



Windows Vista™

Step-By-Step Guide to Device Driver Signing and Staging

Microsoft Corporation

Published: July 2006

Author: Dave Bishop

Editor: Scott Somohano

Abstract

By using device driver signing and staging in Microsoft® Windows Server® Code Name "Longhorn" and Windows Vista™ you can increase the security of your computers by allowing users to install only those device drivers that you approve. This guide describes how to sign and securely stage driver packages on a computer in a manner that allows a user to install the package without requiring administrative privileges.

Microsoft

This document supports a preliminary release of a software product that may be changed substantially prior to final commercial release, and is the confidential and proprietary information of Microsoft Corporation. It is disclosed pursuant to a non-disclosure agreement between the recipient and Microsoft. This document is provided for informational purposes only and Microsoft makes no warranties, either express or implied, in this document. Information in this document, including URL and other Internet Web site references, is subject to change without notice. The entire risk of the use or the results from the use of this document remains with the user. Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2006 Microsoft Corporation. All rights reserved.

Microsoft, MS-DOS, Windows, Windows Server, Windows Vista, and Active Directory are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

References to any third-party products or their hardware identifiers are for illustrative purposes only. These products are not endorsed by Microsoft Corporation.

All other trademarks are property of their respective owners.

Contents

Step-by-Step Guide to Device Driver Signing and Staging	5
Who should use this guide?	5
Benefits of signing and staging driver packages	6
Scenario overview	6
Driver package signing and staging scenarios.....	8
Signing a device driver package.....	8
Staging a device driver package in the driver store.....	8
Configuring clients to search a shared network folder for device driver packages ..	8
Technology review.....	9
Digital signatures and certificates	9
Guidance for Safeguarding Code Signing Keys	11
Device installation	11
Group Policy	13
Requirements for device driver signing and staging	14
Prerequisite procedures	15
Responding to the User Account Control page	15
Enable the Run option in the Start menu.....	16
Disable automatic searching of Windows Update for device drivers.....	17
Install the Windows Driver Kit	17
Configure the Toaster sample device driver package for use in this guide.....	18
Install the Toaster Bus device driver.....	20
Steps for signing a device driver package.....	21
Steps outline: Signing a device driver package	21
Step 1: Create a digital certificate for signing.....	22
Step 2: Add the certificate to the Trusted Root Certification Authorities store.....	24
Step 3: Add the certificate to the per machine Trusted Publishers store.....	25
Step 4: Sign the device driver package with the certificate.....	26
Steps for staging a device driver package in the driver store.....	31
Steps Outline: staging a device driver package in the driver store	31
Step 1: Attempt to stage an unsigned driver package	31
Step 2: Attempt to stage an signed, but improperly modified driver package.....	32
Step 3: Attempt to stage the properly signed driver package	33
Step 4: Test installation of the package	34
Steps for configuring a shared network folder to hold signed device driver packages .	35

Steps outline: Configure a shared network folder to hold signed device driver packages.....	36
Step 1: Create the folder to contain device driver packages	36
Step 2: Configure the client computer to search the folder for driver packages.....	37
Step 3: Configure the client computer to allow standard users to install the device..	37
Step 4: Remove the device driver and driver package installed in the previous procedure	39
Step 5: Attempt installation of the device driver package.	40
Conclusion	42
Logging bugs and feedback	42
Additional Resources.....	42

Step-by-Step Guide to Device Driver Signing and Staging

This step-by-step guide uses a sample device and driver to demonstrate how to securely deliver device driver packages to client computers in a lab environment so that a standard user can install them without any assistance from an administrator or user interface prompts.

Important

The steps provided in this guide are intended only for use in a test lab environment. This Step-by-Step guide is not meant to be used in a production environment, and should be used with discretion as a stand-alone document.

Specifically, in Microsoft® Windows Vista™ and Windows Server® Code Name "Longhorn" you can perform the following tasks:

- Sign device driver packages with digital certificates, and then place those certificates on client computers so that users do not have to determine whether a device driver or its publisher is "trusted."
- Stage device driver packages in the protected driver store on a client computer so that a standard user can install the package without requiring administrator rights.
- Configure client computers to search specific shared network folders for a driver package when a new hardware device is discovered but a driver package is not already staged on the local computer.

Who should use this guide?

This guide is for the following audiences:

- IT professionals responsible for deploying device drivers to client computers running Windows Vista or Windows Server "Longhorn"
- IT planners and analysts who are evaluating Windows Vista and Windows Server "Longhorn"
- Security architects who are responsible for implementing trustworthy computing
- Administrators who want to become familiar with the technology

Benefits of signing and staging driver packages

Because device driver software runs as a part of the operating system with unrestricted access to the entire computer, it is critical that only known and authorized device drivers are permitted. Signing and staging your device driver packages on client computers by using the techniques described in this guide provide the following benefits:

- **Improved security.** Before Windows Vista and Windows Server "Longhorn", standard users could not install device drivers without assistance from an administrator. Users often logged on with user accounts that were members of the Administrator's group. The rights associated with Administrator group membership allows a user to carry out required tasks, but they also allow the user to carry out actions that can compromise security or configure the computer so that it does not run correctly.

With Windows Vista and Windows Server "Longhorn", you can allow standard users to install approved device drivers without compromising computer security or requiring help desk assistance.

- **Reduced support costs.** Users can only install devices that your organization has tested and is prepared to support. You therefore maintain the security of the computer while simultaneously reducing the demands on your helpdesk.
- **Better user experience.** A driver package that is staged in the driver store works automatically when the user plugs in the device. Alternatively, driver packages placed on a shared network folder can be discovered whenever the operating system detects a new hardware device. In both cases, the user is not prompted before installation.

Scenario overview

Windows Server "Longhorn" and Windows Vista include several features that allow an administrator to make device driver installation easier for users. You have the ability to stage driver packages in a protected area of a user's computer called the driver store. A standard user, without any special privileges or permissions, can install a driver package that is in the driver store. You can also configure client computers to automatically search an administrator-specified list of folders (and their subfolders) when a new device is attached to the computer. These folders can be local to the computer or hosted on a network share. When a device driver is accessible in this manner, Windows will not need to prompt the user to insert media. These features improve the user experience and reduce help desk support costs by allowing standard users to install approved driver

packages without requiring additional permissions or the assistance of an administrator. These features also increase the security of your computers by ensuring that your standard users can only install those driver packages which you authorize and trust.

 **Note**

In this guide, the term "device driver" refers to the installed, configured, and operational software required to use a hardware device on a Windows computer.

The terms "device driver package," "driver package," or "package" refer to the complete set of files required to install the device driver.

The term "administrator" refers to any user logged on to the computer with an account that is a member of the local Administrators group.

The term "standard user" refers to any user logged on to the computer with an account that has no elevated permissions through group membership or other delegation of rights.

In this guide, you create a test certificate, and manually install it in the certificate store of the client computer. In an enterprise production environment, use more scalable procedures, including the following:

- A commercially acquired digital certificate. This is an important option if the certificate must be usable by computers outside of your organization. These certificates typically must be purchased.
- Certificates generated by an internal certificate authority computer, such as a computer running Windows Server and Certificate Services. This is a good option when certificates for many purposes are required within the organization, and the cost for acquiring that many commercial certificates is prohibitive.
- The use of Group Policy to deploy the certificates to client computers. Group Policy allows you to have the certificate automatically installed to all managed computers in a domain, organizational unit, or site.

To maintain and safeguard the stability of the operating system, only administrators can install unsigned device drivers. An organization's administrator can use the procedures in this guide to sign packages that are not previously signed by the vendor to make the packages usable in the organization. The administrator can also use this procedure to replace the vendor's signature with one created by the organization's certificate. If all packages are signed with the organization's certificate, then only that one certificate needs to be deployed.

If a standard user attempts to install a device whose driver package is not yet in the store, Windows attempts to stage the driver package. Staging succeeds if the user can

supply administrator credentials, or the package is for a device with a setup class identifier that is permitted via device installation policy on the computer. If the user cannot complete staging, then the user cannot install that device.

Driver package signing and staging scenarios

This guide describes several tasks that involve delivering device driver packages to client computers.

Signing a device driver package

With Windows Vista and Windows Server "Longhorn", you can digitally sign driver packages. This task describes the steps required to create a test certificate, use the certificate to sign a device driver package, validate that the device driver package is properly signed, and configure a client computer to accept that signature.

