

## Re: Scheduled script problem

---

*Source:*

<http://www.tech-archive.net/Archive/Windows/microsoft.public.windows.server.scripting/2007-11/msg00080.html>

---

- *From:* Tom Lavedas <tglbatch@xxxxxxx>
  - *Date:* Fri, 16 Nov 2007 06:49:56 -0800 (PST)
- 

On Nov 16, 9:45 am, Tom Lavedas <tglba...@xxxxxxx> wrote:

On Nov 15, 11:15 pm, "dk" <kaa...@xxxxxxxxxxxx> wrote:

Here is the problem:

A vbs script that sets up a command to run a batch file, calls EXEC on the command. When run from a scheduled task, it hangs on the EXEC command, until the scheduler stops the script after the maximum allowed time to run. If the EXEC command is commented out, the script does some other work and completes.

The scheduled task is set to run with user permissions set explicitly for this type of task. When the same user logs in directly and executes the script from the command line, the batch file and vbs script completes without problem. User has rights to run and read write in the directory where the script resides. The user has rights to run the batch file in the directory where the batch file resides. The user has rights to create files at root of C: where a process the batch file kicks off wants to write. Again, works successfully when run from the command line: cscript (or wscript) EXAMPLENAME.VBS

Any ideas what could be wrong?

Example of what the script is doing:

```
Dim oShell
dim logFile
set outerFSO = CreateObject("Scripting.FileSystemObject")
set logFile = outerFSO.OpenTextFile("c:\tmp\runreport.log", 2, true)
logFile.WriteLine("Log file created")
Set oShell = CreateObject("WScript.Shell")
strPath = Chr(34) & "C:\Program Files\product name\reports\runreport.bat"
```

Re: Scheduled script problem

```
&  
Chr(34) & " " & Chr(34) & CreateYesterdayName & Chr(34) & " >  
c:\tmp\temp.txt"  
logFile.WriteLine(strPath)  
oShell.exec strPath  
logFile.WriteLine("Exiting script")
```

CreateYesterdayName is a function that creates the desired file name.

Thank you,  
D.

Did you specify that the script is to run Interactive? The Exec command needs a command console, which is not possible if the operation does not run interactive. In addition, as I understand it, unless a user is logged in to the workstation such interactive operations still will fail.

The correct approach, I think, is to replace the Exec with a Run with it's operation 'hidden', something like this ...

```
oShell.Run strPath, 0, true
```

If you want to catch the return code for your log function, try this ...

```
Dim nRes  
nRes = oShell.Run strPath, 0, true  
logFile.WriteLine "Exiting script with exit code: " & nRes
```

Tom Lavedas  
===== <http://members.cox.net/tglbatch/wsh/>

Opps, there's an error in that nest to the last line of code. It should read ...

```
nRes = oShell.Run(strPath, 0, true)
```

Tom Lavedas  
===== <http://members.cox.net/tglbatch/wsh/>

.