Scripting a Scheduled Wakeup in Windows 7 (and Vista too!)

By Boyd, October 15, 2010 1:27 pm

When it comes to implementing power saving settings on managed workstations, the easy part is configuring the power management settings themselves. The hard part is ensuring that the systems remain consistently managed and maintained. Once standby settings are configured in Windows Power Management, idle workstations are likely to enter standby overnight, which is great way for conserving energy. But the evening hours are also an ideal time to deploy software and updates, because it's less disruptive to employees that use these workstations throughout the day. How do you balance power savings, maintenance, and the end user experience on these systems?

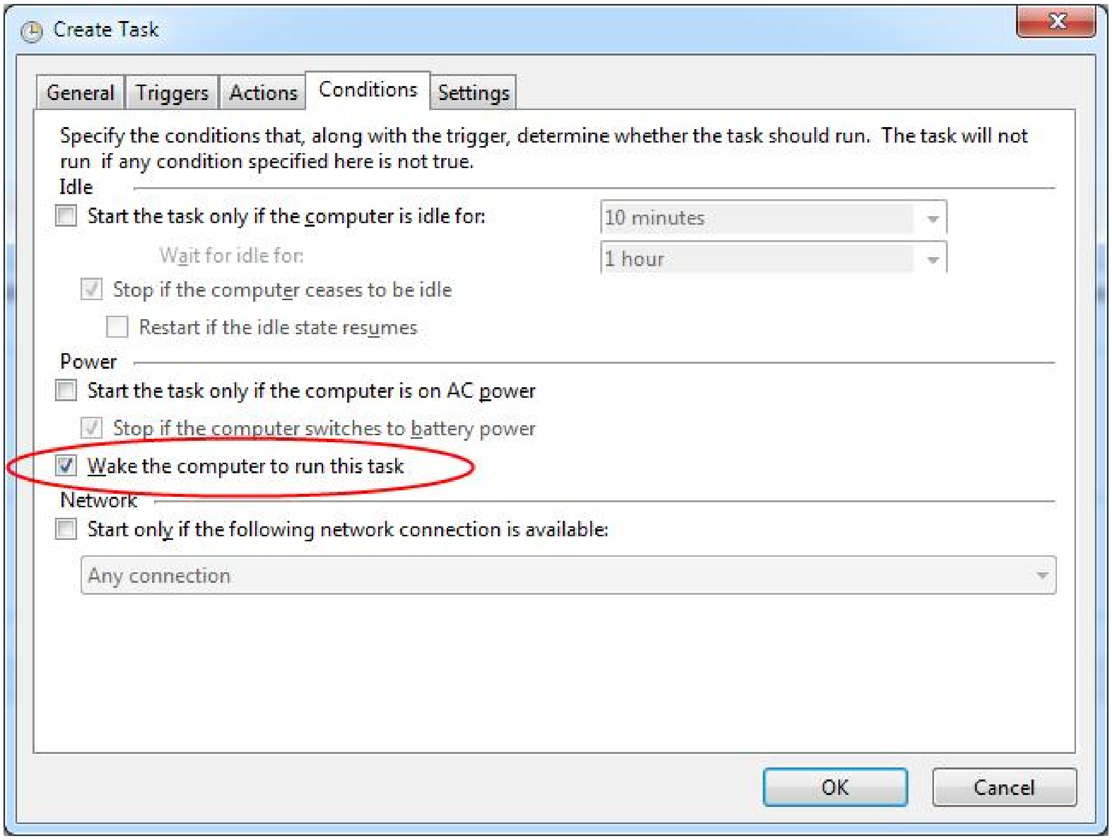
Well, some people will tell you that Wake-on-LAN (WoL) is the solution. It's true that, with WoL, you should be able to wake machines overnight to perform tasks — in theory at least. I say *in theory* because, any sysadmin that has tried to use WoL to wake and manage many workstations (100+) over multiple subnets will tell you that Wake-on-LAN is no magic bullet. There are several reasons for this:

1. To wake up a workstation with Wake-on-LAN, the workstation's network adapter must be properly configured to receive WoL's *Magic Packets* (trust me, these packets are much less magical than their name implies). This can be a lot more difficult than it sounds, especially if you need to script these settings for automated configuration.
2. In most environments, WoL packets will not work across subnets, so you need at least one device on each subnet that can send WoL packets. You'll also need some sort of mechanism (usually software) to tell sender devices to send packets on their subnet to wake them up.
3. Many wireless network adapters do not support WoL, and the ones that do tend to have inconsistent results with receiving WoL packets. If you have workstations that only connect to the network via wireless, this is a problem.
4. If the workstation is disconnected from the LAN, the WoL packet won't make it.

