MFC - DIALOG BOXES

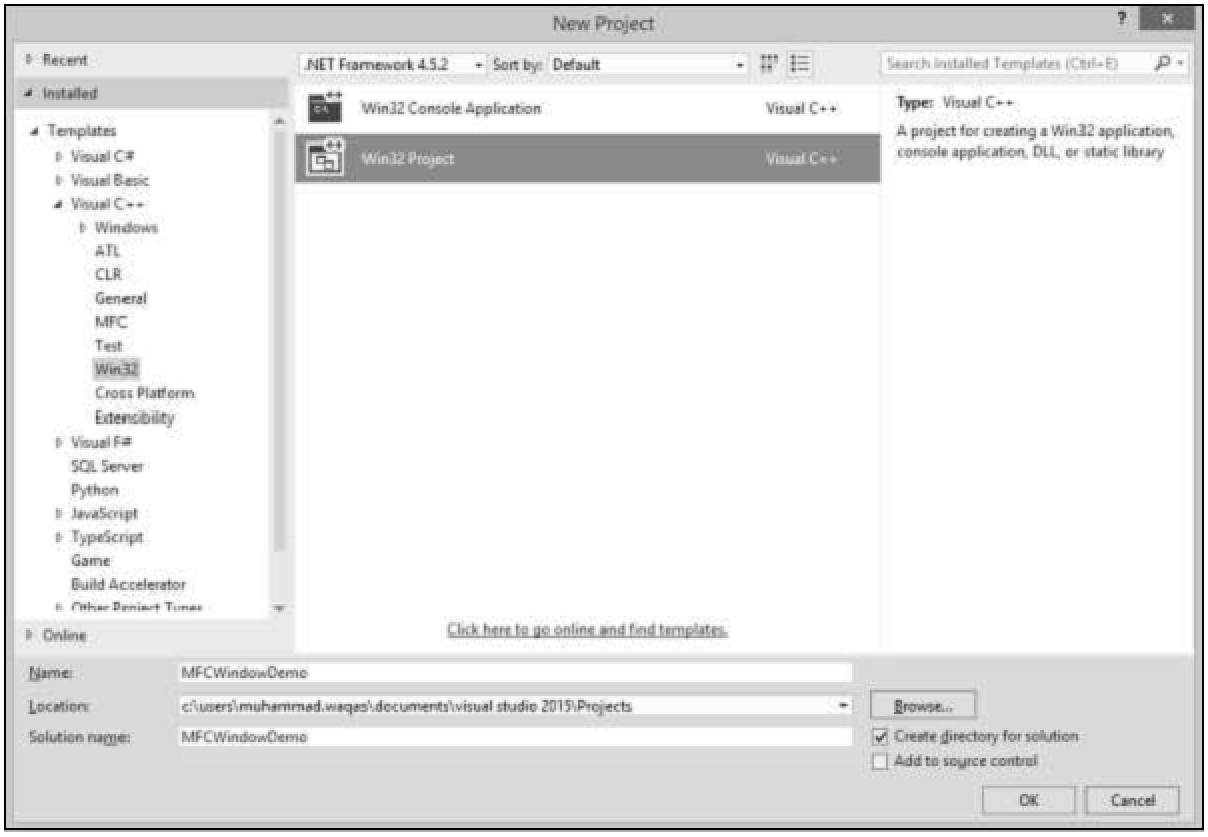
In this chapter, we will be covering the Dialog boxes. Applications for Windows frequently communicate with the user through dialog boxes. **CDialog class** provides an interface for managing dialog boxes. The Visual C++ dialog editor makes it easy to design dialog boxes and create their dialog-template resources.

- Creating a dialog object is a two-phase operation –
    - Construct the dialog object.
    - Create the dialog window.

Let us look into a simple example by creating a new Win32 project.

Step 1 – Open the Visual studio and click on the File → New → Project menu option.

Step 2 – You can now see the New Project dialog box.



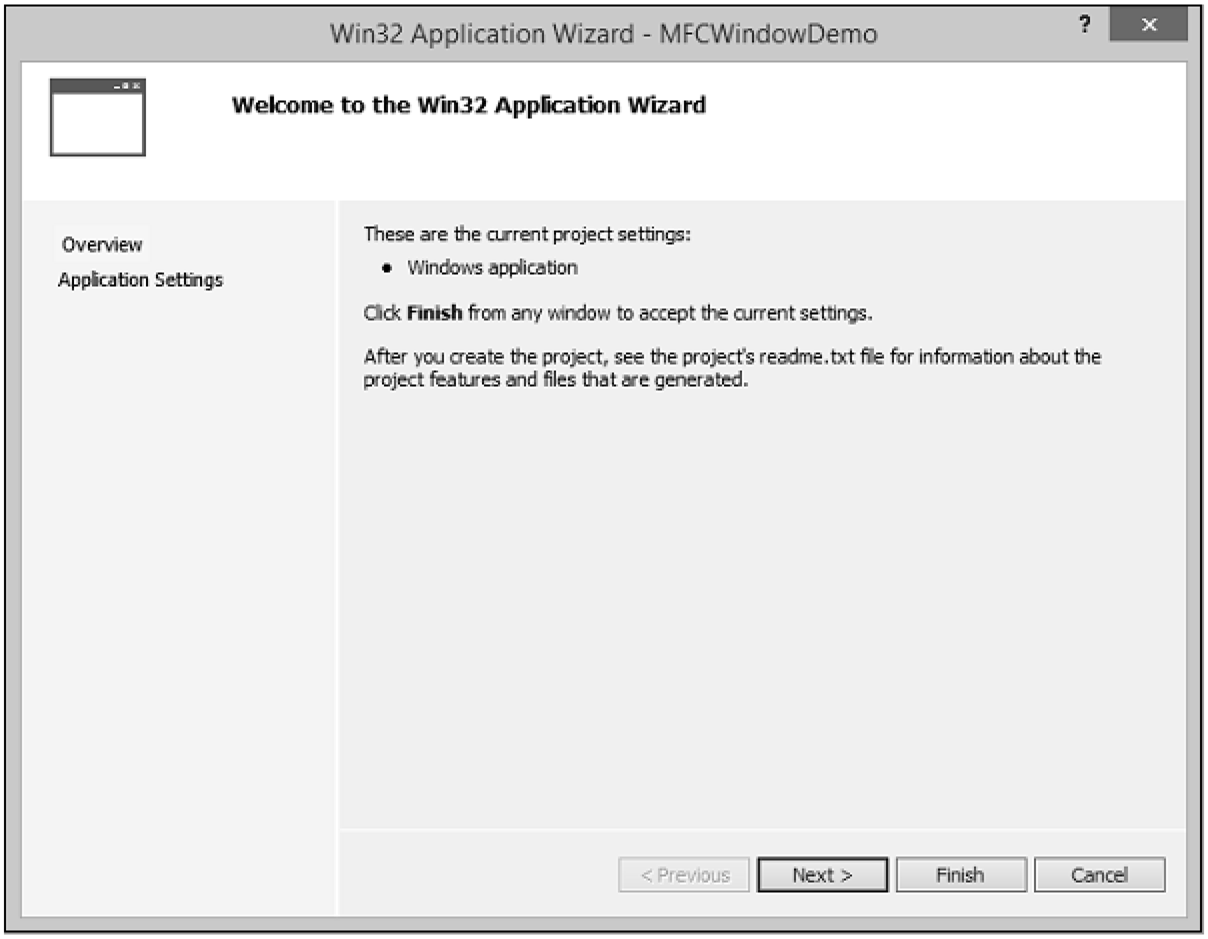
Type: Visual C++
A project for creating a Win32 application, console application, DLL, or static library

Name: MFCWindowDemo
Location: c:\users\muhammad.waqas\documents\visual studio 2015\Projects
Solution name: MFCWindowDemo
[✓] Create directory for solution
[ ] Add to source control

Step 3 – From the left pane, select Templates → Visual C++ → Win32.

Step 4 – In the middle pane, select Win32 Project.

Step 5 – Enter project name ‘MFCDialogDemo’ in the Name field and click OK to continue. You will see the following dialog.



Welcome to the Win32 Application Wizard

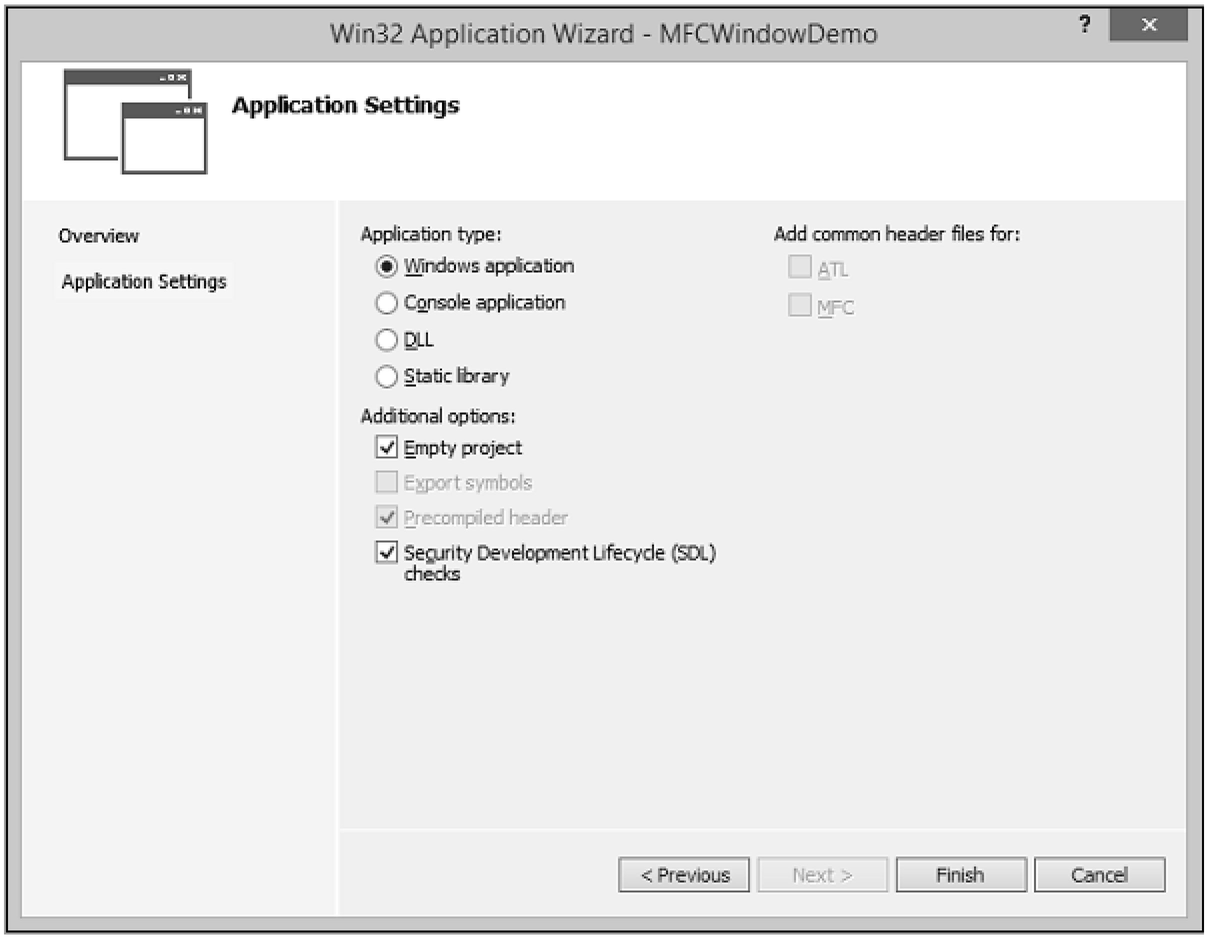
These are the current project settings:

- Windows 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.

Step 6 – Click Next.

