

Driver Setup

Reference Manual

Version: 2.3.0
Date: 23. August 2013

Author: René Möller
Thesycon Systemsoftware & Consulting GmbH
Werner-von-Siemens-Str. 2
D-98693 Ilmenau
Germany
+49 3677 8462 0
<http://www.thesycon.de>

Table of contents

1	Signing	3
2	Customization	3
2.1	Package for delivery	4
2.2	Customization of the driver package	4
2.2.1	Rename drivers (.sys).....	4
2.2.2	Customize installation files (.inf).....	4
2.2.3	Provide catalog files (.cat)	5
2.3	Customization of the setup.ini file	5
3	Silent mode	9
3.1	Silent Installation	10
3.1.1	Command Line Parameters.....	10
3.1.2	Exit codes.....	10
3.2	Silent Uninstallation.....	12
3.2.1	Command Line Parameters.....	12
3.2.2	Exit codes.....	13

1 Signing

Signing of driver packages is optional for older Windows versions but required for newer ones, especially for 64-bit Windows starting with Vista.

To sign a driver package, a catalogue file (.cat) must be created in the first step. This catalogue file contains a hash value of each file belonging to the driver package. In the next step, the catalogue file is signed by means of a certificate. Either the signature is obtained from a WHQL submission (i.e. a submission to the Windows Hardware Quality Labs program of Microsoft) or by means of a vendor certificate (Authenticode). A WHQL submission is a more expensive and time-consuming process while a vendor signature can be created within seconds after you purchased a valid certificate from a certificate authority (CA), e.g. VeriSign. For details you may refer to <http://www.symantec.com/verisign/code-signing/microsoft-authenticode>. Alternatively, search the Web for "VeriSign Code Signing Certificates for Microsoft Authenticode".

Note: Vendor signatures do not have any noteworthy impact on driver installations before Windows Vista. To enjoy the advantages of a signed driver package on Windows XP and corresponding server systems, a WHQL signature must instead be used.

A signature may be required for the following reason:

- Windows setup dialogs can be suppressed, i.e. it is possible to install a driver completely silent (with administrator privileges).
- Windows Vista 64-bit and later 64-bit operating systems require signed driver packages (or at least signed driver files).
- Windows provides built-in drivers that may be compatible in some cases with the device to install. Built-in drivers are always WHQL signed and the operating system prefers signed drivers when selecting the appropriate driver for a device. So your driver package should be signed in this case.
Note: Before Windows 7 WHQL signed drivers are also preferred over vendor-signed drivers according to the default system settings.

The driver package you received from Thesycon contains a set of scripts that generate setup packages signed by means of your vendor certificate. To use these scripts the following steps must be executed:

- Install the "SignTools" package you can get from Thesycon on request. It includes all Microsoft tools required to digitally sign files.
- Perform the steps described in the script "signing\set_vendor_certificate.cmd" that is contained in driver package.
- Run the script "setup\create_and_sign_setup_packages.cmd". It recreates the setup packages located in the folders "_disk" and "_disk_debug" by means of your vendor certificate.

2 Customization

The setup package consists of the setup program 'setup.exe', the corresponding configuration file 'setup.ini', the files of the driver package to install and an optional certificate file of the vendor that contains the public key used to sign the driver package. The setup program is not modified during the customization process. All required changes are performed with the parameters in the .ini file and the files of the driver package.

Note: Customization of the setup package is usually not necessary if it is already prepared for installation of the desired product. To adapt it for installation of another product customization is mandatory in order to avoid potential conflicts with other own products or products of other vendors.

2.1 Package for delivery

The complete setup package that is delivered to the end user consists of the following files:

- setup.exe - The setup program. The setup program supports x86 and x64 systems.
- setup.ini - The configuration file for the setup program. Do not rename it.
- The driver files (.sys) for x86 and x64 systems.
- The INF files (.inf) for x86 and x64 systems.
- The catalogue files (.cat) with a digital signature for x86 and x64 systems. Starting with Windows Vista catalogue files are at least required for x64 systems.
- The optional public key (.cer) of the Authenticode certificate used to sign the driver package.

