

# **PnP Driver Installer**

## **A customizable PnP Driver Installer for Windows**

### **Reference Manual**

Version: 1.5.1  
Date: 10 January 2011

Authors: Günter Hildebrandt  
Rene Möller

Thesycon Systemsoftware & Consulting GmbH  
Werner-von-Siemens-Str. 2  
D-98693 Ilmenau  
Germany

Tel: +49 3677 8462 0  
Fax: +49 3677 8462 18

<http://www.thesycon.de>



# Contents

<b>Table of contents</b>	<b>4</b>
<b>1 Introduction</b>	<b>5</b>
1.1 Supported Operating Systems . . . . .	5
1.2 Features . . . . .	5
<b>2 Operation</b>	<b>6</b>
2.1 Driver Installation . . . . .	6
2.2 Removal of Driver Software . . . . .	6
2.3 Driver Update . . . . .	6
2.4 Demo Version . . . . .	6
2.5 WHQL Certification . . . . .	6
2.5.1 Installing WHQL Certified Drivers . . . . .	6
2.5.2 Installing not WHQL Certified Drivers . . . . .	7
2.5.3 Windows XP . . . . .	7
2.5.4 Windows Vista and Windows 7 . . . . .	7
2.6 Silent mode . . . . .	7
2.6.1 Installation . . . . .	8
Command Line Parameters . . . . .	8
Exit codes . . . . .	8
2.6.2 Uninstallation . . . . .	10
<b>3 Customization</b>	<b>11</b>
3.1 Package for Delivery . . . . .	11
3.2 INF File . . . . .	11
3.3 INI File Parameters . . . . .	12
3.3.1 [Setup] . . . . .	12
SupportedOS . . . . .	12
CompanyName . . . . .	12
ProductName . . . . .	12
ProductVersion . . . . .	12
SoftwareRegPath . . . . .	13
DefaultDestDir . . . . .	13
ShowLicenseText . . . . .	13

	CertificateFile . . . . .	13
	CopyFiles . . . . .	13
	ShortCut[1..n] . . . . .	14
3.3.2	[Setup_x86] . . . . .	14
	CopyFiles . . . . .	14
	PnPDriverInf . . . . .	15
	RegisterFiles . . . . .	15
	SubDir . . . . .	15
	ShortCut[1..n] . . . . .	15
3.3.3	[Setup_x64] . . . . .	16
	CopyFiles . . . . .	16
	PnPDriverInf . . . . .	16
	RegisterFiles . . . . .	17
	SubDir . . . . .	17
	ShortCut[1..n] . . . . .	17
<b>4</b>	<b>Limitations</b>	<b>19</b>

# 1 Introduction

The PnP Driver Installer package can be used to install kernel mode drivers for Windows in a reliable and convenient way. It can handle the first time installation, the update of the driver and the removal of the complete driver software. The driver installer can be run with a graphical user interface or in silent mode. It can support different languages and it can be customized to support different devices and kernel mode drivers. Regardless of the system state the installer makes sure that the required driver is installed.

## 1.1 Supported Operating Systems

The PnP Driver Installer supports the following operating systems:

- Windows 2000
- Windows XP (32 and 64 bit)
- Windows Vista (32 and 64 bit)
- Windows 7 (32 and 64 bit)
- Windows Server 2003 (32 and 64 bit)
- Windows Home Server
- Windows Server 2008 (32 and 64 bit)
- Windows Server 2008 R2

## 1.2 Features

The PnP Driver Installer provides the following features:

- Support for first time installation, driver update and removal of the driver software.
- Removes all other driver packages for the device that match the hardware IDs of the package to install.
- Installs exactly the driver version provided by the installed package (even in the case of a rollback).
- Interactive mode with graphical user interface.
- Silent mode to integrate the installer into other setups.
- Support for different languages.
- Can be customized easily.
- Supports signed and unsigned drivers.

## 2 Operation

### 2.1 Driver Installation

The installer performs a pre-installation of the driver software. During this process the driver becomes known to the system. On Windows XP the INF file is copied to the INF folder and a pre-compiled INF files is created. On Windows Vista and later the driver package is stored in the driver store. When a PnP device is connected to the system it finds the related driver and install it.

At the end of the installation the GUI based driver setup asks the user to disconnect and re-connect the device.

During the driver installation an entry in Windows programs control panel is created. This entry can be used to uninstall the driver software.