Staging a device driver package in the driver store

This scenario describes the steps required to place a device driver package in the driver store, and demonstrates how a driver package is installed from the store when a new hardware device is plugged into the computer.

By staging a driver package, an administrator enables the user to plug in the corresponding device and Windows installs the device driver with no requirement for elevated permissions, or the need for the user to approve a digital signature. The driver package files are all available on the client computer, and have been security checked, so they install silently and the device works.

Configuring clients to search a shared network folder for device driver packages

This scenario describes the steps required to configure a client computer to search for device driver packages on a designated shared network folder when Windows detects a new hardware device, and configure the system to allow standard users to install those driver packages.

Device driver packages can be placed on a network share accessible to all client computers. and Windows Server "Longhorn" support a registry setting that allows you to configure the list of folders that the Windows searches to find driver packages that are not already in the driver store. This list can include a network path to a folder on a server.

Device drivers located on a network share must still be staged in the driver store prior to installation, and all requirements for staging are still enforced. If the driver packages on that network folder are signed with trusted certificates, and you have granted the user permissions (through device installation policy) to devices of this device setup classes, then the installation succeeds silently. If the package has not been signed, or if the appropriate device installation policy is not configured, then the user is prompted to accept the publisher and is prompted to supply administrator credentials.

Technology review

The following sections provide a brief overview of the core technologies discussed in this guide.

Digital signatures and certificates

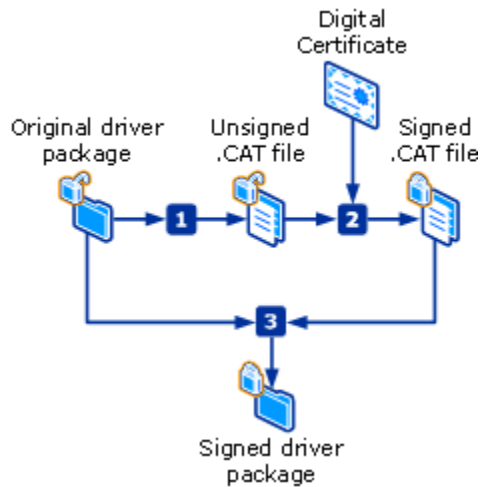
Because device drivers run with system-level privileges and can access anything on your computer, it is critical that you trust the device drivers that you install. Trust, in this context, includes two main principles:

- **Authenticity** is a guarantee that the package came from its claimed source. It cannot be malicious code masquerading as something legitimate.
- **Integrity** is an assurance that the package is 100 percent intact, and has not been modified by anyone after it was released.

Windows uses digital certificates and digital signatures to provide support for these principles. A **digital certificate** identifies an organization, and it is trustworthy because it can be checked electronically by a certification authority (CA). A **digital signature** uses the organization's digital certificate to encrypt specific details about the package.

The encrypted information in a digital signature includes a **thumbprint** for each file included with the package. The thumbprint is generated by a special cryptographic algorithm referred to as a hashing algorithm. The algorithm generates a code that could only be created by that file's contents. Changing a single bit in the file changes the thumbprint. After the thumbprints are generated, they are combined together into a catalog, and then encrypted.

The following figure shows the process used to sign a driver package.



In this process the following steps take place:

The original driver package has no signature, and no .cat file in which a signature can be placed. In Step 1 of the diagram the **Signability** tool is run to create the .cat file, in which it places a thumbprint for each file identified as part of the driver package, as specified in the .inf file. In Step 2, the **SignTool** tool is run, specifying a digital certificate to encrypt, and thus sign, the .cat file. In Step 3, the digitally signed .cat file is included with the driver package and deployed to client computers.

The recipient computer confirms the identity of the package originator by using a copy of the certificate to decrypt the signature on the package. A successful decryption proves that the owner of the certificate is the signer of the package.

The same hashing algorithm used to create the thumbprints is used again during the confirmation process. Windows generates a thumbprint for each file received in the package. If the thumbprints generated by the receiving computer are identical to the ones found encrypted in the signature, then the recipient can be sure that the received package is identical to the original. If the thumbprints do not match, then the files were altered in some way after they were signed, and should not be trusted.

On each computer, Windows maintains a store for digital certificates. As the computer administrator, you can add certificates from publishers that you trust. If a package is received for which a matching certificate cannot be found in the certificate store, then Windows presents a page asking the user to confirm that the publisher is trusted. By placing a certificate in the certificate store on all of your client computers, you are telling Windows that all packages signed by that certificate are trusted.

Important

The 64-bit versions of Windows Vista and Windows Server "Longhorn" require that all kernel mode device drivers be signed with a Software Publishing Certificate issued by a certification authority. If you use a 64-bit version of Windows Vista, then you need a driver package that is already signed or have access to a Software Publishing Certificate with which you can sign the driver package. If you sign a 64-bit kernel mode device driver incorrectly, it will not load or run successfully. If the device driver is required to start the computer, your computer might fail to start. Ensure that you test your packages thoroughly on each type of computer on which you will deploy them.

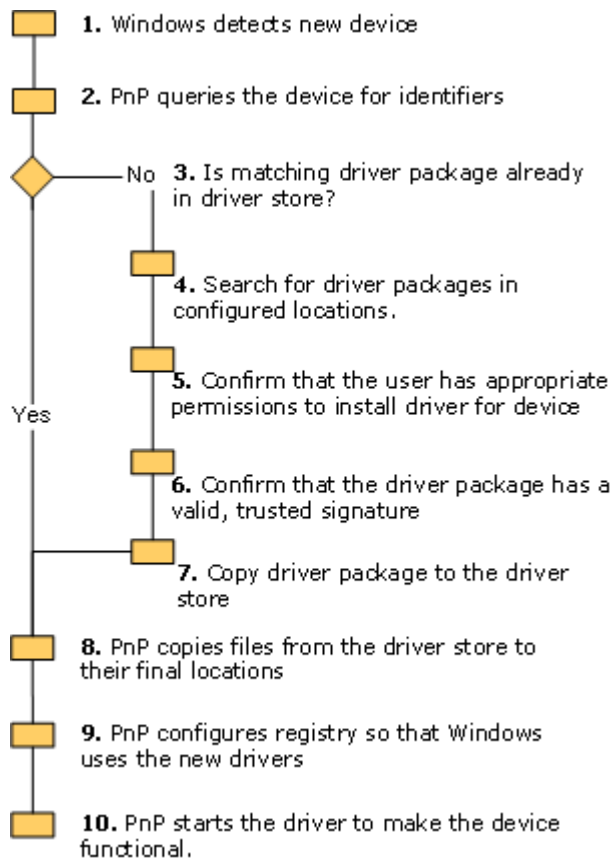
Guidance for Safeguarding Code Signing Keys

The cryptographic keys that are at the heart of the Authenticode signing process must be well protected and treated with the same care as the organization's most valuable assets. These keys represent an organization's identity. Any code that is signed with these keys appears to Windows as if it contains a valid digital signature that can be traced to the organization. If the keys are stolen, they could be used to fraudulently sign malicious code and possibly result in the delivery of code that contains a Trojan or virus that appears to come from a legitimate publisher.

For detailed information about safe guarding private keys, see the "Code Signing Best Practices" document referenced in the [Additional Resources](#) section at the end of this guide.

Device installation

A device is a piece of hardware with which Windows interacts to perform some function. Windows can communicate with a device only by using a piece of software called a device driver. Device and device driver installation in Windows Vista and Windows Server "Longhorn" operate as shown in the following diagram. "PnP" in the diagram refers to the Plug and Play service running in Windows. If any of the described security checks fail, or if Windows cannot find an appropriate device driver package, then the process stops.



1. When a user inserts a device, Windows detects the new hardware, and then signals the Plug and Play service to make the device operational.
2. Plug and Play queries the device for identification strings.
3. Plug and Play searches the driver store for a driver package that matches the identification strings. If a matching package is found, then skip to step 8. If a matching package is not found, then continue with step 4.
4. Windows searches for a matching driver package by looking in the following locations. If more than one package is found, Windows the "best" one. See the article "How Setup Selects Device Drivers" document referenced in the [Additional Resources](#) section at the end of this guide.
 - Searching folders specified by the DevicePath registry entry
 - Searching the Windows Update Web site
 - Prompting the user for media

5. Windows checks that the user has permission to place the driver package in the driver store. The user must have administrator credentials, or computer policy must be set to allow standard users to install specific devices.
6. Windows checks that the driver package has a valid digital signature. If the driver package is signed by a certificate that is valid, but that is not found in the Trusted Publishers store, then Windows prompts the user for confirmation.
7. Windows places a copy of the driver package in the driver store.
8. Plug and Play copies the device driver files from the driver store to their operational locations, typically %systemroot%\windows32\drivers.
9. Plug and Play configures the registry to instruct Windows how to use the newly installed device driver.
10. Plug and Play starts the newly installed device driver. This step is repeated each time the computer is restarted to reload the drivers.