These files should be stored in a folder. Typically a surrounding software installer, that is not part of this package, should extract these files to the hard disk. Alternatively the files can be stored in a folder on a CD ROM or the like.

2.2 Customization of the driver package

2.2.1 Rename drivers (.sys)

Rename each file of type .sys to any name of your choice. The name should be world-wide unique. Keep the extension 'sys'.

2.2.2 Customize installation files (.inf)

The .inf file contains references to the driver and the service. Rename the strings that belong to the driver and the service in the same way as the driver itself. Modify the hardware IDs to fit your product. Rename the strings that describing the product, the vendor and the installation disc. Most of the strings can be found in the Strings section at the end of the .inf file. In some sections like the 'copy files' section the string replacement is not working. For that reason the file names must be replaced manually.

2.2.3 Provide catalog files (.cat)

Provide your own catalog files for your driver package. You can achieve this by obtaining a WHQL signature for your driver package from Microsoft or by creating your own catalog files and sign them with your vendor certificate (Authenticode). Please refer to the section “Signing” for further details.

2.3 Customization of the setup.ini file

The INI file is a text based configuration file. It can be edited with a text editor like Notepad. The INI file is divided into sections. A section is marked with []. Do not change the names of the sections. In each section a set of parameters is defined. Each parameter has a name and a value. Do not modify the names. Change the parameters accordingly to your requirements.

[Setup]

This section contains general keys that are used on x86 and x64 systems.

CompanyName

Company name that is shown in the programs control panel of the system, which is the 'Programs and Features' control panel on Windows Vista and later and the 'Add or Remove Programs' control panel before Windows Vista.

Example: CompanyName=My Company Name

ProductName

Product name shown in dialogs.

Note: The setup automatically appends the product version as well as strings like 'Setup' and 'Uninstall' depending on the purpose of the dialogs. So don't use these strings as part of the name!

ProductName=My Product Name

ProgramsAndFeaturesProductName

Product name shown in the programs control panel of the system.

If empty the value of 'ProductName' is used instead. The name has to be unique at least for all products of the same company.

Note: The setup automatically appends the product version. So the name should not contain any version information.

Example: ProgramsAndFeaturesProductName=My Product Name

ProductVersion

Product version shown in dialogs.

Example: ProductVersion=1.6.0.0

SetupRegPath

Registry path of the setup application located under the main key HKLM\Software.

Note: This parameter must not change in future releases of your setup package.

Otherwise the setup cannot detect whether or not another version already exists on the target system and cannot uninstall the existing version before the current package will be installed.

Note furthermore that the path must be world-wide unique. To achieve this use your company name and your product name for the path.

Example: SetupRegPath=MyCompanyName\MyProductName

DefaultInstDir

Default installation directory on the user's system.

The path is relative to 'Program Files'.

Note: The path must be world wide unique. To achieve this use your company name and your product name for the path.

Example: DefaultInstDir=MyCompanyName\MyProductName

CertificateFile

Name of the file containing the public (!) key of the vendor certificate that has been used to sign the driver package.

To create the file just export the vendor certificate from the certificate store. Choose the export format 'DER encoded binary X.509 (.CER)'. Do not export the private key!

The installer uses this file only to suppress system setup dialogs that will ask the user whether or not to trust the driver publisher. For this purpose the certificate is temporarily installed in the 'Trusted Publisher' store. After installation has completed the certificate is not required anymore and will be removed from the certificate store. This feature is especially useful when running the setup in silent mode.

