

**Developing and Implementing Windows-based Applications with Visual Basic .NET and Visual Studio .NET Training Guide (Exam 70-306)**

**Copyright© 2003 by Que Publishing**

International Standard Book Number: 0789728192

**Warning and Disclaimer**

Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied. The information provided is on an "as is" basis. The author and the publisher shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the CD or programs accompanying it.

When reviewing corrections, always check the print number of your book. Corrections are made to printed books with each subsequent printing. To determine the print number of your book, view the copyright page. The print number is the right-most number on the line below the "First Printing" line. For example, the following indicates that this is the 1<sup>st</sup> printing of this title and it was printed in May 2003.

*First Printing: May 2003*

*06 05 04 03      4 3 2 1*

## First Printing Corrections

Pg	Error	Correction
2	Delete second bullet point	
37	Paragraph above Step By Step, third sentence: The default location of this file is in the bin\debug directory within the project's directory.	The default location of this file is in the <b>bin</b> directory within the project's directory.
53	Add Note:	Read-only and Write-only Properties The Get and Set accessors allow you both read and write access to a property. If you would rather make a property read-only, use the ReadOnly keyword in the property declaration and don't include and Set accessor in the property definition. On the other hand, if you just want a write-only property, use the WriteOnly keyword in the property declaration and don't include a Get accessor in the property definition.
60	Second paragraph, third sentence: Event handlers will normally <del>have a void return type and will</del> accept two arguments.	Event handlers will normally accept two arguments <b>and have no return value.</b>
61	Fourth paragraph, last two sentences: Step By Step 1.11 gets the MouseDown event from the Control class. By registering a handler for this event, Step By Step 1.11 becomes...	Step By Step 1.9 gets the MouseDown event from the Control class. By registering a handler for this event, Step By Step 1.9 becomes...
64-65	Delete Step 5 and renumber Step 6 to 5.	
66	Step 4, last two lines of code	Should read: AddHandler Me.MouseDown, AddressOf Me.StepByStep_1_12_MouseDown2

70	Add Note:	Form Properties In Step By Step 1.13, <u>Width</u> , Height, and Font are all properties of the form that contains the code. If you prefer, you can write these as Me.Width, Me.Height, and Me.Font, but the compiler will recognize them either way.
74	Second code snippet on page: Public Sub DrawString(string, Font, Brush, _ PointF, StringFormat)÷	Public Sub DrawString(string, Font, Brush, _ PointF, StringFormat)
78	Top of page, code lines 6 and 10: Dim penYellow As Pen = New Pen(Color.Blue, 20) grfx.DrawEllipse(penYellow, 20, 10, 100, 100)	Dim penBlue As Pen = New Pen(Color.Blue, 20) grfx.DrawEllipse(penBlue, 20, 10, 100, 100)
97	Question 14, first sentence: ... through a using directive at the top of the program.	... through <b>an Import statement</b> at the top of the program.
118	Step 10, code line 13: ' Attach a event handler to the	' Attach <b>an</b> event handler to the
119	Step 11, code line 8: ' Remove the most recetly added control	' Remove the most <b>recently</b> added control
136	Table 2.3, first member: AcceptReturn	Accept <b>s</b> Return
146	Step 3, first sentence: Change the ListBox control's Text property to ...	Change the ListBox control's <b>Name</b> property to ...

