

Re: Ghostly presence of Excel process

Source:

<http://www.tech-archive.net/Archive/Access/microsoft.public.access.modulesdaovba/2006-03/msg01223.html>

- *From:* RoyVidar <roy_vidarNOSPAM@xxxxxxxxx>
 - *Date:* Wed, 29 Mar 2006 23:45:19 +0200
-

ragtopcaddy via AccessMonster.com wrote in message <5dff1dda7472e@uwe> :

The code below runs at the end of a sub that exports queries via a transferspreadsheet routine to a new workbook which then has to be formatted, due to the fact that Excel inexplicably turns all of my numbers into text, and opened so the user can see his handiwork. Even on those rare occasions when I don't get runtime error '462', and the code executes and opens the workbook, closing the workbook manually leaves an instance of Excel.exe running in Task Manager Processes, though not in Applications. Inevitably, this results in runtime error '462' the next time I try to run the code. What a nightmare! I'm slowly discovering how many ways one can "close" Excel without closing it! Any suggestions would be appreciated. I've spent the entire day searching and reading posts.

Thanks,

Bill R

StripXLFormats strPath & strFile

```
bIXLRunning = IsExcelRunning
If Not bIXLRunning Then
Set xlObj = New Excel.Application
Else
Set xlObj = GetObject(, "Excel.Application")
End If
Set WkBk = xlObj.Workbooks.Open(strPath & strFile)
xlObj.Visible = True
Set WkBk = Nothing
Set xlObj = Nothing
bIXLRunning = IsExcelRunning
If bIXLRunning Then
MsgBox "Excel is STILL running!", vbExclamation & vbOKOnly, "EXCEL"
Stop
End If
End If
```

Re: Ghostly presence of Excel process

```
Function StripXLFormats(strWkBk As String, Optional strWkSht As String)
Dim WkBk As Excel.Workbook
Dim WkSht As Excel.Worksheet
Dim Cell As Excel.Range
Dim bl1Sheet As Boolean
Dim xlObj As Object
Dim blXLRRunning As Boolean

blXLRRunning = IsExcelRunning

If strWkSht <> "" Then
bl1Sheet = True
GoTo OneSheet
End If

'Set xlObj = CreateObject("excel.application")
If Not blXLRRunning Then
Set xlObj = New Excel.Application
Else
Set xlObj = GetObject(, "Excel.Application")
End If
Set WkBk = xlObj.Workbooks.Open(strWkBk)
For Each WkSht In WkBk.Worksheets
OneSheet:
WkSht.Activate
WkSht.Range("A1").Activate
WkSht.Range("A1").CurrentRegion.Select
For Each Cell In Selection
If Len(Cell.Value) >= 1 And Cell.PrefixCharacter = "" Then
With Cell
.NumberFormat = "@" 'keep it text
.Value = .Value
If InStr(.Value, "%") > 0 Then
.Replace What:="%", Replacement:"", LookAt:=xlPart
If IsNumeric(.Value) And .Row > 1 Then
.Value = .Value / 100
.NumberFormat = "#%"
End If
ElseIf InStr(.Value, "$") > 0 Then
.Replace What:="$", Replacement:"", LookAt:=xlPart
If IsNumeric(.Value) And .Row > 1 Then
.Value = .Value
.NumberFormat = "$#,##0.00##"
End If
ElseIf IsNumeric(.Value) And .Row > 1 Then
.Value = .Value
.NumberFormat = "#0.0#####"
End If
End With
End If
Next Cell
```

Re: Ghostly presence of Excel process

```
Cells.Select
Cells.EntireColumn.AutoFit
WkSht.Range("A1").Select
If bl1Sheet Then GoTo OuttaHere
Next WkSht

OuttaHere:
Set WkSht = Nothing

With WkBk
.Sheets(1).Select
.Save
.Close
End With
Set WkBk = Nothing
xlObj.Quit
Set xlObj = Nothing
'After "closing" excel in the above code, the following is always true
blXLRunning = IsExcelRunning

If blXLRunning Then
MsgBox "Excel is STILL running!", vbExclamation & vbOKOnly, "EXCEL"
Stop
End If

End Function

Public Function IsExcelRunning() As Boolean
Dim xl As Object

On Error Resume Next

Set xl = GetObject(, "Excel.Application")
IsExcelRunning = (Err.Number = 0)

Set xl = Nothing
Err.Clear
End Function
```

You are making what I call unqualified referencing

For Each Cell In Selection

uses excels selection object implicitly, and most likely is at least one of the reasons for your extra instance of excel in memory.

Since excel also has a cell object, I'd probably alter the name of that variable Cell -> objCell

For Each objCell In xlObj.Selection

Re: Ghostly presence of Excel process

Re: Ghostly presence of Excel process

Likewise with
Cells.Select
Cells.EntireColumn.AutoFit

Try
xlObj.Cells.Select
xlObj.Cells.EntireColumn.AutoFit

For more details, check out
<http://support.microsoft.com/default.aspx?kbid=178510>

—
Roy-Vidar

.