

JULIA CASE BRADLEY
ANITA C. MILLSPAUGH

**PROGRAMMING IN
VISUAL BASIC 2010**

Chapter 14

**Additional Topics in
Visual Basic**



Objectives

- Validate user input in the Validating event and display messages using an [ErrorProvider component](#).
- Use [code snippets](#) in the editor.
- Create a multiple document project with parent and [child forms](#).
- Arrange the child forms vertically, horizontally, or cascaded.
- Add [toolbars](#) and [status bars](#) to your forms using tool strip and status strip controls.
- Use calendar controls and date functions.
- Display a Web page on a Windows form using a [WebBrowser control](#).
- Capture and check an individual keypress from the user.



Advanced Validation Techniques

- ErrorProvider components
 - Share some characteristics with Web validation controls.
- Useful controls include:
 - Set MaxLength and/or
 - CharacterCasing properties of text boxes
 - Perform **field-level validation** using Validating event of input controls.



The ErrorProvider Component

- ErrorProvider component causes an error message to appear next to the field in error on the form rather than pop-up messages in message boxes.
- Generally, one ErrorProvider can be used to validate all controls on a form.
 - Added to the component tray
- If data value is invalid, a blinking icon displays next to the field in error and a message displays in a pop-up (similar to a ToolTip).



ErrorProvider.SetError Method

- General Form

```
ErrorProviderObject.SetError(ControlName, MessageString)
```

- Examples

```
ErrorProvider1.SetError(QuantityTextBox, "Quantity must be numeric.")  
ErrorProvider1.SetError(CreditCardTextBox, "Required field.")
```



The MaxLength and CharacterCasing Properties

- Helps user to enter correct input data
- **MaxLength property**
 - Set smaximum number of characters that can be entered, and beeps and holds insertion point to indicate error



- **CharacterCasing property**
 - Converts each character entered to Normal, Upper, or Lower case (default is Normal)



Field-Level Validation

- Displays any error message as soon as the user attempts to leave a field with invalid data
- To accomplish field-level validation use
 - Validating event
 - CausesValidation property
 - ErrorProvider components



Using the Validating Event and CausesValidation Property

- Validation event is best location for validation code.
 - Use `CancelEventArgs` argument to cancel the event and return focus to the control.
- Each control on a form has a `CausesValidation` property set to `True` by default.
 - When focus passes from one control to another, the validating event occurs for the control just left.
 - Set `CausesValidation` to `False` on a control such as `Cancel` or `Exit` to give the user a way to bypass the validation.



Capturing Keystrokes from the User

- Check for the key that the user entered in a control's KeyDown, KeyPress, or KeyUp event procedure.
 - These events occur in the order listed for most keyboard keys with the following exception:
 - Keystrokes that ordinarily cause an action to occur, such as the Tab key or Enter key, generate only a KeyUp event.
 - The e argument of the KeyPress event procedure is KeyPressEventArgs.
 - Has a KeyChar property that holds the character pressed
 - Also has Handled property which can be set to True—that the keystroke needs no further processing; the action effectively “throws away” the keystroke just entered.
-



Using the Masked Text Box for Validation

- Set the Mask property of a masked text box to any predefined masks, or users can write their own.
- Easiest way to create a new one is to modify one of the existing masks.

--OR--

- Follow the syntax rules of a regular expression.
- Predefined masks include date, time, phone number, Social Security number, and ZIP code format.



Code Snippets

- Small samples of code that can show users how to accomplish many programming tasks
- Snippet categories include:
 - Collections, Data Types — defined by VB, Interacting with the Application, Maintaining Collections, File System-Processing Drives, Folders and Files, Math, Security, and VB language
 - The above list varies depending upon the location on the insertion.



Sample Projects

- Visual Studio includes many sample projects (all editions except the Express Edition) that can be used to learn new techniques.
- From the Help menu, select Contents; expand the nodes for Development Tools and Languages/Visual Studio/Visual Basic to find the Samples node.
- The Quick Starts are another avenue for pursuing the study of VB.
 - Provide tutorials with step-by-step introduction to many techniques and controls



Multiple Document Interface (1 of 2)

- **SDI** = Single document interface
 - Each form in the project acts independently from the other forms .
- **MDI** = Multiple document interface
 - An example of a MDI application is Word.
 - Word has a **parent form** (the main window) and child forms (each document window).



