

Re: Treeview

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.vb/2004-12/3775.html>

From: Bob Hollness (*bob_at_blockbuster.com*)

Date: 12/20/04

Date: Mon, 20 Dec 2004 16:43:48 +0100

Thanks. I have so far thrown together this code (with a lot of assistance from a webpage i found). It does the trick great. My only problem now is that I would like to somehow differentiate between files and folders by putting the appropriate icons in their places. This has me kinda stumped at the moment. I guess the only way to do this is to look for a file extension and the change the icon. As you can see in the code, there is a rather dodgy IF statement which tries to do this. The problem I have is that VB seems to only apply this rule for the first file in a directory (child node) and then all the others have the folder icon.

Any ideas / improvements?

```
Private Function BuildNodeList(ByVal currentPath As String, ByVal NodeTree
As TreeNodeCollection) As Boolean
```

```
    Dim NodeText As String
    Dim nNode As TreeNode
    Dim strPath() As String
```

```
    If currentPath.Chars(0) = "\" Then currentPath =
currentPath.Substring(1)
```

```
    strPath = currentPath.Split("\")
```

```
    For Each NodeText In strPath
```

```
        For Each nNode In NodeTree
```

```
            If nNode.Text = NodeText Then
```

```
                If PathData = "" Then Exit Function
```

```
                PathData = PathData.Substring(strPath(0).Length)
                Add2Node2(PathData, nNode.Nodes)
```

```
            Exit Function
        End If
```

Next

If NodeText <> "" Then

 If Mid(NodeText, Len(NodeText) - 3, 1) = "." Then

 'file

 NodeTree.Add(NodeText)

 NodeTree.Item(NodeTree.GetEnumerator.Current).ImageIndex

= 2

 NodeTree.Item(NodeTree.GetEnumerator.Current).SelectedImageIndex

= 2

 Else

 'folder

 NodeTree.Add(NodeText)

 NodeTree.Item(NodeTree.GetEnumerator.Current).ImageIndex

= 1

 NodeTree.Item(NodeTree.GetEnumerator.Current).SelectedImageIndex

= 1

 End If

 Add2Node2 = True

End If

Next

End Function

--

Bob

I'll have a B please Bob.

"Nick Stansbury" <nick.stansbury@sage-removepartners.com> wrote in message
news:%23bZSL8p5EHA.2804@TK2MSFTNGP15.phx.gbl...

> Not that I know of. Here's some kind of pseudo code - you'll have to fill
> in

> the blanks

>

> Class FilePath

> Public ChildCollection as Collection

> Public Address as string

> Public DirectoryPath as string

> Public FileName as string

> ParentAddress as string

> End Class

>

> Dim TempCollection as new Collection

> Dim BaseFilePath as FilePath 'the root object

> 'open a stream from the file and read each line into a local variable.

> Return EOF = true when finished with the last line

> dim StrTmp as stirng

> Dim CountOfSlashes as long

> Dim StringIndex as long

> dim LastSlash as boolean

> dim CurrentPath as filePath

>

> Do until EOF

microsoft.public.dotnet.languages.vb: Re: Treeview

```
> CurrentPath = new FilePath
> StringIndex = 0
> strTmp = readLine()
> CurrentPath.Address = StrTmp
> if instr(StrTmp, ".") > 0 then
>     'this is a file, so we have to grab the directory path
> stripping off the filename
>     StringIndex = StrTmp.LastIndexOf("\")
>     CurrentPath.DirectoryPath = Left(StrTmp, StringIndex + 1)
> 'should get the C:\MyFolder\ section of C:\MyFolder\MyFile.txt
>     FileName = Mid(StrTmp, StringIndex + 1) ' should get
> MyFile.txt part of the above sample string
>     else
>         CurrentPath.DirectoryPath = StrTmp
>         FileName = "" 'set to ZLS for testing purposes
>     end if
>     'now get the parent address which should be the same for every line
>     if instr(StrTmp, "\", 3) > 0 then 'start at the 4th character in
> the
> string (i.e. start just after the first slash in C:\MyFolder\MyFile.txt)
> and
> see if there are any following slashes
>     StringIndex = StrTmp.LastIndexOf("/")
>     'now add the whole string up to this point as the parent
> address
>     currentPath.ParentAddress = left(StrTmp, StringIndex)
> 'should add the whole of strtmp up to the last "\" as the parent address
>     end if
>     TempCollection.add(CurrentPath)
> Loop
>
> 'now it should simple be a case of iterating through the collection and
> adding the object in a structured way to the base object.
>
>
> Hope this helps - basically the idea is that you parse the strings to
> break
> them up into the parent section, the filename and the current address. You
> then should compare the currentAddress to the parentAddress and be able to
> group them into a single recursive set of objects. Once you have them in a
> nice useable structure you should be able to draw the tree view very
> easily.
> You could also persist the object tree into memory for later use
>
> Nick
>
>
>
>
> "Bob Hollness" <bob@blockbuster.com> wrote in message
> news:u6j1AMo5EHA.3840@tk2msftngpl3.phx.gbl...
>> do you know of anyone of telling VB that each entry in the treeview must
> be
>> unique?
>>
>> --
>>
>> Bob
>>
>> -----
>> I'll have a B please Bob.
```

microsoft.public.dotnet.languages.vb: Re: Treeview

```
>>
>> "Nick Stansbury" <nick.stansbury@sage-removepartners.com> wrote in
>> message
>> news:%23E769Bo5EHA.1396@tk2msftngpl3.phx.gbl...
>> > Bob,
>> >
>> > You are, I presume, doing the binding code yourself. To make this
> work
>> > you'll need to a recursive binding methodology. My best guess would be
>> > that
>> > you'll need
>> > to parse the text file in such a way that you do the work of marking up
>> > parents and children, and then add them to the tree recursively. Doing
>> > this
>> > by hand is actually quite a complex problem - you might want to look at
>> > how
>> > the text file is populated - outputting it as XML would make the whole
>> > process a lot easier.
>> >
>> > Regards,
>> >
>> > Nick Stansbury
>> >
>> >
>> >
>> > "Bob Hollness" <bob@blockbuster.com> wrote in message
>> > news:ugG4z3n5EHA.2540@TK2MSFTNGP09.phx.gbl...
>> >> Hi.
>> >>
>> >> I have a text file containing a list of files and their path. I.e.
>> >>
>> >> c:\MyDocs\File1.txt
>> >> c:\MyDocs\File2.txt
>> >> c:\MyDocs\File3.txt
>> >> c:\MyDocs\File4.txt
>> >> C:\MyDocs\Folder\File1.txt
>> >>
>> >> etc.
>> >>
>> >> What I want to do is create a treeview that makes this look like a
>> > directory
>> >> listing (i.e. Windows Explorer). The problem I have is getting VB to
>> >> realise that "C:\MyDocs\Folder\File1.txt" should be a child of
>> > "c:\MyDocs\"
>> >> and add it as a child node. So far, all i get is each item listed
>> >> individually in the treeview.
>> >>
>> >> Any ideas/help? Thanks.
>> >>
>> >> --
>> >>
>> >> Bob
>> >>
>> >> -----
>> >> I'll have a B please Bob.
>> >>
>> >>
>> >
>> >
>>
>>
```

microsoft.public.dotnet.languages.vb: Re: Treeview

>
>