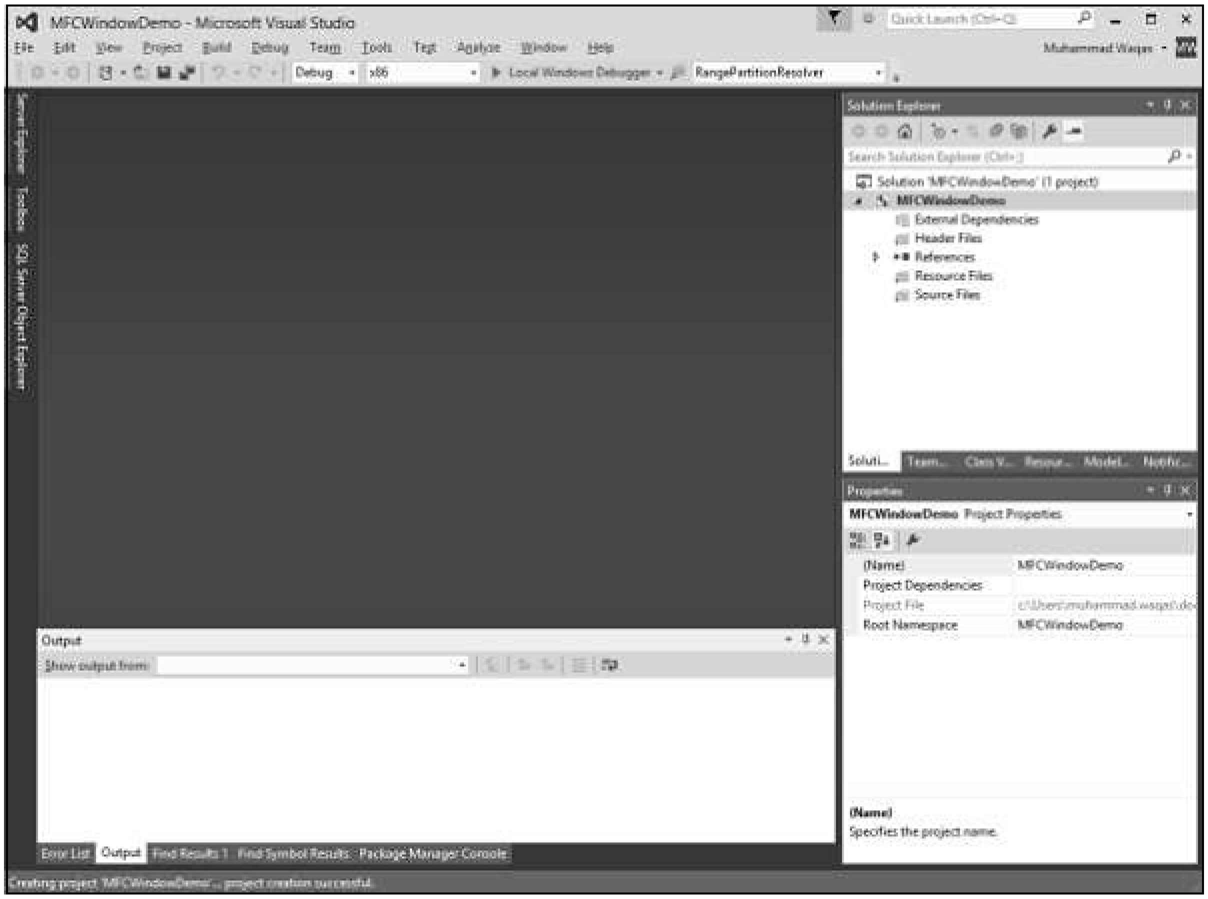


Application type:
(● Windows application) ○ Console application ○ DLL ○ Static library

Additional options:
[✓] Empty project
[ ] Export symbols
[✓] Precompiled header
[✓] Security Development Lifecycle (SDL) checks

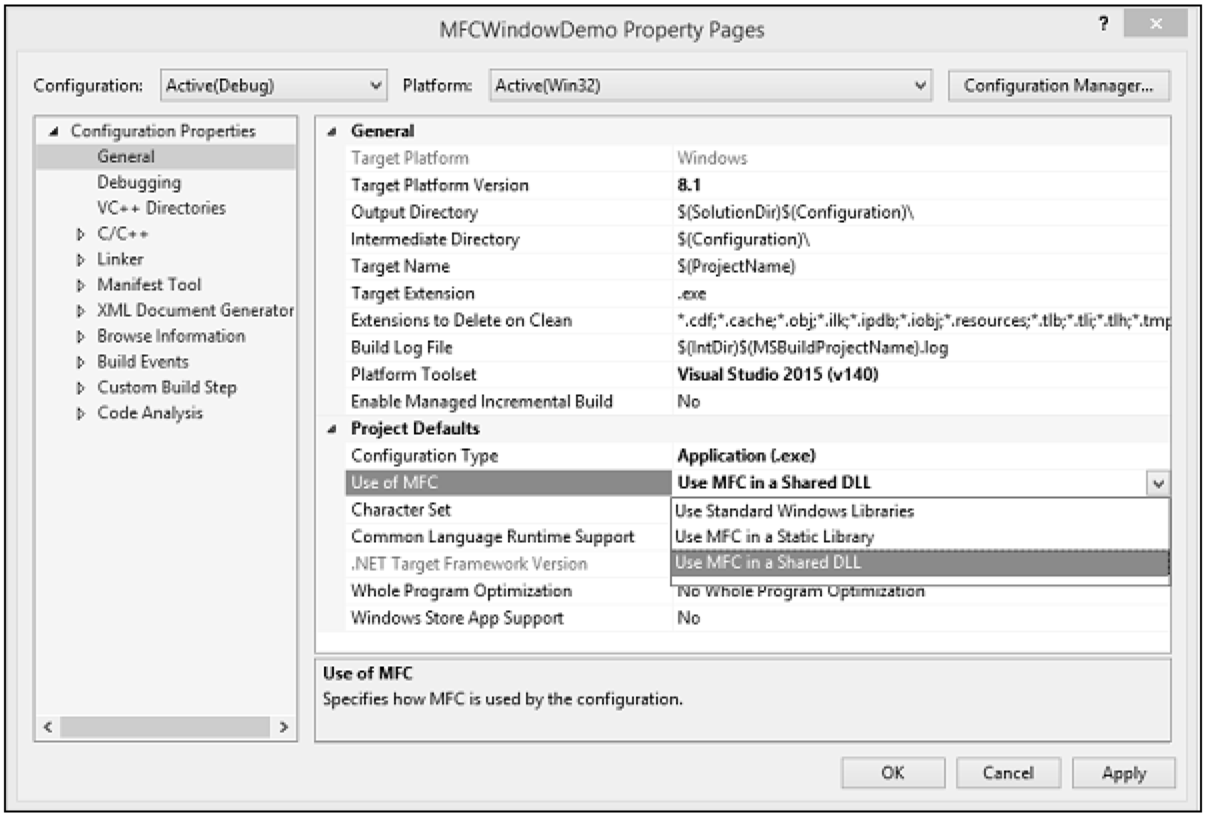
Add common header files for:
[ ] ATL
[ ] MFC

Step 7 – Select the options shown in the dialog box given above and click Finish.


MFCWindowDemo - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug x86 Local Windows Debugger RangePartitionResolver
Solution Explorer
Solution 'MFCWindowDemo' (1 project)
MFCWindowDemo
External Dependencies
Header Files
References
Resource Files
Source Files
Properties
MFCWindowDemo Project Properties
(Name) MFCWindowDemo
Project Dependencies
Root Namespace MFCWindowDemo
Output
Show output from:
Error List Output Find Results 1 Find Symbol Results Package Manager Console
Creating project 'MFCWindowDemo'... project creation successful.

Step 8 – An empty project is created.

Step 9 – To make it a MFC project, right-click on the project and select Properties.



MFCWindowDemo Property Pages

Configuration: Active(Debug) Platform: Active(Win32)

General	
Target Platform	Windows
Target Platform Version	8.1
Output Directory	$(SolutionDir)$(Configuration)\
Intermediate Directory	$(Configuration)\
Target Name	$(ProjectName)
Target Extension	.exe
Extensions to Delete on Clean	*.cdf;*.cache;*.obj;*.ilk;*.ipdb;*.iobj;*.resources;*.tlb;*.tli;*.tlh;*.tmp
Build Log File	$(IntDir)$(MSBuildProjectName).log
Platform Toolset	Visual Studio 2015 (v140)
Enable Managed Incremental Build	No
Project Defaults	
Configuration Type	Application (.exe)
Use of MFC	Use MFC in a Shared DLL
Character Set	Use Standard Windows Libraries
Common Language Runtime Support	Use MFC in a Static Library
.NET Target Framework Version	Use MFC in a Shared DLL
Whole Program Optimization	No Whole Program Optimization
Windows Store App Support	No

Use of MFC
Specifies how MFC is used by the configuration.

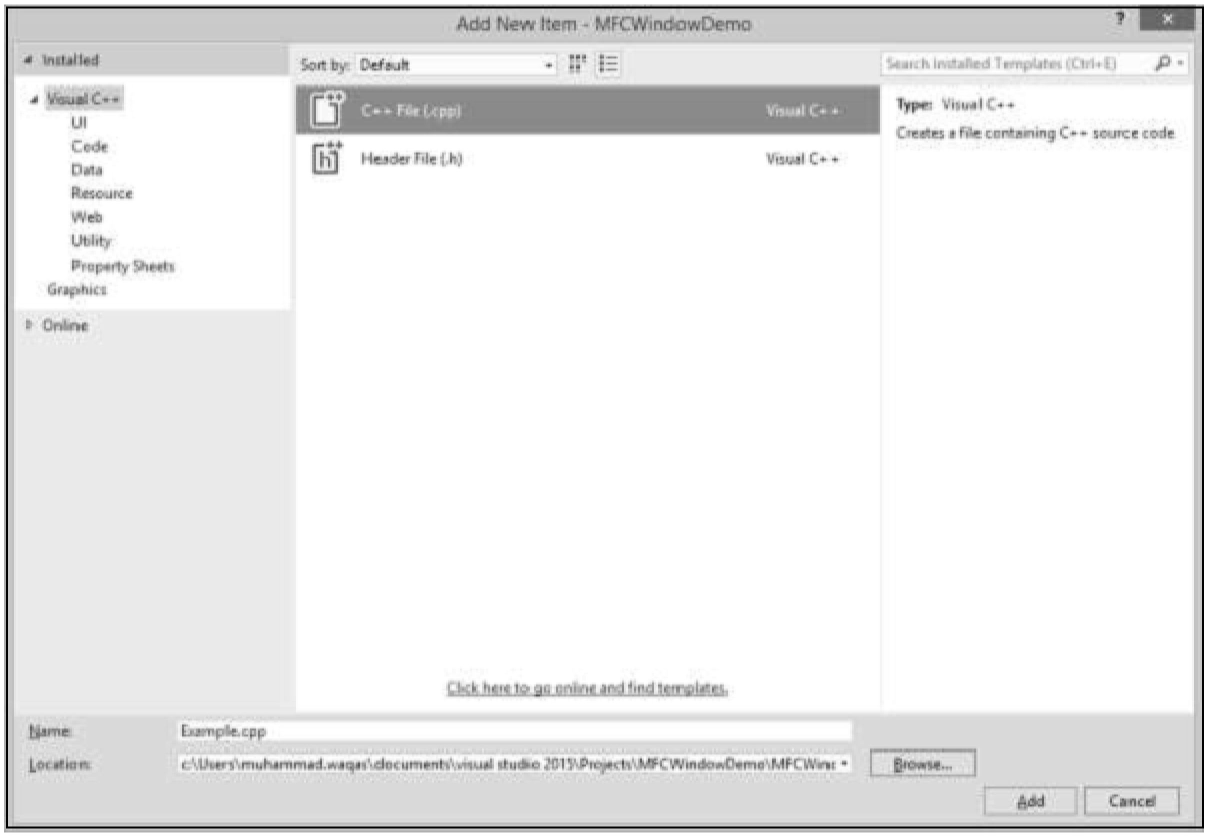
Step 10 – In the left section, click Configuration Properties → General.

Step 11 – Select the Use MFC in Shared DLL option in Project Defaults section and click OK.

Step 12 – Add a new source file.

Step 13 – Right-click on your Project and select Add → New Item.

