

CHAPTER W1

Device Driver Tweaks

A *device driver* is a tiny chunk of programming code that serves as a kind of middleman between Windows and a particular device. For example, if Windows needs a device to do something, it alerts the device driver, and the device driver passes the request along in language that the device understands. Similarly, if a device has data or some other feedback for Windows, it first passes that data to the device driver, and the driver passes it to Windows.

For most folks, device drivers exist in the nether regions of the PC world, shrouded in obscurity and the mysteries of assembly language programming. As the middlemen brokering the dialogue between Windows and our hardware, however, these complex chunks of code perform a crucial task. After all, it's just not possible to unleash the full potential of your system unless the hardware and the operating system coexist harmoniously and optimally. This chapter presents you with a few driver-related tweaks that I hope will help you achieve and maintain that harmony.

IN THIS CHAPTER

- Tell Windows to Ignore Unsigned Device Drivers
- Configure Windows to Always Check for Device Drivers Online
- Write a Complete List of Device Drivers to a Text File

Tell Windows to Ignore Unsigned Device Drivers

Vista

XP



Medium

Device drivers that meet the Designed for Windows Vista (or XP) specifications have been tested for compatibility with Microsoft and are then given a digital signature. This signature tells you that the driver works properly with Windows and that it hasn't been changed since it was tested. (For example, the driver hasn't been infected by a virus or Trojan horse program.) When you're installing a device, if Windows Vista comes across a driver that has not been digitally signed, it displays a dialog box similar to the one shown in Figure W1.1. A Windows XP version of the dialog box is shown in Figure W1.2.

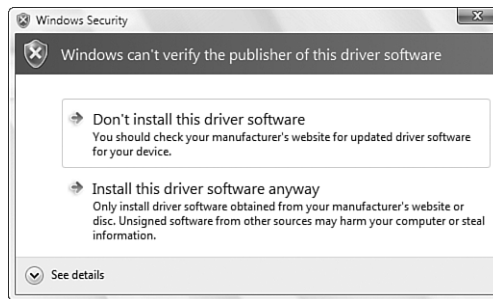


FIGURE W1.1

Windows Vista displays a dialog box similar to this one when it comes across a device driver that does not have a digital signature.

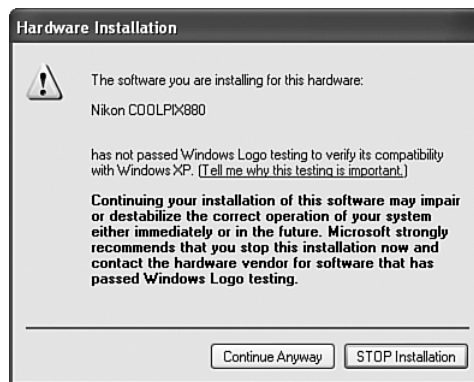


FIGURE W1.2

Windows XP displays a dialog box like this when it trips over a device driver without a digital signature.

If you click Don't Install This Driver Software (in Vista) or Stop Installation (in XP), Windows aborts the driver installation and you won't be able to use the device. This is the most prudent choice in this situation because an unsigned driver can cause all kinds of havoc, including lock-ups, BSODs (Blue Screens of Death), and other system instabilities. You should check the manufacturer's website for an updated driver that's compatible with your version of Windows, or you can upgrade to newer hardware that's supported by your version of Windows.

However, although not installing an unsigned driver is the *prudent* choice, it's not the most *convenient* choice because, in most cases, you probably want to use the device now rather than later. The truth is that *most* of the time these unsigned drivers cause no problems and work as advertised, so it's probably safe to continue with the installation. In any case, Windows always sets a restore point prior to the installation of an unsigned driver, so you can always restore your system to its previous state should anything go wrong.

By default, Windows gives you the option of either continuing or aborting the installation of the unsigned driver. You can change this behavior to automatically accept or reject all unsigned drivers by following these steps:

1. In Vista, select Start, type **gpedit.msc**, press Enter, and then enter your User Account Control credentials. In XP, select Start, Run to open the Run dialog box, type **gpedit.msc**, and click OK. Windows launches the Group Policy Object Editor.
2. Open the User Configuration\Administrative Templates\System\Driver Installation branch.
3. Double-click the Code Signing for Device Drivers policy. Windows displays the Code Signing for Device Drivers Properties dialog box.
4. Click Enable.

