

P2V Post-Configuration Scripting Options (W2K/3/XP)

Document Version 1.2

RTFM Education

Beyond the Manual... with Mike Laverick

By Mike Laverick
© RTFM Education

Contact:
mikelaverick@rtfm-ed.co.uk

Note:

- Firstly, I want to begin with a big disclaimer – this document shows a heavily scripted approach to the post-configuration of a P2V'd Virtual Machine. I accept ABSOLUTELY no responsibility for any unforeseen and unwanted side-effects. Use these scripts at your own risk!
- Due of the difficulty of cut and pasting code from rich-text documents like this – you can download sample versions of these scripts from my website here – I've put them into an ISO file because in the end I decided to run the scripts from within the P2V'd Virtual Machine – and I couldn't rely on network communications to able to run them across a network

<http://www.rtfm-ed.co.uk/downloads/p2v-scripts.iso>

- This section uses the utility called DevCon to interrogate the Virtual Machine and Physical Machine. Unfortunately, its difficult to have both machine on the same network at the same time because of IP and NETBIOS conflicts
- So to work around this I reconfigure my helper to have two network cards – one on a Internal Switch (vmnet) and one on a External Switch (vmnic). I communicate to the Physical Server using the External Switch, and put the P2V'd Virtual Machine on an Internal Switch. This allows the VM helper to have communications to both machines with IP/NETBIOS conflicts occurring
- Then on the P2V Assistant/Help I run the Microsoft Routing and Remote Access Wizard – and allow the system to make this a NAT. I used the External Network connection as my link to the "internet"
- This allows the Virtual Machine to be on its private network but still have network communication to outside network with IP/NETBIO conflicts – as it is "hidden" behind the NAT.
- All the scripts are run from the P2V'd Virtual Machine
- All my tests have been on Windows 2000 P2V's so far
- An extra layer complexity existing if you are downgrading or upgrading the CPU in your P2V – this because of the update required to the ACPI/HAL
- As I've shown elsewhere in my P2V documentation when the Virtual Machine first boots – just acknowledge any warnings and error messages – and don't bother with a reboot

Unattended Installation of VMware Tools

1. **Within a VM**, Choose **Install VMTools** and Click the **Install button** on the dialog box (**do not complete the installation**)

Note:

This should connect the VM's Virtual CD to an VMware Tools ISO file

2. **Within the VM**, browse to the **CD-ROM** using **Windows Explorer**
3. On the **ROOT** of the drive you **should see the contents of the CD...**
4. **Copy ALL of the CD to another location** (local or network the choice is yours)

Note:

Do not just take the MSI. It will not work isolation, and requires the other files.

5. **Lower the driver signing security level of the server** with:

Properties of My Computer, (System Properties Dialog)

Hardware Tab

Driver Signing Button

Choose **Ignore**

Note:

You can change this through the registry (HKLM\Software\Microsoft\Driver Signing). You could lower the security by using regini or a importing a REG file – if you doing this on Windows 2000 this is fine

Beware that in Windows XP/2003 Microsoft disabled this "Registry" method on security grounds, because too many vendors were using a registry hack to suppress the warning message.

Microsoft KB article [Q298503](#) outlines why this change was introduced to W2K3 and WinXP. If you are using W2K you can use a registry hack.

The only way to change the Driver Signing feature programmatically in W2K3 and WinXP is:

- a.) by hand
- b.) by a policy – see the KB article
- c.) by secedit inf file

I prefer to use policies to upgrades of production Virtual Machines
I use the secedit inf file method for P2V as the Virtual Machine may not be part of the domain or even have network communications

6. *This version, allows a reboot:*

```
msiexec -i "c:\vmtools\vmware tools.msi" addlocal=all /qn
```

This version suppresses the reboot:

```
msiexec -i "c:\vmtools\vmware tools.msi" addlocal=all /qn  
REBOOT="ReallySuppress"
```

Note:

Like don't just suppress – really suppress!

Script and install must run in under the context of Administrator account for this to be successful. -i switch does an installation; addlocal=all switch instructs the msiexec tool to install all the available features locally;/qn does a quiet install with no UI. For more information about using msiexec consult MS KB [Q314881](#)

Note:

If you installing VMware Tools to Windows running Terminal Services in a Application Mode – you will need to open and close the install mode and close it too

Here is a sample script which installs vmtools in unattended way across the network

```

@echo off
cls

echo =====
echo ==Temporarily Switching on Ignoring Driver Signing==
echo =====
echo.
echo Please Wait

secedit /configure /db c:\temp.sdb /cfg d:\p2v-scripts\off.inf > nul

echo Done!
echo.

echo =====
echo ==Very Silently Installing VMware Tools!=====
echo =====
echo.
echo Please Wait

change user /install > nul
msiexec -i "d:\p2v-scripts\vmtools\vmware tools.msi" addlocal=all /qn reboot="reallysuppress"
change user /execute > nul

echo Done!
echo.

echo =====
echo ==Re-enabling Driver Signing Warnings=====
echo =====
echo.
echo Please Wait

secedit /configure /db c:\temp.sdb /cfg d:\p2v-scripts\on.inf > nul

echo Done!
echo.

echo =====
echo ==Deleting temporary SDB File (c:\temp.sdb) =====
echo =====
echo.

del c:\temp.sdb

echo Done!
echo.

echo =====
echo ==Script Completed=====
echo =====

```

Note: off.inf looks like this:

```

[Unicode]
Unicode=yes
[Version]
signature="$chicago$"
Revision=1
[Registry Values]
MACHINE\Software\Microsoft\Driver Signing\Policy=3,0

```

and on.inf looks like this:

```

[Unicode]
Unicode=yes
[Version]
signature="$chicago$"
Revision=1
[Registry Values]

```

MACHINE\Software\Microsoft\Driver Signing\Policy=3,1

7. Another thing you might like to do is adjust the resolution in the Remote Console. Unfortunately, this can't be scripted with vbs – unless you use the "sendkey" feature – literally sending keystrokes to the display.cpl applet as if there was human at the keyboard/mouse. There is a nifty utility called multires which handles this much better – and it can be used in an unattended way from a batch file

Download multires from here:

<http://www.entechtaiwan.net/util/multires.shtm>

Although this downloads an installer – all we really need is multires.exe and multires.ini

Here is a sample script:

```
@echo off
cls
start multiRes /800,600
kill multiRes.exe
```

Note:

I use to kill from the MS Resource Kit, as multires loads up a tray-icon which we do not need

Warning:

I've had problems getting this script to run reliably and I don't why it doesn't

Using a VBS Script to Rename a Network Connection

Note:

- You might have noticed when you first boot the P2V'd virtual machine – a new network card is created. You will find that the name of this adapter is not "Local Area Connection" but normally "Local Area Connection 2" or "Local Area Connection 3"
- This is because the "old network" cards are still present and have not been removed – so it uses the next available name
- This cause use a problem in the next section because we are not sure what the name of the adapter is in Windows
- This script scans the system looking for "Local Area Connection n" and renames that connection to just "Local Area Connection"
- This works on with a single NIC virtual machine
- It was stolen from here, and modified to suit our purposes:

<http://www.microsoft.com/technet/scriptcenter/resources/qanda/all.msp#x>

```
Const NETWORK_CONNECTIONS = &H31&
```

```
Set objShell = CreateObject("Shell.Application")
Set objFolder = objShell.Namespace(NETWORK_CONNECTIONS)
Set colItems = objFolder.Items
For Each objItem in colItems
```

```
    If objItem.Name = "Local Area Connection 2" Then
        objItem.Name = "Local Area Connection"
    End If
```

```
    If objItem.Name = "Local Area Connection 3" Then
        objItem.Name = "Local Area Connection"
    End If
```

```
    If objItem.Name = "Local Area Connection 4" Then
        objItem.Name = "Local Area Connection"
    End If
```

```

If objItem.Name = "Local Area Connection 5" Then
    objItem.Name = "Local Area Connection"
End If

If objItem.Name = "Local Area Connection 6" Then
    objItem.Name = "Local Area Connection"
End If
Next

```

Using NetSH to Set a New IP Address

Note:

- NetSh is a Windows Command-Line tool which can manage nearly everything to do with network. Very cool, Very powerful tool. We can use it to set the IP address on the new adapter. The %1 %2 are command-line variables/parameters
- Where 1 = ip, 2 = subnet mask, 3 = default gateway and 4 = Primary DNS
- So if you put this into a batch file you would type:

```
ip 192.168.2.100 255.255.255.0 192.168.2.1 192.168.2.199
```

```

@echo off
cls
echo Changing your IP Settings. Please Wait...
netsh interface ip set address name="Local Area Connection" static %1 %2 %3 1
netsh interface ip set dns name="Local Area Connection" static %4

```

Note:

The number 1 at the end of the first netsh command sets the ip as the primary/preferred route for network traffic

Using a VBS Script to Enumerate Software (MSI)

Note:

- This script will **only** enumerate software installed using MSI – you must create a software.tsv for the script to work properly.
- **This is stolen from:**
<http://support.microsoft.com/default.aspx?scid=kb;en-us;821775>

```

Set objFSO = CreateObject("Scripting.FileSystemObject")
Set objTextFile = objFSO.CreateTextFile("c:\software.tsv", True)
strComputer = "."
Set objWMIService = GetObject("winmgmts:" _
    & "{impersonationLevel=impersonate}!\\" & strComputer & "\root\cimv2")
Set colSoftware = objWMIService.ExecQuery _
    ("SELECT * FROM Win32_Product")
objTextFile.WriteLine "Caption" & vtab & _
    "Description" & vtab & "Identifying Number" & vtab & _
    "Install Date" & vtab & "Install Location" & vtab & _
    "Install State" & vtab & "Name" & vtab & _
    "Package Cache" & vtab & "SKU Number" & vtab & "Vendor" & vtab & _
    & "Version"
For Each objSoftware in colSoftware
objTextFile.WriteLine objSoftware.Caption & vtab & _
    objSoftware.Description & vtab & _
    objSoftware.IdentifyingNumber & vtab & _
    objSoftware.InstallLocation & vtab & _
    objSoftware.InstallState & vtab & _
    objSoftware.Name & vtab & _
    objSoftware.PackageCache & vtab & _
    objSoftware.SKUNumber & vtab & _
    objSoftware.Vendor & vtab & _
    objSoftware.Version
Next
objTextFile.Close

```

There's another much shorter script I found which can just print a list of MSI installed software... I got this from my subscription from WindowsITpro email circular

```
strComputer = "."
Set objWMIService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\" & strComputer &
"\root\cimv2")
Set colSoftware = objWMIService.ExecQuery ("Select * from Win32_Product")
For Each objSoftware in colSoftware
    wscript.echo objSoftware.Caption
Next
```

I run this script from the command-line with `cscript //nologo software.vbs`

I can create a text file which lists all my MSI installed software with:

```
cscript //nologo software.vbs > c:\softwarelist.txt
```

Using a VBS Script to Uninstall Stale Software (MSI)

Note:

- Again this VBS script **only** works with MSI installed software
- **This was stolen from here:**
http://www.microsoft.com/technet/scriptcenter/guide/sas_cpm_qoai.mspx

```
strComputer = "."
Set objWMIService = GetObject("winmgmts:" _
& "{impersonationLevel=impersonate}!\\" & strComputer & "\root\cimv2")
Set colSoftware = objWMIService.ExecQuery _
("SELECT * FROM Win32_Product WHERE Name = 'MetaFrame Presentation Server Client'")
For Each objSoftware in colSoftware
    objSoftware.Uninstall()
Next
```

Using the Vendors Uninstall Routine because its NOT an MSI!

Note:

- Of course some vendors persist in not using MSI's so you might still have software in Add & Remove programs but you cannot use `msiexec` or `vbs` to remove that software
- This is especially true of drivers – because early implementations of MSI didn't support installing drivers...
- In this case you will be lumbered with the vendors uninstall routines which may or may not support scripting
- In my case the source Physical Server has 3 Intel Pro 1000 NIC's and the drivers required are already installed
- So I had to use the vendors uninstall routine to remove the drivers
- Unfortunately NONE of the standard methods worked to remove the drivers. In the end I had to raise a support request with Intel
- They sent me two links – the first utility I download didn't work at all
- The second one did work
- Rubbish eh? All I want to do is uninstall some software from Windows and there is 5 methods available – and only one of them worked how I wanted it to... plus its BETA Software!
- You install the software, which unpacks the `removencs` utility – which is command-line based
- **Link:**

Beta PROSet Uninstaller [P3UNIST.EXE]

http://downloadfinder.intel.com/scripts-df-external/Detail_Desc.aspx?agr=Y&ProductID=416&DwnldID=6036

- This tool is command-line based – unfortunately, it was totally unattended so and asked “are you sure style” questions – and there were no documented switches (nice!)
- The easiest way to handle these utilities is with a bit of DOS Redirection “Answer Files”. Yeah, so those DOS Advanced courses I used to teach in the 1993/4 are still worthwhile. Sometimes I wonder if industry ever really progresses!