### 2.2 Removal of Driver Software

The installer removes all driver installations that match to the hardware IDs provided by the driver package to install. This makes sure that exactly one driver is installed after the setup has been completed. It is important that exactly one driver version is installed to make sure that exactly this version is loaded. During the removal process all installed device nodes including the corresponding registry entries and the pre-installation of the driver software are removed. After this step the system behaves in such a way as when the driver software was never installed.

### 2.3 Driver Update

The driver update is a combination of the Removal of the driver software and a first time installation. The installer detects automatically if an update or a first time installation is required.

### 2.4 Demo Version

The PnP Driver Installer exists in a Demo and in a Full version. The Demo version has the limitation that the Demo Mode Control Panel `DemoModeCpl.exe` must be started before the installer can be used. The Demo Mode Control Panel requests the user to enter a random number. Than the installer can be enabled with the button "Enable". The Demo Mode Control Panel window is always on top of the desktop. It must run until the installer has been finished.

The Full version does not have this limitation.

### 2.5 WHQL Certification

#### 2.5.1 Installing WHQL Certified Drivers

A WHQL signed driver can be installed silent on all operating systems. This is the recommended way of Microsoft to install a kernel mode driver.

### 2.5.2 Installing not WHQL Certified Drivers

If the driver is not WHQL certified the systems show different behavior that is described in the following sections. The system behavior depends on the setting of the Driver Signing Options. The default setting is: "Warn". In this case a warning box is shown. The setting can be changed to "block" or "ignore". The following description assumes the default setting.

### 2.5.3 Windows XP

Windows XP distinguish between device classes where the system creates a warning box and other classes where a kernel driver can be installed without a warning. Proprietary device classes does not cause a warning box. An Authenticode signature is not considered by the system. During the pre-installation of a driver a warning box appears when the driver is installed in a system controlled device class. The warn box appears again when the device is connected the first time to the PC.

When a device is connected and a pre-installed driver exists the Found New Hardware wizard is launched. The user just has to choose the option "No, not this time" on the first page and to continue with "Next" on every page until finished. When the hardware wizard is launched the user needs administrator privileges.

### 2.5.4 Windows Vista and Windows 7

Both systems behaves similar and are described together. The systems trade all device classes in the same way. They consider Authenticode signatures on the CAT and on the SYS file. The signature on the CAT file is considered during the pre-installation of the driver. During the installation of the Authenticode signed driver a warning box is shown that identifies the vendor. The customer can chose to "Always trust this vendor". In this case the following installations from this vendor are performed silent. When the CertificateFile parameter in the INI file is used the warn box can be suppressed.

The x64 editions require at least a valid Authenticode signature. Otherwise the driver does not run and shows the error code 39 in the device manager.

If a device is connected later and the driver is successful pre-installed the installation process is performed silent. Some information hints in the Systray indicates the progress of the installation. When the device is connected no administrator privileges are required.

## 2.6 Silent mode

The PnP Driver Installer supports a silent mode that can be enabled with command line options. This mode is designed to integrate the PnP Driver Installer into a surrounding installation program. When using this mode note the following:

- The silent mode only suppresses the GUI of the setup application. It does not suppress any Windows setup dialog that may appear.
- The installer is no console mode application, even when running in silent mode. I.e., when executed on the command line the command prompt returns immediately while the setup is

still running. To get the setup exit code it should be started by an application that is able to wait for another process.

- The caller is responsible for user guidance which depends on the exit code of the setup (see section "Exit codes").
- The setup only prepares installation and pre-installs the drivers on the PC. After it finished successfully the user can connect the device to the system at any later point in time to complete the installation. If the device is already connected it is required to disconnect and reconnect the device. The caller of the setup has to inform the user about this required step.
- After the device has been (re-)connected the system assigns the pre-installed drivers to the device. I.e., the device is not accessible until this final installation step finished successfully. Under XP the hardware wizard will appear. The user just has to follow the instructions of the hardware wizard. He can choose the option "No, not this time" on the first page and should continue with "Next" on every following page until finished.

### 2.6.1 Installation

#### Command Line Parameters

Parameter	Description
/S	Run installer in silent mode.
/DIR="target path"	Installation directory on the target system. If not specified the default installation directory is used.
/NOPCPL	Prevent the installer from installing uninstallation support in Windows programs control panel.

