

C# 系统应用之ListView控件 (三).添加ContextMenuStrip右键菜单打开删除文件

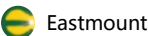
原创 Eastmount 最后发布于2014-05-28 20:53:08 阅读数 13011 ☆ 收藏

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...



¥9.90

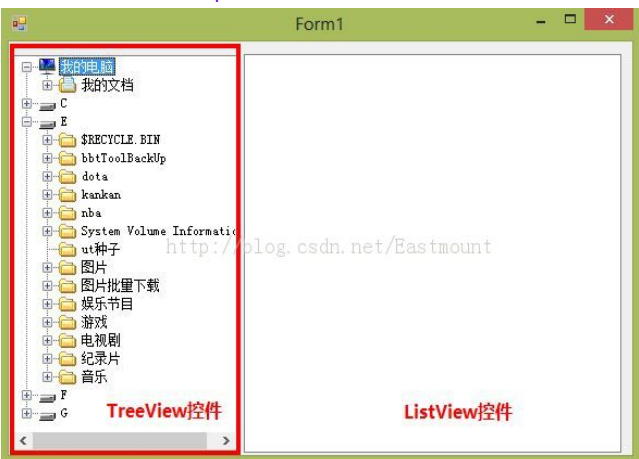
去订阅

在前面讲述过使用TreeView控件和ListView控件显示磁盘目录信息,但仅仅是显示信息是不够的,我们还需要具体的操作.在"个人电脑使用历史痕迹"项目中我还需要添加"打开文件"和"删除文件"两种方法.具体如下:

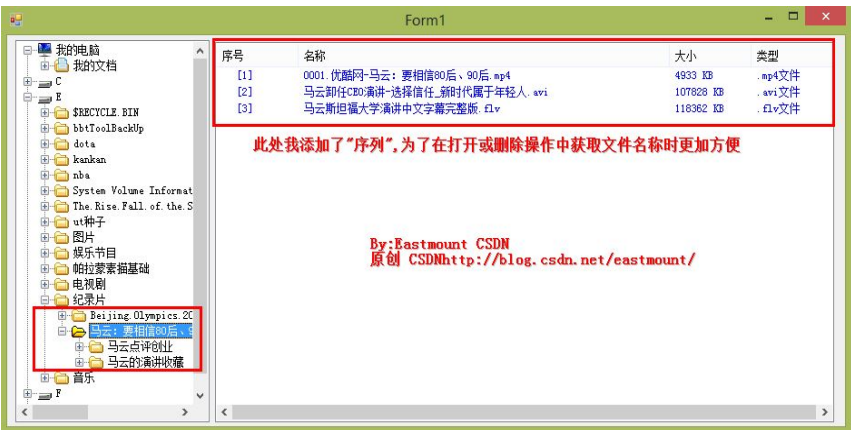
在第一篇文章"C# 系统应用之TreeView控件 (一).显示树状磁盘文件目录及加载图标"中显示如下:

<http://blog.csdn.net/eastmount/article/details/19453107>

免费下载地址: <http://download.csdn.net/detail/eastmount/7414709>



在第二篇文章"C# 系统应用之ListView控件 (二).加载选中节点文件夹下文件信息"中显示加载文件信息,如下图所示:<http://blog.csdn.net/eastmount/article/details/21241313>

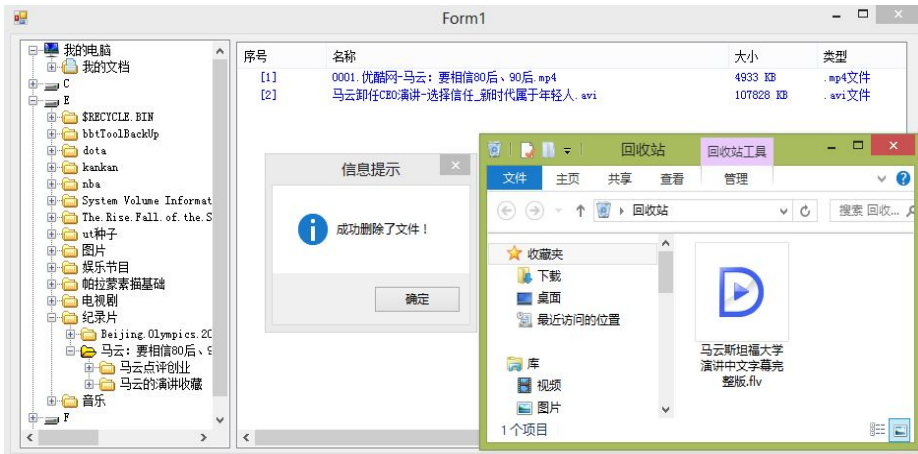


该文章第三篇主要讲述的是为ListView控件中显示的文件添加一个鼠标右键菜单"打开文件"和"删除文件",打开文件能打开具体的文件,删除文件能删除至回收站.效果图如下图所示:

打开文件



删除文件



一.添加控件ContextMenuStrip

首先添加一个ContextMenuStrip控件(右键关联控件显示相关菜单),并在"请在此处键入"处添加"打开文件"和"删除文件"两栏,将添加两项的Name设置为openFileToolStripMenuItem和deleteFileToolStripMenuItem.(否则为中英文名)

然后为filesList(ListView)控件添加鼠标事件,添加代码如下:

```
#region 鼠标右击事件 添加contextMenuStrip控件
private void filesList_MouseClick(object sender, MouseEventArgs e)
{
    // 禁止多选
    filesList.MultiSelect = false;
    // 鼠标右键
    if (e.Button == MouseButtons.Right)
    {
        //filesList.ContextMenuStrip = contextMenuStrip1;
        // 选中列表中数据才显示 空白处不显示
        String fileName = filesList.SelectedItems[0].Text; // 获取选中文件名
        Point p = new Point(e.X, e.Y);
        contextMenuStrip1.Show(filesList, p);
    }
}
#endregion
```

此时,运行软件点击文件就会添加了相应的菜单栏.

二.打开文件

现在需要添加打开文件功能.需要自定义两个全局变量记录打开文件的路径和名称.

```
// 全局变量
```

```
public string fileNamePublic = ""; // 文件名
public string filePathPublic = ""; // 文件路径
```

在更改选定内容AfterSelect事件中为选中的路径赋值,同时为openFileToolStripMenuItem添加Click事件,具体代码如下:

```
#region 打开文件操作
// 设置contextMenuStrip1控件中"打开文件"Name为openFile
private void openFileToolStripMenuItem_Click(object sender, EventArgs e)
{
    // 文件没有选中
    if (this.filesList.SelectedItems.Count == 0)
        return;
    // 获取选中文件
    var selectedItem = this.filesList.SelectedItems[0];
    // 全局变量文件名 SubItems[1]表示文件名
    fileNamePublic = filePathPublic + "\\\" + selectedItem.SubItems[1].Text;
    // MessageBox.Show(fileNamePublic);
    try
    {
        // 实例化一个新的Process类 命名空间using System.Diagnostics;
        using (Process p = new Process())
        {
            p.StartInfo.FileName = fileNamePublic; // 指定要启动的文件路径
            p.StartInfo.CreateNoWindow = false; // 在当前窗口启动程序
            // 指定窗口的显示样式
            p.StartInfo.WindowStyle = System.Diagnostics.ProcessWindowStyle.Normal;
            p.StartInfo.UseShellExecute = true; // 使用操作系统的shell启动进程
            p.Start(); // 开始打开文件
        }
    }
    catch (Exception msg) // 异常处理
    {
        MessageBox.Show(msg.Message);
    }
}
#endregion
```

它是通过实例化一个Process进程类,指定ProcessStartInfo参数,调用Process的Start()方法执行打开文件.