To Create Answer File:

- 1. Open a CMD prompt**
- 2. Copy con answer.dat** [Enter]

{now use the keystrokes you would normally need to interact with the DOS utility}

y[enter]

- 3. [Control+Z]** to break copy con and create your file

- 4. This is my sample script**

```
@echo off
cls
echo =====
echo ==Uninstalling Intel Pro 1000 Drivers=====
echo =====
echo.
echo Please wait

RemoveNCS < answer.txt > nul

echo Done!
echo.

echo =====
echo ==Script Completed=====
echo =====
```

Discovering Phantom Hardware with Comm and DevCon

Note:

- There is a utility from Microsoft called DEVCON. It’s a CLI for Device Manager in Windows and works with Windows 2000, 2003 and XP.
- It is downloadable as a separate package – and also part of the Drive Development Kit for Windows (DDK). It is NOT a Linux command – and only runs on Windows!
- We can generate a report on the Physical Machine and Virtual Machine and parse the output to a text file. We can then use comparison of these two files – to build a 3rd file which lists all the hardware that needs removing from the Virtual Machine
- **NT4:**
There is a DDK for NT4.0. But I’ve not found it yet – and so have no idea if there is an equivalent tool in the DDK for it. If you have access to old MSDN boxes you might have it buried away in there. God, I bet you thought you’d never need those CD’s again did you! ☺
- **Remote Authentication:**
It doesn’t appear to be documented about user rights you need to use this tool remotely. This is has been my most common problem with DevCon.

The only other issue is I have faced is ordinary IP communication

- I've used it in a pure W2K3 environment as the root Domain Admin with success
- I've used it in a W2K Workgroup where the Administrator accounts password on both servers were the same
- I've tried using it on my laptop which has no account matching – and got "devcon failed"
- I've used it in a native W2K3 AD environment against a W2K server from a W2K3 server and got "devcon failed"
- It always works if you logon locally to the machine in question.
- It doesn't seem to support /user /password style switches
- If you think authentication is going to be a concern you might want to think of using PSEXEC from SysInternals to run the command under alternative credentials

<http://www.sysinternals.com/Utilities/PsExec.html>

If you do decide to use psexec it would do you know harm to read this tutorial on WindowsITPro:

<http://www.winnetmag.com/Windows/Issues/IssueID/714/Index.html>

- **Findings:**

I've run devcon findall * which lists every device and ends with a total of devices hidden and unhidden on a clean virtual machine and this is what I discovered

130 Devices:

Clean W2K with Terminal Services Virtual Machine (not P2V'd):

I ran my script against a range of different P2V events on different hardware and software builds

Here are some statistics:

Dell Optiplex GX1 (Single CPU/IDE Disks)	
Physical Machine	197
RAW P2V'd Virtual Machine	231
Virtual Machine after Scripts	152
Notes	I found that after removing the hardware – volume manager was re-detected by Windows – this appeared to only happen on IDE systems
ACPI Issues	P2V Instructions says not to make any changes – P2V Virtual Machine uses Standard PC

Dell PowerEdge 1650 (2xCPU, 2 SCA Disks, Adaptec SCSI Adapter) Windows 2000	
Physical Machine	157
RAW P2V'd Virtual Machine	184
Virtual Machine after Scripts	126
ACPI Issues	P2V Instructions says to downgrade to ACPI Multi-Processor to ACPI UniProcessor PC

Dell PowerEdge 1650 (2xCPU, 2 SCA Disks, Adaptec SCSI Adapter) Windows 2003	
Physical Machine	156
RAW P2V'd Virtual Machine	183
Virtual Machine after Scripts	124
Notes	P2V Instructions says to downgrade to Advanced Configuration and Power Interface (ACPI) PC which is different from a Windows 2000

Dell PowerEdge 1650 (2xCPU, 2 SCA Disks, Adaptec SCSI Adapter) Windows XP (Service Pack 2)	
Physical Machine	166
RAW P2V'd Virtual Machine	188
Virtual Machine after Scripts	135
Notes	P2V Instructions says to downgrade to Advanced Configuration and Power Interface (ACPI) PC which is different from a Windows 2000
Warning	Beware of Microsoft Firewall in Windows XP with Service Pack 2. This can stop communication from the Virtual Machine to the Physical Machine when running DevCon. This may need disabling on both connections

- **Anomalies:**

- If you show all hidden hardware on a cleanly installed virtual machine – you will see “phantom devices”. I think this maybe caused by a complete install of VMware Tools such as RAS Sync Adapter and Sound, Video and Game Controllers

Technically, sound isn't supported in a Virtual Machine on ESX – but my cleanly installed Virtual Machines have Drivers and Codecs. I'm imagine this is done case you want to move the Virtual Machine from ESX to WS where different hardware IS supported in the Virtual Machine

- **When DevCon Remove runs you might see responses** like:

**ROOT*PNP030B\1_0_22_0_32_0 : Remove failed
No devices removed.**

I'm not sure why these happen – I have a feeling it maybe a device

that has a relationship with a parent device. If you remove the parent then the child component cannot be removed because it isn't there.

Microsoft documentation online does touch upon this:

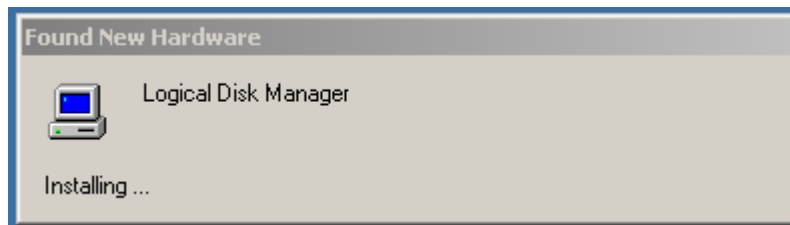
"The update affects only the devices that match the specified hardware ID, and does not affect the child devices"

This does suggest a Parent/Child relationship between some hardware components...

