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.

When reviewing corrections, always check the print number of your book. Corrections are made to printed books with each subsequent printing. To determine the printing of your book, view the copyright page. The print number is right-most number on the line below the "First Printing" line. For example, the following indicates the 4th printing of a title.

First Printing: January 2002
Third Printing with corrections: April 2004
<table>
<thead>
<tr>
<th>Misprint</th>
<th>Correction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page 234 in code listing line 29 HTMLDocumentEvents_onclick</td>
<td>HTMLDocumentEvents2_onclick</td>
</tr>
<tr>
<td>Page 300 2nd paragraph after bullets; last line the CCW holds onto</td>
<td>the RCW holds onto</td>
</tr>
<tr>
<td>Page 418 first code line towards the end ClassName’&lt;/object&gt;</td>
<td>ClassName’&lt;/object&gt;</td>
</tr>
<tr>
<td>Page 421 2nd to last paragraph–add line after last sentence: physical assembly file.</td>
<td>physical assembly file. You must also ensure that your machine policy allows managed code to run in the Internet zone.</td>
</tr>
<tr>
<td>Page 479 middle of page code 4th line f.BorderStyle = 2</td>
<td>f.BorderStyle = 2</td>
</tr>
<tr>
<td>Page 492 2nd paragraph after bullets shutdown of the CLR, including the finalization of any .NET objects that have not yet been finalized, and exits the process with the passed-in error code.</td>
<td>shutdown of the CLR by calling CoEEShutDownCOM, finalizing any .NET objects that have not yet been finalized, then exiting the process with the passed-in error code.</td>
</tr>
<tr>
<td>Page 609 in Listing 2nd line 2: &lt;is blank here&gt;</td>
<td>2: Imports System</td>
</tr>
<tr>
<td>Page 659 last paragraph/last sentence Even if you reference the Interop Assembly type library defining IDispatch and write a class</td>
<td>The only useful thing you can do with a .NET definition of</td>
</tr>
</tbody>
</table>
that implements this interface, it will be completely ignored by the CLR and unusable from COM.

IDispatch (which can be found as an empty interface in the PIA for the OLE Automation type library) is make a .NET class marked with ClassInterfaceType.None implement it, because this is the only way to make the type library exporter create a coclass whose default interface is IDispatch. (Some COM clients may care about this.) At run time, however, the CLR ignores the fact this dummy IDispatch interface is implemented.

Interface IprovideClassInfo

Page 759 first mono listing using (IDisposable)FileWriter f = new FileWriter() {
    foreach(int i in new int[1,2,3,4,5,6,7,8,9,10])
        f.WriteLine(i.ToString());
}

Page 763 4th paragraph, 2nd line

VB.NET

Page 764 line 17
public struct

Page 836 line 17

VB .NET

Page 841 line 17
public class

Page 934 code lines 93-101
93:     if (pcbRead == NULL)
94:     {
95:         // User isn't interested in how many bytes were read
96:         *pcbRead = originalStream->Read(array, 0, cb);
97:     }

93:     if (pcbRead == NULL)
94:     {

else {
  originalStream->Read(array, 0, cb);
}

Page 1018 2nd note Digging Deeper- add line after final sentence
per source interface.

Page 1062 replace figure
Delete existing 22.1

Page 1084 replace both figures
Delete existing 22.5
Delete existing 22.6

Page 1150 figure caption
Three Classes are used

Page 1227 Listing heading- delete mono on Class
The WizardVisualization Class

Page 1252 Last paragraph; last sentence
offset multiplied by the array element size to the pointer value passed to

// User isn't interested in how many bytes were read
originalStream->Read(array, 0, cb);
else {
  pcbRead = originalStream->Read(array, 0, cb);
}
Three classes are used

The WizardVisualization Class
offset (a number of bytes) to the pointer value passed

largeArray, 40));

pointers that the RCW holds onto.

GDI32.DLL

static extern bool Heap32ListNext

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