Matrix42

Matrix42 OS Deployment per WinPE - How-To

Version 1.8.6

23. September 2020

Inhaltsverzeichnis

1 Über dieses Dokument 4
1.1 Voraussetzungen 4
1.2 Aktuelle Funktionalität 4
1.3 Einschränkungen 5
2 Schritt für Schritt Anleitung 6
2.1 Import von Matrix42 PreOS-Paketen 6
2.1.1 Import der notwendigen PreOS-Pakete 7
2.1.2 Freigabe der Matrix42 PreOS-Pakete (bis Version 1.8.0) 11
2.2 Erstellung einer WinPE basierenden Bootkonfiguration 13
2.2.1 Erstellen einer Boot Konfiguration 14
2.2.2 Generierungsprozess des PXE Images 18
2.3 Zuweisung in der Administration 19
2.3.1 Paketreihenfolge in der Depotkonfiguration 20
2.3.2 Globale Variablen 21
2.3.3 DiskPartitioning Paket 22
2.3.4 DriverIntegration Paket 24
2.3.5 WindowsInstallation Paket 24
2.3.6 LocalExperiancePacksInstallation-Paket 28
2.3.7 PxeOffAndReboot-Paket 28
2.3.8 LanguagePacksInstallation-Paket 29
2.3.9 DomainJoin-Paket 30
2.3.10 EmpirumAgentSetup-Paket 32
2.4 PXE-Log 35
3 Matrix42 OS Deployment mit WinPE im Empirum Depot Server-Szenario 36
3.1 Agent-Template im Empirum Depot Server-Szenario 36
3.1.1 DHCP-Optionen 36
3.2 Zusätzliches Empirum Sync-Template 38
4 Treiberintegration für die Betriebssysteminstallation 39
4.1 DriverIntegration Paket 39
4.2 Ermitteln des Herstellers und des Modells 40
4.3 Starten des Matrix42 Treiberassistenten 40
4.3.1 Erstellen der Datei Drivers.json beim ersten Start 40
4.3.2 Import von HardwareInfo-Paket Ausgabedateien 41
4.3.3 Treiberdatei importieren 43
5 SimpleOsDeployment Paket 47
6 Disk Imaging 48

6.1 DiskImageCreate 49
6.2 DiskImageDeploy 52
7 WinPE-basiertes Computer Self Provisioning 56
7.1 Computervorlagen 57
7.1.1 Erstellen von Computervorlagen 57
7.1.2 Konfigurieren und zuweisen von Computervorlagen 59
7.2 Empirum-API Dienst konfigurieren und installieren 60
7.3 Bootkonfigurationen beim Computer Self Provisioning 61
7.3.1 Zusätzliche Empirum-API Verbindungsdaten 62
7.4 Computer Self Provisioning auf dem Empirum Server aktivieren 63
7.4.1 Registry-Anpassungen für den Empirum PXE-Dienst 63
7.5 Computer Self Provisioning am Computer ausführen 64
7.6 Self Provisioning über Depot Server (Offline) 70
8 WinPE-basiertes OS Deployment via http(s) 73
8.1 Aktuelle Einschränkungen 73
8.1.1 http(s) Unterstützung bei selbsterstellten PreOS Paketen 73
8.2 WinPE basiertes Deployment via http(s) einrichten 74
9 End Of Life (EOL) per WinPE 82
9.1 End Of Life konfigurieren (EOL) 82
10 FAQ 85

1 Über dieses Dokument

Es beschreibt zunächst die Voraussetzungen für das Matrix42 OS Deployment basierend auf WinPE. Anschließend folgt eine Schritt-für-Schritt-Anleitung, wie das Deployment vorbereitet und durchgeführt wird.

1.1 Voraussetzungen

Achtung: Matrix42 Client Management (v18.0.3 oder neuer).

- Ein Installiertes Windows ADK für Windows 10 auf dem Empirum Master Server, für die WinPE Erstellung.
- Der Empirum Master Server läuft auf einem Windows Server 2012 R2 oder Windows Server 2016 / 2019 Betriebssystem.
- Die Backend Task Queue benötigt dies für die Erstellung der WinPE Boot Images.

