Füllen eines Strukturansicht-Steuerelements mit XML-Daten in Visual C# 2005 oder Visual C# .NET

Dieser Artikel ist eine Übersetzung des folgenden englischsprachigen Artikels der Microsoft Knowledge Base:
317597 (http://support.microsoft.com/kb/317597/EN-US/)
How to populate a TreeView control with XML data in Visual C# 2005 or in Visual C# .NET

Artikel-ID : 317597
Geändert am : Montag, 25. Juni 2007
Version : 4.3

Eine Version dieses Artikels für Microsoft Visual Basic .NET finden Sie unter 308063 (http://support.microsoft.com/kb/308063/DE/).

Auf dieser Seite

Zusammenfassung

Dieser Artikel beschreibt die Vorgehensweise zum Füllen eines **Strukturansicht**-Steuerelements mit XML-Daten in Microsoft Visual C# 2005 oder Microsoft Visual C# .NET. Da sowohl XML als auch das **Strukturansicht**-Steuerelement die Daten in einem hierarchischen Format darstellen, bietet sich das **Strukturansicht**-Steuerelement für die Anzeige von XML-Daten an.

Das **Strukturansicht**-Steuerelement hat eine **Nodes**-Auflistung mit **TreeNode**-Stammobjekten. Jeder **TreeNode** hat wiederum eine eigene **Nodes**-Auflistung mit mehr als einem untergeordneten **TreeNode**.

Hinweis: In diesem Beispiel werden die DOM-Parsing-Klassen (DOM = Document Object Model) von .NET zur Verarbeitung von XML verwendet.

Weitere Informationen zum Design von XML in .NET Framework finden Sie im Abschnitt "Verweise".

Voraussetzungen

Die folgende Liste führt die empfohlene Hardware, Software und Netzwerkinfrastruktur sowie die benötigten Service Packs auf:

- Microsoft Windows XP, Microsoft Windows 2000 oder Microsoft Windows NT 4.0 Service Pack 6a
- Microsoft Data Access Components (MDAC) 2.6 oder höher
- Microsoft Visual Studio 2005 oder Microsoft Visual Studio .NET

In diesem Artikel wird vorausgesetzt, dass Sie über Erfahrungen in den folgenden Bereichen verfügen:

- Visual C# 2005-Syntax oder Visual C# .NET-syntax
- XML und die zugehörigen Standards
- Windows Forms

Vorgehensweise zum Erstellen und Füllen des Strukturansicht-Steuerelements mit XML

1. Fügen Sie den folgenden XML-Beispielcode in eine neue Textdatei namens "Sample.xml" ein. Diese Datei enthält die XML-Beispieldaten für dieses Beispiel:

```
<?xml version="1.0"?> <family> <parent>id="grandfather" <parent>id="father"
<parent>id="brother" <child>id="niece" </child> </parent> <parent>id="me"
<child>id="son"</child> <child>id="daughter"</child> </parent>
<child>id="sister"</child> </parent> <parent>id="uncle" <parent>id="cousin sister"
<child>id="second cousin"</child> </parent> <child>id="cousin brother"</child> </parent>
</parent> </family>
```

2. Erstellen Sie eine neue Windows-Anwendung in Visual C# 2005 oder Visual C# .NET. Form1 wird standardmäßig zur Anwendung hinzugefügt.
3. Ziehen Sie neue **Strukturansicht**-, **Schaltfläche**-, **Label**- und **TextBox**-Steuerelemente auf Form1.
4. Fügen Sie den folgenden Beispielcode am Ende des Abschnitts mit den **using**-Direktiven in Form1.cs hinzu:

```
using System.Xml;
```

5. Fügen Sie den folgenden Code in das Ereignis **Form1_Load** ein:

```
// Initialize the controls and the form. label1.Text = "File Path"; label1.SetBounds(8,
8, 50, 20); textBox1.Text = Application.StartupPath + "\\Sample.xml";
textBox1.SetBounds(64, 8, 256, 20); button1.Text = "Populate the TreeView with XML";
button1.SetBounds(8, 40, 200, 20); this.Text = "TreeView control from XML"; this.Width =
336; this.Height = 368; treeView1.SetBounds(8, 72, 312, 264);
```

6. Fügen Sie den folgenden Code in das Ereignis **Button1_Click** ein:

```
try { // SECTION 1. Create a DOM Document and load the XML data into it. XmlDocument dom
= new XmlDocument(); dom.Load(textBox1.Text); // SECTION 2. Initialize the TreeView
control. treeView1.Nodes.Clear(); treeView1.Nodes.Add(new
TreeNode(dom.DocumentElement.Name)); TreeNode tNode = new TreeNode(); tNode =
treeView1.Nodes[0]; // SECTION 3. Populate the TreeView with the DOM nodes.
AddNode(dom.DocumentElement, tNode); treeView1.ExpandAll(); } catch(XmlException xmlEx)
{ MessageBox.Show(xmlEx.Message); } catch(Exception ex) { MessageBox.Show(ex.Message); }
```

