Vbs脚本编程简明教程

Vbs脚本编程简明教程之一 为什么要使用Vbs

在Windows中，学习计算机操作也许很简单，但是很多计算机工作是重复性劳动，例如你每周也许需要对一些计算机文件进行复制、粘贴、改名、删除，也许你每天启动计算机第一件事情就是打开WORD，切换到你喜爱的输入法进行文本编辑，同时还要播放优美的音乐给工作创造一个舒心的环境，当然也有可能你经常需要对文本中的某些数据进行整理，把各式各样的数据按照某种规则排列起来……。这些事情重复、琐碎，使人容易疲劳。

第三方软件也许可以强化计算机的某些功能，但是解决这些重复劳动往往事倍功半，我也尝试过使用计算机语言编写程序来解决这些问题，但是随之而来的命令、语法、算法、系统框架和类库常常让我觉得这样是否有必要，难道就是因为猪毛比较难拔，所以我就要去学习机械，为自己设计一个拔猪毛机吗？

Vbs是一种Windows脚本，它的全称是:Microsoft Visual Basic Script Editon.(微软公司可视化BASIC脚本版)，VBS是Visual Basic的的一个抽象子集，是系统内置的，用它编写的脚本代码不能编译成二进制文件，直接由Windows系统执行（实际是一个叫做宿主host的解释源代码并执行），高效、易学，但是大部分高级语言能干的事情，它基本上都具备，它可以使各种各样的任务自动化，可以使你从重复琐碎的工作中解脱出来，极大的提高工作效率。

我个人认为Vbs脚本其实就是一种计算机编程语言，但是由于缺少计算机程序设计语言中的部分要素，对于事件的描述能力较弱，所以称为脚本，它最方便的地方就是提供了对COM对象的简便支持。那么什么是COM对象呢？

我这样理解，COM对象就是一些具有特定函数功能项程序模块，他们一般以ocx或者dll作为扩展名，你只要找到包含有你需要的功能的模块文件，并在脚本中规范的引用，就可以实现特定的功能，也就是说Vbs脚本就是调用现成的"控件"作为对象，用对象的属性和方法实现目的，完全免去了编写代码、设计算法等等麻烦。说白了，我不是觉得拔猪毛麻烦么？我发觉xx机（比如真空离心器）有一个功能可以实现脱毛，ok，我把它拿来给猪脱毛。什么？大材小用？太浪费资源了？天哪，那是计算机芯片的事情，死道友不死贫道，反正我的事情是方便快速的解决了，这就行了。

最方便的是它甚至不需要专门的开发环境，在你的计算机中，只要有notepad，就可以编写Vbs脚本了，并且可以直接执行。

Vbs脚本编程简明教程之二 开始第一个Vbs脚本

就像多数计算机教程一样 ，我们从"Hello World！"程序开始我们的练习。什么？不知道是什么意思？就是说大部分的计算机程序设计教程开篇入门都是编写一个小程序，执行这个程序的结果就是在计算机的屏幕上或者dos窗口中显示一行文字：HelloWorld！好了，我们开始吧。

打开你的"记事本"程序，在编辑窗口填写：msgbox "Hello World!"

然后用鼠标单击"文件"菜单，单击"保存"，把"保存在"一栏设为桌面，在"文件名"一栏中填写kk.vbs，单击"保存"就可以了。然后最小化"记事本"窗口，在桌面上寻找你刚刚保存的kk.vbs，然后双击。看到弹出的对话框了没有，单击"确定"，对话框消失了。难看了点，不过确实是你编写的第一个脚本程序。

说明之一：上面的操作中，保存位置放在桌面，仅仅是为了执行方便，你保存到其他的地方完全没有问题，只要你知道你保存在什么地方就可以了，什么？是废话，自己保存的当然知道保存在那里了。不，自己保存的文件自己找不到的人我见的多了去了。文件名你可以随意填写，不一定非要写kk，只要符合Windows的文件命名规则就可以了，但是扩展名必须是vbs，什么?不知道什么是扩展名？就是文件名中"."后的那部分，简单说，就是vbs脚本文件命名时必须是：xxx.vbs，其中xxx你随意。

说明之二：在记事本编辑窗口中写的这行是什么意思？

Msgbox是VBS内建的函数，每一个函数都可以完成一定的功能，你只需要按照语法要求，在函数的相应部分填写相应的内容就可以了，这部分内容我们称为参数，当然函数执行的结果我们称为返回值，一个函数可以有返回值也可以没有，可以有参数也可以没有。你不用了解函数是怎么运作的，只要了解这个函数能干什么就行了。

Msgbox语法：msgbox "对话框内容", , "对话框的标题"

你不妨用记事本打开刚才的文件在编辑窗口中输入：msgbox "Hello World!" , , "系统提示"

执行一下，看看效果和位置。

说明之三：如果执行失败，看看你的标点符号，所有的标点符号必须是在英文状态下输入的。

当然，这个脚本实在是太简单了，甚至连最简单的交互都没有，所以你可以把脚本这样修改一下：

Dim name

name=Inputbox("请输入你的名字:","名称")

Msgbox name, , "您的名字是"

保存执行一下，看到弹出的对话框了么？填入你的名字，点确定，看到结果了吗？

说明之一：第一句是定义变量，dim是定义变量的语句

其格式为：dim 变量1,变量2……，

Vbs只有一种变量类型，所以不用声明变量类型。系统会自动分辨变量类型。

说明之二：inputbox是VBS内建的函数，可以接受输入的内容，其语法格式为：

Inputbox("对话框内容","对话框标题")

第二句的意思是接受用户的输入，并把输入结果传递给变量name。

好了，到此脚本基本的输入输出函数都有了，已经可以完成一些比较简单的功能了，你可以编写一个简单的脚本，然后拷贝的"程序"—>"启动"中，然后重新启动计算机看看结果

Vbs脚本编程简明教程之三 Vbs的基本语法（牢记）

VBScript基础知识

一、变量

1、所有单引号后面的内容都被解释为注释。

2、在VBScript中，变量的命名规则遵循标准的命名规则，需要注意的是：在VBScript中对变量、方法、函数和对象的引用是不区分大小写的。在申明变量时，要显式地申明一个变量，需要使用关键字DIm来告诉VBScript你要创建一个变量，并将变量名称跟在其后。申明多个同类型变量，可以用逗号分隔。注意：VBScript中不允许在申明变量的时候同时给变量赋值。但是允许在一行代码内同时对两个变量进行赋值，中间用冒号分隔。

3、你可以使用OptionExplicit来告诉宿主变量必须先声明后使用。

4、VBScript在定义时只有一种变量类型，在实际使用中需要使用类型转换函数来将变量转换成相应的变量类型。

Cbool函数将变量转换成布尔值；

Cbyte函数将变量转换为0到255之间的整数。

Ccur函数、Cdbl函数和Csng函数将变量转换为浮点数值，前者只精确到小数点后四位，后两者要更加精确，数值的范围也要大的多。

Cdate函数将变量转换为日期值。

Cint函数和Clng函数将变量转换为整数，后者的范围比前者要大的多。

Cstr函数将变量转换为字符串。

二、数组

数组的定义与变量非常类似，只需要在变量后描述这个数组的个数和维数。需要注意的是：数组的下标总是从0开始，而以数组定义中数值减一结束。也就是说你以要定义一个有十个数据的数组，将这样书写代码：dImarray（9），同样，当你要访问第五个元素时，实际的代码是array(4)。当然，你可以通过不指定数组的个数和维数来申明动态数组。等到数组的个数和维数固定后，使用关键字redim来改变数组。注意，在改变数组的大小时，数组的数据会被破坏，使用关键字preserve来保护数据。例如：