**ROOT\DMIO\0000 : Removed on reboot
Not all of 1 device(s) removed, at least one requires reboot to complete the operation.**

This one is a benign error message – which is self explanatory

- In Windows 2000 P2V processes after DevCon removes the hardware if found for one reason or another Windows re-Plug N Plays "Volume Manager"



The only thing I can relate this too is on a IDE system I got this error but on SCSI system I didn't

- **Some Very Useful Links:**

- **Microsoft Help and Support:**
<http://support.microsoft.com/default.aspx?scid=kb;en-us;311272>
- **Microsoft Technet**
<http://www.microsoft.com/technet/prodtechnol/windowsserver2003/library/TechRef/0f087656-fb2e-4828-9630-e76051a0a608.msp>

This link does not work properly – this is because Microsoft's website is just rubbish. The path to the info on DevCon is Windows Server 2003 Technical Reference > Tools and Settings Collection > Windows Support Tools > Disk and Data Management Support Tools

- **Rob van der Woude's Scripting Pages**
<http://www.robvanderwoude.com/devcon.html>
- **Computing.net - Change from standard PC to ACPI**
<http://www.computing.net/windows2003/wwwboard/forum/3128.html>

- **MSFN Forums – Get Hardware ID's**
<http://www.msfn.org/board/lofiversion/index.php/t50356.html>
- **The Win-32 Bit Versions of GNU Tools**
<http://gnuwin32.sourceforge.net/packages/coreutils.htm>

Note:

This can be done locally and remotely – beware though if BOTH the Physical and Virtual Machine can “see” each other on the SAME network you will face potentially IP and NetBIOS conflicts which can stop communication

1. **Download the DevCon package from Microsoft's Website (see links above)**
2. Unzip and copy from the i386 folder, devcon.exe to somewhere on your hard-drive – I chose to put my copy into the Windows directory

Note:

Doesn't matter what machine you use for this – you may wish to use it at the “helper” or your workstation. Beware you need administration rights over the remote machine to use DevCon. If remote functionality won't work – restore to a local logon

3. Type:

devcon -m:\physicalservername findall * > physicalmachine.txt

Note:

Where physicalservername IP address of the source of your P2V event. If you cannot get DevCon to work remotely – then you can always copy file to the physical machine and run the command locally

4. Repeat this for the Virtual Machine, but with one revision highlighted in bold

devcon findall * | find /v "acpi_hal\pnp0c08\0" > c:\virtualmachine.txt

Note:

The use of Microsoft's FIND /V excludes hardware ID (hwid) of “ACPI_HAL\PNP0C08\0” which the is relates to the friendly name of “Microsoft ACPI-Compliant System”. If this device is removed accidentally it causes a complete refresh of hardware by Windows and breaks the installation of VMware Tools.

Don't imagine that if you update the ACPI drivers first, and then do the comparison that this fixes you this problem. It is the hardware ID that were comparing not drivers

Note:

Next we need to compare each of these text files to each other – using hardware list in physicalmachine.txt to search for the same hardware (which needs removing) within the Virtual Machine.

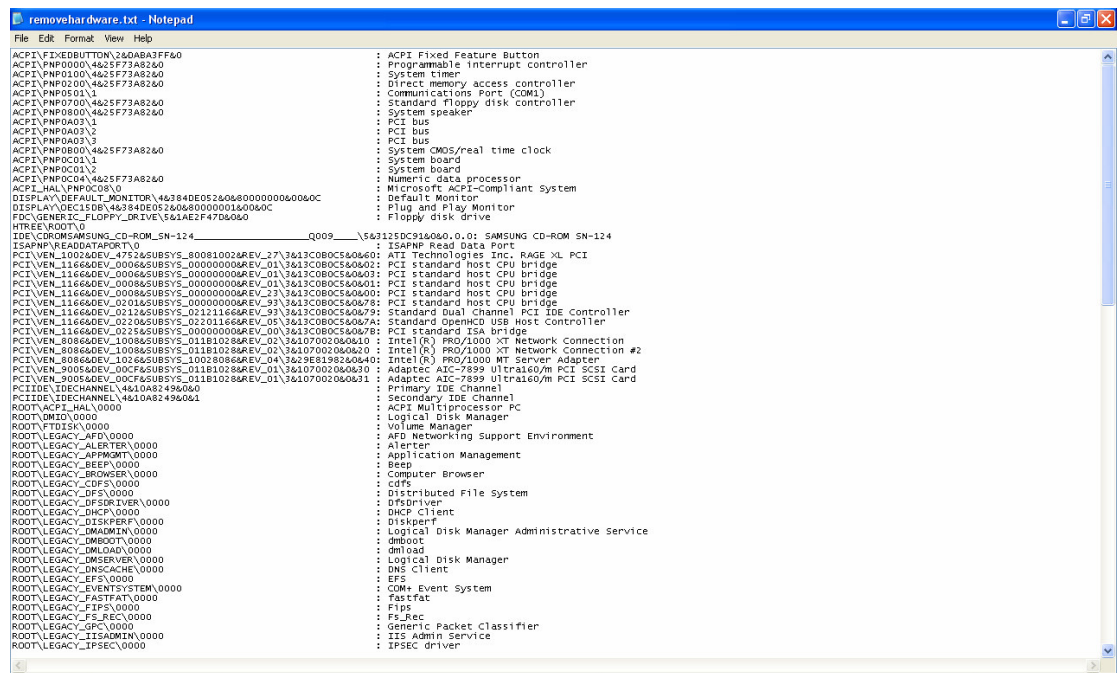
5. **comm -12 physicalmachine.txt virtualmachine.txt > removehardware.txt**

Note:

The comm command is normally only available to users of Linux. To use the comm command in Windows you need to download the GNU Core Utilities for Windows. If you would like to run this in stand-alone way (without having to install it to the Virtual Machine to make it work copy from the C:\Program Files\GnuWin32\bin these file:

comm.exe
libintl3.dll
libiconv2.dll

6. This creates a file within which each line represents a instance of hardware like so that shared in common between the Physical Machine and the Virtual Machine. This is a definitive list of all the "Stale" or "Phantom" hardware



To manually remove a piece of hardware type:

devcon remove "@ACPI\FIXEDBUTTON\2&DABA3FF&0"

Note:

You must use the @ sign – this used to indicate your using the hardware ID not the "friendly name". The speech marks must be used because of the & character which represents a space (I think)

Clearly doing this by hand using the command-line is very laborious – so what need is a batch file or command – which go through each entry and remove the hardware for us. Removing hardware often requires a reboot – so we might as well reboot the server after this process completes.

Below is a sample script of how to discover phantom devices:

Script 1 – Using DevCon where authentication is NOT a problem

```
@echo off
cls

echo =====
echo ==Producing hardware report for PHYSICAL MACHINE==
```

```

echo =====
devcon -m:\\%1 findall * > c:\physicalmachine.txt

echo.
echo          Done!
echo.

echo =====
echo ==Producing hardware report for VIRTUAL Machine==
echo =====

DEVCON FINDALL * | FIND /V "ACPI_HAL\PNP0C08\0" > C:\virtualmachine.txt

echo.
echo          Done!
echo.

echo =====
echo =====Analysis Results=====
echo =====

echo There are
type c:\physicalmachine.txt | findstr /r matching
echo in the PHYSICAL Machine
echo.

echo There are
type c:\virtualmachine.txt | findstr /r matching
echo in the VIRTUAL Machine
echo.

echo =====
echo =====Discovering Phantom Hardware=====
echo =====
echo.

comm -12 c:\physicalmachine.txt c:\virtualmachine.txt > c:\phantomhardware.txt

echo.
echo          Done!
echo.