7. Fügen Sie den folgenden Beispielcode in das Ereignis **Button1_Click** ein:

```
private void AddNode(XmlNode inXmlNode, TreeNode inTreeNode) { XmlNode xNode; TreeNode
tNode; XmlNodeList nodeList; int i; // Loop through the XML nodes until the leaf is
reached. // Add the nodes to the TreeView during the looping process. if
(inXmlNode.HasChildNodes) { nodeList = inXmlNode.ChildNodes; for(i = 0;
i<=nodeList.Count - 1; i++) { xNode = inXmlNode.ChildNodes[i]; inTreeNode.Nodes.Add(new
TreeNode(xNode.Name)); tNode = inTreeNode.Nodes[i]; AddNode(xNode, tNode); } } else { //
Here you need to pull the data from the XmlNode based on the // type of node, whether
attribute values are required, and so forth. inTreeNode.Text =
(inXmlNode.OuterXml).Trim(); } } } }
```

8. Drücken Sie die Taste [F5], um die Anwendung erstellen und ausführen zu lassen. Vergewissern Sie sich, dass der XML-Dateipfad richtig ist, und klicken Sie anschließend auf die **Schaltfläche**. Die XML-Daten sollten jetzt im **Strukturansicht**-Steuerelement erscheinen.

Hinweis: Die Ressource kann eine Datei, ein URL oder ein XML-Stream sein. Informationen zur Verwendung der Klasse "XmlDocument" zum Laden von XML-Daten aus verschiedenen Ressourcen finden Sie im Abschnitt "Verweise".

Vorgehensweise zum Füllen des Strukturansicht-Steuerelements mit erforderlichen Daten

Der vorhergehende Beispielcode ordnet die XML-Strukturdaten direkt der **Strukturansicht** zu und zeigt alle Daten an. Alternativ können Sie der Anzeige zusätzliche Informationen hinzufügen oder nicht gewünschte Daten überspringen.

Häufig soll vielleicht nur ein Teil der XML-Daten angezeigt werden. Der Teil der Daten, den Sie anzeigen möchten, ist eventuell dynamisch erstellt, Ergebnis einer XSL-Transformation (XSL = Extensible Stylesheet Language) oder Ergebnis einer XPath-Abfrage. Dieser Abschnitt beschreibt, wie Sie ein neues XML-Dokument erstellen, dass nur die erforderlichen Knoten enthält, und dann das neue Dokument zum **Strukturansicht**-Steuerelement hinzufügen.

Beispielsweise rufen die folgenden Schritte nur die untergeordneten Elemente der ursprünglichen XML-Daten über eine xPath-Abfrage ab und fügen diese Liste dann als neuen Knoten zur **Strukturansicht** hinzu.

1. Fügen Sie den folgenden Code direkt vor der Zeile **TreeView1.ExpandAll** im vorhergehenden Beispiel ein:

```
// SECTION 4. Create a new TreeView Node with only the child nodes. XmlNodeList nodelist
= dom.SelectNodes("//child"); XmlDocument cDom = new XmlDocument();
cDom.LoadXml("<children></children>"); foreach(XmlNode node in nodelist) { XmlNode
newElem = cDom.CreateNode(XmlNodeType.Element, node.Name, node.LocalName);
newElem.InnerText = node.InnerText; cDom.DocumentElement.AppendChild(newElem); }
treeView1.Nodes.Add(new TreeNode(cDom.DocumentElement.Name)); tNode =
treeView1.Nodes[1]; AddNode(cDom.DocumentElement, tNode);
```

2. Erstellen Sie die Anwendung, und führen Sie sie aus. Diese Anwendung sollte zusätzlich zu den ursprünglichen Daten einen neuen untergeordneten Wurzelknoten in der **Strukturansicht** anzeigen.

Informationsquellen

Weitere Informationen finden Sie im folgenden Artikel der Microsoft Knowledge Base:

313651 (http://support.microsoft.com/kb/313651/DE/) INFO: Wegweiser für XML in dem .NET

Weitere Informationen finden Sie auf den folgenden MSDN-Websites:

http://msdn2.microsoft.com/de-de/library/28hw3sce(vs.71).aspx (http://msdn2.microsoft.com/de-de/library/28hw3sce(vs.71).aspx)

http://msdn2.microsoft.com/de-de/xml/default.aspx (http://msdn2.microsoft.com/de-de/xml/default.aspx)

Die Informationen in diesem Artikel beziehen sich auf:

- Microsoft Visual C# 2005 Express Edition
- Microsoft Visual C# .NET 2003 Standard Edition
- Microsoft Visual C# .NET 2002 Standard Edition
- Microsoft .NET Framework 1.1
- Microsoft .NET Framework 1.0

Keywords: kbhowtomaster kbbcl kbwindowsforms kbctrl KB317597