

Code sample: Compare the code

This is an example of a Visual Basic for Applications (VBA) macro that is used to call one Microsoft Office application from another. In this case, a macro in Microsoft Excel calls some code in Microsoft Word.

The example shows the code that needs to be removed in AppleScript. The code that invokes an existing or new instance of Word has to be removed, as does every Dim declaration, and some error handling.

[The VBA version](#)

The following code is the VBA subroutine:

```
Sub ControlWordFromXL()

    Dim oWord As Word.Application
    Dim WordWasNotRunning As Boolean
    Dim oDoc As Word.Document
    Dim myDialog As Word.Dialog
    Dim UserButton As Long

    'Get existing instance of Word if it's open;
    'otherwise create a new one

    On Error Resume Next

    Set oWord = GetObject(, "Word.Application")
    If Err Then
        Set oWord = New Word.Application
        WordWasNotRunning = True
    End If

    On Error GoTo Err_Handler

    oWord.Visible = True
    oWord.Activate
    Set oDoc = oWord.Documents.Add

    ' code here to manipulate the document

    oDoc.Close savechanges:=wdDoNotSaveChanges

    If WordWasNotRunning Then
        oWord.Quit
    End If

    'Make sure you release object references.

    Set oWord = Nothing
    Set oDoc = Nothing
    Set myDialog = Nothing

    'quit
    Exit Sub

Err_Handler:
    MsgBox "Word caused a problem. " &
        Err.Description, vbCritical, "Error: " _
        & Err.Number
    If WordWasNotRunning Then
        oWord.Quit
    End If

End Sub
```

[The AppleScript version](#)

The following code is the AppleScript subroutine. You use a subroutine rather than top-level script when you want to use the same code frequently in several scripts.

```
to ControlWord()
    -- no need for any Dim statements, a try/error
    -- block to invoke Word, or to create a new
    -- instance of Word specifically, but check to
    -- see if Word is already running, for later
    tell application "System Events"
        set wordWasRunning to exists process "Microsoft Word"
    end tell
    try
        tell application "Microsoft Word"
```

```

        -- no need to make Word visible,
        -- nor the option*
        activate
        set oDoc to make new document

        -- code here to manipulate the document
        close oDoc saving no
        if not wordWasRunning then quit
    end tell
on error errMsg number errNum
    -- bring current application
    -- (e.g., Excel) to the front to view
    -- dialog
    activate
    display dialog "Word caused a problem. "
        & errMsg & return & "Error: " & errNum
        with icon 0
    if not wordWasRunning then
        tell application "Microsoft Word" to quit
    end if
end try
-- no need to release object references
end ControlWord

```

Comparing the two

System Events is a background application with a very interesting dictionary. In this example, System Events checks whether Word as a **process** exists since there's no way to mimic the VBA method of trying to create a new Word instance. In VBA, you receive an error if one exists already.

Note that the **application** class in the Word dictionary does not have a **visible** property. However, **window** does, so you can hide and show individual windows.

If you want to hide and show the application, you do that in System Events, where you can set the **visible** property of any process, including Word. The Finder can do the same thing with "legacy" commands, but these process features are actually done by System Events. The Finder simply passes on the command. This is a basic difference between Macintosh and Windows. The system, rather than the application itself, controls things like the visibility of applications.

Also note that in AppleScript you need to use **try** in order to catch an error. If you want to put the **try** block around the entire `ControlWord()` handler, put it around the call to the handler, not inside of the handler.

In this example, when you are using Mac OS X v10.4 (Tiger), the call to Word to quit inside the `on error` block works. However, in Mac OS X v10.3 (Panther) and earlier versions, if Word has already quit, this new `tell` statement launches Word only to quit it.