```

Script2 - Using DevCon where authentication is IS a problem

```

if (%1) == () goto error

echo =====
echo ==Producing hardware report for PHYSICAL MACHINE==
echo =====
echo.
echo Please Wait

psexec \\%1 /u %2 /p %3 -c devcon.exe findall * > c:\physicalmachine.txt

echo Done!
echo.

echo =====
echo ==Producing hardware report for VIRTUAL Machine==
echo =====
echo.
echo Please Wait

DEVCON FINDALL * | FIND /V "ACPI_HAL\PNP0C08\0" > C:\virtualmachine.txt

echo Done!
echo.

echo =====
echo ==Analysis Results=====

```

```
echo =====
echo.

echo There are
type c:\physicalmachine.txt | findstr /r matching
echo in the PHYSICAL Machine
echo.

echo There are
type c:\virtualmachine.txt | findstr /r matching
echo in the VIRTUAL Machine
echo.

echo =====
echo ==Discovering Phantom Hardware=====
echo =====
echo.

comm -12 c:\physicalmachine.txt c:\virtualmachine.txt > c:\phantomhardware.txt

echo =====
echo ==Script Completed=====
echo =====
echo.

echo =====
echo ==Warning =====
echo =====
echo.
echo If you see "DevCon Failed" under "Producing hardware
echo report for PHYSICAL MACHINE". Then check two things:
echo.
echo 1. Can you ping the physical machine by its IP from
echo   Virtual Machine?
echo.
echo 2. Check you user rights/accounts. You need admin
echo   rights on both the Physical and Virtual Machine
echo   for this script to function correctly
goto end

:error
echo
=====
echo ==SYNTAX ERROR=====
echo You must specify 1 Parameter
echo 6.compare-pm2vm [IP Address of Physical Machine]
echo.
echo EXAMPLE:
echo 6.compare-pm2vm 192.168.2.200
echo
=====
echo
=====
echo.
echo Press Any Key to Continue
pause > nul

:end
```

Some Notes on PSEXEC:

Some anti-virus scanners report that one or more of the tools are infected with a "remote admin" virus. None of the PsTools contain viruses, but they have been used by viruses, which is why they trigger virus notifications. Several viruses use PsExec to propagate within a network, and as a result, several major antivirus products flag PsExec as a Trojan horse program or a worm. Remember that PsExec works on remote systems only if it runs within an account that has administrator group membership on the remote system. In other words, unless the account from which you run it has administrative access to a remote system, PsExec won't be able to execute a process on the remote system. In addition, PsExec's functionality can be achieved in other ways; thus, PsExec is only a convenience for virus writers, who could otherwise easily implement the functionality that PsExec provides.

Note that the password is transmitted in clear text to the remote system.

Requires that both the local and remote computers have file and print sharing (i.e., the Workstation and Server services) enabled and that the default Admin\$ share (a hidden share that maps to the \windows directory) is defined on the remote system.

The -c switch directs PsExec to copy the specified executable to the remote system for execution and delete the executable from the remote system when the program has finished running

PsExec starts an executable on a remote system and controls the input and output streams of the executable's process so that you can interact with the executable from the local system. PsExec does so by extracting from its executable image an embedded Windows service named Psexesvc and copying it to the Admin\$ share of the remote system. PsExec then uses the Windows Service Control Manager API, which has a remote interface, to start the Psexesvc service on the remote system.

The Psexesvc service creates a named pipe, psexecsvc, to which PsExec connects and sends commands that tell the service on the remote system which executable to launch and which options you've specified

Warning:

I've had intermittent problems with DevCon – sometimes it fails to communicate to the Physical Machine. It's not authentication, it's not IP (because I can ping) but it seems a reboot fixes it. It could be because I deliberately avoid the reboot during the VMware Tools installation

Using a Script to Create a Report "Phantom Hardware" with DevCon

Note:

- **Acknowledgement:**

I would like to take the opportunity to extend my heart felt thanks to Rob van der Woude. Rob was invaluable in responding to my emails about scripting this part of the process. Without Rob's input I wouldn't have got this far. Contrary to what you might think I am no scripting guru. I'm just persistent and have access to the internet. It was Rob's website that first led me to DevCon and made me realise that the removal of "phantom devices" could be scripted.

Script 2, is the one where Rob had critical input. It's based on his script [RenewUSB.bat](#). Rob's site is a gold mine of scripting tips and tricks. If you haven't been there yet – I would heartily recommend it

<http://www.robvanderwoude.com>

- This script is dependant on the previous section/script which gives us a definitive list of hardware to be removed in the file "phantomhardware.txt"
- This script merely generates a report of the hardware that could be removed...
- **Revisions to Script2:**
The critical line is in bold. Please note there is a space between the colon and the speech marks in the line =: ". This is slight difference in Rob's original script. Also note the line @echo DEVCON REMOVE stops the remove happening and merely prints the command to a cmd prompt which Script1 redirects to a text file for checking purposes...

Script 1:

```
@echo off
cls

echo =====
echo ==Generating Report of Hardware that will be removed=====
echo =====
echo.

call removehardware-test > c:\Report.txt

echo =====
echo ==Press any key open the report in Notepad=====
echo =====
echo.
echo Once you have read the report
echo Close Notepad to end this script...

pause > nul

notepad c:\Report.txt

echo =====
echo ==Script Completed=====
echo =====
```

