ConfigMgr Driver Injector

Instructions

Mickael PONSOT

April 2013

[Mickael.ponsot@gmail.com](mailto:Mickael.ponsot@gmail.com) – <http://mponsot.wordpress.com>

Version 1.1– Second release

Table of content

[Presentation 3](#_Toc352941209)

[Prerequisites 4](#_Toc352941210)

[How it works 5](#_Toc352941211)

[ConfigMgrDriverGrabber 5](#_Toc352941212)

[ConfigMgrDriverInjector 5](#_Toc352941213)

[How to use 7](#_Toc352941214)

[ConfigMgrDriverGrabber 7](#_Toc352941215)

[Manual use 7](#_Toc352941216)

[Silent use 7](#_Toc352941217)

[Deployment via System Center 2012 Configuration Manager 9](#_Toc352941218)

[ConfigMgrDriverInjector 11](#_Toc352941219)

[How to use a specific driver package in a task sequence 12](#_Toc352941220)

[Notes 14](#_Toc352941221)

# Presentation

ConfigMgr Driver Injector is a tool which can be used to automatically import drivers into System Center 2012 Configuration Manager OSD module.

The tool is made of two utilities:

* ConfigMgrDriverGrabber

This is the client part of the tool, this utility captures all installed drivers on a workstation and then send them to a specified UNC share. Drivers are sorted by driver class in the destination folder. This utility can be used with a GUI or silently from the command line.

* ConfigMgrDriverInjector

This is the server part of the tool, this utility parse the drivers capture folder and import all found drivers into SCCM. Each imported driver will be assigned to proper drivers categories and then assigned to a driver package.

# Prerequisites

Both utilities needs Framework .Net 4.

The ConfigMgrDriverInjector utility must be run on the System Center 2012 Configuration Manager site server.

The driver capture folder must be dedicated to that use. The account used to grab drivers must have read and write permissions on that shared folder.

The account used to run ConfigMgrDriverInjector must have read and write permissions on the capture share, the drivers source folder and the drivers package source folder.

# How it works

## ConfigMgrDriverGrabber

ConfigMgrDriverGrabber lloks for installed drivers on the workstation. Selected drivers are captured in a temporary file. When all drivers are captured, the temporary file is sent to the capture share.

If ConfigMgrDriverGrabber.exe is launched without parameters, a GUI will shows up allowing you to tune the way drivers will be grabbed. ConfigMgrDriverGrabber can be launched silently with the correct command parameters.

The drivers capture folder is named with some randomly generated GUID. In the root of this folder a XML file which is also named with this GUID is generated. The folder also contains a sub folder per driver class. These sub folders then contains a folder per captured driver.

The XML file contains some metadata about the workstation brand, model, OS and architecture. It also contains a “status” node which allows ConfigMgrDriverInjector to know if the folder have already been processed.

A log file compatible with the cmtrace tool is generated in the folder of the application.

## ConfigMgrDriverInjector

ConfigMgrDriverInjector starts by retreiving the paths of the capture folder, the drivers source folder and the drivers packages source folder. These information are stored in the ConfigMgrDriverInjector.exe.config file.

ConfigMgrDriverInjector then parse the capture share, if a folder contains an xml file with the folder name and if the xml file contains a “status” node equals to 0 then ConfigMgrDriverInjector consider it as a valid capture folder.

ConfigMgrDriverInjector will then create a ConfigMgrDriverInjector driver category, this category is necessary for ConfigMgrDriverInjector to work correctly, it must not be deleted. Each imported driver will be assigned to that category.

ConfigMgrDriverInjector will then create all needed drivers categories. The categories are:

* Workstation’s model
* Workstation’s brand
* Workstation’s OS
* Workstation’s architecture
* One category per driver class

After that, ConfigMgrDriverInjector will copy the capture folder to the driver’s source path. The destination folder will be named according to that convention:

<Brand>\_ <Model>\_ <OS>\_ <Architecture>

With the same naming convention ConfigMgrDriverInjector will create a driver package.

Finally ConfigMgrDriverInjector will import all captured drivers, assign them to the proper categories and then assign them to the correct driver package.

If a driver has been previously imported it won’t get imported again. Instead, ConfigMgrDriverInjector will use the already existing driver by adding it the new categories and adding the driver to the new package. In such cases the source folder of the driver will be deleted as it won’t be used. If all driver folders gets deleted (all drivers were already in Config Manager) then the entire source folder will be deleted.

A log file compatible with the cmtrace tool is generated in the folder of the application.