164	Step 6, code line 30: tnNode = New TreeNode	tnNode = New TreeNode(d)
185	Add new Step 25, and renumber current steps 25-29 to 26-30:	25. Double-click the form to open its module, and add the following line at the top: Imports System.IO
188	Step 7, seventh code line from the bottom: If text <> "then" Then	If text <> "" Then
200	Table 2.33, first member: ActiveMdiChild	ActiveMdiChild
208	Step 7, 11th line of code from bottom: If pos >= 1 Then	If pos >= 0 Then
240	Step 4, fourth and sixth lines of code: Dim unhandledException As Exception = # ' If no event source exist, create an event source	Dim unhandledException As Exception = # ' If no event source exists, create an event source
241	Third line of code on page: e1.Source = "Mileage Efficiency Calculator"	e1.Source = "Mileage Efficiency Calculator Log"
295	Step 7, code lines 16 and 22: sf = g.MeasureString(strTime, Font) g.DrawString(strTime, Font, New SolidBrush(ForeColor), pos)	sf = g.MeasureString(strTime, f) g.DrawString(strTime, f, New SolidBrush(ForeColor), pos)
311	Step 5, fourth line of code from end: lblAddResult.Text = String.Format( _	lblAddResult.Text = String.Format( " _
325	Exercise 3, add sentence to end:	Set the Text property of each of the label controls to 0.
325	Exercise 9: ... add a reference for Exercise4-1.dll.	... add a reference for Exercise4-1.dll.

507	Step 6, last sentence: The copy will have all the CustomerState values converted to uppercase.	The copy will have all the CustomerCity values converted to uppercase.
531	Answer 6	Correct answer is B.
609	Step 4, code line: aximp c:\winnt\system32\sysinfo.ocbx	aximp c:\winnt\system32\sysinfo.ocx
786	Step 7	Should read:  In the Solution Explorer window, select View All Files from the toolbar. Navigate to the bin folder. Right-click the folder and select Include In Project. Then right-click the folder and select Add, Add New Item. Choose to create an XML file, and name the XML file Exercise12-1.exe.config.
833	Step 4, last sentence: ... to Embedded Resources	... to Embedded Resource
861	Step 2, add sentence to end:	Delete the default Class1.vb file.
861	Step 3, add line of code to end:	Imports System.Text
906	Step 4, add sentence to end:	Set the Text property of lblLiveCalculators to 0.
922	Step 4, delete first line of code: <del>private void PopulateLogNames</del>	
923	Step 6, delete first two lines of code: <del>private void btnListen_Click(object sender, _ _System.EventArgs e)</del>	

1055	Second column, after bullets, paragraph should read:	You can also count as elective one of the four core exams (70-305, 70-306, 70-315, and 70-316). The elective exam must be form the opposite technology as the exam that you counted as core. For example, if you take the VB Windows-based Applications exam (70-306) as a core exam, you can take either the VB Web Applications exam (70-305) or the C# Web Applications exam (70-315) as an elective.
------	--	--

## Second Printing Corrections

Pg	Error	Correction
51	Seventh code line from the bottom: EditorBrowsable(EditorBrowsableState. <del>Never</del> ), _	EditorBrowsable(EditorBrowsableState. <b>Always</b> ), _
65	Replace entire code section under step 3	<p>With:</p> <pre> ' Delegate to handle the MouseDown event Sub StepByStep_1_12_MouseDown2(ByVal sender As Object, ByVal e As MouseEventArgs)     Dim frm1_12 As Form = CType(sender, Form)     If frm1_12.BackColor.ToArgb = Color.AntiqueWhite.ToArgb Then         frm1_12.BackColor = Color.LightCoral     Else         frm1_12.BackColor = Color.AntiqueWhite     End if End Sub  Private Sub StepByStep1_12_MouseDown(ByVal sender As Object, _ ByVal e As System.Windows.Forms.MouseEventArgs) Handles MyBase.MouseDown     MessageBox.Show(String.Format("x ={0}, Y={1}", e.X, e.Y), _ String.Format("The {0} mouse button hit me at:", e.Button.ToString())) End Sub </pre>