Step 14 – In the Templates section, click C++ File . *cpp*


Add New Item - MFCWindowDemo
Installed
Visual C++
UI
Code
Data
Resource
Web
Utility
Property Sheets
Graphics
Online
Sort by: Default
Search Installed Templates (Ctrl+E)
C++ File (.cpp) Visual C++
Header File (.h) Visual C++
Type: Visual C++
Creates a file containing C++ source code
Click here to go online and find templates.
Browse...
Add
Cancel

Name: Example.cpp

Location: c:\Users\muhammad.waqas\documents\visual studio 2015\Projects\MFCWindowDemo\MFCWin

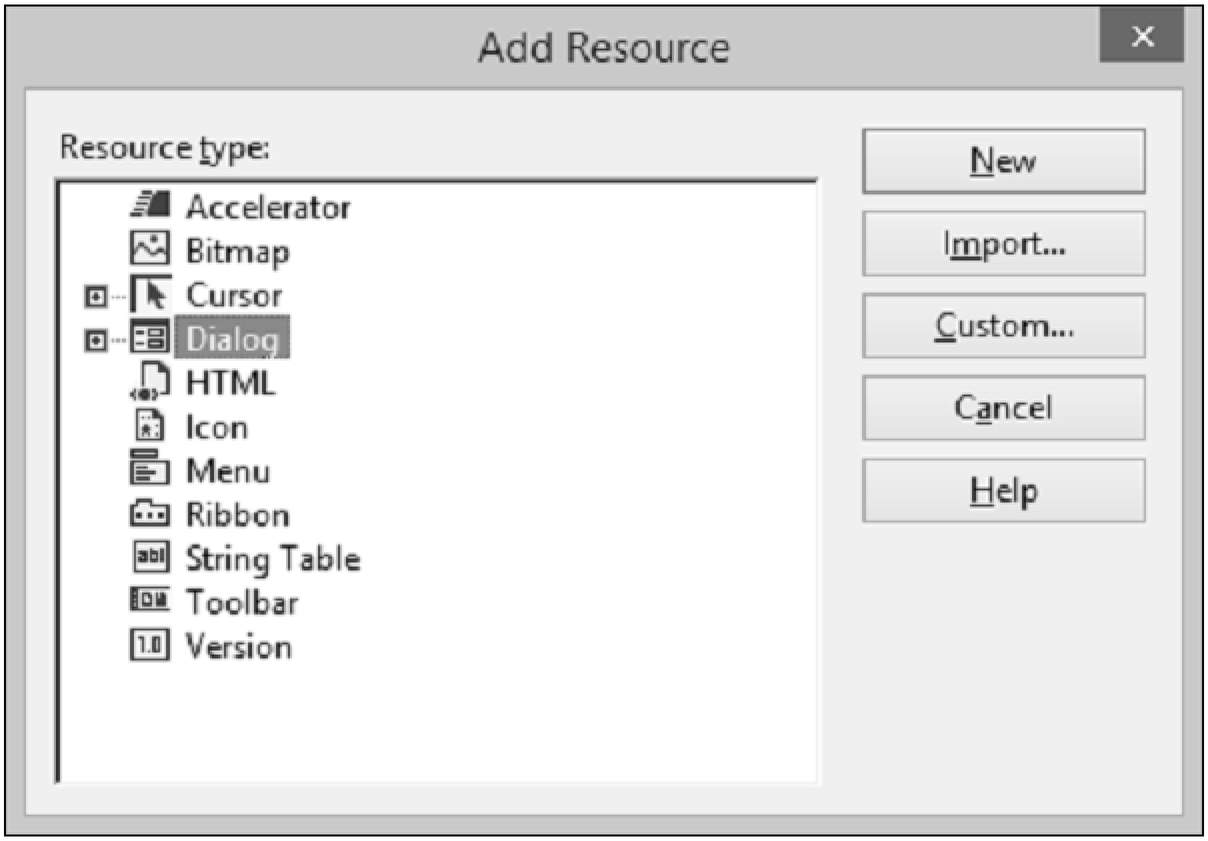
Step 15 – Set the Name as Example and click Add.

Step 16 – To create an application, we need to add a class and derive it from the MFC's CWinApp.

```
#include <afxwin.h>

class CExample : public CWinApp {
   public:
      BOOL InitInstance();
};
```

Dialog Box Creation

Step 1 – To create a dialog box, right-click on the Resource Files folder in solution explorer and select Add → Resource.

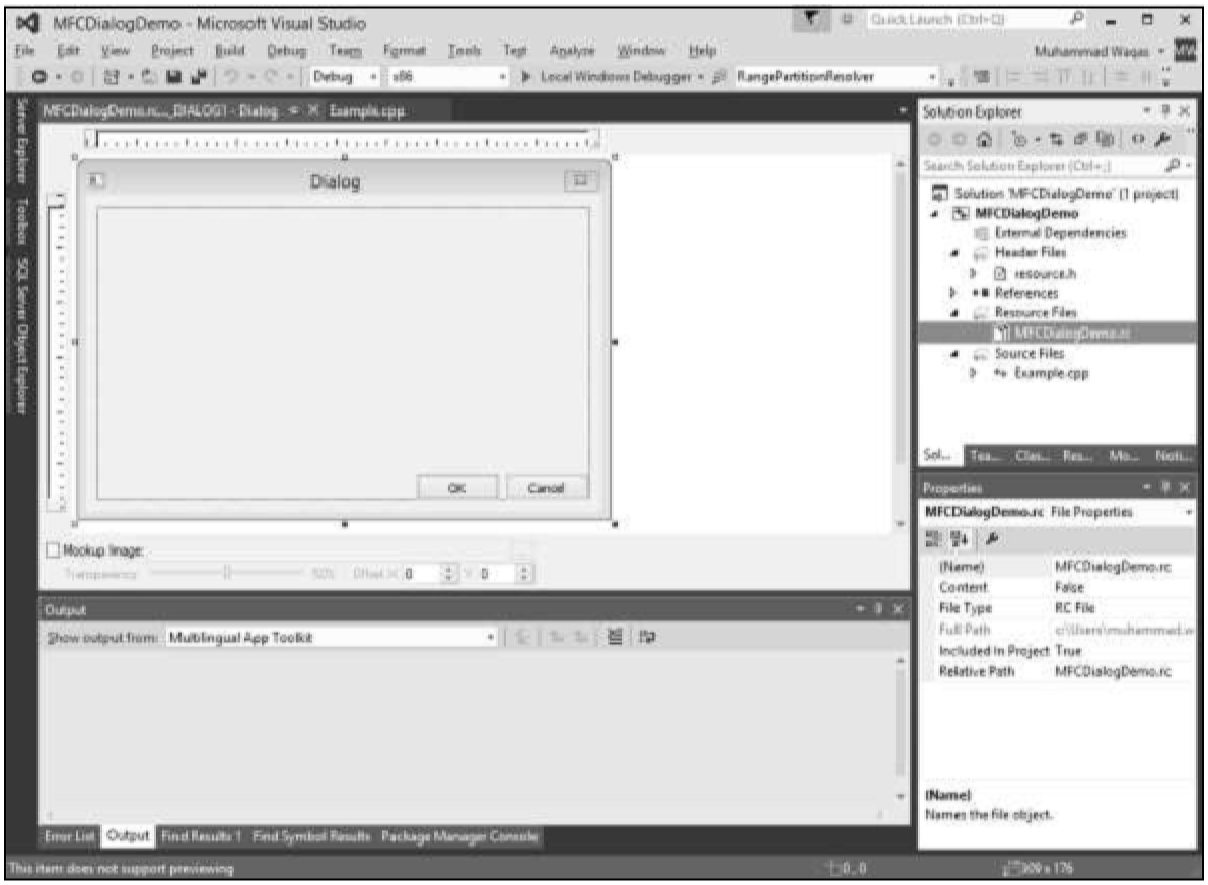


Step 2 – In the Add Resource dialog box, select Dialog and click New.

Step 3 – A dialog box requires some preparation before actually programmatically creating it.

Step 4 – A dialog box can first be manually created as a text file *inaresourcefile*.

Step 5 – You can now see the MFCDialogDemo.rc file created under Resource Files.

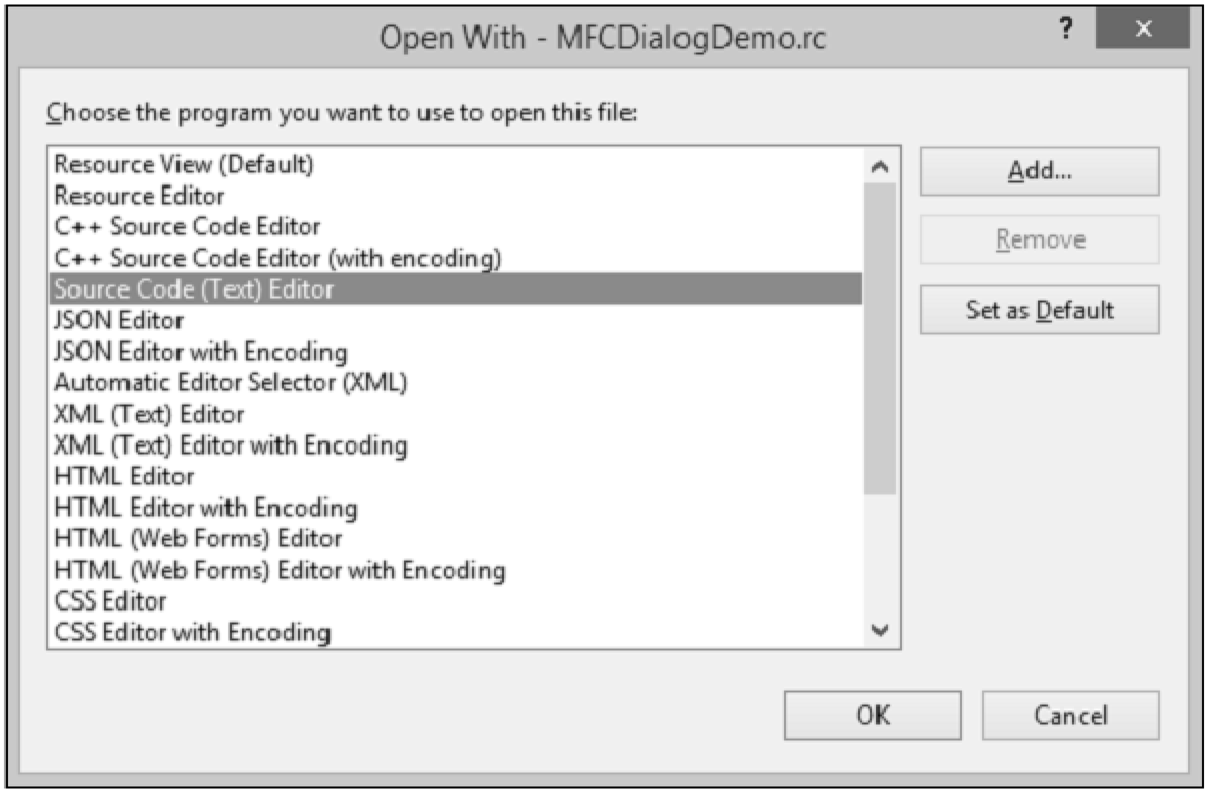

MFCDialogDemo - Microsoft Visual Studio
File Edit View Project Build Debug Team Format Tools Test Analyze Window Help
Debug x86 Local Windows Debugger RangePartitionResolver
MFCDialogDemo.rc..._DIALOG1 - Dialog Example.cpp
Dialog
OK Cancel
Mockup Image
Solution Explorer
Solution 'MFCDialogDemo' (1 project)
MFCDialogDemo
External Dependencies
Header Files
resource.h
References
Resource Files
MFCDialogDemo.rc
Source Files
Example.cpp
Properties
Output
Show output from: Multilingual App Toolkit
Error List Output Find Results 1 Find Symbol Results Package Manager Console
This item does not support previewing