For more information about this process, see Device Manager Help in Windows Vista or Windows Server "Longhorn".

In Windows Vista and Windows Server "Longhorn", the process described in steps 4 through 7 is referred to as staging. Staging involves Windows performing all required security checks, and then placing the driver package in a secure location so it can be accessed by the Plug and Play service. In Windows Vista and Windows Server "Longhorn" staging can be performed by an administrator as a separate step. After a driver package has been staged, any user that logs on to that computer can install the device driver in the driver store by plugging in the appropriate device. There are no prompts, and no special permissions are required. The user plugs in the device and it works, without administrator or helpdesk intervention.

For more information about the driver store see the [Additional Resources](#) section at the end of this guide.

Group Policy

Group Policy is the means by which an administrator can centrally enforce security and configuration settings on managed client computers throughout an organization. It also supports automatic deployment of digital certificates to computers that are members of a domain. Instead of having to visit and manually configure settings on each client computer, or write complicated logon scripts, you configure the desired settings once and distribute them to your managed computers using the Active Directory® directory service that is available with Windows Server "Longhorn", Windows Server 2003, and Windows 2000 Server.

This guide demonstrates procedures that involve manually configuring the client, including the manual installation of the certificates used to sign driver packages. However, in a production environment with many client computers, using Active Directory Group Policy is a much more efficient, less error-prone, and more secure method for enforcing company policy and security settings on your managed computers.

For more information about Group Policy and Active Directory, see the [Additional Resources](#) section at the end of this guide.

Requirements for device driver signing and staging

To successfully complete the procedures in this guide, you must meet the following requirements:

- A client computer running Windows Vista 32-bit edition. This guide refers to this computer as **DMI-Client1**.

Important

The 64-bit versions of Windows Vista and Windows Server "Longhorn" have special signature requirements for kernel mode device drivers. If you use a 64-bit version of Windows, then you must also use a Software Publishing Certificate for this purpose from an approved certificate authority. For more information, see the [Additional Resources](#) section at the end of this guide.

- A device and its corresponding device driver package. The device driver package must not already be present on the computer, as either part of the in-box device driver set supplied by Microsoft with Windows, or already in the driver store on the **DMI-Client1**. If the device was previously installed on the computer then the driver package is likely to be in the driver store, and must be removed before starting the procedures in this guide. As long as the driver package is not one of the in-box driver packages provided with Windows, you can follow the steps in [Remove driver installed in the previous procedure](#) to remove the old copy and prepare the computer for this guide. The procedures in this guide use the sample "Toaster" device driver package which you can get as part of the Windows Driver Kit. If you choose to use some other device with its device driver package, the screens you see may differ from those described in this guide, and you may have to adapt certain steps to work with the driver package you are using.
- Access to a protected administrator account on **DMI-Client1**. This guide calls this account **TestAdmin**. The procedures in this guide require administrator privileges for

most steps. You must be logged into DMI-Client1 using this administrator account at the beginning of each procedure, unless you are directed otherwise.

 **Note**

Windows Vista and Windows Server "Longhorn" introduce the concept of a protected administrator account. This account is a member of the Administrators group, but by default that security token is not directly used. Any attempt to carry out a task that requires the elevated rights of an administrator generates a dialog box asking for permission to perform that task. This dialog box is discussed in the section [Responding to the User Account Control page](#). Microsoft recommends that you use a protected administrator account, rather than the built-in Administrator account whenever possible.

- Access to a standard user account on **DMI-Client1**. This user account has no special memberships that grant any kind of elevated permissions. This guide calls this account **TestUser**. Only log into your computer with this account when instructed to do so. With a standard user account, any attempt to carry out a task that requires the elevated rights of an administrator can generate a dialog box requesting the credentials of an account with administrator privileges. This dialog box is discussed in the section [Responding to the User Account Control page](#).

Prerequisite procedures

Use the following procedures to configure your computer for the procedures in this guide.

1. [Responding to the User Account Control page](#)
2. [Enable the Run option in the Start menu](#)
3. [Disable automatic searching of Windows Update for device drivers](#)
4. [Install the Windows Driver Kit](#)
5. [Configure the Toaster sample device driver package](#)
6. [Install the Toaster Bus device driver](#)

Responding to the User Account Control page

Membership in the Administrators group, or equivalent, is the minimum required to complete many procedures in this guide. In Windows Vista and Windows Server "Longhorn", when you attempt to perform a procedure that requires administrator rights, the following occurs:

- The built-in administrator account is disabled by default. However, if you are logged in as the built-in Administrator account (not recommended) then the operation proceeds.
- If you are logged on as a member of the Administrators group that is not the built-in Administrator account, then a **User Account Control** dialog box appears requests permission to continue. Click **Continue** to allow the operation to proceed.
- If you are logged on as a standard user, then you could be prevented from performing the procedure. Depending on the procedure, a **User Account Control** dialog box might request the user name and password for an administrator account. If you provide valid credentials, then the operation runs in the security context of the administrator account you provided. If you cannot provide administrative credentials, then you are prevented from performing the procedure.

Important

Before providing credentials to run any administrative operation, ensure that the **User Account Control** page is displayed in response to an operation that you initiated. If the page appears unexpectedly, click the **Details** button, and then ensure that the task that is one you wish to allow.

This guide does not specify every occurrence of the **User Account Control** dialog box that you might encounter in performing these procedures. When special steps are required to run specific tasks as an administrator, those steps are documented in the guide.

Enable the Run option in the Start menu

The **Run** menu option does not appear by default on a computer running Windows Vista. Enabling the **Run** option in the Start menu will simplify many steps later in the guide.

To enable the Run option on the Start menu

1. Right-click the **Start** button, and then click **Properties** to display the **Taskbar and Start Menu Properties** page.
2. On the **Start Menu** tab, next to the **Start menu** option, click **Customize**.
3. In the list on the **Customize Start Menu** dialog box, select the **Run command** check box.
4. Click **OK** twice.
5. Click the **Start** button, and then verify that the **Run** option is present.

Disable automatic searching of Windows Update for device drivers

By default, when searching for a device driver package, Windows includes the driver library maintained on the Windows Update Web site.

Inclusion of Windows Update in the search is very useful for home users. However, many administrators need more control over which device drivers users can install. You can configure a computer policy to disable the inclusion of Windows Update in the search for device drivers. If you are using a device whose driver package is available on Windows Update for the scenarios in this guide, then you must use the following procedure for the scenarios to work as described.

To disable automatic searching of Windows Update for device drivers

1. Click **Start**, right-click **Computer**, and then click **Properties**.
2. In the **Tasks** list, click **Advanced System Settings**.
3. On the **System Properties** dialog box, click the **Hardware** tab, and then click **Windows Update Driver Settings**.
4. Select **Never check for drivers when I connect a device**.
5. Click **OK** twice, and then close the **System** dialog box.

Without Windows Update, your computer searches only in the driver store (see [second task in this guide](#)) and in the folders listed in the DriverPath registry entry (see [third task in this guide](#)), before prompting the user for media.

Note

Manual configuration of settings is useful only when managing a small number of computers. If you want to disable Windows Update search for device drivers, then use Group Policy. The **Turn off Windows Update device driver searching** policy can be found in Group Policy Management Console at: **Computer Configuration, Administrative Templates, System, Internet Communication Management, Internet Communication settings**.

Install the Windows Driver Kit

The tools used to digitally sign device driver packages -- MakeCert, Signability, and SignTool -- are part of the Windows Driver Kit (WDK). The sample device driver used for demonstration in this guide, Toaster, is also part of the WDK. If you do not already have a copy of the WDK installed, follow the steps in this procedure.