Script 2:

```
@ECHO OFF
:: Check Windows version
IF NOT "%OS%"=="Windows_NT" GOTO Syntax
IF "%OS%"=="Windows_NT" SETLOCAL
VER | FIND "Windows NT" >NUL && GOTO Syntax

:: Check command line arguments -- none required
IF NOT "%~1"==" " GOTO Syntax

:: Check if DEVCON.EXE is available and if not, prompt for download
SET DevconAvailable=
SET Download=
DEVCON.EXE /? >NUL 2>&1
IF ERRORLEVEL 1 (
    SET DevconAvailable=No
    ECHO This batch file requires Microsoft's DEVCON utility.
    SET /P Download=Do you want to download it now? [y/N]
)

:: Start download if requested
IF /I "%Download%"=="Y" (
    START "DevCon" "http://support.microsoft.com/default.aspx?scid=kb;EN-US;Q311272"
    ECHO.
    ECHO Install the downloaded file and make sure DEVCON.EXE is in the PATH.
    ECHO Then try again.
)
```

```
:: Abort if DEVCON.EXE is not available yet
IF "%DevconAvailable%"=="No" GOTO End
```

```
FOR /F "tokens=1 delims=: " %%%A IN (c:\phantomhardware.txt) DO @echo DEVCON REMOVE "@%%~A"
```

```
:: Rescan for new hardware
rem DEVCON ReScan
```

```
:: Done
ENDLOCAL
GOTO End
```

```
:Syntax
ECHO.
ECHO removestalehardware.bat, Version 1.00 for Windows 2000/3 / XP
ECHO Use DEVCON to remove all Stale or "Phantom" devices let behind on a P2V'd Virtual Machin
ECHO.
ECHO Usage: removestalehardware
ECHO.
ECHO Notes: [1] This batch file requires Microsoft's DEVCON.EXE, available at
ECHO         http://support.microsoft.com/default.aspx?scid=kb;EN-US;Q311272
ECHO         You will be prompted for download if it isn't found.
ECHO         [2] I tested this batch file on my own P2V'd Windows 2000 Adv SP4.
ECHO         However, I cannot guarantee flawless operation on any other Virtual Machine.
ECHO         Use this batch file entirely at your own risk. Make sure you have
ECHO         a recent full backup available in case something might go wrong.
ECHO.
ECHO Written by Rob van der Woude
ECHO http://www.robvanderwoude.com
ECHO Modified by Mike Laverick
ECHO http://www.rtfm-ed.co.uk
```

```
:End
```

Using a Script to Remove "Phantom Hardware" with DevCon

Note:

- This script is EXACTLY the same as Script2 in the previous section – the only difference is that @echo DEVCON REMOVE has had the @echo part removed which does cause the script to execute
- I've also slightly modified the script to give some prompts to after the rescan has completed...

```
@ECHO OFF
:: Check Windows version
IF NOT "%OS%"=="Windows_NT" GOTO Syntax
IF "%OS%"=="Windows_NT" SETLOCAL
VER | FIND "Windows NT" >NUL && GOTO Syntax

:: Check command line arguments -- none required
IF NOT "%~1"==" " GOTO Syntax

:: Check if DEVCON.EXE is available and if not, prompt for download
SET DevconAvailable=
SET Download=
DEVCON.EXE /? >NUL 2>&1
IF ERRORLEVEL 1 (
    SET DevconAvailable=No
    ECHO This batch file requires Microsoft's DEVCON utility.
    SET /P Download=Do you want to download it now? [y/N]
)

:: Start download if requested
IF /I "%Download%"=="Y" (
    START "DevCon" "http://support.microsoft.com/default.aspx?scid=kb;EN-US;Q311272"
    ECHO.
    ECHO Install the downloaded file and make sure DEVCON.EXE is in the PATH.
    ECHO Then try again.
```

```

)

:: Abort if DEVCON.EXE is not available yet
IF "%DevconAvailable%"=="No" GOTO End

echo Removing Hardware NOW....
FOR /F "tokens=1 delims=: " %%A IN (c:\phantomhardware.txt) DO DEVCON REMOVE
"@%%~A"

cls
echo Hardware Removed

:: Rescan for new hardware
DEVCON ReScan
echo Please REBOOT the Virtual Machine

:: Done
ENDLOCAL
GOTO End

:Syntax
ECHO.
ECHO removestalehardware.bat, Version 1.00 for Windows 2000/3 / XP
ECHO Use DEVCON to remove all Stale or "Phantom" devices left behind on a P2V'd Virtual Machine
ECHO.
ECHO Usage: removestalehardware
ECHO.
ECHO Notes: [1] This batch file requires Microsoft's DEVCON.EXE, available at
ECHO          http://support.microsoft.com/default.aspx?scid=kb;EN-US;Q311272
ECHO          You will be prompted for download if it isn't found.
ECHO          [2] I tested this batch file on my own P2V'd Windows 2000 Adv SP4.
ECHO          However, I cannot guarantee flawless operation on any other Virtual Machine.
ECHO          Use this batch file entirely at your own risk. Make sure you have
ECHO          a recent full backup available in case something might go wrong.
ECHO.
ECHO Written by Rob van der Woude
ECHO http://www.robvanderwoude.com
ECHO Modified by Mike Laverick
ECHO http://www.rtfm-ed.co.uk

:End

```

Anomaly:

Logic would state that if you ran the comparison process again, after removing the phantom hardware – that the Physical and Virtual Machine would no longer share any hardware in common. However, after cleaning out “phantom hardware” and then comparing the cleaned P2V’d VM with the Physical Machine – they still appear to share hardware in common

I have a feeling that this is “software” that has been de-installed physical system – but is still lurking with Windows. For example this P2V system had previously had the Citrix Secure Gateway Service – I uninstalled it – but remains listed as a “legacy” device. For some reason DevCon doesn’t remove it from the Virtual Machine and the P2V process.

This is a report from a Dell Optiplex being compared to the Physical Machine after running the DevCon script to remove “Phantom Devices”

```

HTREE\ROOT\0
ISAPNP\READDATAPORT\0           : ISAPNP Read Data Port
ROOT\*PNP030B\1_0_22_0_32_0    : PC/AT Enhanced PS/2 Keyboard (101/102-Key)
ROOT\DMIO\0000                  : Logical Disk Manager
ROOT\FTDISK\0000                : Volume Manager
ROOT\LEGACY_AFD\0000            : AFD Networking Support Environment
ROOT\LEGACY_AIC78XX\0000        : aic78xx
ROOT\LEGACY_ALERTER\0000        : Alerter