1.2 Aktuelle Funktionalität

Die aktuelle Funktionalität von Matrix42 OS Deployment basierend auf WinPE:

- Import von Matrix42 PreOS-Paketen
- Erstellung einer WinPE basierenden Bootkonfiguration
- Zuweisung und Konfiguration des Matrix42 PreOS-Paketes mit Hilfe von Computervariablen
- Die Installation des Matrix42 Software Depot nach dem Deployment
- Deployment auf UEFI oder Legacy/Bios basierenden Computern
- Domain Join des Computers und Zuordnung zur Organizational Unit im Active Directory
- Unterstützung von DHCP Optionen
- Auswahl der zu installierenden Empirum Agenten-Version
- http(s) als Transportprotokoll
- Anzeige der aktuellen Aufgaben auf dem Client während der Installation
- End of Life (EOL)
- Self Provisioning per http(s) ab der WinPE PreBoot Support Version 1.8.3

Ab Empirum v18.0.2:

- Zuweisung von Betriebssystem Editionen
- Language Pack Unterstützung

Ab Empirum v19.0.0:

- Zuweisung und Deployment von Windows 7, Windows Server 2016 und Windows Server 2019 Betriebssystemen

Ab Empirum v19.0.1:

- Computer Self Provisioning basierend auf Computervorlagen **(über den Empirum-API Dienst auf dem Empirum Master Server)**

Ab Empirum v20.0.0:

- Automatisches Einreihen von Matrix42 PreOS-Paketen beim Import in der Depotkonfiguration an bereits vorhanden Vorgängerversionen.

Ab Empirum v20.0.1:

- Verwendung von statischen IP Adresse beim OS Deployment. Die Verwendung der Matrix42 PreOS-Pakete *WindowsInstallation 5.3* und *DriverIntegration 2.14* oder höher wird dafür vorausgesetzt.

Ab Empirum v20.0.2:

- Konfiguration der Fingerabdrücke der Serverzertifikate für die https-Kommunikation über die Bootkonfiguration in der Empirum Management Console.

Hinweis: Die neueste Version des Artikels *WinPE PreBoot Support* ist unter https://marketplace.matrix42.com/product/winpe/ dauerhaft erreichbar. Eine entsprechende Anmeldung wird vorausgesetzt.

1.3 Einschränkungen

Die folgenden Einschränkungen bei der Verwendung von Matrix42 OS Deployment basierend auf WinPE bestehen:

- Die aktuelle Funktionalität von Matrix42 OS Deployment basierend auf WinPE unterstützt ausschließlich 64-Bit Windows Betriebssysteme.
- Das aktuelle Matrix42 PreOS-Paket SimpleOsDeployment unterstützt ausschließlich Windows 10.
- Das WindowsInstallation-Paket unterstützt die Windows 7-Installation nur unter Matrix42 Client Management v19.0.0 oder höher.
- Computer Self Provisioning wird erst ab Matrix42 Client Management v19.0.1 oder höher unterstützt.
- WinPE basiertes Self-Provisioning funktioniert mit allen (EFI) Clients, aber der lokale Boot funktioniert nur auf Clients, welche bereits mit Empirum v19.0.1 / WinPE Support 1.6.1 aufgesetzt wurden. Die alternative ist es kein Default PXE Image in DBUtil zu konfigurieren, sondern von einem USB-Stick zu booten. Ein USB-Stick kann in der Empirum Konsole unter Boot Konfigurationen erstellt werden. Beim Ausführen einer Self Provisioning-Boot-Konfiguration werden im aktuellen Stand zur Konfiguration hinzugefügte Treiber nicht installiert.
- WinPE basiertes Self-Provisioning ist im aktuellen Zustand nur für Installation-Straßen gedacht und sollte nicht direkt in produktiven Umgebungen eingesetzt werden.

2 Schritt für Schritt Anleitung

Dieses Kapitel enthält eine “Schritt für Schritt” Anleitung für die Verwendung des Matrix42 OS Deployment - basierend auf WinPE.