NOTE Test your system thoroughly after installing the driver: Use the device, open and use your most common applications, and run some disk utilities. If anything seems awry, roll back the driver. (In Device Manager, double-click the device, click the Driver tab, and then click Roll Back Driver.) If that doesn't work, use the restore point to roll back the system to its previous configuration.

NOTE If you're running a version of Windows that doesn't come with the Group Policy Editor, I'll show you a bit later how to perform this tweak using the Registry.

5. Use the When Windows Detects a Driver File Without a Digital Signature list to select one of the following items:
 - **Ignore**—Choose this option if you want Windows to install all unsigned drivers.
 - **Warn**—Choose this option if you want Windows to warn you about an unsigned driver by displaying one of the dialog boxes shown earlier in Figures W1.1 and W1.2.
 - **Block**—Choose this option if you do not want Windows to install any unsigned drivers.
6. Click OK.

TIP There are some device drivers that Windows knows will cause system instabilities. Windows will simply refuse to load these problematic drivers, no matter which action you choose in the Driver Signing Options dialog box. In this case, you'll see a dialog box similar to the one in either Figure W1.1 or W1.2, except this one tells you that the driver will not be installed, and your only choice is to cancel the installation.

If your version of Windows doesn't support the Group Policy Editor, follow these steps to set the driver signing options via the Registry:

1. In Vista, select Start, type **regedit**, press Enter, and then enter your User Account Control credentials. In XP, select Start, Run to open the Run dialog box, type **regedit**, and click OK. Windows launches the Registry Editor.
2. Navigate to the following key:
HKCU\Software\Policies\Microsoft\
3. If you don't see a Windows NT key, select Edit, New, Key, type **Windows NT**, and click OK.
4. Select Edit, New, Key, type **Driver Signing**, and click OK.
5. Select Edit, New, DWORD, type **BehaviorOnFailedVerify**, and click OK.
6. Double-click the BehaviorOnFailedVerify setting to open it for editing.
7. Type one of the following values:
 - **1**—(Ignore) Use this value if you want Windows to install all unsigned drivers.
 - **2**—(Warn) Use this value if you want Windows to warn you about an unsigned driver by displaying one of the dialog boxes shown earlier in Figures W1.1 and W1.2.

- 3—(Block) Use this value if you do not want Windows to install any unsigned drivers.
8. Click OK.

Configure Windows to Always Check for Device Drivers Online

Vista



Easy

If Windows Vista can't find drivers when you initially attach a device, it automatically checks Windows Update to see whether any drivers are available online. If Vista finds a driver, it installs the software automatically. In most cases, this is desirable behavior because it requires almost no input from you. However, lots of people don't like to use Windows on automatic pilot all the time because doing so can lead to problems. For example, it could be that you've downloaded the driver you actually want to use from the manufacturer's website, so you don't want whatever is on Windows Update to be installed. (And because manufacturer drivers often offer more features than Microsoft's drivers, you have another good reason to turn off the automatic pilot here.)

To gain control over Windows Update driver downloads, follow these steps:

1. Select Start, type **systempropertieshardware**, press Enter, and then enter your User Account Control credentials. Vista opens the System Properties dialog box with the Hardware tab displayed.
2. Click Windows Update Driver Settings. Vista displays the Windows Update Driver Settings dialog box, shown in Figure W1.3.



FIGURE W1.3

Use the Windows Update Driver Settings dialog box to control how Vista uses Windows Update to locate and install device drivers.

3. You have three choices:
 - **Check for Drivers Automatically**—This is the default setting and it tells Vista to go ahead and locate and install Windows Update drivers each time you attach a new device.
 - **Ask Me Each Time I Connect a New Device Before Checking for Drivers**—Activate this option to tell Vista to prompt you before it connects to Windows Update for drivers. If you want to control Windows Update driver installation, this is the ideal setting because it enables you to prevent those installs when you don't need them and to approve those installs when you do.
 - **Never Check for Drivers When I Connect a Device**—Activate this option to tell Vista to bypass Windows Update for all new devices. Use this option if you always use the manufacturer's device driver, whether it's on a disc that comes with the device or via the manufacturer's website.
4. Click OK to return to the System Properties dialog box.
5. Click OK.