```

ROOT\LEGACY_BEEP\0000 : Beep
 ROOT\LEGACY_BROWSER\0000 : Computer Browser
 ROOT\LEGACY_CLIENT_FOR_NFS\0000 : Client for NFS
 ROOT\LEGACY_CTXSECGWY\0000 : Secure Gateway Service
 ROOT\LEGACY_DFS\0000 : Distributed File System
 ROOT\LEGACY_DFSDRIVER\0000 : DfsDriver
 ROOT\LEGACY_DHCP\0000 : DHCP Client
 ROOT\LEGACY_DISKPERF\0000 : Diskperf
 ROOT\LEGACY_DMLOAD\0000 : dmload
 ROOT\LEGACY_DMSERVER\0000 : Logical Disk Manager
 ROOT\LEGACY_DNS\0000 : DNS Server
 ROOT\LEGACY_DNSCACHE\0000 : DNS Client
 ROOT\LEGACY_EFS\0000 : EFS
 ROOT\LEGACY_EVENTSYSTEM\0000 : COM+ Event System
 ROOT\LEGACY_FIPS\0000 : Fips
 ROOT\LEGACY_FS_REC\0000 : Fs_Rec
 ROOT\LEGACY_GPC\0000 : Generic Packet Classifier
 ROOT\LEGACY_IISADMIN\0000 : IIS Admin Service
 ROOT\LEGACY_IPFILTERDRIVER\0000 : IP Traffic Filter Driver
 ROOT\LEGACY_IPNAT\0000 : IP Network Address Translator
 ROOT\LEGACY_IPSEC\0000 : IPSEC driver
 ROOT\LEGACY_KEPCNFSD\0000 : KePcnfsd
 ROOT\LEGACY_KSECDD\0000 : KSecDD
 ROOT\LEGACY_LANMANSERVER\0000 : Server
 ROOT\LEGACY_LANMANWORKSTATION\0000 : Workstation
 ROOT\LEGACY_LICENSESERVICE\0000 : License Logging Service
 ROOT\LEGACY_LMHOSTS\0000 : TCP/IP NetBIOS Helper Service
 ROOT\LEGACY_MAPSVC\0000 : User Name Mapping
 ROOT\LEGACY_MESSENGER\0000 : Messenger
 ROOT\LEGACY_MNMDD\0000 : mnmdd
 ROOT\LEGACY_MRXSMB\0000 : MRXSMB
 ROOT\LEGACY_MSRTC\0000 : Distributed Transaction Coordinator
 ROOT\LEGACY_MSFS\0000 : Msfs
 ROOT\LEGACY_MUP\0000 : Mup
 ROOT\LEGACY_NDIS\0000 : NDIS System Driver
 ROOT\LEGACY_NDPROXY\0000 : NDProxy
 ROOT\LEGACY_NETBIOS\0000 : NetBIOS Interface
 ROOT\LEGACY_NETBT\0000 : NetBios over Tcpip
 ROOT\LEGACY_NETMAN\0000 : Network Connections
 ROOT\LEGACY_NFSNP\0000 : Nfsnp
 ROOT\LEGACY_NFSRDR\0000 : NfsRdr
 ROOT\LEGACY_NGDBSERV\0000 : Symantec Ghost Database Service
 ROOT\LEGACY_NGSERVER\0000 : Symantec Ghost Win32 Configuration
 Server
 ROOT\LEGACY_NPFS\0000 : Npfs
 ROOT\LEGACY_NTLMSSP\0000 : NT LM Security Support Provider
 ROOT\LEGACY_NTMSVC\0000 : Removable Storage
 ROOT\LEGACY_NULL\0000 : Null
 ROOT\LEGACY_PARVDM\0000 : ParVdm
 ROOT\LEGACY_PCNFSD\0000 : Server for PCNFS
 ROOT\LEGACY_POLICYAGENT\0000 : IPSEC Policy Agent
 ROOT\LEGACY_PORTMAP\0000 : Portmap
 ROOT\LEGACY_PROTECTEDSTORAGE\0000 : Protected Storage
 ROOT\LEGACY_RASACD\0000 : Remote Access Auto Connection Driver
 ROOT\LEGACY_RASAUTO\0000 : Remote Access Auto Connection Manager
 ROOT\LEGACY_RASMAN\0000 : Remote Access Connection Manager
 ROOT\LEGACY_RDBSS\0000 : Rdbss
 ROOT\LEGACY_RDPWD\0000 : RDPWD
 ROOT\LEGACY_REMOTEREGISTRY\0000 : Remote Registry Service
 ROOT\LEGACY_RPCSS\0000 : Remote Procedure Call (RPC)
 ROOT\LEGACY_RPCXDR\0000 : RpcXdr
 ROOT\LEGACY_SAMSS\0000 : Security Accounts Manager
 ROOT\LEGACY_SCARDDRV\0000 : Smart Card Helper
 ROOT\LEGACY_SCARDSVR\0000 : Smart Card
 ROOT\LEGACY_SCHEDULE\0000 : Task Scheduler
 ROOT\LEGACY_SECMON\0000 : RunAs Service
 ROOT\LEGACY_SENS\0000 : System Event Notification
 ROOT\LEGACY_SHAREDACCESS\0000 : Internet Connection Sharing
 ROOT\LEGACY_SMTPSVC\0000 : Simple Mail Transport Protocol (SMTP)
 ROOT\LEGACY_SPOOLER\0000 : Print Spooler
 ROOT\LEGACY_SPUD\0000 : Special Purpose Utility Driver

```

ROOT\LEGACY_SRV\0000           : Srv
ROOT\LEGACY_SVKP\0000         : SVKP
ROOT\LEGACY_TAPISRV\0000     : Telephony
ROOT\LEGACY_TCPIP\0000       : TCP/IP Protocol Driver
ROOT\LEGACY_TDTCP\0000       : TDTCP
ROOT\LEGACY_TERMDD\0000      : Terminal Device Driver
ROOT\LEGACY_TERMSERVICE\0000 : Terminal Services
ROOT\LEGACY_TRKWKS\0000      : Distributed Link Tracking Client
ROOT\LEGACY_VGASAVE\0000     : VgaSave
ROOT\LEGACY_WANARP\0000      : Remote Access IP ARP Driver
ROOT\LEGACY_WINMGMT\0000     : Windows Management Instrumentation
ROOT\LEGACY_WMII\0000        : Windows Management Instrumentation Driver
Extensions
ROOT\LEGACY_WUAUSERV\0000    : Automatic Updates
ROOT\MEDIA\MS_MMACHM         : Audio Codecs
ROOT\MEDIA\MS_MMDRV         : Legacy Audio Drivers
ROOT\MEDIA\MS_MMMCI         : Media Control Devices
ROOT\MEDIA\MS_MMVCD         : Legacy Video Capture Devices
ROOT\MEDIA\MS_MMVID         : Video Codecs
ROOT\MS_L2TPMINIPORT\0000    : WAN Miniport (L2TP)
ROOT\MS_NDISWANIP\0000       : WAN Miniport (IP)
ROOT\MS_PPTPMINIPORT\0000    : WAN Miniport (PPTP)
ROOT\MS_PSCHEMMP\0000       : WAN Miniport (IP) - Packet Scheduler
Miniport
ROOT\MS_PTMINIPORT\0000      : Direct Parallel
ROOT\PCI_HAL\0000            : Standard PC
ROOT\SYSTEM\0000             : Plug and Play Software Device Enumerator
ROOT\SYSTEM\0001             : Microcode Update Device
SW\{6C1B9F60-C0A9-11D0-96D8-00AA0051E51D}\{9B365890-165F-11D0-A195-0020AFD156E4}:
Microsoft Kernel GS Wavetable Synthesizer
SW\{8C07DD50-7A8D-11D2-8F8C-00C04FBF8FEF}\DMUSIC           : Microsoft DirectMusic SW Synth
(WDM)
SW\{A7C7A5B0-5AF3-11D1-9CED-00A024BF0407}\{9B365890-165F-11D0-A195-0020AFD156E4}:
Microsoft Kernel System Renderer
SW\{B7EAFDC0-A680-11D0-96D8-00AA0051E51D}\{9B365890-165F-11D0-A195-0020AFD156E4}:
Microsoft Kernel Audio Mixer
SW\{CD171DE3-69E5-11D2-B56D-0000F8754380}\{9B365890-165F-11D0-A195-0020AFD156E4}:
Microsoft WINMM WDM Audio Compatibility Driver
SW\{EEAB7790-C514-11D1-B42B-00805FC1270E}\ASYNCMAC         : RAS Async Adapter

