



细细品味 C#

——文件操作

精
华
集
锦

csAxp

虾皮工作室

<http://www.cnblogs.com/xia520pi/>

2011年10月4日

目录

1、文件操作之一.....	2
1.1、版权声明.....	2
1.2、内容详情.....	2
1.2.1 追加文件.....	2
1.2.2 拷贝文件.....	2
1.2.3 删除文件.....	2
1.2.4 移动文件.....	3
1.2.5 创建目录.....	3
1.2.6 文件夹内容拷贝.....	4
1.2.7 文件夹内容删除.....	5
2、文件操作之二.....	7
2.1、版权声明.....	7
2.2、内容详情.....	7
2.2.1 文件操作函数.....	7
2.2.2 目录操作实现.....	7
2.2.3 文件操作实现.....	8
2.2.4 编码转换操作.....	9
2.2.5 文本文件操作.....	9
2.2.6 流文件操作.....	10

1、文件操作之一

1.1、版权声明

文章出处：<http://kb.cnblogs.com/page/42225/>

文章作者：李超

1.2、内容详情

1.2.1 追加文件

//C#追加文件

```
StreamWriter sw = File.AppendText(Server.MapPath(".")+"\\myText.txt");
sw.WriteLine("追逐理想");
sw.WriteLine("kzlll");
sw.WriteLine(".NET 笔记");
sw.Flush();
sw.Close();
```

1.2.2 拷贝文件

//C#拷贝文件

```
string OrignFile,NewFile;
OrignFile = Server.MapPath(".")+"\\myText.txt";
NewFile = Server.MapPath(".")+"\\myTextCopy.txt";
File.Copy(OrignFile,NewFile,true);
```

1.2.3 删除文件

//C#删除文件

```
string delFile = Server.MapPath(".")+"\\myTextCopy.txt";
File.Delete(delFile);
```

1.2.4 移动文件

//C#移动文件

```
string OrignFile,NewFile;  
OrignFile = Server.MapPath(".")+"\\myText.txt";  
NewFile = Server.MapPath(".")+"\\myTextCopy.txt";  
File.Move(OrignFile,NewFile);
```

1.2.5 创建目录

//C#创建目录

// 创建目录 c:\sixAge

```
DirectoryInfo d=Directory.CreateDirectory("c:\\sixAge");
```

// d1 指向 c:\sixAge\sixAge1

```
DirectoryInfo d1=d.CreateSubdirectory("sixAge1");
```

// d2 指向 c:\sixAge\sixAge1\sixAge1_1

```
DirectoryInfo d2=d1.CreateSubdirectory("sixAge1_1");
```

// 将当前目录设为 c:\sixAge

```
Directory.SetCurrentDirectory("c:\\sixAge");
```

// 创建目录 c:\sixAge\sixAge2

```
Directory.CreateDirectory("sixAge2");
```

// 创建目录 c:\sixAge\sixAge2\sixAge2_1

```
Directory.CreateDirectory("sixAge2\\sixAge2_1");
```

//递归删除文件夹及文件

```
<% @ Page Language=C#%>
```

```
<%@ Import namespace="System.IO"%>
<Script runat=server>

public void DeleteFolder(string dir)
{
    if (Directory.Exists(dir)) //如果存在这个文件夹删除之
    {
        foreach(string d in Directory.GetFileSystemEntries(dir))
        {
            if(File.Exists(d))
                File.Delete(d); //直接删除其中的文件
            else
                DeleteFolder(d); //递归删除子文件夹
        }
        Directory.Delete(dir); //删除已空文件夹
        Response.Write(dir+" 文件夹删除成功");
    }
    else
        Response.Write(dir+" 该文件夹不存在"); //如果文件夹不存在则提示
}

protected void Page_Load (Object sender ,EventArgs e)
{
    string Dir="D:\\gbook\\11";
    DeleteFolder(Dir); //调用函数删除文件夹
}
```