RedIm空格preserve空格array括号个数逗号维数括号

三、操作符

在VBScript运算符中，加减乘除都是我们常用的符号，乘方使用的是 ^ ，取模使用的Mod。

在比较操作符中，等于、小于、大于、小于等于、大于等于都与我们常用的符号是一致的，而不等于是小于和大于连用。

逻辑运算符为：和操作—>AND 非操作—>NOT 或操作—>OR；

你可以使用操作符 + 和操作符 & 来连接字符串，一般使用&操作符；

另外还有一个比较特殊的操作符Is用来比较对象，例如按钮对象，如果对象是同一类型，结果就是真，如果对象不是同一类型，结果就是假。

四、条件语句主要有if……then语句和selectcase语句两种形式

在if……then语句中，其基本形式为：

If 条件 then

处理条件的语句；

……

Endif

基本形式只能对单个条件进行验证，如果有两个条件，则需要在基本形式中添加单行语句else，如果还有更多的条件需要验证，则需要添加语句

Elseif 条件 then

处理条件语句

在selectcase语句中，其基本形式为：

Select case 变量

Case 条件值

处理条件语句

并对上两句进行重复

最后一句应为

case else

处理语句

当然不要忘记将条件结束语句End select放在最后一行

注意：在执行字符串比较时，需要特别注意大小写，一般情况下，我们在比较前，使用lcase函数将字符串转换成小写，使用ucase函数将字符串转换成大写大写。

五、循环控制语句

循环控制语句有for……next循环、for……each循环、do……while循环、do……until循环、while循环五种形式。

在使用循环控制语句前，首先要对循环条件进行判断，如果循环次数是有固定次数的，那么使用For……next循环，其结构为：

For 计数器变量＝开始计数值 to 最后计数值

执行循环体

Next

如果是需要对数组或对象集合中的每一个元素进行判断，则需要使用for……each循环，其结构为：

For each 循环计数变量 in 要查看的对象或数组

执行处理语句

Next

注意：在上述两种循环中随时可以使用exit for来退出循环

如果你希望在条件满足时执行一段代码则使用do……while语句，结构为：

Do while 条件

执行循环体

Loop

如果你希望在条件不满足时执行代码，则使用do……until语句，结构为：

Do　until　条件

执行循环体

Loop

当然，在这两种循环语句中，你可以使用exit do来退出循环

最后一种循环语句是条件满足时一直执行循环，

While 条件

执行循环体

Wend

六、使用过程

常用的过程有两种，一种为函数，给调用者返回值，一种为子程序，无返回值，还有一种叫事件的特殊子程序，用的比较少。

函数的基本定义方法为：

Function 函数名称（参数列表）

函数代码

函数名称＝某值 ‘用来返回值

end function

子程序一些都类似，不过没有返回值

注意：尽管在定义子程序的时候，参数列表要加括号，但在调用子程序的时候，参数列表不加括号，括号只在函数中使用。另外，子程序不能在表达式中使用。

而函数只能出现在赋值语句的右边，或者表达式中，函数不能直接使用，如果必须直接使用函数，则必须使用call语句调用，并取消返回值

Vbs脚本编程简明教程之四 利用Vbs运行外部程序

Vbs只提供了编程的一个基本框架，用户可以使用Vbs来定义变量、过程和函数，vbs也提供了一些内部函数和对象，但是Vbs没有提供任何命令来访问Windows系统内部的部件，但是值得庆幸的是，Vbs虽然不能自己完成这些任务，但是它提供了一条极为方便、功能也相当强的命令——CreateObject，这条命令可以访问windows系统内安装的所有com对象，并且可以调用这些部件中存放的命令。

于是问题解决了，比如说，我手头有1000个小文本，我首先要对每一个文本的语法进行查错和修改，然后按照预先定义好的规则对这些文本进行排序，最后将这些文本合并成为一个文件。正常情况下，我们需要把打开第一个小文本，然后把它复制到WORD中，然后利用里面的除错功能进行除错和修改，然后再导入到EXCEL中进行排序，将这个过程重复1000遍，然后再将所有得到的文本复制到一个大文本中。实在是太枯燥、工作量太大了。有了Vbs和CreateObject，问题得到解决，我只需要找到相应的模块，调用相应的功能就可以了，作为脚本，把一个枯燥的过程重复1000次，本就是它的拿手好戏。

好了，我们走入正题，从最简单的——只启动一个程序开始。

WSH也就是用来解析Vbs的宿主，本身包含了几个个常用对象：

1、Scripting.FileSystemObject —> 提供一整套文件系统操作函数

2、Scripting.Dictionary —> 用来返回存放键值对的字典对象

3、Wscript.Shell —> 提供一套读取系统信息的函数，如读写注册表、查找指定文件的路径、读取DOS环境变量，读取链接中的设置

4、Wscript.NetWork —> 提供网络连接和远程打印机管理的函数。（其中，所有Scripting对象都存放在SCRRUN.DLL文件中，所有的Wscript对象都存放在WSHOM.ocx文件中。）

现在我们需要的是第三个对象，好了，让我们先连接一下对象看看，在记事本的编辑窗口中输入：

Set objShell = CreateObject("Wscript.Shell")

objShell.Run "notepad"

同样，保存执行。那么看到了一个什么样的结果呢？在桌面上又打开了一个记事本。

说明之一：Set是Vbs指令，凡是将一对象引用赋给变量，就需要使用set关键字。那么什么是对象引用呢？凡是字符串、数值、布尔值之外的变量都是对象引用。Objshell是变量名，可以随意修改。

说明之二：反是正确引用的对象，其本身内置有函数和变量，其引用方法为在变量后加". "，后紧跟其实现功能的函数就可以了。Objshell.run 的意思就是调用Wscript.shell中的运行外部程序的函数——run，notepad是记事本程序的文件名。当然你也可以改成"calc"，这是计算器的文件名，winword是word的文件名，等等吧，所有可执行文件的文件名都可以。但是需要注意的是，如果你要执行的可执行文件存放的地方不是程序安装的常用路径，一般情况下，需要提供合法的路径名，但是run在运行解析时，遇到空格会停止，解决的方法是使用双引号，例如：在我的机器上运行qq，代码为：

objshell.run """C:\Program Files\QQ2006\QQ.exe""" ‘注：三个引号

好， 我们再进一步，启动两个程序会如何呢？

输入如下代码：

Set objShell = CreateObject("Wscript.Shell")

objShell.Run "notepad"

objShell.Run "calc"

执行会如何呢？两个程序基本上同时启动了。如果我们需要先启动notepad再启动calc将如何呢？很简单在需要顺序执行的代码后加 , , True参数就可以了。

好了输入代码：

Set objShell = CreateObject("Wscript.Shell")

objShell.Run "notepad" ,,true

objShell.Run "calc"

看看执行的结果怎么样吧！

总结：run函数有三个参数，第一个参数是你要执行的程序的路径，第二个程序是窗口的形式，0是在后台运行；1表示正常运行；2表示激活程序并且显示为最小化；3表示激活程序并且显示为最大化；一共有10个这样的参数我只列出了4个最常用的。第三个参数是表示这个脚本是等待还是继续执行，如果设为了true,脚本就会等待调用的程序退出后再向后执行。

其实，run做为函数，前面还有一个接受返回值的变量，一般来说如果返回为0，表示成功执行，如果不为0，则这个返回值就是错误代码，可以通过这个代码找出相应的错误

Vbs脚本编程简明教程之五 错误处理

引发错误的原因有很多，例如用户输入了错误类型的值，或者脚本找不到必需的文件、目录或者驱动器，我们可以使用循环技术来处理错误，但是VBS本身也提供了一些基本技术来进行错误的检测和处理。

1、最常见的错误是运行时错误，也就是说错误在脚本正在运行的时候发生，是脚本试图进行非法操作的结果。例如零被作为除数。在vbs中，任何运行时错误都是致命的，此时，脚本将停止运行，并在屏幕上显示一个错误消息。你可以在脚本的开头添加

On Error Resume Next

这行语句可以告诉vbs在运行时跳过发生错误的语句，紧接着执行跟在它后面的语句。

发生错误时，该语句将会把相关的错误号、错误描述和相关源代码压入错误堆栈。

2、虽然On Error Resume

Next语句可以防止vbs脚本在发生错误时停止运行，但是它并不能真正处理错误，要处理错误，你需要在脚本中增加一些语句，用来检查错误条件并在错误发生时处理它。

vbscript提供了一个对象err对象，他有两个方法clear，raise，5个属性：description，helpcontext，helpfile，number，source

err对象不用引用实例，可以直接使用，例如：

on error resume next

a=11

b=0

c=a/b

if err.number<>0 then

wscript.echo err.number & err.description & err.source

end if

Vbs脚本编程简明教程之六 修改注册表

Vbs中修改注册表的语句主要有：

1、读注册表的关键词和值：

可以通过把关键词的完整路径传递给wshshell对象的regread方法。例如：

set ws=wscript.createobject("wscript.shell")

v=ws.regread("HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\nwiz")

wscript.echo v

2、写注册表

使用wshshell对象的regwrite方法。例子：

path="HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\"

set ws=wscript.createobject("wscript.shell")

t=ws.regwrite(path & "jj","hello")

这样就把HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\jj这个键值改成了hello.不过要注意：这个键值一定要预先存在。

如果要创建一个新的关键词，同样也是用这个方法。

path="HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\run\sssa2000\love\"

set ws=wscript.createobject("wscript.shell")

val=ws.regwrite(path,"nenboy")

val=ws.regread(path)

wscript.echo val

删除关键字和值

使用regdelete方法，把完整的路径传递给regdelete就可以了

例如

val=ws.regdel(path)