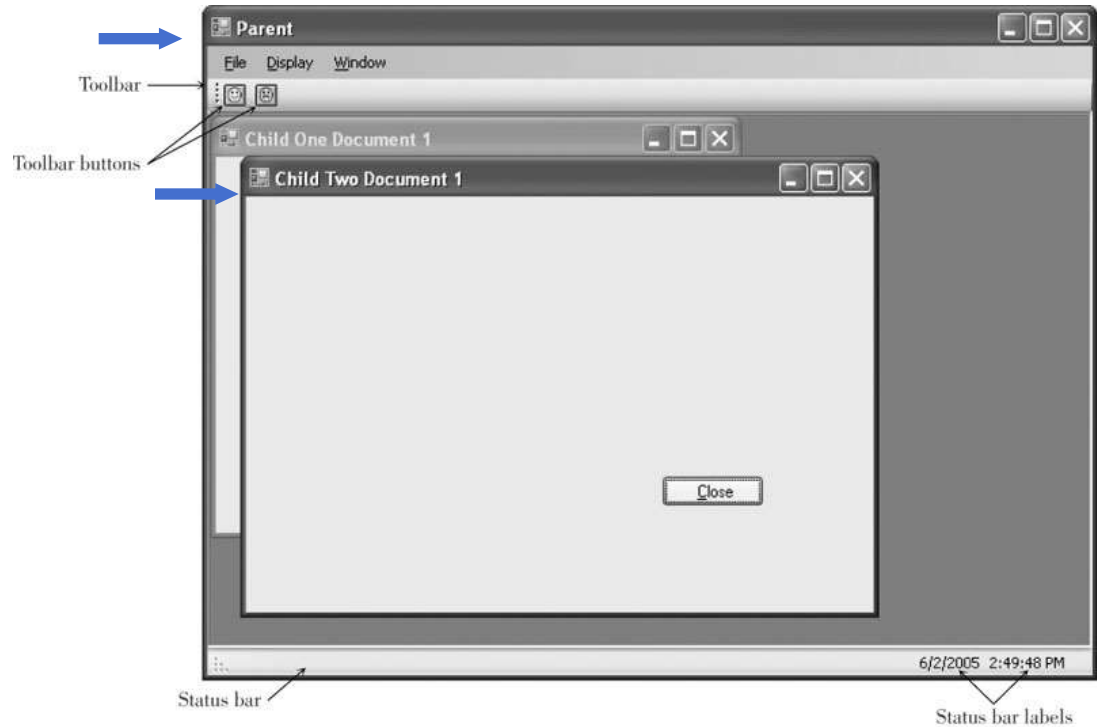
Multiple-Document Interface (2 of 2)

- Child form always stays within boundaries of the parent window.
- Close the parent window and all child windows close automatically.
- Child form always appears inside parent's area.
- Window menu displays list of open windows, allows movement from one active document to another.



MDI Application Forms

The main form is the parent and the smaller forms are the child forms.





Creating an MDI Project (1 of 2)

- At design time designate a form as Parent.
 - `IsMdiContainer` property = True
- At run time designate Child forms
 - Before displaying the Child form, from the Parent set the Child's `MdiParent` property to the current (parent) form.



Creating an MDI Project (2 of 2)

- If multiple child windows are displayed, the title bar of each child should be unique.
- Solution
 - Append a number to the title bar before displaying the form (like MS Word).