1.2.6 文件夹内容拷贝

```
// =====
// 实现一个静态方法将指定文件夹下面的所有内容 copy 到目标文件夹下面
// 如果目标文件夹为只读属性就会报错。
// April 18April2005 In STU
// =====
```

```
Codepublic static void CopyDir(string srcPath,string aimPath)
{
    try
    {
        // 检查目标目录是否以目录分割字符结束如果不是则添加之
        if(aimPath[aimPath.Length-1] != Path.DirectorySeparatorChar)
            aimPath += Path.DirectorySeparatorChar;
    }
}
```

```

// 判断目标目录是否存在如果不存在则新建之
if(!Directory.Exists(aimPath))
    Directory.CreateDirectory(aimPath);

// 得到源目录的文件列表，该里面是包含文件以及目录路径的一个数组
// 如果你指向 copy 目标文件下面的文件而不包含目录请使用下面的方法
// string[] fileList = Directory.GetFiles(srcPath);
string[] fileList = Directory.GetFileSystemEntries(srcPath);

// 遍历所有的文件和目录
foreach(string file in fileList)
{
    // 先当作目录处理如果存在这个目录就递归 Copy 该目录下面的文件
    if(Directory.Exists(file))
        CopyDir(file,aimPath+Path.GetFileName(file));
    // 否则直接 Copy 文件
    else
        File.Copy(file,aimPath+Path.GetFileName(file),true);
}
}
catch (Exception e)
{
    MessageBox.Show (e.ToString());
}
}

```

1.2.7 文件夹内容删除

```

// =====
// 实现一个静态方法将指定文件夹下面的所有内容 Delete
// 测试的时候要小心操作，删除之后无法恢复。
// April 18April2005 In STU
// =====

```

```

Codepublic static void DeleteDir(string aimPath){
    try
    {
        // 检查目标目录是否以目录分割字符结束如果不是则添加之
        if(aimPath[aimPath.Length-1] != Path.DirectorySeparatorChar)
            aimPath += Path.DirectorySeparatorChar;
    }
}

```

```
// 得到源目录的文件列表, 该里面是包含文件以及目录路径的一个数组
// 如果你指向 Delete 目标文件下面的文件而不包含目录请使用下面的方法
// string[] fileList = Directory.GetFiles(aimPath);
string[] fileList = Directory.GetFileSystemEntries(aimPath);

// 遍历所有的文件和目录
foreach(string file in fileList)
{
    // 先当作目录处理如果存在这个目录就递归 Delete 该目录下面的文件
    if(Directory.Exists(file))
    {
        DeleteDir(aimPath+Path.GetFileName(file));
    }
    // 否则直接 Delete 文件
    else
    {
        File.Delete (aimPath+Path.GetFileName(file));
    }
}
//删除文件夹
System.IO .Directory .Delete (aimPath,true);
}
catch (Exception e)
{
    MessageBox.Show (e.ToString());
}
}
```

2、文件操作之二

2.1、版权声明

文章出处: http://blog.csdn.net/mane_yao/article/details/5886576

文章作者: mane_yao

2.2、内容详情

2.2.1 文件操作函数

若要执行此操作...	请参阅本主题中的示例...
创建文本文件	向文件写入文本
写入文本文件	向文件写入文本
读取文本文件	从文件读取文本
向文件中追加文本	File.AppendText FileInfo.AppendText
重命名或移动文件	File.Move FileInfo.MoveTo
删除文件	File.Delete FileInfo.Delete
复制文件	File.Copy FileInfo.CopyTo
获取文件大小	FileInfo.Length
获取文件属性	File.GetAttributes
设置文件属性	File.SetAttributes
确定文件是否存在	File.Exists
读取二进制文件	对刚创建的数据文件进行读取和写入
写入二进制文件	对刚创建的数据文件进行读取和写入
检索文件扩展名	Path.GetExtension
检索文件的完全限定路径	Path.GetFullPath
检索路径中的文件名和扩展名	Path.GetFileName
更改文件扩展名	Path.ChangeExtension

2.2.2 目录操作实现

System.IO 类

目录操作

```
string[] drives = Directory.GetLogicalDrives(); //本地驱动器的名,如:C:/等
string path = Directory.GetCurrentDirectory(); //获取应用程序的当前工作目录
Path.GetFileName(@"c:/dir/file.txt"); //获取子目录的名字,result的结果是 file.txt
```



```
Directory.GetFiles(路径及文件名) //获取指定目录中的文件名(文件列表)
DirectoryInfo di = new DirectoryInfo(@"f:/MyDir"); //构造函数创建目录
DirectoryInfo di=Directory.CreateDirectory(@"f:/bbs"); //创建对象并创建目录
if (di.Exists == false) //检查是否存在此目录
di.Create(); //创建目录
DirectoryInfo dis = di.CreateSubdirectory("SubDir"); //以相对路径创建子目录
dis.Delete(true); //删除刚创建的子目录
di.Delete(true); //删除创建目录
```

