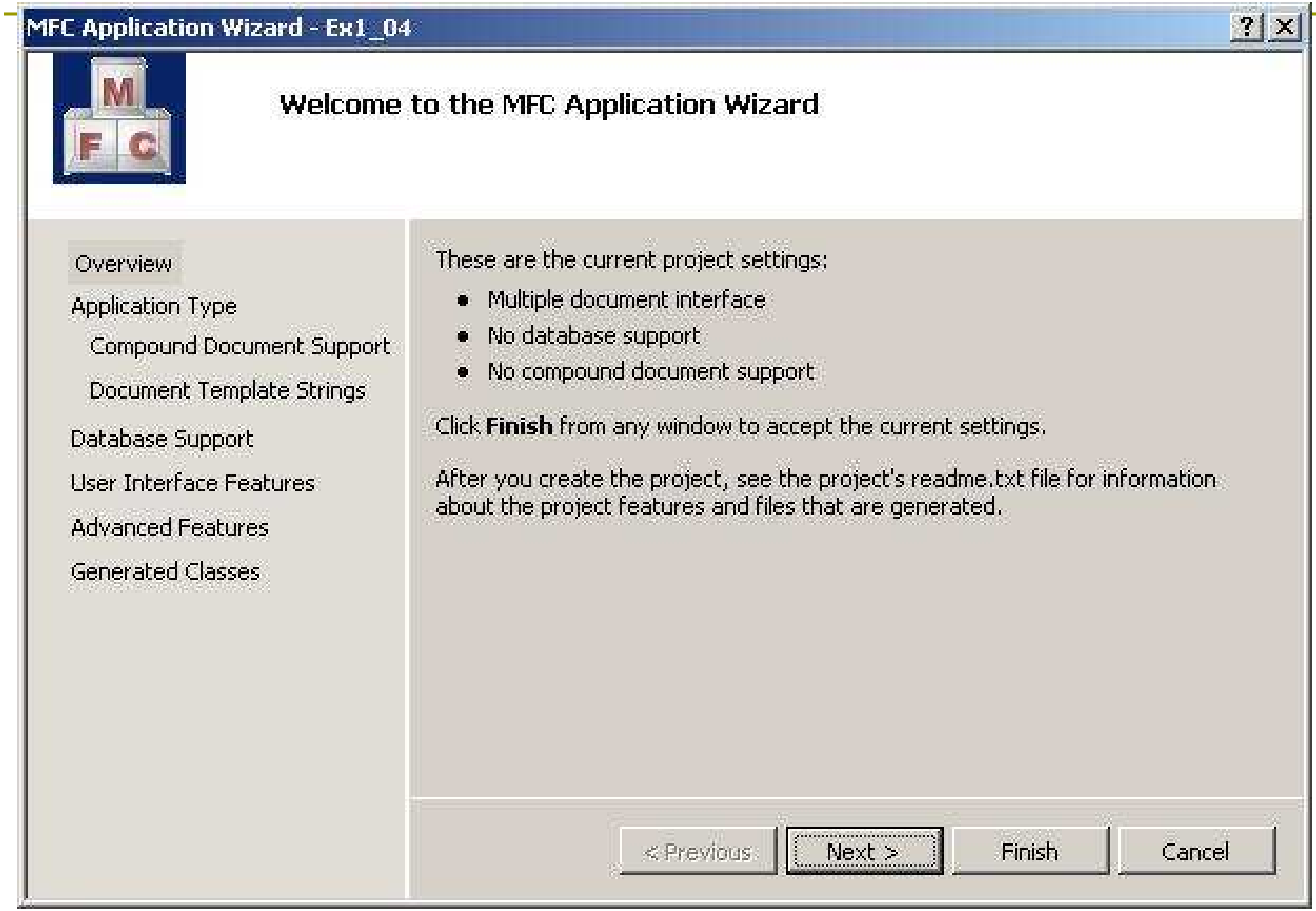
Location: D:\CS101\Exercises

Solution: Create new Solution

☑ Create directory for solution

Solution Name: Ex1_04

MFC Applications Wizard



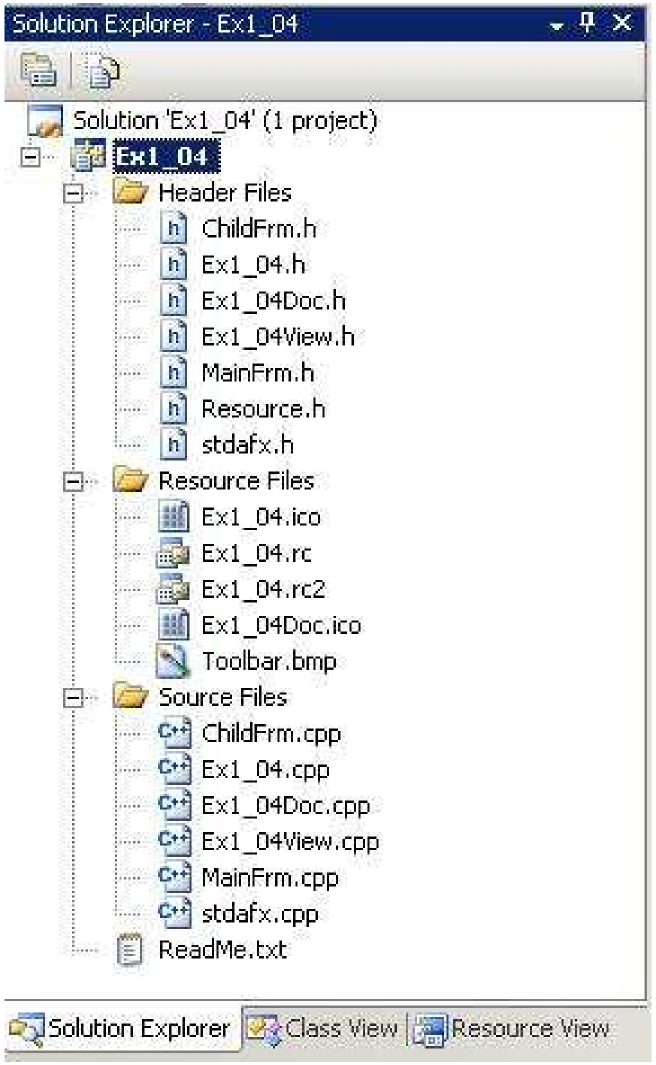
These are the current project settings:

- Multiple document interface
- No database support
- No compound document support

Click **Finish** from any window to accept the current settings.

After you create the project, see the project's readme.txt file for information about the project features and files that are generated.