注意，如果要删除关键词的值的话 一定要在路径最后加上"\"，如果不加斜线，就会删除整个关键词。

类 型类型索引大 小说 明

REG\_BINARY30~多个字节可以包含任何数据的二进制对象颜色描述

REG\_COLOR\_RGB\*4字节颜色描述

REG\_DWORD44字节DWORD值

REG\_DWORD\_BIG\_ENDIAN54字节一个DWORD值的逆序存储形式

REG\_DWORD\_LITTLE\_ENDIAN44字节DWORD值

REG\_EXPAND\_SZ20~多个字节包含环境变量占位符的字符串

REG\_FILE\_NAME\*0~多个字节文件名

REG\_FILE\_TIME\*未知文件时间

REG\_FULL\_RESOURCE\_DESCRIPTOR9未知硬件资源列表

REG\_LINK60~多个字节命名一个符号链接(symbolic link)的Unicode串

REG\_MULTI\_SZ70~多个字节以null字符分隔的字符串集合，集合中的最后一个字符串以两个null字符结尾

REG\_NONE0未知REG\_NONE类型的对象无须分类，它与REG\_UNKNOWN类型不同

REG\_RESOURCE\_LIST8未知设备使用的资源列表

REG\_RESOURCE\_REQUIREMENTS\_

LIST10未知驱动程序要求的资源列表

REG\_SZ10~多个字节以一个null字符线的字符串

REG\_UNKNOWN（未定义）未知因为数据类型索引无效而无法确定的对象类型

Vbs脚本编程简明教程之七 FSO的常见对象和方法

文件系统是所有操作系统最重要的部分之一，脚本经常会需要对文件及文件夹进行访问和管理，在Vbs中对桌面和文件系统进行访问的顶级对象是FileSystemObject(FSO)，这个对象特别复杂，是vbs进行文件操作的核心。此节内容应了如指掌。

FSO包含的常见对象有：

Drive对象：包含储存设备的信息，包括硬盘、光驱、ram盘、网络驱动器

Drives集合：提供一个物理和逻辑驱动器的列表

File 对象：检查和处理文件

Files 集合：提供一个文件夹中的文件列表

Folder对象：检查和处理文件夹

Folders集合：提供文件夹中子文件夹的列表

Textstream对象：读写文本文件

FSO的常见方法有：

BulidPath：把文件路径信息添加到现有的文件路径上

CopyFile：复制文件

CopyFolder：复制文件夹

CreateFolder：创建文件夹

CreateTextFile：创建文本并返回一个TextStream对象

DeleteFile：删除文件

DeleteFolder：删除文件夹及其中所有内容

DriveExits：确定驱动器是否存在

FileExits：确定一个文件是否存在

FolderExists：确定某文件夹是否存在

GetAbsolutePathName：返回一个文件夹或文件的绝对路径

GetBaseName：返回一个文件或文件夹的基本路径

GetDrive：返回一个dreve对象

GetDriveName：返回一个驱动器的名字

GetExtensionName：返回扩展名

GetFile：返回一个file对象

GetFileName：返回文件夹中文件名称

GetFolder：返回一个文件夹对象

GetParentFolderName：返回一个文件夹的父文件夹

GetSpecialFolder:返回指向一个特殊文件夹的对象指针

GetTempName:返回一个可以被createtextfile使用的随机产生的文件或文件夹的名称

MoveFile：移动文件

MoveFolder：移动文件夹

OpenTextFile：打开一个存在的文件并返回一个TextStream对象

Vbs脚本编程简明教程之八 FSO的基本操作

文件夹操作

1、使用fso

由于fso不是wsh的一部分，所以我们需要建立他的模型

例如set fs=wscript.createobject("scripting.filesystemobject")

这样就建立了fso的模型。如果要释放的话也很简单，set fs=nothing

2、使用文件夹

在创建前，我们一般需要检查该文件夹是否存在例如：

dim fs,s //定义fs、s两个变量

set fs=wscript.createobject("scripting.filesystemobject") //fs为FSO实例

if (fs.folderexists("c:\temp")) then //判断c:\temp文件夹是否存在

s="is available"

else

s="not exist"

set foldr=fs.createfolder("c:\temp") //不存在则建立

end if

删除： set fs=wscript.createobject("scripting.filesystemobject")

fs.deletefolder("c:\windows")

拷贝： set fs=wscript.createobject("scripting.filesystemobject")

fs.copyfolder "c:\data" "d:\data"

注意：如果c:\data 和d:\data都存在，脚本会出错，复制也就会停止，如果要强制覆盖，使用fs.copyfolder "c:\data"

"d:\data"，true

移动： set fs=wscript.createobject("scripting.filesystemobject")

fs.movefolder "c:\data" "d:\data"

我们可以使用统配符，来方便操作：

例如， fs.movefolder :c:\data\te\*" , "d:\working"

注意：在目的路径最后没有使用"\" 也就是说我没有这样写：

fs.movefolder c:\data\te\*" , "d:\working\"

这样写的话，如果d:\working 目录不存在，windows就不会为我们自动创建这个目录。

注意：上面我们所举的例子都是在利用fso提供的方法，如果使用folder对象也完全是可以的：

set fs= wscript.createobject("scripting.filesystemobject")

set f=fs.getfolder("c:\data")

f.delete //删除文件夹c:\data。如果有子目录，也会被删除

f.copy "d:\working",true //拷贝到d:\working

f.move "d:\temp" //移动到d:\temp

3、特殊文件夹

一般指的就是系统文件夹：\windows\system32，临时文件夹，windows文件夹，在前几篇的时候，我们提过一下：例如

set fs=wscript.createobject("scripting.filesystemobject")

set wshshell=wscript.createobject("wscript.shell")

osdir=wshshell.expandenvironmentstrings("%systemroot%")

set f =fs.getfolder(osdir)

wscript.echo f

当然，还有简单的方法 那就是使用getspecialfolder()

这个方法使用3种值：

0 表示windows文件夹，相关常量是windowsfolder

1 系统文件夹，相关常量是systemfolder

2 临时目录，相关常量temporaryfolder

例如：

set fs=wscript.createobject("scripting.filesystemobject")

set wfolder=fs.getspecialfolder(0) ‘返回windows目录

set wfolder=fs.getspecialfolder(1) ‘返回system32\

set wfolder=fs.getspecialfolder(2)'返回临时目录

文件的操作

一、文件属性：

在windows中，文件的属性一般用数字来表示：

0代表normal，即普通文件未设置任何属性。 1代表只读文件。2代表隐藏文件。 4代表系统文件。 16代表文件夹或目录。32代表存档文件。 1024代表链接或快捷方式。例如：

set fs=wscript.createobject("scripting.filesystemobject")

set f=fs.getfile("d:\index.txt")

msgbox f.Attributes ‘attributes函数的作用是显示文件属性

需要说明的是：msgbox显示的结果往往不是上面说明的数字，而是有关属性代表数字的和。

二、创建文件：object.createtextfile方法，注意创建前一般需要检查文件是否存在。

例如：set fso=wscript.createobject("scripting.filesystemobject")

if fso.fileexists("c:\kk.txt") then

msgbox "文件已存在"

else

set f=fso.createtextfile("c:\kk.txt")

end if

如需要强制覆盖已存在的文件，则在文件名后加true参数。

三、复制、移动、删除文件：使用copyfile方法、movefile方法、deletefile方法。例如：

set fso=wscript.createobject("scripting.filesystemobject")

fso.copyfile "c:\kk.txt","d:\1\kk.txt",true //如上文说述，true代表强制覆盖

fso.movefile "c:\kk.txt", "d:\" //移动文件

fso.deletefile "c:\kk.txt" //删除文件

四、文件的读写：

1、打开文件：使用opentextfile方法

如：set ts=fso.opentextfile("c:\kk.txt",1,true)

说明：第二个参数为访问模式1为只读、2写入、8为追加

第三个参数指定如文件不存在则创建。

2、读取文件：read(x)读x个字符；readline读一行；readall全部读取

如：set ffile=fso.opentextfile("c:\kk.txt",1,true)

value=ffile.read(20)

line=ffile.readline

contents=ffile.readall

3、常见的指针变量：

atendofstream属性：当处于文件结尾的时候这个属性返回true。一般用循环检测是否到达文件末尾。例如：

do while ffile.atendofstream<>true

ffile.read(10)

loop

atendofline属性：如果已经到了行末尾，这个属性返回true。

Column属性(当前字符位置的列号)和line属性(文件当前行号)：在打开一个文件后，行和列指针都被设置为1。

4、在文件中跳行：skip(x) 跳过x个字符；skipline 跳过一行

5、在文件中写入字符：可以用2－写入和8－追加的方式来写入

其方法有：write(x)写入x字符串；writeline(x)写入x代表的一行

writeblanklines(n) 写入n个空行

注意：最后一定要使用close方法关闭文件。

Vbs脚本编程简明教程之九 妙用SendKeys简化重复

每次开机的时候，你想自动登陆你的QQ或者网志吗？巧妙使用VBS中的SendKeys命令（这个命令的作用就是模拟键盘操作，将一个或多个按键指令发送到指定Windows窗口来控制应用程序运行），可以极大的方便我们的常用操作。其使用格式为：

Object.SendKeys string其中：

Object：为WshShell对象，即脚本的第一行为：

Set WshShell=WScript.CreateObject("WScript.Shell")

将Object替换为WshShell

"string"：表示要发送的按键指令字符串，需要放在英文双引号中。它包含如下内容：

1．基本键：一般来说，要发送的按键指令都可以直接用该按键字符本身来表示，例如要发送字母"x"，使用"WshShell.SendKeys "x""即可。当然，也可直接发送多个按键指令，只需要将按键字符按顺序排列在一起即可，例如，要发送按键"cfan"，可以使用"WshShell.SendKeys "cfan""。

