

Script not Functioning Properly

Source:

http://www.tech-archive.net/Archive/Outlook/microsoft.public.outlook.program_vba/2004-09/0406.html

From: Vincenzo Demasi (*vincenzodemasi_at_hotmail.com*)

Date: 09/16/04

Date: 16 Sep 2004 13:23:17 -0700

Hi,

I was wondering if someone could help me with a problem I've encountered using a script I downloaded from CDOLive.

Basically, the script is supposed to automatically reply to any new messages in the inbox, much the same way that an Out Of Office reply would work. I previously had the script modified to create a task based on the email, and to do a couple of other things as well, so I don't think using the Out of Office Reply or any of the included rules with Outlook 2000 is possible.

>From what I can gather, everything necessary is installed, and the script functions up to a certain point, but then doesn't create the new message or send it out. We're currently using Outlook 2000 and Exchange Server 5.5.

It used to function absolutely flawlessly, so I'm not sure what may have caused the problem, I was away from the organization when it stopped functioning, and nobody remembers exactly when it stopped working.

Any help would be greatly appreciated!

The log:

```
09/15/04 09:46:13 @HelpDesk AutoReply - Processing startet
New message with subject: <Test.> arrived
Message is not a status message, create reply
AutoReply - Processing finished
@HelpDesk AutoReply - Processing startet
New message with subject: <Test.> arrived
Message is not a status message, create reply
AutoReply - Processing finished
```

The script in question:

```
<SCRIPT RunAt=Server Language=VBScript>
```

```
'THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT  
'WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED,  
'INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES  
'OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR  
'PURPOSE
```

```
'-----  
'  
' NAME: AutoReply  
'  
' FILE DESCRIPTION: Automatically replies to all incoming messages  
with a  
' predefined text. Messages that contain status  
information  
' (e.g. delivery reports) are detected and omitted.  
'  
' Copyright (c) CdoLive 1999. All rights reserved.  
' http://www.cdolive.com  
' Mailto:samples@cdolive.com  
'  
' Portions:  
' Copyright (c) Microsoft Corporation 1993-1997. All rights reserved.  
'  
'-----
```

Option Explicit

```
'-----  
' Global Variables  
'-----
```

```
Dim g_bstrDebug ' Debug String
```

```
'-----  
' CONSTANTS  
'-----
```

```
' MAPI property tags used in this script  
Const CdoPR_ACTION = &H10800003  
Const CdoPR_ACTION_FLAG = &H10810003  
Const CdoPR_ACTION_DATE = &H10820040  
Const CdoPR_AUTO_FORWARDED = &H0005000B  
Const CdoPR_SENT_REPRESENTING_ADDRTYPE = &H0064001E
```

microsoft.public.outlook.program_vba: Script not Functioning Properly

```
Const ACTION_REPLY = 261
Const ACTION_FORWARD = 262
Const ACTION_REPLY_SENDER = 102
Const ACTION_REPLY_ALL = 103
Const ACTION_FORWARD_FORWARD = 104

' Reply text file
Dim g_Const_ReplyText

'Auto reply message text
g_Const_ReplyText = "Thank you for your e-mail!" & Chr(13) & Chr(13)
&_
    "We will get in touch with you immediately."

'-----
' EVENT HANDLERS
'-----

' DESCRIPTION: This event is fired when a new message is added to the
folder
Public Sub Folder_OnMessageCreated

    ' Declare variables
    Dim objSession ' Session
    Dim objFolder ' Outbox folder
    Dim objCurrentMsg ' Current message
    Dim objReplyMsg ' Reply message
    Dim objStatusMsg ' Status message
    Dim objAttachment ' Attachment
    Dim objFields ' Message fields
    Dim objField ' Message field
    Dim objRecipients ' Recipients collection
    Dim objRecipient ' Recipients object
    Dim strRecipients ' Recipients list
    Dim strMessageBody ' Message body
    Dim blnStatusMsg ' True if message is a status message

    ' Initialize variables
    Set objSession = Nothing
    Set objFolder = Nothing
    Set objCurrentMsg = Nothing
    Set objReplyMsg = Nothing
    Set objStatusMsg = Nothing
    Set objAttachment = Nothing
    Set objFields = Nothing
    Set objField = Nothing
    Set objRecipients = Nothing
    Set objRecipient = Nothing
    blnStatusMsg = False
```