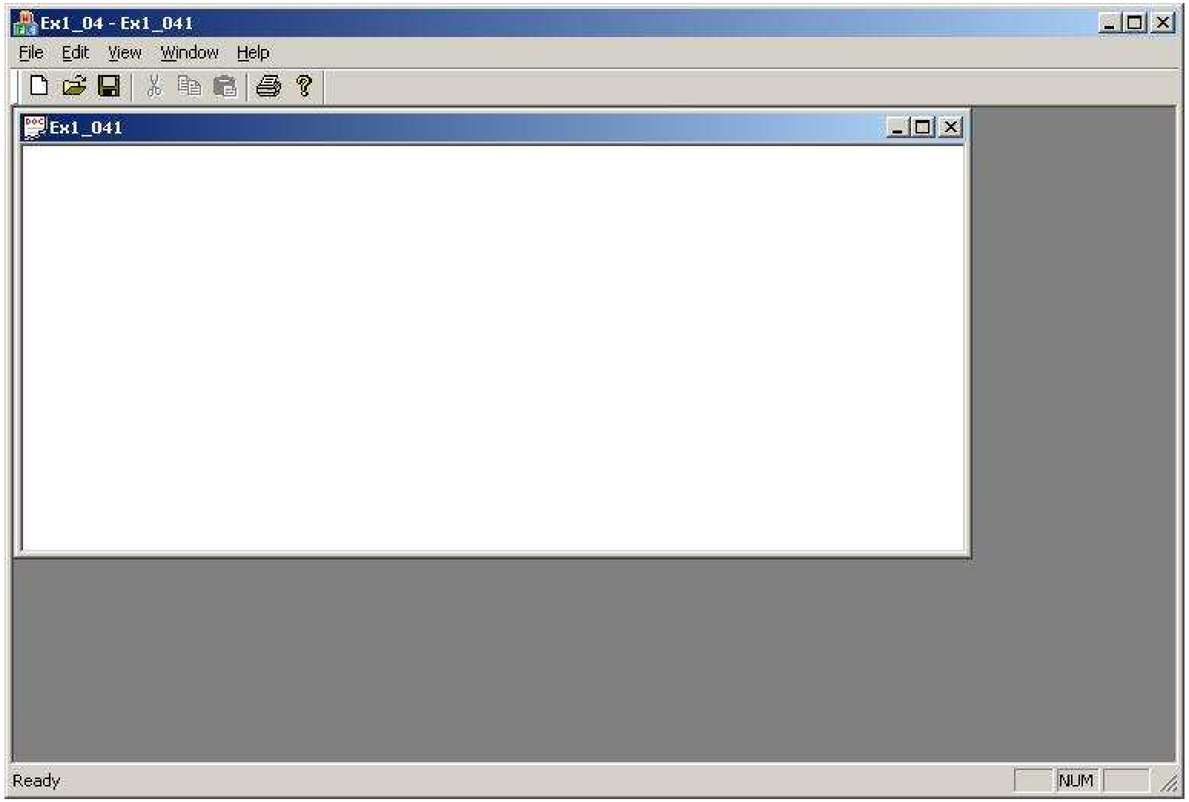
24 Files Are Generated Automatically



```
Solution 'Ex1_04' (1 project)
  Ex1_04
    Header Files
      ChildFrm.h
      Ex1_04.h
      Ex1_04Doc.h
      Ex1_04View.h
      MainFrm.h
      Resource.h
      stdafx.h
    Resource Files
      Ex1_04.ico
      Ex1_04.rc
      Ex1_04.rc2
