

18. 脚本集锦

序号	内容	页码
18.1	脚本使用数组	18-2
18.2	变量数值写入文本文件	18-2
18.3	查询或设置注册表	18-3
18.4	脚本错误继续运行	18-3
18.5	检测并处理脚本异常错误	18-3
18.6	变量数值写入 Excel 并显示	18-4
18.7	变量值追加到数据表	18-4
18.8	文本内容滚动显示	18-5
18.9	发送电子邮件	18-5
18.10	访问 XML 文件	18-6
18.11	Excel 实时读取变量值	18-7
18.12	打开 PDF 文档	18-8
18.13	判断 Excel 是否被启动	18-8
18.14	得到某天距年初的周数和天数	18-9
18.15	通过资源管理器打开某文件夹	18-9
18.16	防火墙开放端口	18-9
18.17	使用消息队列 (MSMQ)	18-9
18.18	脚本内容换行输入	18-12
18.19	访问 Webservice	18-13
18.20	无符号整数问题	18-14
18.21	向 HTTP 服务器发送 POST 请求	18-14
18.22	打开 IE 浏览器	18-15
18.23	判断当前时间是否在 5 月 1 日与 10 月 1 日之间	18-15
18.24	模拟变量值格式化为科学计数法表示的文本	18-16
18.25	从某文本文件读取内容并赋值给变量	18-16
18.26	读写 INI 文本文件	18-18
18.27	获取某文件目录下最新建立的文件名称	18-18
18.28	获取某文件目录下最新修改的文件名称	18-19
18.29	退出 (Exit) 脚本执行	18-20
18.30	访问注册表	18-21
18.31	某批变量存储成 CSV 格式文件并 FTP 同步到其他计算机	18-22
18.32	使用 Split 函数转换字符串到数组	18-25
18.33	记录操作日志到文本文件	18-25

18.1 脚本使用数组

- VBScript 脚本, 数组不允许被多次重复定义;
- 建议数组的定义放在画面脚本或全局脚本的初始脚本中, 在其他脚本中使用数组;
- 举例:

在初始脚本中定义和初始化数组:

```
Dim myar(10)
For i=0 To 10
    myar(i)=0
Next
```

在其他脚本中使用数组:

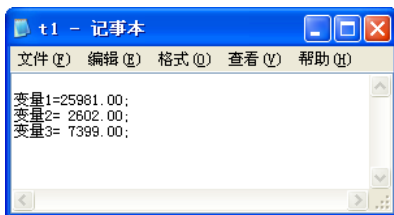
```
For i=0 To 10
    myar(i)=myar(i)+1
Next
```

18.2 变量数值写入文本文件

- 通过脚本, 把变量数值格式化后成字符串, 文本内容写入文本文件;

```
x1=RunSys.GetVarValue(VA,"%RAND",-1)
x2=RunSys.GetVarValue(VA,"%X=X+1",-1)
x3=RunSys.GetVarValue(VA,"%X=X-1",-1)
bValue=TxtFileObj.OpenFile("c:\fameview\t1.txt")
If bValue=1 Then
    StringObj.Format "变量1=%8.2f;", x1
    s1=StringObj.GetText()
    StringObj.Format "变量2=%8.2f;", x2
    s2=StringObj.GetText()
    StringObj.Format "变量3=%8.2f;", x3
    s3=StringObj.GetText()
    s=s1&vbCrLf&s2&vbCrLf&s3
    TxtFileObj.WriteFile s,0
    TxtFileObj.CloseFile
End If
```

- 写入完成后, 文件内容:



18.3 查询或设置注册表

- 访问注册表得到计算机名称:

```
Set myObj=CreateObject("WScript.Shell")
path="HKLM\System\CurrentControlSet\Control\ComputerName\ComputerName\ComputerName"
PCName=myObj.RegRead(path)
MsgBox("计算机名为"&PCName)
Set myObj=Nothing
```

- 访问注册表自动启动程序:

```
Set myObj=WScript.CreateObject("WScript.Shell")
RegPath="HKLM\Software\Microsoft\Windows\CurrentVersion\Run\"
Type_Name="REG_SZ"
Key_Name="AutoRun"
Key_Data="C:\Myfile\autorun.exe"
myObj.Write RegPath&Key_Name,Key_Data,Type_Name
Set myObj=Nothing
```

18.4 脚本错误继续运行

- 脚本开始的地方加入语句"On Error Resume Next", 脚本发生错误继续运行:

```
On Error Resume Next
n=100/0
RunSys.VarOnOff VA,"%VA1",-1
```

18.5 检测并处理脚本异常错误

- 通过 Err 对象检测并处理脚本产生的异常错误:

例 1:

```
On Error Resume Next
n=100/0
If Err.Number<>0 Then
    MsgBox Err.Description
    Err.Clear
End If
```

例 2:

```
On Error Resume Next
Set conn = CreateObject("ADODB.Connection")
strConn="Driver={SQL Server};Server=(local);Database=UserDatabase;Uid=;Pwd=;"
conn.Open strConn
If Err.Number<>0 Then
    MsgBox Err.Description
    Err.Clear
End If
```

18.6 变量值写入Excel并显示

```
x1=RunSys.GetVarValue(VA,"%X=X+1",-1)
x2=RunSys.GetVarValue(VA,"%X=X-1",-1)
Set ExcelObj = CreateObject("Excel.Application")
ExcelObj.Visible = False
ExcelObj.Application.DisplayAlerts= False
ExcelFile = "C:\FameView\ExcelFile\test.xls"
ExcelObj.Workbooks.Open ExcelFile
ExcelObj.Sheets("Sheet1").Select
ExcelObj.Cells(1,3) = x1
ExcelObj.Cells(2,3) = x2
ExcelObj.Visible = True
Set ExcelObj=Nothing
```