Note: It is rare that the root certificate belonging to the vendor certificate is not yet present on the target system. In this case, it is automatically loaded in the background from a Microsoft server. Unfortunately this doesn't always happen during installation on Windows Vista. So the system may warn the user not to install an unsigned driver from an unknown vendor - even though the driver is signed and the vendor certificate is installed in the certificate store. The installation will succeed and the driver will be loaded anyway, but because of the system warning the user may decide to abort the installation. For this reason the installer explicitly requests the system to load the corresponding root certificate if required. Note that this requires a connection to the internet. Administrators that don't allow an online connection of their target systems can download specific root certificate packages from a dedicated Microsoft server (e.g. <http://catalog.update.microsoft.com>).

For further information please search the internet for 'root update' or 'KB931125'.

The value is optional but because of the reasons given above it is highly recommended.

Note: This parameter is ignored for operating systems before Windows Vista.

Example: CertificateFile=MyCertificate.cer

[Setup_x86]

Settings in this section will only be applied to 32-bit operating systems.

CopyFiles

Files of the setup package to be copied to the installation directory (e.g. .sys and .cat files). Several files may be specified as a comma separated list of file names without a path. If the parameter SubDir of this section is empty the files must be located in the same directory like the setup application, otherwise they must be located in the given sub-folder.

Note that files specified by other parameters of this section are implicitly copied and should not be specified by this parameter.

Example: CopyFiles=MyMiDrv.sys, MyMiDrv.cat, MyComDrv.sys, MyComDrv.cat

BusDriverInf

Name of the .inf file provided for the bus driver. If the parameter SubDir of this section is empty the file must be located in the same directory like the setup application, otherwise it must be located in the given sub-folder.

Example: BusDriverInf=MyMiDrv.inf

ClientDriverInf

Names of all .inf files provided for client drivers. Several files may be specified as a comma separated list of file names without a path. If the parameter SubDir of this section is empty the files must be located in the same directory like the setup application, otherwise they must be located in the given sub-folder.

Example: ClientDriverInf=MyComDrv.inf, MyNcmDrv.inf

ExcludeClientOnWin8AndNewer

Selected client drivers of the parameter ClientDriverInf that should not be installed on Windows 8 and newer operating systems. The parameter lists the names of the corresponding .inf files, without a path. Several files may be specified as a comma separated list.

The value is optional.

Note: The driver package specified by all other parameters of the section is always copied as a whole to the installation directory. The parameter only prevents installation of the given client drivers.

Example: ExcludeClientOnWin8AndNewer=MyNcmDrv.inf

SubDir

Sub-folder of all files listed in the section [Setup_x86]. If the parameter is empty the files must be located in the same directory like the setup application, otherwise they must be located in the given sub-folder. The folder will also be created on the destination system.

The value is optional.

Example:

Location of the setup application: *y:\install*

Installation directory: *c:\Program Files\Company\Product*

a) SubDir=x86

The files will be copied from *y:\install\x86*
to *c:\Program Files\Company\Product\x86*.

b) SubDir=

The files will be copied from *y:\install*
to *c:\Program Files\Company\Product*.

[Setup_x64]

Settings in this section will only be applied to 64-bit operating systems.

CopyFiles

Files of the setup package to be copied to the installation directory (e.g. .sys and .cat files). Several files may be specified as a comma separated list of file names without a path. If the parameter SubDir of this section is empty the files must be located in the same directory like the setup application, otherwise they must be located in the given sub-folder.

Note that files specified by other parameters of this section are implicitly copied and should not be specified by this parameter.

Example: CopyFiles=MyMiDrv_x64.sys, MyMiDrv_x64.cat, MyComDrv_x64.sys, MyComDrv_x64.cat

BusDriverInf

Name of the .inf file provided for the bus driver. If the parameter SubDir of this section is empty the file must be located in the same directory like the setup application, otherwise it must be located in the given sub-folder.

Example: BusDriverInf =MyMiDrv_x64.inf

ClientDriverInf

Names of all .inf files provided for client drivers. Several files may be specified as a comma separated list of file names without a path. If the parameter SubDir of this section is empty the files must be located in the same directory like the setup application, otherwise they must be located in the given sub-folder.