85	14th line of code after step 4:  ' Call the <code>Fill Rectangle</code> method	' Call the <code>FillRectangle</code> method
93	Fifth and seventh lines of code up from step 4:  'Increase the y Coordinate by the  <code>intYCoord += Font.Height</code>	'Increase the y Coordinate by the  <code>intYCoord += fnt.Height</code>
96	Second line of answer D to question 11:  <code>this.Font.Bold)</code>	<code>Me.Font.Bold)</code>
99	First sentence of answer 9:  <code>Graphics.Hwnd</code> is the function that gets a Graphics object for a Windows Form.	<code>Graphics.FromHwnd</code> is the function that gets a Graphics object for a Windows Form.
151	Fourth line of code under Step 4:  ' Add a list of System Fonts to  <code>theComboBox</code>	' Add a list of System Fonts to the <code>ComboBox</code>

170	<p>Replace block of code under Step 4</p>	<p>With:</p> <pre> Private Sub hscr_Scroll(ByVal sender As System.Object, _     ByVal e As System.Windows.Forms.ScrollEventArgs) Handles vscr.Scroll     pbImage.Left = CInt(GetHScrollAdjustment()) End Sub  Private Sub vscr_Scroll(ByVal sender As System.Object, _     ByVal e As System.Windows.Forms.ScrollEventArgs) Handles vscr.Scroll     pbImage.Top = CInt(GetVScrollAdjustment()) End Sub  Private Function GetHScrollAdjustment() As Double     Dim hPos As Double = CDb1(hscr.Value - hscr.Minimum)     Dim hDiff As Double = CDb1(hscr.Width - pbImage.Width)     'Adjust ticks for scroll thumb width     Dim hTicks As Double = CDb1 ((hscr.Maximum - hscr.LargeChange + 1)         hscr.Minimum)     GetHScrollAdjustment = (hDiff / hTicks) * hPos End Function  Private Function GetVScrollAdjustment() As Double     Dim vPos As Double = CDb1(vscr.Value - vscr.Minimum)     Dim vDiff As Double = CDb1(vscr.Height - pbImage.Height)     'Adjust ticks for scroll thumb height     Dim vTicks As Double = CDb1(vscr.Maximum - vscr.Minimum)     Get VScrollAdjustment = (vDiff / vTicks) * vPos End Function </pre>
194	<p>Sixth line of code in Step 6:</p> <p>' Check which menu it is and set the appropriate help text.</p>	<p>' Check which menu it is and set the appropriate help text.</p>
229	<p>Add to the end of Step 2</p>	<p>Set the MultiLine property of txtText to True.</p>



232	Last sentence of Step 2: Name the empty Label control <code>lblResults</code> .	Name the empty Label control <code>lblResult</code> .
319	Second paragraph up from Step By Step; last sentence:  You could also create a RESX file into a resources file from the command line by using the following command:	You could also convert a RESX file into a resources file from the command line by using the following command:
319	Step 1 at bottom of page; first two words:  Select <code>State</code>	Select <code>Start</code>
343	Second-to-last code line from the bottom of the page:  <code>Me.BindingContext (mdtBound) .Position -=1</code>	<code>Me.BindingContext (mdtBound) .Position -=1</code>
343	Second-to-last code line above Step 6:  <code>Me.BindingContext (mdtBound) .Position +=1</code>	<code>Me.BindingContext (mdtBound) .Position +=1</code>
433	Fifth line of code in Step 6:  <code>procedure</code>	<code>' procedure</code>
447	Add two lines of code between <code>End If</code> and <code>fsOut.Flush()</code> , and align them with <code>fsOut.Flush()</code>	<code>Loop</code>  <code>' Clean up</code>

