

Microsoft Visual C# 2013 Step by Step

John Sharp

ISBN: 978-0-7356-8183-5

First printing: November, 2013

To ensure the ongoing accuracy of this book and its companion content, we've reviewed and confirmed the errors listed below. If you find a new error, we hope you'll report it to us on our website: www.microsoftpressstore.com/contact-us/errata.

Page	Location	Description	Date corrected
	Companion content	<p>An extra folder level (C_Sharp_SBS) seems to have crept in to the file paths in the zip, and if you do not extract the files directly into the Documents folder the names of some of the generated .pp files for the later exercises cause this error. These files are not critical.</p> <p>However, to avoid this problem, open the ZIP file after downloading it, move to the C_Sharp_SBS folder in the ZIP, and then drag the Microsoft Press folder to your Documents folder. This should extract the files and put them in the correct place as documented by the exercises in the book, and hopefully the path names should no longer be too long.</p>	11/12/2014

Page	Location	Description	Date corrected
	Companion content	<p>For a brief period after the book was published, the uploaded sample files for this chapter were the wrong version.</p> <p>The first line in the compareClick method should be exactly as reproduced in the book:</p> <pre>int diff = dateCompare(first, second);</pre> <p>The problem is that the DatePicker controls in Windows 7 and Windows 8.1 expose different properties for accessing the current date. I hid these differences by using two properties named first and second in the MainWindow class, which is what the code in the book uses. For interest, these properties look like this in Windows 7:</p> <pre>#region Conversion properties to cater for the differences between the DatePicker control in Windows 7 and Windows 8.1 private DateTime first { get { return this.firstDate.SelectedDate.GetValueOrDefault(this.firstDate.DisplayDate); } } private DateTime second { get { return this.secondDate.SelectedDate.GetValueOrDefault(this.secondDate.DisplayDate); } } #endregion</pre> <p>In Windows 8.1 they look like this:</p> <pre>#region Conversion properties to cater for the differences between the DatePicker control in Windows 7 and Windows 8.1 private DateTime first { get { return this.firstDate.Date.LocalDateTime; } } private DateTime second { get { return this.secondDate.Date.LocalDateTime; } } #endregion</pre>	11/12/2014

Page	Location	Description	Date corrected
		It is not actually necessary to understand how these properties work (properties are covered later in the book).	
xxiv	"Code Samples" section (three occurrences)	The Code Samples section refers to the file C#_SBS.zip. However, the name of the download containing the code files is actually 9780735681835_files.zip.	11/12/2014
32	12 lines from bottom	Reads: typeof(GraphWindow) should be typeof(MainPage). Should read: if (!rootFrame.Navigate(typeof(MainPage), e.Arguments))	11/12/2014
36	Step 5, new reader aid	The following new Note reader aid should be added at the end of step 5: This code will display a warning concerning the use of an asynchronous method when it is compiled. You do not have to be concerned about this warning. Asynchronous methods are explained more fully in Chapter 24.	11/12/2014
46	Note	Reads: you can press Windows key+B to return... Should read: you can press the Windows key+D to return...	
71	Step 3	Two instances. Reads: class Should read: type	11/12/2014

Page	Location	Description	Date corrected
90	Item #9 in the To Do code, second case statement, four lines down from "above the default label:"	<p>It looks like there are some formatting issues with the EPUB and MOBI editions of the book. The completed code for the CopyOne method should look like this:</p> <pre>private void copyOne(char current) { switch (current) { case '<': target.Text += "&lt;"; break; case '>': target.Text += "&gt;"; break; case '&': target.Text += "&amp;"; break; case '\": target.Text += "&#34;"; break; case '\': target.Text += "&#39;"; break; default: target.Text += current; break; } }</pre> <p>In the Nook version, the following steps will allow the user to write the code:</p> <p>8. Add the following statements shown in bold to the switch statement after the opening brace for the statement and directly before the default label:</p> <pre>switch (current) { case '<': target.Text += "&lt;"; break; default: target.Text += current; break; }</pre> <p>If the current character being copied is a left angle bracket (<), the preceding code appends the string "&lt;" to the text being output in its place.</p> <p>9. Add the following statements to the switch statement after the</p>	11/12/2014

Page	Location	Description	Date corrected
		<p>break statement you have just added and above the default label:</p> <pre> case '>' : target.Text += "&gt;"; break; case '&' : target.Text += "&amp;"; break; case '\"' : target.Text += "&#34;"; break; case '\\"' : target.Text += "&#39;"; break; </pre> <p>Lubbock, TX</p>	
105	"Comparing dates in real-world applications" sidebar, first paragraph	The reference to the "Compare" method (5 lines from the end of the box) should read "CompareTo."	11/12/2014
131	Third line on top	<p>Reads:</p> <pre>current = current + digit;</pre> <p>Should read:</p> <pre>current = digit + current;</pre>	
221	Fourth paragraph, step 10	<p>Reads:</p> <p>HoldingRoutedEventArgs e</p> <p>Should read:</p> <p>RightTappedRoutedEventArgs e</p>	11/12/2014
301, 303	Bottom, top of page(s)	<p>Pg 301, last paragraph should read:</p> <p>The Windows.UI namespace contains the definition of the Colors class which you will use when you set the color of a shape as it is drawn. In WPF, this class is defined in the System.Windows.Media namespace, which is already referenced by the DrawingPad.xml file</p> <p>Pg 303, sentence (in brackets) at the top should read: (The Colors class is provided as part of the .NET Framework)</p>	11/12/2014
305	Top of page	The semi-colon after the closing "}" for the two implementations of the ChewGrass methods (for the Horse and Sheep classes) should be removed.	11/12/2014

Page	Location	Description	Date corrected
397	Step 15	<p>Reads:</p> <pre>if (this.LeftTree != null) { result = this.LeftTree.WalkTree(); }</pre> <p>Should read:</p> <pre>if (this.LeftTree != null) { result += this.LeftTree.WalkTree(); }</pre>	
445	Second paragraph, second sentence	<p>Should read:</p> <p>The compiler uses this code to generate an implementation of the <code>IEnumerator<T></code> interface containing a <code>Current</code> property and a <code>MoveNext</code> method.</p>	11/12/2014

Page	Location	Description	Date corrected
449	Quick Reference table, Do This column	<p>The first row should read:</p> <pre>public class Tree<TItem> : IEnumerable<TItem> { ... IEnumerator<TItem> IEnumerable<TItem>.GetEnumerator() { ... } }</pre> <p>Third row should read:</p> <pre>IEnumerator<TItem> IEnumerable<TItem>.GetEnumerator() { for (...) { yield return ... } }</pre> <p>Second row, Current and MoveNext members should reference the IEnumerator<TItem> type:</p> <pre>public class TreeEnumerator<TItem> : IEnumerator<TItem> { ... TItem IEnumerator<TItem>.Current { get { ... } } bool IEnumerator<TItem>.MoveNext() { ... } }</pre>	11/12/2014
506	Step 7	The paragraph after the code block in Step 7 should be deleted.	11/12/2014
564	First line of code after second paragraph	<p>Reads:</p> <pre>cancellationToken,Register(doAdditionalWork);</pre> <p>Should read:</p> <pre>cancellationToken.Register(doAdditionalWork);</pre>	
679	Both figures	In both figures, the ComboBox should be empty.	11/12/2014