```

Downgrading/Upgrading ACPI from Multi-Processor to Uni-Processor

Note:

- Again we can use DevCon to update our hardware
- I focused on downgrading Dual-Processor Physical Machines to Single Processor Virtual Machines – as I guess this is the most common practise
- Remember if you not altering the Processor state during the P2V then no downgrade/upgrade procedures are required
- So a single-CPU physical server can be P2V'd to a single-CPU Virtual Machine without the follow script
- I discovered and then modified this process based on a forum post on computing.net

<http://www.computing.net/windows2003/wwwboard/forum/3128.html>

- Put simply the Hardware ID (hwid) is associated with the drivers specified in the hal.inf. Devcon update then updates the driver using the instructions from hal.inf followed by the keyword which allows set which driver to use in the inf file. At the end a devcon reboot forces a reboot of the Virtual Machine. You may get a second reboot message from the Virtual Machine after this... thank you Windows!
- Bear in mind that when you downgrade a W2K Physical Machine, VMware's P2V Instructions files recommends using "ACPI UniProcessor PC"

- Whereas when you downgrade a W2K3/XP Physical Machine, VMware's P2V Instructions file recommends using "Advanced Configuration and Power Interface (ACPI) PC"
- These are two different options in the inf file – therefore we would need a downgrade script for W2K and a different one for W2K3/XP. W2K uses the ACPI**APIC**_UP, whereas W2K3/XP uses ACPI**PIC**_UP

Sample Downgrade Script (W2K)

```
@echo off
cls

echo =====
echo ==Downgrading ACPI to Uni-Processor=====
echo =====
echo.
echo Please Wait

devcon sethwid @ROOT\PCI_HAL\0000 := !E_ISA_UP !ACPIPIC_UP !ACPIPIC_UP !ACPIPIC_MP
!MPS_UP !MPS_MP !SGI_MPS_MP !SYSPRO_MP !SGI_MPS_MP > nul
devcon sethwid @ROOT\ACPI_HAL\0000 := !E_ISA_UP !ACPIPIC_UP !ACPIPIC_UP !ACPIPIC_MP
!MPS_UP !MPS_MP !SGI_MPS_MP !SYSPRO_MP !SGI_MPS_MP > nul;
devcon sethwid @ROOT\PCI_HAL\0000 := +ACPIPIC_UP > nul
devcon sethwid @ROOT\ACPI_HAL\0000 := +ACPIPIC_UP > nul
devcon update %windir%\inf\hal.inf ACPIPIC_UP > nul

echo Done!
echo.

echo =====
echo ==Script Completed=====
echo =====
echo.

echo =====
echo ==Press any key to reboot the Virtual Machine=====
echo =====

pause > nul

devcon reboot
```

Sample Downgrade Script (W2K3/XP)

```
@echo off
cls

echo =====
echo ==Downgrading ACPI to Uni-Processor=====
echo =====
echo.
echo Please Wait

devcon sethwid @ROOT\PCI_HAL\0000 := !E_ISA_UP !ACPIPIC_UP !ACPIPIC_UP !ACPIPIC_MP
!MPS_UP !MPS_MP !SGI_MPS_MP !SYSPRO_MP !SGI_MPS_MP > nul
devcon sethwid @ROOT\ACPI_HAL\0000 := !E_ISA_UP !ACPIPIC_UP !ACPIPIC_UP !ACPIPIC_MP
!MPS_UP !MPS_MP !SGI_MPS_MP !SYSPRO_MP !SGI_MPS_MP > nul
devcon sethwid @ROOT\PCI_HAL\0000 := +ACPIPIC_UP > nul
devcon sethwid @ROOT\ACPI_HAL\0000 := +ACPIPIC_UP > nul
devcon update %windir%\inf\hal.inf ACPIPIC_UP > nul

echo Done!
echo.

echo =====
echo ==Script Completed=====
echo =====
echo.

echo =====
echo ==Press any key to reboot the Virtual Machine=====
echo =====

pause > nul

devcon reboot
```

Sample Upgrade Script (W2K)

```
@echo off
cls

echo =====
echo ==Upgrading ACPI to Multi-Processor=====
echo =====
echo.
echo Please Wait

devcon sethwid @ROOT\PCI_HAL\0000 := !E_ISA_UP !ACPIPIC_UP !ACPIPIC_UP !ACPIPIC_MP
!MPS_UP !MPS_MP !SGI_MPS_MP !SYSPRO_MP !SGI_MPS_MP > nul
devcon sethwid @ROOT\ACPI_HAL\0000 := !E_ISA_UP !ACPIPIC_UP !ACPIPIC_UP !ACPIPIC_MP
!MPS_UP !MPS_MP !SGI_MPS_MP !SYSPRO_MP !SGI_MPS_MP > nul
devcon sethwid @ROOT\PCI_HAL\0000 := +ACPIPIC_MP > nul
devcon sethwid @ROOT\ACPI_HAL\0000 := +ACPIPIC_MP > nul
devcon update %windir%\inf\hal.inf ACPIPIC_MP > nul

echo Done!
echo.

echo =====
echo ==Script Completed=====
echo =====
echo.

echo =====
echo ==Press any key to reboot the Virtual Machine=====
echo =====

pause > nul

devcon reboot
```