Scheduled Wakeups

WoL is very useful for many situations, especially for impromptu wakeups. But if you want your workstations to wake from standby at night, or any time, you may not want to depend soley on WoL. What you really need is something that tells Windows to resume from standby on a schedule. But how? Well, Microsoft Windows actually includes the capability to resume from standby at certain times. This functionality is a part of the Task

Scheduler service, and it can be enabled by simply clicking a check box:



Specify the conditions that, along with the trigger, determine whether the task should run. The task will not run if any condition specified here is not true.

Idle

□ Start the task only if the computer is idle for: 10 minutes

Wait for idle for: 1 hour

☑ Stop if the computer ceases to be idle

□ Restart if the idle state resumes

Power

□ Start the task only if the computer is on AC power

☑ Stop if the computer switches to battery power

☑ Wake the computer to run this task

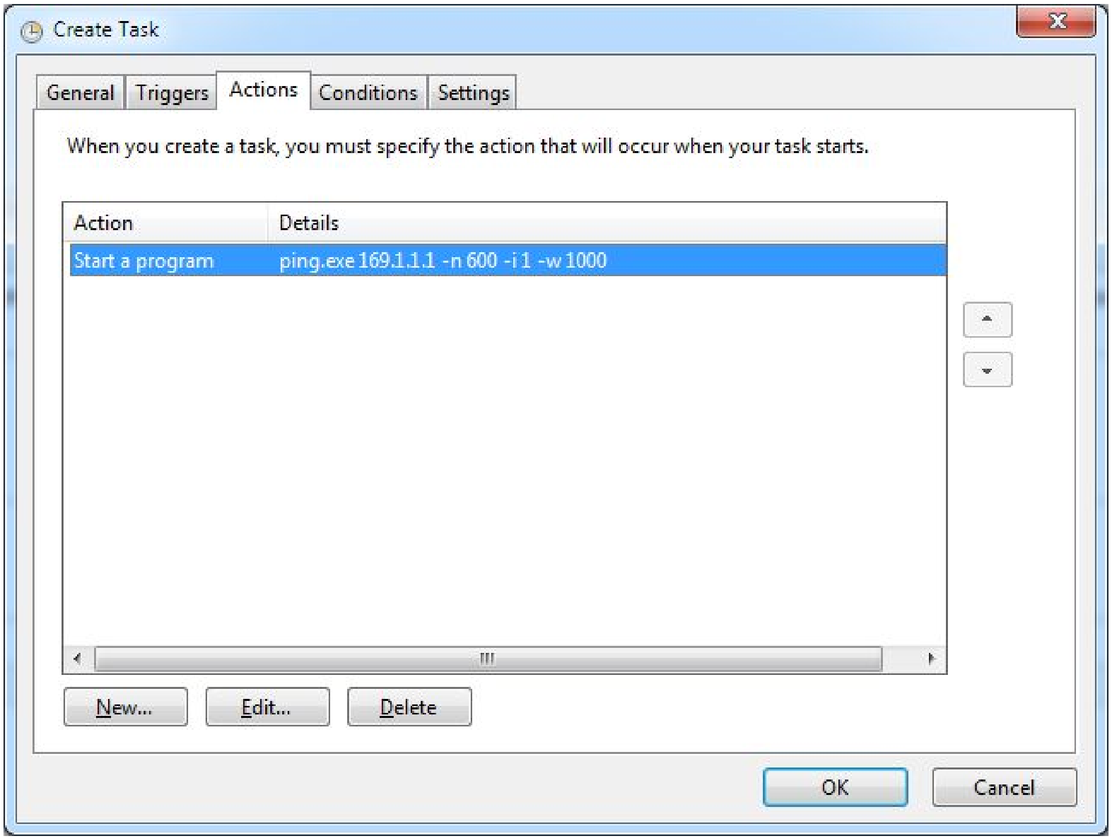
Network

□ Start only if the following network connection is available:

Any connection

With the "Wake the computer to run this task" checkbox set on a scheduled task, the system will resume from standby at whatever time interval has been configured on the Triggers tab. It is important to note, however, that this won't work if the system is completely powered off.

The next question is, what should the task do once it has woken up the system? The answer is, just about anything. For example, it could run a script that starts Windows Update, run a virus scan, or start a backup. If you can script it or call it from the command line, you can do it. Here's a simple example of how you might keep the system awake for at ~10 minutes by using the ping command:



When you create a task, you must specify the action that will occur when your task starts.