      Ex1_04Doc.ico
      Toolbar.bmp
    Source Files
      ChildFrm.cpp
      Ex1_04.cpp
      Ex1_04Doc.cpp
      Ex1_04View.cpp
      MainFrm.cpp
      stdafx.cpp
    ReadMe.txt
```

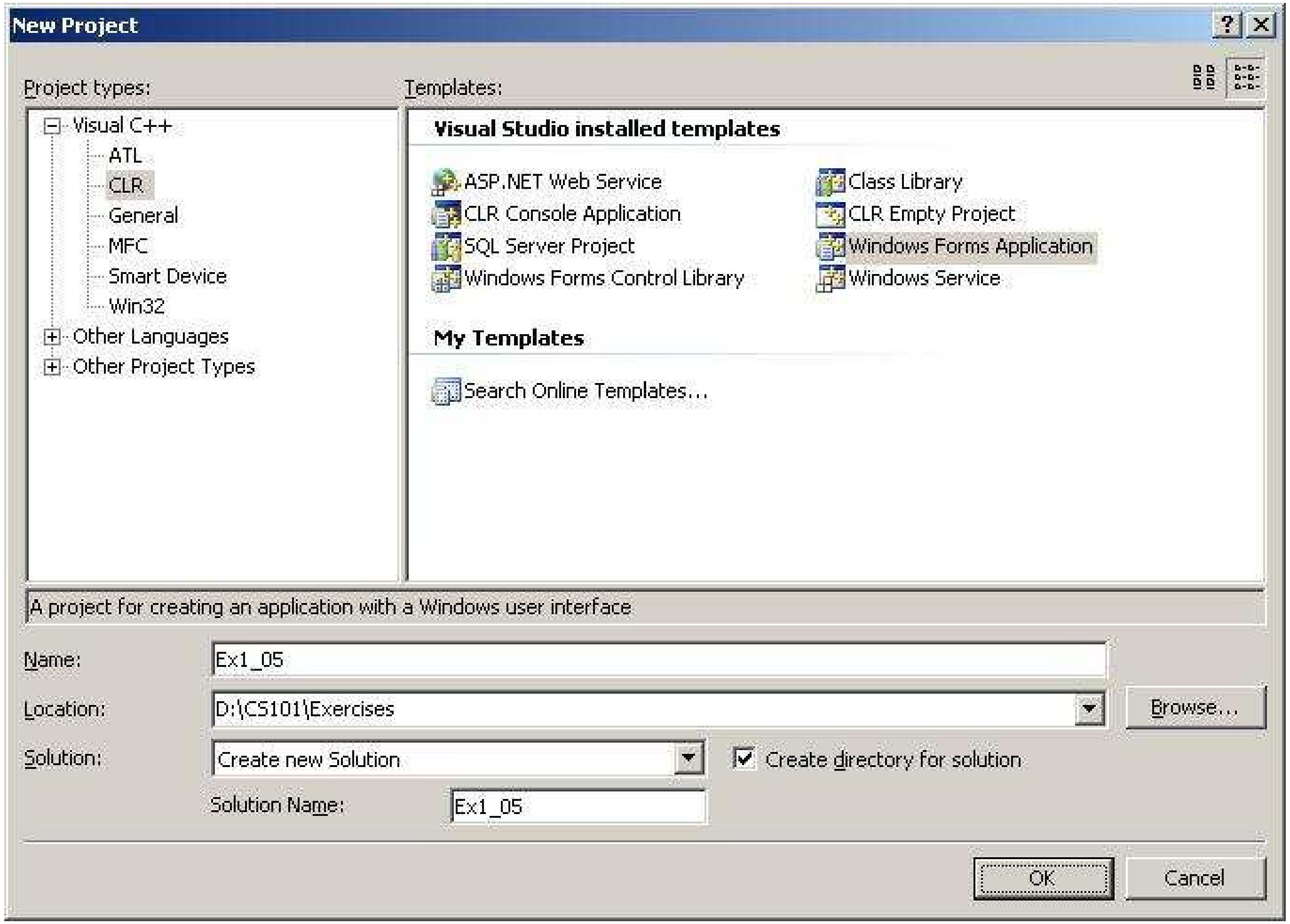
- 3 are in the solution folder
- 17 are in the project folder
- Four are in a subfolder res to the project folder

