

Matrix42

Matrix42 Driver integration for WinPE - HowTo

This documentation is protected by copyright. All rights reserved by Matrix42 AG.

Any other use, in particular the passing on to third parties, storage within a data system, distribution, processing, presentation, performance and demonstration are prohibited. This applies to the entire document as well as parts thereof. We reserve the right to make changes.

No part of this publication may be reproduced in any form or by any means without the prior written permission of Matrix42 AG.

The software described in this documentation is subject to permanent further development, which may result in differences between the documentation and the actual software. This documentation makes no claim to the actual functional scope of the software.

Apple and **Mac OS X** are registered trademarks of Apple Inc.

Citrix® software or **Citrix® server** are Trademarks and Registered Trademarks of Citrix Systems, Inc. in the United States and other countries.

cygwin is copyrighted by Red Hat Inc. 1996-2003.

expat is copyrighted by Thai Open Source Software Center Ltd.

gSOAP is copyrighted by Robert A. van Engelen, Genivia, Inc. All rights reserved.

Iconv is copyrighted by 1999-2003 Free Software Foundation, Inc.

lperf is copyrighted by the University of Illinois, except for the `gnu_getopt.c`, `gnu_getopt_long.c`, `gnu_getopt.h` files, and `inet_aton.c`, which are under the GNU General Public License.

Libmspack (C) 2003-2004 by Stuart Caie <kyzer@4u.net>.

OpenSSL This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit.

PuTTY is copyrighted by Simon Tatham. Portions copyright Robert de Bath, Joris van Rantwijk, Delian Delchev, Andreas Schultz, Jeroen Massar, Wez Furlong, Nicolas Barry, Justin Bradford, Ben Harris, Malcolm Smith, Ahmad Khalifa, Markus Kuhn, and CORE SDI S.A.

RSA Data Security, Inc. MD5 Message-Digest Algorithm is copyrighted by RSA Data Security Inc. Created 1991. All rights reserved.

rsync is an open source utility that provides fast incremental file transfer. `rsync` is freely available under the GNU General Public License version 2.

runcontrol The Initial Developer of the Original Code is James Clark. Portions created by James Clark are Copyright (c) 1998 James Clark. All rights reserved.

SNMP++ Copyright (c) 1996 Hewlett-Packard Company.

VMware, the **VMware "boxes" logo and design**, **Virtual SMP**, **VMotion vSphere**, **vSphere Hypervisor (ESXi)**, **ESX**, **View**, **ThinApp**, **vCenter** and **vCloud** are registered trademarks or trademarks of VMware, Inc. in the United States and/or other jurisdictions.

Windows, **Windows Vista**, **Windows Server 2008**, **Windows 7**, **Windows Server 2008 R2**, **Windows 8 (8.1)**, **Windows Server 2012**, **Windows 10** and **Windows Server 2016** are registered trademarks of Microsoft Corporation.

Others, at this point not explicitly listed, company, brand and product names are trademarks or registered trademarks of their respective owners and are subject to trademark protection.

Table of Contents

1	About this document	4
1.1	Prerequisites	4
2	Instructions for the provision of WinPE drivers	5
2.1	Driver integration for a boot configuration	5
2.2	Reading Log File Information	10
3	Instructions for the Provision of Operating System Drivers	12
3.1	Concept.....	12
3.2	Determination of the vendor and the model	14
3.3	Matrix42 Driver Assistant for WinPE	14
3.3.1	Prerequisites	14
3.3.2	Current functionality	15
3.3.3	Limitations.....	15
3.3.4	Starting the Matrix42 Driver Assistant.....	15
3.3.5	Creating a Drivers.json file at first startup	16
3.3.6	The Matrix42 Driver Assistant interface	17
3.3.7	Importing HardwareInfo package output files.....	17
3.3.8	Import driver file	21
3.3.9	Manual extension.....	23
3.3.10	Conflicts	25
3.3.11	Filtering functions.....	26
4	Injecting Drivers into a Boot Configuration	29
5	FAQ	31

1 About this document

This document provides an overview of driver integration in WinPE based OS deployment. There is a step by step guide on how to integrate the drivers.

1.1 Prerequisites

Empirum Unified Endpoint Management (v17.0.2 or higher).

Empirum WinPE PreBoot Support (version 1.4.15 or higher).

2 Instructions for the provision of WinPE drivers

In the following chapter, we describe the provisioning of WinPE drivers, which are necessary to be able to communicate with hardware that is not yet natively supported by the current WinPE.

Despite frequent updates of the Microsoft ADK, it is possible that certain hardware types are not or not correctly addressed by the supplied drivers. The manufacturers usually offer special WinPE driver packages for this, which provide a remedy.

Note: Please note that this does not always have to be a "fully functional" driver for operating system that is deployed. If the manufacturer offers drivers for your operating system, we suggest preferring them for the operating system installation.

For a detailed list of the supported operating systems, you can generally refer to the driver package README file or ask the manufacturer if necessary.

2.1 Driver integration for a boot configuration

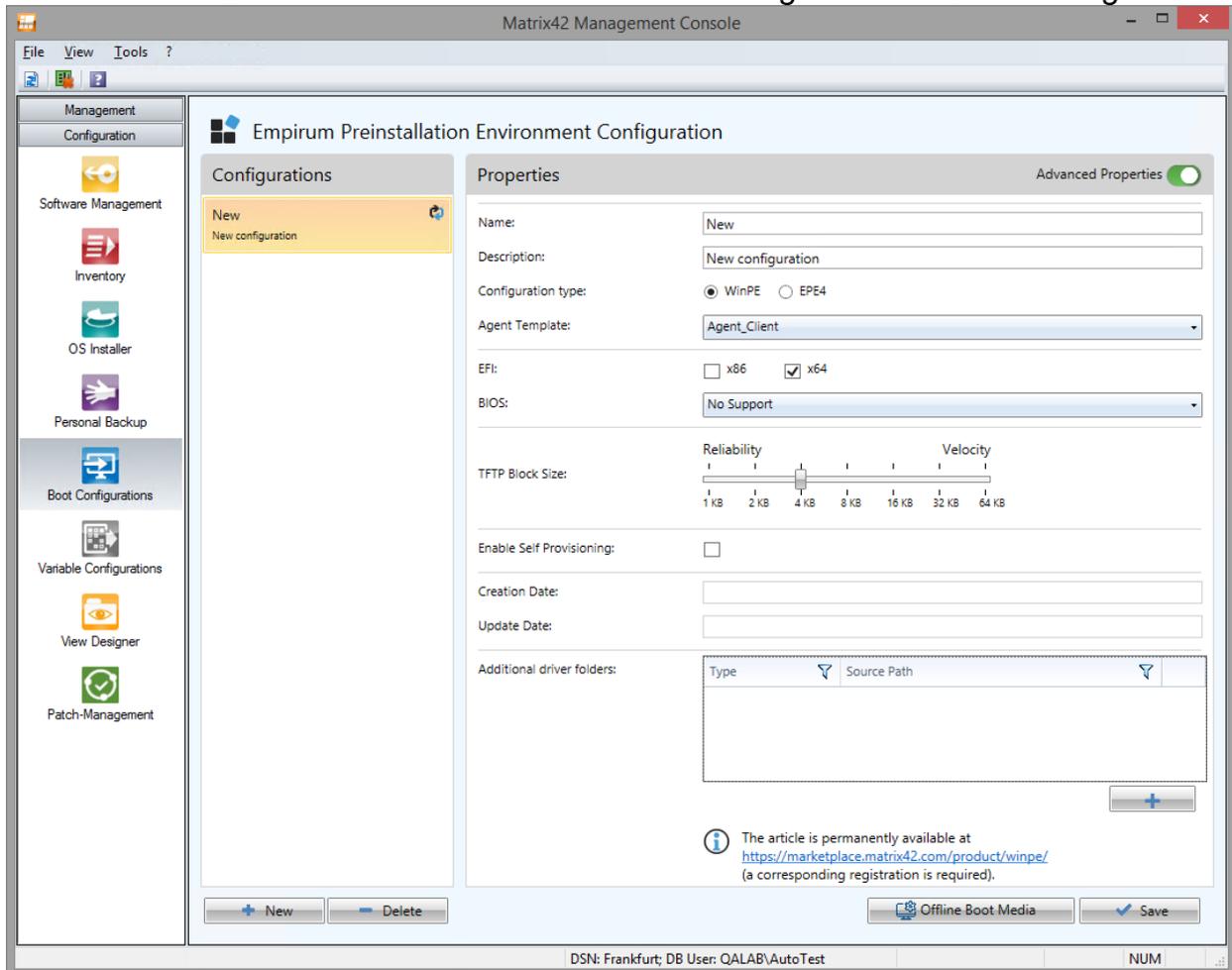
For our case in the step by step instructions, we will use a Dell workstation and integrate the necessary drivers.

Dell already provides the necessary mass storage drivers under the common name F6 driver. This name is derived from the F6 third-party storage driver installation method during Windows Setup.

As with the F6 installation method, you will need unpacked drivers to deploy the mass storage drivers (typically these consist of at least one of * .inf, * .cat, and * .sys files, often multiple files of those types may be included in a driver package).

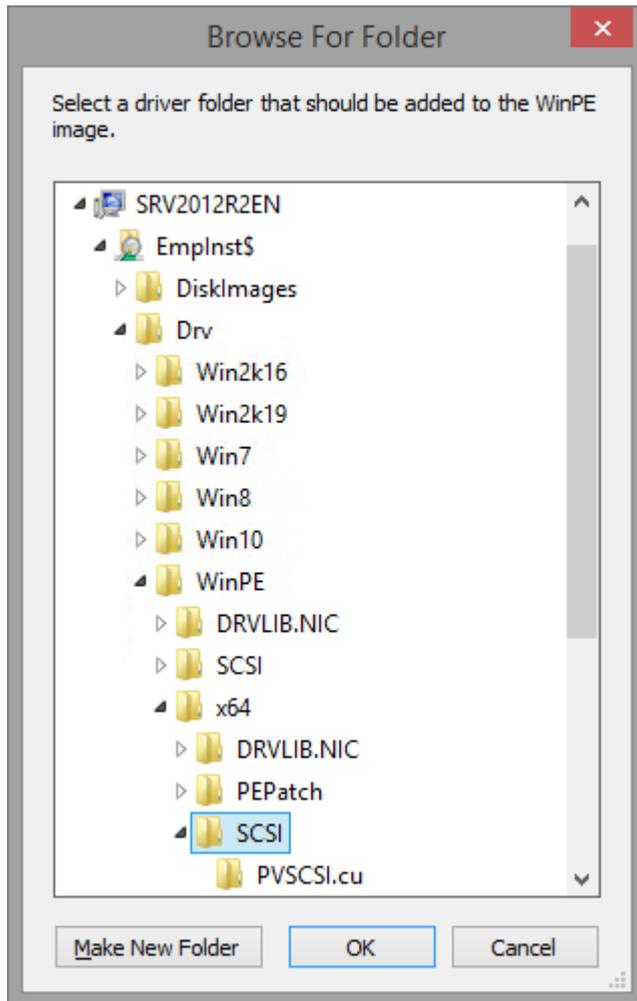
Matrix42 Driver integration for WinPE - HowTo

- 1 After ensuring that the drivers have been unpacked correctly in the desired location, switch to the *EMC* and select the menu item *Configuration > Boot Configurations*.



Note: It is recommended to create a new boot configuration and thus to test the deployment. After successful tests, the necessary drivers can be transferred to a productive boot configuration.

- 2 In the Boot Configuration overview, you will find "Additional driver folders" on the bottom right.
- 3 If already integrated driver directories exist for this boot configuration, they are displayed here. In our case, we create a new boot configuration and the window "Additional driver directories" shows nothing now.
- 4 Please select the  button and navigate in the file browser to the driver directory where you have previously unzipped the drivers.



Note: It is a good idea to always choose speaking names, so that you will be able to understand at a later date which driver should be replaced by a successor, but also to make it easier for other colleagues to identify the driver.

The selected directories must be located on a share, which can then also be accessed by the activation service. Make sure that the user under which the Activation Service runs has access to the share and that the share can be accessed from the computer running the Activation Service.

For the file storage and all path specifications, a maximum path length of **256** characters applies. The following characters ~ "# % & * : < > ? / \ { / } are not allowed in folder, file and path specifications.

The Emplnst\$ share must be used here.

- 5 In this case, we add the driver directories "SCSI" and "DRVLIB.NIC" to the boot configuration.

