

### 一、简介

文件系统对象 FSO 的英文全称是 File System Object，这种对象模型提出了有别于传统的文件操作语句处理文件和文件夹的方法。通过采用 object.method 这种在面向对象编程中广泛使用的语法，将一系列操作文件和文件夹的动作通过调用对象本身的属性直接实现。

FSO 对象模型不仅可以象使用传统文件操作语句那样实现文件的创建、改变、移动和删除，而且可以检测是否存在指定的文件夹，如果存在，那么，这个文件夹又位于磁盘上的什么位置。更令人高兴的是 FSO 对象模型还可以获取关于文件和文件夹的信息，如名称、创建日期或最近修改日期等以及当前系统中使用的驱动器的信息，如驱动器的种类是 CD - ROM 还是可移动磁盘，当前磁盘的剩余空间还有多少。而以前要获取这些信息必须通过调用 Windows API 函数集中的相应函数才能实现。

FSO 对象模型包含在 Scripting 类型库 (Sccrun.Dll) 中，它同时包含了 Drive、Folder、File、FileSystemObject 和 TextStream 五个对象。其中 Drive 用来收集驱动器的信息，如可用磁盘空间或驱动器的类型；Folder 用于创建、删除或移动文件夹，同时可以进行向系统查询文件夹的路径等操作；File 的基本操作和 Folder 基本相同，所不同的是 Files 的操作主要是针对磁盘上的文件进行的；FileSystemObject 是 FSO 对象模型中最主要对象，它提供了一套完整的可用于创建、删除文件和文件夹，收集驱动器、文件夹、文件相关信息的方法。需要

注意的是， FSO 对象模型提供的方法是冗余的，也就是说在实际使用中， FSO 对象模型中包含的不同对象的不同方法进行的却是同样的操作，而且 FileSystemObject 对象的方法直接作用于其余对象，所以在后面的文章中并没有单独提到 FileSystemObject 对象，千万不要以为没有提到就不重要，事实上 FileSystemObject 对象在整个 FSO 对象模型中无处不在；最后的 TextStream 对象则是用来完成对文件的读写操作的。

在初步了解了 FSO 对象模型之后，下面我们通过实际的代码对不同的对象进行进一步的阐述。

## 二、 FSO 对象模型的应用

### （一）创建 FSO 对象模型

由于 FSO 对象包含在 Scripting 类型库 (Sccrun.Dll) 中，所以在使用前首先需要在在工程中引用这个文件，单击“工程”，“引用”，然后在“引用”对话框中选中“Microsoft Scripting Runtime”前的复选框，然后单击“确定”。

要创建 FSO 对象可以采用两种方法，一种是将一个变量声明为 FSO 对象类型：`Dim fsoTest As New FileSystemObject`；另一种是通过

CreateObject 方法创建一个 FSO 对象：`Set fsoTest =`

`CreateObject("Scripting.FileSystemObject")`。在实际使用中具体采

用哪种声明方法，可根据个人的使用习惯而定。

完成了 FSO 对象模型的创建之后，就可以利用创建的对象模型的方法

访问下属各个对象的属性来获取所需信息或进行相关操作了，具体

的方法在下面结合各个对象分别讲述。

## （二） Drive 对象

上面已经提到 Drive 对象是用来获取当前系统中各个驱动器的信息的。由于 Drive 对象没有方法，其应用都是通过属性表现出来的，所以我们必须熟悉 Drive 对象的属性：

AvailableSpace ：返回在指定的驱动器或网络共享上的用户可用的空间容量。

DriveLetter ：返回某个指定本地驱动器或网络驱动器的字母，这个属性是只读的。

DriveType ：返回指定驱动器的磁盘类型。

FileSystem ：返回指定驱动器使用的文件系统类型。

FreeSpace ：返回指定驱动器上或共享驱动器可用的磁盘空间，这个属性是只读的。

IsReady ：确定指定的驱动器是否准备好。

Path ：返回指定文件、文件夹、或驱动器的路径。

RootFolder ：返回一个 Folder 对象，该对象表示一个指定驱动器的根文件夹。只读属性。

SerialNumber ：返回用于唯一标识磁盘卷标的十进制序列号。

ShareName ：返回指定驱动器的网络共享名

TotalSize ：以字节为单位，返回驱动器或网络共享的总空间大小。

VolumeName ：设置或返回指定驱动器的卷标名。

从上面的属性可以看到 Drive 对象基本上包含了日常操作所需的全部

的驱动器信息， 因此在使用中是非常方便的。 下面通过一个实例讲述 Drive 对象的使用。首先在 VB 中建立一个工程，然后添加一个命令按钮，将其 Caption 设置为 “ TestDrive ， ” 然后在 click 事件中加入以下代码：

```
Dim fsoTest As New FileSystemObject

Dim drv1 As Drive, sReturn As String

Set drv1 = fsoTest.GetDrive(      \") " C:

sReturn =      " Drive & "      " C:\" &      vbCrLf

sReturn = sReturn      & " VolumeName" &      drv1.VolumeName      &

vbCrLf

sReturn = sReturn      & " Total Space: "      &

FormatNumber(drv1.TotalSize / 1024, 0)

sReturn = sReturn      & " Kb" &      vbCrLf

sReturn = sReturn      & " Free Space: "      &

FormatNumber(drv1.FreeSpace / 1024, 0)

sReturn = sReturn      & " Kb" &      vbCrLf

sReturn = sReturn      & " FileSystem:" &      drv1.FileSystem      &      vbCrLf

MsgBox sReturn
```