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



Project types:

- Visual C++
  - ATL
  - CLR
  - General
  - MFC
  - Smart Device
  - Win32
- Other Languages
- Other Project Types

Templates:

Visual Studio installed templates

ASP.NET Web Service	Class Library
CLR Console Application	CLR Empty Project
SQL Server Project	Windows Forms Application
Windows Forms Control Library	Windows Service

My Templates

Search Online Templates...

A project for creating an application with a Windows user interface

Name: Ex1_05

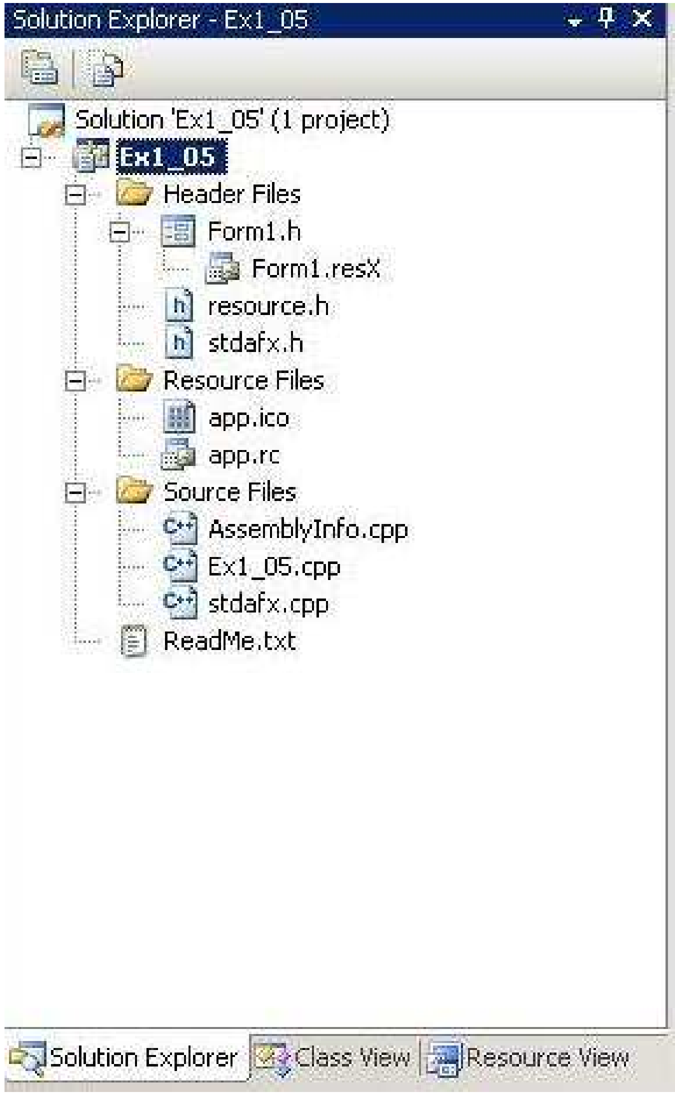
Location: D:\CS101\Exercises

Solution: Create new Solution

☑ Create directory for solution

Solution Name: Ex1_05

Total of 15 Files



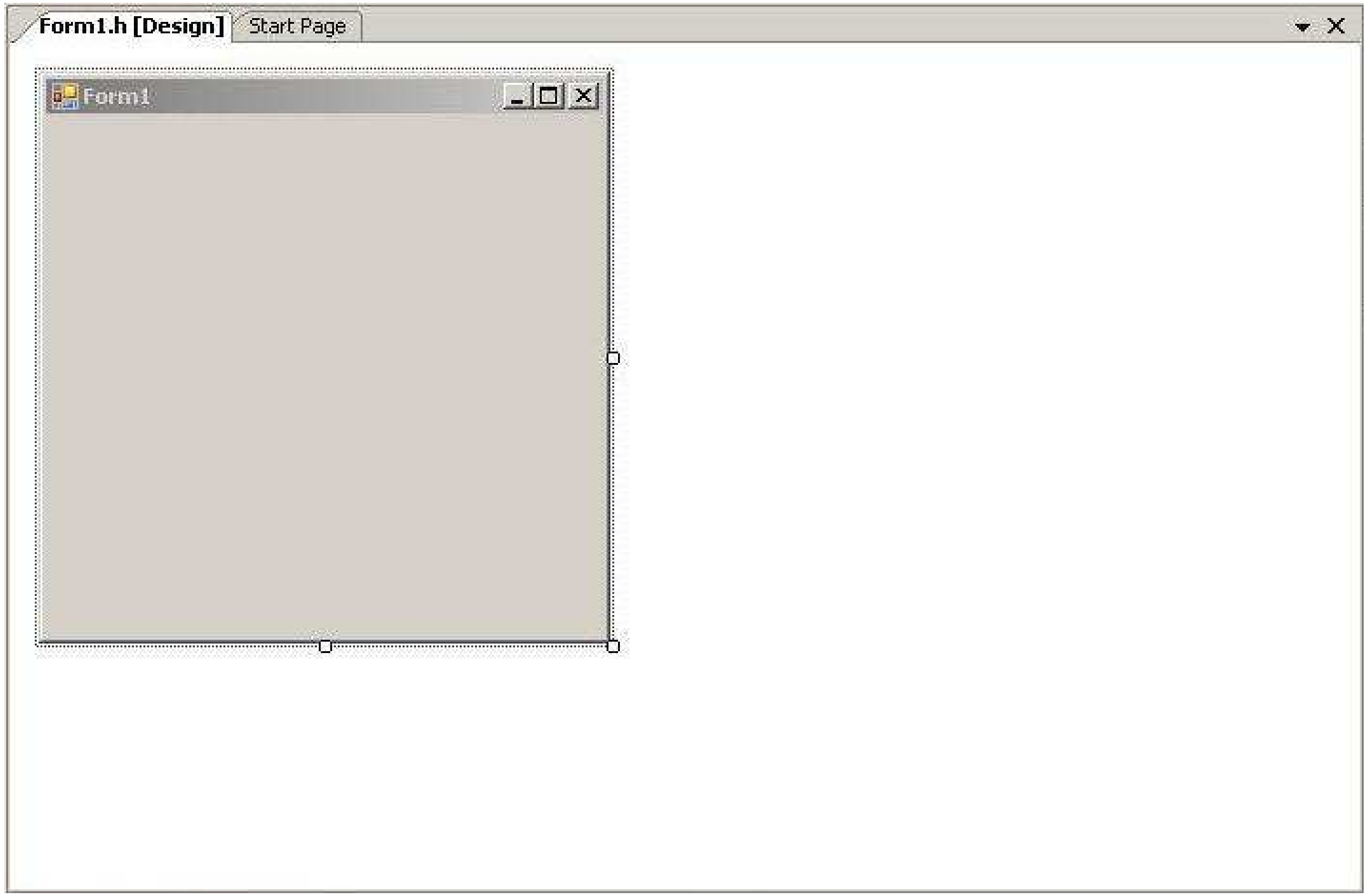
Solution Explorer - Ex1_05

```
Solution 'Ex1_05' (1 project)
  Ex1_05
    Header Files
      Form1.h
        Form1.resX
      resource.h
      stdafx.h
    Resource Files
      app.ico
      app.rc
    Source Files
      AssemblyInfo.cpp
      Ex1_05.cpp
      stdafx.cpp
    ReadMe.txt
```

