

C# 系统应用之TreeView控件 (一).显示树状磁盘文件目录及加载图标

原创 Eastmount 最后发布于2014-02-19 15:30:15 阅读数 32786 ☆ 收藏

展开



Python+TensorFlow人工智能

该专栏为人工智能入门专栏,采用Python3和TensorFlow实现人工智能相关算法。前期介绍安装流程、基础语法...

Eastmount

¥9.90

去订阅

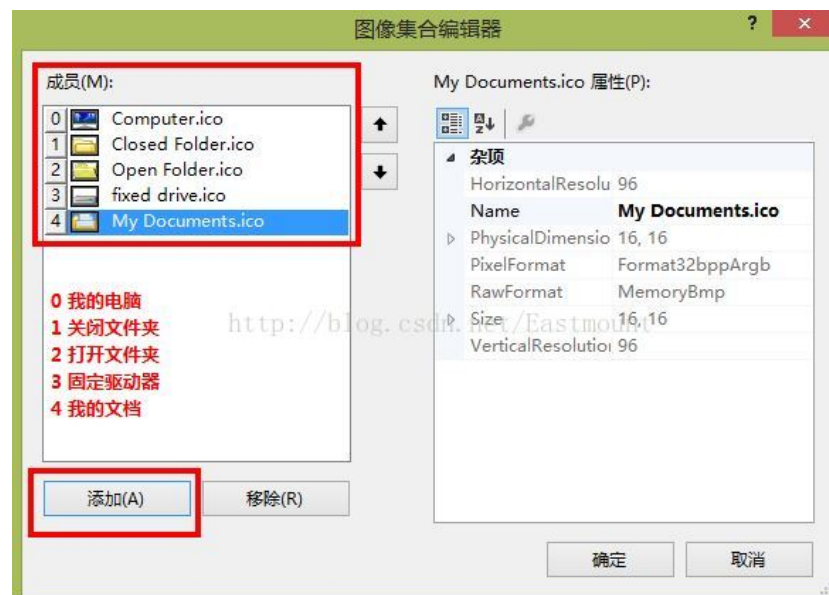
在C#系统应用毕设U盘防御软件中需要实现文件不可恢复的删除,首先需要实现类似于资源管理器的界面,通过TreeView控件显示"我的电脑"所有磁盘文件树状目录并加载相应图标.显示结果如下图所示:



一. 界面设计

主窗体是一个Windows窗体文件,工程命名为SecureTree.分别从工具箱中左边拖一个TreeView控件,名称(Name)为directoryTree;右侧文件列表ListView控件,名称为filesList.

然后从工具箱中拖两个ImageList控件,分别命名为directoryIcons和filesIcons.右键directoryIcons点击"选择图像",从本地Resource文件夹中添加图片如下图所示5张图片,同理filesIcons添加Closed Folder.ico图标.



添加完成后,指定directoryTree控件的ImageList属性为directoryIcons,filesList控件的SmallImageList属性为filesIcons绑定图标.我上传了常用的ICON图标
免费资源:<http://download.csdn.net/detail/eastmount/6934111>

二. 代码实现

1.添加IconIndexes枚举类型类,这个枚举类型值与之前添加到directoryIcons图标对应.这样TreeView控件便可显示该结点相对应的图标.

```
/// <summary>
/// IconIndexes类 对应ImageList中5张图片的序列
/// </summary>
private class IconIndexes
{
    public const int MyComputer = 0;        //我的电脑
    public const int ClosedFolder = 1;      //文件夹关闭
    public const int OpenFolder = 2;        //文件夹打开
    public const int FixedDrive = 3;        //磁盘盘符
    public const int MyDocuments = 4;      //我的文档
}
```