449	<p>Code lines 9 and 13 up from Step 5:</p> <pre>Dim intBytesRead As Integer = 1 FsOut.Write(buf, 0, intBytesRead</pre>	<pre>Dim intBytesRead As Integer = -1 FsOut.Write(buf, 0, intBytesRead)</pre>
494	<p>Delete the sixth code line in Step 5 which reads:</p> <pre>' Find the specified row and delete it</pre>	
587	<p>Step 3, first sentence:</p> <p>Double-click the <b>ComboBox</b> control to open the form's module.</p>	<p>Double-click the <b>Button</b> control to open the form's module.</p>
642	<p>Bottom of the page, Selecting a Help Compiler, second paragraph, first sentence:</p> <p>An alternative if you'd like more modern-looking help is HTML Help 2, which is in beta as I write this chapter and should ship some time in 2003.</p>	<p>An alternative if you'd like more modern-looking help is HTML Help 2, which is in beta as I write this chapter and will probably ship some time in the future.</p>
647	<p>Step 5:</p> <p>Add a second <b>form</b> to the project. Name this form frmOptions. Set its Text property to Options, its MaximizeButton and MinimizeButton properties to False, its HelpButton property to True, and its FormBorderStyle property to FixedDialog.</p>	<p>Add a second <b>form</b> to the project. Name this form frmOptions. Set its Text property to Options, its MaximizeBox and MinimizeBox properties to False, its HelpButton property to True, and its FormBorderStyle property to FixedDialog.</p>

670	<p>Tip:</p> <p>To display a help button on a form, you must set the form's HelpButton property to True, its <b>MinButton</b> property to False, and its MaxButton property to False.</p>	<p>To display a help button on a form, you must set the form's HelpButton property to True, its <b>MinimizeBox</b> property to False, and its MaximizeBox property to False.</p>
672	<p>AutomaticDelay bullet at top of page, last word:</p> <p>AutoDelay</p>	<p>AutomaticDelay</p>
683	<p>Step 6, add line of code to top of code listing:</p>	<pre>Imports System.Runtime.InteropServices</pre>
766	<p>Fourth line of code in No. 4:</p> <pre>' Add a event log listener to the Listeners collection</pre>	<pre>' Add an event log listener to the Listeners collection</pre>
786	<p>Exercise 12.2, add right below Estimated Time: 30 minutes:</p>	<p>Note: Debugging SQL Server stored procedures requires the Enterprise Developer or Enterprise Architect version of Visual Studio.NET. If you are using the Professional version, you won't be able to complete this exercise.</p>
849	<p>Top of page, last sentence before No. 9:</p> <p>Set the Value property to <code>[CommonFilesFolder]RandNumCorp/</code>, where <i>CommonFilesFolder</i> is where the assembly file copy is stored.</p>	<p>Set the Value property to <code>[CommonFilesFolder]RandNumCorp/</code></p>
1031	<p>No. 59, in A, B, C, and D</p> <p>MinButton and MaxButton</p>	<p>Should be:</p> <p>MinimizeBox and Maximize box</p>

1039	No. 75:  C.	D.
------	-------------------	----

### Third printing corrections

96	<p>Question 11, Answer choices A and B:</p> <p>A. Me.Font.Bold = <del>true</del></p> <p>B. Me.Font = <del>new</del> Font(Me.Font,</p>	<p>A. Me.Font.Bold = <b>True</b></p> <p>B. Me.Font = <b>New</b> Font(Me.Font,</p>
99	<p>Question 15</p> <p>C. GraphicsPath class belongs to System.Drawing.Drawing2D namespace. <b>This</b> namespace must be included with an Imports directive to uniquely identify GraphicsPath class in your program.</p>	<p>C. GraphicsPath class belongs to System.Drawing.Drawing2D namespace. <b>The</b> namespace must be included with an Imports directive to uniquely identify GraphicsPath class in your program.</p>
202	<p>Step 5, last sentence</p> <p>Add an OpenFileDialog component named <b>dlgOpen</b> to this form.</p>	<p>Add an OpenFileDialog component named <b>dlgOpenFile</b> to this form.</p>
429	<p>Third set of code on the page:</p> <pre>UPDATE Products SET Discontinued = 1 FROM Suppliers INNER JOIN Products ON Suppliers.SupplierID = Products.<b>Supplier.ID</b> WHERE Suppliers.Country = 'Italy'</pre>	<pre>UPDATE Products SET Discontinued = 1 FROM Suppliers INNER JOIN Products ON Suppliers.SupplierID = Products.<b>SupplierID</b> WHERE Suppliers.Country = 'Italy'</pre>