Example: ClientDriverInf=MyComDrv_x64.inf, MyNcmDrv_x64.inf

ExcludeClientOnWin8AndNewer

Selected client drivers of the parameter ClientDriverInf that should not be installed on Windows 8 and newer operating systems. The parameter lists the names of the corresponding .inf files, without a path. Several files may be specified as a comma separated list.

The value is optional.

Note: The driver package specified by all other parameters of the section is always copied as a whole to the installation directory. The parameter only prevents installation of the given client drivers.

Example: ExcludeClientOnWin8AndNewer=MyNcmDrv_x64.inf

SubDir

Sub-folder of all files listed in the section [Setup_x64]. If the parameter is empty the files must be located in the same directory like the setup application, otherwise they must be located in the given sub-folder. The folder will also be created on the destination system.

The value is optional.

Example:

Location of the setup application: *y:\install*

Installation directory: *c:\Program Files\Company\Product*

a) SubDir=x64

The files will be copied from *y:\install\x64*
to *c:\Program Files\Company\Product\x64*.

b) SubDir=

The files will be copied from *y:\install*
to *c:\Program Files\Company\Product*.

3 Silent mode

The setup supports a silent mode that can be enabled with command line options. This mode is designed to integrate the setup application into a surrounding setup program. When using this mode the following facts have to be considered:

- The silent mode only suppresses the graphical user interface of the setup application. It does not suppress any Windows setup dialog that may appear. Windows setup dialogs can be suppressed by signing the driver package.
- The setup is not a console mode application, even when running in silent mode. This means when executed on the command line the command prompt returns immediately while the setup is still running. To get the setup's exit code it should be started by an application that is able to wait for another process.
- The caller of the setup is responsible for user guidance which depends on the exit code.
- The setup only preinstalls the driver on the system. To complete the installation the system has to detect a connected device to which the preinstalled driver could be assigned. Therefore at the end of the installation the setup has to request the user to connect the device. But if the device is already connected it could happen that the system is not able to detect this connection. Therefore the user should be prompted instead to disconnect and reconnect the device. Since the setup is running silently this is the task of the surrounding setup program.

3.1 Silent Installation

3.1.1 Command Line Parameters

Parameter	Description
/S	Run installer in silent mode.
/DIR="target path"	Installation directory on the target system. If not specified, the installation directory given by the setup.ini file is used.
/NOPCPL	Prevent the installer from installing uninstallation support in the programs control panel of the system, which is the 'Programs and Features' control panel on Windows Vista and newer operating systems and the 'Add or Remove Programs' control panel before Windows Vista respectively.

Example: `setup.exe /S /DIR="c:\MyCompany\MyProduct"`

3.1.2 Exit codes

Status Code	Description	Solution
0	The setup finished successfully.	-
100	Setup aborted: Another PnP	Inform the user to close all open

	installation process is currently running on the system.	hard-ware installation wizards. If no wizard is currently open the system is probably performing some installation steps in the background. Wait some time and run setup again.
101	Setup finished successfully, but to complete driver installation a restart is required.	Reboot the system. If the device is already connected, no disconnect/reconnect is required after reboot.
102	Setup aborted: The current operating system is not supported by the setup.	Inform the user and abort.
103	Setup aborted: Unexpected error.	Inform the user and abort. The setup creates a log file in the user's temp directory. This file may help to analyse the problem.
104	Setup aborted: The current user has no administrator privileges which are required to proceed.	Inform the user and abort.
109	Setup aborted: Invalid command line parameters.	Call the setup with correct command line parameters.
111	Setup aborted: Another installer/uninstaller is already running.	The user has to finish the running (un-) installation at first.
112	Setup aborted: The installation directory could not be created on the destination system.	Inform the user or choose another installation directory and run setup again.
113	Setup aborted: The uninstaller file could not be extracted to the installation directory.	Inform the user or choose another installation directory and run setup again.
114	Setup aborted: Not all required files could be installed (Probably one or more files already exist in the installation directory and could not be overwritten.).	Inform the user or choose another installation directory and run setup again.
115	Setup aborted: Any driver could not be preinstalled on the system. The most likely reason is that the user didn't accept installation of unsigned drivers.	Inform the user that he should accept installation of unsigned drivers despite the system warnings and run setup again.
123	Setup aborted: setup.ini is not present or is corrupt.	Provide a correct setup.ini file.
125	Setup interrupted: Installation cannot continue because of	Reboot the system and run installation again.