MFCDialogDemo.rc File Properties

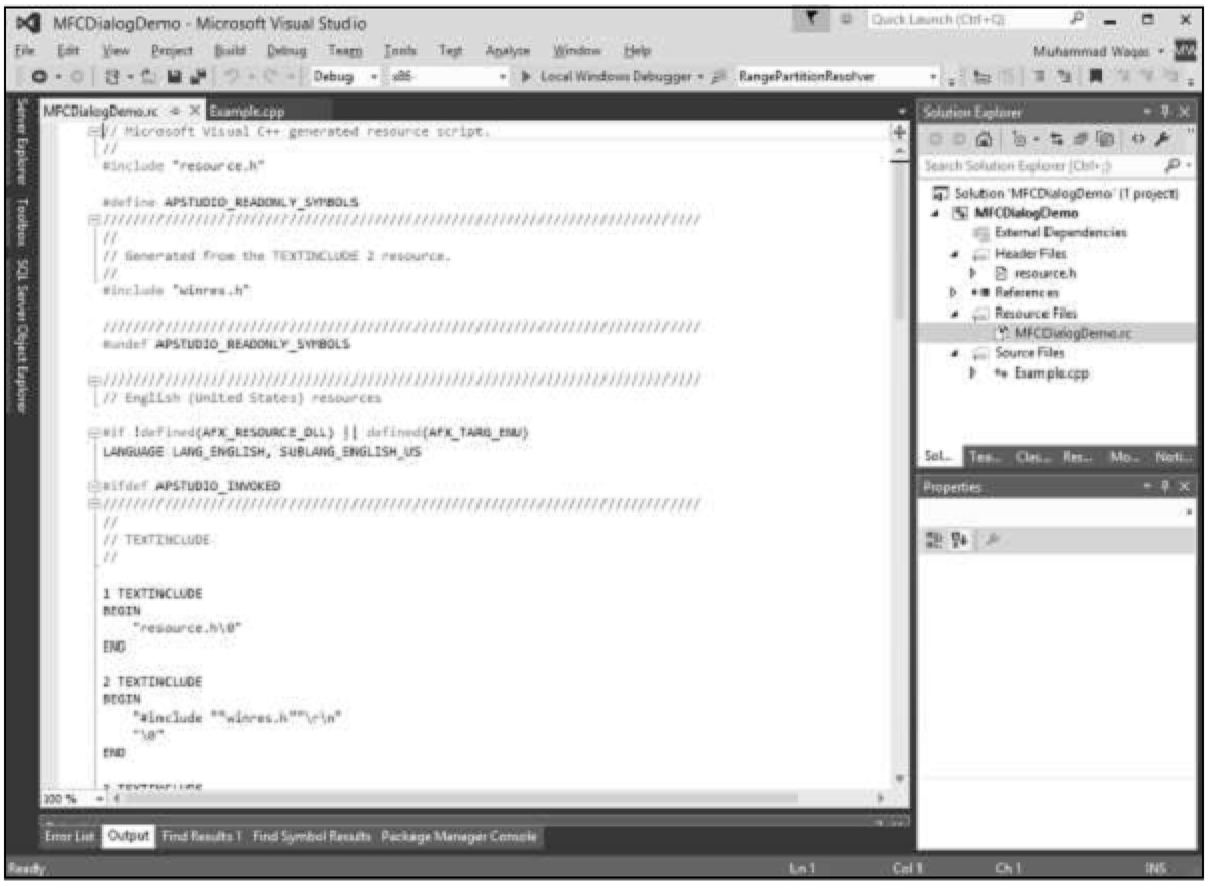
(Name)	MFCDialogDemo.rc
Content	False
File Type	RC File
Full Path	c:\Users\muhammad.w
Included In Project	True
Relative Path	MFCDialogDemo.rc

(Name)
Names the file object.

Step 6 – The resource file is open in designer. The same can be opened as a text file. Rightclick on the resource file and select Open With.

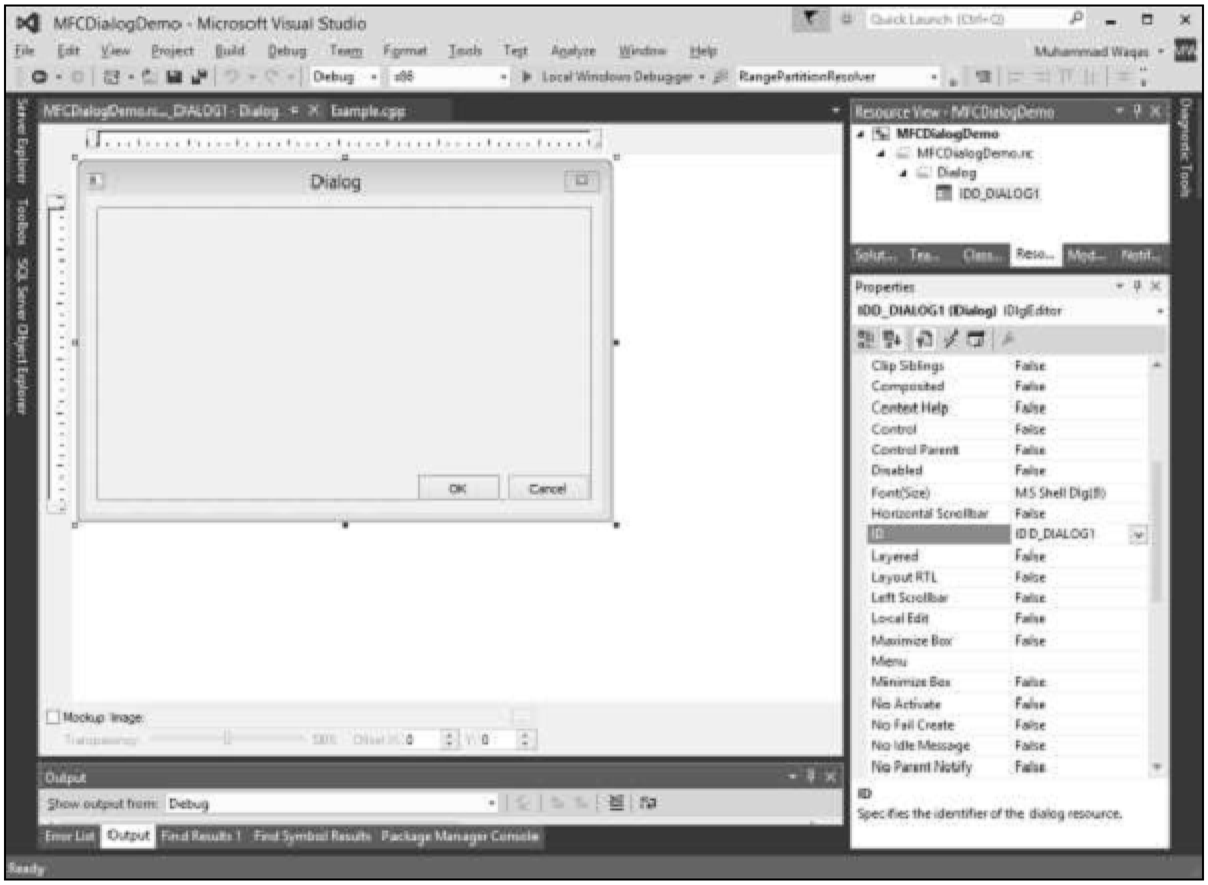

Open With - MFCDialogDemo.rc
?
×
Choose the program you want to use to open this file:
Resource View (Default)
Resource Editor
C++ Source Code Editor
C++ Source Code Editor (with encoding)
Source Code (Text) Editor
JSON Editor
JSON Editor with Encoding
Automatic Editor Selector (XML)
XML (Text) Editor
XML (Text) Editor with Encoding
HTML Editor
HTML Editor with Encoding
HTML (Web Forms) Editor
HTML (Web Forms) Editor with Encoding
CSS Editor
CSS Editor with Encoding
Add...
Remove
Set as Default
OK
Cancel

Step 7 – Select the Source Code $Text$ editor and click Add button.


MFCDialogDemo - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug x86 Local Windows Debugger RangePartitionResolver
MFCDialogDemo.rc Example.cpp
Solution Explorer
Solution 'MFCDialogDemo' (1 project)
MFCDialogDemo
External Dependencies
Header Files
resource.h
References
Resource Files
MFCDialogDemo.rc
Source Files
Example.cpp
Properties
Error List Output Find Results 1 Find Symbol Results Package Manager Console
Ready Ln 1 Col 1 Ch 1 INS

```
// Microsoft Visual C++ generated resource script.
//
#include "resource.h"

#define APSTUDIO_READONLY_SYMBOLS
/////////////////////////////////////////////////////////////////////////////
//
// Generated from the TEXTINCLUDE 2 resource.
//
#include "winres.h"

/////////////////////////////////////////////////////////////////////////////
#undef APSTUDIO_READONLY_SYMBOLS

/////////////////////////////////////////////////////////////////////////////
// English (United States) resources

#if !defined(AFX_RESOURCE_DLL) || defined(AFX_TARG_ENU)
LANGUAGE LANG_ENGLISH, SUBLANG_ENGLISH_US

#ifdef APSTUDIO_INVOKED
/////////////////////////////////////////////////////////////////////////////
//
// TEXTINCLUDE
//

1 TEXTINCLUDE
BEGIN
    "resource.h\0"
END

2 TEXTINCLUDE
BEGIN
    "#include ""winres.h""\r\n"
    "\0"
END
```

Step 8 – Go back to the designer and right-click on the dialog and select Properties.


MFCDialogDemo - Microsoft Visual Studio
File Edit View Project Build Debug Team Format Tools Test Analyze Window Help
Debug x86 Local Windows Debugger RangePartitionResolver
MFCDialogDemo.rc...IDIALOG1 - Dialog Example.cpp
Dialog
OK
Cancel
Resource View - MFCDialogDemo
MFCDialogDemo
MFCDialogDemo.rc
Dialog
IDD_DIALOG1
Output
Show output from: Debug
Error List Output Find Results 1 Find Symbol Results Package Manager Console
Ready

Properties

IDD_DIALOG1 (Dialog) IDlgEditor

Clip Siblings	False
Composited	False
Context Help	False
Control	False
Control Parent	False
Disabled	False
Font(Size)	MS Shell Dlg(8)
Horizontal Scrollbar	False
ID	IDD_DIALOG1
Layered	False
Layout RTL	False
Left Scrollbar	False
Local Edit	False
Maximize Box	False
Menu	
Minimize Box	False
No Activate	False
No Fail Create	False
No Idle Message	False
No Parent Notify	False

ID
Specifies the identifier of the dialog resource.

