

# A Typical Snippet Explorer Form Code

```
Imports System.Diagnostics
Imports System.Windows.Forms

Public Class Explorer1

    Public MyFolder As String

    Private Sub Explorer1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        SetUpListViewColumns()
        Opener()

    End Sub
    Public Sub Opener()
        Dim MyStarter As String
        MyStarter = CurDir() & "\Snippets"

        MyFolder = MyStarter
        SecondOpener()
    End Sub
    Private Sub SecondOpener()
        TreeView.Nodes.Clear()
        Dim Nd As New TreeNode
        ' change the name of the folder to be scanned in the following statement
        Dim initFolder As String = MyFolder
        Nd = TreeView.Nodes.Add(initFolder)
        ' Only the statement that calls the ScanFolder() subroutine is required to populate
the TreeView control
        ' Because quite a bit of code is executed from within the application's Load event
handler,
        ' it will take a while before the form is displayed.
        ' Instead of waiting to populate the control before showing the form, you can call
the Show
        ' method to show the form. In the ScanFolder() subroutine you must call the form's
Refresh method
        ' every now and then to update the form.
        Me.Show()
        Application.DoEvents()
        Me.Cursor = Cursors.WaitCursor
        ScanFolder(initFolder, Nd)
        Me.Cursor = Cursors.Default
    End Sub
    Sub ScanFolder(ByVal folderSpec As String, ByRef currentNode As TreeNode)
        Dim thisFolder As String
        Dim allFolders() As String
        Try
            allFolders = System.IO.Directory.GetDirectories(folderSpec)
            For Each thisFolder In allFolders
                Dim Nd As TreeNode
                Nd = New TreeNode(System.IO.Path.GetFileName(thisFolder))
                Nd = currentNode.Nodes.Add(Nd.Text)
                folderSpec = thisFolder
                ScanFolder(folderSpec, Nd)
                ' Call the DoEvents method to update the display.
                ' Because it may take a while to populate the TreeView control with the
structure
                ' of a folder with many subfolders/files, we update the display so that the
user
                ' doesn't think the application isn't responding.
            Next
        Catch
        End Try
    End Sub
End Class
```

```

        ' An alternative would be to display the name of the folder being scanned
        ' on the form's caption bar with the following statement:
    '
    Me.Text = "Scanning " & folderSpec
    Application.DoEvents()
    Next
Catch ex As Exception
    ErrorMess()
End Try

End Sub
Private Sub TreeView_AfterSelect(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.TreeViewEventArgs) Handles TreeView.AfterSelect
    Dim Nd As TreeNode
    Dim pathName As String
    Nd = TreeView.SelectedNode
    pathName = Nd.FullPath
    Me.Text = pathName
    ShowFiles(pathName)
End Sub
Sub ShowFiles(ByVal selFolder As String)
    ListView.Items.Clear()
    Dim files() As String
    Dim file As String
    files = System.IO.Directory.GetFiles(selFolder)
    Dim TotalSize As Long
    For Each file In files
        Dim LItem As New ListViewItem

        LItem.Text = System.IO.Path.GetFileName(file)
        Dim FI As New System.IO.FileInfo(file)
        LItem.SubItems.Add(FI.Length.ToString("#,###"))
        LItem.SubItems.Add(FormatDateTime(FI.CreationTime, DateFormat.ShortDate))
        LItem.SubItems.Add(FormatDateTime(FI.LastAccessTime, DateFormat.ShortDate))
        LItem.ImageKey = "WeblogPost"
        ListView.Items.Add(LItem)
        TotalSize += file.Length
    Next
    Me.Text = Me.Text & " [" & TotalSize.ToString("#,###") & " bytes]"
End Sub
Private Sub ErrorMess()
    Message.TextBox1.Text = " Error"
    Message.Width = 200
    Message.Timer1.Interval = 2000
    Message.Show()
    Message.Visible = True
End Sub
Private Sub ListView_DoubleClick(sender As Object, e As EventArgs) Handles
ListView.DoubleClick
    Try
        Dim SourceFile As String
        SourceFile = ToolStripTextBox1.Text.Trim
        ToolStripTextBox1.SelectionStart = ToolStripTextBox1.Text.Length - 4
        ToolStripTextBox1.SelectionLength = 4
        ' If ToolStripTextBox1.SelectedText = ".rtf" Or ".RTF" Then '

        'Else
        If ToolStripTextBox1.SelectedText = ".RTF" Then
            Frist.Show()
            Frist.ToolStripTextBox1.Text = SourceFile
            Frist.RichTextBox1.LoadFile(SourceFile)
        End If
    End Try
End Sub