Matrix42 Driver integration for WinPE - HowTo

The screenshot shows the 'Empirum Preinstallation Environment Configuration' window. On the left, a 'Configurations' sidebar has a 'New' button and a 'New configuration' entry. The main 'Properties' panel is active, with 'Advanced Properties' toggled on. The configuration details are as follows:

- Name: New
- Description: New configuration
- Configuration type: WinPE (selected), EPE4
- Agent Template: Agent_Client
- EFI: x86 (unchecked), x64 (checked)
- BIOS: No Support
- TFTP Block Size: A slider set to 4 KB, with options from 1 KB to 64 KB.
- Enable Self Provisioning: (unchecked)
- Creation Date: (empty)
- Update Date: (empty)
- Additional driver folders: A table with two entries:

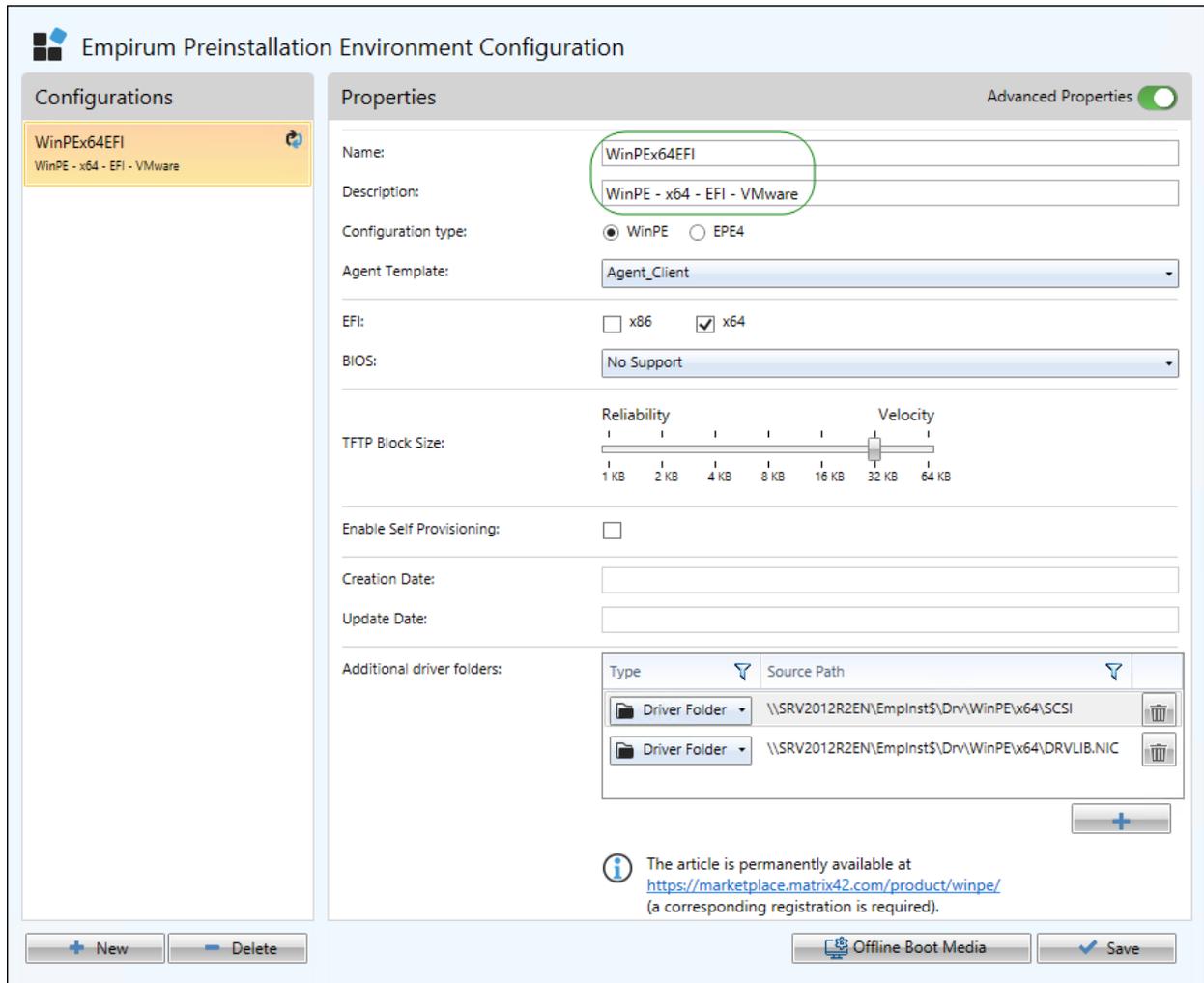
Type	Source Path
Driver Folder	\\SRV2012R2EN\EmpInst\$\Drv\WinPE\w64\SCSI
Driver Folder	\\SRV2012R2EN\EmpInst\$\Drv\WinPE\w64\DRVLIB.NIC

At the bottom, there is a note: 'The article is permanently available at <https://marketplace.matrix42.com/product/winpe/> (a corresponding registration is required)'. Navigation buttons include '+ New', '- Delete', 'Offline Boot Media', and 'Save'.

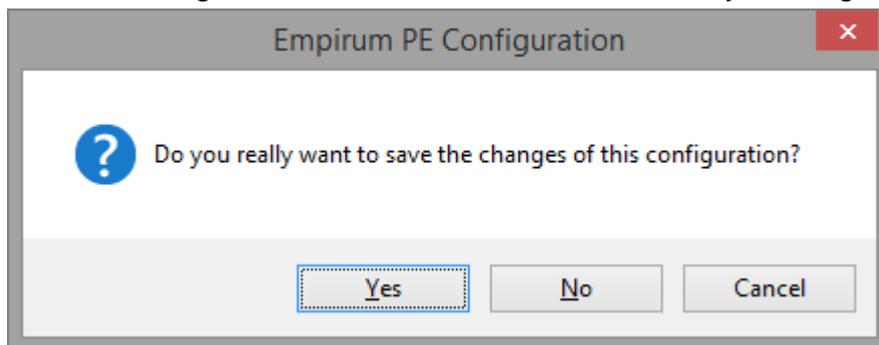
Note: Note that each driver directory must be added separately.

- 6 Enter a description in the description field, which facilitates later identification of the boot configuration.

Matrix42 Driver integration for WinPE - HowTo



7 After all changes have been made, confirm them by clicking on the "Save" button.



8 The WinPE boot configuration is now being created and can then be assigned to the client.

Note: The WinPE Agent tries to find suitable drivers for devices that are not supported by WinPE's own drivers. That Any existing WinPE driver will always be given priority over a subsequently added driver.

Matrix42 Driver integration for WinPE - HowTo

2.2 Reading Log File Information

The log file 'Matrix42.Platform.Service.Host.log' can be found on the client under 'x:\UAF' or on the server in the 'Auto\MAC' or 'Auto\UUID' directory. There you will find a section which indicates the missing device drivers or the installed devices if additional drivers have been stored in the boot configuration.

Here in the example an 'Ethernet Controller' is listed as 'Unsupported Device'. And you can see in the last lines that no devices have been installed, because no drivers have been stored in the boot configuration:

```
[Information] [PeAgent.LogUnsupportedDevices] =====
[Information] [PeAgent.LogUnsupportedDevices] Listing unsupported devices which require drivers...
[Information] [PeAgent.LogUnsupportedDevices] =====
[Information] [PeAgent.LogUnsupportedDevices] Unsupported device: 'Ethernet Controller'
[Information] [PeAgent.LogUnsupportedDevices] Hardware ids: PCI\VEN_15AD&DEV_07B0&SUBSYS_07B015AD&REV_01,
[Information] [PeAgent.LogUnsupportedDevices] =====
[Information] [PeAgent.LogUnsupportedDevices] Unsupported device: 'Base System Device'
[Information] [PeAgent.LogUnsupportedDevices] Hardware ids: PCI\VEN_15AD&DEV_0740&SUBSYS_074015AD&REV_10,
[Information] [PeAgent.LogUnsupportedDevices] =====
[Information] [PeAgent.LogUnsupportedDevices] Unsupported device: 'Video Controller (VGA Compatible)'
[Information] [PeAgent.LogUnsupportedDevices] Hardware ids: PCI\VEN_15AD&DEV_0405&SUBSYS_040515AD&REV_00,
[Information] [PeAgent.InstallDeviceDrivers] =====
[Information] [PeAgent.InstallDeviceDrivers] Installing required device drivers if provided...
[Information] [PeAgent.InstallDeviceDrivers] There were 0 devices installed.
[Information] [PeAgent.InstallDeviceDrivers] Finished installing required device drivers.
```

The hardware id can be used to search for a required driver.

If you have stored drivers in the boot configuration, you can see from the log that the appropriate drivers have been installed. Additionally, the driver files are listed, and which devices they support.

The appropriate drivers are then filtered for storage and network devices - everything else is not needed and therefore not installed. (Strictly speaking, only the device classes Volume, DiskDrive, Net, SCSIAdapter and HDC are used.)

And finally, it comes to the installation of the drivers, as you can see in the example below: The driver for the network device has been installed.

```
[Information] [PeAgent.InstallDeviceDrivers] =====
[Information] [PeAgent.InstallDeviceDrivers] Installing required device drivers if provided...
[Information] [DriverInfParser.ProcessInfFile] =====
[Information] [DriverInfParser.ProcessInfFile] Driver: x:\drivers\drivers_1\pvscsi\pvscsi.inf
[Information] [DriverInfParser.GetDeviceHardwareIDs] Device: VMware PVSCSI Controller (PCI\VEN_15AD&DEV_07B0&SUBSYS_07B015AD&REV_01)
[Information] [DriverInfParser.ProcessInfFile] =====
[Information] [DriverInfParser.ProcessInfFile] Driver: x:\drivers\drivers_1\vmwaresvgaii\vm3d.inf
[Information] [DriverInfParser.GetDeviceHardwareIDs] Device: VMware SVGA 3D (PCI\VEN_15AD&DEV_0405&SUBSYS_040515AD&REV_00)
[Information] [DriverInfParser.GetDeviceHardwareIDs] Device: VMware SVGA 3D (PCI\VEN_15AD&DEV_0405&SUBSYS_040515AD&REV_00)
[Information] [DriverInfParser.ProcessInfFile] =====
[Information] [DriverInfParser.ProcessInfFile] Driver: x:\drivers\drivers_1\vmwarevmcibusdevice\vmci.inf
[Information] [DriverInfParser.GetDeviceHardwareIDs] Device: VMware VMCI Bus Device (PCI\VEN_15AD&DEV_0740&SUBSYS_074015AD&REV_10)
[Information] [DriverInfParser.GetDeviceHardwareIDs] Device: VMware VMCI Host Device (ROOT\VMWVMCIHOSTDEV)
[Information] [DriverInfParser.ProcessInfFile] =====
[Information] [DriverInfParser.ProcessInfFile] Driver: x:\drivers\drivers_1\vmxnet3\vmxnet3ndis6.inf
[Information] [DriverInfParser.GetDeviceHardwareIDs] Device: vmxnet3 Ethernet Adapter (PCI\VEN_15AD&DEV_07B0&SUBSYS_07B015AD&REV_01)
[Information] [DriverLoader.FilterAvailableDriversByDeviceClasses] =====
[Information] [DriverLoader.FilterAvailableDriversByDeviceClasses] Filtering available drivers by: Volume, Net, DiskDrive, SCSIAdapter, HDC
[Information] [DriverLoader.InstallRequiredDrivers] =====
[Information] [DriverLoader.InstallRequiredDrivers] Installing available drivers for 1 devices.
[Information] [DriverLoader.InstallRequiredDrivers] Installing available driver: Ethernet Controller (PCI\VEN_15AD&DEV_07B0)
[Information] [DriverLoader.InstallRequiredDrivers] Driver for Ethernet Controller (PCI\VEN_15AD&DEV_07B0)
[Information] [PeAgent.InstallDeviceDrivers] There were 1 devices installed.
[Information] [PeAgent.InstallDeviceDrivers] Device: Ethernet Controller (PCI\VEN_15AD&DEV_07B0) - Driver: x:\drivers\drivers_1\pvscsi\pvscsi.inf
[Information] [PeAgent.InstallDeviceDrivers] Finished installing required device drivers.
```

Matrix42 Driver integration for WinPE - HowTo

In principle, unsupported devices are always listed - the installed devices are not listed by default. If you want to see all devices or more information about the state of the devices and the drivers included in the installation, you must change the configuration file 'Matrix42.Platform.Service.Host.exe.config' on the server under 'EmpInst\$\Sys\Images\WinPE\binaries\UAF' and rebuild related boot configurations.

