

## Re: Checking for running process

**Source:** <http://www.tech-archive.net/Archive/Scripting/microsoft.public.scripting.wsh/2004-03/0059.html>

---

**From:** Joe Earnest (joeearnestNO\_at\_SPAMqwest.netPLEASE)

**Date:** 03/03/04

Date: Wed, 3 Mar 2004 10:30:18 -0700

Hi,

"Joel" <anonymous@discussions.microsoft.com> wrote in message news:FAFB314B-F8DA-4C10-B366-D927EAA13BEA@microsoft.com...

> *Hi,*

>

> *I've created a simple WSH script for the Win2000 computers in my kid's classroom that attempts to solve the problem of the kids launching multiple instances of slow-loading apps. If the app doesn't respond quickly the kids get impatient and click multiple times and launch the app a bunch of times. The record so far is 23 instances.*

>

> *The script I created accepts the app's path and executable name as a command line argument and then makes use of WMI to enumerate the running processes on the computer to check to see if the app is already running:*

>

> *var e = new*

*Enumerator(GetObject("winmgmts:").InstancesOf("Win32\_process"))*

> *for (;!e.atEnd();e.moveNext())*

> *{*

> *var Process = e.item();*

>

> *if (AppNameNoPath.toUpperCase() ==*  
*Process.Name.toUpperCase())*

> *{*

> *//*

> *// We have a name match which means the application is already*

> *// running. Activate the already running instance and then quit.*

> *//*

>

> *WshShell.AppActivate(*

*Process.processid );*

> *WScript.Quit();*

>

> *If the app isn't already running the script launches it via*  
*WshShell.Run().*

>

> *This all works great except with some old shareware DOS-based programs*

microsoft.public.scripting.wsh: Re: Checking for running process

they have. The DOS-based apps run under NTVDM and as a result my name comparison never sees a match. I've looked at the command line arguments that the NTVDM process was launched with but there's nothing useful there.

>

> *Does anyone have any ideas as to how I can detect that an instance of a specific app is running under NTVDM?*

>

> *I really don't want to have to resort to doing something like creating/checking for a temp file to specify that a given app has been launched.*

>

> *Thanks in advance for any ideas.*

If you were running WinXp, you could check the CommandLine property of the process in WMI. But this is WinXp only.

Otherwise, a *\*possibility\** would be to modify the trick for a script to get its own PID (since all scripts show up as WSCRIPT.EXE). Run the program through WshShell.Exec, instead of WshShell.Run, if you can. (I've never tried this with a DOS program – you may have to run it as a parameter of cmd.exe.) The Exec method returns a ProcessId property, although it's not listed in the MS WSH CHM documentation. The PID will need to be listed, using a temporary text file, a junk registry key or an IePipe (an invisible IE window opened just to allow scripts to store and pass data back and forth — sort of an elaborate memory sink). A script can check the list to see if the program has an entry for the program with an associated PID. If so, it can check the PID to see if it's name is "NTVDM" and is active. If so, it shuts down. If not, then run the program through WshShell.Exec and add/replace the PID. To remove all chance of reassigned PID conflicts, you should recreate an empty temp file or cleanse the registry key at shutdown or bootup. You could also leave the initiating script active while the program is active, looping (with something like WScript.Sleep 1000 to preserve resources) to check the WshShell.Exec's Status flag, and have it cleanse the list at app shutdown. If using this last method, you could simply create a specifically name temp file for each application, and not have to deal with a list. (If the file exists, the app is running.) The initiating script can then just delete the temporary file when the app is closed.

I'd throw some code at you, but I script entirely in VBS and you seem to use JS. There shouldn't be anything difficult in the coding, however.

Joe Earnest

---

Outgoing mail is certified Virus Free.

Checked by AVG anti-virus system (<http://www.grisoft.com>).

Version: 6.0.564 / Virus Database: 356 - Release Date: 01-19-04