2．特殊功能键：对于需要与Shift、Ctrl、Alt三个控制键组合的按键，SendKeys使用特殊字符来表示：Shift —— +；Ctrl —— ^；Alt —— %

如要发送的组合按键是同时按下Ctrl＋E，需要用"WshShell.SendKeys "^e""表示，如果要发送的组合按键是按住Ctrl键的同时按下E与C两个键，这时应使用小括号把字母键括起来，书写格式为"WshShell.SendKeys

"^(ec)""，这里要注意它与"WshShell.SendKeys "^ec""的区别，后者表示组合按键是同时按住Ctrl和E键，然后松开Ctrl键，单独按下"C"字母键。

由于"+"、"^"这些字符用来表示特殊的控制按键了，如何表示这些按键呢？只要用大括号括住这些字符即可。例如，要发送加号"+"，可使用"WshShell.SendKeys "{+}""。另外对于一些不会生成字符的控制功能按键，也同样需要使用大括号括起来按键的名称，例如要发送回车键，需要用"WshShell.SendKeys "{ENTER}""表示，发送向下的方向键用"WshShell.SendKeys "{DOWN}""表示。

如果需要发送多个重复的单字母按键，不必重复输入该字母，SendKeys允许使用简化格式进行描述，使用格式为"{按键数字}"。例如要发送10个字母"x"，则输入"WshShell.SendKeys "{x 10}""即可。

例一：WshShell.SendKeys "^{ESC}u"

代码的含义为：按下Ctrl＋Esc组合键（相当于按Win键）打开"开始"菜单，接着按U键打开"关机"菜单。

例二：让VBS脚本自动在记事本中输入一行文字"hello, welcome to cfan"。

Dim WshShell

Set WshShell=WScript.CreateObject("WScript.Shell")

WshShell.Run "notepad"

WScript.Sleep 2000 //本行的含义为是脚本暂停2秒，给notepad一个打开的时间，有时时间太短可能导致后面的字符无法进入编辑区

WshShell.AppActivate "无标题 - 记事本"//AppActivate为寻找可执行程序的标题框，"无标题－记事本"内容你的自己打开看一下

WshShell.SendKeys "hello, welcome to cfan"

作业1:让脚本自动输入下面两段小短句

This is the most wonderful day of my life

because I'm here with you now

作业2：让脚本在输入短句后自动关闭记事本，并保存文件名为"test"，注意关闭记事本可以直接使用组合按键Alt＋F4来实现

例三：制作能自动定时存盘的记事本

我们最常用的记事本没有Word、WPS那样的自动定时存盘功能，其实利用VBS脚本再加上SendKeys命令，就能弥补这个遗憾。打开记事本，输入以下内容（为容易描述和分析，把代码分为四个部分）：

'第一部分：定义变量和对象

Dim WshShell, AutoSaveTime, TXTFileName

AutoSaveTime=300000

Set WshShell=WScript.CreateObject("WScript.Shell")

TXTFileName=InputBox("请输入你要创建的文件名(不能用中文和纯数字)：")

'第二部分：打开并激活记事本

WshShell.Run "notepad"

WScript.Sleep 200

WshShell.AppActivate "无标题 - 记事本"

'第三部分：用输入的文件名存盘

WshShell.SendKeys "^s"

WScript.Sleep 300

WshShell.SendKeys TXTFileName

WScript.Sleep 300

WshShell.SendKeys "%s"

WScript.Sleep AutoSaveTime

'第四部分：自动定时存盘

While WshShell.AppActivate (TXTFileName)=True

WshShell.SendKeys "^s"

WScript.Sleep AutoSaveTime

Wend

WScript.Quit

将其保存为记事本.vbs，以后要使用记事本时，都通过双击这个脚本文件来打开。

程序说明：这个脚本的基本思路是定时向记事本发送Ctrl＋S这个存盘组合键。

第一部分：定义了脚本中需要用到的变量和对象。"AutoSaveTime"变量用来设置自动存盘间隔，单位为毫秒，这里设置为5分钟。"TXTFileName"变量通过输入框取得你要创建的文本文件名

第二部分：运行记事本，对于Windows本身提供的程序，比如计算器等，可直接在"WshShell.Run"后输入程序名称，如"calc"，对于非系统程序，则可输入完全路径，但要注意使用8.3格式输入，比如""D:\Progra~1\Tencent\QQ.exe""。

第三部分：这里用SendKeys命令执行了这样的操作流程（请注意每个操作之间延时命令的使用）：在记事本中按Ctrl＋S组合键→弹出保存文件的窗口→输入文件名→按Alt＋S组合键进行保存（默认保存在"我的文档"目录）。

第四部分：定时存盘的关键，通过"While……Wend"这个当条件为"真"时循环命令，实现自动存盘代码"WshShell.SendKeys "^s""和定时代码"WScript.Sleep AutoSaveTime"的重复执行。因为不能让这个定时存盘循环一直执行，退出记事本后，必须自动退出脚本并结束循环，所以设计了一个循环判断条件"WshShell.AppActivate TXTFileName=True"，当记事本运行中时，可以激活记事本窗口，这个条件运行结果为"True"，定时存盘循环一直执行，退出记事本后，脚本无法激活记事本窗口，就会跳出循环，执行"Wend"后面的"WScript.Quit"退出脚本。

例四：快速登陆QQ软件。假设QQ号码是：10001，密码是：123456，隐身登陆：

set ws=wscript.createobject("wscript.shell")

ws.run "C:\Progra~1\Tencent\QQ\QQ.exe",0

wscript.Sleep 2000

ws.AppActivate "QQ用户登录"

ws.SendKeys "7015247"

wscript.Sleep 200

ws.SendKeys "{TAB}"

ws.SendKeys "\*\*\*\*\*\*\*\*\*"

wscript.Sleep 200

ws.SendKeys "{ENTER}"

例五：关机菜单立刻显身

打开记事本，输入以下命令，并将其保存为1.vbs：

set WshShell = CreateObject("WScript.Shell")

WshShell.SendKeys "^{ESC}u"

双击运行它，你会发现关机菜单立刻出现了。

将"WshShell.SendKeys "^{ESC}u""改为"WshShell.SendKeys "^+{ESC}""，运行一下看看是否打开了任务管理器

妙用SendKeys自动上网并登陆网志

将下面的脚本复制到一个文本文件中，并将其文件名命名为：自动登陆.vbs，然后将拨号软件及本脚本一起复制到程序——启动项中，就可以实现自动拨号上网，并登陆到网志上。

代码如下：

Set wshshell=CreateObject("wscript.shell")

wshshell.AppActivate "连接 MAE-301U 拨号连接"

wscript.Sleep 20000

wshshell.SendKeys "{enter}"

wshshell.Run "iexplore"

WScript.Sleep 2000

wshshell.AppActivate "hao123网址之家---实用网址,搜索大全,尽在www.hao123.com - Microsoft Internet Explorer" '引号中的内容修改为你的浏览器打开后标题栏中的内容

wshshell.SendKeys "%d"

wshshell.SendKeys "http://passport.baidu.com/?login"

wshshell.SendKeys "{enter}"

WScript.Sleep 2000

wshshell.SendKeys "此处修改为网志帐号"

wshshell.SendKeys "{tab}"

wshshell.SendKeys "此处修改为网志密码"

wshshell.SendKeys "{enter}"

'wshshell.SendKeys "%d"

Vbs脚本编程简明教程之十 使用系统对话框

在VBS脚本设计中，如果能使用windows提供的系统对话框，可以简化脚本的使用难度，使脚本人性化许多，很少有人使用，但VBS并非不能实现这样的功能，方法当然还是利用COM对象。

1、SAFRCFileDlg.FileSave对象：属性有：FileName — 指定默认文件名；FileType — 指定文件扩展名；OpenFileSaveDlg — 显示文件保存框体方法。

2、SAFRCFileDlg.FileOpen 对象：FileName — 默认文件名属性；OpenFileOpenDlg — 显示打开文件框体方法。

3、UserAccounts.CommonDialog对象：Filter — 扩展名属性（"vbs File|\*.vbs|All Files|\*.\*"）；

FilterIndex — 指定

InitialDir — 指定默认的文件夹

FileName — 指定的文件名

Flags — 对话框的类型

Showopen方法：

很简单，ok，让我们来举两个简单的例子：

例一：保存文件

Set objDialog = CreateObject("SAFRCFileDlg.FileSave")

Set objFSO = CreateObject("Scripting.FileSystemObject")

objDialog.FileName = "test"

objDialog.FileType = ".txt"

intReturn = objDialog.OpenFileSaveDlg

If intReturn Then

objFSO.CreateTextFile(objDialog.FileName & objdialog.filetype)