447	Step 4, code line 6 from the end:  ' clean up	' <b>Clean</b> up
642	Note added to the page:	In February 2003, Microsoft announced it was discontinuing the HTML Help 2.0 compiler and would no longer be releasing it.
696	Last line on the page:  Dim intCurrentY As Integer = 0	Dim intCurrentY As Integer = <b>e.MarginBounds.Top</b>
697	Line 5 from the top:  Brushes.Black, 0, intCurrentY)	Brushes.Black, <b>e.MarginBounds.Left,</b> <b>intCurrentY)</b>
698	6th code line from the bottom:  Dim intCurrentY As Integer = 0  Last line on the page:  If intCurrentY <= e.MarginBounds.Height <b>Then</b>	Dim intCurrentY As Integer = <b>e.MarginBounds.Top</b> <b>' Reset the line counter</b> <b>mintLines = 0</b>  If intCurrentY <= e.MarginBounds.Height <b>+ e.MarginBounds.Top Then</b>

699	<p>Line 3 from the top:</p> <pre>If mintLines &lt; nudCopies.Value Then</pre> <p>Line 8 from the top:</p> <pre>Brushes.Black, 0, intCurrentY)</pre>	<pre>If mintLines &lt;= nudCopies.Value Then</pre> <pre>Brushes.Black, e.MarginBounds.Left, intCurrentY)</pre>
704	<p>Line 12 from the bottom:</p> <pre>Dim intCurrentY As Integer = 0</pre> <p>Line 7 from the bottom:</p> <pre>If intCurrentY &lt;= e.MarginBounds.Height Then</pre> <p>Line 4 from the bottom:</p> <pre>If mintLines &lt; nudCopies.Value Then</pre> <p>Last line on the page:</p> <pre>Brushes.Black, 0, intCurrentY)</pre>	<pre>Dim intCurrentY As Integer = e.MarginBounds.Top</pre> <pre>' Reset the line counter</pre> <pre>mintLines = 0</pre> <pre>If intCurrentY &lt;= e.MarginBounds.Height + e.MarginBounds.Top Then</pre> <pre>If mintLines &lt;= nudCopies.Value Then</pre> <pre>Brushes.Black, e.MarginBounds.Left, intCurrentY)</pre>

709

Line 20 from the bottom

```
Dim intCurrentY As Integer = 0
```

Line 11 from the bottom:

```
If intCurrentY <= e.MarginBounds.Height Then
```

Last line on the page:

```
Brushes.Black, 0, intCurrentY)
```

```
Dim intCurrentY As Integer = e.MarginBounds.Top
```

```
If intCurrentY <= e.MarginBounds.Height  
+ e.MarginBounds.Top Then
```

```
Brushes.Black, e.MarginBounds.Left  
intCurrentY)
```

713

Line 7 from the top:

```
Dim mintLines As Integer = 0
```

Line 20 from the top:

```
Dim intCurrentY As Integer = 0
```

Line 25:

```
If intCurrentY <= e.MarginBounds.Height Then
```

Line 28:

```
If mintLines < nudCopies.Value Then
```

Line 14 from the bottom:

```
Brushes.Black, 0, intCurrentY)
```

```
Dim mintLines As Integer = 0  
' Print settings for this page  
Dim pgsCustom As PageSettings
```

```
Dim intCurrentY As Integer = e.MarginBounds.Top  
' Reset the line counter  
mintLines = 0
```

```
If intCurrentY <= e.MarginBounds.Height  
+ e.MarginBounds.Top Then
```

```
If mintLines <= nudCopies.Value Then
```

```
Brushes.Black, e.MarginBounds.Left,  
intCurrentY)
```