▶ **To install the WDK**

1. Log on to **DMI-Client1** as **DMI-Client1\TestAdmin**.
2. Browse to the local or shared network folder where you have a copy of the Windows Driver Kit installation files.
3. Double-click **Setup.exe**.
4. On the **Welcome to the Microsoft Windows Driver Kit Setup Wizard** page, click **Next**.
5. On the **End-User License Agreement** page, carefully read the license agreement in the text box. If you agree with the terms, select **I accept the terms in the License Agreement**, and then click **Next**.
6. On the **Custom Setup** page, click **Next**.
7. On the **Ready to Install** page, click **Install**.
8. When the installation is complete, click **Finish** to close the wizard.

Configure the Toaster sample device driver package for use in this guide

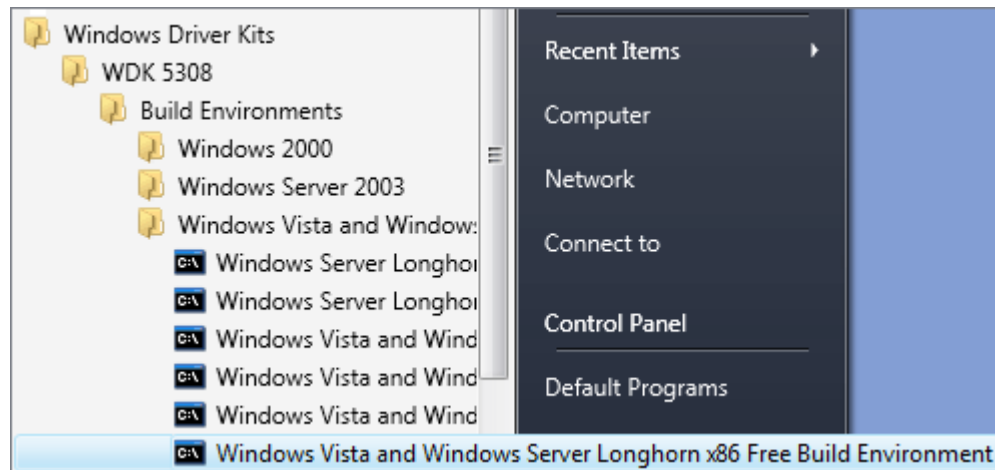
If you have access to the Windows Driver Kit, you can use the following procedure to configure the Toaster sample device drivers for use with this guide.

If you do not have access to the Windows Driver Kit, then you can use any device for which the device driver is not already present in the driver store. The driver package for such a device must already be signed.

Use this procedure to compile the sample device drivers, and to copy them to a folder in a way that resembles how a third-party commercial driver package is typically provided.

▶ **To configure the Toaster sample device driver package**

1. Log onto **DMI-Client1** as **DMI-Client1\TestAdmin**.
2. Click **Start**, and then click **All Programs**.
3. Click the following: **Windows Driver Kits**, **WDK *YourBuildNumber***, **Build Environment**, **Windows Vista and Longhorn**, and then **Windows Vista and Windows Server Longhorn x86 Free Build Environment**.



A command prompt window configured to build device drivers appears.

 **Note**

You cannot use a standard **Command Prompt** window. The **Build Environment** menu option configures the Path and other environment variables to specifically support the tools used for building device drivers.

4. Start Notepad by typing the following at the command prompt. You must still be in the `c:\winddk\YourBuildNumber` folder.

```
notepad copytoastfiles.cmd
```

5. In the confirmation dialog box, click **Yes** to create a new file.
6. Copy and paste the following text into the Notepad window:

```
REM -----START COPY HERE-----
@echo off
Echo Creating destination folder structure:
Md c:\toaster
Md c:\toaster\bus
Md c:\toaster\device
Md c:\toaster\device\i386
Echo Compiling the Bus device driver:
Cd .\src\general\toaster\bus
Build -cZ
Echo Compiling the Plug-in Utility:
Cd ..\exe\enum
Build -cZ
Echo Copying the device driver files to the destination folders:
Cd ..\..
Copy .\bus\objfre_wlh_x86\i386\busenum.sys c:\toaster\bus
Copy .\inf\i386\bus.inf c:\toaster\bus
```

```

Copy .\toastpkg\toastcd\toastpkg.inf c:\toaster\device
Copy .\toastpkg\toastcd\i386\toaster.sys c:\toaster\device\i386
Copy .\toastpkg\toastcd\i386\tostrcls.dll c:\toaster\device\i386
Copy .\exe\enum\objfre_wlh_x86\i386\enum.exe c:\toaster
Echo Toaster sample device driver is ready to use in c:\toaster
Cd ..\..\..
REM -----END COPY HERE-----

```

Note

This script creating the Toaster folder cannot work if you do not have write permissions for the C: drive root. If you logged on by using an administrator account, you have those permissions on a default installation of Windows. If you want to place the folder somewhere else, modify the script as needed.

7. Save the file, and then close Notepad.
8. In the **Build Environment** command window, run the .cmd file you just created.

```
copytoastfiles
```

Important

The file must be run at the command prompt in the folder specified, or else it cannot work.

Install the Toaster Bus device driver

There is no physical Toaster device that you plug in and unplug for this guide. Instead, the sample Toaster device is simulated by a device driver, and supported by the combination of a special bus driver and a tool. Like a USB bus, the Toaster Bus device driver starts the installation of a device driver when it detects the insertion of the sample Toaster device. The insertion of the Toaster device is simulated by the Enum.exe tool included with the Toaster sample package. Before you can simulate insertion and removal of the device by running Enum.exe, the Toaster Bus device driver must be installed.

To install the Toaster Bus device driver

1. At the **Build Environment** command prompt window, run the following command:


```
mmc devmgmt.msc
```
2. Right-click the top node that represents your computer, and then click **Add legacy hardware**.

3. On the **Welcome to the Add Hardware Wizard** page, click **Next**.
4. On the **The wizard can help you install other hardware** page, select **Install the hardware that I manually select from a list (Advanced)**, and then click **Next**.
5. In the list, scroll down and select **System devices**, and then click **Next**.
6. On the **Select the device driver you want to install for this hardware** page, click **Have Disk**.
7. In the text box, type **c:\toaster\bus**, and then click **OK**.
8. In the **Select the device driver you want to install for this hardware** dialog box, select **Toaster Bus Enumerator**, and then click **Next**.
9. On the **The wizard is ready to install your hardware** page, click **Next**.
The **Windows Security** dialog box appears, because there is not a valid digital signature for the device driver. Click **Install** to allow installation to proceed.
10. After installation completes, on the **Completing the Add Hardware Wizard** page, click **Finish**.
11. In **Device Manager**, double-click **System Devices** to expand the list.
12. Confirm that **Toaster Bus Enumerator** is in the list, and then close **Device Manager**.
13. Close the **Build Environment** command prompt window.

Steps for signing a device driver package

To sign a device driver package, you must have a code signing certificate. For more details about the various types of certificates that are available and how to acquire one, see the [Additional Resources](#) section at the end of this guide. This guide shows you how to create a certificate that you can use for testing purposes.

Steps outline: Signing a device driver package

1. [Create a digital certificate for signing](#)
2. [Add the certificate to the Trusted Root Certification Authorities store](#)
3. [Add the certificate to the Trusted Publishers store](#)
4. [Sign the device driver package with the certificate](#)

Step 1: Create a digital certificate for signing

In this step you create a certificate that can be used to sign the sample Toaster driver package.

First, open the Certificates MMC snap-in to see the current certificates.

Important

Do not run `certmgr.msc` to open the snap-in. By default, that opens the Current User version of the certificate stores. This procedure requires the certificates to be placed in the stores for the Computer Account instead.

To open the Certificates MMC snap-in

1. Click **Start**, click **Run**, and then in the Run box, type: `mmc`
2. In **Console Root**, click **File**, and then click **Add/Remove Snap-in**.
3. In **Add or Remove Snap-ins**, click **Certificates**, and then click **Add**.
4. In **Certificates snap-in**, click **Computer Account**, and then click **Next**.
5. On the **Select Computer** dialog box, select **Local computer: (the computer this console is running on)**, and then click **Finish**.
6. Click **OK** to close the **Add or Remove Snap-ins** page.

The Certificates snap-in appears in the console.

Now you can create the certificate.

Note

You cannot use the previous **Build Environment** command prompt window, because it was not running with the elevated permissions required by the `MakeCert` tool. If you attempt to run `MakeCert` without elevated permissions, it will fail with error code 0x5 (Access Denied).