Generell basiert diese neue Variante des Deployments zum einen auf einem WinPE basiertem PXE-Boot-Image, das beim Booten das Matrix42 Universal Agent Framework startet. Das Matrix42 Universal Agent Framework sorgt dann dafür, dass sich der bootende Computer mit dem zugewiesenen Empirum Server verbindet und die zugewiesenen Matrix42 PreOS-Pakete ausführt. Für die Verbindung zum Empirum Server werden die Angaben in dem ausgewählten Agent-Template herangezogen.

Das WindowsInstallation Paket ist eines dieser Matrix42 PreOS-Pakete und enthält die Funktionalität, um eine Windows Installation durchzuführen. Es kommt mit zusätzlichen Variablendefinitionen, die das Installationsverhalten steuern. Das WindowsInstallation Paket ab Version 4 ist auf das DiskPartitioning Paket angewiesen, welches vor dem WindowsInstallation Paket ausgeführt werden muss. Außerdem müssen die Pakete PxeOffAndReboot, DomainJoin und EmpirumAgentSetup nach dem WindowsInstallation-Paket ausgeführt werden.

Die Matrix42 PreOS-Pakete können über die Depot-Konfiguration der Matrix42 Management Konsole importiert werden und stehen dann als spezielles Softwarepaket in Empirum zur Verfügung. Damit können Sie wie gewohnt die Matrix42 PreOS-Pakete über die Administration einem Computer zuweisen, der diese dann während eines WinPE-Bootvorgangs ausführt.

In den folgenden Kapiteln werden nun die Schritte beschrieben, um das WinPE basierte Deployment durchzuführen:

- Import von Matrix42 PreOS-Paketen
- Insbesondere die Pakete *DiskPartitioning, (DriverIntegration), WindowsInstallation, PxeOffAndReboot, DomainJoin and EmpirumAgentSetup*
- Erstellen einer WinPE basierten Bootkonfiguration
- Konfiguration der benötigten Variablen; Zuweisung der erstellten Konfigurationen und Aktivierung
- Besonderheiten bei der Verwendung von Empirum Depot Servern

2.1 Import von Matrix42 PreOS-Paketen

Ein Matrix42 PreOS-Paket kann wie ein Softwarepaket über die *Matrix42 Management Console* importiert werden und steht damit in Empirum zur Verfügung.

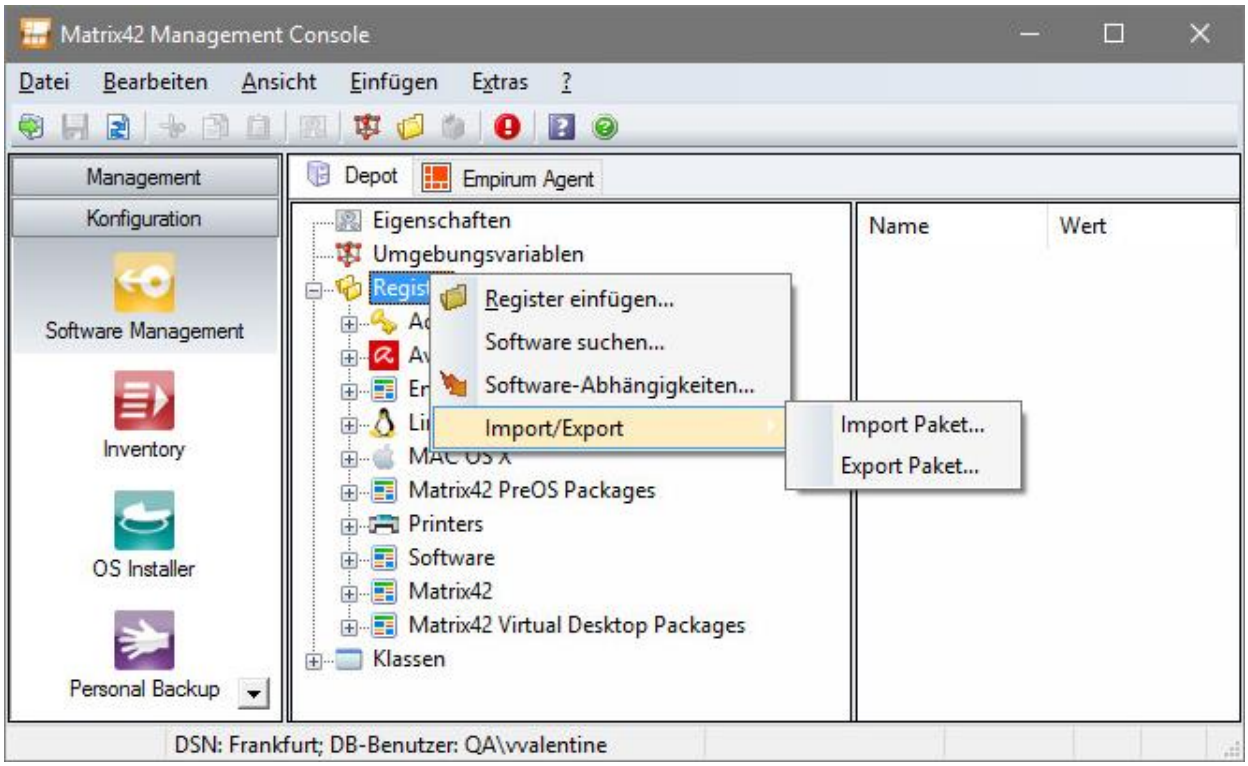
Im Folgenden wird beschrieben, wie Matrix42 PreOS-Pakete importiert werden und wie sie im Anschluss freigegeben werden.