If you activated the Ask Me Each Time I Connect a New Device Before Checking for Drivers option, the next time you attach a device for which Vista can't install drivers automatically, you see a Found New Hardware dialog box that offers three choices:

- **Yes, Always Search Online**—Click this option to restore the automatic Windows Update driver installation. (This is the same as choosing the Check for Drivers Automatically option in the Windows Update Driver Settings dialog box.)
- **Yes, Search Online This Time Only**—Click this option to have Vista search Windows Update for drivers only for the current device. (This is the same as choosing the Ask Me Each Time I Connect a New Device Before Checking for Drivers option in the Windows Update Driver Settings dialog box.)
- **Don't Search Online**—Click this option to bypass Windows Update for this device.

Write a Complete List of Device Drivers to a Text File

Vista

XP



There are times when you wish you had a list of all the drivers installed on your PC. For example, if your system crashes, it would be nice to have some kind of record of what drivers are in there. More likely, such a list would come in handy if you had to set up your PC from scratch and you wanted to know which drivers you had to update.

How do you get such a list? Oddly, Windows doesn't give you any straightforward way to do this. However, you can make your own list by using a script like the one shown in Listing W1.1.

NOTE The file containing the script in Listing W1.1—`WriteDeviceDrivers.vbs`—is available from my website at <http://mcfedries.com/cs/content/TweakItFreakIt.aspx>. See Chapter 37, "Running Scripts," to learn how to run the script on your PC.

Listing W1.1 A Script That Writes a Complete List of a PC's Installed Device Drivers to a Text File

```
Option Explicit
Dim strComputer, objWMI, collDrivers, objDriver, intDrivers
Dim objFSO, strFolder, objFile
'
' Change the following value to the path of the folder
' where you want to store the text file
'
strFolder = "d:\backups\"
'
' Initialize the file system object
'
Set objFSO = CreateObject("Scripting.FileSystemObject")
'
' Create the text file
'
Set objFile = objFSO.CreateTextFile(strFolder & "drivers.txt", True)
'
' Get the WMI object
'
strComputer = "."
Set objWMI = GetObject("winmgmts:\\." & strComputer)
'
```

Continues

Listing W1.1 Continued

```

' Return the collection of device drivers on the computer
'
Set collDrivers = objWMI.ExecQuery _
    ("Select * from Win32_PnPSignedDriver")
'
' Run through each item in the collection
'
intDrivers = 0
For Each objDriver in collDrivers
    '
    ' Write the driver data to the text file
    '
    objFile.WriteLine(objDriver.DeviceName)
    objFile.WriteLine("=====")
    objFile.WriteLine("Device Class: " & objDriver.DeviceClass)
    objFile.WriteLine("Device Description: " & objDriver.Description)
    objFile.WriteLine("Device ID: " & objDriver.DeviceID)
    objFile.WriteLine("INF Filename: " & objDriver.InfName)
    objFile.WriteLine("Driver Provider: " & objDriver.DriverProviderName)
    objFile.WriteLine("Driver Version: " & objDriver.DriverVersion)
    objFile.WriteLine("Driver Date: " &
↳ReturnDriverDate(objDriver.DriverDate))
    objFile.WriteLine("")
    intDrivers = intDrivers + 1
Next
'
' Close the text file
'
objFile.Close
WScript.Echo "Wrote " & intDrivers & " drivers to the text file."

'
' ReturnDriverDate()
' This function takes the driver datetime value and converts
' it to a friendlier date and time format
'
Function ReturnDriverDate(dDriverDate)
    Dim eventDay, eventMonth, eventYear

```



```
Dim eventSecond, eventMinute, eventHour
eventYear = Left(dDriverDate, 4)
eventMonth = Mid(dDriverDate, 5, 2)
eventDay = Mid(dDriverDate, 7, 2)
eventHour = Mid(dDriverDate, 9, 2)
eventMinute = Mid(dDriverDate, 11, 2)
eventSecond = Mid(dDriverDate, 13, 2)
ReturnDriverDate = DateSerial(eventYear, eventMonth, eventDay) & _
    " " & TimeSerial(eventHour, eventMinute, eventSecond)
```

End Function

The script uses VBScript's `FileSystemObject` to connect to the PC's file system. In this case, the script uses `FileSystemObject` to create a new text file in the folder specified by `strFolder`. The script then sets up the usual Windows Management Instrumentation (WMI) object, and then uses WMI to return the collection of installed device drivers. A `For Each...Next` loop goes through each device and writes various data to the text file, including the device name and description, and the driver version and date.

