

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;
using System.Xml;

namespace FileTree
{
    public partial class TreeForm : Form
    {
        MySorter sorter = new MySorter ();

        public TreeForm ()
        {
            InitializeComponent ();

            // pripoj sorter, pokracovani: listView_ColumnClick
            listView.ListViewItemSorter = sorter;
            listView.Sorting = SortOrder.Ascending; // nezbytne
        }

        private void listView_ColumnClick (object sender, ColumnClickEventArgs e)
        {
            if ( sorter.column_index == e.Column )
            {
                // zmena poradi
                sorter.inv = ! sorter.inv;
            }
            else
            {
                // vyber jiny sloupec
                sorter.column_index = e.Column;
                sorter.inv = false;
            }
            listView.Sort (); // nezbytne pro aktualizaci
            showStatus ("Column " + e.Column);
        }

        private void runButton_Click (object sender, EventArgs e)
        {
            DriveInfo [] drives = DriveInfo.GetDrives ();
            foreach ( DriveInfo d in drives )
            {
                MyTreeNode n = new MyTreeNode ();
                n.Text = d.Name;
                n.Path = d.Name;
                try
                {
                    // n.ToolTipText = d.VolumeLabel;
                }
                catch ( Exception ex )
                {
                    n.ToolTipText = ex.Message;
                }
                treeView.Nodes.Add (n);

                // addSubdirs (n, 2);
            }
        }

        private void addSubdirs (MyTreeNode top, int level)
        {
            if (top.ready)
            {

```

```

        if (level > 1)
            foreach (MyTreeNode n in top.Nodes)
                addSubdirs (n, level-1);
    }
    else
    {
        top.ready = true;
        try
        {
            DirectoryInfo top_dir = new DirectoryInfo (top.Path);
            DirectoryInfo [] subdirs = top_dir.GetDirectories ();
            foreach (DirectoryInfo d in subdirs)
            {
                MyTreeNode n = new MyTreeNode ();
                n.Text = d.Name;
                n.ToolTipText = d.FullName;
                n.Path = d.FullName;
                top.Nodes.Add (n);
                showStatus (n.Path);

                if ( level > 1 )
                    addSubdirs (n, level-1);
            }
        }
        catch (Exception ex)
        {
            // top.ToolTipText = ex.Message;
            showStatus (ex.Message);
        }
    }
}

private void showFiles (MyTreeNode top)
{
    listView.Items.Clear ();
    try
    {
        DirectoryInfo top_dir = new DirectoryInfo (top.Path);
        FileInfo [] files = top_dir.GetFiles ();
        foreach (FileInfo f in files)
        {
            MyListNode n = new MyListNode ();

            n.FilePath = f.FullName;
            n.FileName = f.Name;
            n.FileSize = f.Length;
            n.FileTime = f.LastWriteTime;

            n.Text = n.FileName;
            n.ToolTipText = n.FilePath;

            n.SubItems.Add (n.FileSize.ToString ());

            // n.SubItems.Add (n.FileTime.ToString ("u"));
            n.SubItems.Add (n.FileTime.ToString ("dd-MM-yyyy HH:mm:ss"));

            listView.Items.Add (n);
        }
    }
    catch (Exception ex)
    {
        // top.ToolTipText = ex.Message;
        showStatus (ex.Message);
    }
}

private void showStatus (string s)
{
    statusField.Text = s;
}

```

```
}

private void treeView_BeforeSelect (object sender, TreeViewCancelEventArgs e)
{
    MyTreeNode n = e.Node as MyTreeNode;
    addSubdirs (n, 2);
    showStatus ("Before select " + n.Path);
}

private void treeView_BeforeExpand (object sender, TreeViewCancelEventArgs e)
{
    MyTreeNode n = e.Node as MyTreeNode;
    addSubdirs (n, 2);
    showStatus ("Before expand " + n.Path);
}

private void treeView_AfterSelect (object sender, TreeViewEventArgs e)
{
    MyTreeNode n = e.Node as MyTreeNode;
    showFiles (n);
    showStatus ("Selected " + n.Path);
}

private void treeView_AfterExpand (object sender, TreeViewEventArgs e)
{
    MyTreeNode n = e.Node as MyTreeNode;
    showStatus ("Expanded " + n.Path);
}