三.删除文件

删除文件开始准备的方法是使用File.Delete删除,但是删除的文件并不是添加至回收站的,同时系统文件不能访问,所以采用了SHFileOperation外壳函数实现文件操作,为deleteFileToolStripMenuItem添加Click事件,具体代码如下:

```
#region 删除文件
// 删除文件
// 文件删除到回收站中
private const int FO_DELETE = 3; // 删除
private const int FOF_SILENT = 0x0004; // 不显示进度条提示框
private const int FOF_NOCONFIRMATION = 0x0010; // 不出现任何对话框
private const int FOF_ALLOWUNDO = 0x0040; // 允许撤销
private const int FOF_NOCONFIRMMKDIR = 0x0200; // 创建文件夹的时候不用确认

// 添加命名空间using System.Runtime.InteropServices;
[StructLayout(LayoutKind.Sequential)]
private struct SHFILEOPSTRUCT
{
    public int hwnd; // 父窗口句柄, 0为桌面
    public int wFunc; // 功能标志 FO_COPY复制 FO_DELETE删除 FO_MOVE移动 FO_RENAME重命名
    public string pFrom; // source file源文件或者源文件夹
}
```

```

        public string pTo;                //destination目的文件或文件夹
        public int fFlags;                //控制文件的标志位 FOF_ALLOWUNDO 准许撤销 FOF_CONFIRMMOUSE 没有被使用
        public bool fAnyOperationsAborted;
        public int hNameMappings;
        public string lpszProgressTitle;
    }

    //SHFileOperation外壳函数 实现文件操作 参数SHFILEOPSTRUCT结构
    [DllImport("shell32.dll")]
    private static extern int SHFileOperation(ref SHFILEOPSTRUCT FileOp);

    /// <summary>
    /// 删除文件 Delete("c:\\test.txt",true) 把"c:/test.txt"删除到回收箱
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    private static int Delete(string sPath, bool recycle)
    {
        SHFILEOPSTRUCT FileOp = new SHFILEOPSTRUCT();
        FileOp.hwnd = 0;
        FileOp.wFunc = FO_DELETE;        //实现操作是删除文件
        FileOp.fFlags = 0;
        FileOp.fFlags = FileOp.fFlags | FOF_SILENT;
        FileOp.fFlags = FileOp.fFlags | FOF_NOCONFIRMATION;
        FileOp.fFlags = FileOp.fFlags | FOF_NOCONFIRMMKDIR;
        if (recycle)
        {
            FileOp.fFlags = FileOp.fFlags | FOF_ALLOWUNDO;
        }
        FileOp.pFrom = sPath + "\\0";
        return SHFileOperation(ref FileOp);
    }

    //设置contextMenuStrip1控件中"删除文件"Name为deleteFile
    private void deleteFileToolStripMenuItem_Click(object sender, EventArgs e)
    {
        //文件没有选中
        if (this.filesList.SelectedItems.Count == 0)
            return;
        //获取选中文件
        var selectedItem = this.filesList.SelectedItems[0];
        //全局变量文件名 SubItems[1]表示文件名
        fileNamePublic = filePathPublic + "\\\" + selectedItem.SubItems[1].Text;
        //MessageBox.Show(fileNamePublic);
        try
        {
            if (MessageBox.Show("确认要删除该最近浏览文件信息?", "提示",
                System.Windows.Forms.MessageBoxButtons.YesNo,
                System.Windows.Forms.MessageBoxIcon.Question) ==
                System.Windows.Forms.DialogResult.Yes)
            {
                //删除文件
                Delete(fileNamePublic, true);
                //移除文件
                foreach (ListViewItem item in this.filesList.SelectedItems)
                {
                    this.filesList.Items.Remove(item);
                }
            }
            MessageBox.Show(this, "成功删除了文件!", "信息提示", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
    }

```

```
        catch (Exception msg) // 异常处理    {
            MessageBox.Show(msg.Message);
        }
    }
#endregion
```

四.总结

最后,希望该文章对大家有所帮助.这三篇文章主要讲述了如何使用C#的TreeView控件加载磁盘目录信息,同时在ListView控件中显示具体信息,为ListView控件添加鼠标右键ContextMenuStrip控件菜单栏.在这几篇文章中你可以实现到项目中的:

1.如何使用TreeView控件添加图标,添加节点及其展开事件

2.如何使用ListView控件加载数据,设置标题头、添加数据、listItem.SubItems.Add等,尤其在数据库处理等操作或显示信息时,经常会应用到该控件显示内容

3.如何为控件添加右键菜单栏ContextMenuStrip,并设置其Item,为其添加相应的响应事件

最后,希望文章对大家有所帮助,如果文章中有错误或不足之处,还请海涵.我这一系列文章最后的运行效果如下图所示,高仿360同时能实现各种功能,具体参照博客每部分:



上面的打开删除文件源代码地址:<http://download.csdn.net/detail/eastmount/7414709>

```
//全局变量
public string fileNamePublic = ""; //文件名
public string filePathPublic = ""; //文件路径

TreeView树形显示磁盘下文件夹
设置ListView控件初始化
ListView显示选中文件夹中文件内容
鼠标右击事件 添加contextMenuStrip控件
打开文件操作
删除文件操作
```

(By:Eastmount 2014-5-28 夜8点 原创CSDN<http://blog.csdn.net/eastmount/>)

👍 点赞 4 ☆ 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