Child Form Code Example

```
Private Sub ChildOneMenuItem_Click(ByVal sender As _  
    System.Object, ByVal e As System.EventArgs) Handles _  
    ChildOneMenuItem.Click  
    ' Display Child One form.  
  
    Dim AChildOneForm As New childOneForm( )  
    AChildOneForm.MdiParent = Me  
    AchildOneForm.Show( )  
End Sub
```



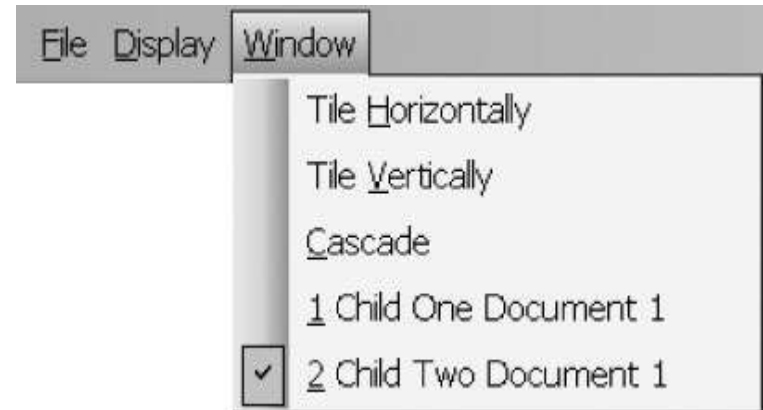
MDI Child Title Bar Example

```
' Module-level declarations.  
Dim childOneCountInteger As Integer  
Private Sub ChildOneMenuItem_Click(ByVal sender As _  
System.Object, ByVal e As System.EventArgs) Handles _  
ChildOneMenuItem.Click  
    ' Display Child One form.  
    Dim AChildOneForm As New childOneForm  
    AChildOneForm.MdiParent = Me  
    childOneCountInteger += 1  
    AChildOneForm.Text - "ChildOne Document " _  
    & childOneCountInteger.ToString( )  
    AChildOneForm.Show( )  
End Sub
```



Adding a Window Menu

- Parent form should include a Window menu to
 - List open Child forms
(Set menu's MdiList property to True.)
 - Allow the user to arrange multiple Child forms
 - Allow the user to switch between windows





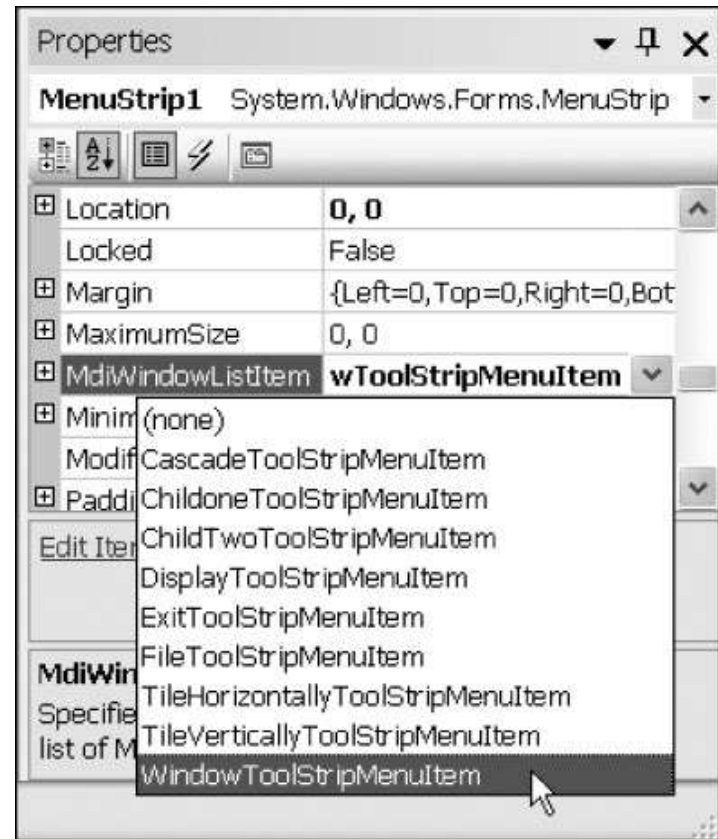
MenuStrip Control (1 of 2)

- After the menus are created for a MenuStrip control, the menus can be set to display a list of open child windows.
 - Display the properties of the MenuStrip (not a menu item) in the Properties window.
 - Drop down the list for the MidiWindowList Item property displaying all of the menu items that belong to the MenuStrip.
 - Select WindowToolStripMenuItem.
- Arranging the windows requires additional code.



