

Vbs 操作 EXCEL 方法总结:

' \--**打开 Excel 电子表格**

' |--**说明**

' |--**演示脚本打开名为 C:\Scripts\New_users.xls 的现有 Excel 电子表格。**

```
1. Set objExcel = CreateObject("Excel.Application")
2. Set objWorkbook = objExcel.Workbooks.Open("C:\Scripts\New_users.xls")
```

复制代码

' \--**读取 Excel 电子表格**

' |--**说明**

' |--**演示脚本读取名为 C:\Scripts\New_users.xls 的电子表格中存储的值。**

```
1. Set objExcel = CreateObject("Excel.Application")
2. Set objWorkbook = objExcel.Workbooks.Open("C:\Scripts\New_users.xls")
3. intRow = 2
4. Do Until objExcel.Cells(intRow,1).Value = ""
5.     Wscript.Echo "CN: " & objExcel.Cells(intRow, 1).Value
6.     Wscript.Echo "sAMAccountName: " & objExcel.Cells(intRow, 2).Value
7.     Wscript.Echo "GivenName: " & objExcel.Cells(intRow, 3).Value
8.     Wscript.Echo "LastName: " & objExcel.Cells(intRow, 4).Value
9.     intRow = intRow + 1
10. Loop
11. objExcel.Quit
```

复制代码

' \--**自动调整 Microsoft Excel 工作表中的列**

' |--**说明**

'|--演示脚本检索计算机上安装的服务的相关信息，将该数据写入到 Excel 电子表格中，然后使用 Autofit() 方法根据该数据来自动调整列。

```
1. Set objExcel = CreateObject("Excel.Application")
2. objExcel.Visible = True
3. Set objWorkbook = objExcel.Workbooks.Add()
4. Set objWorksheet = objWorkbook.Worksheets(1)
5. x = 1
6. strComputer = "."
7. Set objWMIService = _
8.     GetObject("winmgmts:\\." & strComputer & "\root\cimv2")
9. Set colItems = objWMIService.ExecQuery _
10.    ("Select * from Win32_Service")
11. For Each objItem in colItems
12.     objWorksheet.Cells(x, 1) = objItem.Name
13.     objWorksheet.Cells(x, 2) = objItem.DisplayName
14.     objWorksheet.Cells(x, 3) = objItem.State
15.     x = x + 1
16. Next
17. Set objRange = objWorksheet.UsedRange
18. objRange.EntireColumn.Autofit()
```

复制代码

'|--将数据添加到电子表格单元格中

```
1. Set objExcel = CreateObject("Excel.Application")
2. objExcel.Visible = True
3. objExcel.Workbooks.Add
4. objExcel.Cells(1, 1).Value = "Test value" '向单元格 1,1 中写入"Test value"
```

复制代码

```
' \---将带格式的数据添加到电子表格中
' |---说明
' |---演示脚本将文字“test value”添加到新的电子表格中，然后设置包含该值的单元格的格式。
```

```
1. Set objExcel = CreateObject("Excel.Application")
2. objExcel.Visible = True
3. objExcel.Workbooks.Add
4. objExcel.Cells(1, 1).Value = "Test value" '写入内容
5. objExcel.Cells(1, 1).Font.Bold = TRUE '加粗为真
6. objExcel.Cells(1, 1).Font.Size = 24 '单元格大小为 24
7. objExcel.Cells(1, 1).Font.ColorIndex = 3 '颜色号为 3
```

[复制代码](#)

```
' \---列出 Excel 颜色值
' |---说明
' |---演示脚本显示在以编程方式控制 Microsoft Excel 时可用的各种颜色（及其相关颜色索引）。
```

```
1. Set objExcel = CreateObject("Excel.Application")
2. objExcel.Visible = True
3. objExcel.Workbooks.Add
4. For i = 1 to 56 '一共有 56 种颜色可供选择
5.     objExcel.Cells(i, 1).Value = i
6.     objExcel.Cells(i, 1).Interior.ColorIndex = i
7. Next
```

[复制代码](#)