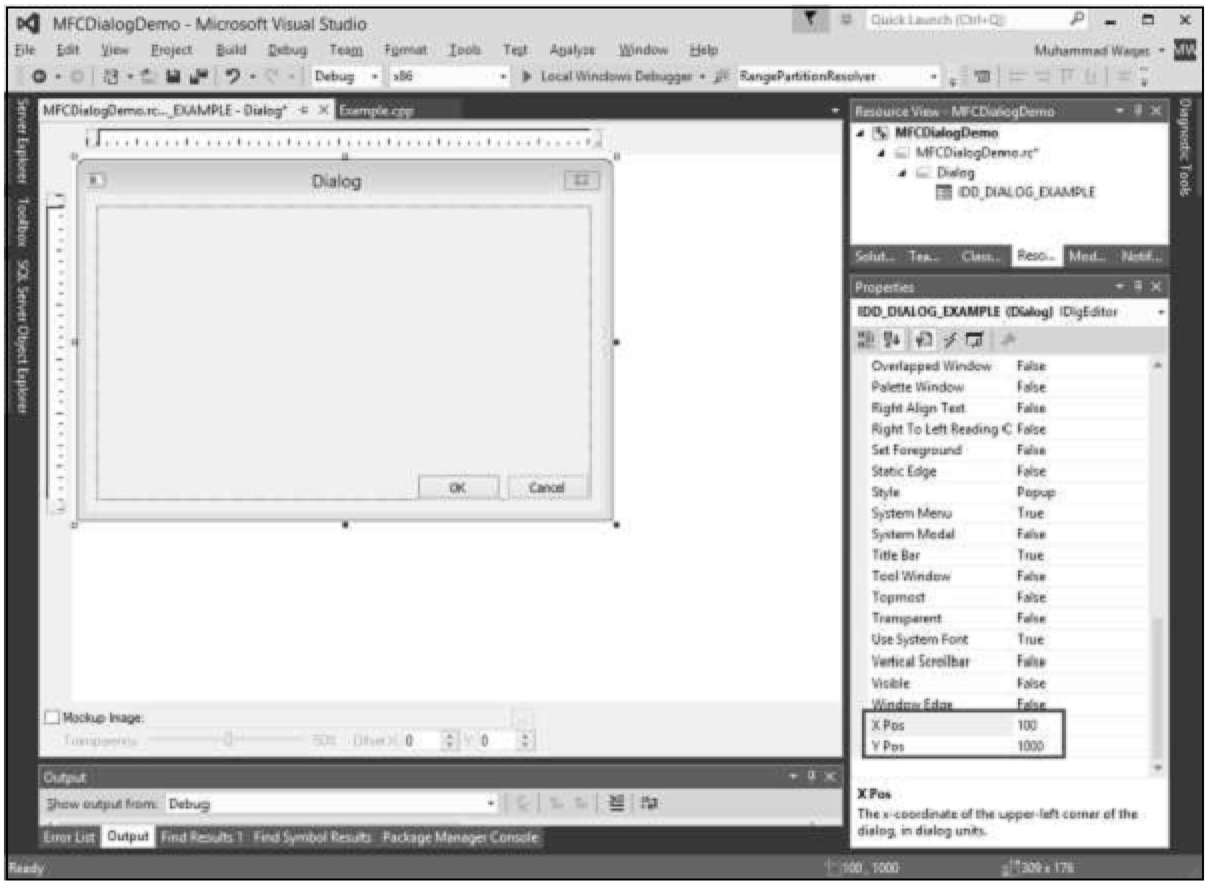
Step 9 – You need to choose out of the many options.

Step 10 – Like most other controls, a dialog box must be identified. The identifier *ID* of a dialog box usually starts with IDD_, Let us change the ID to IDD_EXAMPLE_DLG.

Dialog Location

A dialog box must be "physically" located on an application. Because a dialog box is usually created as a parent to other controls, its location depends on its relationship to its parent window or to the desktop.

If you look and the Properties window, you see two fields, X Pos and Y Pos.


MFCDialogDemo - Microsoft Visual Studio
File Edit View Project Build Debug Team Format Tools Test Analyze Window Help
Debug x86 Local Windows Debugger RangePartitionResolver
MFCDialogDemo.rc..._EXAMPLE - Dialog* Example.cpp
Dialog
OK Cancel
Resource View - MFCDialogDemo
MFCDialogDemo
MFCDialogDemo.rc*
Dialog
IDD_DIALOG_EXAMPLE
Output
Show output from: Debug
Error List Output Find Results 1 Find Symbol Results Package Manager Console
Ready

Properties

IDD_DIALOG_EXAMPLE (Dialog) IDlgEditor

Overlapped Window	False
Palette Window	False
Right Align Text	False
Right To Left Reading C	False
Set Foreground	False
Static Edge	False
Style	Popup
System Menu	True
System Modal	False
Title Bar	True
Tool Window	False
Topmost	False
Transparent	False
Use System Font	True
Vertical Scrollbar	False
Visible	False
Window Edge	False
X Pos	100
Y Pos	1000

X Pos
The x-coordinate of the upper-left corner of the dialog, in dialog units.

100, 1000 309 x 176

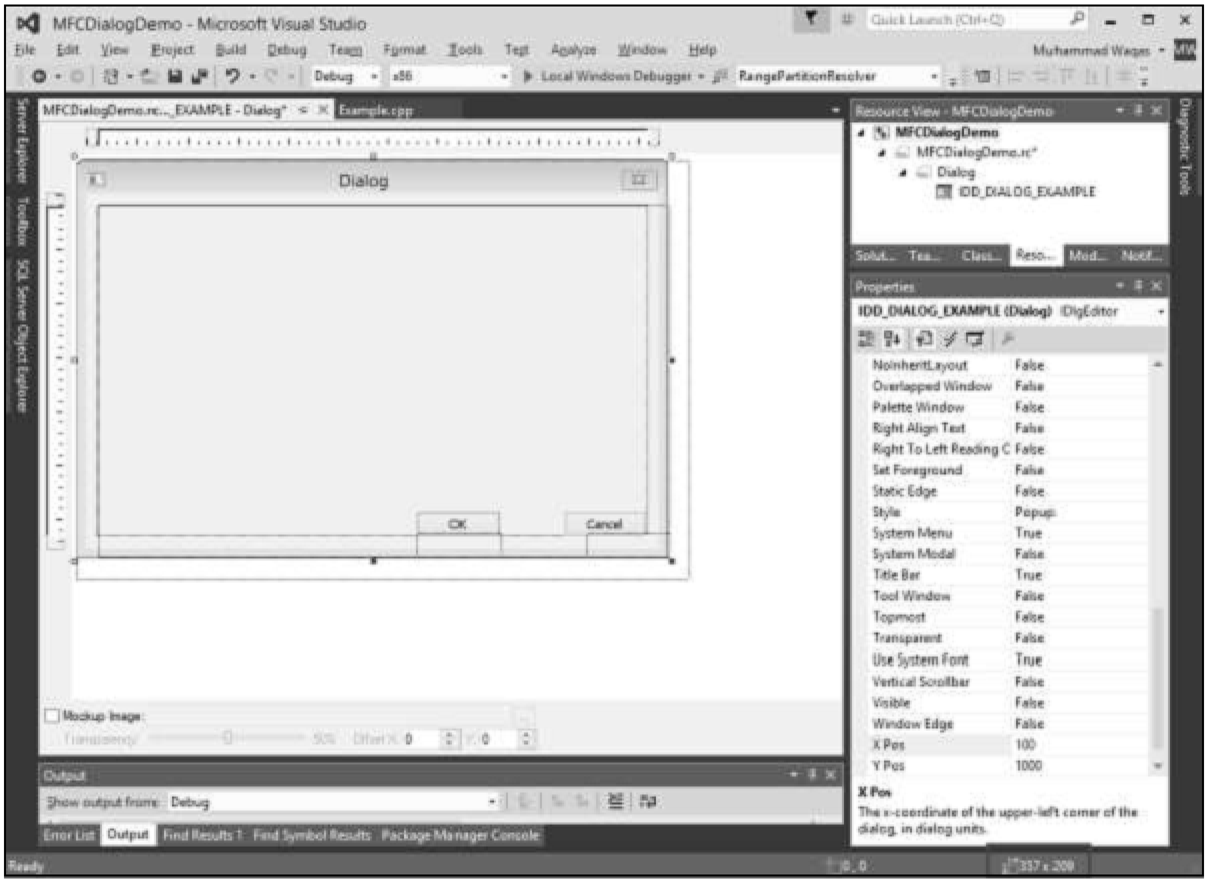
- X is the distance from the left border of the monitor to the left border of the dialog box.
- Y is the distance from the top border of the monitor to the top border of the dialog box.

By default, these fields are set to zero. You can also change as shown above.

If you specify these two dimensions as 0, the left and top borders of the dialog box would be set so the object appears in the center-middle of the screen.

Dialog Box Dimensions

The dimensions of a dialog box refer to its width and its height. You can resize the width and height with the help of mouse in designer window.


MFCDialogDemo - Microsoft Visual Studio
File Edit View Project Build Debug Team Format Tools Test Analyze Window Help
Debug x86 Local Windows Debugger RangePartitionResolver
MFCDialogDemo.rc..._EXAMPLE - Dialog*
Example.cpp
Dialog
OK
Cancel
Resource View - MFCDialogDemo
MFCDialogDemo
MFCDialogDemo.rc*
Dialog
IDD_DIALOG_EXAMPLE
Output
Show output from: Debug
Error List Output Find Results 1 Find Symbol Results Package Manager Console
Ready
0, 0
337 x 209

Properties

IDD_DIALOG_EXAMPLE (Dialog) IDlgEditor

NoInheritLayout	False
Overlapped Window	False
Palette Window	False
Right Align Text	False
Right To Left Reading C	False
Set Foreground	False
Static Edge	False
Style	Popup
System Menu	True
System Modal	False
Title Bar	True
Tool Window	False
Topmost	False
Transparent	False
Use System Font	True
Vertical Scrollbar	False
Visible	False
Window Edge	False
X Pos	100
Y Pos	1000

X Pos
The x-coordinate of the upper-left corner of the dialog, in dialog units.

You can see the changes in width and height on the Status Bar.

Dialog Box Methods

The base class used for displaying dialog boxes on the screen is CDialog class. To create a dialog box, we need to derive a class from CDialog. The CDialog class itself provides three constructors which are as follows –

```
CDialog();
CDialog(UINT nIDTemplate, CWnd* pParentWnd = NULL);
CDialog(LPCTSTR lpszTemplateName, CWnd* pParentWnd = NULL);
```

Let us create another class CExampleDlg and derive it from CDialog. We will implement its default constructor destructor as shown in the following code.

```
class CExampleDlg : public CDialog {
   public:
      enum { IDD = IDD_EXAMPLE_DLG };

      CExampleDlg();
      ~CExampleDlg();
};

CExampleDlg::CExampleDlg():CDialog(CExampleDlg::IDD) {

}
```

```
CExampleDlg::~CExampleDlg() {

}
```

We need to instantiate this dialog on CExample::InitInstance method as shown in the following code.

```
BOOL CExample::InitInstance() {
   CExampleDlg myDlg;
   m_pMainWnd = &myDlg;

   return TRUE;
}
```

Modal Dialog Boxes

There are two types of dialog boxes – **modeless** and **modal**. Modal and modeless dialog boxes differ by the process used to create and display them.

Modeless Dialog Box

- For a modeless dialog box, you must provide your own public constructor in your dialog class.
- To create a modeless dialog box, call your public constructor and then call the dialog object's Create member function to load the dialog resource.
- You can call Create either during or after the constructor call. If the dialog resource has the property WS_VISIBLE, the dialog box appears immediately.
- If not, you must call its ShowWindow member function.

Modal Dialog

- To create a modal dialog box, call either of the two public constructors declared in CDialog.
- Next, call the dialog object's **DoModal** member function to display the dialog box and manage interaction with it until the user chooses OK or Cancel.
- This management by DoModal is what makes the dialog box modal. For modal dialog boxes, DoModal loads the dialog resource.