MenuStrip Control (2 of 2)

Set the **MdiWindowListItem** property to make the Window menu display the list of open MDI child windows.





Layout Options

- Use an argument of the `LayoutMdi` method to set the type of layout.
- Examples

```
Me.LayoutMdi(MdiLayout.TileHorizontal)  
Me.LayoutMdi(MdiLayout.TileVertical)  
Me.LayoutMdi(MdiLayout.Cascade)
```



Toolbars and Status Bars

- Enhance usability of programs
 - A toolbar requires a **ToolStrip control** and the images in Resources to appear on the ToolStrip buttons.
 - Toolbars are an easy shortcut for menu items.
 - Images can be collected using the Items Collection Editor window.
 - With the ToolStrip selected, Standard Items can be inserted such as New, Open, Save, Print buttons and pictures are automatically added; however, the code must be written for each button.
 - Status bars appear at bottom of screen; display information for the user; require a **StatusStrip control**.
-



Toolbars

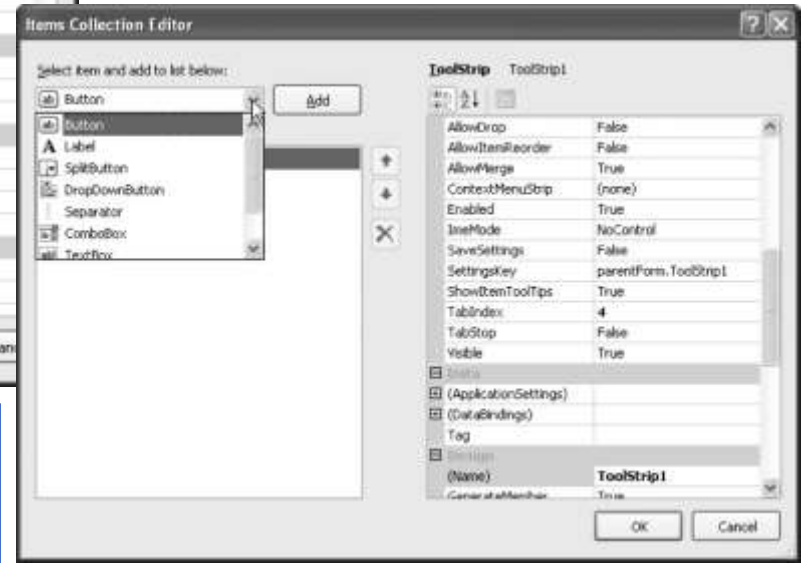
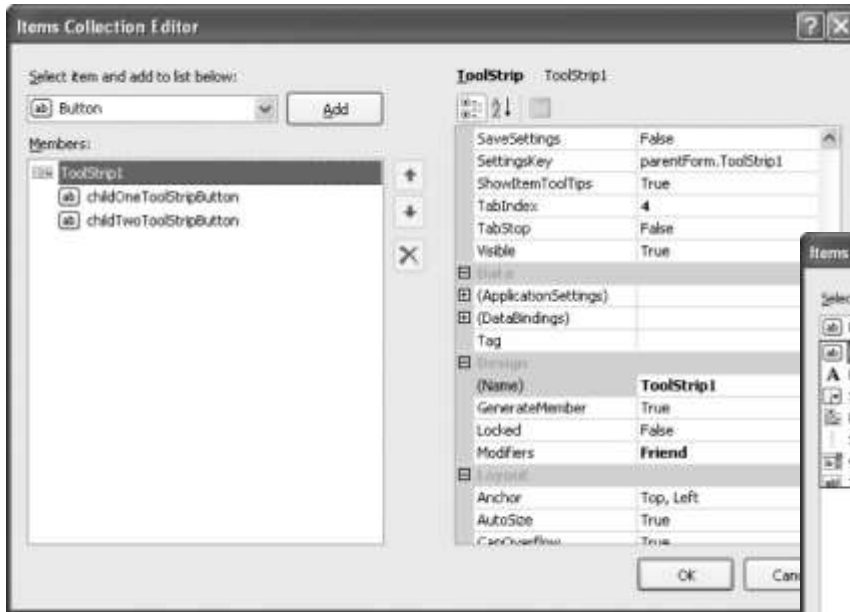
- Use the ToolStrip control to create a ToolStrip object (container) which holds several type of objects.