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

Editor Window

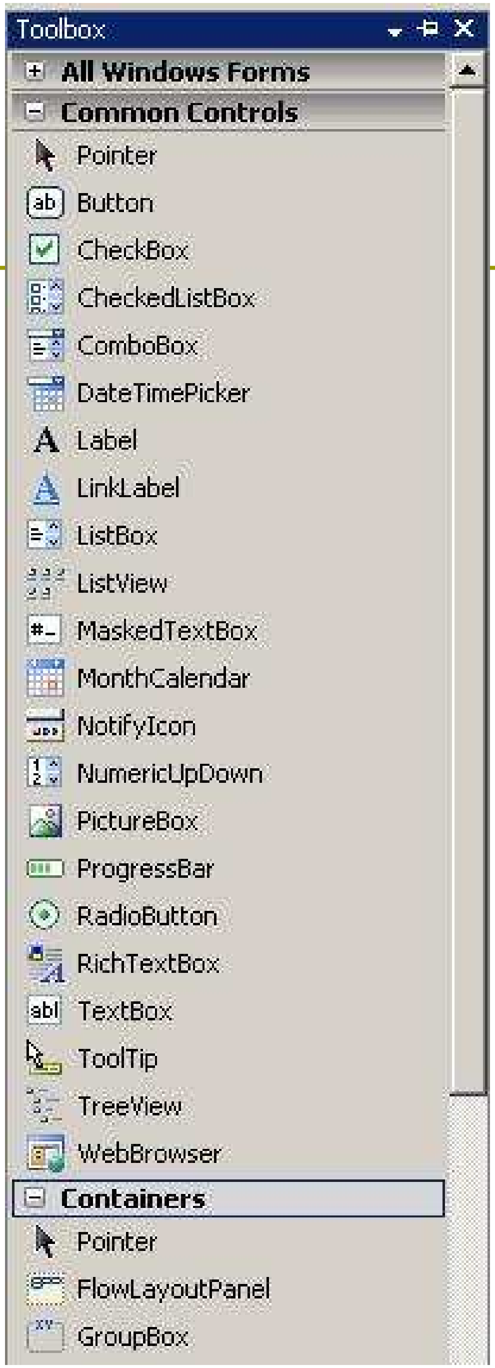
- The Editor window shows an image of the application window rather than code.