2.2.3 文件操作实现

//删除目录及其子目录和内容(如为假不能删除有内容的目录包括子目录)

```
Directory.Delete(@"f:/bbs2", true);
```

//获取指定目录中子目录的名称

```
Directory.GetDirectories 方法
string[] dirs = Directory.GetDirectories(@"f:/", "b*");
Console.WriteLine("此目录中以 b 开头的子目录共{0}个!", dirs.Length);
foreach (string dir in dirs) { Console.WriteLine(dir); }
```

//获取指定目录中的目录及文件名

```
Directory.GetFileSystemEntries
```

//检索此计算机上格式为“<驱动器号>:/”的逻辑驱动器的名称,【语法同上】

```
Directory.GetLogicalDrives
```

```
Directory.GetParent //用于检索父目录的路径。
DirectoryInfo a = Directory.GetParent(path);
Console.WriteLine(a.FullName);
Directory.Move //移动目录及其在内的所有文件
//将文件夹 1 内的文件剪到文件夹 2 内 文件夹 2 是刚创建的
Directory.Move(@"f:/bbs/1", @"f:/bbs/2");
Stream // 对字节的读写操作(包含对异步操作的支持) Reading Writing Seeking
BinaryReader 和 BinaryWriter // 从字符串或原始数据到各种流之间的读写操作
FileStream 类通过 Seek()方法进行对文件的随机访问,默认为同步
TextReader 和 TextWriter //用于 gb2312 字符的输入和输出
StreamReader 和 StreamWriter //在字符串中读写字符
```

```
StreamReader 和 StreamWriter //在流中读写字符  
BufferedStream 为诸如网络流的其它流添加缓冲的一种流类型。  
MemoryStream 无缓冲的流  
NetworkStream 互联网络上的流
```

2.2.4 编码转换操作

//编码转换

```
Encoding e1 = Encoding.Default; //取得本页默认代码  
Byte[] bytes = e1.GetBytes("中国人民解放军"); //转为二进制  
string str = Encoding.GetEncoding("UTF-8").GetString(bytes); //转回 UTF-8 编码
```

2.2.5 文本文件操作

//文本文件操作：创建/读取/拷贝/删除

```
using System;  
using System.IO;  
class Test  
{  
    string path = @"f:t.txt";  
    public static void Main()  
    {  
        //创建并写入(将覆盖已有文件)  
        if (!File.Exists(path))  
        {  
            using (StreamWriter sw = File.CreateText(path))  
            {  
                sw.WriteLine("Hello");  
            }  
        }  
        //读取文件  
        using (StreamReader sr = File.OpenText(path))  
        {  
            string s = "";  
            while ((s = sr.ReadLine()) != null)  
            {  
                Console.WriteLine(s);  
            }  
        }  
    }  
}
```

```
}  
//删除/拷贝  
try  
{  
    File.Delete(path);  
    File.Copy(path, @"f:/tt.txt");  
}  
catch (Exception e)  
{  
    Console.WriteLine("The process failed: {0}", e.ToString());  
}  
}
```

2.2.6 流文件操作

//流文件操作

```
private const string name = "Test.data";  
public static void Main(String[] args)  
{  
    //打开文件() ,或通过 File 创建如: fs = File.Create(path, 1024)  
    FileStream fs = new FileStream(name, FileMode.CreateNew);  
    //转换为字节 写入数据(可写入中文)  
    Byte[] info = new UTF8Encoding(true).GetBytes("This is some text in the file.");  
    //字节数组,字节偏移量,最多写入的字节数  
    fs.Write(info, 0, info.Length);  
    w.Close();  
    fs.Close();  
    //打开文件  
    fs = new FileStream(name, FileMode.Open, FileAccess.Read);  
    //读取  
    BinaryReader r = new BinaryReader(fs);  
    for (int i = 0; i < 11; i++)  
    {  
        Console.WriteLine(r.ReadInt32());  
    }  
    w.Close();  
    fs.Close();  
}
```