- Use the Items property and open the Items Collection Editor.
- In the ToolStrip's Items Collection Editor, drop down the list of available types of objects.



Items Collection Editor



Drop down list of available objects such as buttons.



Status Bars

- Usually located at the bottom of a form to display date, time, status of CapsLock or NumLock, error or informational messages
- Place the StatusBar control on the form.
 - Add a StatusStrip control.
 - Add [ToolStripStatusLabel](#) objects to the StatusStrip.
- Like ToolStrips, the easiest way to add items to the StatusStrip object is to select its items property in the Properties window to open the ItemsCollection Editor.
- Set where labels appear by setting the StatusStrip's RightToLeft property.



Assigning Values to ToolStripStatusLabel

- Assign values to the Text property at run time.
- Examples

```
DateToolStripStatusLabel.Text = Now.ToShortDateString( )  
TimeToolStripStatusLabel.Text = Now.ToLongTimeString( )  
InformationToolStripLabel.Text = "It's very late."
```



Displaying the Date and Time

- Use the properties and methods of the **DateTime structure** to retrieve and format the current date and time.
- **Now property** holds system date and time in numeric format that can be used for calculations.
- Generally, set initial values in Form_load event and use a Timer component to update the time.
- DateTime methods
 - ToShortDateString ToLongDateString
 - ToShortTimeString ToLongTimeString



Some Helpful Date Controls

- Provide the ability to display calendars on your form.
- DateTimePicker
 - Takes less screen space
 - Displays only day and date unless user drops down the calendar
- MonthCalendar
 - Displays calendar



DateTimePicker Control

- Value Property
 - Contains the date
 - Initially set to current date
 - User can select a date or can assign a date value to the property.



Calendar Controls

The screenshot shows a Windows application window titled "Calendar Controls". At the top left, there is a "DateTimePicker" control displaying "Wednesday, June 01, 2005". Below it is a smaller "MonthCalendar" control showing the month of June 2005 with the 1st highlighted. To the right, a larger "MonthCalendar" control also shows June 2005 with the 1st highlighted. Both month calendars include a "Today: 6/1/2005" indicator at the bottom.

DateTimePicker → Wednesday, June 01, 2005

MonthCalendar → June, 2005

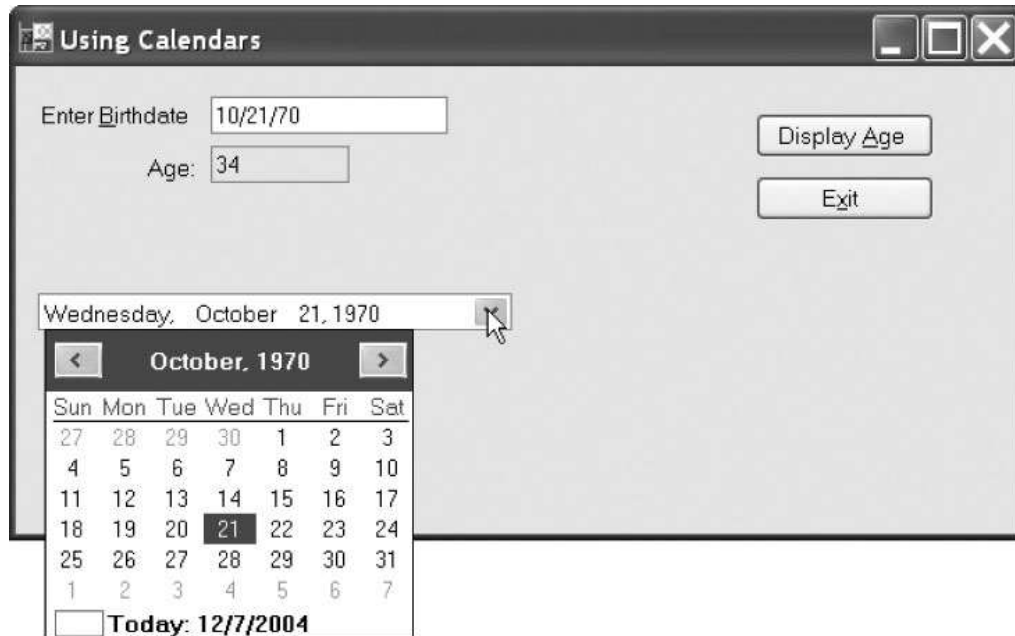
Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	1	2
3	4	5	6	7	8	9