To achieve that you must add the listener for the log file as XML node to the debug switch in the logging configuration:

At <loggingConfiguration><categorySources>

```
<add switchValue="Verbose" name="Debug">
  <listeners>
    <add name="OutputDebug TraceListener"/>
  </listeners>
</add>
```

you must add following listener

```
<add name="RollingFlatFile TraceListener"/>
```

So that the configuration looks like below and debug logs are written to the log file:

```
<add switchValue="Verbose" name="Debug">
  <listeners>
    <add name="OutputDebug TraceListener"/>
    <add name="RollingFlatFile TraceListener"/>
  </listeners>
</add>
```

3 Instructions for the Provision of Operating System Drivers

The following chapter first describes the concept of operating system driver integration. The following describes how to get the necessary information using the Matrix42 PreOS HardwareInfo package and how to integrate the information and drivers using the Matrix42 Driver Assitant.

3.1 Concept

Before we move on to provisioning operating system drivers, we'd like to discuss how hardware discovery works in our case and the mapping of a driver package based on it.

The Matrix42 PreOS package DriverIntegration is responsible for the integration of the operating system drivers. The package offers the possibility to decide which driver archive or driver folder should be integrated into the operating system.

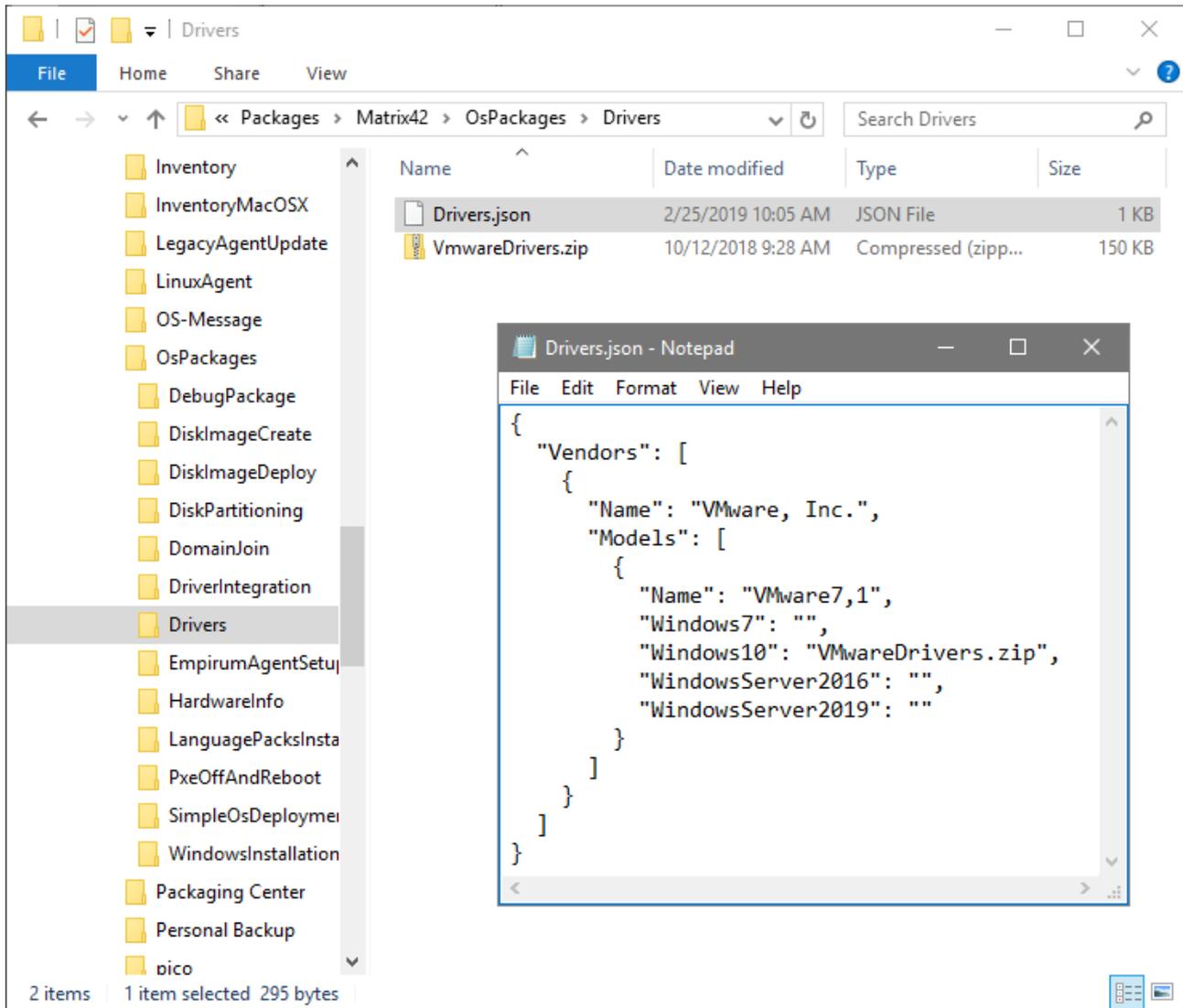
- ▶ Direct specification of the driver archive or driver folder via the computer variable *DriverIntegration.DriverArchiveFileName*
- ▶ Automatic determination of the driver archive or driver folder via the control file *Drivers.json* based on the vendor name, model name and the operating system type (*Windows 7, Windows 10, Windows Server 2016 or Windows Server 2019*)

The control file can be found on the Empirum server under:

```
\\%EmpirumServer%\Configurator$\PackagesMatrix42\OsPackages\Drivers\drivers.json
```

The mapping is based on the vendor name, the model name of a hardware and the assigned operating system source. Exactly this information is summarized in the *drivers.json*.

Matrix42 Driver integration for WinPE - HowTo



The names for vendor and model are not arbitrary. They are specified by WinPE at runtime via the Windows Management Instrumentation (WMI) classes (gwmi win32_computersystem) and must be the same. This information can also be obtained by executing the Matrix42 PreOS package HardwareInfo.

Note: With the DriverIntegration PreOS package since version 2.13 the model name in the *Drivers.json* does not have to match completely with the WMI read out model name. It is enough if the WMI read out model name starts with the model name in the *Drivers.json*.

Example:

WMI model name is **VMware7,1**

Model name in the *Drivers.json* is **VMware7**

This means that the driver specified at the model name **VMware7** is used, even if the name does not match completely.

If there are multiple model names in the *Drivers.json* that match the model name read from the WMI, the model with the greatest match in the name is selected.

Example:

WMI model name is **VMware7,1**

Model names in the *Drivers.json* are **VMware7** and **VMware**

This means that the driver specified at the model name **VMware7** is used, because it has a better match with the WMI read out model name.

Some manufacturers offer driver packages for several models. The only difference between the model names is the different name endings. With this modification the entries in the *Drivers.json* can be combined. It is no longer necessary to create an entry for each individual model variant.

3.2 Determination of the vendor and the model

In order to be able to execute the automated assignment of driver packages consistently, we need - as already mentioned - the vendor and model names as determined by the Windows Management Instrumentation.

For this purpose, we provide the Matrix42 PreOS package HardwareInfo, with that you can determine all necessary information to be able to enter it into the control file by using the Matrix42 Driver Assistant.

- 1 Assign a working WinPE boot configuration and the Matrix42 PreOS package HardwareInfo to the client.
- 2 Activate the client (software and PXE activation).
- 3 Start the client via PXE Boot afterwards.
- 4 After successful execution of the package, you will find a file in the form "debug_drivers_<client>.<domain>.json" in the Mac8 or UUID directory (\\%EmpirumServer%\EmpInst\$\Wizard\OS\Auto\{<Mac8>|<UUID>}) of the given computer.
- 5 The content can now be imported into the Drivers.json control file using the Matrix42 Driver Assistant. Chapter 3.3.4 *Starting the Matrix42 Driver Assistant* on page 15 describes the assistant and the import of the information in more detail.

3.3 Matrix42 Driver Assistant for WinPE

The Matrix42 Driver Assistant is a tool to customize the Drivers.json control file. The control file is used by the Matrix42 PreOS package DriverIntegration to determine the driver archive or driver folder to be integrated.

3.3.1 Prerequisites

- ▶ Matrix42 Client Management (v17.0.2 or higher).
- ▶ WinPE Support 1.4.14 or higher.
- ▶ PreOS package DriverIntegration 2.7 or higher:
 - ▶ To use the Driver Assistant, you need the updated PreOS package DriverIntegration 2.7 (or higher). DriverIntegration 2.7 (or higher) supports the JSON format which is

Matrix42 Driver integration for WinPE - HowTo

created by the Driver Assistant. Older versions of DriverIntegration do not support this JSON format.

3.3.2 Current functionality

The current functionality of Matrix42 Driver Assistant for WinPE

- ▶ Support of drivers for the Operating Systems *Windows 7, Window 10, Windows Server 2016, Windows Server 2019*
- ▶ Validation of the entries in the control file Drivers.json
- ▶ Import of driver entries created by the HardwareInfo PreOS package
- ▶ Import of driver entries via the previous version of the control file *Drivers.ini*
- ▶ Automatic migration of the previously used Drivers.ini to the new Drivers.json, if it is was not already present
- ▶ Importing driver files or directories
 - ▶ Reference to already imported driver files or directories via import
- ▶ Manual adjustment of the control file
 - ▶ Add, delete and customize vendor entries
 - ▶ Add, delete and customize model entries
 - ▶ Customize driver assignments

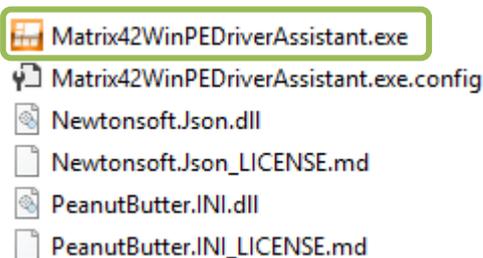
3.3.3 Limitations

The following limitations exist when using the Matrix42 Driver Assistant for WinPE:

- ▶ Currently the Driver Assistant can only be used on the Empirum master server.
- ▶ The created control file is only compatible with the current **DriverIntegration package version 2.7 or higher**.

3.3.4 Starting the Matrix42 Driver Assistant

To start the Matrix42 Driver Assistant, the Matrix42WinPEDriverAssistant.exe application must be started in the Empirum\AddOns\WinPEDriverAssistant directory.

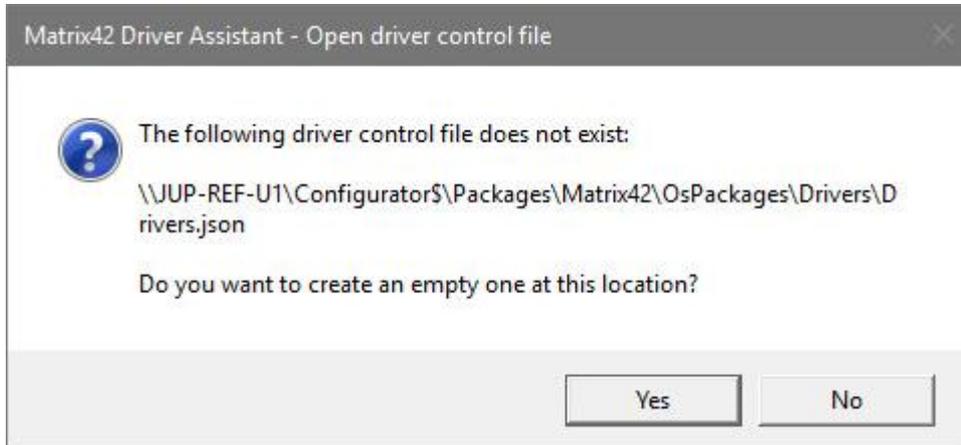


Note: Make sure that the user who starts the assistant has enough permissions to open, edit, recreate and delete the Drivers.json file on the Empirum share. The user should also have the permissions to copy files and folders there.

Matrix42 Driver integration for WinPE - HowTo

3.3.5 Creating a Drivers.json file at first startup

The Matrix42 Driver Assistant automatically tries to open the driver control file Drivers.json on the Empirum share at startup. If the driver control file does not exist, an empty driver control file will be created by confirming the following message with Yes.