2.添加响应主窗体的Load事件,在窗体加载后,获取最顶层电脑驱动器和我的文档信息并加载到TreeView控件中.并添加命名空间:using System.IO;

```
/// <summary>
/// 窗体加载Load事件 初始化
/// </summary>
private void Form1_Load(object sender, EventArgs e)
{
    //实例化TreeNode类 TreeNode(string text,int imageIndex,int selectImageIndex)
    TreeNode rootNode = new TreeNode("我的电脑",
        IconIndexes.MyComputer, IconIndexes.MyComputer); //载入显示 选择显示
    rootNode.Tag = "我的电脑";                          //树节点数据
    rootNode.Text = "我的电脑";                          //树节点标签内容
    this.directoryTree.Nodes.Add(rootNode);              //树中添加根目录

    //显示MyDocuments(我的文档)结点
    var myDocuments = Environment.GetFolderPath          //获取计算机我的文档文件夹
        (Environment.SpecialFolder.MyDocuments);
    TreeNode DocNode = new TreeNode(myDocuments);
    DocNode.Tag = "我的文档";                            //设置结点名称
    DocNode.Text = "我的文档";
    DocNode.ImageIndex = IconIndexes.MyDocuments;        //设置获取结点显示图片
    DocNode.SelectedImageIndex = IconIndexes.MyDocuments; //设置选择显示图片
    rootNode.Nodes.Add(DocNode);                         //rootNode目录下加载节点
    DocNode.Nodes.Add("");

    //循环遍历计算机所有逻辑驱动器名称(盘符)
    foreach (string drive in Environment.GetLogicalDrives())
    {
        //实例化DriveInfo对象 命名空间System.IO
        var dir = new DriveInfo(drive);
        switch (dir.DriveType) //判断驱动器类型
        {
            case DriveType.Fixed: //仅取固定磁盘盘符 Removable-U盘
            {
                //Split仅获取盘符字母
                TreeNode tNode = new TreeNode(dir.Name.Split(':')[0]);
                tNode.Name = dir.Name;
                tNode.Tag = tNode.Name;
                tNode.ImageIndex = IconIndexes.FixedDrive; //获取结点显示图片
                tNode.SelectedImageIndex = IconIndexes.FixedDrive; //选择显示图片
            }
        }
    }
}
```

```

        directoryTree.Nodes.Add(tNode); // 加载驱动节点
        tNode.Nodes.Add("");
    }

    break;
}
}
rootNode.Expand(); // 展开树状视图
}

```

代码首先用TreeNode(string text,int imageIndex,int selectImageIndex)实例化一个根节点"我的电脑"(该结点没指定路径),其3个参数分别是:结点名称、加载时显示图标,选中结点时显示图标,通过IconIndexes类枚举访问.然后通过GetFolderPath(Environment.SpecialFolder.MyDocuments)获取MyDocuments(我的文档)文件夹路径,并添加为第一个子结点,同时调用foreach循环遍历逻辑磁盘盘符,DriveType判断固定磁盘加载至子结点.其中代码tNode.Nodes.Add("")是加载空节点,使"我的文档"和"CDEFG盘"显示+号.当点击加号时,后面的代码会清除该结点.

3.添加TreeView的AfterExpand事件和BeforeExpand事件.分别是展开结点后实现展开,将要展开前实现加载子目录结点.选择文件夹时会显示不同icon图标.

```

/// <summary>
/// 在结点展开后发生 展开子结点
/// </summary>
private void directoryTree_AfterExpand(object sender, TreeViewEventArgs e)
{
    e.Node.Expand();
}

/// <summary>
/// 在将要展开结点时发生 加载子结点
/// </summary>
private void directoryTree_BeforeExpand(object sender, TreeViewCancelEventArgs e)
{
    TreeViewItems.Add(e.Node);
}

/// <summary>
/// 自定义类TreeViewItems 调用其Add(TreeNode e)方法加载子目录
/// </summary>
public static class TreeViewItems
{
    public static void Add(TreeNode e)
    {
        //try..catch异常处理
        try
        {
            //判断"我的电脑"Tag 上面加载的该结点没指定其路径
            if (e.Tag.ToString() != "我的电脑")
            {
                e.Nodes.Clear(); // 清除空节点再加载子节点
                TreeNode tNode = e; // 获取选中\展开\折叠结点
                string path = tNode.Name; // 路径

                //获取"我的文档"路径
                if (e.Tag.ToString() == "我的文档")
                {
                    path = Environment.GetFolderPath // 获取计算机我的文档文件夹
                        (Environment.SpecialFolder.MyDocuments);
                }

                // 获取指定目录中的子目录名称并加载结点
                string[] dics = Directory.GetDirectories(path);
                foreach (string dic in dics)