Today: 6/1/2005



Using Calendars — Example

```
BirthdateDateTimePicker.Value =  
Convert.ToDateTime(BirthdateTextBox.Text)
```





Displaying Web Pages on a Windows Form

- The toolbox includes a WebBrowser control.



- Forms resemble a browser window in Internet Explorer.
- Use the control to display an HTML page, online or offline.



The WebBrowser Control

- Some useful properties, methods and events of the WebBrowser control are:

Url property	Set this property to a URL at design time or run time to navigate to the entered page. <pre>WebBrowser1.Url = New Uri("http://www.microsoft.com")</pre>
Navigate method	Execute this method at run time to navigate to the desired page. <pre>WebBrowser1.Navigate(New Uri("http://www.microsoft.com"))</pre>
DocumentCompleted event	A page has finished loading. You can use this event to add the page to the Items property of the combo box. <pre>With Me .ToolStripComboBox1.Items.Add(.WebBrowser1.Url) End With</pre>
DocumentTitleChanged event	A page with a new title finished loading. Use this event to change the form's title bar to the Web page's title. <pre>With Me .Text = .WebBrowser1.DocumentTitle End With</pre>
DocumentTitle property	Retrieves the title of the current Web page.



A WebBrowser Program





XML Data Files(1 of 5)

- XML is a platform-independent format that is not tied to a specific language or vendor.
- VB 2008 includes some new tools that make working with XML files easier than in the past.
- The Load method of an Xdocument can be used to read an XML file.



XML Data Files(2 of 5)

XDocument.Load Method—General Forms

```
Dim Identifier = XDocument.Load(Filename)  
Dim Identifier As XDocument = XDocument.Load(Filename)
```

Examples

```
Dim BookXDocument = XDocument.Load("books.xml")  
Private CustomerXDocument As XDocument =  
XDocument.Load("C:\Data\customers.xml")  
Dim InventoryDocument = XDocument.Load(CurDir() &  
"inventory.xml")
```



XML Data Files(3 of 5)

- Loading an XML File into an XElement Object

XML file can also be loaded into an XElement object. The difference between XDocument and XElement is that the XDocument contains the information about the document from the top of the file, while the root node is the first item in an XElement object.

```
Dim BookData = XElement.Load("books.xml")
```



XML Data Files(4 of 5)

LINQ Queries

- The axes notation can be used on the In clause of LINQ as well as in the Select clause.
- LINQ offers many operators.

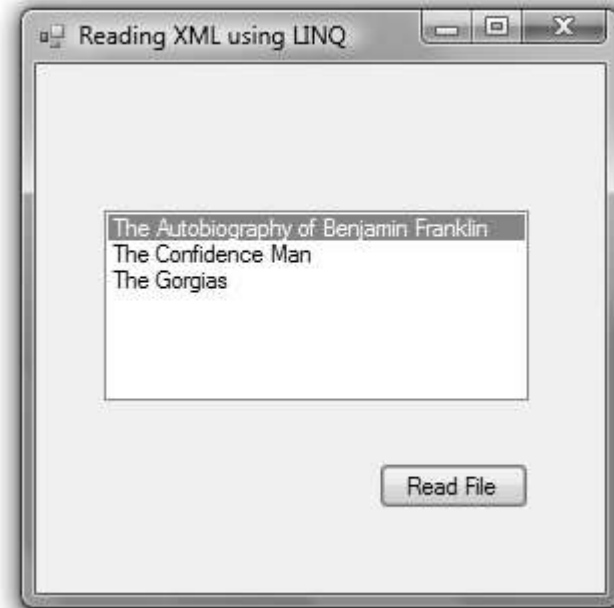
```
Dim MyQuery = From book In BookData...<price> _  
Select book.Value 'Gets the price.
```




XML Data Files(5 of 5)

- LINQ to XML Program Example

The program reads the books.xml file, uses a LINQ to XML query, and loads the resulting list of titles into a list box.