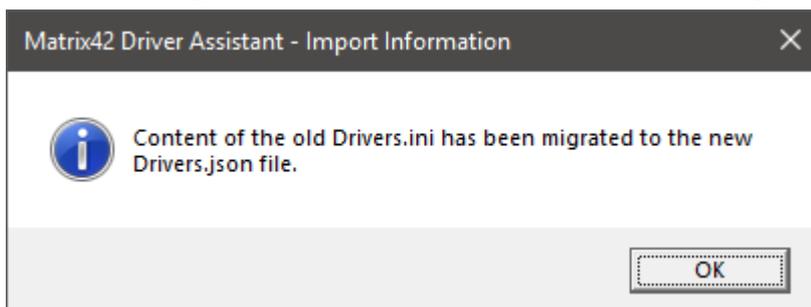


If a driver control file has already been created manually, this message will not appear and the Matrix42 driver wizard will open directly and display the contents of the file.

3.3.5.1 Migration of Drivers.ini Configuration from Previous Versions

If driver integration was already implemented in an earlier version via the Drivers.ini file, the import function can be used to transfer the entries of the Drivers.ini file to the new JSON format.

The migration of the previously used Drivers.ini to the new Drivers.json takes place automatically at the start of the tool if no Drivers.json already existed.



Note: As soon as no DriverIntegration package with version less than 2.7 is used anymore, the old Drivers.ini file can be safely removed from the server. DriverIntergration 2.7 and later uses a Drivers.JSON to map drivers, but it is also able to manage a Drivers.INI if the migration from INI to JSON has not yet taken place.

Chapter 3.3.7 *Importing HardwareInfo package output files* at page 17 describes the general procedure for importing. In this case, instead of importing an output file of the HardwareInfo package, you import the Drivers.ini file which is located on the Empirum server in the following directory:

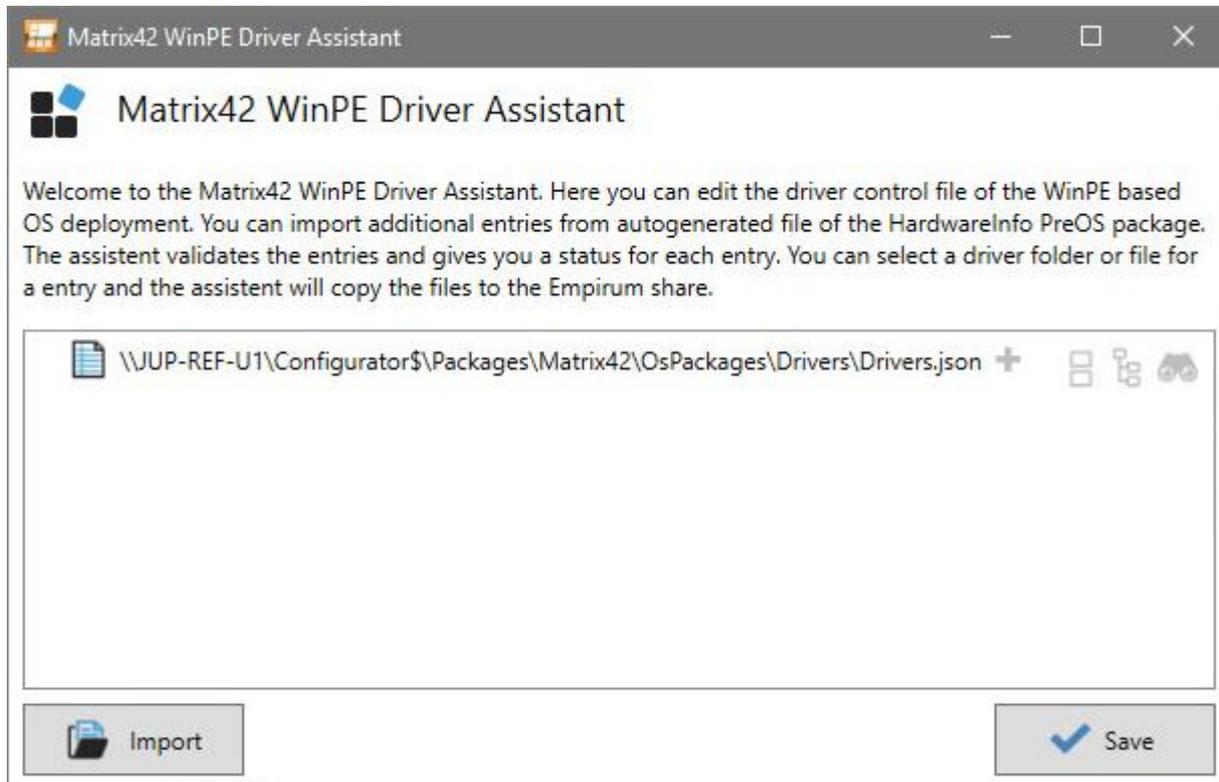
```
Configurator$\Packages\Matrix42\OsPackages\Drivers\Drivers.ini
```

Matrix42 Driver integration for WinPE - HowTo

In this case the driver file or directory specifications in the Drivers.ini file are always automatically assigned to the *Windows 10* operating system in the new JSON format.

3.3.6 The Matrix42 Driver Assistant interface

After creating an empty driver control file, the assistant opens and displays the opened file.



Because the driver control file does not contain any entries, no subordinate vendor or model entries are displayed.

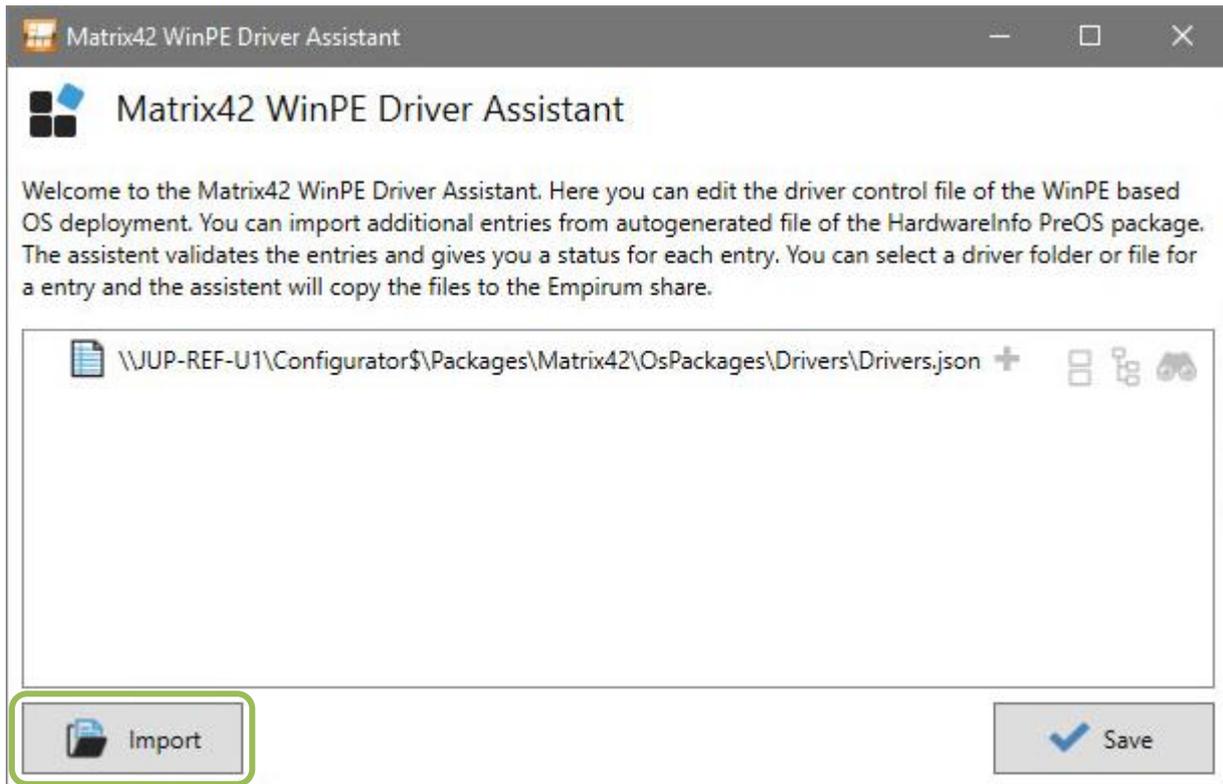
3.3.7 Importing HardwareInfo package output files

The easiest and safest way to extend the driver control file is to import *HardwareInfo* PreOS package output files, which the *HardwareInfo* package automatically creates in the `EmpInst$\Wizard\OS\Auto` subdirectory of the given client. Chapter 3.2 *Determination of the vendor and the model* at page 14 describes how to generate the output file with help of the *HardwareInfo* package.

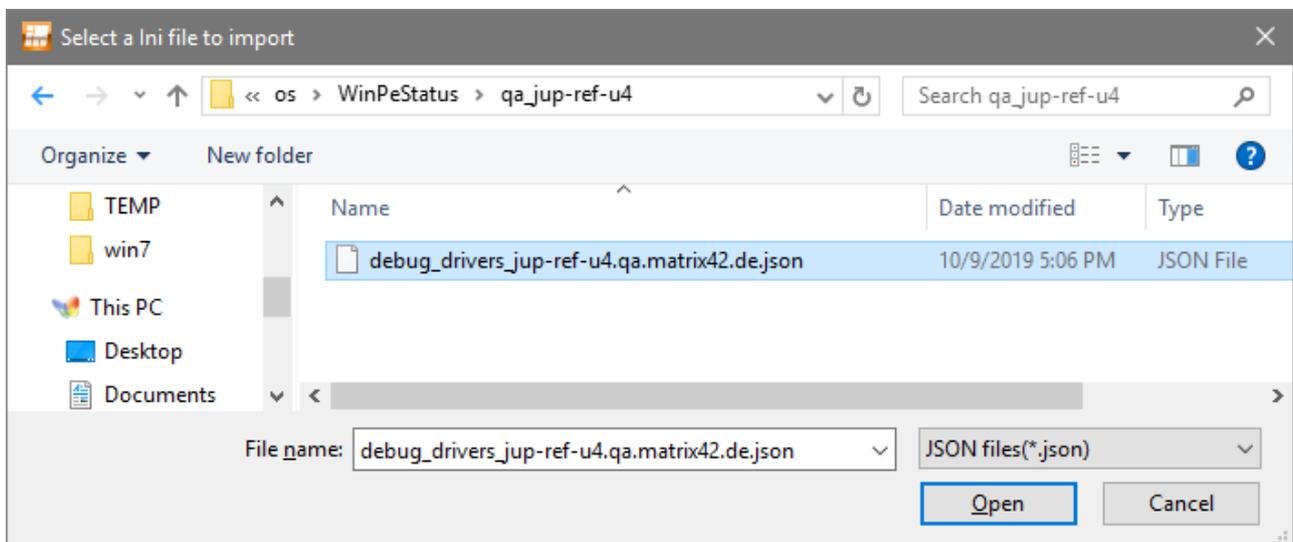
This file contains the information about the vendor and the model of the given client.

The import can be started via the *Import* button.

Matrix42 Driver integration for WinPE - HowTo



The file browser opens in the `EmpInst$\Wizard\OS\Auto` folder which can be used to open the output file of the *HardwareInfo* package.

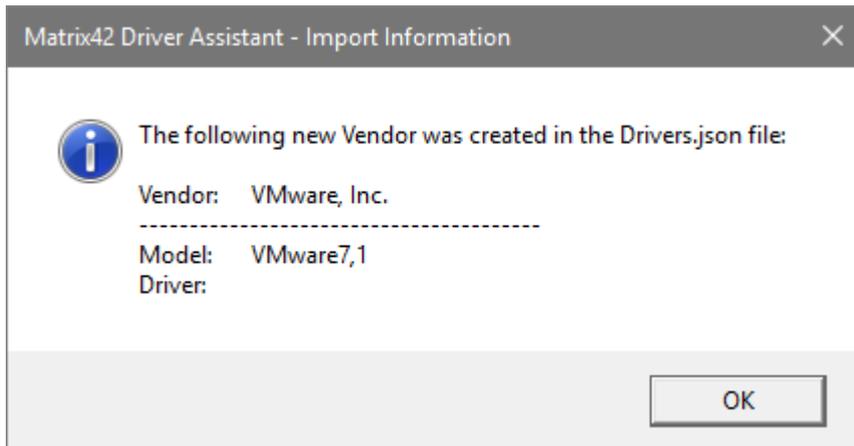


The output file of the *HardwareInfo* package is in the subfolder of the given client. This is either the Mac8 or the GUID folder.