To create a digital certificate by using the `MakeCert` tool

1. Open a **Build Environment** command prompt with elevated permissions, by right-clicking **Build Environment** on the **Start** menu, and then selecting **Run as administrator**.
2. At the **Build Environment** command prompt, type the following command on a single line (it appears here on multiple lines for clarity and to fit space limitations):

```
makecert -r -n "CN=MyCompany - for test use only"  
-ss MyCompanyCertStore
```

`-sr LocalMachine`

-r

Specifies that the certificate is to be "self-signed," rather than signed by a CA. Also called a "root" certificate.

-n "CN=MyCompany - for test use only"

Specifies the name associated with this new certificate. It is recommended that you use a certificate name that clearly identifies the certificate and its purpose.

-ss MyCompanyCertStore

Specifies the name of certificate store in which the new certificate is placed.

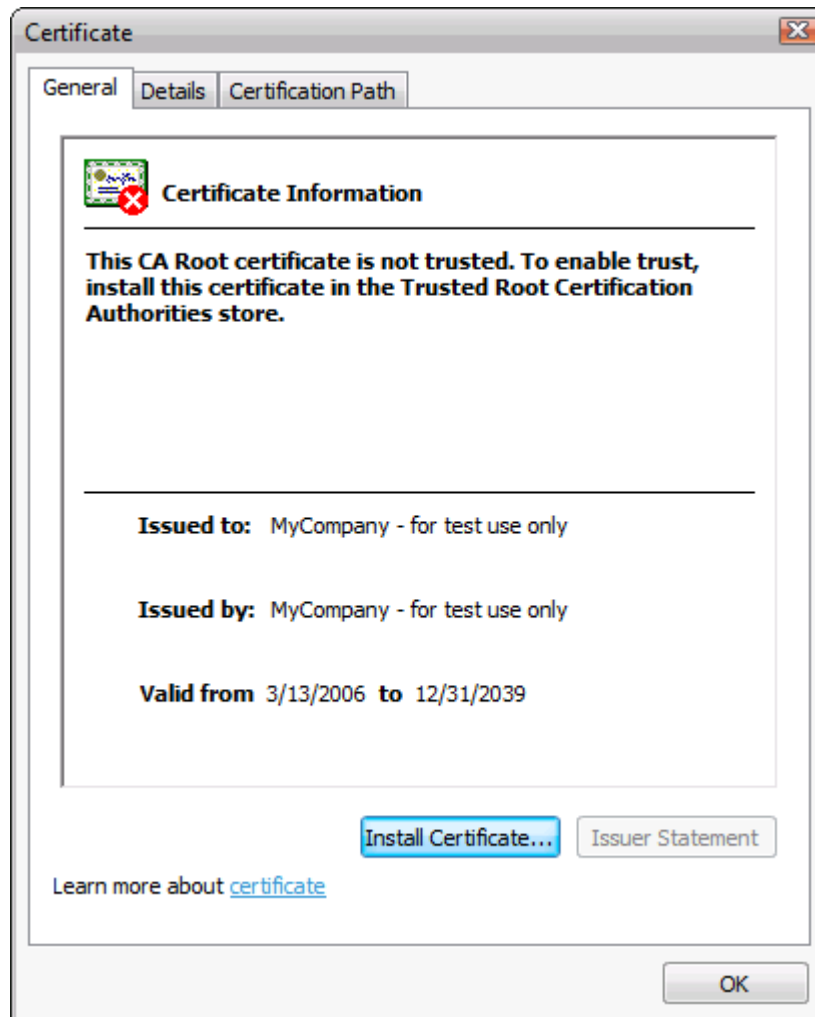
-sr LocalMachine

Specifies that the certificate store created by the `-ss` option is in the per computer store, instead of the default per user store.

The command returns the message "Succeeded" when the store and certificate are created.

3. Verify that your new certificate was created correctly. In the **Certificates** MMC snap-in that you opened earlier, open the node **Certificates (Local Computer)**, then **MyCompanyCertStore**, and then **Certificates**.
4. In the right-hand pane, double-click **MyCompany - for test use only**.

The certificate dialog appears showing your new certificate.



5. Click **OK** to close the **Certificate** page.

Step 2: Add the certificate to the Trusted Root Certification Authorities store

This step is required for locally created certificates, such as those created by using MakeCert, which are not directly traceable to a Trusted Root Certification Authority certificate.

By default, your new certificate is marked "Untrusted" because Windows cannot validate the certificate against any of the trusted certificates in the per computer Trusted Root

Certification Authorities store. In Windows Vista and Windows Server "Longhorn", all certificates must be traceable to a certificate in this store to be considered valid.

This step is not required for commercial certificates created for you by a certification authority because they already have their root certificate in the per computer Trusted Root Certification Authorities store.

 **Note**

Certificates that are placed in the per user Trusted Root Certification Authorities store will not validate signatures of device driver packages.

 **To add the test certificate to the Trusted Root CA certificate store**

1. In the Certificates snap-in, right-click **MyCompany - for test use only**, and then click **Copy**.
2. Right-click **Trusted Root Certification Authorities**, and then click **Paste**.
3. Open **Trusted Root Certification Authorities** and **Certificates**, and then double-click your certificate.
4. Confirm that the "Untrusted" message no longer appears, and then click **OK** to close the certificate.

Step 3: Add the certificate to the per machine Trusted Publishers store

To use your new certificate to confirm the valid signing of device drivers, it must also be installed in the per computer Trusted Publishers store.

 **Note**

Certificates that are placed in the per user Trusted Publishers store cannot validate signatures of device driver packages.

 **To add the test certificate to the Trusted Publishers certificate store**

1. In the Certificates snap-in, right-click your certificate, and then click **Copy**.
2. Right-click **Trusted Publishers**, and then click **Paste**.
3. Open **Trusted Publishers** and **Certificates**, and then confirm that a copy of your certificate is in the folder.
4. Click **OK** to close the certificate.

Step 4: Sign the device driver package with the certificate

If you are using the sample Toaster device and driver -- or if your organization wants to implement a policy where all device drivers must be signed by your organization's own certificate -- then follow these steps to replace the existing signature with your own.

If you are using a driver package that has already been signed by the vendor, then your driver package already has a useful catalog file that is referenced by the .inf file. In this case, you can skip the first two steps below, and begin with [Sign the catalog file using Signtool](#).

To sign the device driver, you need to do the following:

1. [Prepare the driver package .inf file](#)
2. [Create a catalog file for the driver package](#)
3. [Sign the catalog file by using Signtool](#)

Prepare the driver package .inf file

The .inf file controls the installation of the driver package. The digital signature for a device driver package resides in a catalog file, with a .cat file name extension. Ensure that the .inf file used to install the driver package includes a reference to the .cat file.

In addition, for the sample Toaster device driver used in this guide, you must also change the timestamp and version number of the device driver.

A co-installer is code provided by the device driver manufacturer that can be invoked during the driver package installation process. It gives the installation program more flexibility in what can be done during the installation process. In the sample Toaster device driver, the co-installer displays optional programs that the user can install. You do not need the Toaster co-installer for these scenarios, so in this procedure you delete it from the .inf file.

Note

If your driver package has already been signed by the vendor, then the .inf file already has a reference to a valid catalog file, and you can skip this procedure.

To prepare the driver package .inf file

1. At the **Build Environment** command prompt with elevated permissions, change to the folder that contains your driver package. Type the following command:

```
cd \toaster\device
```

2. Then type the command:

```
Notepad toastpkg.inf
```

Notepad opens with the .inf file displayed.

3. Find the **[Version]** section. The original file includes the lines:

```
CatalogFile.NTx86 = tostx86.cat  
CatalogFile.NTIA64 = tostia64.cat  
CatalogFile.NTAMD64 = tstamd64.cat
```

4. Delete those three lines, and replace them with following single line:

```
CatalogFile = toaster.cat
```

5. In the **[Version]** section, find the line that begins with **DriverVer=**. Replace the date and version number so that the line appears as follows:

```
DriverVer=04/01/2006,9.9.9.9
```

6. In the **[Toaster_Device.NT.CoInstallers]** section, find and delete these three lines:

```
[Toaster_Device.NT.CoInstallers]  
AddReg=CoInstaller_AddReg  
CopyFiles=CoInstaller_CopyFiles
```

7. Save your changes, and then close Notepad.

Create a catalog file for the driver package

Next, run the Signability tool to create an unsigned catalog file for the sample Toaster driver package. Signability parses the driver package .inf file, and then generates unique hashes for every file referenced in the .inf file. The recipient of the package uses the hashes to confirm that the files received are exactly the same as those that were signed.