Windows Presentation Foundation(WPF)

- WPF provides the ability to create richer user interfaces for multiple platform development.
- The user interface in WPF applications uses XAML (pronounced “zammel”) code rather than HTML.
- **XAML (Extensible Application Markup Language)** is an XML-based language that is much more interactive than the traditional HTML.
- XBAP is used to refer to a **XAML Browser Application that** runs in an Internet browser.



WPF Interoperability

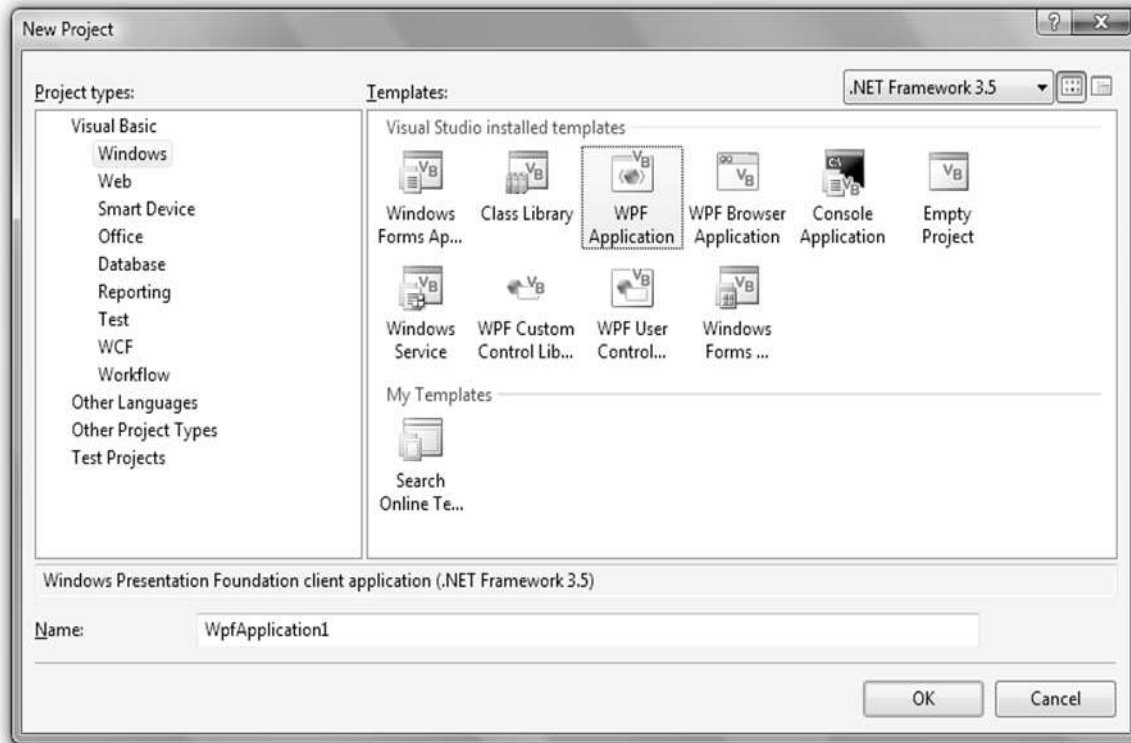
- Using WPF Interoperability, WPF controls can be used in a Windows Forms application