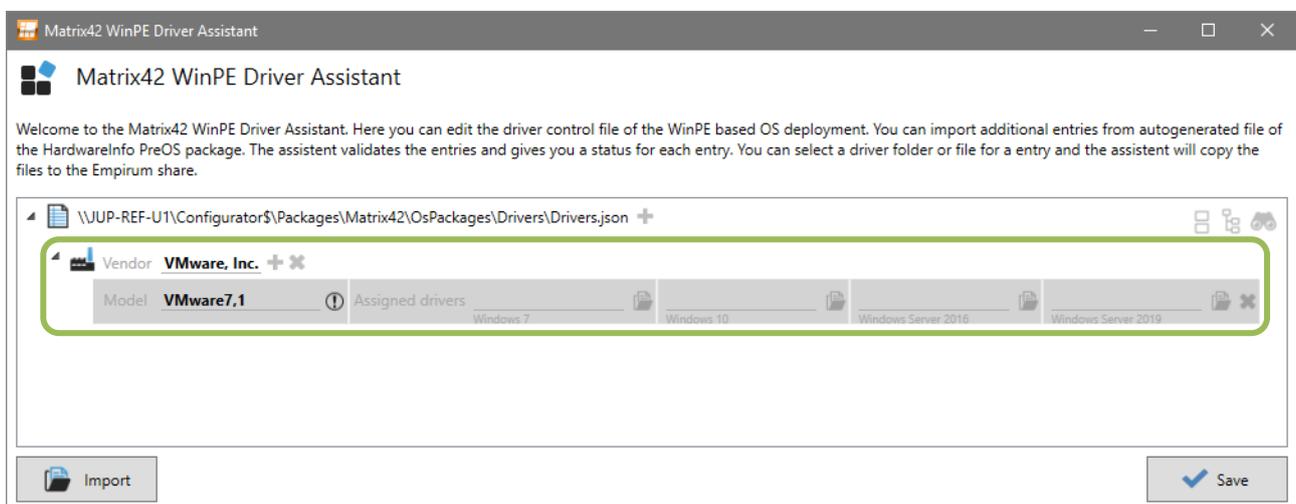
Note: You can also use the search function of the file browser to search for the client name. The output file of the *HardwareInfo* package always contains the client name and domain of the client.

Once the output file has been found and selected, the import can be started via Open. The wizard then gives feedback about which vendor and which model was imported.

Matrix42 Driver integration for WinPE - HowTo



After confirmation with OK the new entry is inserted into the control file.



An entry was created for the recognized vendor *VMware, Inc.* Subordinate an entry for the recognized model *VMware7,1* was created.

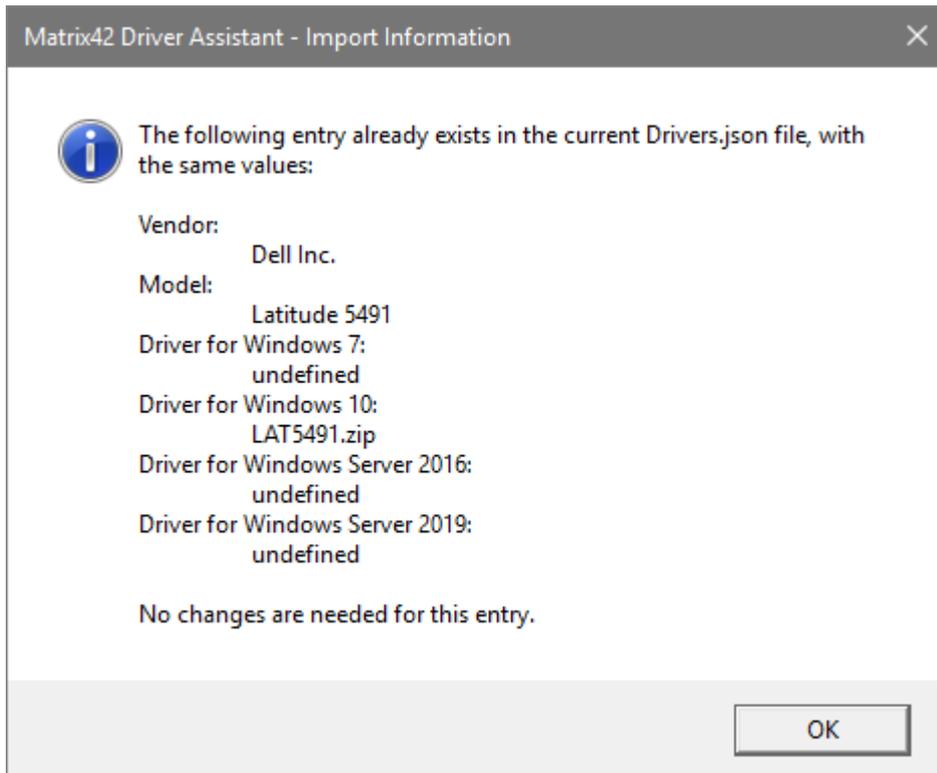
No driver files or driver directories are currently assigned to the operating systems *Windows 7*, *Windows 10*, *Windows Server 2016* or *Windows Server 2019* of the model entry. For this reason, the entry is highlighted in light gray and marked with an exclamation mark . This indicates that the entry is not yet valid because there is no driver is specified, or the specified driver is not available on the Empirum share.

3.3.7.1 Conflict situations at import

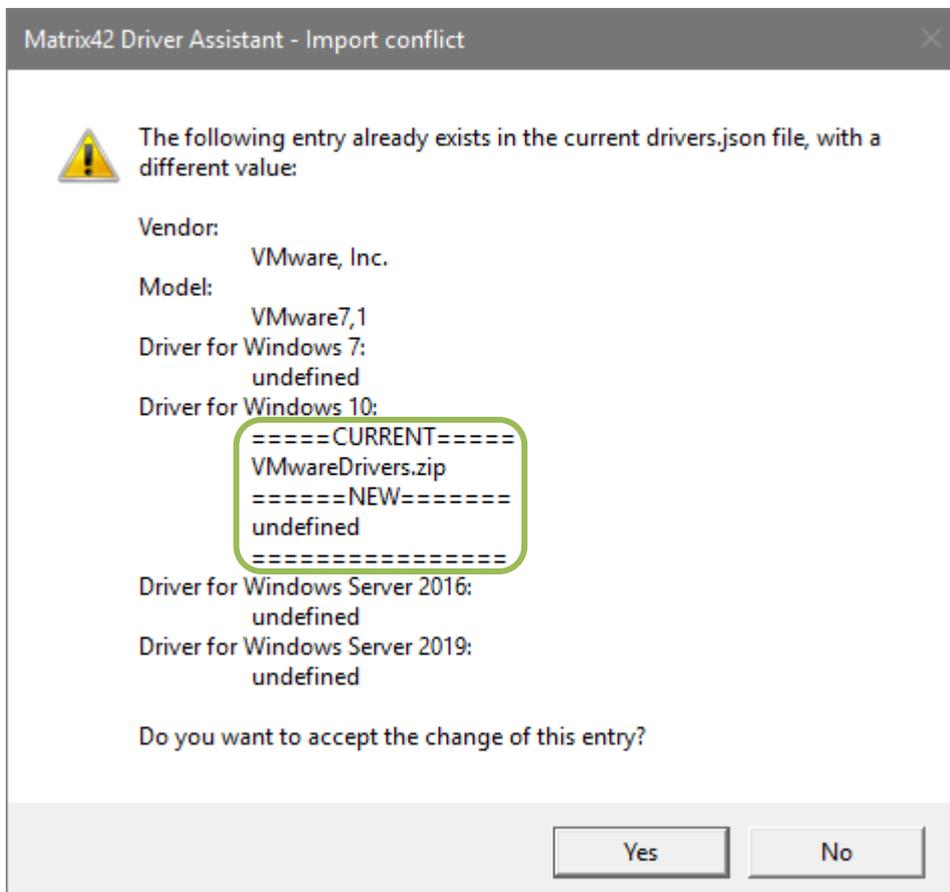
When importing HardwareInfo package output, it can happen that the models to import already exist in the driver control file

If the model values are identical to the existing values, the following message is displayed, indicating that no changes are necessary.

Matrix42 Driver integration for WinPE - HowTo



If a model is imported that already exists and where the driver specification differs, the following message is displayed.



Matrix42 Driver integration for WinPE - HowTo

Here it must be decided whether the existing driver specification is to be kept or whether the new specification is to be adopted. With *Yes* the changes are accepted. If you select *No*, the existing entries are kept.

Note: Driver specifications for the Windows 7, Windows Server 2016 and Windows Server 2019 operating systems always remain unchanged during import. These driver specifications remain unchanged even if the new specifications are accepted.

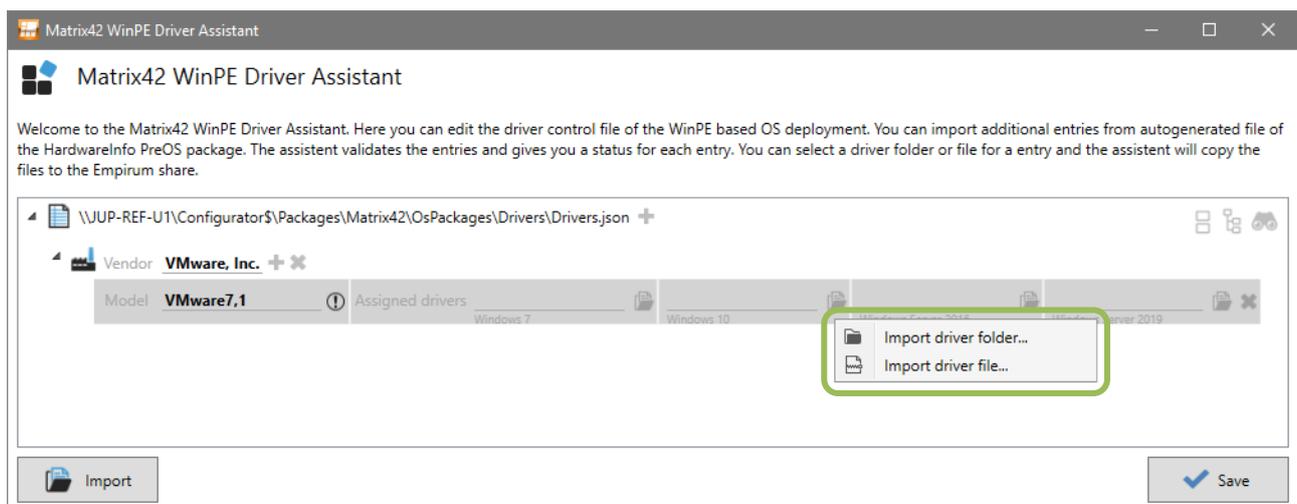
3.3.8 Import driver file

To assign a driver to the model entry, you can specify either a file name or a folder name directly in the entry of the corresponding operating system.

In this case you must make sure that the folder of the *Drivers.json* file contains the corresponding file or folder.

You can also press the driver import  button behind the corresponding operating system to select a driver file or folder to be copied to the Empirum share.

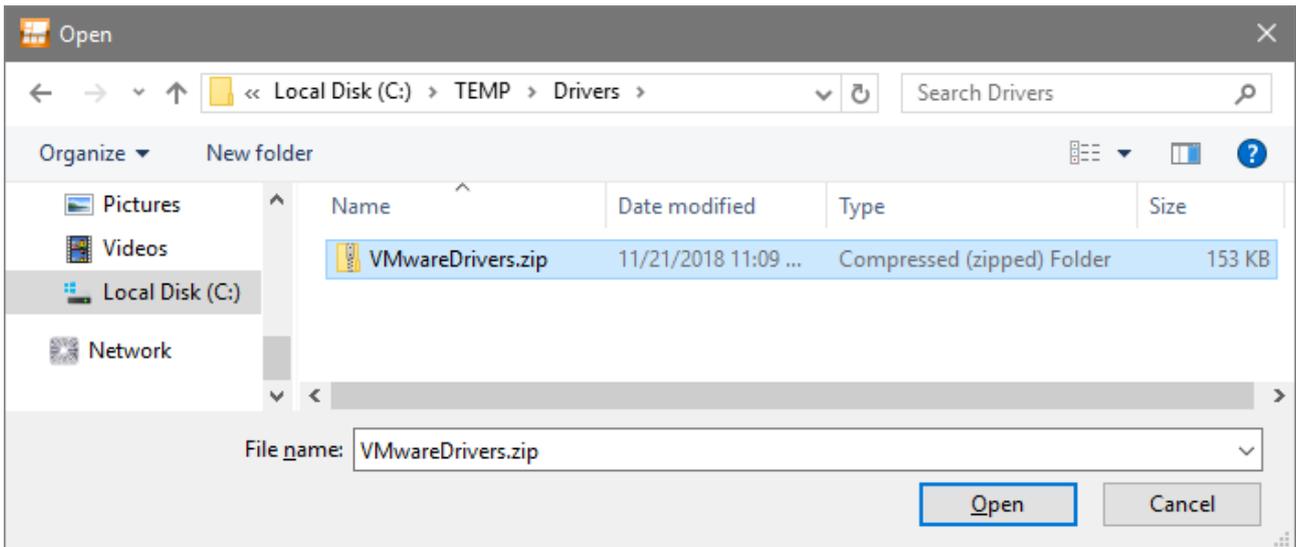
In this case the driver import button of the *Windows 10* entry is selected to import a driver for *Windows 10* to the model.