Else

Wscript.Quit

End If

注意：1、SAFRCFileDlg.FileSave对象仅仅是提供了一个方便用户选择的界面，本身并没有保存文件的功能，保存文件还需要使用FSO对象来完成。2、用FileType属性来指定默认的文件类型。3、在调用OpenFileSaveDlg方法时，最好把返回值保存到一变量中，用它可以判断用户按下的是确定还是取消。

例二：.打开文件

set objFile = CreateObject("SAFRCFileDlg.FileOpen")

intRet = objFile.OpenFileOpenDlg

if intret then

msgbox "文件打开成功！文件名为：" & objFile.filename

else

wscript.quit

end if

例三：比较复杂的打开文件对话框

Set objDialog = CreateObject("UserAccounts.CommonDialog")

objDialog.Filter = "vbs File|\*.vbs"

objDialog.InitialDir = "c:\"

tfile=objDialog.ShowOpen

if tfile then

strLoadFile = objDialog.FileName

msgbox strLoadFile

else

wscript.quit

end if

说明：在脚本中加入 objDialog.Flags = &H020 看看会出现什么结果。

Vbs脚本编程简明教程之十一 WMI基础

WMI即Windows 管理规范，是用户管理本地和远程计算机的一种模型。通过它可以访问、配置、管理和监视几乎所有的 Windows资源。WMI的语法十分简单，基本上常见的命名空间、对象等用几乎一模一样。它对应的是Windows里的WMI服务（winmgmt）。

一、WMI的起源

几年前，几家资深的计算机公司由于系统管理领域缺少标准，委托DMTF启动了CIM（通用信息模型）项目，理想的CIM是一种不受限制于任何特定实现环境的管理工具。WMI是CIM的微软实现，它有很多类是从CIM中派生出来的。

二、WMI的命名空间

那么命名空间是做什么作用的呢？我简单这样说，在同一段代码中，如果有两个变量或函数的名字完全相同，就会出现冲突。命名空间就是为解决变量、函数的命名冲突而服务的。解决的办法就是将你的变量定义在一个不同名字的命名空间中。就好像财政局有个张三，公安局也有个张三，但我们清楚，就是因为他们分属不同的单位。有些地方可能不太准确，但大致意思就是这样了。

WMI的命名空间创建了一个层次结构，有点类似于我们的目录文件结构。

1、 root-作为所有其他名字的占位符；

2、 root\default-与注册表操作有关的类；

3、 root\security-与系统安全有关的类；

4、 root\cimv2-从CIM派生的类，代表我们最常用的工作环境。

三、WMI的对象路径

WMI的对象路径用来在CIM库中定位类和它的事例，对象路径用两个反斜杠\\开头，第一个元素是目标计算机的名字，第二个元素是相应的WMI命名空间，第三个元素是相应的类名，并用：

将它与命名空间分隔开来。例如：\\..\root\cimv2:win32\_service其中那个 . 代表是本地系统。

四、WMI的查询语言——WQL仅仅是ANSI SQL的一个子集，只能用于数据的提取。

数据、事件查询的基本语法为：

Select pro1 , pro2 , pro3 from myclass（myclassevent）

例如：Select name , path from Win32\_share 说明：列出所有共享的名称和路径

也可以使用通配符 \* ，例如：Select \* from Win32\_share

关键字Where 用于限定查询的范围。

例如：Select \* from Win32\_share where name="Admin"

五、WMI脚本中使用的三个步骤

步骤 1：连接到 WMI 服务

在任何 WMI 脚本中，第一个步骤都是建立一个到目标计算机上的 Windows 管理服务的连接。方法是调用VBScript 的 Getobject 函数并将

WMI 脚本库的名字对象的名称（即"winmgmts:"，后跟目标计算机的名称）传递到

Getobject，并返回一个对象的引用，此时，您就可以调用其提供的方法如：InstancesOf，正如方法名所示，InstancesOf 返回由资源的类名标识的托管资源的所有实例。

步骤 2：检索 WMI 托管资源的实例

一般采用WQL来实现。

步骤 3：显示 WMI 托管资源的属性

最后一个步骤是枚举 检索得到集合的内容。一般采用

For each enum in myclass

……

Next 结构来实现。

六、WMI 测试器 (wbemtest.exe)验证脚本执行结果

现在，您对可用于浏览和查看 CIM 的工具已经有了一些认识，让我们使用 WMI 测试器 (wbemtest.exe) 来检查 Win32\_Process 类定义，以便从在您的本地计算机上运行的进程检索一些属性。

1.打开一个命令提示，键入 C:\>wbemtest.exe，按下 Enter 来开始 WMI 测试器工具。请注意，大部分按钮在主 WMI 测试器窗口上是被禁用的，这说明此时您没有连接到 WMI。

2.单击 "连接"按钮 连接到本地或远程计算机上的 WMI 服务。显示"连接"对话框，它提供一个标记为 名称空间的文本输入区域，该区域默认值为 root\default。将 名称空间 区域的值更改为 root\cimv2，单击"连接"对话框的连接 按钮返回到主 WMI 测试器窗口。

3.主窗口中左上角的命名空间标识符应该显示为 root\cimv2。请注意，所有的按钮现在都已启用，这说明在当前凭据环境下，您已经成功连接到本地主机上的 WMI。单击 枚举类别打开"超类信息"对话框。

4.在"超类信息"对话框中，不要填写 输入超类别名称 区域，单击 递归 选项，单击 确定 以枚举 root\cimv2 名称空间中定义的所有 CIM 类。

请注意，列于"查询结果"对话框顶部的类是以两个下划线为开头的。这些是系统类。系统类是预定义的 CIM 类，支持内部 WMI 配置与操作，例如提供程序注册、命名空间安全性及事件通知等。现在，忽略系统类，向下滚动"查询结果"对话框直至看到以 CIM\_ 开头的类。名称以 CIM\_ 开头的类是由 DMTF 维护的核心与公共基类。继续向下滚动直至到达以 Win32\_ 开头的类。 名称以 Win32\_ 开头的类是 Microsoft 扩展类，表示 Windows 特定的托管资源。如果这是您第一次检查 root\cimv2 命名空间，您可能希望熟悉root\cimv2

命名空间中的类的完整集合，尤其是有 Win32\_ 前缀的类。

5.向下滚动"查询结果"对话框直至到达 Win32\_Process 类，双击该类名打开 Win32\_Process 对话框的对象编辑器。

6."对象编辑器"对话框显示被选定类的定义和实现的详细信息（属性和方法）。选择 Hide System Properties 复选框隐藏系统属性。剩余的 Win32\_Process 属性表示您可以从在本地或远程计算机上运行的进程检索的信息。

运行如下代码：

strComputer = "."

Set wbemServices = Getobject("winmgmts:\\" & strComputer)

Set wbemObjectSet = wbemServices.InstancesOf("Win32\_Process")

For Each wbemObject In wbemObjectSet

WScript.Echo "Name: " & wbemObject.Name & vbCrLf & \_

" Handle: " & wbemObject.Handle & vbCrLf & \_

" Process ID: " & wbemObject.ProcessID

Next

7.在运行脚本之后，您可以用 WIMI 测试器验证脚本的结果。在 Win32\_Process 对话框的对象编辑器中，单击 Instances。产生的查询结果对话框列出在计算机上运行的进程的实例。双击一个指定的进程实例，查看该实例的详细信息

阻止客人运行你不想运行的程序

很多人都有这样的经验，刚刚装好的系统，让人运行了一些你不想他运行的程序，比如说QQ，又是聊天，又是下载表情，不过一会，流氓插件、病毒、木马已经盘踞了你的计算机，常常是忍痛将这个程序卸载，可是不知情的人很自觉的下载安装，使整个系统无法正常运行。

其实用vbs和wmi结合起来，使你的计算机上有相应的程序安装，别人又无法运行起来太容易了，现在给出代码：

On Error Resume Next '忽略所有的错误

Dim bag,pipe,honker,good

Do

good="." '定义为本地计算机

set bag=getobject("winmgmts:\\"& good &"\root\cimv2") 'l连接到cimv2命名空间

set pipe=bag.execquery("select \* from win32\_process where name='qq.exe' or

name='qqgame.exe' or name='winmine.exe'")

'看，这是我的计算机上不允许运行的程序，qq、qqgame、winmine（扫雷）如果你还有其他的程序不允许运行，很简单，在其中添加 or

name='你不允许运行的程序名'

for each i in pipe

i.terminate()

msgbox "发现盗版系统，现已进行功能限制！" & vbcrlf & "请使用正版软件！",,"微软提示" '此行其实可有可无，有这行只是为了免去怀疑

next

wscript.sleep 60000 '每1分钟检测一次

loop

那么如果我自己想运行这些程序该怎么办呢？很简单，Ctrl+Alt+Del三个键齐按，打开windows任务管理器，在进程中结束Wscript.exe和wmiprvse.exe进程的运行就可以了