18.7 变量值追加到数据表

```
f1=RunSys.GetVarValue(VA,"%VA1",-1)
f2=RunSys.GetVarValue(VA,"%VA2",-1)
f3=RunSys.GetVarValue(VA,"%VA3",-1)
' 连接数据库
Set conn = CreateObject("ADODB.Connection")
strConn="Driver={SQL Server};Server=(local);Database=UserDatabase;Uid=;Pwd=;"
conn.Open strConn
' 打开数据表
Set rs= CreateObject("ADODB.Recordset")
strSQL="SELECT * FROM demo"
rs.Open strSQL, conn, 2, 2
' 追加记录
rs.AddNew
rs("dt")=Now()
rs("f1")=f1
rs("f2")=f2
rs("f3")=f3
rs.Update
' 关闭数据库
rs.Close
conn.Close
Set rs=Nothing
Set conn=Nothing
```

18.8 文本内容滚动显示

- 假设文本变量 VT.VT1, 脚本改编文本内容实现滚动;
- 画面中通过定时器执行脚本:

```

bValue=RunSys.OpenDatabase(3000)
If bValue=1 Then
    s=RunSys.GetVTtext("VT1",-1)
    s1=Left(s,1) : s2=Right(s,Len(s)-1) : s=s2+s1
    RunSys.SetVTtext "VT1",-1,s
    RunSys.CloseDatabase
End If

```

18.9 发送电子邮件

On Error Resume Next

Set cdoMail = CreateObject("CD0.Message")

Set cdoConf = CreateObject("CD0.Configuration")

```

cdoMail.Subject = "Test Mial"           ' 邮件主题
cdoMail.From     = "jkgong@fameview.com" ' 发件地址
cdoMail.To       = "jktong@fameview.com" ' 收件地址
cdoMail.TextBody = "Test Text"          ' 邮件正文
cdoMail.AddAttachment "c:\temp\test.rar" ' 附件
cdoConfAddr="http://schemas.microsoft.com/cdo/configuration/"
cdoConf.Fields.Item(cdoConfAddr & "sendusing")           = 2
cdoConf.Fields.Item(cdoConfAddr & "smtpserver")          = "smtp.fameview.com" ' 发送服务器
cdoConf.Fields.Item(cdoConfAddr & "smtpserverport")       = 25 ' 服务器端口
cdoConf.Fields.Item(cdoConfAddr & "smtpconnectiontimeout") = 10 ' 连接超时
cdoConf.Fields.Item(cdoConfAddr & "smtpauthenticate")     = 1 ' 需要身份验证
cdoConf.Fields.Item(cdoConfAddr & "sendusername")         = "帐户名称"
cdoConf.Fields.Item(cdoConfAddr & "sendpassword")         = "帐户口令"
cdoConf.Fields.Item(cdoConfAddr & "smtpusessl")           = False
cdoConf.Fields.Update
Set cdoMail.Configuration = cdoConf
cdoMail.Fields.Item("urn:schemas:mailheader:X-MSMail-Priority") = "High"
cdoMail.Fields.Item("urn:schemas:mailheader:X-Priority")        = 2
cdoMail.Fields.Item("urn:schemas:httpmail:importance")          = 2
cdoMail.Fields.Update
n=1 : n=cdoMail.Send() ' 发送邮件成功返回0
RunSys.SetVarValue VA,"%VA1",-1,n
Set cdoMail = Nothing
Set cdoConf = Nothing

```

18.10 访问 XML 文件

□ 例子 1(建立 XML 文件):

```

' 建立XML对象
Set doc = CreateObject("Microsoft.XMLDOM")
' 同步打开XML文件
doc.async = False
doc.Load "c:\temp\myfile.xml"
' 建立根节点
Set root = doc.SelectSingleNode("Root")
If root Is Nothing Then
    Set root=doc.CreateElement("Root")
    doc.AppendChild root
End If
' 在根节点下建立某子节点
Set node = root.SelectSingleNode("node")
If node Is Nothing Then
    Set node=doc.CreateElement("node")
    root.AppendChild node
End If
' 设置子节点的值
node.Text="Node1"
' 设置子节点下的某参数
node.setAttribute "Param1", "p1"
doc.Save "c:\temp\myfile.xml"
Set doc=Nothing

```

□ 例子 2(读取 XML 文件参数):

```

' 建立XML对象
Set doc = CreateObject("Microsoft.XMLDOM")
' 同步打开XML文件
doc.async = False
doc.Load "c:\temp\myfile.xml"
' 得到根节点
Set root = doc.SelectSingleNode("root")
If Not (root Is Nothing) Then
    ' 得到根节点下的某子节点
    Set node = root.SelectSingleNode("subnode")
    If Not (node Is Nothing) Then
        ' 读取节点下的某参数
        s=node.getAttribute("param1")
    End If
End If

```

```

        RunSys.SetVTtext "%VT1",-1,s
    End If
End If
Set doc=Nothing

```

□ 例子 3(修改 XML 文件参数):

```

' 建立XML对象
Set doc = CreateObject("Microsoft.XMLDOM")
' 同步打开XML文件
doc.async = False
doc.Load "c:\temp\myfile.xml"
' 得到根节点
Set root = doc.SelectSingleNode("root")
If Not (root Is Nothing) Then
    ' 得到根节点下的某子节点
    Set node = root.SelectSingleNode("subnode")
    If Not (node Is Nothing) Then
        ' 修改节点下的某参数
        node.setAttribute "Param1","test"
    End If
End If
' 保存文件
doc.Save "c:\temp\myfile.xml"
Set doc=Nothing

```

18.11 Excel 实时读取系统变量值

□ 在某 Excel 文件中添加以下 3 个宏, 即可实现:

' 宏 1:

' 宏名称为[auto_open]时, Excel 打开时会自动调用, 并执行另一个宏[runtimer]

```

Sub auto_open()
    Call runtimer
End Sub

```

' 宏 2:

' 宏[runtimer]去启动Excel定时器, 从现在时刻开始, 等待5秒后, 执行另一个宏[ReadVar]

```

Sub runtimer()
    Application.OnTime Now() + TimeValue("00:00:05"), "ReadVar"
End Sub