microsoft.public.outlook.program_vba: Script not Functioning Properly

```
' Clear error buffer
Err.Clear

' Get session information
On Error Resume Next
Set objSession = EventDetails.Session

' No errors detected ?
If Err.Number = 0 Then

    ' Write some logging
    Call DebugAppend(objSession.CurrentUser & " AutoReply – Processing
startet", False)

    ' Get outbox folder
    Err.Clear
    On Error Resume Next
    Set objFolder = objSession.Outbox

    ' No errors detected ?
    If Err.Number = 0 Then

        ' Get current message
        Err.Clear
        On Error Resume Next
        Set objCurrentMsg =
objSession.GetMessage(EventDetails.MessageID,Null)

        ' Error detected ?
        If Err.Number <> 0 Then

            ' Error reading current message
            Call DebugAppend("Error – Could not read message", True)
        Else

            ' Write some logging
            Call DebugAppend("New message with subject: <" &
objCurrentMsg.Subject & "> arrived", False)

            ' Check if message is a non-delivery report
            If objCurrentMsg.Type = "REPORT.IPM.NOTE.NDR" Then
                blnStatusMsg = True

            ' Check if message is a delivery report
            ElseIf objCurrentMsg.Type = "REPORT.IPM.Note.DR" Then
                blnStatusMsg = True

            ' Check if message is a read notification
            ElseIf objCurrentMsg.Type = "REPORT.IPM.Note.IPNRN" Then
                blnStatusMsg = True
```

microsoft.public.outlook.program_vba: Script not Functioning Properly

```
' Check if message is a not-read notification
ElseIf objCurrentMsg.Type = "REPORT.IPM.Note.IPNNRN" Then
    blnStatusMsg = True

' Check if message is an out of office reply
ElseIf objCurrentMsg.Type = "IPM.Note.Rules.OofTemplate.Microsoft"

Then

    blnStatusMsg = True

' Check if message is a meeting item
ElseIf Left(objCurrentMsg.Type, 12) = "IPM.Schedule" Then
    blnStatusMsg = True

' Check for some special cases
Else

    ' Get fields collection of current message
    On Error Resume Next
    Set objFields = objCurrentMsg.Fields

    ' Check if we've got a fields collection
    If Not objFields Is Nothing Then

        ' Get auto-forwarded status field
        On Error Resume Next
        Set objField = objFields.Item(CdoPR_AUTO_FORWARDED)

        ' Check if field found
        If Not objField Is Nothing Then

            ' Check if message is auto-forwarded
            If objField.Value = True Then
                blnStatusMsg = True
            End If
        End If

        ' Get sender address type field
        On Error Resume Next
        Set objField = objFields.Item(CdoPR_SENT_REPRESENTING_ADDRTYPE)

        ' Check if message is from an external address
        If objField.Value <> "EX" Then

            ' Get a reference to the first attachment
            Err.Clear
            On Error Resume Next
            Set objAttachment = objCurrentMsg.Attachments.Items(1)

            ' No errors detected ?
            If Err.Number = 0 Then
```