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

#### Exit codes

Status Code	Description	Solution
0	The setup finished successfully.	-
100	Setup 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 probably performs some installation steps in the background. Just wait some time. Then 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 anymore 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 analyze 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: The 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 (e.g., because of missing permissions).	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 (e.g., because of missing permissions to write 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 pre-installed on the system. The most likely reason is that the user didn't accepted installation of unsigned drivers.	Inform the user that he should accept installation of unsigned drivers despite the system warnings and run setup again.
118	Setup aborted: Self-registration of some modules failed.	Inform the user and abort.
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 some locked resources. A restart is required.	Reboot the system and run installation again.
126	Setup aborted: Setup aborted: Running this version of the setup application requires activation of the Demo Mode by means of the corresponding "Demo Mode Control Panel".	This code is only returned in the demo edition of the setup which is not delivered to the end user. It has not to be handled by a calling application. Check the log file of the setup in the user's TEMP folder for further information. If the setup is executed in non-silent mode the information is displayed by a message box.

### 2.6.2 Uninstallation

During installation the setup extracts an uninstaller application to the main installation folder (uninstall.exe). It is not required to explicitly run the uninstaller in silent mode because of the following reasons:

- If an installation is performed and the setup detects an existing installation on the destination system that differs from the current one the uninstaller of the existing installation is executed before the new installation starts. I.e., it is never required to explicitly cleanup the system before an installation. This is implicitly done by the installer.
- The user may run uninstallation by means of Windows programs control panel. The uninstaller is implicitly called by the control panel.
- If the installation failed (see section "Exit codes") the user should be informed. If the setup is not repeated he may also optionally be informed to use the control panel for cleanup. If the setup is repeated the installer will call the uninstaller implicitly.

If the uninstaller is called explicitly it copies itself to a temporary folder and starts this copy. The copy actually performs the uninstallation while the primary uninstaller exits immediately. This approach allows the uninstallation of the uninstaller. By reason of this approach it makes no sense to wait for the uninstaller to check whether the uninstallation finished, because the called instance terminates while the uninstallation is still running in the background.

## 3 Customization

The driver package contains the program Setup.exe and the .ini file. The setup program is not modified during the customization process. All required changes are performed with the parameters in the .ini file.

### 3.1 Package for Delivery

The complete installer 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.
- <driver\_name\_x86>.sys - The kernel driver(s) for x86 systems.
- <driver\_name\_x86>.inf - The INF file(s) for x86 systems.
- <driver\_name\_x86>.cat - The catalog file(s) with digital signature for x86 systems. This files are optional but recommended.
- <COM\_module\_name\_x86>.dll - The optional COM DLL(s) to register for x86 (and x64) systems.
- <driver\_name\_x64>.sys - The kernel driver(s) for x64 systems.
- <driver\_name\_x64>.inf - The INF file(s) for x64 systems.
- <driver\_name\_x64>.cat - The catalog file(s) with digital signature for x64 systems. This files are required for Windows Vista x64 and later OS.
- <COM\_module\_name\_x64>.dll - The optional COM DLL(s) to register for x64 systems.
- license.txt - The optional license text.
- <certificate\_name>.cer - The optional certificate of the Authenticode key.

These files should be stored in a folder. Optionally sub-folders can be used to separate files of the x86 and x64 driver packages. 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.

The CAT files contain either a digital signature obtained from a WHQL submission or an Authenticode signature added by the vendor. For details how to create a CAT file please refer to the user manual of the driver package if the package is provided by Thesycon.

### 3.2 INF File

The INF file describes how the kernel driver is installed. The full customization of this file is described in the user manual of the driver package if the package is provided by Thesycon.

The copy file section in the INF file describes how the files are copied. It can contain one or more lines like this:

```
<driver_name>.sys,,,0x00000004
```

The flag 0x00000004 means COPYFLG\_NOVERSIONCHECK. If this flag is set the installation can overwrite a newer version of the driver without showing a warning box to the user. It is recommended to set this flag if the user should be able to install an older version of the software over a new one.

### 3.3 INI File Parameters

The INI file is a text based configuration file. It can be edited with a text editor like Notepad.exe. The INI file is divided into sections. A sections 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. The meaning of the parameters are explained by comments in the INI file and in the following sections. The comments should be removed after the customization process has been finished.

#### 3.3.1 [Setup]

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

##### SupportedOS

Operating systems supported by the setup.

The key has to contain a comma-separated list of the following tokens:

2000, XP\_X86, XP\_X64, Server2003\_X86, HomeServer, Server2003\_X64, Vista\_X86, Vista\_X64, Server2008\_X86, Server2008\_X64, 7\_X86, 7\_X64, Server2008R2\_X64

Example: SupportedOS = XP\_X86, Vista\_X86, Vista\_X64, 7\_X86, 7\_X64

##### CompanyName

This is your company name shown in Windows programs control panel.

##### ProductName

This is the Product name shown in dialogs.

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

##### ProductVersion

The Product version shown in dialogs.