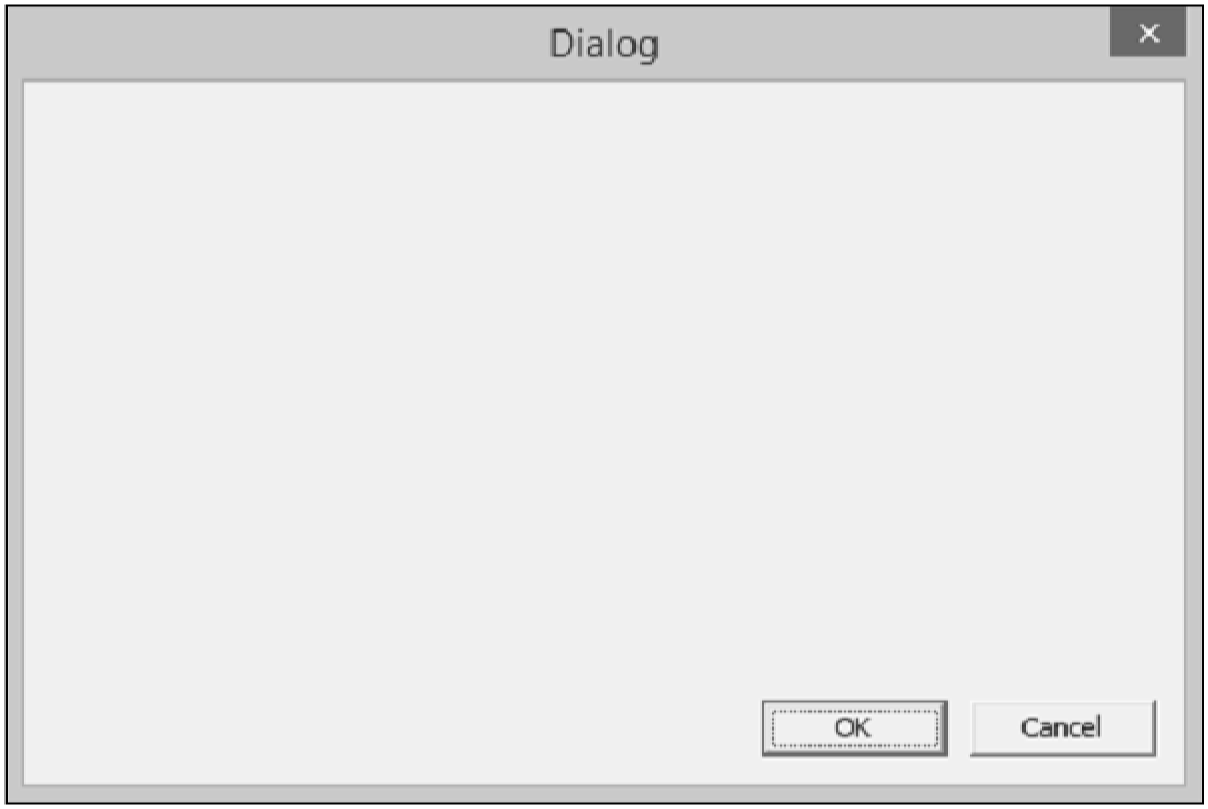
Step 1 – To display the dialog box as modal, in the CExample::InitInstance event call the DoModal method using your dialog variable –

```
BOOL CExample::InitInstance() {
   CExampleDlg myDlg;
   m_pMainWnd = &myDlg;
   myDlg.DoModal();
   return TRUE;
}
```

Step 2 – Here is the complete implementation of Example.cpp file.

```
#include <afxwin.h>
#include "resource.h"

class CExample : public CWinApp {
   public:
      BOOL InitInstance();
};

class CExampleDlg : public CDialog {
   public:
      enum { IDD = IDD_EXAMPLE_DLG };

      CExampleDlg();
     ~CExampleDlg();
};

CExampleDlg::CExampleDlg():CDialog(CExampleDlg::IDD) {

}

CExampleDlg::~CExampleDlg() {

}

BOOL CExample::InitInstance() {
   CExampleDlg myDlg;
   m_pMainWnd = &myDlg;
   myDlg.DoModal();
   return TRUE;
}
CExample MyApp;
```

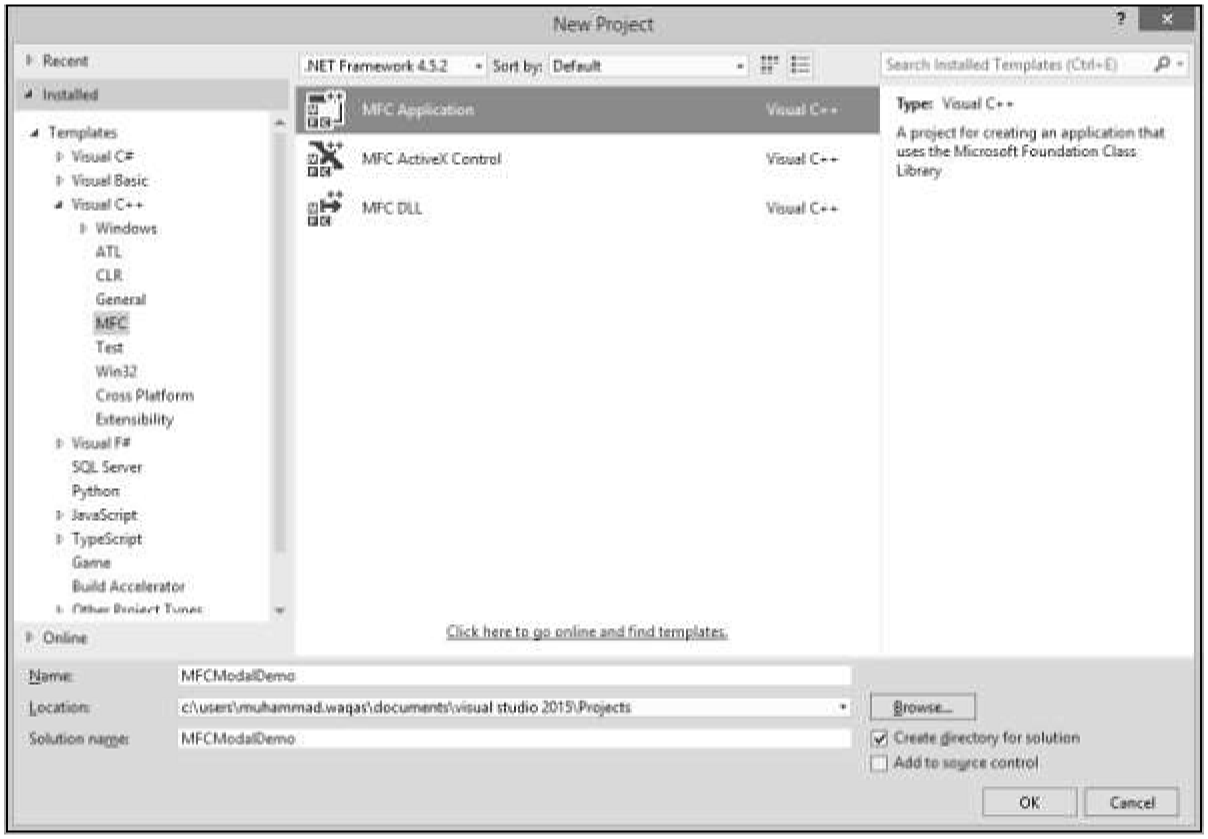
Step 3 – When the above code is compiled and executed, you will see the following dialog box.



Dialog-Based Applications

Microsoft Visual Studio provides an easier way to create an application that is mainly based on a dialog box. Here are the steps to create a dialog base project using project templates available in Visual Studio –

Step 1 – Open the Visual studio and click on the File → New → Project menu option. You can see the New Project dialog box.



Type: Visual C++

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

Name: MFCModalDemo

Location: c:\users\muhammad.waqas\documents\visual studio 2015\Projects

Solution name: MFCModalDemo

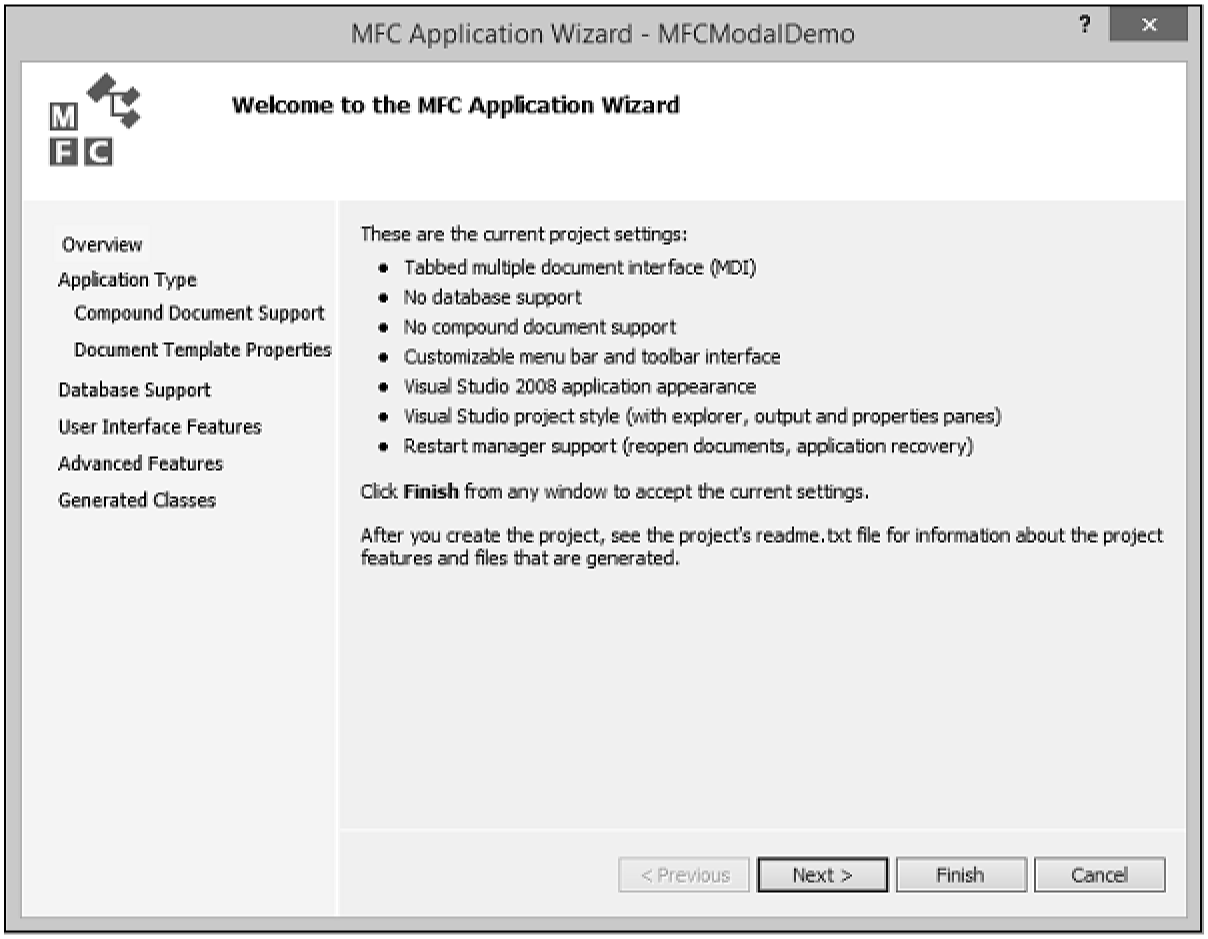
[✓] Create directory for solution

[ ] Add to source control

Step 2 – From the left pane, select Templates → Visual C++ → MFC.

Step 3 – In the middle pane, select MFC Application.

Step 4 – Enter project name 'MFCModalDemo' in the Name field and click OK to continue. You will see the following dialog box.



Welcome to the MFC Application Wizard

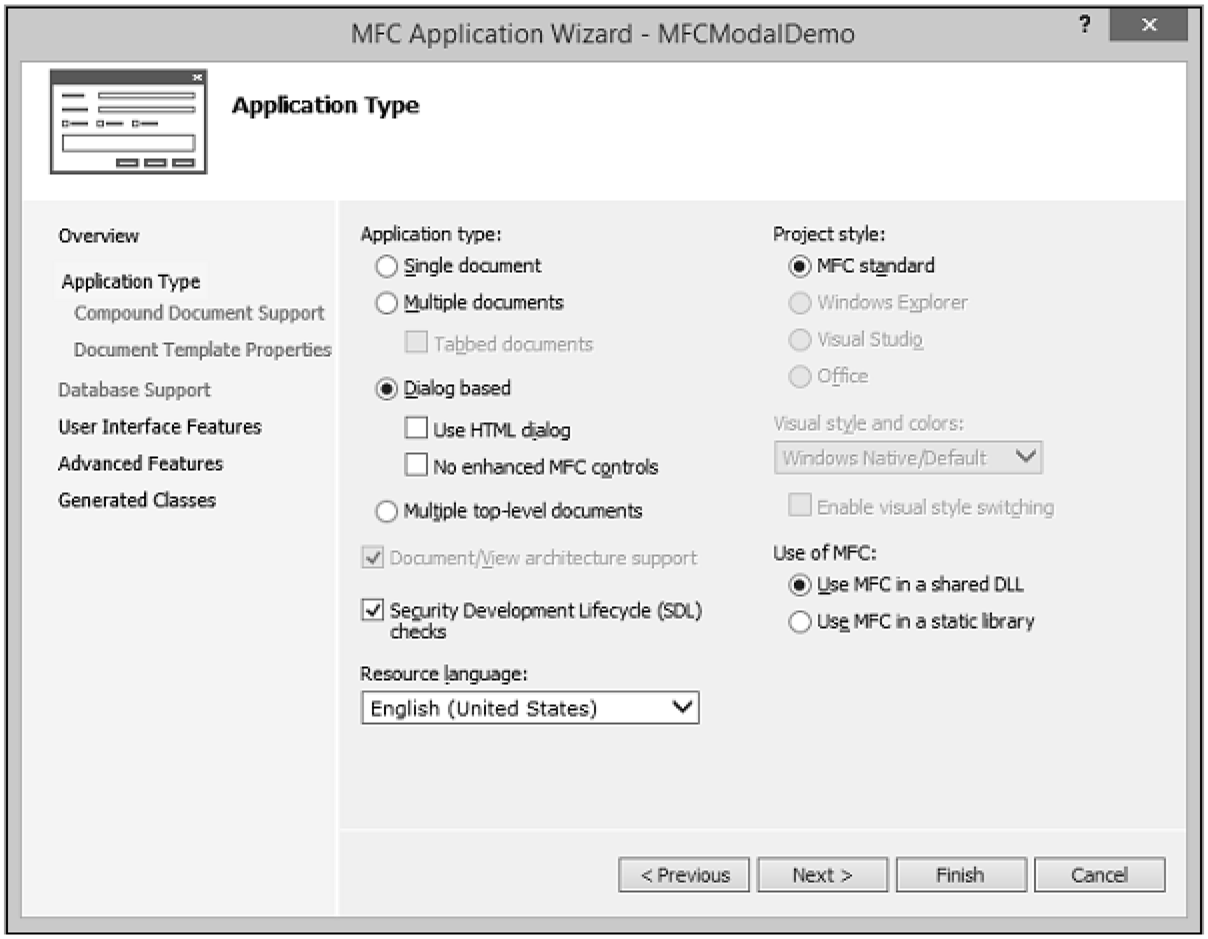
These are the current project settings:

- Tabbed multiple document interface (MDI)
- No database support
- No compound document support
- Customizable menu bar and toolbar interface
- Visual Studio 2008 application appearance
- Visual Studio project style (with explorer, output and properties panes)
- Restart manager support (reopen documents, application recovery)

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.

Step 5 – Click Next.



Application type:
- ( ) Single document
- ( ) Multiple documents
  - [ ] Tabbed documents
- (●) Dialog based
  - [ ] Use HTML dialog
  - [ ] No enhanced MFC controls
- ( ) Multiple top-level documents

[✓] Document/View architecture support

[✓] Security Development Lifecycle (SDL) checks

Resource language: English (United States)

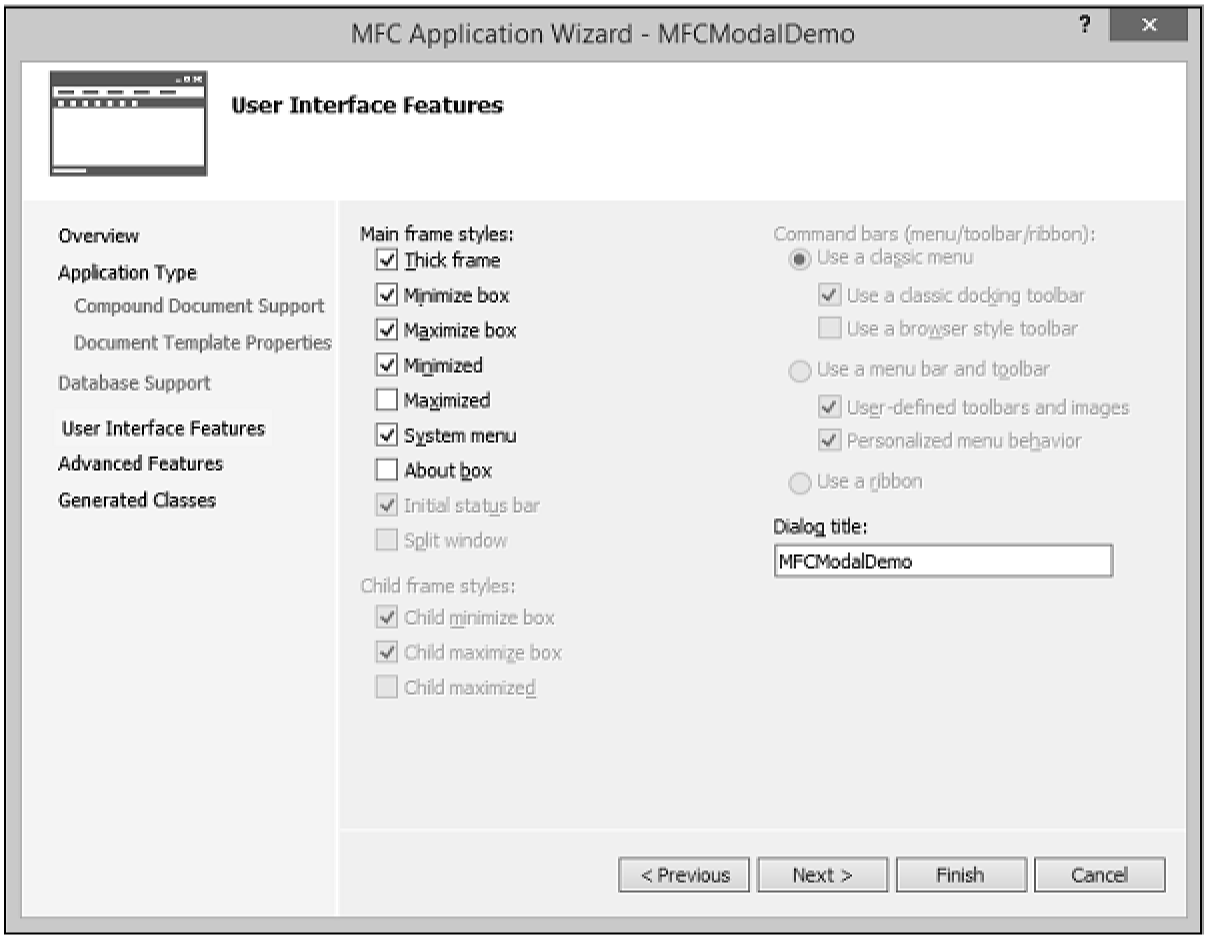
Project style:
- (●) MFC standard
- ( ) Windows Explorer
- ( ) Visual Studio
- ( ) Office

Visual style and colors: Windows Native/Default

[ ] Enable visual style switching

Use of MFC:
- (●) Use MFC in a shared DLL
- ( ) Use MFC in a static library

Step 6 – Select the options shown in the above dialog box and click Next.



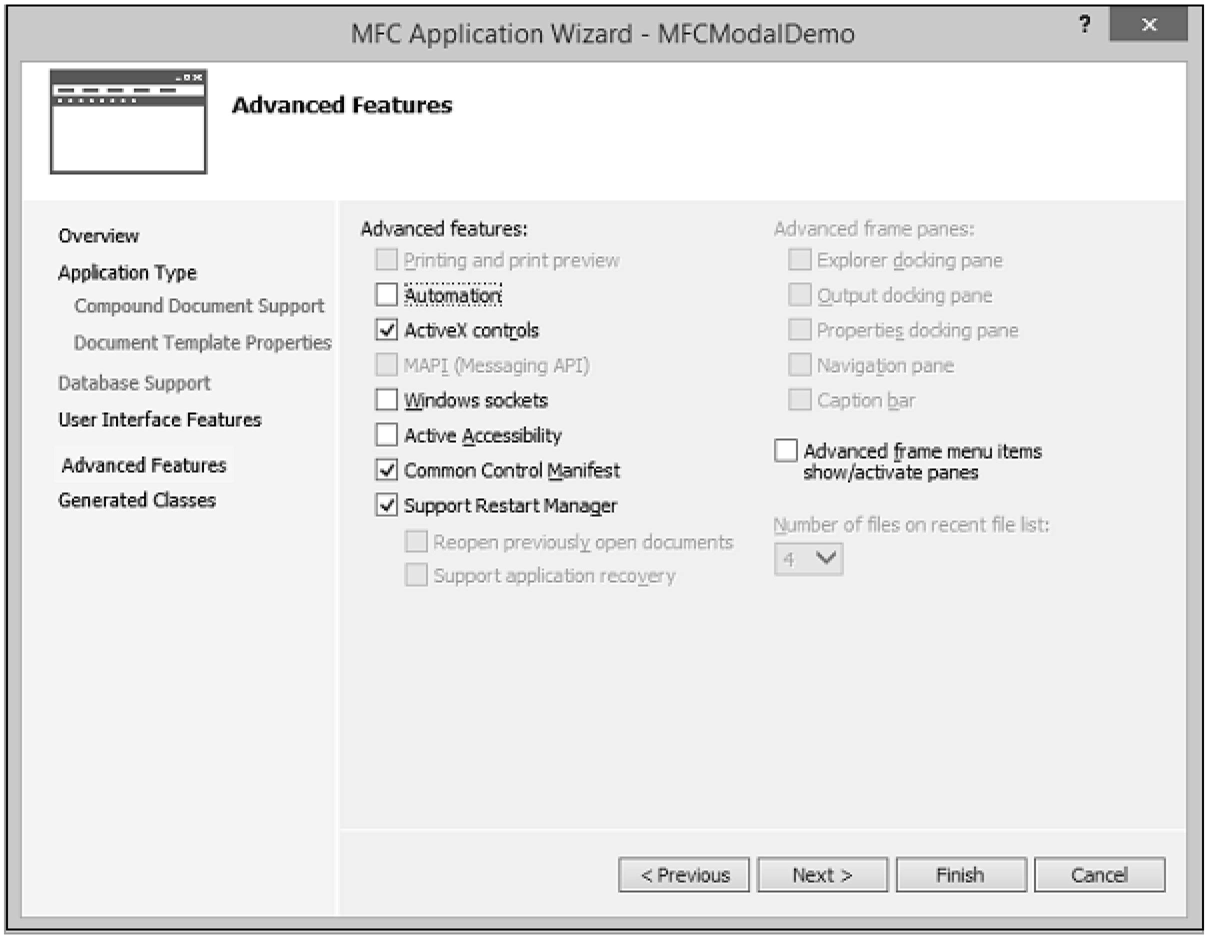
Main frame styles:
- [✓] Thick frame
- [✓] Minimize box
- [✓] Maximize box
- [✓] Minimized
- [ ] Maximized
- [✓] System menu
- [ ] About box
- [✓] Initial status bar
- [ ] Split window

Child frame styles:
- [✓] Child minimize box
- [✓] Child maximize box
- [ ] Child maximized

Command bars (menu/toolbar/ribbon):
- ● Use a classic menu
  - [✓] Use a classic docking toolbar
  - [ ] Use a browser style toolbar
- ○ Use a menu bar and toolbar
  - [✓] User-defined toolbars and images
  - [✓] Personalized menu behavior
- ○ Use a ribbon

Dialog title: MFCModalDemo

Step 7 – Check all the options that you choose to have on your dialog box like Maximize and Minimize Boxes and click Next.