2.1.1 Import der notwendigen PreOS-Pakete

Der Import eines Matrix42 PreOS-Paketes wird über die *Matrix42 Management Console* durchgeführt. Folgen Sie der Anleitung, um die *Matrix42 PreOS Pakete* zu importieren.

1 Starten Sie die *Matrix42 Management Console (EMC)*, wechseln Sie in der Navigationsleiste zu *Konfiguration > Software Management > Register Depot*.

2 Wählen Sie in der Baumansicht den Eintrag *Register* aus und starten Sie über das Kontextmenü *Import/Export > Import Paket* den Import Wizard.



Der Dialog für den Import von Softwarepaketen öffnet sich.

3 Klicken Sie auf *Weiter*.

4 Geben Sie nun das *Verzeichnis* an, in dem sich die Matrix42 PreOS Pakete befinden.

```
\\%EmpirumServer%\Configurator$\PackageStore\PreOsPackages
```

Achtung: Wählen Sie die Option *Pakete nach erfolgreichem Import aus Verzeichnis löschen* nicht aus, damit die Matrix42 PreOS-Pakete im Quellverzeichnis bestehen bleiben.

5 Klicken Sie auf *Weiter*.

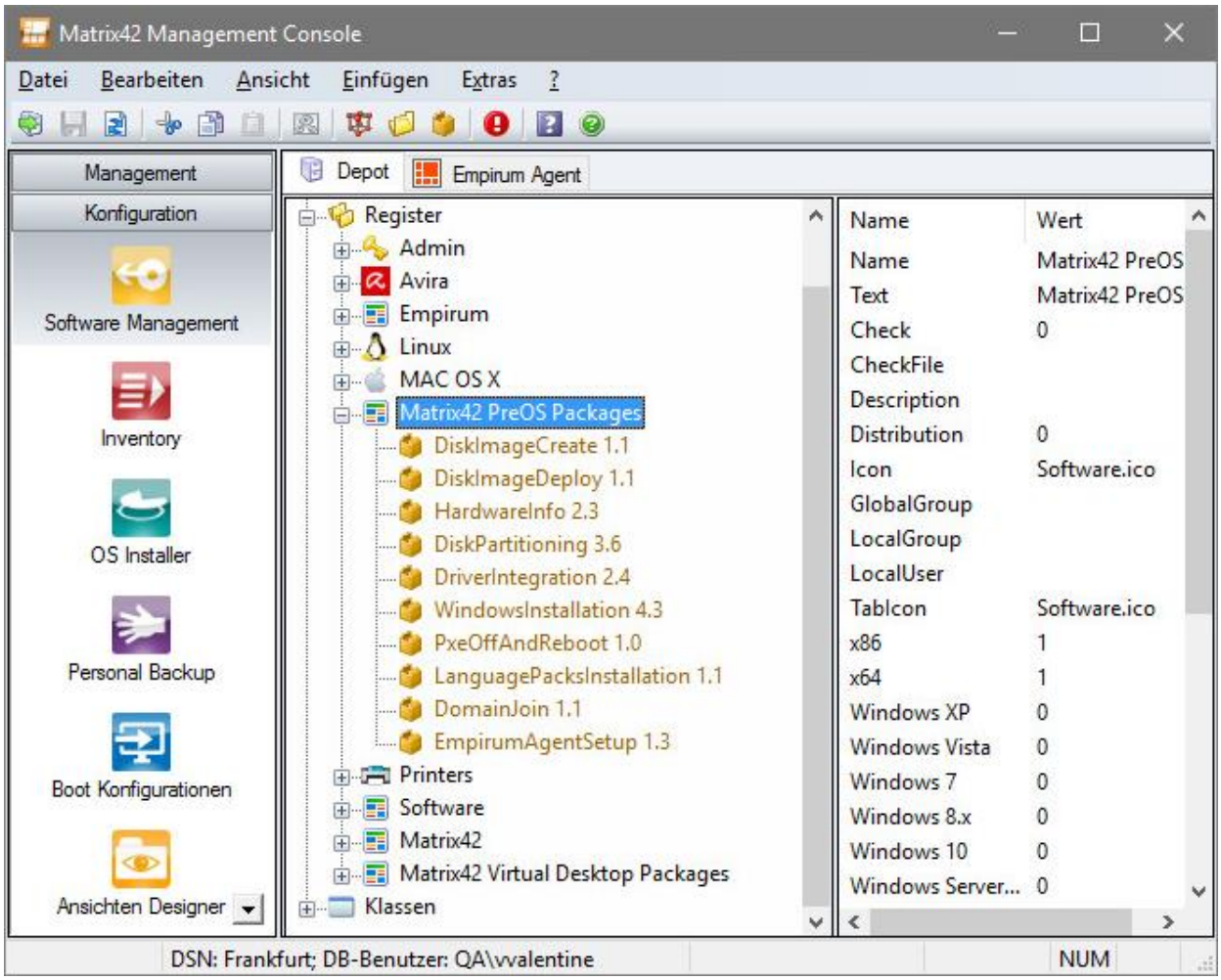
6 Der Import Wizard zeigt alle erkannten OS Softwarepakete an. Wählen Sie in der Liste die neuesten Pakete für *DiskPartitioning, DomainJoin, DriverIntegration, EmpirumAgentSetup, LanguagePacksInstallation, PxeOffAndReboot* und *WindowsInstallation* aus.

7 Klicken Sie auf *Weiter*.

Achtung: Ist das Matrix42 PreOS-Paket bereits vorhanden, fragt der Import Wizard nach, ob das Paket trotzdem importiert werden soll. Wird der Import trotzdem durchgeführt, wird ein Fehler auftreten.

