

C# 원품 TreeView Control

TreeView 메뉴 트리형태(계층적인 형태)로 아이템 컬렉션의 내용을 표시한다. 상위의 부모노드는 자식을 Nodes 속성을 통해 추가하면 된다.

[노드 추가, 제거]

```
TreeNode treeView1= new TreeNode("Text for new node");  
TreeNode newNode= new TreeNode("First Node");  
treeView1.SelectedNode.Nodes.Add(newNode);  
  
treeView1.Nodes.Remove(treeView1.SelectedNode);  
// Clears all nodes.  
TreeView1.Nodes.Clear();
```

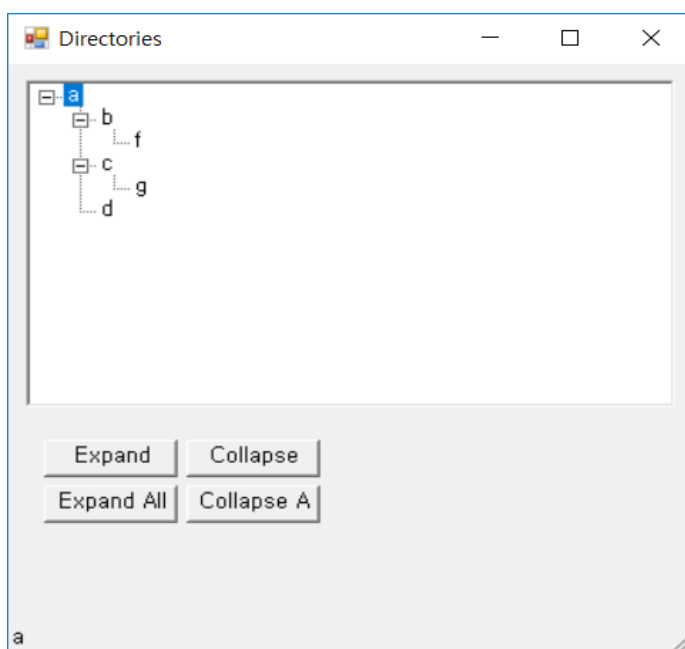
[TreeView에 이미지 표시]

TreeView 컨트롤의 ImageList 속성을 사용할 기존 ImageList 컨트롤로 설정한다.

이 속성은 디자이너의 속성 창을 사용하여 설정하거나 코드로 설정할 수 있다.

```
treeView1.ImageList = imageList1;  
// the TreeView control contains a selected image.)  
treeView1.SelectedNode.ImageIndex = 0;  
treeView1.SelectedNode.SelectedIndex = 1;
```

[예제]TreeView 메뉴를 이용한 디렉토리 표시



```

using System;
using System.Drawing;
using System.Windows.Forms;
using System.IO;

public class MForm : Form
{
    private TreeView tv;
    private Button expand;
    private Button expandAll;
    private Button collapse;
    private Button collapseAll;
    private StatusBar sb;

    private const string HOME_DIR = @"c:\Wa";

    public MForm()
    {
        Size = new Size(400, 400);
        Text = "Directories";

        tv = new TreeView();

        SuspendLayout();

        tv.Parent = this;
        tv.Location = new Point(10, 10);
        tv.Size = new Size(ClientSize.Width - 20, Height - 200);
        tv.Anchor = AnchorStyles.Top | AnchorStyles.Left |
            AnchorStyles.Right;

        tv.FullRowSelect = false;
        tv.ShowLines = true;
        tv.ShowPlusMinus = true;
        tv.Scrollable = true;
        tv.AfterSelect += new TreeViewEventHandler(AfterSelect);

        expand = new Button();
        expand.Parent = this;
        expand.Location = new Point(20, tv.Bottom + 20);
        expand.Text = "Expand";
        expand.Anchor = AnchorStyles.Left | AnchorStyles.Top;
        expand.Click += new EventHandler(OnExpand);

        expandAll = new Button();
        expandAll.Parent = this;
        expandAll.Location = new Point(20, expand.Bottom + 5);
        expandAll.Text = "Expand All";
        expandAll.Anchor = AnchorStyles.Left | AnchorStyles.Top;
        expandAll.Click += new EventHandler(OnExpandAll);
    }
}

```

```

collapse = new Button();
collapse.Parent = this;
collapse.Location = new Point(expandAll.Right + 5, expand.Top);
collapse.Text = "Collapse";
collapse.Anchor = AnchorStyles.Left | AnchorStyles.Top;
collapse.Click += new EventHandler(OnCollapse);

collapseAll = new Button();
collapseAll.Parent = this;
collapseAll.Location = new Point(collapse.Left, collapse.Bottom + 5);
collapseAll.Text = "Collapse All";
collapseAll.Anchor = AnchorStyles.Left | AnchorStyles.Top;
collapseAll.Click += new EventHandler(OnCollapseAll);

sb = new StatusBar();
sb.Parent = this;

ShowDirectories(tv.Nodes, HOME_DIR);

ResumeLayout();

CenterToScreen();
}

void AfterSelect(object sender, TreeViewEventArgs e)
{
    sb.Text = e.Node.Text;
}

void ShowDirectories(TreeNodeCollection trvNode, string path)
{
    DirectoryInfo dirInfo = new DirectoryInfo(path);
    if (dirInfo != null)
    {
        DirectoryInfo[] subDirs = dirInfo.GetDirectories();
        TreeNode tr = new TreeNode(dirInfo.Name);

        if (subDirs.Length > 0)
        {
            foreach (DirectoryInfo dr in subDirs)
            {
                if (!dr.Name.StartsWith("."))
                {
                    ShowDirectories(tr.Nodes, dr.FullName);
                }
            }
            trvNode.Add(tr);
        }
    }
}

```

```
void OnExpand(object sender, EventArgs e)
{
    tv.SelectedNode.Expand();
}

void OnExpandAll(object sender, EventArgs e)
{
    tv.ExpandAll();
}

void OnCollapse(object sender, EventArgs e)
{
    tv.SelectedNode.Collapse();
}

void OnCollapseAll(object sender, EventArgs e)
{
    tv.CollapseAll();
}

static void Main()
{
    Application.Run(new MForm());
}
}
```