714	<p>Code section above step 6:</p> <pre>Private Sub btnPageSetup_Click(_   ByVal sender As System.Object, _     ByVal e As System.EventArgs) Handles btnPageSetup.Click   ' Create a PageSettings object and send it to the dialog   Dim pgsCustom As PageSettings = New PageSettings()   PageSetupDialog1.PageSettings = pgsCustom    ...</pre>	<pre>Private Sub btnPageSetup_Click(_   ByVal sender As System.Object, _     ByVal e As System.EventArgs) Handles btnPageSetup.Click   ' Create a PageSettings object if this is the first time in   If pgsCustom Is Nothing Then     pgsCustom = New PageSettings()   End If   PageSetupDialog1.PageSettings = pgsCustom</pre>
716	<p>Step 5, line 6:</p> <pre>Dim mintLines As Integer = 0</pre> <p>Line 19:</p> <pre>Dim intCurrentY As Integer = e</pre> <p>Line 24:</p> <pre>If intCurrentY &lt;= e.MarginBounds.Height Then</pre> <p>Line 13 from the bottom:</p> <pre>Brushes.Black, e, intCurrentY)</pre>	<pre>Dim mintLines As Integer = 0 ' Print settings for this page <b>Dim pgsCustom As PageSettings</b>  Dim intCurrentY As Integer = <b>e.MarginBounds.Top</b>   ' Reset the line counter   mintLines = 0  If intCurrentY &lt;= e.MarginBounds.Height <b>+ e.MarginBounds.Top Then</b>  Brushes.Black, <b>e.MarginBounds.Left</b> intCurrentY)</pre>

717	<p>Line 19:</p> <pre>Private Sub btnPageSetup_Click( _     ByVal sender As System.Object, _         ByVal e As System.EventArgs) Handles btnPageSetup.Click <del>    ' Create a PageSettings object and send it to the</del> <del>    dialog</del> <del>    Dim pgsCustom As PageSettings = New PageSettings()</del> </pre>	<pre>Private Sub btnPageSetup_Click( _     ByVal sender As System.Object, _         ByVal e As System.EventArgs) Handles btnPageSetup.Click     ' Create a PageSettings object if this is the first time in     If pgsCustom Is Nothing Then         pgsCustom = New PageSettings()     End If </pre>
719	<p>Step 5, code line 6:</p> <pre>Dim mintLines As Integer = 0</pre> <p>Line 3 from the bottom:</p> <pre>Dim intCurrentY As Integer = 0</pre>	<pre>Dim mintLines As Integer = 0 ' Print settings for this page Dim pgsCustom As PageSettings  Dim intCurrentY As Integer = e.MarginBounds.Top ' Reset the line counter mintLines = 0</pre>

720

Line 3:

```
If intCurrentY <= e.MarginBounds.Height Then
```

Line 6:

```
If mintLines < nudCopies.Value Then
```

Line 9:

```
Brushes.Black, 0, intCurrentY)
```

Line 6 from the bottom:

```
' Create a PageSettings object and send it to the dialog  
Dim pgsCustom As PageSettings = New PageSettings()
```

```
If intCurrentY <= e.MarginBounds.Height  
+ e.MarginBounds.Top Then
```

```
If mintLines <= nudCopies.Value Then
```

```
Brushes.Black, e.MarginBounds.Left, _  
intCurrentY)
```

```
' Create a PageSettings object if this is the first time in  
If pgsCustom Is Nothing Then  
    pgsCustom = New PageSettings()  
End If
```