8 Der Import Wizard zeigt eine Zusammenfassung, welche Pakete im Anschluss importiert werden. Klicken Sie auf *Fertig stellen,* um den Import abzuschließen.

9 Nach dem erfolgreichen Import schließt sich der Import Wizard und die Pakete *DiskPartitioning, DomainJoin, DriverIntegration, EmpirumAgentSetup, LanguagePacksInstallation, PxeOffAndReboot* und *WindowsInstallation* werden im Register *Matrix42 PreOS Packages* angezeigt.



Hinweise: Matrix42 PreOS-Pakete werden immer in das Register importiert, das in der Konfigurationsdatei (EmpirumPackageData.xml) des Paketes angegeben ist. Dieses Verhalten unterscheidet sich von dem Verhalten beim Import von Softwarepaketen. Startet man dort den Import über ein spezielles Register (beispielsweise Software), dann werden die zu importierenden Softwarepakete auch in diesem Register abgelegt.

Sind noch keine Matrix42 PreOS-Pakete vorhanden, wird die Reihenfolge der Pakete anhand der Angabe in der Konfigurationsdatei (EmpirumPackageData.xml) gesetzt.

Ab Empirum v20.0: Ist bereits eine Version eines Matrix42 PreOS-Paketes vorhanden, dann wird beim Import zunächst dessen Position im Register bestimmt und auf das neu zu importierende Paket übertragen. Die neue Version des Pakets wird dann direkt unter der bereits vorhandenen Version angezeigt.