```

’ 宏 3:

’ 宏[ReadVar]从系统的运行数据库中读取变量，并把变量值更新到Excel单元格中;

```
Sub ReadVar()
    Dim obj As Object
    Set obj = CreateObject("fmDM0.Rundb")
    If obj.Open() = 1 Then
        Dim n As Long
        Dim x As Single
        n = obj.GetVarIndex(7, "%X=X+1")
        x = obj.GetVarValue(7, n)
        Sheets("Sheet1").Cells(1, 1) = x
        n = obj.GetVarIndex(7, "%X=X-1")
        x = obj.GetVarValue(7, n)
        Sheets("Sheet1").Cells(2, 2) = x
        n = obj.GetVarIndex(7, "%RAND")
        x = obj.GetVarValue(7, n)
        Sheets("Sheet1").Cells(3, 3) = x
        obj.Close
    End If
    Set obj = Nothing
    ’ 继续启动定时器
    Call runtime
End Sub
```

18.12 打开 PDF 文档

- 一般使用脚本即可(直接选择输入pdf文件):

```
RunSys.RunProcess "c:\temp\test.pdf"
```

- 如果不能打开,可尝试脚本(找到Adobe Reader执行程序):

```
RunSys.RunProcess "C:\Program Files\Adobe\Reader 9.0\Reader\AcroRd32.exe  
c:\temp\test.pdf"
```

18.13 判断 Excel 是否被启动

```
On Error Resume Next
Dim ExcelApp
Set ExcelApp=GetObject(,"Excel.Application")
If TypeName(ExcelApp)="Application" Then
    ExcelApp.Quit
    Set ExcelApp=Nothing
End If
```


18.14 得到某天距年初的周数和天数

VBScript 提供的 DatePart 函数可直接得到某日期距年初的星期数和天数：

' 得到周数

```
n=DatePart("ww",Now())
```

' 得到天数

```
n=DatePart("y",Now())
```

18.15 通过资源管理器打开某文件夹

□ 使用脚本打开<c:\temp>目录, 查看其中的内容:

```
RunSys.RunProcess "C:\WINDOWS\explorer.exe c:\temp"
```

18.16 防火墙开放端口

```
Set objFwMgr = CreateObject("HNetCfg.FwMgr")
Set objProfile = objFwMgr.LocalPolicy.CurrentProfile
Set colOpenPorts = objProfile.GloballyOpenPorts
Set objOpenPort = CreateObject("HNetCfg.FWOpenPort")
objOpenPort.Name = "TCP10000" ' 端口名称
objOpenPort.Port = 10000 ' 端口号
objOpenPort.Protocol = 6 ' 协议类型, 6=TCP, 17=UDP
objOpenPort.Scope = 0
objOpenPort.Enabled = True
colOpenPorts.Add objOpenPort
Set objFwMgr=Nothing
```

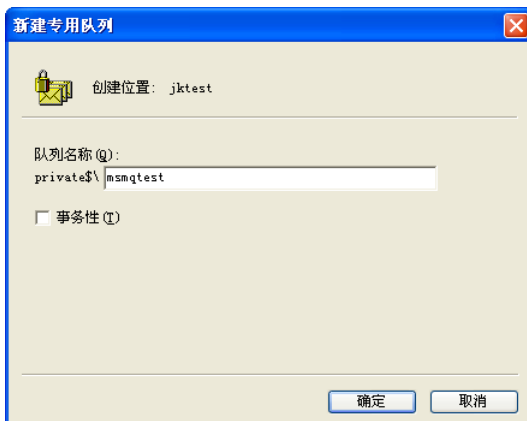
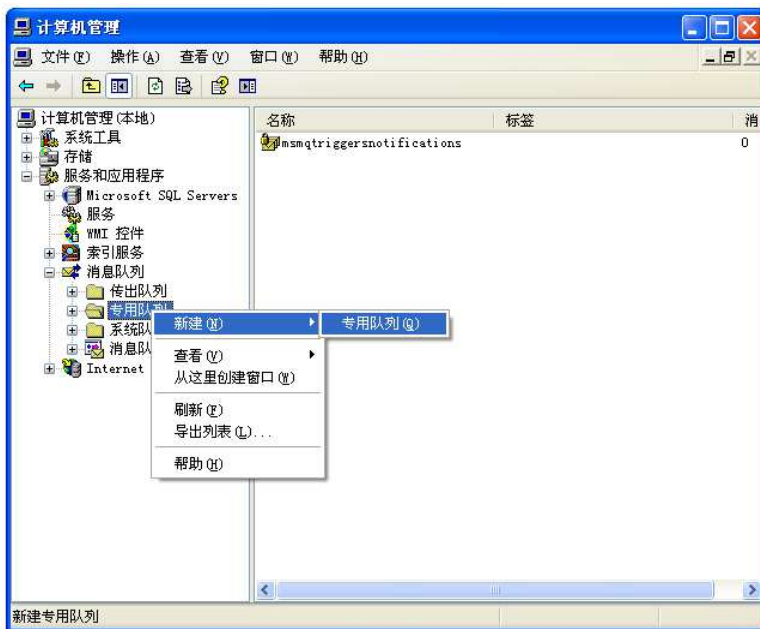
18.17 使用消息队列(MSMQ)

□ 安装MSMQ, 控制面板->安装Windows程序:





- 配置MSMQ, 控制面板->管理工具->计算机管理, 建立专有队列msmqtest:



□ 编写脚本发送消息到队列:

```
' 建立消息队列对象
Set MSMQQueueInfo = CreateObject("MSMQ.MSMQQueueInfo")
Set MSMQQueue      = CreateObject("MSMQ.MSMQQueue")
Set MSMQMessage     = CreateObject("MSMQ.MSMQMessage")
' 定义消息队列
' 队列路径格式1:"DIRECT=OS:计算机名\private$\队列名"
' 队列路径格式2:"DIRECT=TCP:IP\private$\队列名"
MSMQQueueInfo.FormatName = "DIRECT=OS:jkgong_fameview\private$\msmqtest"
' 传送方式
Transaction=0 ' MQ_NO_TRANSACTION=0
If MSMQQueueInfo.IsTransactional2 Then Transaction=3 ' MQ_SINGLE_MESSAGE=3
' 开始发送
Set MSMQQueue = MSMQQueueInfo.Open(2,0) ' MQ_SEND_ACCESS=2, MQ_DENY_NONE=0
MSMQMessage.Body = "Test Message Body" ' 发送内容
MSMQMessage.Label = "Test Message Label" ' 发送标签
MSMQMessage.Send MSMQQueue, Transaction
Set MSMQMessage = Nothing
Set MSMQQueue = Nothing
Set MSMQQueueInfo = Nothing
```

□ 编写脚本从队列接收消息, 如果队列中没消息一直等待:

```
' 建立消息队列对象
Set MSMQQueueInfo = CreateObject("MSMQ.MSMQQueueInfo")
Set MSMQQueue      = CreateObject("MSMQ.MSMQQueue")
Set MSMQMessage     = CreateObject("MSMQ.MSMQMessage")
' 定义消息队列
' 队列路径格式1:"DIRECT=OS:计算机名\private$\队列名"
' 队列路径格式2:"DIRECT=TCP:IP\private$\队列名"
MSMQQueueInfo.FormatName = "DIRECT=OS:jkgong_fameview\private$\msmqtest"
' 开始同步接收
Set MSMQQueue= MSMQQueueInfo.Open(1,0) ' MQ_RECEIVE_ACCESS=1, MQ_DENY_NONE=0
Set MSMQMessage = MSMQQueue.Receive
' 接收到的内容
MsgBox MSMQMessage.Body
MsgBox MSMQMessage.Label
Set MSMQMessage = Nothing
Set MSMQQueue = Nothing
Set MSMQQueueInfo = Nothing
```

18.18 脚本内容换行输入

□ 例1:

```
MsgBox "123" _  
    &"456" _  
    &"789" _  
    &"ABCDEFG" _  
    &"abcdefg"
```



□ 例2:

```
MsgBox "123" _  
    &"456" _  
    &"789" _  
    &"&vbCrLf&" _  
    &"ABCDEFG" _  
    &"&vbCrLf&" _  
    &"abcdefg"
```



□ 例3:

```
ret=MsgBox("描述:1234567890", _  
    vbYesNo+vbInformation+vbSystemModal, _  
    "标题:abcdefg")
```



18.19 访问 Webservice

- 网上搜索、下载并安装soapsdk3.0

(<http://download.microsoft.com/download/2/e/0/2e068a11-9ef7-45f5-820f-89573d7c4939/soapsdk.exe>)

- 例1(查询手机号码信息):

```
On Error Resume Next
Set soap = CreateObject("MSSOAP.SoapClient30")
If Err.Number = 0 Then
    soap.ClientProperty("ServerHTTPRequest") = True
    soap.mssoapinit "http://www.webxml.com.cn/WebServices/MobileCodeWS.asmx?wsdl", _
        "", "", ""
    s=soap.getMobileCodeInfo("13501234567", "")
    MsgBox s
End If
Set soap = Nothing
```

- 例2(上传数据):

```
On Error Resume Next
strTime=CurrentTime.GetFormatTime("%Y-%m-%d %H:%M:%S")
strData="100,200,300,400,500,600"

Set soap = CreateObject("MSSOAP.SoapClient30")
If Err.Number=0 Then
    Soap.ClientProperty("ServerHTTPRequest")=True
    Soap.mssoapinit "http://spp.foxconn.com/eqcs/EQCSservice.asmx?wsdl", "", "", ""
    ResultFlag=""
    ResultDesc=""
    s=soap.UploadEQValue("SZ1", "E05", "E05-1-001", "1", strTime, strData, _
        ResultFlag, ResultDesc)

    If s=True Then
        ' 上传成功 MsgBox 1
    Else
        ' 上传失败 MsgBox 0
    End If
End If
Set soap=Nothing
```

- MSSOAP全称是Microsoft Soap Toolkit

Webservice 其实就是一个基于网络的应用程序,它向外部程序提供一定的调用接口。外部(远程)的程序,可以经由Web对其进行调用;由于Webservice 经由web服务器发布,所以比DCOM而言,可以顺利穿过防火墙,对Webservice 的调用,是通过SOAP协议进行的,简单对象访问协议(SOAP)提供了标准的远程过程调用(RPC)方法来调用Web service,而SOAP也是基于XML和XSD的,XML是SOAP的数据编码方式,对Webservice 的调用也可以是跨平台的;

18.20 无符号整数问题

- 下面脚本：

```
a=&H8000
```

```
b=1
```

```
c=a+b
```

希望执行结果是32769, 但实际结果为-32767, 为什么呢?

- 原来VBScript脚本没有无符号整数类型, 16进制&H8000应该为32768;
但因为占1个字的数值最高位为1, 被作为符号位使用, 表示成了-32767;

- 以下脚本可正确执行：

```
x=&H8000
```

```
If x>=0 Then
```

```
    a=x
```

```
Else
```

```
    a=(x And &H0FFF)+32768
```

```
End If
```

```
b=1
```

```
c=a+b
```

18.21 向 HTTP 服务器发送 POST 请求

' 建立Http对象

```
Set HttpObj = CreateObject("FameView.Http")
```

' 连接到Http服务器, 参数分别为:服务器名称, 端口号, 页面

```
retCode=HttpObj.ConnectToHttpServer("218.26.184.148",18445,"/janus-ws/devices")
```

' 得到P12证书, 支持HTTPS协议, HTTP协议不需要证书

```
' If retCode=0 Then
```

```
    retCode=HttpObj.GetCertContext("c:\temp\certfile.p12","password")
```

```
' End If
```