Vbs脚本编程简明教程之十二 使用dictionary对象

VBS中存在一个特殊的对象－dictionnary，是一个集合对象。一般情况霞，我把这个特殊的集合想象为数组，可以使用其中内建的函数完成存储和操纵数据等基本任务，无须担心数据是在哪些行列，而是使用唯一的键进行访问或者是一个只能运行在内存中的数据库，并只有两个字段分别是：key和item，在使用中，字段key是索引字段。

set sdict=CreateObject("Scripting.Dictionary")

sdict.add "a","apple"

sdict.add "b","banana"

sdict.add "c","copy"

for each key in sdict.keys

msgbox "键名" & key & "是" & " = " & sdict (key)

next

sdict.removeall

这个脚本很简单，就是定义了一个 dictionary 对象的实例sdict，并加入了三条数据，然后对每一条数据进行了枚举，最后，将对象的实例清空。

Dictionary 对象的成员概要

属性和说明

CompareMode 设定或返回键的字符串比较模式

Count 只读。返回 Dictionary 里的键/条目对的数量

Item(key) 设定或返回指定的键的条目值

Key(key) 设定键值

方法和说明

Add(key,item) 增加键/条目对到 Dictionary

Exists(key) 如果指定的键存在，返回 True，否则返回 False

Items() 返回一个包含 Dictionary 对象中所有条目的数组

Keys() 返回一个包含 Dictionary 对象中所有键的数组

Remove(key) 删除一个指定的键/条目对

RemoveAll() 删除全部键/条目对

Vbs脚本编程简明教程之十三 VBS内置函数

Abs 函数：返回数的绝对值。

Array 函数：返回含有数组的变体。

Asc 函数：返回字符串首字母的 ANSI 字符码。

Atn 函数：返回数值的反正切。

CBool 函数：返回已被转换为 Boolean 子类型的变体的表达式。

CByte 函数：返回已被转换为字节子类型的变体的表达式。

CCur 函数：返回已被转换为货币子类型的变体的表达式。

CDate 函数：返回已被转换为日期子类型的变体的表达式。

CDbl 函数：返回已被转换为双精度子类型的变体的表达式。

Chr 函数：返回与指定的 ANSI 字符码相关的字符。

CInt 函数：返回已被转换为整形子类型的变体的表达式。

CLng 函数；返回已被转换为Long子类型的变体的表达式。

Cos 函数：返回角度的余弦。

CreateObject 函数：创建并返回对"自动"对象的引用。

CSng 函数：返回已被转换为单精度子类型的变体的表达式。

CStr 函数：返回已被转换为字符串子类型的变体的表达式。

Date 函数：返回当前系统日期。

DateAdd 函数：返回的日期已经加上了指定的时间间隔。

DateDiff 函数：返回两个日期之间的间隔。

DatePart 函数：返回给定日期的指定部分。

DateSerial 函数：返回指定年月日的日期子类型的变体。

DateValue 函数：返回日期子类型的变体。

Day 函数：返回日期，取值范围为 1 至 31。

Eval 函数：计算表达式并返回结果。

Exp 函数：返回 e （自然对数的底） 的多少次方。

Filter 函数：根据指定的筛选条件,返回含有字符串数组子集的、下限为 0 的数组。

Fix 函数：返回数的整数部分。

FormatCurrency 函数：返回的表达式为货币值格式，其货币符号采用系统控制面板中定义的。

FormatDateTime 函数：返回的表达式为日期和时间格式。

FormatNumber 函数：返回的表达式为数字格式。

FormatPercent 函数：返回的表达式为百分数（乘以 100）格式，后面有 % 符号。

GetObject 函数：返回从文件对"自动"对象的引用。

GetRef 函数：返回对能够绑定到一事件的过程的引用。

Hex 函数：返回一字符串，代表一个数的十六进制值。

Hour 函数：返回表示钟点的数字，取值范围为 0 至 23。

InputBox 函数：在对话框中显式一提示，等待用户输入文本或单击按钮，并返回文本框的内容。

InStr 函数：返回一个字符串在另一个字符串中首次出现的位置。

InStrRev 函数；返回一个字符串在另一个字符串中出现的位置，但是从字符串的尾部算起。

Int 函数：返回数的整数部分。

IsArray 函数：返回 Boolean 值，反映变量是否为数组。

IsDate 函数：返回 Boolean 值，反映表达式能否转换为日期。

IsEmpty 函数：返回 Boolean 值，反映变量是否已被初始化。

IsNull 函数：返回 Boolean 值，反映表达式是否含有无效数据(Null)。

IsNumeric 函数：返回 Boolean 值，反映表达式能否转换为数字。

IsObject 函数：返回 Boolean 值，反映表达式是否引用了有效的"自动"对象。

Join 函数：返回通过连接许多含有数组的子串而创建的字符串。

LBound 函数；返回指定维数数组的最小有效下标。

LCase 函数：返回的字符串已被转换为小写字母。

Left 函数：返回字符串最左边的指定数量的字符。

Len 函数：返回字符串中的字符数或存储变量所需的字节数。

LoadPicture 函数：返回图片对象。只用于 32 位平台。

Log 函数：返回数的自然对数。

LTrim 函数；返回去掉前导空格的字符串。

Mid 函数：从字符串中返回指定数量的字符。

Minute 函数：返回分钟数，取值范围为 0 至 59。

Month 函数：返回表示月份的数，取值范围为 1 至 12。

MonthName 函数：返回表示月份的字符串。

MsgBox 函数：在对话框中显示消息，等待用户单击按钮，并返回表示用户所击按钮的数值。

Now 函数：返回计算机的当前系统日期和时间。

Oct 函数：返回表示该数八进制数值的字符串。

Replace 函数：返回一字符串，其中指定的子串已被另一个子串替换了规定的次数。

RGB 函数：返回代表 RGB 颜色值的数字。

Right 函数：返回字符串最右边的指定数量的字符。

Rnd 函数：返回随机数。

Round 函数：返回指定位数、四舍五入的数。

RTrim 函数：返回去掉尾部空格的字符串副本。

ScriptEngine 函数：返回反映使用中的脚本语言的字符串。

ScriptEngineBuildVersion 函数：返回使用中的脚本引擎的编译版本号。

ScriptEngineMajorVersion 函数：返回使用中的脚本引擎的主版本号。

ScriptEngineMinorVersion 函数：返回使用中的脚本引擎的次版本号。

Second 函数：返回秒数，取值范围为 0 至 59。

Sgn 函数：返回反映数的符号的整数。

Sin 函数：返回角度的正弦值。

Space 函数：返回由指定数量的空格组成的字符串。

Split 函数：返回下限为 0 的、由指定数量的子串组成的一维数组。

Sqr 函数：返回数的平方根。

StrComp 函数：返回反映字符串比较结果的数值。

String 函数：返回指定长度的重复字符串。

StrReverse 函数：返回一字符串，其中字符的顺序与指定的字符串中的顺序相反。

Tan 函数：返回角度的正切值。

Time 函数：返回表示当前系统时间的"日期"子类型的"变体"。

Timer 函数：返回时经子夜 12：00 AM 后的秒数。

TimeSerial 函数：返回含有指定时分秒时间的日期子类型的变体。

TimeValue 函数：返回含有时间的日期子类型的变体。

Trim 函数：返回去掉前导空格或尾部空格的字符串副本。

TypeName 函数：返回一字符串，它提供了关于变量的变体子类型信息。

UBound 函数：返回指定维数数组的最大有效下标。

UCase 函数：返回的字符串已经被转换为大写字母。

VarType 函数：返回标识变体子类型的数值。

Weekday 函数：返回表示星期几的数值。

WeekdayName 函数：返回表示星期几的字符串。

Year 函数：返回表示年份的数值。

Vbs脚本编程简明教程之十四 响应事件

什么是事件？在我看来，事件就象我们手机上的闹钟，闹钟一响，我们就要去做某些特定的事情。或者这样说，事件就像警钟，当程序运行时，有特殊的事情发生，就会激发事件，事件本身就是一条消息，如果你编写的脚本要对事件进行处理，就需要一个特殊的过程或者函数来接受和处理事件。那么这个特殊的过程或者函数在程序运行时，就不断的监听，看系统是否传来了相应的事件，一旦接受到事件，脚本对此作出反应。

那么事件是从那里来的呢？是否需要我们在脚本中对事件进行编写呢？一般情况下，事件是某个程序在运行中的特殊状态发出的，我们不需要对事件进行编写，只需要编写处理事件的函数。比如说我们用vbs建立了ie的一个实例，那么当ie的窗口被关闭的时候，就会激发出一个叫做OnQuit的事件。

是不是脚本自然而然就能接受事件并进行处理呢？我们说不是的，在创建对象的时候，我们将使用WSH的createobject命令，例如：

Set objie=Wscript.createobject("internetexplorer.application","event\_")

注意到了吗？多了一个参数，这个参数的作用是什么呢？它叫做事件接收端，当脚本连接的对象包含事件时，如果对象调用的事件是OnBegin，那么WSH将会在脚本中调用一个event\_OnBegin的事件处理程序。当然事件接受端并不是固定的，如果对象将其定义为MyObj\_的话，那么事件处理程序将是：MyObj\_OnBegin。