' \---列出电子表格中的服务数据

' |---说明

' |---演示脚本检索计算机上运行的每种服务的相关信息，然后在 Excel 电子表格中显示该数据。

```
1. Set objExcel = CreateObject("Excel.Application")
2. objExcel.Visible = True
3. objExcel.Workbooks.Add
4. x = 1
5. strComputer = "."
6. Set objWMIService = GetObject _
7.     ("winmgmts:\\\" & strComputer & "\\root\cimv2")
8. Set colServices = objWMIService.ExecQuery _
9.     ("Select * From Win32_Service")
10. For Each objService in colServices
11.     objExcel.Cells(x, 1) = objService.Name
12.     objExcel.Cells(x, 2) = objService.State
13.     x = x + 1
14. Next
```

[复制代码](#)

' \---对 Excel 电子表格应用自动套用格式

' |---说明

' |---演示脚本将数据添加到 Excel 电子表格中，然后对该数据应用 AutoFormat(xpRangeAutoFormatList2)。

```
1. Const xpRangeAutoFormatList2 = 11
2. Set objExcel = CreateObject("Excel.Application")
3. objExcel.Visible = True
4. Set objWorkbook = objExcel.Workbooks.Add()
5. Set objWorksheet = objWorkbook.Worksheets(1)
```

```
6. k = 1
7. For i = 1 to 10
8.     For j = 1 to 10
9.         objWorksheet.Cells(i,j) = k
10.        k = k + 1
11.     Next
12. Next
13. Set objRange = objWorksheet.UsedRange
14. objRange.AutoFormat(xpRangeAutoFormatList2)
```

复制代码

```
' \--基于电子表格中的信息来创建用户帐户
' |--说明
' |--演示脚本基于 Excel 电子表格中存储的信息来创建 Active Directory 用户帐户。
```

```
1. Set objExcel = CreateObject("Excel.Application")
2. Set objWorkbook = objExcel.Workbooks.Open("C:\Scripts\New_users.xls")
3. intRow = 2
4. Do Until objExcel.Cells(intRow,1).Value = ""
5.     Set objOU = GetObject("ou=Finance, dc=fabrikam, dc=com")
6.     Set objUser = objOU.Create("User", "cn=" & objExcel.Cells(intRow, 1).Value)
7.     objUser.sAMAccountName = objExcel.Cells(intRow, 2).Value
8.     objUser.GivenName = objExcel.Cells(intRow, 3).Value
9.     objUser.SN = objExcel.Cells(intRow, 4).Value
10.    objUser.AccountDisabled = FALSE
11.    objUser.SetInfo
12.    intRow = intRow + 1
13. Loop
14. objExcel.Quit
```

复制代码

' \---对 Microsoft Excel 工作表进行排序

' |---说明

' |---演示脚本将数据添加到 Microsoft Excel 电子表格中，然后在列 A 中对该数据进行排序。

```
1. Set objExcel = CreateObject("Excel.Application")
2. objExcel.Visible = True
3. Set objWorkbook = objExcel.Workbooks.Add
4. Set objWorksheet = objWorkbook.Worksheets(1)
5. objExcel.Cells(1, 1).Value = "4"
6. objExcel.Cells(2, 1).Value = "1"
7. objExcel.Cells(3, 1).Value = "2"
8. objExcel.Cells(4, 1).Value = "3"
9. objExcel.Cells(1, 2).Value = "A"
10. objExcel.Cells(2, 2).Value = "B"
11. objExcel.Cells(3, 2).Value = "C"
12. objExcel.Cells(4, 2).Value = "D"
13. Set objRange = objWorksheet.UsedRange
14. Set objRange2 = objExcel.Range("A1")
15. objRange.Sort(objRange2)
```

[复制代码](#)

' \---设置一组单元格的格式

' |---说明

' |---演示脚本将数据添加到电子表格的四个不同的单元格中，然后使用 Range 对象同时设置多个单元格的格式。

```
1. Set objExcel = CreateObject("Excel.Application")
2. objExcel.Visible = True
```

```
3. objExcel.Workbooks.Add
4. objExcel.Cells(1, 1).Value = "Name"
5. objExcel.Cells(1, 1).Font.Bold = TRUE
6. objExcel.Cells(1, 1).Interior.ColorIndex = 30
7. objExcel.Cells(1, 1).Font.ColorIndex = 2
8. objExcel.Cells(2, 1).Value = "Test value 1"
9. objExcel.Cells(3, 1).Value = "Test value 2"
10. objExcel.Cells(4, 1).Value = "Tets value 3"
11. objExcel.Cells(5, 1).Value = "Test value 4"
12. Set objRange = objExcel.Range("A1","A5")
13. objRange.Font.Size = 14
14. Set objRange = objExcel.Range("A2","A5")
15. objRange.Interior.ColorIndex = 36
16. Set objRange = objExcel.ActiveCell.EntireColumn
17. objRange.AutoFit()
```

