



Products Services Developer Resources Contact STS About STS



## Communicating With Microsoft Outlook from Revolution



> Basically, what I would like to do is allow a user to generate an email  
 > from an email address field within the stack, attach a pdf file (the  
 > path of which will also be stored somewhere within the stack), and then  
 > either send the message automatically or open it on screen for review  
 > and allow the user to click the "send" button manually.  
 >  
 > A more critical task is scheduling- I could save myself a tremendous  
 > amount of time and energy in development if I could somehow schedule a  
 > task within Outlook's calendar. Being able to use Revolution to stick an  
 > appointment in for two hours on May 25th at 2PM with a reminder 15  
 > minutes before would be a huge benefit to me.

OK, I'll do them in reverse order (Calendar, then Mail) because the mail stuff is a bit trickier...

### **Manipulating Outlook's Calendar**

Here's the basic VBScript, with placeholders surrounded in double angle brackets << >> :

```
Const olCalendarFolder = 9

Set oOutlook = WScript.CreateObject("Outlook.Application")
Set mNamespace = oOutlook.GetNameSpace("MAPI")
Set oCalFolder = mNamespace.GetDefaultFolder(olCalendarFolder)
Set oApptItem = oCalFolder.Items.Add
With oApptItem
  .Start = "<<START_DATE_TIME>>"
  .End = "<<END_DATE_TIME>>"
  .Subject = "<<SUBJECT>>"
  .Body = "<<BODY>>"
  .ReminderMinutesBeforeStart = <<REMIND_MINS>>
End With
```

Of course, there are other properties that can be set for an appointment item, but you get the picture.

Now put that into a custom property (I use "uVBScript" below), and then when you need it, retrieve it, replace the placeholders with real info, and then run it like this:

```
on mouseUp
  put "4/12/05 9:00AM" into tStart
  put "4/12/05 10:00AM" into tEnd
  put "Doctor's Appointment" into tSubject
  put "Call 555-5555 beforehand to confirm." into tBody
  put 30 into tRemindMins
  SetAppointment tStart,tEnd,tSubject,tBody,tRemindMins
end mouseUp

on SetAppointment pStart,pEnd,pSubject,pBody,pRemindMins
  put the uVBScript of this stack into tScript
  replace "<<START_DATE_TIME>>" with pStart in tScript
  replace "<<END_DATE_TIME>>" with pEnd in tScript
  replace "<<SUBJECT>>" with pSubject in tScript
  replace "<<BODY>>" with pBody in tScript
  replace "<<REMIND_MINS>>" with pRemindMins in tScript
  runScript tScript
end SetAppointment

on runScript pVBS
  set the hideConsoleWindows to true
  put "C:\temp.vbs" into tTempPath
  put pVBS into url ("file:" & tTempPath)
  get shell("cscript.exe //nologo" && tTempPath)
  send "delete file" && quote & tTempPath & quote to me in 1 second
  -- this gives enough time for the script to run before you delete it
end runScript
```

### **Manipulating Outlook's Mail System**

Now, back to your original request about creating a mail message in Outlook with an attachment. Here's the basic VBScript again, without placeholders this time (so I can show you what the deal is) - it's similar to the calendar stuff above, though:

```
Const olInbox = 6
Const olCC = 2
Const olBCC = 3
```

```
Set oOutlook = WScript.CreateObject("Outlook.Application")
Set mNamespace = oOutlook.GetNameSpace("MAPI")
Set oMailFolder = mNamespace.GetDefaultFolder(olInbox)
Set oMailItem = oMailFolder.Items.Add
With oMailItem
    .Recipients.Add("kray@sonsofthunder.com")
    .Recipients.Add("jrosenberg@gilsonco.com").Type = olCC
    .Recipients.Add("hiddenguy@uknowwho.com").Type = olBCC
    .Subject = "Here's my attached file"
    .Body = "This is line 1" & vbCrLf & "This is line 2"
    .Attachments.Add("C:\myfile.txt")
    .Display
End With
```

A few notes:

1. For a multiline "body", you'll need to replace CRs in your variable you want to use with the "vbCrLf", and make sure things are quoted, so using the <<BODY>> variable, I might set it up like:

```
.Body = <<BODY>>
```

(note no quotes around the placeholder, unlike what I did for the Appointment Item), and then use a function like:

```
function MakeBody pMultiLineVal
    replace CR with (quote & " & vbCrLf & " & quote) in pMultiLineVal
    return quote & pMultiLineVal & quote
end MakeBody
```

to format it properly.

2. If you want to show it in Outlook and not send it, use the ".Display" command just before the "End With". If you want to send it, use ".Send".
3. Outlook security will bring up a dialog for the end user asking them for permission to allow an outside entity to do something with their Outlook program (I have Outlook 2002, so it may be a bit different in 2003). Either way, the user has to acknowledge the intrusion and allow it to happen.

In any event, I hope the above gives you some insight on how to use VB Script to automate Outlook.

*Posted 4/12/2005 by Ken Ray to the Use-Revolution List*

---

 [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](mailto:webmaster@sonsofthunder.com).

---