723	<p>Line 6 of code:</p> <pre>Dim mintLines As Integer = 0</pre> <p>Line 19:</p> <pre>Dim intCurrentY As Integer = 0</pre> <p>Line 24:</p> <pre>If intCurrentY &lt;= e.MarginBounds.Height Then</pre> <p>Line 27:</p> <pre>If mintLines &lt; nudCopies.Value Then</pre> <p>Line 32:</p> <pre>Brushes.Black, 0, intCurrentY)</pre>	<pre>Dim mintLines As Integer = 0 ' Print settings for this page Dim pgsCustom As PageSettings</pre> <pre>Dim intCurrentY As Integer = e.MarginBounds.Top ' Reset the line counter mintLines = 0</pre> <pre>If intCurrentY &lt;= e.MarginBounds.Height + e.MarginBounds.Top Then</pre> <pre>If mintLines &lt;= nudCopies.Value Then</pre> <pre>Brushes.Black, e.MarginBounds.Left, intCurrentY)</pre>
724	<p>Middle of page:</p> <pre>Private Sub btnPageSetup_Click( _   ByVal sender As System.Object, _   ByVal e As System.EventArgs) Handles btnPageSetup.Click   ' Create a PageSettings object and send it to the   dialog   Dim pgsCustom As PageSettings = New PageSettings()   ...</pre>	<pre>Private Sub btnPageSetup_Click( _   ByVal sender As System.Object, _   ByVal e As System.EventArgs) Handles btnPageSetup.Click   ' Create a PageSettings object if this is the first time in   If pgsCustom Is Nothing Then     pgsCustom = New PageSettings()</pre>

727	<p>Question 5, code line 10:</p> <pre>Dim mintRowsPrinted As Integer = 0</pre>	<pre>Dim mintRowsPrinted As Integer = 0 ' Print settings for this page Dim pgsCustom As PageSettings</pre>
728	<p>Left column, line 3:</p> <pre>Dim intCurrentY As Integer = 0</pre> <p>Line 25:</p> <pre>e.MarginBounds.Height Then</pre>	<pre>Dim intCurrentY As Integer = e.MarginBounds.Top</pre> <pre>e.MarginBounds.Height + e.MarginBounds.Top Then</pre>
729	<p>Step 4, code line 6:</p> <pre>Dim mintLines As Integer = 0</pre>	<pre>Dim mintLines As Integer = 0 ' Print settings for this page Dim pgsCustom As PageSettings</pre>
730	<p>Left column, line 11:</p> <pre>Dim intCurrentY As Integer = 0</pre> <p>Line 17:</p> <pre>e.MarginBounds.Height Then</pre> <p>Line 17 from the bottom:</p> <pre>Brushes.Black, 0, _</pre>	<pre>Dim intCurrentY As Integer = e.MarginBounds.Top ' Reset the line counter mintLines = 0</pre> <pre>e.MarginBounds.Height + e.MarginBounds.Top Then</pre> <pre>Brushes.Black, e.MarginBounds.Left _</pre>

731	Exam Question 1, code line 12 from the bottom:  If mintLines < _	If mintLines <= _
862	Line 24:  "ngen.exe", "" & strArgs & " ")	"ngen.exe", "" & strArgs & " ")
889	Step 3, right column:  Download the setup.exe bootstrapper sample from http://msdn.microsoft.com/downloads/sample.asp?url=/msdnfiles/027/001/830/msdncompositedoc.xml.	Download the setup.exe bootstrapper sample from http://msdn.microsoft.com/vstudio/downloads/tools/bootstrapper/.

#### Fourth printing corrections

53	Last paragraph, second sentence:  ... specify the runtime behavior ...	... specify the <b>design-time</b> behavior ...
91	Step 6, 11th, 12th, and 13th lines of code:  sbText.Remove( sbText.Length - 1, 1)	sbText.Remove( sbText.Length - 1, 1)

107	Step 3, seventh line of code:  <code>' apecify the form's size</code>	<code>' sspecify the form's size</code>
397	Question 12, first sentence:  Select the SqlDataAdapter4 object.	Select the SqlDataAdapter2 object.
447	Step 4, eighth through 12th code lines from the end:  <code>Do While b &gt; -1     b = fsIN,ReadByte()     If b &gt; -1 Then         fsOut.WriteByte(CType (b, Byte))     End If</code>	<code>Do While b &gt; -1     b = fsIN,ReadByte()     If b &gt; -1 Then         fsOut.WriteByte(CType (b, Byte))     End If</code>
449	Step 4, 15th and 16 lines of code from the bottom:  <code>' Copy all date from in to out, using a 4K buffer</code>	<code>' Copy all date from in to out, using a 4K buffer</code>
449	Step 4, eighth through 10th lines of code from the bottom:  <code>If intBytesRead &gt; 0 Then     fsOut.Write(buf, 0, intBytesRead) End If</code>	<code>If intBytesRead &gt; 0 Then     fsOut.Write(buf, 0, intBytesRead) End If</code>
504	Delete final two lines on page:  <del><code>Dim tnodWorking As TreeNode Dim tnodAttribute As TreeNode</code></del>	