	some locked resources. A restart is required.	
--	---	--

3.2 Silent Uninstallation

During every installation, the uninstallation support is installed as the very first step, so that if the installation fails for any reason the uninstallation can be performed to clean up the system. The uninstallation is supported in the following ways:

- If an installation is performed and the setup detects an existing installation on the target system that differs from the current one, the uninstaller of the existing installation is executed before the new installation starts. This means it is not required to explicitly cleanup the system before an installation. This is done implicitly by the installer.
- Unless specified otherwise, the installer creates an entry in the programs control panel of the system, which is the 'Programs and Features' control panel on Windows Vista and later and the 'Add or Remove Programs' control panel before Windows Vista. The user may run the driver uninstallation by means of this control panel. In this case the uninstaller shows a graphical user interface. You can prevent the installer from creating an entry in the control panel by means of the appropriate command line parameter. This may be useful if you provide an enclosing setup and you don't want to allow the user to uninstall the driver package separately.
- The setup allows you to perform an explicit silent uninstallation. Enclosing setups, such as application setups, are especially addressed by this feature. Usually the uninstallation of an enclosing setup should also involve the uninstallation of the driver package. The driver setup provides the appropriate command line parameters. Note that it is not required to use the same version of the setup executable that installed the existing driver package, but it must be a setup executable of the same product that is defined by the setup.ini. The called setup executable may be located anywhere on the system but the corresponding setup.ini file must be located in the same folder. The caller of the uninstallation is responsible for the deletion of the setup executable file, if required, because the application doesn't delete itself when uninstallation is finished.

Note: You should always wait for the result of the uninstallation! The uninstallation may be aborted for reasons that require some user interaction (see Exit Codes). Furthermore, any following installation will be aborted as long as an uninstallation is running. Even if your enclosing setup application does not intend to run an installation immediately after, the user may do so.

3.2.1 Command Line Parameters

Parameter	Description
/SU	The setup silently uninstalls the currently installed version, if there is any.

Example: setup.exe /SU

3.2.2 Exit codes

Status Code	Description	Solution
0	The uninstallation finished successfully. "Successfully" means that the uninstallation is as complete as possible. The uninstaller may detect that the existing installation is corrupt (e.g., because the user deleted some installed files). This may prevent the uninstaller from performing all required steps or may even prevent the whole uninstallation in the worst case. Since neither the user nor any calling application can resolve such problems no error code is returned.	-
100	Uninstallation aborted: Another PnP installation process is currently running on the system.	Inform the user to close all open hardware installation wizards. If currently no wizard is open the system is probably performing some installation steps in the background. Wait some time and run setup again.
101	Uninstallation finished successfully, but a restart is required for completion.	Reboot the system.
102	Uninstallation aborted: The current operating system is not supported by the uninstaller.	Inform the user and abort.
103	Uninstallation aborted: unexpected error.	Inform the user and abort. The uninstaller creates a log file in the user's temp directory. This file may help to analyse the problem.
104	Uninstallation aborted: The current user has no administrator privileges, which are required to proceed.	Inform the user and abort.
109	Uninstallation aborted: Invalid command line parameters.	Call the uninstaller with correct command line parameters.
111	Uninstallation aborted: Another installer / uninstaller is already running.	The user has to finish the running (un-) installation at first.

123	Uninstallation aborted: setup.ini is not present or is corrupt.	Provide a correct setup.ini file.
-----	---	-----------------------------------