2.1.2 Freigabe der Matrix42 PreOS-Pakete (bis Version 1.8.0)

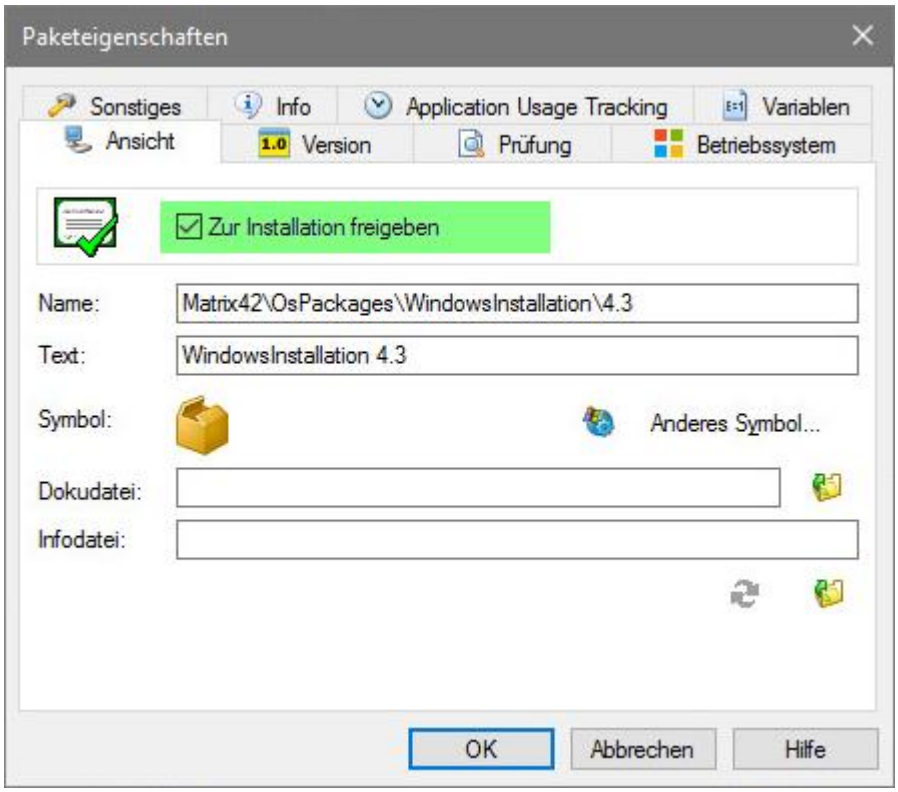
Bis einschließlich der WinPE Support-Version 1.8.0 müssen nach dem Import der DiskPartitioning-, *DriverIntegration-,* WindowsInstallation-, PxeOffAndReboot-, DomainJoin- und EmpirumAgentSetup-Pakete diese noch über die Depot-Konfiguration zur Installation freigegeben werden. Hierzu gehen Sie wie folgt vor:

1 Starten Sie die *Matrix42 Management Console (EMC)*, wechseln Sie in der Navigationsleiste zu *Konfiguration > Software Management > Register Depot*.

2 Öffnen Sie das Register *Matrix42 PreOS Packages* in der Baumstruktur und wählen Sie ein *Paket* aus.

> **Hinweis:** Wird ein neu importiertes Paket noch nicht angezeigt, muss die Depot-Konfiguration aktualisiert werden. Hierzu wählen Sie in der Menüleiste *Ansicht > Aktualisieren*. Damit wird die Depot-Konfiguration erneut aus der Datenbank gelesen.

3 Öffnen Sie über das *Kontextmenü* des neuen Pakets, dessen *Eigenschaften*.



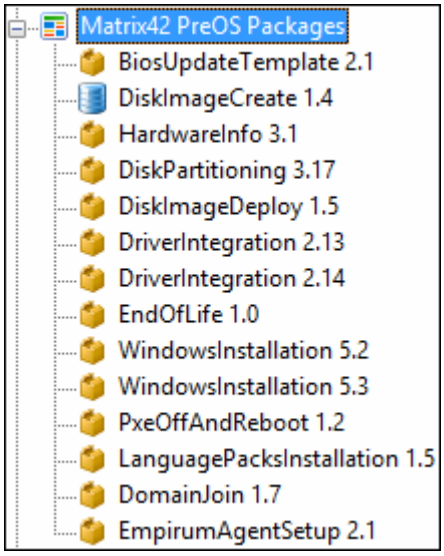
4 Unter dem Reiter *Ansicht* muss die Eigenschaft *Zur Installation freigeben* angehakt werden.

