



Products Services Developer Resources Contact STS About STS



Communicating With Microsoft Excel from Revolution



> Is there any way in which data (e.g. a number) calculated > in a Revolution stack can be automatically transferred to a > given cell (field) in an Excel program? Thanks again.

Sure! Here you go...

Mac Approach

Here's the way to do it for Mac. The basic script for AppleScript is like this:

```
tell application "Microsoft Excel"
    set value of cell "B3" to 500
end tell
```

To translate this to Revolution, you'd do this (I use a q() function for quoting data that makes it easier to work with):

```
on mouseUp
    SendToXL "500", "B3"
end mouseUp

on SendToXL pWhat, pCell
    put "tell app" && q("Microsoft Excel") & cr & \
        "set value of cell" && q(pCell) && "to" && q(pWhat) & cr & \
        "end tell" into tScript
    do tScript as AppleScript
end SendToXL

function q pWhat
    return quote & pWhat & quote
end q
```

Windows Approach

Here's the way to do it for Windows. The basic VB Script code looks like this:

```
Dim ObjXL
Set ObjXL = GetObject(,"Excel.Application")
ObjXL.Range("B3").Value = "500"
```

To translate this to Revolution, you need to output this to a .vbs file and "run" it, and then delete it when you're done. Here's how:

```
on mouseUp
    SendToXL "500", "B3"
end mouseUp

on SendToXL pWhat, pCell
    put "Dim ObjXL" & cr & \
        "Set ObjXL = GetObject(," & q("Excel.Application") & ")" & cr & \
        "ObjXL.Range(" & q(pCell) & ").Value =" && q(pWhat) into tScript

    -- Put the script into a file on disk
    put "C:\VBSTemp.vbs" into tFile
    put tScript into url("file:" & tFile)

    -- Run the file
    set the hideConsoleWindows to true
    get shell("cscript.exe //nologo" && tFile)

    -- Now, delete the file; best way is to give it 1 second to complete
    -- before deleting, so I'll use the "send in " method to
    -- do this:
    send "delete file" && q(tFile) to me in 1 second
end SendToXL
```

If you're developing cross-platform, you can merge them into the same handler by checking "the platform":

```
on SendToXL pWhat, pCell
    switch (the platform)
        case "MacOS"
            -- put the Mac code here
            break
        case "Win32"
            -- put the Windows code here
            break
```

```

end switch
end SendToXL