其中 GetDrive 方法返回一个与指定路径中的驱动器相对应的 Drive 对象。该方法的语法格式为 object.GetDrive drivespec ， object 是一个 FSO 对象的名称， drivespec 用于指定驱动器的名称。

按 F5 运行上述代码，按下 TestDrive 按钮就会弹出一个消息框显示

C 盘的信息。

### （三）Folder 对象

在 FSO 对象模型中，提供了丰富的有关文件夹操作的方法，这些方法分别是：

FileSystemObject 对象有关文件夹的方法：

CreateFolder ：创建一个文件夹 。

DeleteFolder ：删除一个文件夹 。

MoveFolder ：移动一个文件夹 。

CopyFolder ：复制一个文件夹 。

FolderExists ： 查找一个文件夹是否在驱动器上 。

GetFolder ：获得已有 Folder 对象的一个实例 。

GetParentFolderName ： 找出一个文件夹的父文件夹的名称。

GetSpecialFolder ： 找出系统文件夹的路径。

Folder 对象的方法：

Delete ：创建一个文件夹 。

Move ：移动一个文件夹 。

Copy ：复制一个文件夹。

Name ：检索文件夹的名称。

在此需要强调一点，前面我们曾经提到过 FSO 对象模型包含的方法是冗余的，所以 Folder 对象的 Delete 、 Move 、 Copy 方法和 FileSystemObject 对象的 DeleteFolder 、 MoveFolder 、 CopyFolder 方法实际上是相同的，因此在实际使用中可以任选其中的一种。

和 Drive 对象一样，下面通过实例演示 Folder 对象的应用。在 VB 下

新建一个工程，然后在上面添加三个命令按钮，然后在 Form1 的通用部分加入以下代码：

```
Option Explicit
```

```
Dim fsoTest As New FileSystemObject
```

```
Dim folder1 As Folder
```

并且分别在三个命令按钮的 click 事件输入以下代码：

```
Private Sub CmdCreate_Click()
```

```
' 获取 Folder 对象。
```

```
Set folder1 = fsoTest.GetFolder(" C:")
```

```
'创建文件夹
```

```
fsoTest.CreateFolder (" \Test") " C:
```

```
MsgBox " folder C:\Test has created"
```

```
End Sub
```

```
Private Sub CmdDelete_Click()
```

```
' 获取 Drive 对象。
```

```
Set folder1 = fsoTest .GetFolder(" C:")
```

```
'删除文件夹
```

```
fsoTest.DeleteFolder (" \Test") " C:
```

```
MsgBox " folder C:\Test has deleted"
```

```
End Sub
```

```
Private Sub CmdGetPro_Click()
```

'获取文件夹的有关信息

```
Dim sReturn As String
```

```
Set folder1 = fsoTest.GetFolder(           \Windows") " C:
```

```
'sReturn =      " The folder's Attributes is    & " folder1.Attributes    &  
vbCrLf
```

'获取最近一次访问的时间

```
sReturn = sReturn    & " The folder's last access time is "    &  
folder1.DateLastAccessed    & vbCrLf
```

'获取最后一次修改的时间

```
sReturn = sReturn    & " The folder's last modify time is "    &  
folder1.DateLastModified    & vbCrLf
```

'获取文件夹的大小

```
sReturn = sReturn    & " The folder's size is "    &  
FormatNumber(folder1.Size / 1024, 0) sReturn =  
sReturn    & " Kb" & vbCrLf
```

'判断文件或文件夹类型

```
sReturn = sReturn    & " The type is "    & folder1.Type    & vbCrLf  
MsgBox sReturn
```

```
End Sub
```

上述代码中提到的 CreateFolder 方法的语法形式为

object.CreateFolder(foldername) 。 foldername 指定了要创建的文件  
夹的名称，而 DeleteFolder 方法的语法形式为 object.DeleteFolder

folderspec[,force] , 其中, folderspec 用来指定要删除的文件夹的名称, force 是一个可选的布尔型参数, 如果希望删除只读属性的文件夹则将该值设为 TRUE , 默认为 FALSE 。至于代码中用到的 Folder 对象的属性限于篇幅, 就不详细介绍了, 读者可参考 VB 文档中的相关内容。

#### (四) File 对象和 TextStream 对象

由于有关 File 对象的复制, 删除, 移动等操作和 Folder 对象类似, 所以这部分内容就不再重复。这里主要讲述利用 File 对象和 TextStream 对象操作文本文件。

通常对文本文件的操作包括在文本文件中创建数据, 在文本文件中添加数据, 删除文本文件的数据等操作。这些操作都可以通过 File 对象和 FileSystemObject 对象的相关方法完成。不过在使用之前, 首先要创建一个文本文件, 这可以通过三种方法完成。一种方法是使用 FileSystemObject 对象的 CreateTextFile 方法。要创建一个空文本文件, 可以用以下语句:

```
Dim fsoTest As New FileSystemObject, fil1 As File
```

```
Set fil 1= fsoTest.CreateTextFile( \testfile ".c:txt", True)
```

第二种方法是使用 FileSystemObject 对象带 ForWriting 标志设置的 OpenTextFile 方法,

```
Dim fsoTest As New FileSystemObject, ts1 As New TextStream
```

```
Set ts1 = fsoTest.OpenTextFile( " \testfilec:.txt", ForWriting)
```

第三种方法是使用 File 对象的带 ForWriting 标志设置的



OpenAsTextStream      方法 :

Dim fsoTest As