507	Delete 15th line of code from end of Step 4: <del>Dim tnodWorking As TreeNode</del>	
521	Question 4, 10th line of code: ' Load the <del>customer</del> list	' Load the <b>product</b> list
626	Step 6, second to last line of code: buf.ToString.SubString(4, len)	buf.ToString.SubString( <b>0</b> , len)
653	Step 6, last two sentences: Click OK to add this file to the <del>Welcome</del> table of contents entry. <del>Click OK to create the entry.</del>	Click OK to add this file to the table of contents.
671	Step 4, second line: Your <del>a</del> random number.	Your random number.
679	Exercise 4, third and fourth lines of code:  ByVal e As Microsoft.Win32.UserPreferenceChangedEventArgs)	ByVal e As Microsoft.Win32.UserPreferenceChangedEventArgs)
851	Exercise 10, delete second-to-last sentence: <del>Enter a query and click the button to get the results of the query.</del>	
906	Step 4: Set the Text property of <del>lbl</del> LiveCalculators to 0.	Set the Text property of <b>lbl</b> LiveCalculators to 0.



927	Step 3, last sentence:  Add to ... Counter Name (chCounterName)...	Add to ... Counter Name (chCounterName)...
930	Add to top of page (just under <i>continued</i> ):  Private Const PERFORMANCE_STR As String = _ "Current performance level:" "	
930	Replace the four code lines on the page that read:  <del>lblCurrentLevel.Text =</del> <del>pc14_5.NextValue().ToString()</del>	with:  lblCurrentLevel.Text = PERFORMANCE_STR & pc14_5.NextValue().ToString()
940	Question 6, 20th line of code from bottom of column 1:  ' SelectedIndexChaged event	' SelectedIndexChang <u>e</u> d event

978 - 979	Replace all the code in Step 4:	with:  Private Sub btnGetProperties_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnGetProperties.Click ' Tell the CLR which principal policy is in use AppDomain.CurrentDomain.SetPrincipalPolicy(PrincipalPolicy.WindowsPrincipal) lbProperties.Items.Clear()  ' Get the current identity Dim wi As WindowsIdentity = WindowsIdentity.GetCurrent() ' Dump its properties to the listbox With lbProperties.Items .Add("WindowsIdentity:") .Add(" Authentication type: " & wi.AuthenticationType) .Add(" Is Anonymous: " & wi.IsAnonymous.ToString()) .Add(" Is Authenticated: " & wi.IsAuthenticated.ToString()) .Add(" Is Guest: " & wi.IsGuest.ToString()) .Add(" Is System: " & wi.IsSystem.ToString()) .Add(" Name: " & wi.Name) .Add(" Token: " & wi.Token.ToString()) End With  ' Get the current principal Dim prin As WindowsPrincipal = New WindowsPrincipal(wi) ' Dump its properties to the listbox With lbProperties.Items .Add("WindowsPrincipal:") .Add(" Authentication Type: " & prin.Identity.AuthenticationType) .Add(" Is Authenticated: " & prin.Identity.IsAuthenticated.ToString()) .Add(" Name: " & prin.Identity.Name) .Add(" Member of Domain Users: " & prin.IsInRole("LARKGROUP\Domain Users").ToString()) End With  End Sub
-----------------	---------------------------------	---

This errata sheet is intended to provide updated technical information. Spelling and grammar misprints are under process, but are not listed on this errata sheet.