**SoftwareRegPath**

The Registry path of the installed software. It is also used to store setup information. The path is relative to HKEY\_LOCAL\_MACHINE\Software.

Note: This key must not be changed in future releases of your setup package. Otherwise 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: The path must be world wide unique. To reach this use your company name and your product name for the path e.g. <your company name>\<your product name>.

**DefaultDestDir**

Is the Default destination directory on the user's system. The path is relative to 'Program Files'.

Note: The path must be world wide unique. To reach this use your company name and your product name for the path e.g. <your company name>\<your product name>.

**ShowLicenseText**

Define whether or not a license agreement should be displayed and accepted by the user before installation begins. The license text has to be provided in a plain text file named "license.txt" if this option is enabled (ShowLicenseText=1). The value is either 0 or 1.

**CertificateFile**

The name of the file that contains the vendor certificate used to sign the drivers (public key!). The certificate is installed in the "Trusted Publisher" store. So the user is never asked whether or not to trust the driver publisher. After the installation completed the certificate is not required anymore and therefore removed from the certificate store.

The value is optional.

Note: This certificate is only installed under Windows Vista / Vista x64 and above. Otherwise it is ignored.

Example: CertificateFile=MyCertificate.cer

**CopyFiles**

This key specifies common files to be copied to the destination directory (e.g. ReadMe.txt). Several files may be specified as a comma separated list of file names without path. The files have to be located in the same directory like the setup application.

The value is optional.

Note that files mentioned in other keys are copied by default and are not to be included in this key (see below).

**ShortCut[1..n]**

Optional keys of type `ShortCutX=ValueString` that define shortcuts to be created. Start with `X=1` and increment `X` with each shortcut definition. `ValueString` has the following format:

```
SC_ROOT | SC_PATH | SC_NAME | TARGET | PARAMS | START_PATH
```

<b>SC_ROOT</b>	Root directory of the shortcut. This directory will not be deleted during uninstallation. It can be set to the following variables: \$SM_PROG_CURRENT_USER (start menu\programs of the current user) \$SM_STARTUP_CURRENT_USER (start menu\startup of the current user) \$SM_PROG_ALL_USERS (start menu\programs of all users) \$SM_STARTUP_ALL_USERS (start menu\startup of all users)
<b>SC_PATH</b>	Optional path of the shortcut to be created. This path is relative to <code>SC_ROOT</code> .
<b>SC_NAME</b>	Name of the shortcut to be created.
<b>TARGET</b>	Full path and name of the executable file on the target system the shortcut points to. One of the variables <code>\$INSTDIR</code> (main installation directory) <code>\$WINDIR</code> (Windows directory) and <code>\$SYSDIR</code> (system32 directory) should be used to reference the target file.
<b>PARAMS</b>	Optional parameters passed to the target executable file when it is launched.
<b>START_PATH</b>	Optional start directory for the target executable file.

Example:

```
ShortCut1=$SM_PROG_ALL_USERS | MySubDir | License.lnk | $INSTDIR\License.txt | |  
ShortCut2=$SM_PROG_ALL_USERS | MySubDir | ReadMe.lnk | $INSTDIR\ReadMe.txt | |
```

**3.3.2 [Setup\_x86]**

Settings in this section are only executed on a x86 system.

**CopyFiles**

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

The value is optional.

Note that files mentioned in other keys are copied by default and are not to be included in this key (see below).

**PnPDriverInf**

This key specifies the .inf files that describe the Plug'n'Play drivers to be installed. Several files may be specified as a comma separated list of file names without path. If the key `SubDir` of this section is empty the files have to be located in the same directory like the setup application, otherwise they have to be located in the given sub-folder.

The files will also be copied to the destination directory.

Note: If multiple INF files are specified they may not be related in a bus driver architecture (see section "Limitations").

**RegisterFiles**

Optional key that specifies the COM DLLs to register. The DLLs have to support self-registration. Several files may be specified as a comma separated list of file names without path. If the key `SubDir` of this section is empty the files have to be located in the same directory like the setup application, otherwise they have to be located in the given sub-folder.

The files will also be copied to the destination directory.

**SubDir**

Optional key that specifies a sub-folder. All files listed in section [Setup\_x86] are expected to be located in this folder. The folder will also be created on the destination system.

Example:

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

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

a) `SubDir=x86`

The files are copied from `y:\install\x86`  
to `c:\Program Files\Company\Product\x86`.

b) `SubDir=`

The files are copied from `y:\install`  
to `c:\Program Files\Company\Product`.