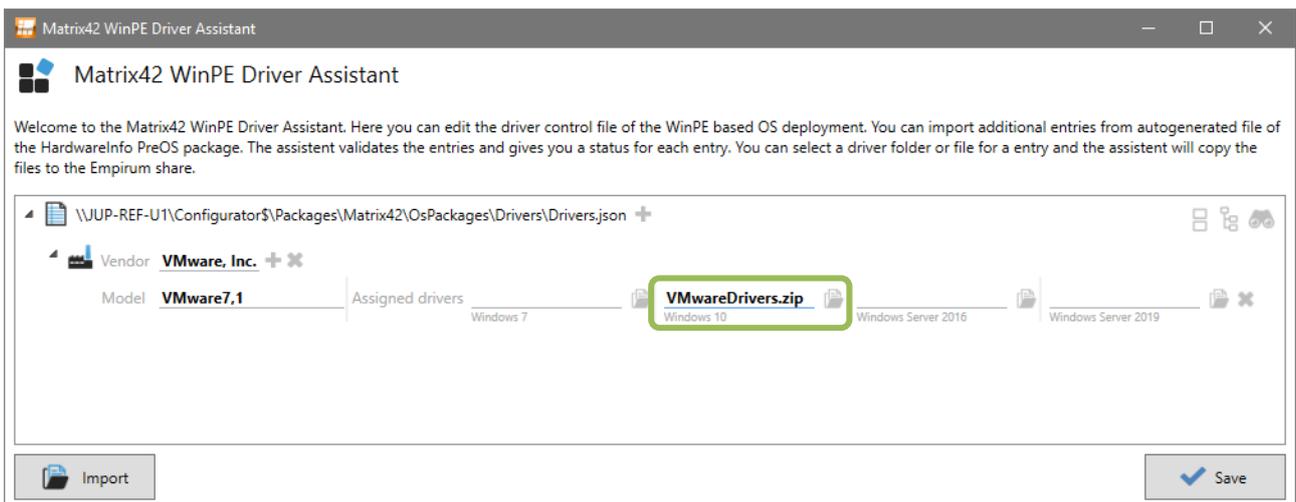
A context menu opens, which can be used to decide whether to import a single driver file or a complete driver directory. In this case, a single Zip archive is to be imported, therefore the *Import driver file...* entry should be selected.

The file browser opens, which can be used to select the driver archive.

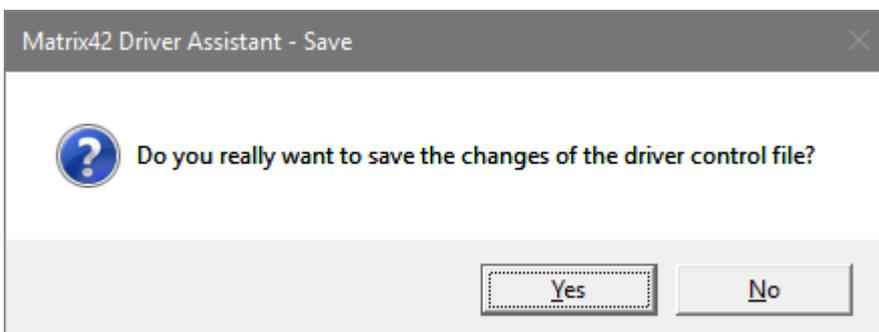
Matrix42 Driver integration for WinPE - HowTo



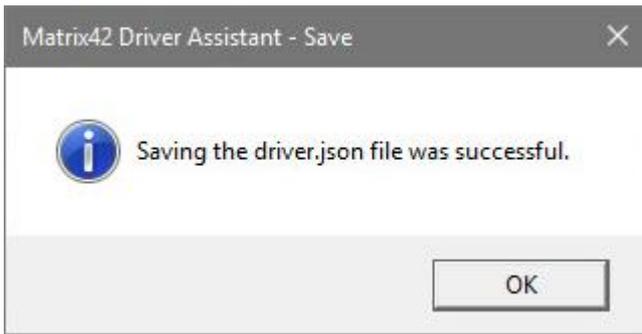
If the selection is confirmed with Open, the selected driver file is copied to the Empirum share and the Assigned driver value of the model entry is set.



You can now save the changes with the help of the Save button. The confirmation dialog opens.



Confirm the message with Yes, so the change will be saved in the driver control file on the Empirum server.



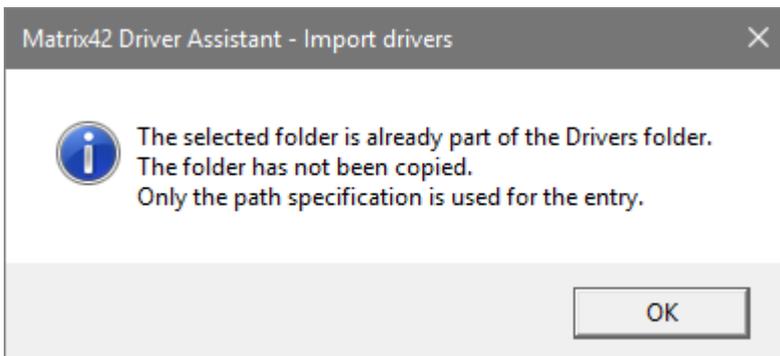
The confirmation of the successful saving process appears, click *OK* and this will return you to the Driver Assistant view.

3.3.8.1 References to already imported driver files or directories

You can also use the import function to create references to already imported driver files or driver directories.

To do this, simply select a file or directory from the Drivers directory when importing the driver file or directory. The Driver Assistant automatically detects that the file or directory to be imported comes from the Drivers directory.

In this case the file or directory will not be imported again. Only the correct path specification in the driver entry is applied. The following message appears during the import:

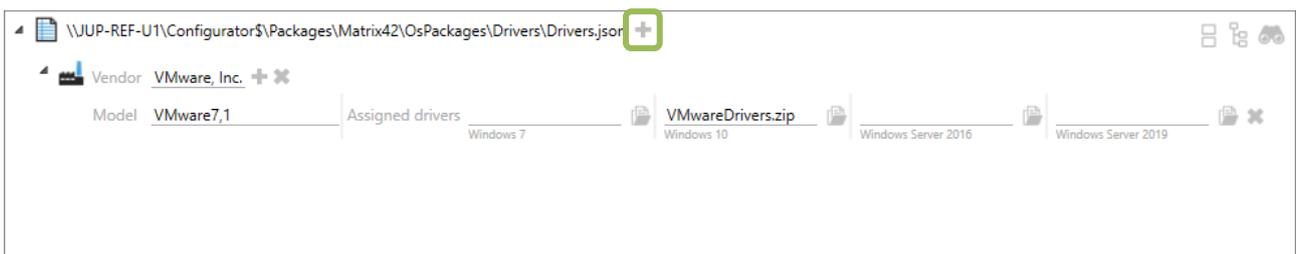


3.3.9 Manual extension

If there are no HardwareInfo package outputs, the driver control file can also be extended manually. For this the vendor name and the model name of a computer are necessary.

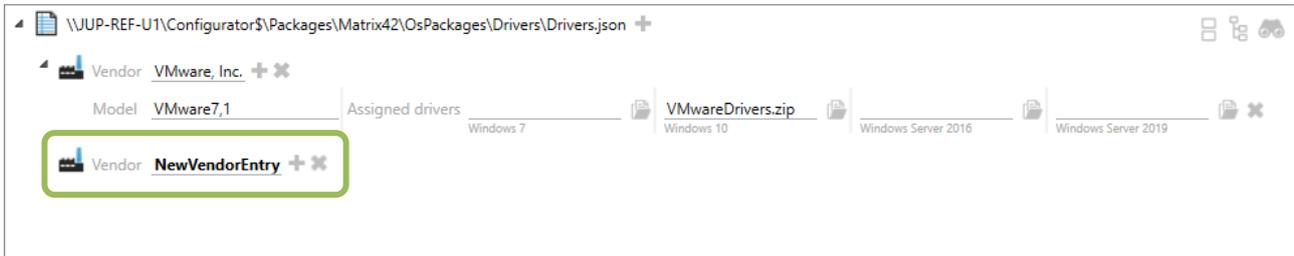
3.3.9.1 New vendor entry

By clicking **+** behind the driver control file entry a new manufacturer entry can be created.

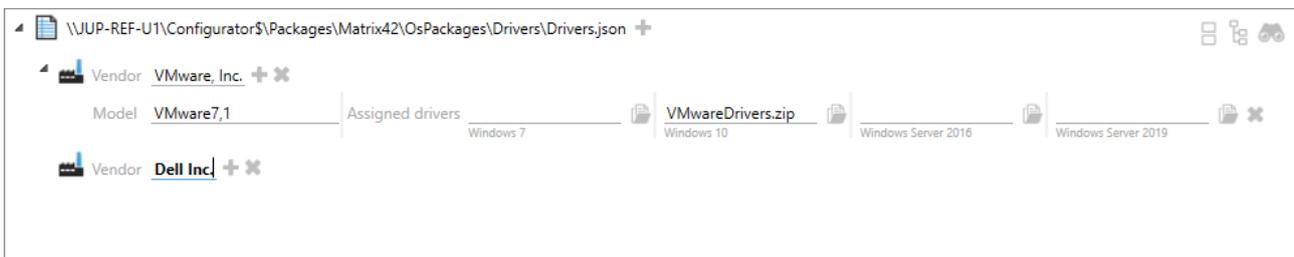


Matrix42 Driver integration for WinPE - HowTo

A new vendor entry with the default name *NewVendorEntry* appears below the structure. The vendor entry does not yet contain any model entries.

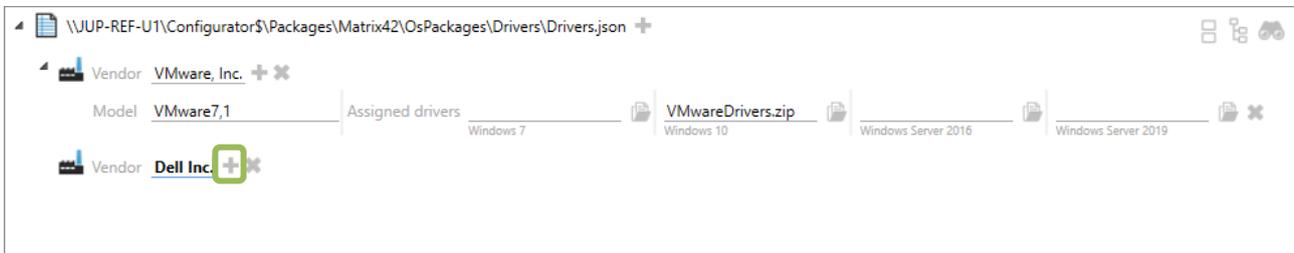


The vendor name can now be adjusted manually and extended with model entries. In this example the manufacturer *Dell Inc.* will be entered.

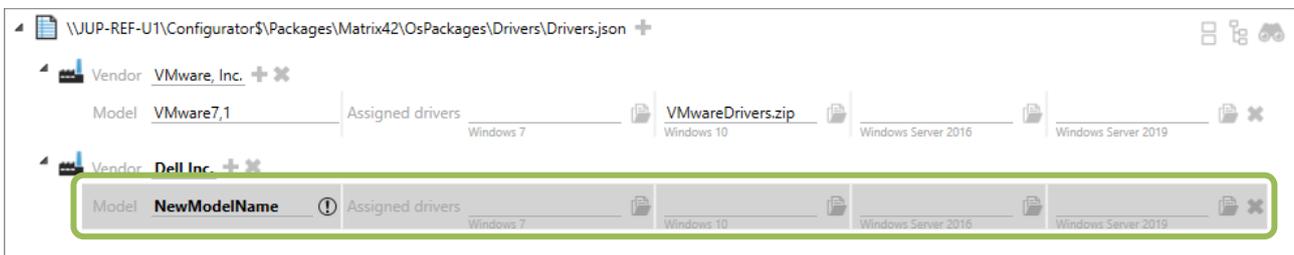


3.3.9.2 New model entry

By clicking **+** behind the vendor entry a new model entry can be created for this vendor.