' 发送内容

```
If retCode=0 Then
```

```
    ' 发送Header
```

```
    HttpObj.SendHttpRequestHeader "Content-Type:application/json"
```

```
    HttpObj.SendHttpRequestHeader "Version:1"
```

```
    ' 发送Body
```

```
    strBody="{\"authId\":\"1180998613232575329\",\"type\":1,\"deviceId\":20}"
```

```
    retCode=HttpObj.SendHttpRequestBody(strBody)
```

```
End If
```

```

'接收内容
If retCode=0 Then
    '接收状态代码(200, 201, 400, 405...)
    statusCode=HttpObj.GetHttpStatusCode()
    If statusCode=201 Then
        '接收Header
        strHeader=HttpObj.ReceiveHttpHeader()
        '接收应答Response
        strResponse=HttpObj.ReceiveHttpResponse()
    End If
End If

'关闭Http连接
HttpObj.CloseHttpConnect
'销毁Http对象
Set HttpObj=Nothing

```

18.22 打开 IE 浏览器

```

Set objIE=CreateObject("InternetExplorer.Application")
objIE.Navigate "http://www.fameview.com"
Do
Loop While objIE.Busy
objIE.Resizable=True
objIE.Width=800
objIE.Height=800
objIE.Left=10
objIE.Top=10
objIE.Visible=True

```

18.23 判断当前时间是否在 5 月 1 日与 10 月 1 日之间

```

imeSpanObj.SetStartTime -1, 5, 1, 0, 0, 0
TimeSpanObj.SetEndTime -1, -1, -1, -1, -1, -1
n1=TimeSpanObj.GetTotalMinutes()
TimeSpanObj.SetStartTime -1, 10, 1, 0, 0, 0
TimeSpanObj.SetEndTime -1, -1, -1, -1, -1, -1
n2=TimeSpanObj.GetTotalMinutes()

If n1>0 And n2<0 Then

End If

```

18.24 模拟变量值格式化为科学计数法表示的文本

- `x=1234000`, 要求 `s=1.234000e+006`
`StringObj.Format "%e", x`
`s=StringObj.GetText()`
- `x=1234000`, 要求 `s=1.234e+006`
`StringObj.Format "%.3e", x`
`s=StringObj.GetText()`
- `x=1234000`, 要求 `s=1.234e+06`
`StringObj.Format "%.3e", x`
`StringObj.Replace "+00", "+0"`
`StringObj.Replace "-00", "-0"`
`s=StringObj.GetText()`

18.25 从某文本文件读取内容并赋值给变量

- 假设文本文件内容如下:



- 只读取其中排放量浓度值, 脚本内容如下:

' 打开文本文件

```
bValue=TxtFileObj.OpenFile("c:\temp\HWReport_A.txt")
```

```
If bValue=1 Then
```

' 得到文本文件的长度

```
n=TxtFileObj.GetFileLength()
```

```
If n>0 Then
```

' 读取文本文件内容

```
If n>30000 Then n=30000
```

```
s=TxtFileObj.ReadFile(n)
```



```
' 建立字符串对象
StringObj.SetText s

' 得到C0值
n=StringObj.Find(" C0 ")
s=StringObj.Mid(n+4, 12)
x1=CSng(s)

' 得到CH4值
n=StringObj.Find(" CH4 ")
s=StringObj.Mid(n+5, 12)
x2=CSng(s)

' 得到C2H4值
n=StringObj.Find(" C2H4 ")
s=StringObj.Mid(n+6, 12)
x3=CSng(s)

' 得到C2H6值
n=StringObj.Find(" C2H6 ")
s=StringObj.Mid(n+6, 12)
x4=CSng(s)

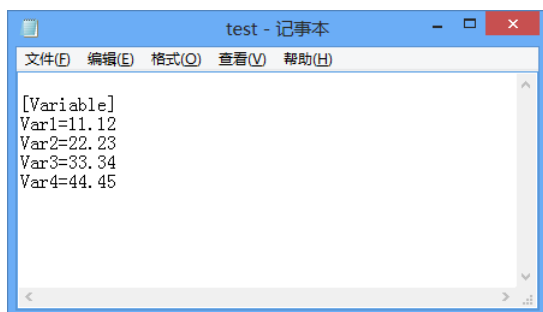
' 把值赋给变量
bValue=RunSys.OpenDatabase(3000)
If bValue=1 Then
    RunSys.SetVarValue VA, "%VA1", -1, x1
    RunSys.SetVarValue VA, "%VA2", -1, x2
    RunSys.SetVarValue VA, "%VA3", -1, x3
    RunSys.SetVarValue VA, "%VA4", -1, x4
    RunSys.CloseDatabase
End If
End If

' 关闭文本文件
TxtFileObj.CloseFile
End If
```

18.26 读写 INI 文本文件

□ 写入文本文件:

```
x1=RunSys.GetVarValue(VA,"%VA1",-1)
x2=RunSys.GetVarValue(VA,"%VA2",-1)
x3=RunSys.GetVarValue(VA,"%VA3",-1)
x4=RunSys.GetVarValue(VA,"%VA4",-1)
INIFile.WriteStringText "c:\temp\test.txt","Variable","Var1",CStr(x1)
INIFile.WriteStringText "c:\temp\test.txt","Variable","Var2",CStr(x2)
INIFile.WriteStringText "c:\temp\test.txt","Variable","Var3",CStr(x3)
INIFile.WriteStringText "c:\temp\test.txt","Variable","Var4",CStr(x4)
```



□ 读取INI文件:

```
s1=INIFile.GetStringText("c:\temp\test.txt","Variable","Var1")
s2=INIFile.GetStringText("c:\temp\test.txt","Variable","Var2")
s3=INIFile.GetStringText("c:\temp\test.txt","Variable","Var3")
s4=INIFile.GetStringText("c:\temp\test.txt","Variable","Var4")
bValue=RunSys.OpenDatabase(3000)
If bValue=1 Then
    RunSys.SetVarValue VA,"%VA1",-1,CSng(s1)
    RunSys.SetVarValue VA,"%VA2",-1,CSng(s2)
    RunSys.SetVarValue VA,"%VA3",-1,CSng(s3)
    RunSys.SetVarValue VA,"%VA4",-1,CSng(s4)
    RunSys.CloseDatabase
End If
```