If the driver package you are using was signed by the vendor, then a catalog file already exists, and you do not need to create a new one. Skip this procedure, and go to the next procedure [Sign the catalog by using SignTool](#) to replace the vendor's signature with your own.

Note

The Signability tool must be run at a command prompt with elevated permissions.

To create a catalog file for the driver package

1. At the **Build Environment** command prompt with elevated permissions, type the

following command:

```
signability /driver:c:\toaster\device /os:256 /auto /cat
```

```
/driver:c:\toaster\device
```

Specifies the location of the .inf file for the driver package. You must specify the complete folder path. A '.' character will not work here to represent the current folder.

```
/os:256
```

Identifies the 32-bit version of Windows Vista as the operating system. Run the command **signability /?** for a complete list of supported operating systems and their codes.

```
/auto
```

Specifies the operation is to be completed without any user interaction.

```
/cat
```

Specifies the tool is to create a catalog file as specified in the .inf file.

2. Review the output of the Signability tool. Note that a warning is generated for each file. The warnings indicate that until you sign the catalog file, it is not trusted.

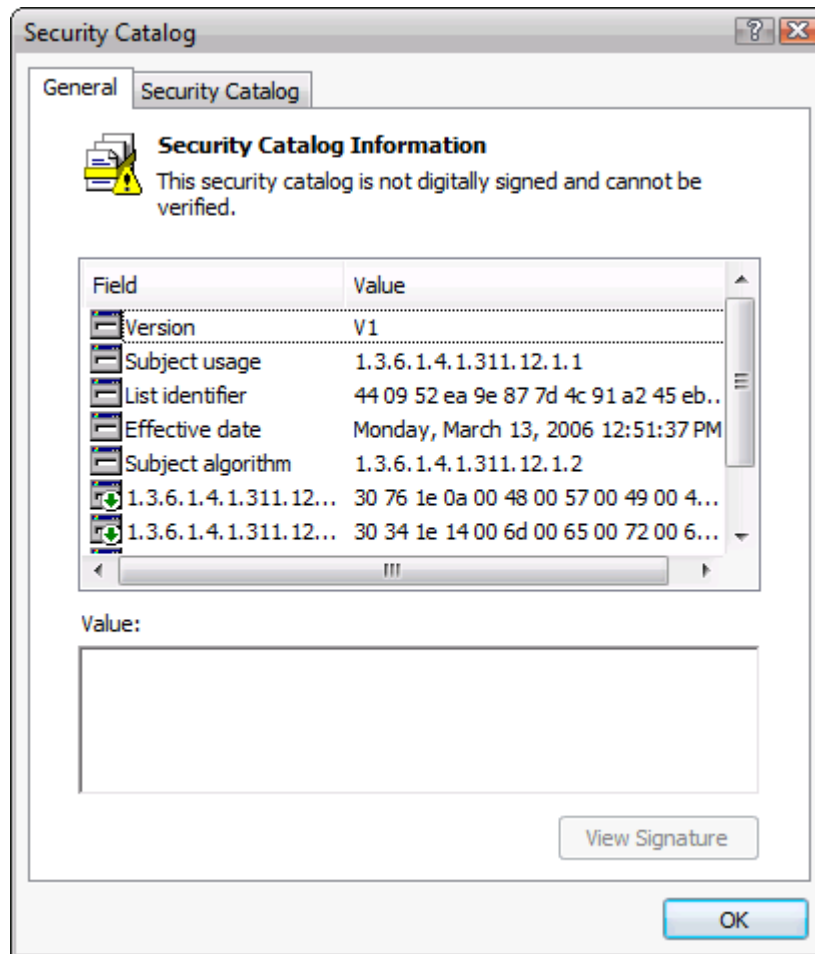
```
Signability v2.19 (engine v01.01.0005)
=====
Test Results for c:\toaster\device
Testing versus the following operating systems:
    Windows Vista (32-bit)
Final result:  Test passed with warnings.  (Details below.)
Warnings:
WARN:  \toastpkg.inf is not represented by a signed catalog file.
WARN:  \i386\toaster.sys is not represented by a signed catalog file.
WARN:  \i386\tostrcls.dll is not represented by a signed catalog file.
Catalog files successfully generated.
```

When you are done reviewing the results, close the Notepad window.

3. Review the completed .cat file, which is in the c:\toaster\driver folder. At the command prompt, type:

```
start toaster.cat
```

The **Security Catalog** dialog box appears, indicating that the catalog is not digitally signed. Because the .cat file is not signed, the **View Signature** button is disabled.



4. Click the **Security Catalog** tab. There are three entries in the **Catalog entries** section, one each for the .inf file, the .sys file, and the .dll file of the driver package. Click each entry, and note in the Entry Details section that each file in the package has an entry, along with a "thumbprint" (the hash) that can be used to confirm the integrity of the file.
5. Click **OK** to close the **Security Catalog** dialog box.

Sign the catalog file by using SignTool

Now that you have a catalog file, you can sign it by using the SignTool tool.

Use this procedure regardless of whether you are using the sample Toaster device driver or not.

 **Important**

When signing a driver package, you should always use the option to timestamp the signature. This timestamp specifies when the signature was created. If a certificate expires or is revoked for security reasons, then only signatures created before the expiration or revocation are valid. If a timestamp is not included in the signature, then Windows cannot determine if the package was signed before or after the expiration or revocation, and will reject the signature.

 **To sign a catalog file using SignTool**

1. At the **Build Environment** command prompt with elevated permissions, type the following command all on one line. It appears here on multiple lines for clarity and to fit space limitations:

```
SignTool sign /s MyCompanyCertStore /n MyCompany  
/t http://timestamp.verisign.com/scripts/timestamp.dll  
toaster.cat
```

/s MyCompanyCertStore

Specifies the name of the certificate store in which SignTool searches for the certificate specified by the parameter `/n`.

/n MyCompany

Specifies the name of the certificate to be used to sign the package. You must include enough of the name to allow SignTool to distinguish it from others in the store. If this name includes spaces, then you must surround the name with double quotes.

/t path to time stamping service

Specifies the path to a time stamping service at an approved certification authority. If you purchase your certificate from a commercial vendor, they should provide you with the appropriate path to their service.

toaster.cat

Specifies the path and file name of the catalog file to be signed.

Signtool indicates completion with the following message:

```
Successfully signed and timestamped: toaster.cat
```

2. To view and verify your signed catalog file, at the command prompt, type:

```
start toaster.cat
```

3. Make sure that the header of the **Security Catalog** property page now states that the security catalog is "Trusted," and that the **View Signature** button is enabled.
4. Click **View Signature**, and then confirm the details of the signature you added to the package. No other details of the catalog file have changed.

Steps for staging a device driver package in the driver store

Staging a device driver package in the driver store on the client computer ensures the smoothest user experience. After the signed driver package is in the driver store, Windows considers the package trusted. As long as you do not have a device installation restriction policy in effect for a specific device, the user can simply plug in the device and Windows silently installs the device driver.

Windows includes a tool called PnPUtil that you can use to manage the driver store, including adding driver packages, removing driver packages, and listing the driver packages that are in the store.

Important

You can only run the PnPUtil tool from a command prompt that is running with elevated permissions. The tool cannot invoke the **User Account Control** dialog box. If you attempt to use the PnPUtil tool to add or remove packages from a command prompt that is not running as administrator, the command will fail.

Steps Outline: staging a device driver package in the driver store

[Step 1: Attempt to stage an unsigned driver package](#)

[Step 2: Attempt to stage a signed, but improperly modified driver package](#)

[Step 3: Attempt to stage the properly signed driver package.](#)

[Step 4: Test installation of the staged driver package.](#)

Step 1: Attempt to stage an unsigned driver package

Windows interrupts an attempt to install an improperly signed driver package.