microsoft.public.outlook.program_vba: Script not Functioning Properly

```
' Assign the source property of the attachment to a
' previously defined message object
On Error Resume Next
Set objStatusMsg = objAttachment.Source

' Check if status message found
If Not objStatusMsg Is Nothing Then

    ' Check if message is a non-delivery report
    If objStatusMsg.Type = "REPORT.IPM.NOTE.NDR" Then
        blnStatusMsg = True

    ' Check if message is a delivery report
    ElseIf objStatusMsg.Type = "REPORT.IPM.Note.DR" Then
        blnStatusMsg = True

    ' Check if message is a read notification
    ElseIf objStatusMsg.Type = "REPORT.IPM.Note.IPNRN" Then
        blnStatusMsg = True

    ' Check if message is a not-read notification
    ElseIf objStatusMsg.Type = "REPORT.IPM.Note.IPNNRN"

Then
        blnStatusMsg = True

    ' Check if message is an out of office reply
    ElseIf objStatusMsg.Type =
"IPM.Note.Rules.OofTemplate.Microsoft" Then
        blnStatusMsg = True
    End If
End If
End If
End If
End If
End If
End If

' Check if message does not contain status information
If blnStatusMsg = False Then

    ' No status information found, write logging
    Call DebugAppend("Message is not a status message, create reply",

False)

    ' Reply message using Message.Reply()
    On Error Resume Next
    Set objReplyMsg = objCurrentMsg.Reply()

    ' Check if we've got a copy of the message
    If Not objReplyMsg Is Nothing Then
```

microsoft.public.outlook.program_vba: Script not Functioning Properly

```
' Check if current message subject does not contain
' reply prefix
If Left(UCase(objCurrentMsg.Subject), 3) <> "RE:" Then

    ' Set reply subject with reply prefix
    objReplyMsg.Subject = "RE: " & objCurrentMsg.Subject
Else

    ' Set reply subject without reply prefix
    objReplyMsg.Subject = objCurrentMsg.Subject
End If

' Get recipients list of current message
Err.Clear
On Error Resume Next
Set objRecipients = objCurrentMsg.Recipients

' No errors detected ?
If Err.Number = 0 Then

    ' Loop through recipients collection and add recipient names
    For Each objRecipient In objRecipients
        If strRecipients <> "" Then
            strRecipients = strRecipients & "; " & objRecipient.Name
        Else
            strRecipients = objRecipient.Name
        End If
    Next
Else

    ' Set current user as only recipient
    strRecipients = objSession.CurrentUser
End If

' Constuct message body
strMessageBody = Chr(13) & Chr(13) & Chr(13) & "-----Original
Message-----" & Chr(13) _
    & "From: " & objCurrentMsg.Sender & Chr(13) & "Sent: " &
objCurrentMsg.TimeReceived & Chr(13) _
    & "To: " & strRecipients & Chr(13) & "Subject: " &
objCurrentMsg.Subject & Chr(13) & Chr(13)

' Set message body
objReplyMsg.Text = g_Const_ReplyText & strMessageBody &
objCurrentMsg.Text

' Update and send message
Err.Clear
On Error Resume Next
objReplyMsg.Update
objReplyMsg.Send
```

microsoft.public.outlook.program_vba: Script not Functioning Properly

```
' Errors detected ?
If Err.Number <> 0 then

    ' Could not sent reply message, write logging
    Call DebugAppend("Error – Could not send reply message", True)
Else

    ' Reply message successfully sent
    Call DebugAppend("Success – Reply message send successfully",

False)

    ' Get fields collection of current message
    On Error Resume Next
    Set objFields = objCurrentMsg.Fields

    ' Check if we've got a fields collection
    If Not objFields Is Nothing Then

        ' Set the reply flags of the current message
        On Error Resume Next
        objFields.Add CdoPR_ACTION_DATE, Now
        On Error Resume Next
        objFields.Add CdoPR_ACTION, ACTION_REPLY
        On Error Resume Next
        objFields.Add CdoPR_ACTION_FLAG,

ACTION_REPLY_SENDER

        ' Update current message
        On Error Resume Next
        objCurrentMsg.Update True, True

        ' Mark current message as read
        objCurrentMsg.Unread = False
    End If
End If
Else

    ' Status information found, write logging
    Call DebugAppend("Message is a status message, no reply sent",

False)

    End If
End If
Else
```