Action	Details
Start a program	ping.exe 169.1.1.1 -n 600 -i 1 -w 1000

Scripting Wakeups

Alright, we can use a scheduled task to wake workstations. That's great, but not very useful unless we can use a script to automate the creation of a task that does this. As you may already know, a scheduled task can be created with the command line utility: SCHTASKS.EXE. This is a relatively powerful utility for creating tasks, and once you understand all of the command line options, creating a task with this utility is fairly straightforward:

```
schtasks /create /TN "My Wakeup Task" /SC DAILY /ST 23:00 /TR "ping.exe 16
```

Unfortunately, it appears as though there's no way to set "Wake the computer to run this task" via SCHTASKS. However, Windows Vista and 7 come with a robust Task Scheduler API that can configure this setting. I wrote a VB script that does just that:

```
 the computer to run this task" setting on Windows Vista and Windows 7 sy
uccessfully, both in the GUI or via this script,the task compatibility
he setting gets reverted.
eToRun.vbs "" [enable | disable]
akeToRun.vbs "My Scheduled Task" enable
he computer to run this task" to enabled.
```

```
:e, objRootFolder, objTask, objDefinition


:s!"





ıbleWakeToRun to true by default if enable/disable was not specified.

ıuter to run this task'"
:NABLE" Then
ıuter to run this task'"


ıputer to run this task'"



 Service")

·("\")
'askName)


ı CStr(objDefinition.Settings.WakeToRun)
  " & objDefinition.Settings.Compatibility

:eToRun

:.Name, objDefinition, TASK_UPDATE or  TASK_DONT_ADD_PRINCIPAL_ACE, , , ob

 Service")

·("\")
ıe)
:r(objTask.Definition.Settings.WakeToRun)
ı objDefinition.Settings.Compatibility
```

To use this script, create a task first by using SCHTASKS. Then, run a command similar to this:

```
cscript.exe ScheduledTaskSetWakeToRun.vbs "My Scheduled Task" enable
```

The script will output both the previous and new values of the "WakeToRun" setting. You can verify that it worked by opening the Task Scheduler GUI and verifying that "Wake the computer to run this task" is set. This script can also be used to disable this setting.

You may also notice that the code in this script sets the "task compatibility mode" version to 2. I found issues with tasks that were using a different compatibility mode. It seems that "Wake the computer to run this task" would always be reverted, even if it was set through the Task Scheduler GUI. I believe the only disadvantage to changing the compatibility mode is that the task will not be backwards compatible with Windows XP.

By using a combination of Wake-on-LAN and scheduled wakeups, it’s much easier to successfully manage and maintain workstations in standby with better precision and accuracy. Also, by performing maintenance tasks overnight, you can keep workstations reliable without impacting the end user. It’s a win-win situation!

More »

Power Management, Scripting, System Administration, Windows 7 | power management, scheduled wakeup, standby, task scheduler, Wake-on-LAN

10 Responses to “Scripting a Scheduled Wakeup in Windows 7 (and Vista too!)”

1. *Future001* says:
   October 27, 2010 at 10:08 am

   Hi nice script, because I searched the hole wwww. But I got an error on line 53,83 if I tried to use this script. I think that’s these strings “” on the end of the script. Please – can you fix this!

   greetings

   Reply

   - *Boyd* says:
     October 27, 2010 at 12:03 pm

     I think some something was wrong with the HTML there. I updated the article. You may want to give the script another try. If you are still getting an error, please post the full message. Thanks.

     Reply

2. *Future001* says:
   October 27, 2010 at 2:24 pm

   Thanks a lot for changing the code. But if I use it, then the script stops at line 47:
   ScheduledTaskSetWakeToRun.vbs(47, 1) (null): (47,4):Task:
   Any ideas?
   I’m using Win7 64bit (german)

   Greetings

   Reply

   - *Boyd* says:
     October 28, 2010 at 11:44 am

Can you confirm that you're running the script in an elevated command prompt?