▶ **To attempt staging of an unsigned driver package**

1. At the **Build Environment** command prompt with elevated permissions, temporarily rename the .cat file to effectively remove the signature from the driver package. Type the following command:

```
ren toaster.cat toaster.nosig
```

2. Attempt to stage the unsigned package. At the command prompt running with elevated permissions, type the command:

```
pnputil.exe -a toastpkg.inf
```

The **Windows Security** dialog box appears because the .inf file is not signed. Windows cannot match it against the certificates that are trusted by the computer.

3. Click **Don't Install**.

The PnPUtil tool indicates that the staging operation failed:

```
Adding the driver package failed : A file could not be verified because it  
does not have an associated catalog signed via Authenticode(tm).  
Adding at least one driver package failed!
```

4. Rename the catalog file back to its correct name. At the command prompt, type:

```
Ren toaster.nosig toaster.cat
```

Step 2: Attempt to stage an signed, but improperly modified driver package

Windows will also interrupt an attempt to install a driver package that has been modified after it was signed. Because the signature includes thumbprints for each file, making a change to any of the files in the package causes the validity check for the signature to fail.

▶ **To attempt staging a signed, but modified driver package**

1. Save a copy of the correct toastpkg.inf file. At the command prompt type:

```
Copy toastpkg.inf toastpkg.orig
```

2. Modify toastpkg.inf so that its thumbprint is no longer valid. Open it in Notepad:

```
notepad toastpkg.inf
```

3. With the cursor at the very beginning of the file, press **Enter** to add a blank line,

and then save your changes and close Notepad.

4. Attempt to stage the modified package. At the command prompt, type:

```
pnputil.exe -a toastpkg.inf
```

Because the package was modified after being signed, the **Windows Security** dialog box appears, warning you that the signature is invalid.

5. Click **Don't Install**.
6. Overwrite the modified .inf with the original. At the command prompt, type:

```
Copy /y toastkg.orig toastpkg.inf
```

Step 3: Attempt to stage the properly signed driver package

To attempt staging a properly signed package

1. Attempt to stage the package. At the command prompt, type:

```
pnputil.exe -a toastpkg.inf
```

Because the signature attached to the package is valid, the files are unmodified, and the file thumbprints match the signature, Windows successfully stages the package, with no prompts. The output includes the published name with the OEM number that you can use to remove the driver package from the store later, if needed. Make note of the number assigned to your package.

```
Processing inf : toastpkg.inf  
Driver Package added successfully.  
Published name : oem4.inf
```

Note

The number assigned to your package might be different due to the number of driver packages that are already installed on your computer.

You can view the package in the store by running the PnPUtil tool with the `-e` (for 'enumerate') parameter.

To examine the package in the driver store

1. At the command prompt, type:

```
pnputil.exe -e
```

2. Look for the package with your `OEM###` listed in the output. Make note of this

number because you might need it later. You can also see the version number and date that you entered in the .inf file.

```
Published name : oem4.inf
Driver package provider : Toast'R'Us
Class : Unknown driver class
Driver version and date : 04/01/2006 9.9.9.9
Signer name : MyCompany - for test use only
```

Step 4: Test installation of the package

At this point, the driver package is now in the driver store. The driver package was staged by an account that has the required administrative rights, and Windows has checked the validity of the digital signature, so the device driver can be installed by a standard user by simply attaching the device.

Note

In this procedure, you run the Enum.exe tool as an administrator, even though Windows can install a device driver from the store as a standard user. The elevated permissions are required because of the simulation of the hardware in software, not because of the device driver installation process. If you follow these procedures with a real hardware device, using a vendor-provided device driver, you do not need to be logged on as an administrator when inserting the device.

To test installation of the staged package

1. Log off, and then log on as **DMI-Client1\TestUser**.
2. Open a command prompt with administrator rights. Click **Start** and **All Programs**, and then click **Accessories**. Right-click **Command Prompt**, and then click **Run as administrator**.
3. On the **User Account Control** page, you are asked to specify an administrator account and its password. Select **DMI-Client1\TestAdmin** and enter its password.

The command prompt opens.

4. Start **Device Manager** so you can view the installed device. At the command prompt with elevated permissions, type the following command:

```
mmc devmgmt.msc
```

5. Rearrange the windows so you can use the command prompt while still seeing the contents of **Device Manager**.

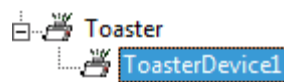
6. Change to the c:\toaster folder. At the command prompt, type the following command:

```
cd \toaster
```

7. Run the Enum.exe tool, which simulates plugging in a Toaster device. At the command prompt, type:

```
Enum -p 1
```

After the device driver finishes installing, the device appears in **Device Manager**.



Note

Do not attempt to uninstall the device driver until instructed to do so in the following procedure.

Steps for configuring a shared network folder to hold signed device driver packages

You might not want to stage every driver package that is approved for use. Instead, you can place the signed driver packages on a shared network folder and configure your client computers to search that folder whenever a new device is plugged into the computer.

A driver package hosted in a shared network folder must be properly signed with a certificate that is installed on the client computer. A driver package must still be staged in the driver store before it can be installed. Because a standard user cannot stage a driver package by default, a driver installation computer policy can be applied to allow the specific device driver to be installed without elevated permissions. For more information about the policies to allow and restrict installation of specific device setup classes, see the [Additional Resources](#) section at the end of this guide.

Important

For simplicity, this guide uses a local folder to demonstrate the use of the DevicePath registry entry. In a production environment, use a shared network folder to which all of your users have read permissions.

Steps outline: Configure a shared network folder to hold signed device driver packages

[Step 1: Create the folder to contain device driver packages](#)

[Step 2: Configure the client computer to search the additional folder for driver packages](#)

[Step 3: Configure the client computer to allow standard users to install the device](#)

[Step 4: Remove the device driver and driver package installed in the previous procedure](#)

[Step 5: Attempt installation of the device driver package from the other folder](#)

Step 1: Create the folder to contain device driver packages

With Windows Vista and Windows Server "Longhorn", you can configure the client computers to search additional folders for driver packages that are not found in the driver store. If a driver package is found in an additional folder, then Windows will not prompt the user for media for the device being installed.

Even after placing a driver package in a shared network folder, you must still have the ability to place a driver package in the driver store. You must also still approve the signature if the package's certificate is not in the Trusted Publishers certificate store.

In this procedure, you create a folder on DMI-Client1, and then copy the signed device driver package to the folder.

▶ To create a folder to contain device driver packages

1. Log off, and then log back in as **DMI-Client1\TestAdmin**.
2. Open a command prompt by clicking **Start, All Programs, Accessories**, and then **Command Prompt**.
3. Create a new folder. At the command prompt, type:

```
md c:\drivershare
```
4. At a command prompt, type the following command to place a copy of your signed driver package on the folder:

```
xcopy /s c:\toaster\device c:\drivershare
```

Step 2: Configure the client computer to search the folder for driver packages

Windows Vista and Windows Server "Longhorn" support a Registry setting that allows you to specify additional folders that Windows searches for a driver package for newly detected hardware. By default this value specifies only the folder **%SystemRoot%\Inf**. You can add other folders to this value, separated by semicolons, to make Windows search additional folders. These other locations can be local folders, or specified with a network path, such as `\\servername\sharename`.

▶ To configure the client computer to search the added folder for driver packages

1. At the command prompt type:

```
RegEdit
```



Caution

Incorrectly editing the registry can severely damage your system. Before making changes to the registry, back up any valued data on the computer.

2. In Registry Editor, navigate to:

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion

3. In the details pane, double-click **DevicePath**.
4. Add a semi-colon to end of the existing text, and then add the path to your folder. The result should be similar to:

```
%SystemRoot%\inf;c:\drivershare
```



Important

Do not remove the `%SystemRoot%\inf` file path from the DevicePath registry entry.

5. Click **OK** to save the new value, and then minimize, but do not close **Registry Editor**. You will use **Registry Editor** again later.

Step 3: Configure the client computer to allow standard users to install the device

By default, a standard user cannot place a driver package in the driver store. However, in Windows Vista and Windows Server "Longhorn" you can use a computer policy together

with the globally unique ID (GUID) of the device's setup class to allow standard users to stage the device driver package.

 **Note**

The procedures shown here work well on a single computer, but do not scale well to a large number of computers. To apply computer configuration to a large number of managed systems, use Group Policy and Active Directory. For more information about Group Policy and Active Directory, see the [Additional Resources](#) section at the end of this guide.

The device setup class GUID can be found in two places, the driver package .inf file, and the Device Properties dialog box for a currently installed device.