private MyListNode currentNode ()
{
    if (listView.SelectedItems.Count == 0)
        return null;
    else
        return listView.SelectedItems [0] as MyListNode;
}

private void listView_ItemActivate (object sender, EventArgs e)
{
    MyListNode n = currentNode ();
    if ( n == null )
    {
        showStatus ("No files selected");
    }
    else
    {
        showStatus ("Selected " + n.FilePath);
    }
}

private void listView_DoubleClick (object sender, EventArgs e)
{
    MyListNode n = currentNode ();
    if (n != null)
    {
        string ext = System.IO.Path.GetExtension (n.FileName);
        ext = ext.ToLower ();

        switch (ext)
        {
            case ".txt":
            case ".bat":
            case ".cs":
            case ".c":
            case ".cpp":
                richTextBox.LoadFile (n.FilePath, RichTextBoxStreamType.PlainText);
                tabControl.SelectedTab = textTab;
                break;
        }
    }
}
```

```

        case ".rtf":
            richTextBox.LoadFile (n.FilePath);
            tabControl.SelectedTab = textTab;
            break;

        case ".bmp":
        case ".jpg":
        case ".jpeg":
        case ".png":
        case ".gif":
            pictureBox.Image = new Bitmap (n.FilePath);
            tabControl.SelectedTab = pictureTab;
            break;

        case ".htm":
        case ".html":
            webBrowser.Url = new Uri ("file://" + n.FilePath);
            tabControl.SelectedTab = htmlTab;
            break;

        case ".xml":
            tabControl.SelectedTab = xmlTab;
            readXML (n.FilePath);
            break;

        default:
            tabControl.SelectedTab = dataTab;
            readData (n.FilePath);
            break;
    }
}

private void readXML (string fileName)
{
    richTextBox.LoadFile (fileName, RichTextBoxStreamType.PlainText);

    // http://support.microsoft.com/kb/307548

    TreeNode top = null;

    // usning System.Xml.XmlTextReader
    XmlTextReader reader = new XmlTextReader (fileName);
    while ( reader.Read () )
    {
        if ( reader.NodeType == XmlNodeType.Element)
        {
            TreeNode n = new TreeNode ();
            n.Text = "Element " + reader.Name;

            if ( top == null )
                treeBox.Nodes.Add (n);
            else
                top.Nodes.Add (n);

            top = n; // novy pracovni vrchol

            if (reader.HasAttributes)
                for ( int i = 0; i < reader.AttributeCount; i++ )
                {
                    TreeNode a = new TreeNode ();
                    a.Text = "Attribute " + reader.Name + " = " + reader.Value;
                    n.Nodes.Add (a);
                }
        }
        else if ( reader.NodeType == XmlNodeType.Text )
        {

```

```
        TreeNode n = new TreeNode ();
        n.Text = "Text " + reader.Value;

        if ( top == null )
            treeBox.Nodes.Add (n);
        else
            top.Nodes.Add (n);
    }
    else if ( reader.NodeType == XmlNodeType.EndElement )
    {

        if (top != null) // nemelo by byt treba
            top = top.Parent; // jsi o patro vys
    }

}

private void readData (string fileName)
{
    dataGridView.ColumnCount = 1;

    // string [] data = { "1", "22", "3" };
    // dataGridView.Rows.Add (data);

    StringBuilder buf = new StringBuilder (); // jedna polozka
    List<String> items = new List<string> (); // cely radek

    TextReader reader = new StreamReader (fileName);
    int n = reader.Read ();
    while ( n != -1 )
    {
        char c = (char) n;
        if ( c == ';' )
        {
            items.Add (buf.ToString ());
            buf = new StringBuilder (); // novy prazdny StringBuilder
        }
        else if ( c == '\n' )
        {
            string [] txt = items.ToArray ();

            if ( dataGridView.ColumnCount < txt.Length )
                dataGridView.ColumnCount = txt.Length;

            dataGridView.Rows.Add (txt);

            buf = new StringBuilder (); // novy prazdny StringBuilder
            items = new List<string> (); // novy radek
        }
        else if ( c != '\r' )
        {
            buf.Append (c);
        }

        n = reader.Read ();
    }
}

}
```