5 Betätigen Sie die Änderung mit *OK*.

6 Im Anschluss muss die Änderung in der Depot-Konfiguration noch über das Menü *Datei > Speichern* abgespeichert werden.

> **Hinweise:** Generell ist es möglich, mehrere Matrix42 PreOS-Pakete beim Booten des WinPE basierenden PXE Images ausführen zu lassen. Die ***Ausführungsreihenfolge*** dieser Pakete kann, wie auch bei anderen Softwarepaketen, global über die Reihenfolge der Pakete im Depot gesteuert werden.
>
> Ab Version **1.8.1** sind die Pakete direkt nach dem Import automatisch freigegeben.

Folgende Reihenfolge ist für die Windows-Installation Voraussetzung:



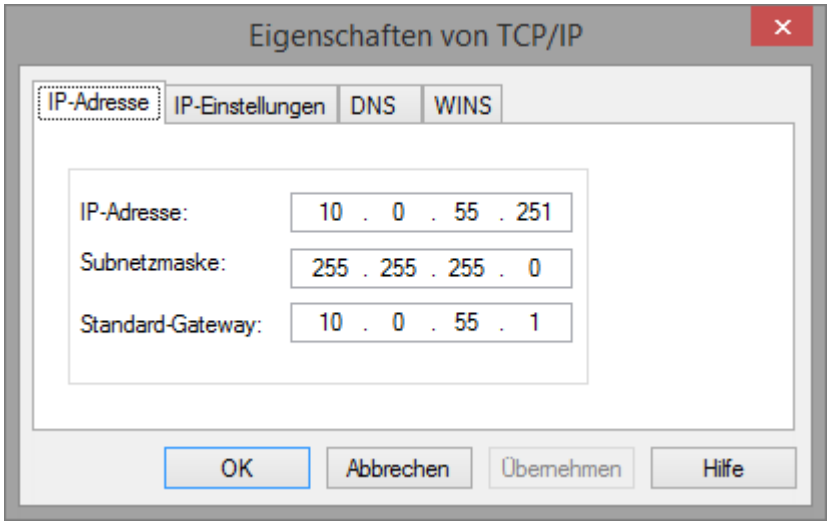
Im Kapitel *2.3.1 Paketreihenfolge in der Depotkonfiguration* auf *Seite 20* wird das detailliert erläutert.

2.2 Erstellung einer WinPE basierenden Bootkonfiguration

Das folgende Kapitel beschreibt die Verwendung der neuen WinPE basierten Empirum Preinstallation Environment Konfiguration.

> **Hinweis:** Ab Empirum *v20.0.1* in Verbindung mit den Matrix42 PreOS-Paketen *WindowsInstallation 5.3* und *DriverIntegration 2.14* oder höher, kann auch eine manuelle IP-Konfiguration für die Betriebssysteminstallation in der WinPE / Windows Phase verwendet werden.

Dazu muss in den *Client Eigenschaften* (EMC > Administration > Eigenschaften Computer > IP-Adresse > Statisch) die IP-Konfiguration im Register *IP-Adresse* und *DNS* vorgenommen werden.



Die hier eingetragenen Werte werden dann in die *<Computername>.ini* in der Sektion `[MS_TCPIP]` eingetragen.

```
[MS_TCPIP]
DHCP=NO
IPADRESS=10.0.55.251
SNMASK=255.255.255.0
GATEWAY=10.0.55.1
DNSName=WinPE-C10-U1
DNSDomain=QAlab.Matrix42.de
DNSSRV1=10.0.2.2
DNSSRV2=10.0.2.4
```