18.27 获取某文件目录下最新建立的文件名称

'指定文件目录

```
strFolder = "c:\Temp"
```

'得到文件目录下的文件集合

```
Set objFSO = CreateObject("Scripting.FileSystemObject")
Set objFolder = objFSO.GetFolder(strFolder)
Set colFiles = objFolder.Files
```

```

' 枚举文件集合, 获取最新文件
newFile=""
Dim fileDate
For Each strFile in colFiles
    If newFile="" Then
        newFile=strFile.path
        fileDate=strFile.DateCreated
    Else
        fileDate1=strFile.DateCreated
        If DateDiff("s",fileDate,fileDate1)>0 Then
            newFile=strFile.path
            fileDate=strFile.DateCreated
        End If
    End If
Next
Set objFSO = Nothing
MsgBox "最新建立的文件是: "& newFile & " ("&fileDate&")"

```

18.28 获取某文件目录下最新修改的文件名称

```

strFolder = "c:\Temp"
Set objFSO = CreateObject("Scripting.FileSystemObject")
Set objFolder = objFSO.GetFolder(strFolder)
Set colFiles = objFolder.Files
' 枚举文件集合, 获取最新文件
newFile=""
Dim fileDate
For Each strFile in colFiles
    If newFile="" Then
        newFile=strFile.path
        fileDate=strFile.DateLastModified
    Else
        fileDate1=strFile.DateLastModified
        If DateDiff("s",fileDate,fileDate1)>0 Then
            newFile=strFile.path
            fileDate=strFile.DateLastModified
        End If
    End If
Next
Set objFSO = Nothing
MsgBox "最新修改的文件是: "& newFile & " ("&fileDate&")"

```

18.29 退出(Exit)脚本执行

□ 需求：如果条件不满足，则不再执行后续脚本，而退出执行；

□ 假设常规脚本内容：

```
x1=RunSys.GetVarValue(VA,"%VA1",-1)
If x1>0 Then
    x2=RunSys.GetVarValue(VA,"%VA2",-1)
    If x2>0 Then
        x3=RunSys.GetVarValue(VA,"%VA3",-1)
        If x3>0 Then
            RunSys.SetVarValue VA,"%VA4",-1,1
        End If
    End If
End If
```

用到了很多的if语句，条件很多时脚本的编写会很复杂；

□ 使用SUB过程，改进后内容：

```
Call mySub
Sub mySub
    x1=RunSys.GetVarValue(VA,"%VA1",-1)
    If x1=0 Then Exit Sub
    x2=RunSys.GetVarValue(VA,"%VA2",-1)
    If x2=0 Then Exit Sub
    x3=RunSys.GetVarValue(VA,"%VA3",-1)
    If x2=0 Then Exit Sub
    RunSys.SetVarValue VA,"%VA4",-1,1
End Sub
```

脚本内容结构和执行效率都会好些！

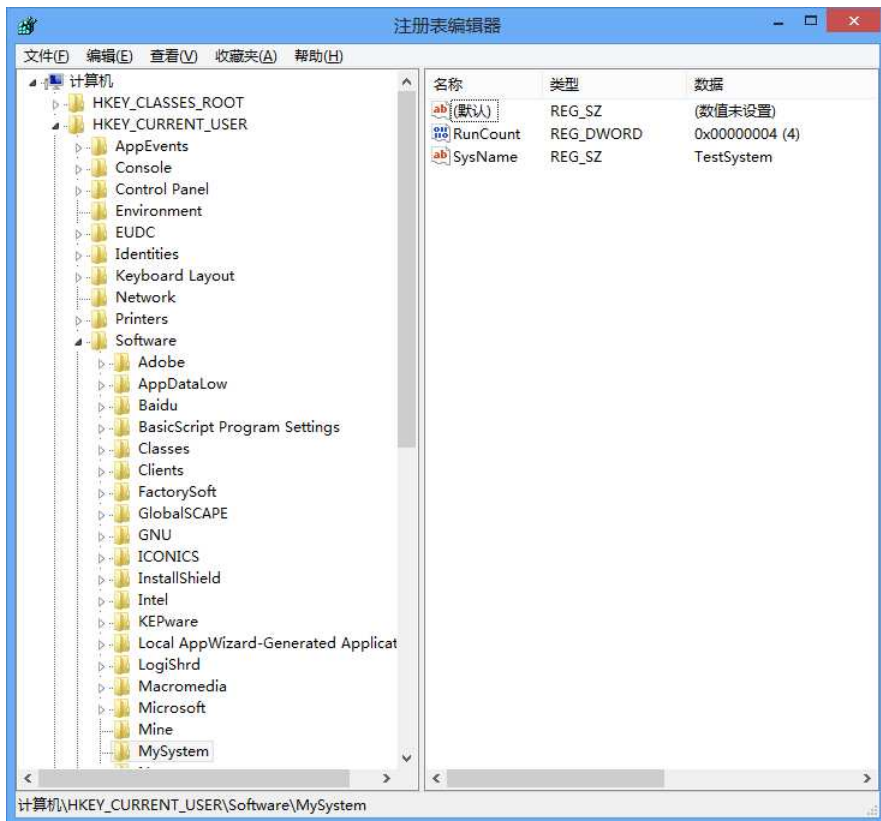
□ 也可以使用Function函数，改进后内容：

```
RunSys.SetVarValue VA,"%VA4",-1,myFunction()
Function myFunction()
    myFunction=0
    x1=RunSys.GetVarValue(VA,"%VA1",-1)
    If x1=0 Then Exit Function
    x2=RunSys.GetVarValue(VA,"%VA2",-1)
    If x2=0 Then Exit Function
    x3=RunSys.GetVarValue(VA,"%VA3",-1)
    If x3=0 Then Exit Function
    myFunction=1
End Function
```