```

```

        Frist.Visible = True
        Frist.Dirty = False
        Exit Sub
    ElseIf ToolStripTextBox1.SelectedText = ".rtf" Then
        Frist.Show()
        Frist.ToolStripTextBox1.Text = SourceFile
        Frist.RichTextBox1.LoadFile(SourceFile)
        Frist.Visible = True
        Frist.Dirty = False
        Exit Sub
    Else
        ' MsgBox(SourceFile)
        ErrorMess()
        Exit Sub
    End If
Catch ex As Exception
    ErrorMess()
End Try

End Sub
Private Sub ListView_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
ListView.SelectedIndexChanged
    Try
        ToolStripTextBox1.Text = TreeView.SelectedNode.FullPath & "\" &
ListView.FocusedItem.Text

        Catch ex As Exception

    End Try
End Sub
Private Sub OpenToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
OpenToolStripMenuItem.Click
    Opener()

End Sub

Private Sub SetView(ByVal View As System.Windows.Forms.View)
    'Figure out which menu item should be checked
    Dim MenuItemToCheck As ToolStripMenuItem = Nothing
    Select Case View
        Case View.Details
            MenuItemToCheck = DetailsToolStripMenuItem
        Case View.LargeIcon
            MenuItemToCheck = LargeIconsToolStripMenuItem
        Case View.List
            MenuItemToCheck = ListToolStripMenuItem
        Case View.SmallIcon
            MenuItemToCheck = SmallIconsToolStripMenuItem
        Case View.Tile
            MenuItemToCheck = TileToolStripMenuItem
        Case Else
            Debug.Fail("Unexpected View")
            View = View.Details
            MenuItemToCheck = DetailsToolStripMenuItem
    End Select

    'Check the appropriate menu item and deselect all others under the Views menu
    For Each MenuItem As ToolStripMenuItem In ListViewToolStripButton.DropDownItems
        If MenuItem Is MenuItemToCheck Then
            MenuItem.Checked = True

```

```

        Else
            MenuItem.Checked = False
        End If
    Next

    'Finally, set the view requested
    ListView.View = View
End Sub
Private Sub ListToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ListToolStripMenuItem.Click
    SetView(View.List)
End Sub
Private Sub DetailsToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles DetailsToolStripMenuItem.Click
    SetView(View.Details)
End Sub
Private Sub LargeIconsToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles LargeIconsToolStripMenuItem.Click
    SetView(View.LargeIcon)
End Sub
Private Sub SmallIconsToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles SmallIconsToolStripMenuItem.Click
    SetView(View.SmallIcon)
End Sub
Private Sub TileToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TileToolStripMenuItem.Click
    SetView(View.Tile)
End Sub
Private Sub SetUpListViewColumns()

    ' TODO: Add code to set up listview columns
    Dim lvColumnHeader As ColumnHeader

    ' Setting Column widths applies only to the current view, so this line
    ' explicitly sets the ListView to be showing the SmallIcon view
    ' before setting the column width
    SetView(View.SmallIcon)

    lvColumnHeader = ListView.Columns.Add("Column1")
    ' Set the SmallIcon view column width large enough so that the items
    ' do not overlap
    ' Note that the second and third column do not appear in SmallIcon
    ' view, so there is no need to set those while in SmallIcon view
    lvColumnHeader.Width = 150
    lvColumnHeader.Text = "File Name"

    ' Change the view to Details and set up the appropriate column
    ' widths for the Details view differently than SmallIcon view
    SetView(View.Details)

    ' The first column needs to be slightly larger in Details view than it
    ' was for SmallIcon view, and Column2 and Column3 need explicit sizes
    ' set for Details view
    lvColumnHeader.Width = 250

    lvColumnHeader = ListView.Columns.Add("Column2")
    lvColumnHeader.Width = 70
    lvColumnHeader.Text = "File Size"

    lvColumnHeader = ListView.Columns.Add("Column3")
    lvColumnHeader.Width = 70