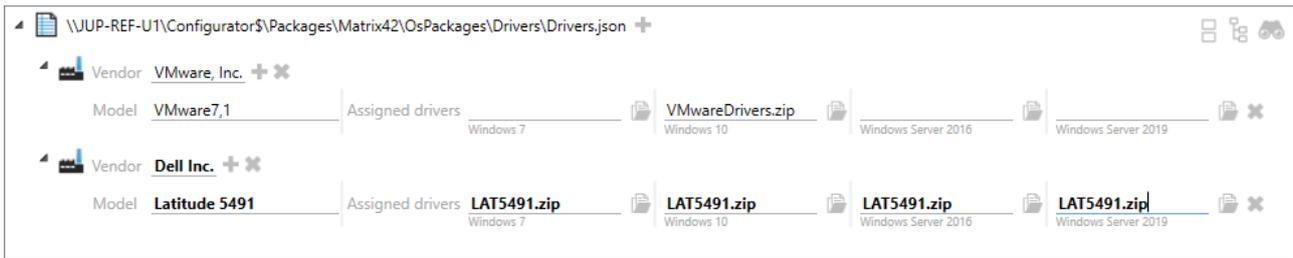


A new model entry with the default name *NewModelName* appears under the vendor.



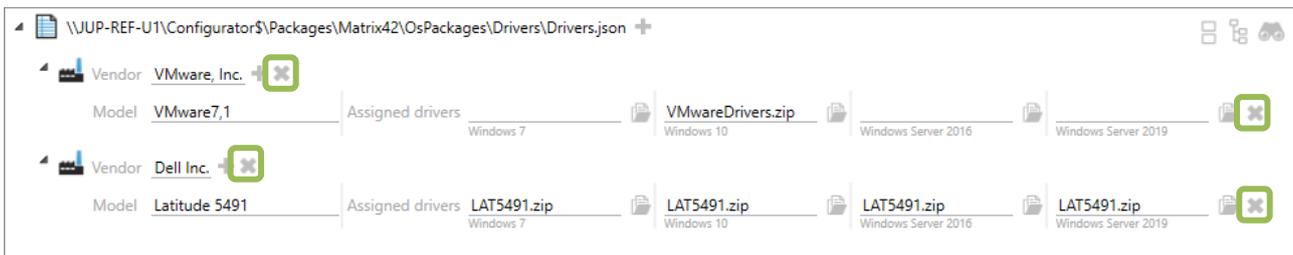
The model name can now be adjusted manually, and a driver can be assigned to the entry or imported. In this case the model name was *Latitude 5491* and all operating systems were assigned the driver archive *LAT5491.zip*.

Matrix42 Driver integration for WinPE - HowTo



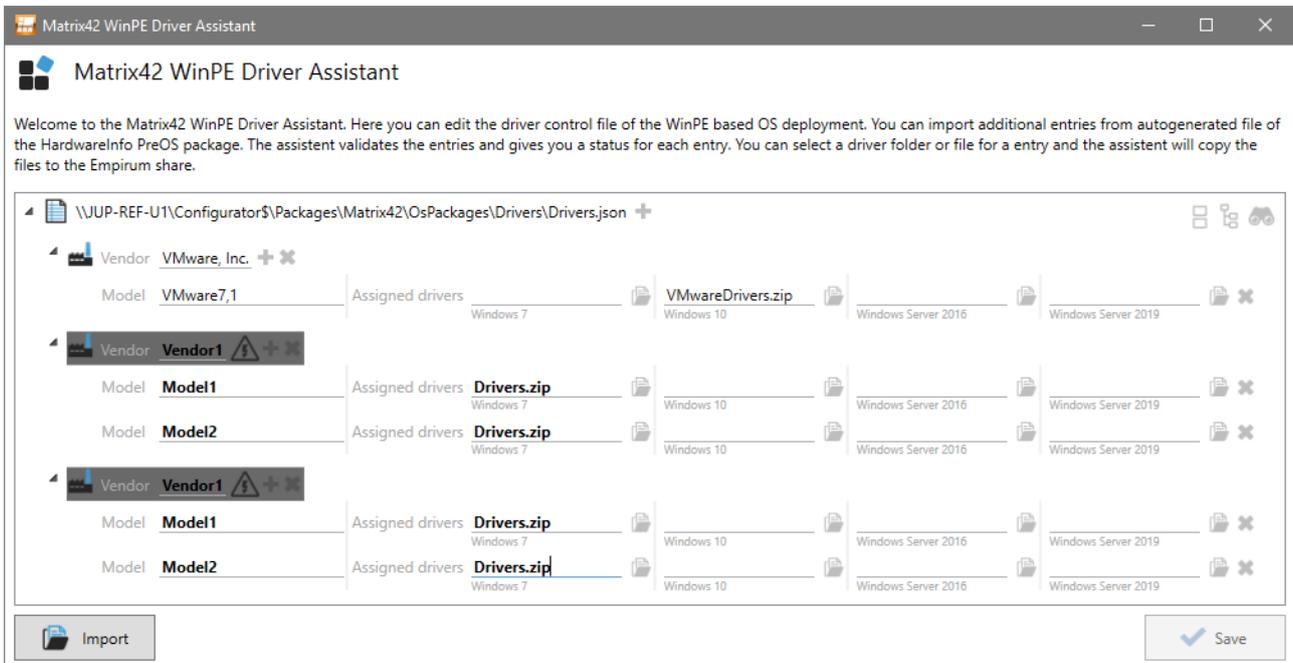
3.3.9.3 Removing Entries

If vendors, or model entries are no longer needed, they can be removed by clicking on the icon  after the desired entry.

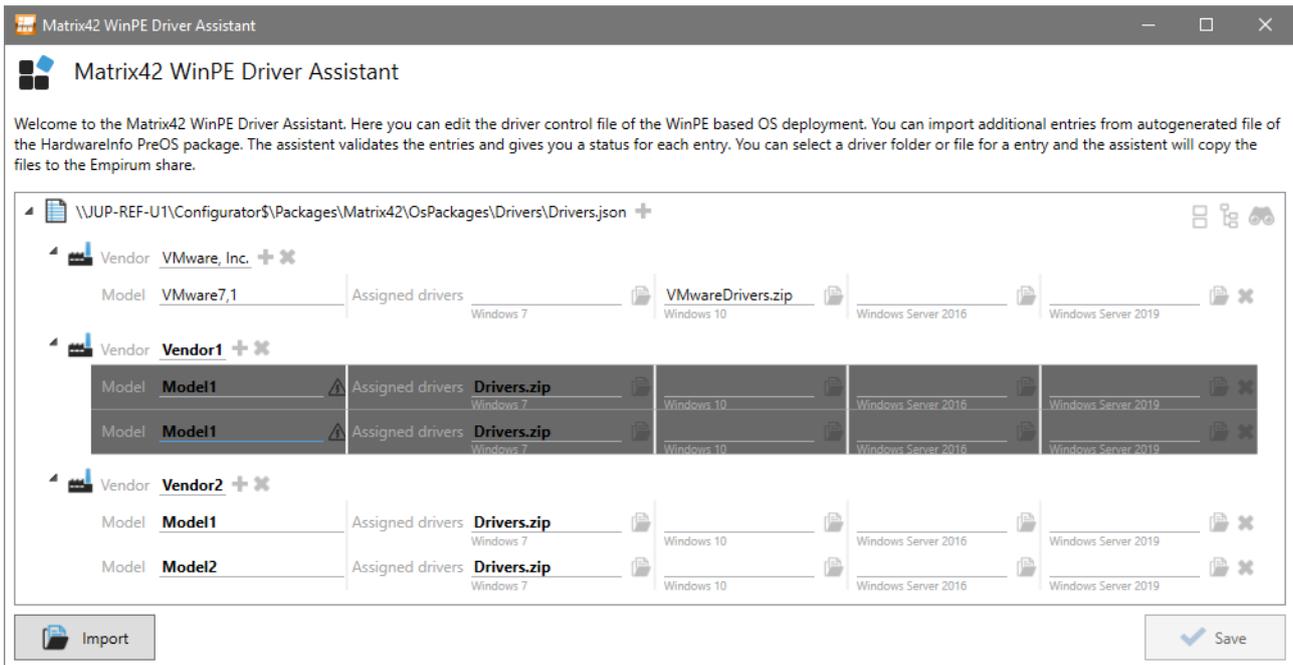


3.3.10 Conflicts

When manually extending the driver control file, conflicts can occur when creating vendor entries with the same name, or when creating model entries with the same name under a single vendor entry. In these cases, the entries are highlighted in dark grey. A warning icon  is also displayed.



Matrix42 Driver integration for WinPE - HowTo

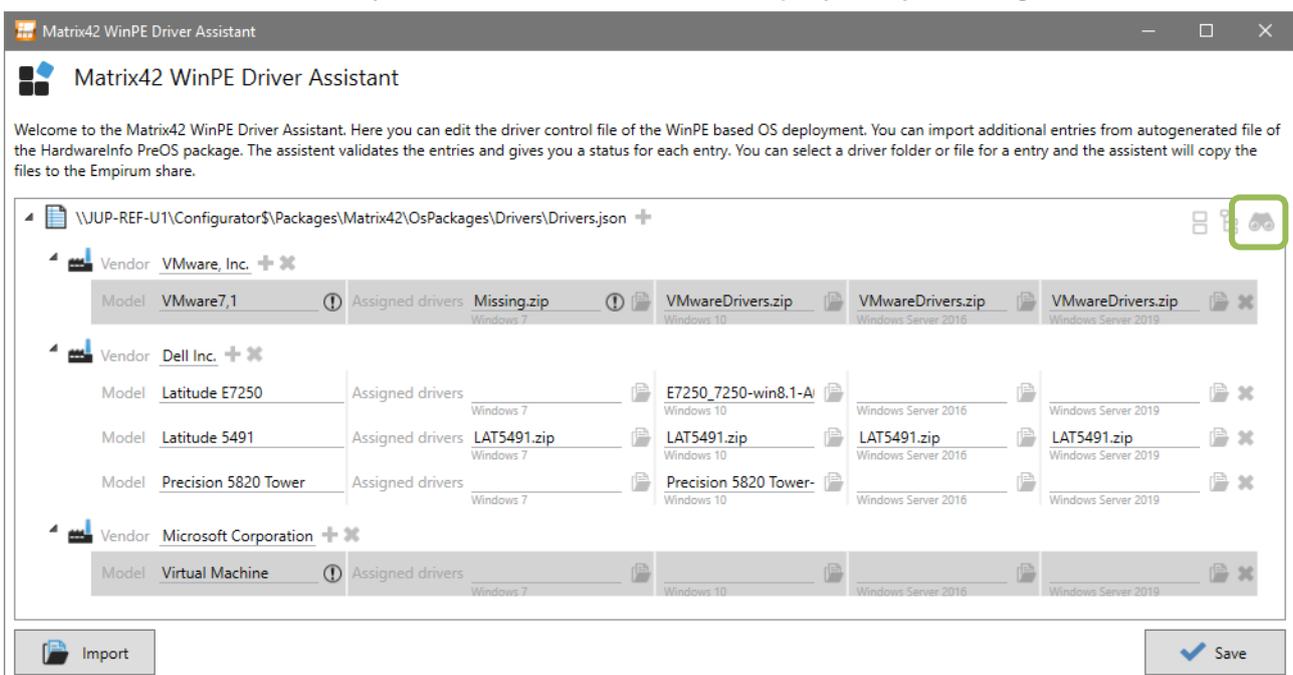


If there are conflicts in the driver control file, the file cannot be saved. The Save button is disabled in these cases.