18.30 访问注册表

□ 假设存在如下注册表项:

HKEY_CURRENT_USER\Software\MySystem\RunCount, HKEY_CURRENT_USER\Software\MySystem\SysName



□ 建立注册表项:

```
Set shell=CreateObject("WScript.Shell")
shell.RegWrite "HKEY_CURRENT_USER\Software\MySystem", ""
shell.RegWrite "HKEY_CURRENT_USER\Software\MySystem\RunCount", 0, "REG_DWORD"
shell.RegWrite "HKEY_CURRENT_USER\Software\MySystem\SysName", "", "REG_SZ"
Set shell=Nothing
```

□ 读写注册表项:

```
Set shell=CreateObject("WScript.Shell")
x=shell.RegRead("HKEY_CURRENT_USER\Software\MySystem\RunCount")
s=shell.RegRead("HKEY_CURRENT_USER\Software\MySystem\SysName")
x=x+1
s="TestSystem"
shell.RegWrite "HKEY_CURRENT_USER\Software\MySystem\RunCount", x, "REG_DWORD"
shell.RegWrite "HKEY_CURRENT_USER\Software\MySystem\SysName", s, "REG_SZ"
Set shell=Nothing
```

18.31 某批变量存储成 CSV 格式文件并 FTP 同步到其他计算机

- 需求:300个变量,以1秒频率存储到CSV格式文件中;每隔10分钟形成1个文件,压缩为Rar文件;
通过FTP方式同步到其他计算机;每天的24*6个文件放置到包含日期的子目录中,最多存储15天;
- 编写全局脚本文件,如脚本1:



脚本内容为: `UserDB.SaveVariablesToCsvFile "csvInfo.txt"`

脚本函数SaveVariablesToCsvFile描述:

- `UserDB.SaveVariablesToCsvFile csvInfoFile`
`csvInfoFile`, 配置文件, 缺省存放于MyFile目录下, 格式必须为:
`[Storage]`
`RootPath = C:\temp`
`FilePath = %Y%m%d`
`FileName = DATA_%H%M.csv`
`Interval = 10 (Minute)`
`HoldTime = 15 (Day)`
`Compress = Yes|No`
`RarPath = C:\Program Files\WinRAR`
`FileLog = CreateFile.log`
`[Ftp]`
`Upload = Yes|No`
`HostName = 192.168.1.100`
`UserName = anonymous`
`Password =`
`RootPath = \r`
`[Variables]`
`Var0001 = id1,1001`
`Var0002 = ID2,2001`

Var0003 = ID3, 3001

Var0004 = 时间, %Y-%m-%d %H:%M:%S

Var0005 = 温度1, AI. T1, %. 2f

.....

[Storage]定义文件存储方式:

RootPath, 存储根目录, 不设置则缺省为组态目录下的temp目录;

FilePath, 存储路径, 1天对应1个目录, 路径名称中须包含当前时间值: 年(%Y)月(%m)日(%d);

FileName, 文件名称, 文件名称中可包含当前时间值: 年(%Y)月(%m)日(%d)时(%H)分(%M);

Interval, 文件时间长度间隔, 时间单位为分钟(Minute);

HoldTime, 文件保持时间, 超出时间的文件被删除, 时间单位为天(Day);

Compress, 是否对文件经RAR进行压缩(Yes或No);

RarPath, Rar压缩程序所在路径;

FileLog, 是否存在存储根目录下生成日志文件, 进行FTP文件同步时用到;

[Ftp]定义是否经FTP与其他计算机进行文件同步;

Upload, 是否上传同步(Yes或No);

HostName, FTP主机名称或IP地址;

UserName, FTP登录用户名称, 匿名登录使用'anonymous';

Password, FTP登录用户口令, 匿名登录为空;

RootPath, FTP存储路径;

[Variables]定义变量信息:

Varxxxx, 可以定义1-9999个变量信息;

变量信息格式1: "标题, 常量", 如"Var0001=ID1, 1001";

变量信息格式2: "标题, 当前时间", 如"Var0004 = 时间, %Y-%m-%d %H:%M:%S";

变量信息格式3: "标题, 变量名称, 数值格式", 如"Var0005 = 温度1, AI. T1, %. 2f";

□ 定义运行策略:

全局脚本运行策略(最多100个)...

编号: [001] << >> 保存

名称: [] X

执行脚本[最多16个]:

- + 脚本1

☒ 时间触发

☐ 开始时间: 2000-01-01 00:00:00

☒ 周期执行: 1 秒

☐ 首次执行:

☐ 退出执行:

☐ Or (或) ☐ And (与)

☐ 变量触发[精度>=200毫秒]

☐ 变量值发生变化时, 执行1次:

☐ 变量值正跳变/增加时, 执行1次:

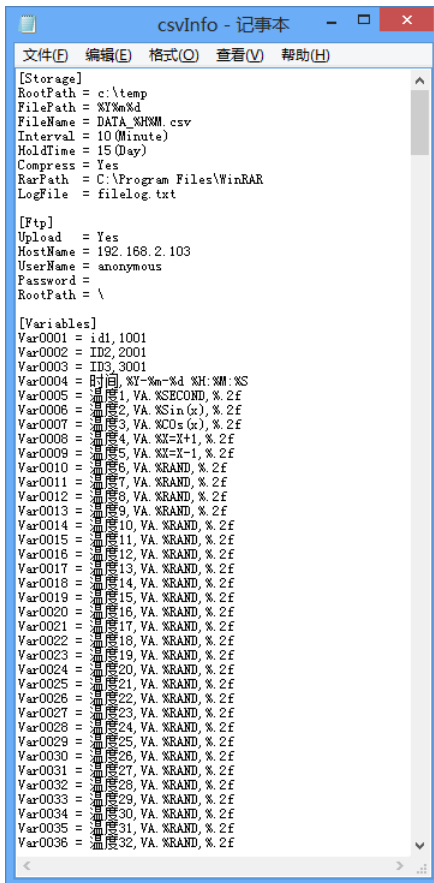
☐ 变量值负跳变/减少时, 执行1次:

☐ 变量值 > 0 时, 循环执行:

☐ 变量值 = 0 时, 循环执行:

☐ 变量值 < 0 时, 循环执行:

□ MyFile目录下编写csvInfo.txt文件:



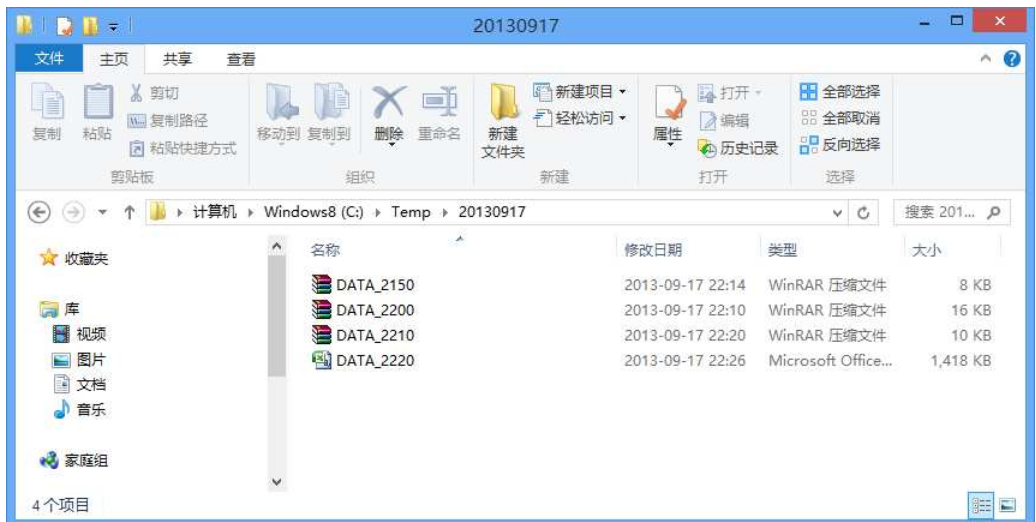
```

[Storage]
RootPath = c:\temp
FilePath = %X%M%k%
FileName = DATA_%X%M.csv
Interval = 10 (Minute)
HoldTime = 15 (Day)
Compress = Yes
RarPath = C:\Program Files\WinRAR
LogFile = filelog.txt

[Ftp]
Upload = Yes
HostName = 192.168.2.103
UserName = anonymous
Password =
RootPath = \

[Variables]
Var0001 = id1,1001
Var0002 = ID2,2001
Var0003 = ID3,3001
Var0004 = 时间, %X-%M-%d %H:%M:%S
Var0005 = 温度1, VA.%SECOND, %2f
Var0006 = 湿度2, VA.%Sin(x), %2f
Var0007 = 湿度3, VA.%Cos(x), %2f
Var0008 = 湿度4, VA.%X=X+1, %2f
Var0009 = 湿度5, VA.%X=X-1, %2f
Var0010 = 湿度6, VA.%RAND, %2f
Var0011 = 湿度7, VA.%RAND, %2f
Var0012 = 湿度8, VA.%RAND, %2f
Var0013 = 湿度9, VA.%RAND, %2f
Var0014 = 湿度10, VA.%RAND, %2f
Var0015 = 湿度11, VA.%RAND, %2f
Var0016 = 湿度12, VA.%RAND, %2f
Var0017 = 湿度13, VA.%RAND, %2f
Var0018 = 湿度14, VA.%RAND, %2f
Var0019 = 湿度15, VA.%RAND, %2f
Var0020 = 湿度16, VA.%RAND, %2f
Var0021 = 湿度17, VA.%RAND, %2f
Var0022 = 湿度18, VA.%RAND, %2f
Var0023 = 湿度19, VA.%RAND, %2f
Var0024 = 湿度20, VA.%RAND, %2f
Var0025 = 湿度21, VA.%RAND, %2f
Var0026 = 湿度22, VA.%RAND, %2f
Var0027 = 湿度23, VA.%RAND, %2f
Var0028 = 湿度24, VA.%RAND, %2f
Var0029 = 湿度25, VA.%RAND, %2f
Var0030 = 湿度26, VA.%RAND, %2f
Var0031 = 湿度27, VA.%RAND, %2f
Var0032 = 湿度28, VA.%RAND, %2f
Var0033 = 湿度29, VA.%RAND, %2f
Var0034 = 湿度30, VA.%RAND, %2f
Var0035 = 湿度31, VA.%RAND, %2f
Var0036 = 湿度32, VA.%RAND, %2f
  
```

□ 运行结果:



18.32 使用 Split 函数转换字符串到数组

```
s="100, 200, 300, 400"
ts=Split(s, ",")
RunSys.SetVarValue VA, "%VA1", -1, CDb1(ts(0))
RunSys.SetVarValue VA, "%VA2", -1, CDb1(ts(1))
RunSys.SetVarValue VA, "%VA3", -1, CDb1(ts(2))
RunSys.SetVarValue VA, "%VA4", -1, CDb1(ts(3))
```

18.33 记录操作日志到文本文件

```
' 得到组态安装主目录
path=RunSys.GetInstallPath()
' 每天1个日志文件, 文件名含时间
fileName=CurrentTime.GetFormatTime("File_%Y%m%d.log")
' 记录到安装目录的临时子目录下
If TxtFileObj.OpenFile(path+"\temp\"+fileName)=1 Then
    ' 移动光标到文件最后
    TxtFileObj.SeekPosToEnd
    ' 组织日志文本: yyyy-mm-dd HH:MM:SS = text
    s=CurrentTime.GetFormatTime("%Y-%m-%d %H:%M:%S = ")
    s=s+"启动设备"+vbCrLf
    ' 写入日志内容
    TxtFileObj.WriteFile s, 0
    ' 关闭文件
    TxtFileObj.CloseFile
End If
```