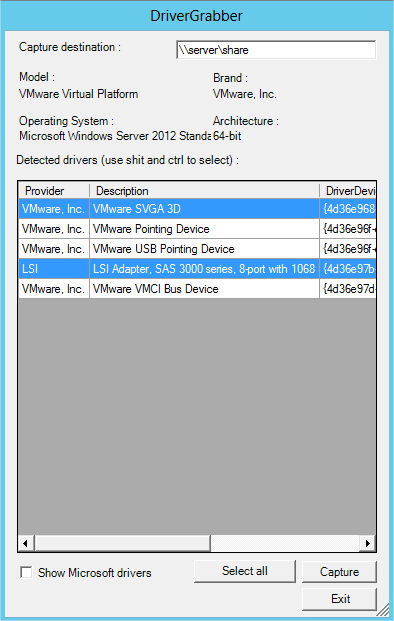
# How to use

## ConfigMgrDriverGrabber

### Manual use

Launch as admin ConfigMgrDriverGrabber.exe. (To get some verbose logs, use the /debug switch).

The following windows open:



Fill in the Capture destination field the UNC path of the capture share.

Choose the drivers to capture. You can use shift and ctrl for multi selection.

You can choose if you want to show or not Microsoft’s drivers. These drivers are included in Windows so capturing them is not recommended.

### Silent use

ConfigMgrDriverGrabber can be used in a silent way (to be deployed via System Center 2012 Configuration Manager for exemple).

Launch as admin ConfigMgrDriverGrabber.exe with the following parameters:

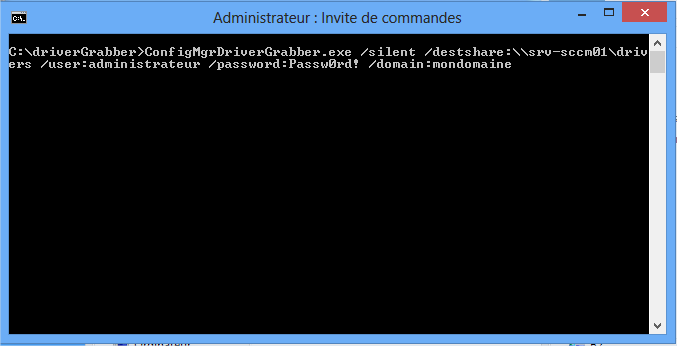
/silent /destshare:<unc\_capture\_target>

The following optionnal parameters are also available:

* /debug : To get verbose logs
* /captureMSDriver : Will capture Microsoft’s drivers (not recommended)
* /user:<username> : the name of the account that will be used to map the target share
* /password:<password> : the password of the user
* /domain:<domain> : the domain of the user

Note: If /user:<username> if used, then /password:<password> et /domain:<domain> are mandatory.

Example:



### Deployment via System Center 2012 Configuration Manager

|  |  |
| --- | --- |
| **Step** | **Result** |
| Create a new package |  |
| Specifiy basic informations.  Specify the package sources. |  |
| Choose to create a standard program. |  |
| Specify the program’s information.  The command line must looks like that :  ConfigMgrDriverGrabber.exe /silent /destshare:<UNC\_capture\_path> |  |

Then distribute the package to a distribution point and deploy it to the desired collection.

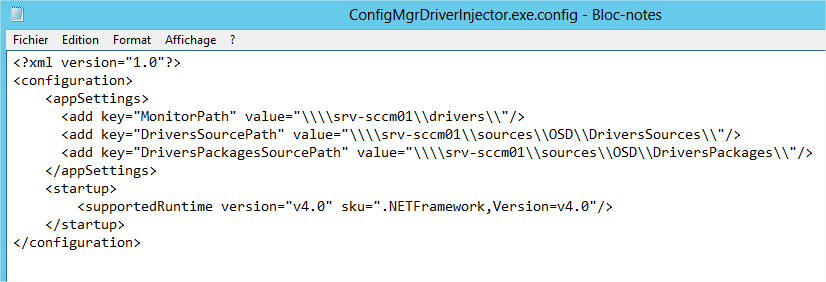
## ConfigMgrDriverInjector

Copy the ConfigMgrDriverInjector’s sources in a local folder of the System Center 2012 Configuration Manager site’s server.

Create a shared folder for drivers capture (example: c:\drivers). Make sure that the account that will be used to launch ConfigMgrDriverGrabber and ConfigMgrDriverInjector can read and write in that shared folder.

If it is not already done, create two shared folders. One for driver’s sources and one for drivers package’s sources (example: c:\sources\OSD\drivers and c:\sources\OSD\driverPackages). Make sure that the account that will be used to launch ConfigMgrDriverInjector can read and write in theses shared folders.

Edit ConfigMgrDriverInjector.exe.config, with the folders you just created (bacslashs must be escaped with a second backslash, folder’s names must end with \\).



« MonitorPath » is the capture folder.