```

Full Version

Here's a fully comprehensive version that will get and set both individual cells and ranges (both named and in A1:B1 style) that is cross-platform to boot:

```

on SendToXL pWhat,pRangeRef
-- assumes tab & cr delimited data
set the itemDel to tab
switch (the platform)
case "MacOS"
    put "" into tList
    repeat for each line tLine in pWhat
        replace quote with numToChar(1) in tLine -- temporary
        replace tab with (quote & "," & quote) in tLine
        put quote & tLine & quote into tList
        replace numToChar(1) with ("\" & quote) in tList
        put "{" & tLine & "}" & "," after tList
    end repeat
    delete char -1 of tList
    put "tell app" && q("Microsoft Excel") & cr & \
        "set the value of range" && q(pRangeRef) && "to" && "{" & tList & "}" & cr & \
        "end tell" into tScript
    do tScript as AppleScript
    break
case "Win32"
    put the number of items of line 1 of pWhat into tNumCols
    put the number of lines of pWhat into tNumRows

    put "Dim ObjXL,tRetVal,tRow,tCol" & cr & \
        "Dim tDataA(" & tNumCols & "," & tNumRows & ")" & cr & \
        "Set ObjXL = GetObject(", " & q("Excel.Application") & ")") into tScript
    repeat with x = 1 to tNumCols
        repeat with y = 1 to tNumRows
            put tScript & cr & "tDataA(" & x & "," & y & ") = " & q(item x of line y of pWhat) into tScript
        end repeat
    end repeat
    put tScript & cr & "For tRow = 1 To" && tNumRows & cr & \
        "For tCol = 1 to" && tNumCols & cr & \
        "ObjXL.Range(" & q(pRangeRef) & ").Cells(tRow,tCol).Value = tDataA(tCol,tRow)" & cr & \
        "Next" & cr & "Next" into tScript

    put "C:\VBSTemp.vbs" into tFile
    put tScript into url("file:" & tFile)
    set the hideConsoleWindows to true
    get shell("cscript.exe //nologo" && tFile)
    send "delete file" && q(tFile) to me in 1 second
    break
end switch
end SendToXL

function GetFromXL pRangeRef
switch (the platform)
case "MacOS"
    put "tell app" && q("Microsoft Excel") & cr & \
        "get the value of range" && q(pRangeRef) & cr & \
        "end tell" into tScript
    do tScript as AppleScript
    put the result into tData

    replace "{" with "{" in tData
    replace "}" with "}" in tData
    replace ", " with "(" & cr) in tData
    replace ("\" & quote) with numToChar(1) in tData
    replace (quote & ", ") with tab in tData
    if char 1 of tData = "{" then delete char 1 of tData
    if char -1 of tData = "}" then delete char -1 of tData
    replace ("}" & cr & "{") with CR in tData
    replace quote with "" in tData
    replace numToChar(1) with quote in tData
    return tData
    break
case "Win32"
    put "Dim ObjXL,tNumRows,tNumCols,tRetVal,tRow,tCol" & cr & \
        "Set ObjXL = GetObject(", " & q("Excel.Application") & ")") & cr & \
        "tNumRows = ObjXL.Range(" & q(pRangeRef) & ").Rows.Count" & cr & \
        "tNumCols = ObjXL.Range(" & q(pRangeRef) & ").Columns.Count" into tScript

    put tScript & cr & "For tRow = 1 To tNumRows" & cr & \
        "For tCol = 1 to tNumCols" & cr & \
        "If tCol <> tNumCols Then" & cr & \
        "tRetVal = tRetVal & ObjXL.Range(" & q(pRangeRef) & ").Cells(tRow,tCol).Value & vbTab" & cr & \

```

```

"Else" & cr & \
"RetVal = tRetVal & ObjXL.Range(" & q(pRangeRef) & ").Cells(tRow,tCol).Value & vbCrLf" & cr & \
"End If" & cr & "Next" & cr & "Next" into tScript

put tScript & cr & "RetVal = Left(tRetVal,Len(tRetVal) - 2)" & cr & \
"WScript.Echo tRetVal" into tScript

put "C:\VBSTemp.vbs" into tFile
put tScript into url("file:" & tFile)
set the hideConsoleWindows to true
get shell("cscript.exe //nologo" && tFile)
send "delete file" && q(tFile) to me in 1 second
if char -1 of it is CR then delete char -1 of it -- strip any trailing CR
return it
break
end switch
end GetFromXL

```

Examples

```

SendToXL "500","A1" -- specific cell
SendToXL "Ken" & tab & "100" & cr & "John" & tab & "500","A1:B2" -- specified range
SendToXL "Ken" & tab & "100" & cr & "John" & tab & "500","Scores" -- named range

```

```

put GetFromXL("A1")
--> 500

```

```

put GetFromXL("A1:B2")
--> Ken      100
    John     500

```

```

put GetFromXL("Scores")
--> Ken      100
    John     500

```

Posted 3/18/2005 by Ken Ray to the Use-Revolution List

Updated 3/3/2006 by Ken Ray to add sending/receiving ranges

 [Print this tip](#)

[News and Rumors](#)
 [Products](#)
 [Services](#)
 [Developer Resources](#)
 [Contact STS](#)
 [About STS](#)

Copyright ©1997-2013 Sons of Thunder Software, Inc. All rights reserved.
 Send all comments to webmaster@sonsofthunder.com.