This is the error I get (on the same line) when I run it in a regular command prompt (UAC enabled):
../script.vbs(47, 1) Microsoft VBScript runtime error: Permission denied

Reply

- *Future001* says:
  October 28, 2010 at 1:58 pm

  I have an Batch with some parameters which looks like that:
  schtasks.exe /CREATE /TN NewTask3 /TR "ping 127.0.0.1″ /SC EINMAL /ST 13:30:45 /SD 27/10/2011 /RU user /RP password
  cscript.exe ScheduledTaskSetWakeToRun.vbs "NewTask3″ enable
  The task will be created but the vbs will be aboted. I tried this step by step in an prompt with and without Administrator rights, with and without UAC enabled. But everytime the same error. I tried to disable the 'wake up' from an manually created task (compat. set to Server 2008 R2); same error on line 47:

  ————————————

  Task Name: NewTask3
  Action: DISABLE 'Wake the computer to run this task'
  Current WakeToRun Setting: Wahr
  Current Compatibility Setting: 3
  —

  Path\..\Run.vbs(47, 1) (null): (47,4):Task:

- Boyd says:
  October 28, 2010 at 4:55 pm

  I see the problem now. Your task is being run as a user, and I incorrectly assumed that it would be run under the service account SYSTEM. I updated the script in this article and changed this line of code:
  objRootFolder.RegisterTask objTask.Name, objDefinition.XmlText, TASK_UPDATE or TASK_DONT_ADD_PRINCIPAL_ACE, , , &H5
  to this:
  objRootFolder.RegisterTaskDefinition objTask.Name, objDefinition, TASK_UPDATE or TASK_DONT_ADD_PRINCIPAL_ACE, , , objDefinition.Principal.LogonType

  I also updated the SCHTASKS command to run as SYSTEM, as I think this is better than running it as the logged on user. However, there are cases where it may be required (not everything can be run as SYSTEM).

  Thanks!

- *Future001* says:
  October 29, 2010 at 4:39 pm

  Got it!
  Your script shown no errors, but now it tells me, that I have no permission (as you said on top). I think, the script has no permission to change the task from the 'user'. My little workaround seems to correct this, because I need to run it in the user account: 1st I create an schtasks with /NP, 2nd I run your script and lastly I /Change it to my /RU /RP

  One more thing: can you implement to set the flag, that 'the task will be deleted immediatly if there is no other schedule' (sorry – don't know the real english flag name)

  Regards

- *Boyd* says:
  October 29, 2010 at 5:51 pm

  I'm glad you managed to get it working. I like your workaround. Regarding the 'delete after final run' flag, have you tried using the /Z switch with Schtasks? I found it in this MSDN article: http://msdn.microsoft.com/en-us/library/bb736357%28VS.85%29.aspx . If that doesn't work, I could probably add it to the script. But I don't really want to if Schtasks can do it already.

  Thanks

- *Future001* says:
  October 29, 2010 at 6:49 pm

  Nope, you can't, because with the /Z Option and v2 or v3 compatibility you got an error (Link): http://www.vistax64.com/vista-general/139897-cannot-create-run-once-task-schtasks.html
  The workaround for that ist to enable V1 – wich is no option in our environment 🙁

  To use the /Z you have to add another trigger in you task with a duration or ending time …(?)… very complicated, in the GUI too.
  BUT with every batch I create 3 unique schtasks which will only be run once in their life. They should be deleted after the trigger event occours. If not, after 3weeks or so, may be here will 40-50 tasks in scheduler, all out of date!

  greetings

3. *Win 7 task scheduler - MediaPortal Forum* says:
   January 5, 2011 at 9:10 am

   [...] [...]

Reply

Leave a Reply

Name (required)

Mail (will not be published) (required)

Web site

Add your Reply

← ATI Radeon Causes a UAC Prompt at User Logon
Updated Dell Warranty Information Script →

- **Recent Posts**
  - Testing for Blocked UDP Ports
  - Applying KB2506143 to an offline Windows 7 SP1 WIM = Windows Setup fail
  - Recovering Data from a Failing Bitlocker Hard Drive
  - Updated Dell Warranty Information Script
  - Scripting a Scheduled Wakeup in Windows 7 (and Vista too!)
-

October 2010

M	T	W	T	F	S	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

« Jun Feb »

- **Vitals**
  - *Users* 1
  - *Posts* 28
  - *Comments* 278

  - *Pages* 1

- **Archives**

  Select Month ▾

- **Categories**

  - BitLocker
  - Data Recovery
  - ITS
  - OS Deployment
  - Power Management
  - Programming
  - Scripting
  - Security
  - System Administration
  - Troubleshooting
  - Uncategorized
  - Windows 7
  - Work
-

- **Disclaimer**

  The expressed opinions, informational content and links displayed on this website do not necessarily reflect a position or policy of The Pennsylvania State University or its affiliates.

(iBoyd) is powered by WordPress

Panorama Theme by Themocracy