

# Leaving Footprints

*using footprints to limit how often code is executed*

by Bob Kelly

November 2006



No, I'm not talking about how much space a script takes, or how much memory it may consume. In this context, a footprint is something a script leaves behind to show that it has been executed. By using this footprint as a condition of your script's execution, you can ensure that your code runs only one time (or however often you wish it to run).

Those familiar with the assignment of applications using Group Policy may be compelled to deploy scripts as Windows Installer packages. However, this adds unnecessary overhead and complicates the target system by making it manage installations that are merely scripts. Group Policy is an excellent means of deploying scripts in a Windows environment, but you should limit yourself to the Startup, Logon, Logoff and Shutdown scripts (keeping in mind that logon and logoff run in the security context of the logged on user). But you don't want it to run every time you say? Group Policy does its own check each time to see if something has already been installed, executing only when necessary- but you can do the same in your scripts using footprints.

Particularly in something like a logon script, there may be actions you want to take only the first time a user logs on. In this case, dropping a registry entry or file in the user profile provides an effective way to make sure your script does not run again and again:

## On Error Resume Next

```
FootPrints = "HKEY_CURRENT_USER\Software\Scripts\  
Set WSHShell = CreateObject ("WScript.Shell")  
sRunCheck = WSHShell.RegRead(FootPrints & WScript.ScriptName)
```

```
If sRunCheck <> Then  
    WScript.Echo "Perform Action"  
    WSHShell.RegWrite FootPrints & WScript.ScriptName, Date(), "REG_SZ"  
Else  
    WScript.Echo "No Action Taken"  
End If
```

With just a slight change, we can modify this code to ensure code run by the logon script is limited to execution only once per week, per user...

## On Error Resume Next

```
FootPrints = "HKEY_CURRENT_USER\Software\Scripts\  
Set WSHShell = CreateObject ("WScript.Shell")  
dRunDate = WSHShell.RegRead(FootPrints & WScript.ScriptName)
```

```
If DateDiff("d", dRunDate, Date()) > 7 Then  
    WScript.Echo "Perform Action"  
    WSHShell.RegWrite FootPrints & WScript.ScriptName, Date(), "REG_SZ"  
Else  
    WScript.Echo "No Action Taken"  
End If
```

By using the date the script was executed as the footprint, we can do date calculations to determine if we want to execute or skip certain actions. Depending upon where you place your footprint, you can have the condition apply to users (place the footprint in the profile) or to the system (place the footprint outside the user profile). Thinking about your own environment, I'm sure you can come up with smart ways to make use of this technique that will work for you. By running actions in a more controlled manner instead of at every logon or startup will speed user logon times and make you the hero of the day! [M](#)

*Bob Kelly is president and co-founder of [AdminScriptEditor.com](http://AdminScriptEditor.com), home to an integrated suite of powerful scripting tools as well as a shared library of scripts and language help. He has authored books on Scripting and Desktop Administration as well as several white papers. Bob is also is the owner and operator of [AppDeploy.com](http://AppDeploy.com) where he writes and produces videos on topics related to software deployment. You can contact Bob about "Leaving Footprints" at [bkelly@adminsripteditor.com](mailto:bkelly@adminsripteditor.com).*