 **To find the GUID for a device setup class in the driver package .inf file**

1. Open the .inf file by using Notepad. At the command prompt, type:

```
Notepad c:\toaster\device\toastpkg.inf
```

2. In the **[Version]** section, find the line that begins with **ClassGuid=**, and make note of the value. For the sample Toaster device it looks like:

```
ClassGuid={B85B7C50-6A01-11D2-B841-00C04FAD5171}
```

3. Select and right-click the GUID value, including the { } characters, and then click **Copy**.
4. Close Notepad, ensuring that you do not save any changes.

Alternatively, if you have a computer with the device already installed and operational, you can see the GUID as part of the device properties.

 **To find the GUID for a device setup class in the device properties page**

1. In Device Manager, find and right-click the **Toaster Package Sample Toaster** device, and then click **Properties**.
2. On the **Toaster Package Sample Toaster Properties** dialog box, click the **Details** tab.
3. In the **Property** list, select **Device class guid**.
4. Make note of the value. It is the same value that you saw in the .inf file.
5. Right-click the GUID, and then select **Copy**.
6. Click **OK** to close the device properties page.

Now that you have the GUID that applies to the device you want to install, you can add it to the list in the computer policy that specifies which devices can be installed by standard users.

▶ **To configure the computer to allow standard users to install devices that have a specified device setup class**

1. At the command prompt, type:

```
Mmc gpedit.msc
```

2. In the navigation pane of the **Group Policy Object Editor**, navigate to **Computer Configuration/Administrative Templates/System/Driver Installation**.
3. In the right-hand pane, double-click the policy **Allow non-administrators to install devices for these device classes**.
4. In the policy dialog box, select **Enabled**, and then click **Show**.
5. In the **Show Contents** dialog box, click **Add**.
6. In the **Add Item** text box, right-click and select **Paste** to insert the GUID.
{B85B7C50-6A01-11D2-B841-00C04FAD5171}
7. Click **OK** three times to close the dialog boxes and return to the Policy Editor.
8. Close the **Group Policy Object Editor**.
9. At the **Build Environment** command prompt with elevated permissions, apply the policy to your current session by typing:

```
gpupdate /force
```

📌 **Note**

GPUpdate cannot display the User Account Control dialog box to request administrative credentials, so make sure you run it with administrator rights.

Step 4: Remove the device driver and driver package installed in the previous procedure

Before you can install the device driver from the additional folder, you must first uninstall the current device driver and remove its driver package from the driver store.

 **Note**

You need to remove the previously installed packages only because this guide is demonstrating an additional way to install a driver package.

 **To uninstall the currently installed device**

1. In Device Manager, right-click on the **Toaster Package Sample Toaster** device entry, and then click **Uninstall**.
2. In the **Confirm Device Removal** dialog box, click **OK**.

The device disappears from the Device Manager window.

3. Run the Enum.exe tool that simulates unplugging the Toaster device. At the command prompt with elevated permissions, type:

```
Enum -u 1
```

The device is unplugged, and the device driver removed from memory.

4. At the command prompt, type the command to remove the driver package from the driver store:

```
pnputil.exe -d oem##.inf
```

In this command, ## is the number you noted in an earlier procedure. If you do not remember the number, run `pnputil -e`, and then look for the Toaster device in the output list.

The package is deleted from the driver store.

5. Run the command `pnputil.exe -e` again to verify that the package is deleted.

Step 5: Attempt installation of the device driver package.

Now that the driver package is in the folder, and the client computer is configured to search there for driver packages when new devices are plugged in, you can install the device.

 **To install the driver package from the network share**

1. Log off, and then log on as **DMI-Client1\TestUser**.
2. Open a command prompt with administrator rights. Click **Start, All Programs, and Accessories**. Right-click **Command Prompt**, and then click **Run as administrator**.

3. On the **User Account Control** page, you are asked to specify an administrator account and its password. Select **DMI-Client1\TestAdmin**, and then enter its password.

The command prompt opens.

4. Start **Device Manager** so you can view the installed device. At the command prompt with elevated permissions, type the following command:

```
mmc devmgmt.msc
```

5. Rearrange the program windows so you can use the command prompt while still seeing the contents of **Device Manager**.

6. Change to the `c:\toaster` folder. At the command prompt, type the following command:

```
cd \toaster
```

7. Run the Enum.exe tool that simulates plugging in a Toaster device. At the command prompt, type:

```
Enum -p 1
```

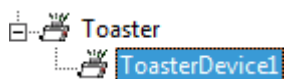
Windows will start the installation process. A new appears in **Device Manager** in the **Other devices** section.

8. In the **Found New Hardware** dialog box, click **Locate and install driver software (recommended)**.

Because it can't find the driver package in the driver store, Windows searches the folders identified in the **DevicePath** registry entry, and finds the driver package in the folder.

9. It might take a few moments to complete the staging of the driver package, and the subsequent installation. If you click the device installation icon, a message appears indicating that drivers are installing, followed by a message that states: **Toaster Package Sample Toaster installed**.

Because the computer policy allows a standard user to place the driver package for this class in the driver store, and because the package is properly signed by a trusted publisher, the installation of the driver package completes with no further user interruptions. The **Unknown Device** entry is replaced by the **Toaster** entry.



Conclusion

In this guide you used a sample device and driver in a lab environment to learn how to securely deliver device driver packages to client computers. With this configuration, a standard user can install device drivers without any assistance from an administrator.

The tasks used to complete this configuration included how to:

- Sign a device driver package to allow Windows to trust the driver package. This task included procedures for creating a signing certificate, configuring the client computers to recognize the certificate, creating a catalog file to contain the signature, and then signing the catalog file and including it in the driver package.
- Stage the driver package in the driver store on the client computer. This task included procedures that showed you how to use the PnPUtil.exe tool to place driver packages in the driver store as an administrator, so that they can be installed by any user.
- Configure a client computer to search additional folders for driver packages when the computer does not find them in the driver store. These procedures demonstrated modifying a Registry entry to add a local folder or network location to the list of folders Windows searches for driver packages when it detects a new hardware device. This eliminates the need for the user to enter the path manually, or to provide media. The procedures also demonstrated how to configure computer policy to allow a standard user to successfully stage, and thus install, devices that are members of approved device setup classes.

Logging bugs and feedback

Your feedback is welcome. If the scenarios included do not work as described or if they fail to capture the way you want to use the technology, please tell us. We will use the feedback that you provide to improve the quality of this documentation. Send your comments on this documentation to [Vista Feedback](mailto:vistafb@microsoft.com) (vistafb@microsoft.com).

For product feedback, please use [Contact Us](#) link at the bottom of the Windows Vista Web page at <http://go.microsoft.com/fwlink/?linkid=65454>.

Additional Resources

For more information about device installation:

- Device Management and Installation

<http://go.microsoft.com/fwlink/?LinkId=59274>

- How Setup Selects Device Drivers
<http://go.microsoft.com/fwlink/?LinkId=54881>
- Device Identification Strings
<http://go.microsoft.com/fwlink/?LinkId=52665>
- Step-By-Step Guide to Controlling Device Installation Using Group Policy
<http://go.microsoft.com/fwlink/?linkid=63416>

For more information about User Account Control in Windows Vista:

- User Account Control
<http://go.microsoft.com/fwlink/?LinkId=68249>

For more information about digital certificates and digital signatures:

- Code Signing Best Practices
<http://go.microsoft.com/fwlink/?LinkId=68250>
- Code Signing Requirements for 64-bit Kernel Mode Drivers
<http://go.microsoft.com/fwlink/?LinkId=66262>
- Introduction to Code Signing
<http://go.microsoft.com/fwlink/?LinkId=59273>
- About Cryptography
<http://go.microsoft.com/fwlink/?LinkId=59544>
- Creating, Viewing and Managing Certificates
<http://go.microsoft.com/fwlink/?LinkId=59275>
- Microsoft Root Certificate Program Member List
<http://go.microsoft.com/fwlink/?LinkId=59547>
- Windows Server PKI Operations Guide
<http://go.microsoft.com/fwlink/?LinkId=59548>

Building an Enterprise Root Certification Authority in Small and Medium Business

<http://go.microsoft.com/fwlink/?LinkId=59549>

For more information about Group Policy:

- Group Policy

<http://go.microsoft.com/fwlink/?LinkId=55625>