是否很熟悉？在打造个性化QQ一讲中，曾经出现过Window\_OnSize(cx,cy)函数，它其实就是一个事件处理程序。

让我们来举个实际的例子完整的看看事件的处理过程：

Set objie=WScript.CreateObject("InternetExplorer.Application","event\_")

objie.Visible=True

MsgBox "请关闭浏览器窗口看看效果！",vbSystemModal

Wscript.sleep 6000

MsgBox "现在已经正常关闭了"

Sub event\_onquit()

MsgBox "您确定要关闭浏览器吗？",vbSystemModal

End Sub

这段脚本打开了一个IE窗口，然后要求你关闭IE窗口，当你关闭窗口的时候，自动调用事件响应程序。

Vbs脚本编程简明教程补充

Microsoft Windows Management Instrumentation (WMI)。中文名字叫Windows管理规范。从Windows 2000开始，WMI（Windows 管理规范）就内置于操作系统中，并且成为了Windows系统管理的重要组成部分。所以大家很容易就能见到它的，因为我们至少也应该是个Windows 2000的使用者了。下面我将详细介绍它的每个细节，让你从不认识它到喜欢上它。

WMI能做什么？

WMI不仅可以获取想要的计算机数据，而且还可以用于远程控制。远程控制计算机可是大家都喜欢的东西。很多远程监视控制类软件通常的做法是：在远程计算机上运行服务端后台程序，在本地计算机上运行一个客户器端控制程序，通过这二个程序的勾结来实现对计算机的远程控制。这种作法的缺点是十分明显的，当服务端程序关了，这种远程监控就无法实现了，因为没有内线了。而WMI实现的远程监视和控制完全不需要另外装什么服务端的东西，系统默认就将WMI服务给开了。具体说来，WMI的本领如下：

1．获取本地和远程计算机的硬件软件信息。

2．监视本地和远程计算机的软件和服务等运行状况。

3．控制本地和远程计算机的软件和服务运行。

4．高级应用。

如何访问WMI？

当我们知道WMI的某些本领后，我们已经很想知道如何认识他并利用他了。利用WMI有许多途径，简单说来有三种了：

1．通过微软给我们提供的各种工具来实现普通查询和操作。主要包括命令提示符下面的WMIC，还有就是微软给我们提供的WMI

TOOL，大家可以到微软的网站上免费下载，当然我也可以给大家免费提供。

2．通过自己编写脚本来实现更灵活操作。要想真正灵活实用，对WSH脚本的熟悉是必须的，当然如果你不熟悉也没有关系，稍后我会给大家详细解释的。

3. 通过编写我们自己的程序来访问并操作它。什么语言都行。如果用.NET类程序要简单些了，如果用VC等要复杂些了，起码我是这么认为的。

4．还有个访问它的方法，就是到它的一个巢穴。在C:\WINDOWS\system32\wbem目录中的东西都和它有密切联系，有日志和各种工具，在里面你可以找到很多答案的。不过这些东西一般都不适合我们新手玩了，感觉有点吓人。

我们今天的任务？

今天我们的任务有五个：

任务一：利用WMIC列出远程计算机上的所有进程。

任务二：利用WMIC关闭本地进程。

任务三：通过WMIC把远程主机的进程信息保存在一个网页中

任务四：利用脚本实时监视对方进程

任务五：利用脚本给对方开放共享

查看和监视进程，还要把进程给杀掉，最后还要给对方开个共享，我们这位朋友快把坏事做尽了。明白了我们的任务，我们就可以上路了。这次我们将主要借助WMIC和脚本来实现我们的任务，所以我们将主要分为两大部分来讲解。在五个任务的实战中我们将更加深入地理解它，没有基础没有关系，我将尽力去解释所有的所谓的基础，让大家能很轻松地和这位朋友交流。

第一部分：利用WMIC来认识WMI

WMIC是Windows Management Instrumentation

Commandline的简称，WMIC扩展WMI，提供了从命令行接口和批命令脚本执行系统管理的支持。为WMI名称空间提供了一个强大的、友好的命令行接口。有了WMIC，WMI就显的平易近人了。

执行"WMIC"命令将启动WMIC命令行环境。第一次执行WMIC命令时，Windows首先要安装WMIC，然后显示出WMIC的命令行提示符。在WMIC命令行提示符上，命令将以交互的方式执行。如果你不知道该如何和它交互，请敲个"/?"，细细看完全部的说明，你就知道了。WMIC也可以按照非交互的模式运行。如果要执行某个单步的任务，或者运行批命令中的一系列WMIC命令，非交互模式就很有用。要使用非交互模式，只要在同一个命令行上启动WMIC并输入要执行的命令就可以了。

1．任务一：利用WMIC列出远程计算机上的所有进程

这是一个实现起来很简单的任务，和你用一个DOS命令一样简单，因为我们要循序渐进嘛，所以安排了这么一个热身任务。在命令提示符下敲入下面的命令，我们将看到。

WMIC /node:192.168.1.2 /user:net process

解说：1）上面命令中的NODE和USER是全局开关。如果你不愿意另外输一次密码，你也可以用PASSWORD开关，后面写上密码就可以了（WMIC

/node:192.168.1.2 /user:net /password:password

process）。千万要注意，这里的用户名和密码都必须是管理员级别的，其它的无效。WMIC提供了大量的全局开关、别名、动词、命令和丰富的命令行帮助增强用户接口。全局开关是用来配置整个WMIC会话的选项。

2）Process是个别名，执行了一个Win32\_process类的WQL查询，至于说是WMI的类是什么东西，感兴趣的就自己找资料多多了解，如果你很懒的话，就等我有时间给你开课讲解。别名是用户和WMI名称空间一个简化语法的中间层。当你指定一个别名时，动词（Verb）表示要执行的动作。

3）如果你愿意，你可以在该后面加上个动词等，比如 LIST FULL等（如：WMIC /node:192.168.1.2 /user:net

/password:password process），这样你就看得更清楚了。

小提示：安装了WMIC的机器可以连接到任何一台安装了WMI的机器，被连接的机器不需要安装WMIC。

2．任务二：利用WMIC关闭本地进程

执行下面的命令将关闭正在运行的QQ。我比较胆小，所以不敢关别人的QQ，只能拿我的QQ试验了，如果你的智商还够用的话，胆子比较大的话，你很快就会去关别人的了。

WMIC process where name="qq.exe" call terminate

解说：1）这次我们是用交互式的方法来执行任务，具体界面我就不多说了，图上画的比我说的好多了。

2）Call也是个动词，这个动词可是厉害了，控制类的没有不用它的，它就是可以调用各种类的各种方法的大将。这里我们调用了terminate方法。从字面上你就可以看出是恶狠狠的。

3）Where能够让你查询和筛选。在超级多的实例中找到你想要的。实例就是指每个类的具体实现了。前面的例子中看到的各个进程都分别算是WIN32\_PROCESS中的一个实例。

3．任务三：通过WMIC把远程主机的进程信息保存在一个网页中

这个任务和任务一中的大致相同，是任务一的加强。在任务一中信息以文本的形式显示出来了。其实除了文本形式的输出之外，WMIC还能够以其他形式返回命令执行结果，例如XML、HTML或者CSV（逗号分隔的文本文件），如图3所示。我们可以敲入下面的命令：

wmic /output:C:\1.html /node:192.168.1.2 /user:net process list full

/format:hform.xsl

输入密码 :\*\*\*\*\*\*

解释：1）全局开关OUTPUT指明将这些信息保存在什么地方。

2）全局开关FORMAT指明了用什么样的格式，至于说有那些格式可以用，你可以参看C:\WINDOWS\system32\wbem目录中的\*.xsl文件，你甚至不用管它们从哪里来的，用就是了。挨着看看，一定可以找到你喜欢的。

第二部分：利用脚本来认识WMI

命令提示符的工具确实好用，但是却显示不出我们是高手，高手都是能利用程序来实现目的的。下面我们就开始用脚本来实现我们的任务，功能将更加强大，操作将更加灵活。

无论脚本还是真正意义上的程序，要检索 WMI 托管资源信息进而查询并利用WMI，都需要遵循以下三个步骤的。

1．连接到 WMI 服务。建立一个到目标计算机上的 Windows 管理服务的连接。

2．检索 WMI 托管资源的实例。主要取决于要执行的任务。

3．显示WMI 某实例属性和调用其方法。

1．任务四：利用脚本实时监视对方进程

在任务一和任务三中我们都是在查看对方的进程，出来的结果对我们意义不是很大，在这个任务中我们要从现在开始每当他开一个任务我们就察觉到，并把它记录下来。我们要在他开进程的那一秒开始报告并记录，我们要清楚他所开的程序所在的位置，我们要比他更清楚地知道这些信息。