```

```

        {
            TreeNode subNode = new TreeNode(new DirectoryInfo(dic).Name); // 实例化
            subNode.Name = new DirectoryInfo(dic).FullName; // 完整目录
            subNode.Tag = subNode.Name;
            subNode.ImageIndex = IconIndexes.ClosedFolder; // 获取节点显示图片
            subNode.SelectedImageIndex = IconIndexes.OpenFolder; // 选择节点显示图片
            tNode.Nodes.Add(subNode);
            subNode.Nodes.Add(""); // 加载空节点 实现+号
        }
    }
}
catch (Exception msg)
{
    MessageBox.Show(msg.Message); // 异常处理
}
}
}

```

注意事项:

- (1).该项目中"我的电脑"是作者自定义的root节点,没有设定其路径,所以BeforeExpand事件中会从它开始依次遍历,但"我的电脑"会提示"没有指定路径".故需要if(e.Tag.ToString() != "我的电脑")判断.同时"我的文档"需要再次获取其路径,依次实现Add子节点;
- (2).同时在"我的文档"和盘符中需要添加tNode.Nodes.Add("")加载空节点形成+号,如果没有该+号,BeforeExpand事件不会被调用,子目录无法获取加载,在BeforeExpand事件调用TreeViewItems.Add加载其子结点需要e.Nodes.Clear();清除该结点的子目录再加载.
- (3).提供两篇类似文章供大家学习,经过对比可以发现:第一篇仅从驱动器(C盘)开始加载,所以BeforeExpand简单展开子目录即可,不需要判断"我的电脑"和"我的文档".第二篇含"桌面",因此需要判断路径:"C# TreeView磁盘文件,AfterSelect显示加号-骆驼祥子" 和"Treeview树状显示文件夹".同时补充一篇很优秀的文章供大家学习"WinForm应用: ListView做图像浏览"
- (4).补充TreeView(树视图)事件:更详见"c# 树状视图(TreeView类)".

事件	描述
AfterCheck	在选中节点复选框后引发
AfterCollapse	在折叠一个节点后引发
AfterExpand	在扩展一个节点后引发
AfterSelect	在选中一个节点后引发
BeforeCheck	在选中节点复选框之前引发
BeforeCollapse	在折叠一个节点之前引发
BeforeExpand	在扩展一个节点之前引发
BeforeSelect	在选中一个节点之前引发

- (5).补充两个关于论坛讨论"c#怎样动态读取资源文件里的图片"和"在C#中怎么调用Resources文件中的图片"
- (6).在《C#典型模块与项目实战大全》(清华大学出版社-丁士锋)书中谈到,出于对程序响应性能考虑,它先加载盘符结点,没有使用递归一次性加载所有文件到树状列表中,代码通过AfterSelect事件和FileSystemWatcher控件,监听加载.并使用线程池Task更新加载TreeView,希望大家去学习.

三. 总结

该篇通过TreeView加载了磁盘目录路径,并通过ImageList加载图标.那么怎样实现阅读文件夹下文件,获取其图标、文件大小、扩展名等信息,并双击打开文件呢?下一篇将接着讲述.最后希望该文章对大家有所帮助,文章中很多链接都可以供觉得有用的同学学习,感谢上面提到的文章及书籍作者.同时如果文章中有错误或不足之处请原谅,有问题或建议者亦可提出.希望尊重作者劳动果实勿喷.

资源下载地址:<http://download.csdn.net/detail/eastmount/6937355>

想使用C++ MFC实现同样功能的文章见:<http://blog.csdn.net/eastmount/article/details/19120567>

(By:Eastmount 2014-2-19 下午3点<http://blog.csdn.net/eastmount/>)

👍 点赞 8 ☆ 收藏 ➦ 分享 ...



Eastmount  博客专家

发布了450 篇原创文章 · 获赞 6227 · 访问量 496万+

他的留言板

关注