« DriversSourcePath » Is the driver’s sources folder for System Center 2012 Configuration Manager.

« DriversPackagesSourcePath » » Is the drivers package’s sources folder for System Center 2012 Configuration Manager.

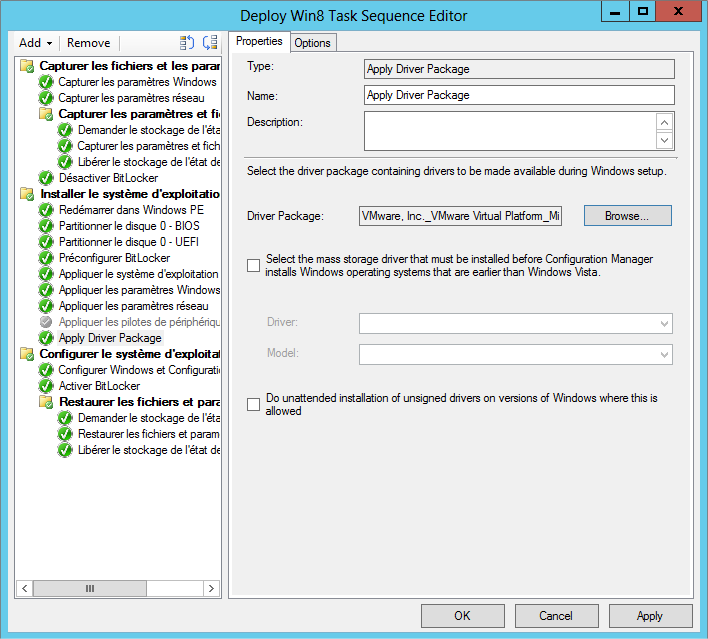
Run ConfigMgrDriverInjector.exe to launch the integration process. (You can use /debug parameter to get verbose logs).

You can create a scheduled task in Windows to get a recurring execution of ConfigMgrDriverInjector.

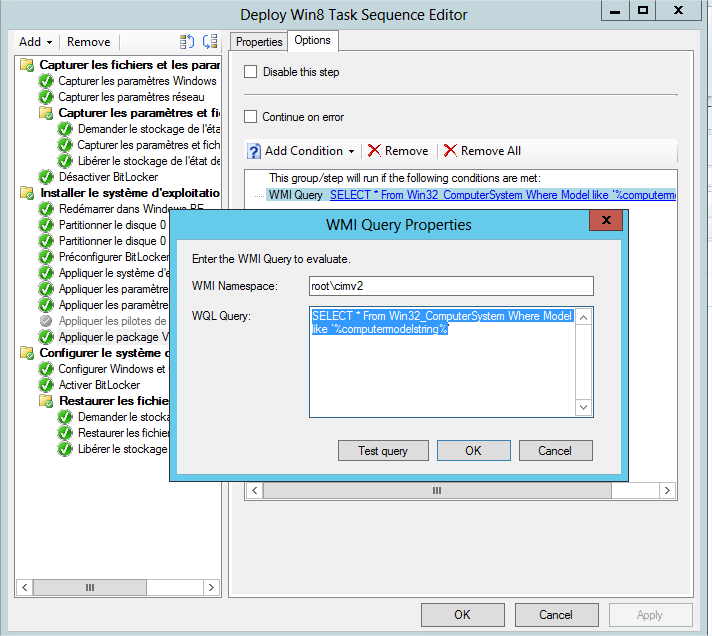
# How to use a specific driver package in a task sequence

To force a specific driver package in a task sequence:

Edit the task sequence, disable the default inject drivers step and add a apply driver package step. Choose the appropriate driver package:



Go into the option pane. Click on « add condition » and add a WMI query contidion :



Enter the following query: SELECT \* From Win32\_ComputerSystem Where Model like ‘%computermodelstring%’

Replace computermodelstring with the workstation’s model name (it’s the same model name used in the categories and in the package name), keep the ‘%’ character as it is used as wildcard.

Warning ! Some manufacturer (especially HP) add a string between parentheses at the end of the model string. The funny thing is that this string can change between two computers of the same model. If you encounter the case just remove the parenthesis string in your WMI query.

Repeat these steps for each driver packages.

# Notes

Added in Version 1.1:

* Corrected a bug that prevented the Drivergrabber to work on Windows XP (XP x64 will be seen as a x86 Operating System)

This tool is provided “as is”, I can’t guarantee it will grabs and install 100% of drivers.

Even if it was tested many time you are the only responsible for potential issue caused by the use of this tool.

You are free to use and distribute this tool as you want.

The tool automatically switch between French and English. If you want it in your language I can provide you the string file and then I’ll be able to publish a localized version.