Advanced features:

- [ ] Printing and print preview
- [ ] Automation
- [x] ActiveX controls
- [ ] MAPI (Messaging API)
- [ ] Windows sockets
- [ ] Active Accessibility
- [x] Common Control Manifest
- [x] Support Restart Manager
  - [ ] Reopen previously open documents
  - [ ] Support application recovery

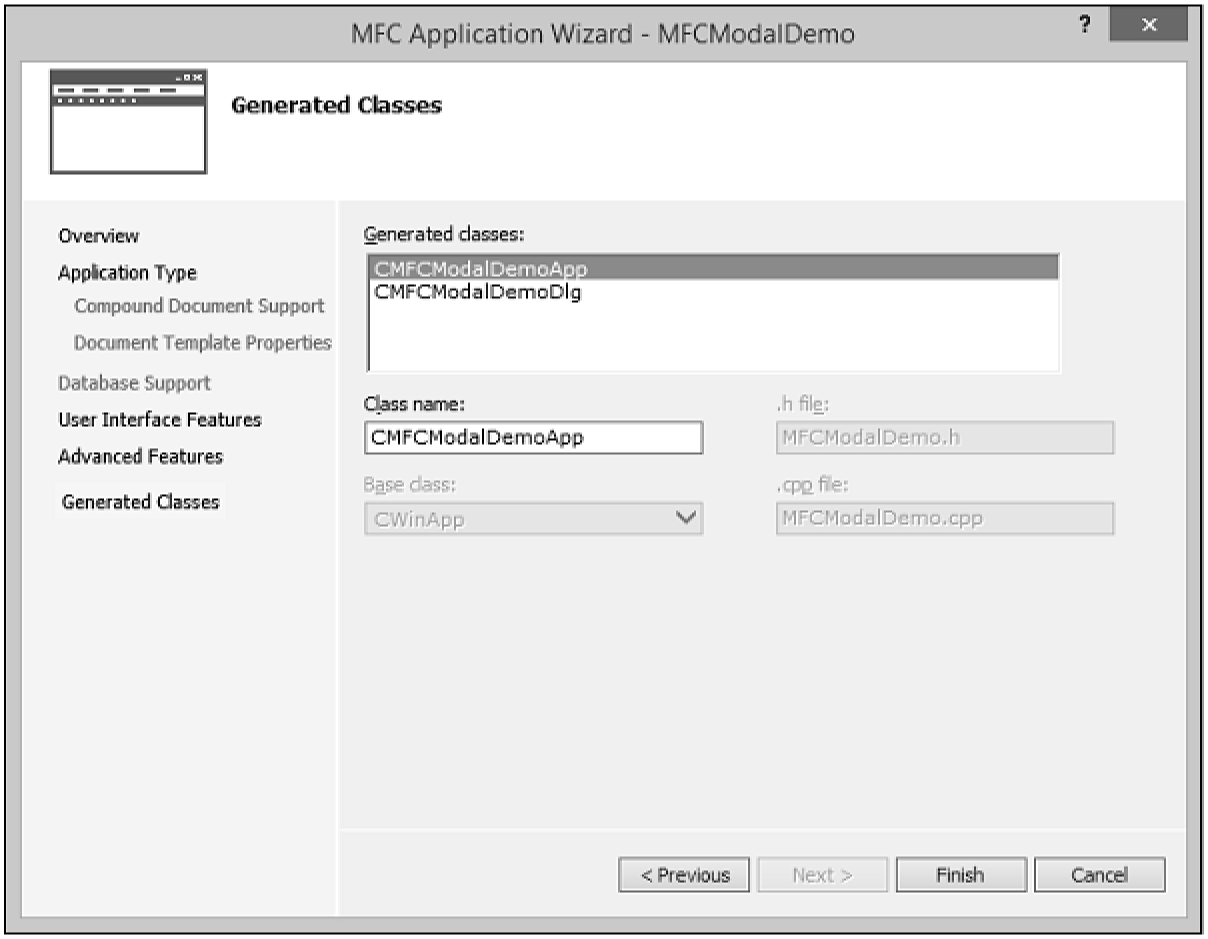
Advanced frame panes:

- [ ] Explorer docking pane
- [ ] Output docking pane
- [ ] Properties docking pane
- [ ] Navigation pane
- [ ] Caption bar

- [ ] Advanced frame menu items show/activate panes

Number of files on recent file list: 4

Step 8 – Click Next.



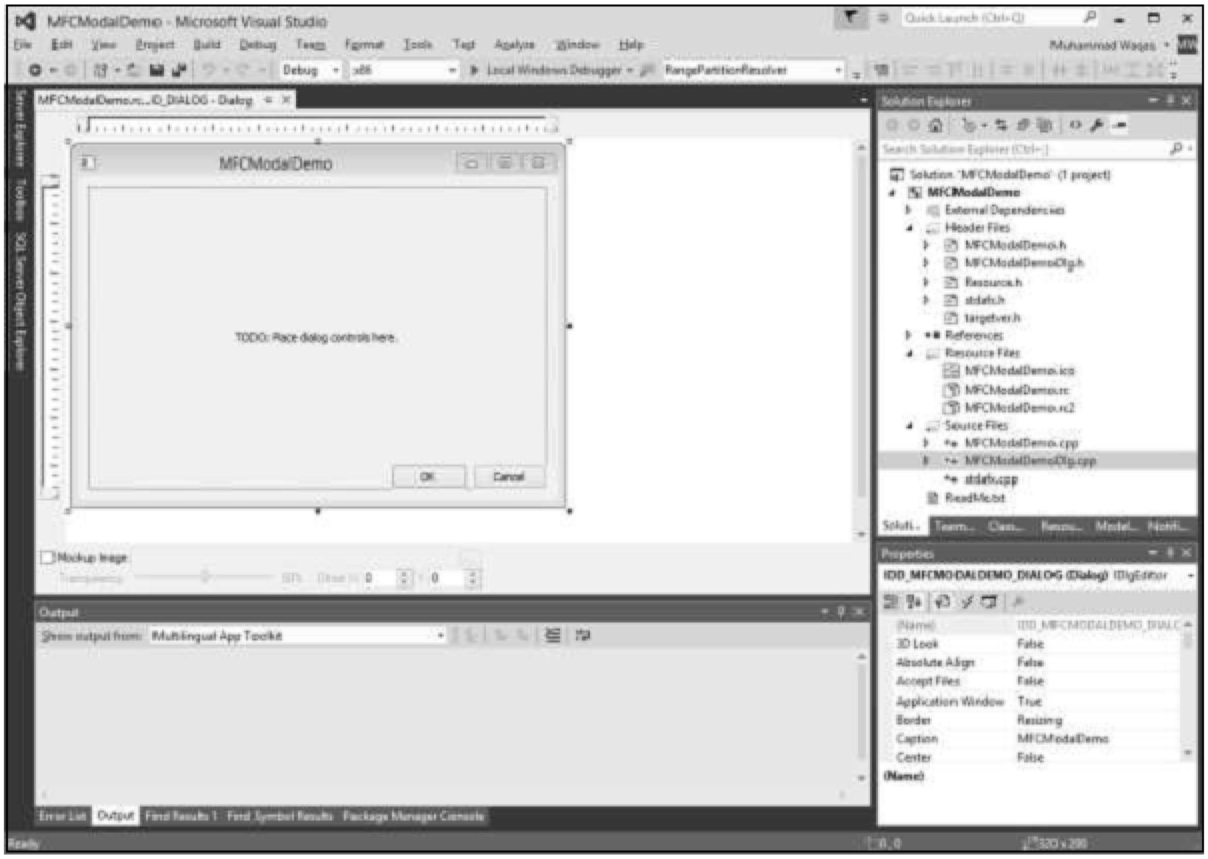
Generated Classes

Generated classes:
CMFCModalDemoApp
CMFCModalDemoDlg

Class name: CMFCModalDemoApp
.h file: MFCModalDemo.h
Base class: CWinApp
.cpp file: MFCModalDemo.cpp

Step 9 – It will generate these two classes. You can change the name of the classes and click Finish.

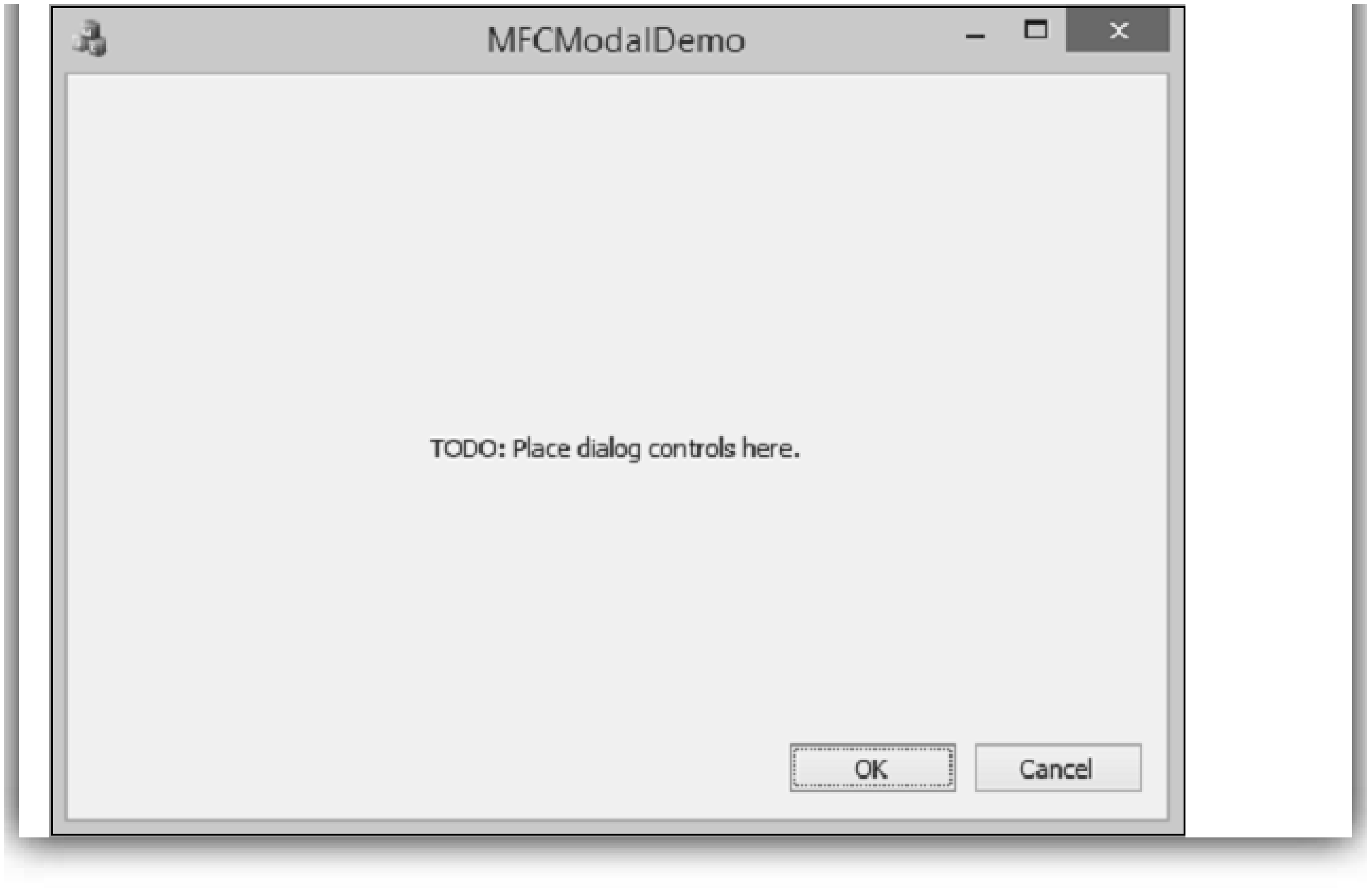
Step 10 – You can now see that the MFC wizard creates this Dialog Box and the project files by default.


MFCModalDemo - Microsoft Visual Studio
File Edit View Project Build Debug Team Format Tools Test Analyze Window Help
Debug x86 Local Windows Debugger RangePartitionResolver
MFCModalDemo.rc...D_DIALOG - Dialog
MFCModalDemo
TODO: Place dialog controls here.
OK
Cancel
Solution Explorer
Solution 'MFCModalDemo' (1 project)
MFCModalDemo
External Dependencies
Header Files
MFCModalDemo.h
MFCModalDemoDlg.h
Resource.h
stdafx.h
targetver.h
References
Resource Files
MFCModalDemo.ico
MFCModalDemo.rc
MFCModalDemo.rc2
Source Files
MFCModalDemo.cpp
MFCModalDemoDlg.cpp
stdafx.cpp
ReadMe.txt
Output
Show output from: Multilingual App Toolkit
Error List Output Find Results 1 Find Symbol Results Package Manager Console
Ready

Properties
IDD_MFCMODALDEMO_DIALOG (Dialog) IDlgEditor

(Name)	IDD_MFCMODALDEMO_DIALOG
3D Look	False
Absolute Align	False
Accept Files	False
Application Window	True
Border	Resizing
Caption	MFCModalDemo
Center	False

Step 11 – When you run this application, you will see the following output.


MFCModalDemo
TODO: Place dialog controls here.
OK
Cancel