**ShortCut[1..n]**

Optional keys of type `ShortCutX=ValueString` that define shortcuts to be created. Start with `X=1` and increment `X` with each shortcut definition. `ValueString` has the following format:

`SC_ROOT | SC_PATH | SC_NAME | TARGET | PARAMS | START_PATH`

SC_ROOT	Root directory of the shortcut. This directory will not be deleted during uninstallation. It can be set to the following variables: \$SM_PROG_CURRENT_USER (start menu\programs of the current user) \$SM_STARTUP_CURRENT_USER (start menu\startup of the current user) \$SM_PROG_ALL_USERS (start menu\programs of all users) \$SM_STARTUP_ALL_USERS (start menu\startup of all users)
SC_PATH	Optional path of the shortcut to be created. This path is relative to SC_ROOT.
SC_NAME	Name of the shortcut to be created.
TARGET	Full path and name of the executable file on the target system the shortcut points to. One of the variables \$INSTDIR (main installation directory) \$WINDIR (Windows directory) and \$SYSDIR (system32 directory) should be used to reference the target file. If the key SubDir of this section defines a sub-folder it has to be specified too.
PARAMS	Optional parameters passed to the target executable file when it is launched.
START_PATH	Optional start directory for the target executable file.

Example:

```
Shortcut1=$SM_STARTUP_ALL_USERS | |My CPL.lnk| $INSTDIR\MyCpl.exe | -s |
```

### 3.3.3 [Setup\_x64]

Settings in this section are only executed on a x64 system.

#### CopyFiles

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

The value is optional.

Note that files mentioned in other keys are copied by default and are not to be included in this key (see below).

#### PnPDriverInf

This key specifies the .inf files that describe the Plug'n'Play drivers to be installed. Several files may be specified as a comma separated list of file names without path. If the key SubDir of this section is empty the files have to be located in the same directory like the setup application, otherwise they have to be located in the given sub-folder.

The files will also be copied to the destination directory.

Note: If multiple INF files are specified they may not be related in a bus driver architecture (see section "Limitations").

### **RegisterFiles**

Optional key that specifies the COM DLLs to register. The DLLs have to support self-registration. Several files may be specified as a comma separated list of file names without path. If the key `SubDir` of this section is empty the files have to be located in the same directory like the setup application, otherwise they have to be located in the given sub-folder.

The files will also be copied to the destination directory.

### **SubDir**

Optional key that specifies a sub-folder. All files listed in section [Setup\_x64] are expected to be located in this folder. The folder will also be created on the destination system.

Example:

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

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

a) `SubDir=x64`

The files are copied from `y:\install\x64`

to `c:\Program Files\Company\Product\x64`.

b) `SubDir=`

The files are copied from `y:\install`

to `c:\Program Files\Company\Product`.

### **ShortCut[1..n]**

Optional keys of type `ShortCutX=ValueString` that define shortcuts to be created. Start with `X=1` and increment `X` with each shortcut definition. `ValueString` has the following format:

`SC_ROOT | SC_PATH | SC_NAME | TARGET | PARAMS | START_PATH`

SC_ROOT	Root directory of the shortcut. This directory will not be deleted during uninstallation. It can be set to the following variables: \$SM_PROG_CURRENT_USER (start menu\programs of the current user) \$SM_STARTUP_CURRENT_USER (start menu\startup of the current user) \$SM_PROG_ALL_USERS (start menu\programs of all users) \$SM_STARTUP_ALL_USERS (start menu\startup of all users)
SC_PATH	Optional path of the shortcut to be created. This path is relative to SC_ROOT.
SC_NAME	Name of the shortcut to be created.
TARGET	Full path and name of the executable file on the target system the shortcut points to. One of the variables \$INSTDIR (main installation directory) \$WINDIR (Windows directory) and \$SYSDIR (system32 directory) should be used to reference the target file. If the key SubDir of this section defines a sub-folder it has to be specified too.
PARAMS	Optional parameters passed to the target executable file when it is launched.
START_PATH	Optional start directory for the target executable file.

Example:

```
Shortcut1=$SM_STARTUP_ALL_USERS | |My CPL.lnk | $INSTDIR\x64\MyCpl_x64.exe | -s |
```

## 4 Limitations

The PnP Driver Installer cannot install driver packages containing drivers related in a bus driver architecture. This is true if the driver package contains

- a driver that creates child devices when loaded **and**
- drivers for these created child devices.

Such driver packages require a specific setup. Please contact Thesycon ([www.thesycon.de](http://www.thesycon.de)) for further information.

Nevertheless, installation of driver packages containing multiple INF files (drivers) which are not related in a bus driver architecture is supported by the PnP Driver Installer.