```

```

lvColumnHeader.Text = "Date Last altered"

End Sub
Private Sub ToggleFoldersVisible()
    'First toggle the checked state of the associated menu item
    FoldersToolStripMenuItem.Checked = Not FoldersToolStripMenuItem.Checked

    'Change the Folders toolbar button to be in sync
    FoldersToolStripButton.Checked = FoldersToolStripMenuItem.Checked

    ' Collapse the Panel containing the TreeView.
    Me.SplitContainer.Panel1Collapsed = Not FoldersToolStripMenuItem.Checked
End Sub
Private Sub FoldersToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles FoldersToolStripMenuItem.Click
    ToggleFoldersVisible()
End Sub
Private Sub FoldersToolStripButton_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles FoldersToolStripButton.Click
    ToggleFoldersVisible()
End Sub

Private Sub ExitToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ExitToolStripMenuItem.Click
    'Close this form
    Me.Close()
End Sub
Private Sub ToolBarToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ToolBarToolStripMenuItem.Click
    'Toggle the visibility of the toolstrip and also the checked state of the associated
menu item
    ToolBarToolStripMenuItem.Checked = Not ToolBarToolStripMenuItem.Checked
    ToolStrip.Visible = ToolBarToolStripMenuItem.Checked
End Sub
Private Sub StatusBarToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles StatusBarToolStripMenuItem.Click
    'Toggle the visibility of the statusstrip and also the checked state of the
associated menu item
    StatusBarToolStripMenuItem.Checked = Not StatusBarToolStripMenuItem.Checked
    StatusStrip.Visible = StatusBarToolStripMenuItem.Checked
End Sub
Private Sub HowToPathToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
HowToPathToolStripMenuItem.Click
    Try
        Dim MyPath As String
        MyPath = ListView.FocusedItem.Text '.ToString
        MsgBox(MyPath)
    Catch ex As Exception
        ErrorMess()
    End Try
End Sub
Private Sub MakeNewDirectoryToolStripMenuItem_Click(sender As Object, e As EventArgs)
Handles MakeNewDirectoryToolStripMenuItem.Click
    Dim logDirectoryProperties As System.IO.DirectoryInfo
    Dim strResult As String
    Dim NewName As String
    strResult = InputBox("Please enter New Directory Name.", "New File Name")
    If strResult = "" Then
        MsgBox("You clicked cancel")
    Exit Sub

```

```

ElseIf strResult.Length >= 1 Then
    MsgBox("you entered " & strResult & ".")
    NewName = strResult
    If My.Computer.FileSystem.DirectoryExists(MyFolder & "\" & NewName) Then
        logDirectoryProperties = My.Computer.FileSystem.GetDirectoryInfo(MyFolder &
"\NewName")
        MsgBox("That Directory already Exist" & vbCrLf & "Please try another Name")
        Exit Sub
    Else
        Mkdir(MyFolder & "\" & NewName)
        Opener()
    End If
Else
    Exit Sub
End If

End Sub
Private Sub DeleteSnippetToRecycleBinToolStripMenuItem_Click(sender As Object, e As
EventArgs) Handles DeleteSnippetToRecycleBinToolStripMenuItem.Click
    Dim MyPath As String
    Dim MyNum As Integer
    MyNum = ListView.FocusedItem.Index
    MyPath = TreeView.SelectedNode.FullPath & "\" & ListView.FocusedItem.Text '.ToString
'MsgBox(MyPath)
    My.Computer.FileSystem.DeleteFile(MyPath,
Microsoft.VisualBasic.FileIO.UIOption.AllDialogs,
Microsoft.VisualBasic.FileIO.RecycleOption.DeletePermanently,
Microsoft.VisualBasic.FileIO.UICancelOption.DoNothing)
    ListView.Items.RemoveAt(MyNum) ' = MyNum
    ListView.Refresh()
    Opener()
    TreeView.ExpandAll()

End Sub
Private Sub DeleteDirectoryToRecycleBinToolStripMenuItem_Click(sender As Object, e As
EventArgs) Handles DeleteDirectoryToRecycleBinToolStripMenuItem.Click
    Dim MyPath As String
    Dim MyNum As Integer
    MyNum = TreeView.SelectedNode.Index
    MyPath = TreeView.SelectedNode.FullPath ' & "\" & ListView.FocusedItem.Text
'.ToString
    MsgBox(MyPath)
    My.Computer.FileSystem.DeleteDirectory(MyPath, FileIO.UIOption.AllDialogs,
FileIO.RecycleOption.SendToRecycleBin)
    Opener()
    TreeView.ExpandAll()
    TreeView.SelectedNode = TreeView.TopNode

End Sub
Private Sub AddNewSnippetToolStripMenuItem_Click(sender As Object, e As EventArgs)
Handles AddNewSnippetToolStripMenuItem.Click
    If TreeView.SelectedNode.Index = 0 Then
        MsgBox("YOU MUST FRIST SELECT A DIRECTORY")
        Exit Sub
    End If

    Frist.ToolStripTextBox1.Text = TreeView.SelectedNode.FullPath
    Frist.Show()
    Frist.SaveSnippetToolStripMenuItem.Enabled = False
    Frist.Visible = True

```

```
End Sub  
End Class
```