3.3.11 Filtering functions

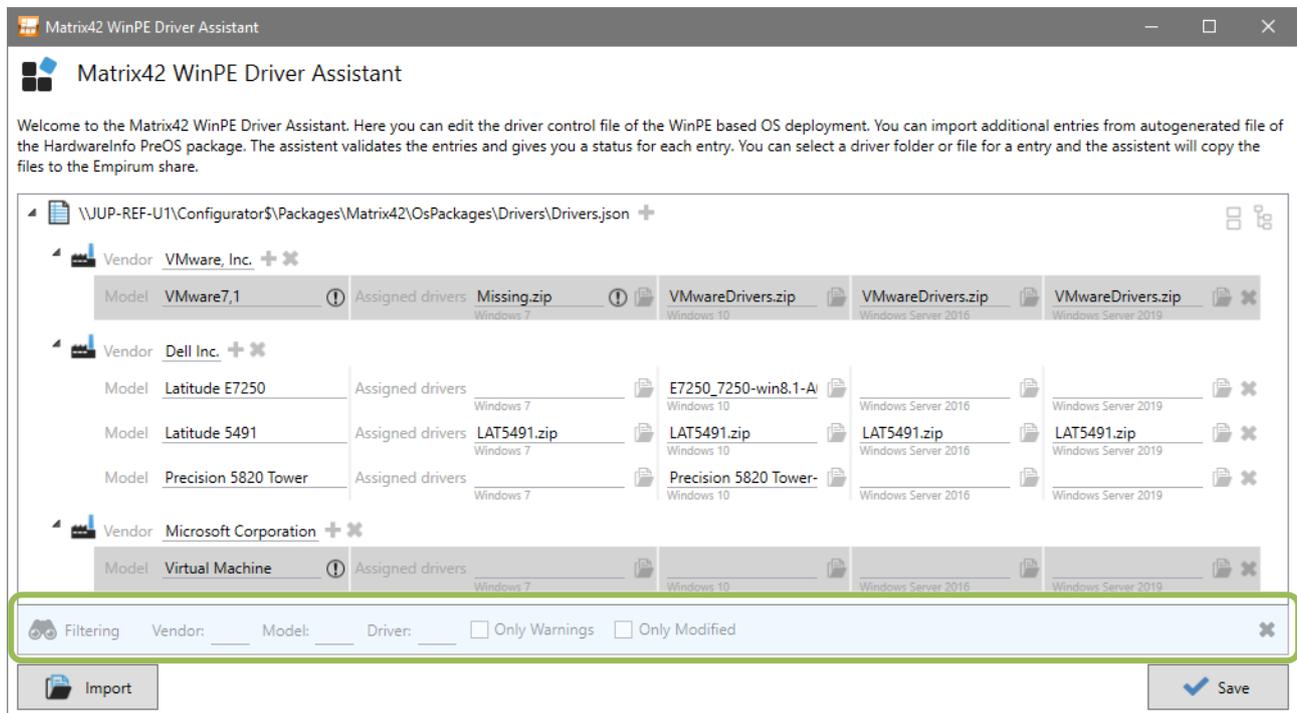
If the driver control file is extended and more and more entries are made for different vendors and models, the view quickly becomes confusing. In order to be able to find the correct entries quickly, a filter function has been provided, which can be used to filter out certain vendor, model or driver entries.

The filter function is initially deactivated and can be displayed by clicking the  icon.



Matrix42 Driver integration for WinPE - HowTo

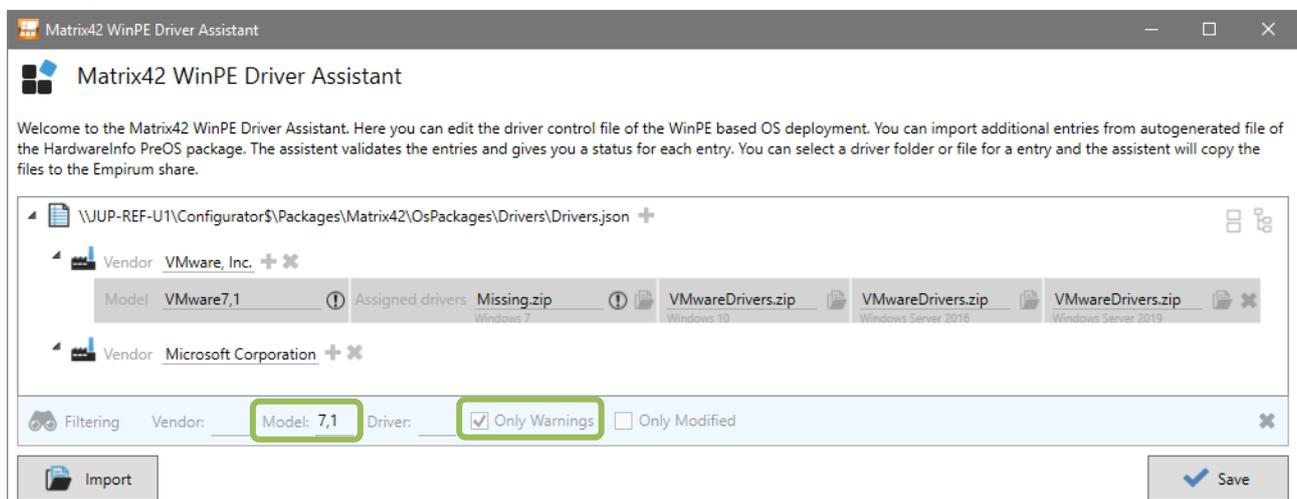
The filter bar appears at the bottom of the window and can be used to define the filter.



It is possible to filter by the following properties:

- ▶ Vendor name
- ▶ Model name
- ▶ Driver name
- ▶ Entries with warning (e.g. non-existent drivers, or conflicts)
- ▶ Modified entries

The filter properties can also be combined. In the following example, only entries that contain *warnings* are displayed. These are two entries for VMware and Microsoft. In addition, the display is restricted to the models contained in the name 7,1.



3.3.11.1 Deactivating filtering

If the filtering of the entries should be deactivated again, the filtering values and check boxes can be removed again, or you can close the filtering display directly via , so that all filtered entries are displayed again.



4 Injecting Drivers into a Boot Configuration

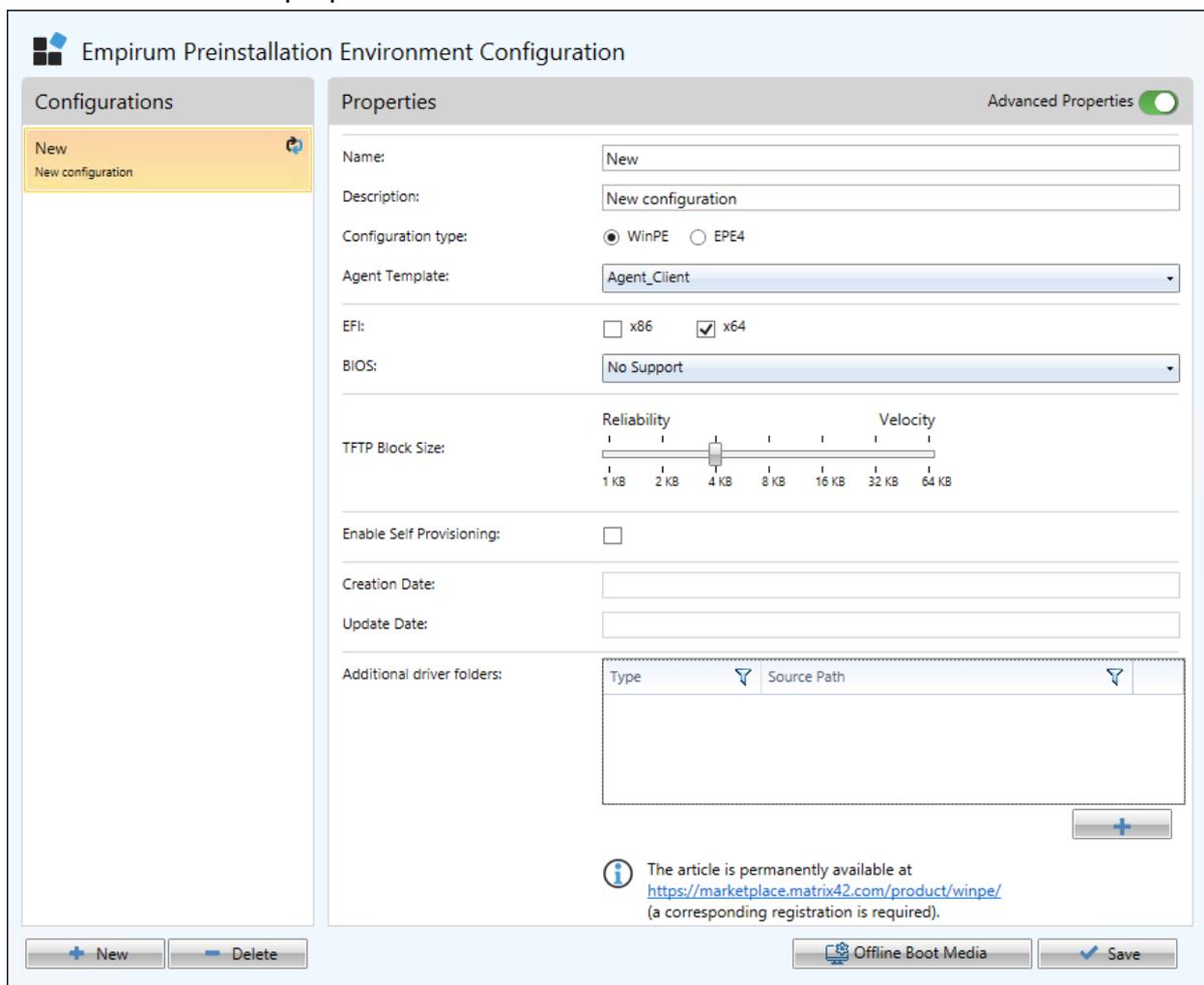
Some deployment scenarios may require another approach to deploying certain drivers.

Some devices need to be fully initialized before Windows HAL, devices like some docking stations that support mac address pass through as an example.

To allow an easy and comprehensive method of addressing such scenarios, WinPE based deployment by Matrix42 offers now the option to declare driver packages that are to be injected into the boot image.

In the following, short, step-by step guide we will add an exemplary driver to be injected into the boot-image.

- 1 We create a new boot configuration.
- 2 Switch to advanced properties.



Note that the user interface has changed slightly.

We will go into details once we added the drivers.

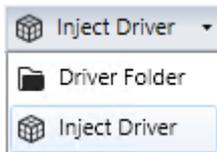
- 3 Now add a driver to the boot configuration by clicking on the  icon below the driver list and selecting the desired driver.

Added drivers can now be deleted by clicking the recycle bin icon on the right side of the driver list.

Matrix42 Driver integration for WinPE - HowTo

Type	Source Path	
Inject Driver	\\SRV2012R2EN\EmpInst\$\Drv\WinPE\x64\VMware\VMCIBusDevice	
Driver Folder	\\SRV2012R2EN\EmpInst\$\Drv\WinPE\x64\SCSI\PVSCSI.cu	

A driver was added to the boot image, in this case a VMware *VMCIBusDevice* driver.



This driver is marked as "*Inject Driver*" type and will be injected into the boot configuration afterwards.

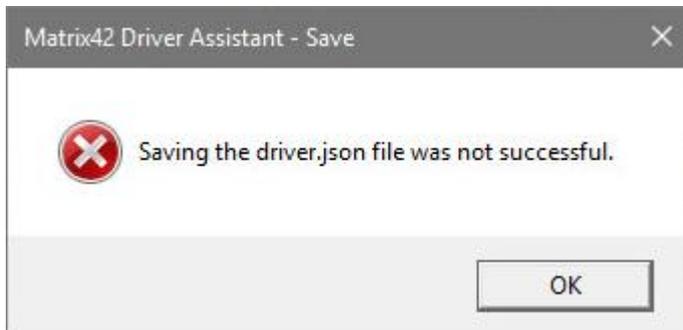
Another driver has also been added, in this case a *PVSCSI* driver, to show the visual difference between the two driver integration options.

Note that per default, a driver that is newly added will be declared a "Driver Folder" type driver, meaning that the driver will be loaded into WinPE during the runtime.

To change the Type of the driver, click on the list item, that will bring up a pulldown menu, where you can change the driver Type for this specific boot image.

5 FAQ

Question: I get the error message 'Saving the driver.json was not successful'. What could be the reason?



Answer: This is probably due to the permissions of the user who started the Matrix42 Driver Assistant.

Make sure that this user is allowed to create, delete and modify files and directories in the directory:

```
\\%EmpirumServer%\Configurator$\Packages\Matrix42\OsPackages\Drivers
```

Question: I have modified the control file using the Matrix42 driver assistant. However, the drivers are not used. In the log there is an error "Copy-Item: Illegal characters in path". What could be the reason?

Answer: Make sure you are using a current version of the DriversIntegration package (at least 2.6). Older versions of the package will not work with the slightly modified specification of the driver that should be used.

Matrix42 AG

Elbinger Straße 7

60487 Frankfurt am Main

Germany

Phone: +49 (0)69 6677 38 380

Fax: +49 (0)69 6677 88 657

E-mail: info@matrix42.de

Web: <https://www.matrix42.com/en/>