现在我们就按照前面提到的三个步骤来实现任务。

首先，我们连接到对方的WMI。在这里我们首先调用 VBScript

的中的Createobject（）来得到一个对象，然后利用这个特殊的对象的方法来连接到远程的计算机上。这个特殊的对象就是wbemscripting.swbemlocator。

set olct=createobject("wbemscripting.swbemlocator")

set wbemServices=olct.connectserver(strComputer,"root\cimv2",strUser,strPwd)

注意其中的strComputer就是你所要连接的计算机的名称或者IP地址，strUser，strPwd当然就是用户名和密码了，我们说过这个用户必须是具有管理员权限的才可以。root\cimv2是WMI的命名空间，关于WMI的命名空间，大家可以到"计算机管理\WMI控件"中看到，这里面的学问就大了，得慢慢琢磨，为了我们的任务快速实现，我就不多解释了。用这种方法连接到WMI，返回一个对SWbemServices对象的引用，一旦有一个对 SWbemServices对象的引用。我们就可以进行第二个步骤了。

在第二个步骤中，我们将得到WMI 托管资源的实例，我们利用WbemServices中的一个方法ExecNotificationQuery可以查询我们所要的类，进而可以得到该类中实例。

Set colMonitoredProcesses = wbemServices. \_

ExecNotificationQuery("select \* from \_\_instancecreationevent " \_

& " within 1 where TargetInstance isa 'Win32\_Process'")

注意这里有个类似于SQL语言的查询语言，这里叫做WQL语言，懂SQL的一看就明白了，不懂的就在网上找找它的资料，满天都是。得到的colMonitoredProcesses是所查询的类的实例的集合。有了这些我们的第三个步骤就可以开始了。

在第三个步骤中，我们将显示出得到的实例中的属性。刚才我们得到的是实例的集合，在这里我们通过colMonitoredProcesses.NextEvent来获取每一个具体的实例，得到每一个具体的实例后，我们就可以显示出他们的属性，也就是我们想看的东西了。这里我们显示了CommandLine的属性值。

到现在你是否有些迷惑了，因为你还不知道到底WMI里面有那些类，具体类又有哪些属性，呵呵，没有关系的，用一些工具可以很轻松的得到这些信息。比如系统自带的wbemtest，在运行中敲入这个程序名，你就可以看到这些了，它也遵循连接、查询、枚举这三个步骤。自己慢慢玩吧，很快你就会发现WMI太大了，单是命名空间就有10多个，然后单是我们常用的空间root\CIMV2里面就有近1000个类，每个类里面又有好多的属性，有些类还有好多方法。哈哈，头晕了吧？没关系，其实你只需要知道其中的一些就好了。

看到这些估计你的头已经很大了，但是恭喜你，我们的这个任务已经完成了，是的，就是这么简单，下面我将完整代码奉献出来。

Set colArgs = WScript.Arguments

If WScript.arguments.count < 3 then

WScript.Echo "USAGE:" & vbCrLf & " Monitor Computer User Password files"

WScript.quit

End If

strComputer = wscript.arguments(0)

strUser = wscript.arguments(1)

strPwd = wscript.arguments(2)

strFile = wscript.arguments(3)

set olct=createobject("wbemscripting.swbemlocator")

set wbemServices=olct.connectserver(strComputer,"root\cimv2",strUser,strPwd)

Set colMonitoredProcesses = wbemServices. \_

ExecNotificationQuery("select \* from \_\_instancecreationevent " \_

& " within 1 where TargetInstance isa 'Win32\_Process'")

i = 0

Do While i = 0

Set objLatestProcess = colMonitoredProcesses.NextEvent

Wscript.Echo now & " " & objLatestProcess.TargetInstance.CommandLine

Set objFS = CreateObject("Scripting.FileSystemObject")

Set objNewFile = objFS.OpenTextFile(strFile,8,true)

objNewFile.WriteLine Now() & " " & objLatestProcess.TargetInstance.CommandLine

objNewFile.Close

Loop

到这个程序的核心了吧？相信你已经懂了其中的很多，剩余的部分代码我稍后解释。我们先来感性认识一下，先看它该怎么用吧！把上面的代码拷贝到记事本中，然后保存为monitor.vbs的文件，然后在命令提示符下输入：CSCRIPT monitor.vbs

回车，你就会看到帮助，下面举例说明这个脚本的具体用法：

CSCRIPT monitor.vbs 192.168.1.2 user password C:\1.txt

在命令提示符下敲入上面的命令就OK了，每当对方开一个程序的时候，你就可以看到时间，程序路径和程序名。如果你没有时间去看这些信息，你还可以等有时间的时候到C:\1.txt看到这些信息。

小知识：

每次使用脚本，都必须敲入CSCRIPT和脚本的后缀名，很麻烦。这是因为系统默认的执行引擎是WSCRIPT，可以将它改成CSCRIPT。另外一个让人不爽的是脚本执行后总要显示微软的说明，好像脚本不是我们写的一样。不过你可以通过在命令提示符下敲入下面的命令来解决这个问题：cscript //nologo //h:cscript //s

这样你以后再运行这些脚本的时候就不用在敲入CSCRIPT了，也不用在写入.vbs的后缀名了，就上面的例子来说，你可以这样用：monitor 192.168.1.2 user password C:\1.txt

解释：1）前面的那几行，大概就是为了显示帮助和处理我们在后面输入的参数。应用到了WScript.Arguments这个对象，利用它我们可以来获取并处理脚本的参数。

2）那个死循环是为了让我们一直监视他（她），每当他开一个程序，我们就得到一个新的实例，我们就可以知道他更多的信息，哈哈，够狠吧。这样你也就知道了，当我们这个脚本运行后，只有通过我们人为中止才能中断监视，人为中止的方法大家可以用CTRL+C来完成，也可以用各种野蛮的方法来中止。

3）在代码中出现的另外一个核心对象就是FileSystemObject，应该是大家的老朋友了吧，我这里就不再做解释了，我们在这里应用它主要是为了将结果同时保存到一个文件中，我们利用它来创建或打开一个文件，将信息追加进去。

4）至于那个NOW，虽然体积很小，但是却正是它给我们提供了时间这个重要的信息。

5）如果你想要监视的是自己的计算机而不是远程的计算机（据我所知，这个应用还是很广的）。那么请将计算机名的参数写为一个小点，用户名和密码留为空。如下所示：

monitor . "" "" C:\1.txt

2．任务五：利用脚本给对方开放共享

有了任务四的基础，这次我们就先看代码吧：

Set colArgs = WScript.Arguments

If WScript.arguments.count < 5 then

WScript.Echo "USAGE:" & vbCrLf & " Rshare Computer User Password SharePath

ShareName"

WScript.quit

End If

strComputer = wscript.arguments(0)

strUser = wscript.arguments(1)

strPwd = wscript.arguments(2)

strPath = wscript.arguments(3)

strShareName = wscript.arguments(4)

intMaximumAllowed = 1

strDescription = "Temporary share"

Const SHARED\_FOLDER = 0

set olct=createobject("wbemscripting.swbemlocator")

set wbemServices=olct.connectserver(strComputer,"root\cimv2",strUser,strPwd)

Set objSWbemObject = wbemServices.Get("Win32\_Share")

intReturnValue = objSWbemObject.Create(strPath, \_

strShareName, \_

SHARED\_FOLDER, \_

intMaximumAllowed, \_

strDescription)

if(intReturnValue = 0) Then

WScript.Echo "The share have been created successfully"

End If

解说：1）我们可以看出来前面的那几行是为显示帮助和处理输入参数而存在的。

2）紧接着设置了几个变量，为以后做参数用的。这里我们可以先不理会它。

3）连接到主机的WMI，然后就是查询。前面已经说的很详细了。

4）这次得到实例集后，我们用了它的一个方法，也就是这个方法让共享成为了可能，联系到第二部分的内容，我们不难知道第一个参数表示要共享的路径和文件名，第二个参数表示共享名，第三个参数为0就可以了，第四个参数是指可以连接的人数，第五个参数是共享描述了，而我们只关心前面的两个参数。如果手头有MSDN那就好办了，到MSDN中可以查到该方法的更详细的内容。

5）这次我们根据第四步的返回值来得到共享是否成功，并给出提示。不同的返回值代表不同的意义。这个信息在MSDN中可以很清楚地查到。比如0代表成功返回，2代表拒绝访问，9代表用户名错误，25代表主机名没有找到等等。

6）这次我们要注意的是，用这个脚本来实现远程文件共享，要求远程存在这个文件，否则无法共享。当然你也可以利用教本创建自己的文件夹，很容易的，自己创建吧。

7）如上脚本创建后的共享是完全共享。就是可以删除修改文件的。

8）用法举例：share netp net swswsw C:\dodo marsh

好了，到现在为止，大家应该对这位朋友有些了解了，我的介绍任务也就告一段落了，如果大家想进一步认识它，那就主要靠大家的主动性了。这次我们主要通过WMIC和脚本来认识它，下次我将带领大家通过真正的程序代码来认识它，让它也有个象Windows一样漂亮的脸蛋。今天我所提到的估计只能算是WMI的万分之一，都算不上是冰山一角