复制代码

'**---在三个不同的列上对 Excel 电子表格进行排序**

'**---说明**

'**---演示脚本打开 Excel 电子表格，然后对列 A（升序）、列 B（降序）和列 C（升序）中的数据进行排序。此脚本假定工作表包含一个标题行（由 xlYes 来指示，它是 Sort 方法中的最后一个参数）。脚本需要一个名为 C:\Scripts\Sort_test.xls 的 Excel 文件。**

```
1. Const xlAscending = 1
2. Const xlDescending = 2
3. Const xlYes = 1
4. Set objExcel = CreateObject("Excel.Application")
5. objExcel.Visible = True
6. Set objExcel = CreateObject("Excel.Application")
7. objExcel.Visible = True
```

```
8. Set objWorkbook = _
9.     objExcel.Workbooks.Open("C:\Scripts\Sort_test.xls")
10. Set objWorksheet = objWorkbook.Worksheets(1)
11. Set objRange = objWorksheet.UsedRange
12. Set objRange2 = objExcel.Range("A1")
13. Set objRange3 = objExcel.Range("B1")
14. Set objRange4 = objExcel.Range("C1")
15. objRange.Sort objRange2,xlAscending,objRange3,,xlDescending, _
16.     objRange4,xlDescending,xlYes
```

复制代码

```
' \--列出电子表格中的 Active Directory 数据
' |--说明
' |--演示脚本从 Active Directory 检索数据，然后在 Excel 电子表格中显示该数据。
```

```
1. Const ADS_SCOPE_SUBTREE = 2
2. Set objExcel = CreateObject("Excel.Application")
3. objExcel.Visible = True
4. objExcel.Workbooks.Add
5. objExcel.Cells(1, 1).Value = "Last name"
6. objExcel.Cells(1, 2).Value = "First name"
7. objExcel.Cells(1, 3).Value = "Department"
8. objExcel.Cells(1, 4).Value = "Phone number"
9. Set objConnection = CreateObject("ADODB.Connection")
10. Set objCommand = CreateObject("ADODB.Command")
11. objConnection.Provider = "AdsDSOObject"
12. objConnection.Open "Active Directory Provider"
13. Set objCommand.ActiveConnection = objConnection
14. objCommand.Properties("Page Size") = 100
15. objCommand.Properties("Searchscope") = ADS_SCOPE_SUBTREE
16. objCommand.CommandText = _
```



```
17.     "SELECT givenName, SN, department, telephoneNumber FROM " _
18.         & "'LDAP://dc=fabrikam,dc=microsoft,dc=com' WHERE " _
19.         & "objectCategory='user'"
20. Set objRecordSet = objCommand.Execute
21. objRecordSet.MoveFirst
22. x = 2
23. Do Until objRecordSet.EOF
24.     objExcel.Cells(x, 1).Value = _
25.         objRecordSet.Fields("SN").Value
26.     objExcel.Cells(x, 2).Value = _
27.         objRecordSet.Fields("givenName").Value
28.     objExcel.Cells(x, 3).Value = _
29.         objRecordSet.Fields("department").Value
30.     objExcel.Cells(x, 4).Value = _
31.         objRecordSet.Fields("telephoneNumber").Value
32.     x = x + 1
33.     objRecordSet.MoveNext
34. Loop
35. Set objRange = objExcel.Range("A1")
36. objRange.Activate
37. Set objRange = objExcel.ActiveCell.EntireColumn
38. objRange.Autofit()
39. Set objRange = objExcel.Range("B1")
40. objRange.Activate
41. Set objRange = objExcel.ActiveCell.EntireColumn
42. objRange.Autofit()
43. Set objRange = objExcel.Range("C1")
44. objRange.Activate
45. Set objRange = objExcel.ActiveCell.EntireColumn
46. objRange.Autofit()
47. Set objRange = objExcel.Range("D1")
48. objRange.Activate
49. Set objRange = objExcel.ActiveCell.EntireColumn
50. objRange.Autofit()
```

```
51. Set objRange = objExcel.Range("A1").SpecialCells(11)
52. Set objRange2 = objExcel.Range("C1")
53. Set objRange3 = objExcel.Range("A1")
```

[复制代码](#)