Add GUI Components

- `View > Toolbox`
  or `Ctrl+Alt+X`

- Add two buttons



Toolbox

⊞ All Windows Forms

⊟ Common Controls

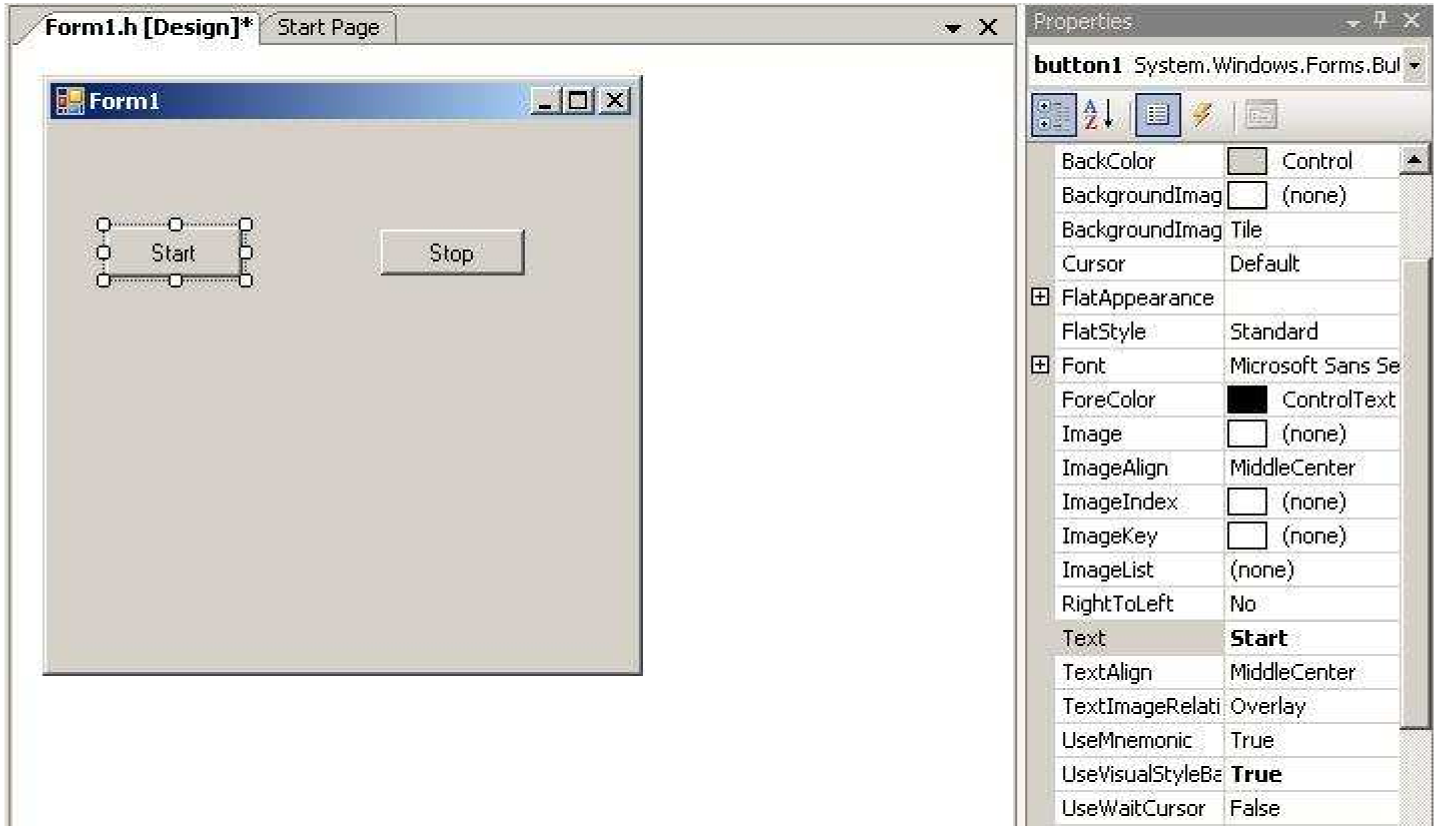
- Pointer
- Button
- CheckBox
- CheckedListBox
- ComboBox
- DateTimePicker
- Label
- LinkLabel
- ListBox
- ListView
- MaskedTextBox
- MonthCalendar
- NotifyIcon
- NumericUpDown
- PictureBox
- ProgressBar
- RadioButton
- RichTextBox
- TextBox
- ToolTip
- TreeView
- WebBrowser

⊟ Containers

- Pointer
- FlowLayoutPanel
- GroupBox

Figure 1-29

- Creating a Windows application cannot be easier!



button1 System.Windows.Forms.Bu

BackColor	Control
BackgroundImag	(none)
BackgroundImag	Tile
Cursor	Default
⊞ FlatAppearance	
FlatStyle	Standard
⊞ Font	Microsoft Sans Se
ForeColor	ControlText
Image	(none)
ImageAlign	MiddleCenter
ImageIndex	(none)
ImageKey	(none)
ImageList	(none)
RightToLeft	No
Text	**Start**
TextAlign	MiddleCenter
TextImageRelati	Overlay
UseMnemonic	True
UseVisualStyleBa	**True**
UseWaitCursor	False