

Writing a Script Accountable for its Actions

Include a function to log and optionally display script actions

by Bob Kelly
April 2007



What the heck did that script do!? No matter if you are being asked by a victim of your script or you are asking the question yourself, there is nothing handier than a script written to log what it does. I often use the same function to post status during execution to also log what it is doing. Some times you'll want to log something and not display it as status to the user. To handle this I pass not only what should be logged to the function, but also if it should be displayed so it can be handled on a case by case basis by the same function.

Following best practices, we will use a variable to identify our log file so it may be easily changed in the future or when using this code in another script:

```
sLogFile = "c:\temp\mylog.txt"
```

In case you later decide you want the ability to force not to display anything regardless of how you called the function, we will set a global variable to override the setting.

```
ForceQuiet = False
```

Because the File System Object is something you will typically make use of in a script outside this simple function call, we will place the creation of the object in the body of the script so it is available to any and all other function in your script (as well as the one we are writing now).

```
Set oFSO = CreateObject("Scripting.FileSystemObject")
```

And now for the actual function...

```
Function LogAction (sMessage, IsVisible)
```

```
    Set oLogFile = oFSO.OpenTextFile(sLogFile, 8, True)
    oLogFile.WriteLine Date() & " " & Time() " | " & sMessage
    If IsVisible And Not ForceQuiet Then
        WScript.Echo sMessage
    End If

    oLogFile.Close
```

```
End Function
```

The function uses the File System Object to open a text file (and creates it if it does not exist) and then writes in the date and time followed by the message that is passed. If IsVisible is true (also passed to the function) and if the ForceQuiet variable is not true, the message written to the log is also echoed to the screen. The function should be called like this:

```
LogAction "Starting Script", False  
LogAction "Mapping Drives", True
```

Just call this function whenever you perform an action in your script and if things should go wrong, you'll see just where the problem is with ease! [M](#)

Bob Kelly is president and co-founder of AdminScriptEditor.com, home to an integrated suite of scripting tools and a shared library of scripts and language help. He has authored books on scripting and desktop administration as well as several white papers. Bob also owns and operates AppDeploy.com, where he writes and produces videos on topics related to software deployment. You can contact Bob about "Writing a Script Accountable for its Actions" at bkelly@adminscrepteditor.com.