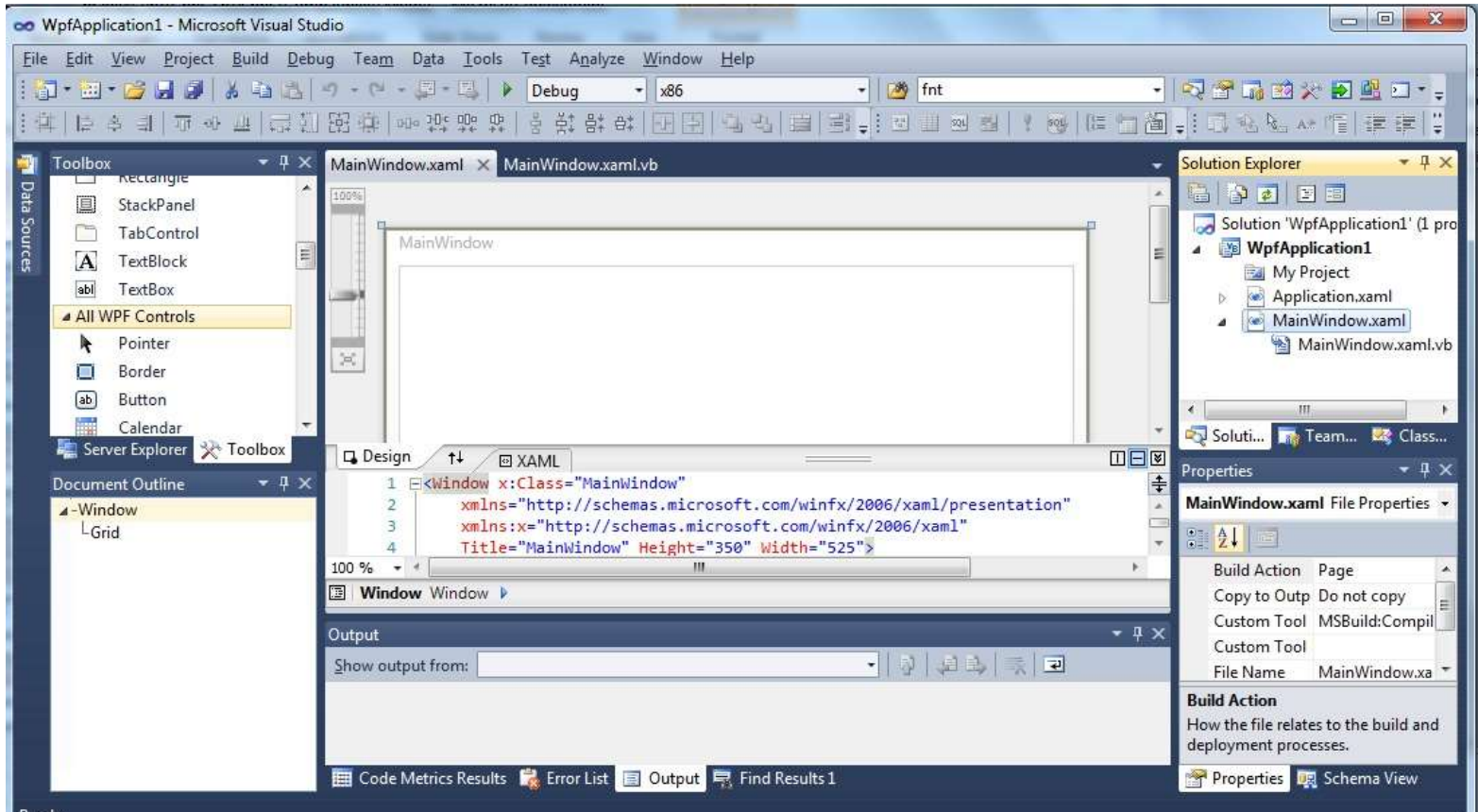
- Add an ElementHost control to a Windows Form to host WPF controls.



WPF Application Project Selection



WPF Project





WPF Project Development

- Interface design is similar to a Web Form
- WPF controls are similar to, but not identical to Windows controls.
- WPF creates a Window with a Title property, as opposed to a Form with a Text property
- See the step-by-step instructions for `Ch14WpfHelloWorld` for an introduction to WPF development.