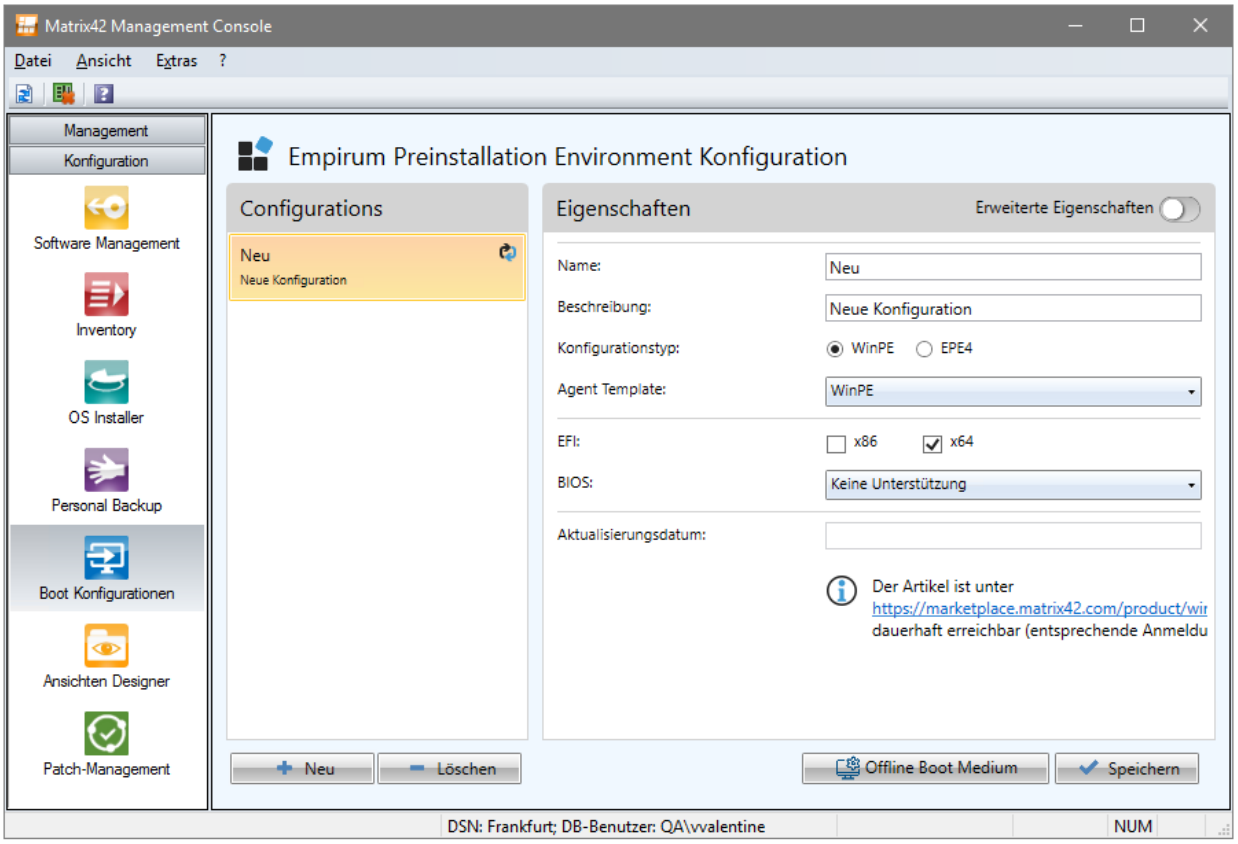
2.2.1 Erstellen einer Boot Konfiguration

In der Boot-Konfiguration kann ein WinPE basiertes PXE-Boot-Image auf Basis des auf dem Empirum Master Server installiertem Windows ADK erstellt werden. Hierzu sind folgende Schritt notwendig:

1 Wechseln Sie in der Navigationsleiste zu *Konfiguration* und wählen Sie *Boot Konfigurationen* aus.

> **Hinweis:** Um eine Boot-Konfiguration neu erstellen oder ändern zu können benötigt der angemeldete Benutzer die Rolle *EMP_I_DISK_CONFIG,* die in Matrix42 DBUtil in der Benutzerverwaltung zugewiesen werden kann. Besitzt der angemeldete Benutzer die Rolle nicht wird der Inhalt der Boot Konfiguration ausgegraut.

2 Klicken Sie auf die Schaltfläche *Neu*, um eine *neue Konfiguration* zu erstellen.



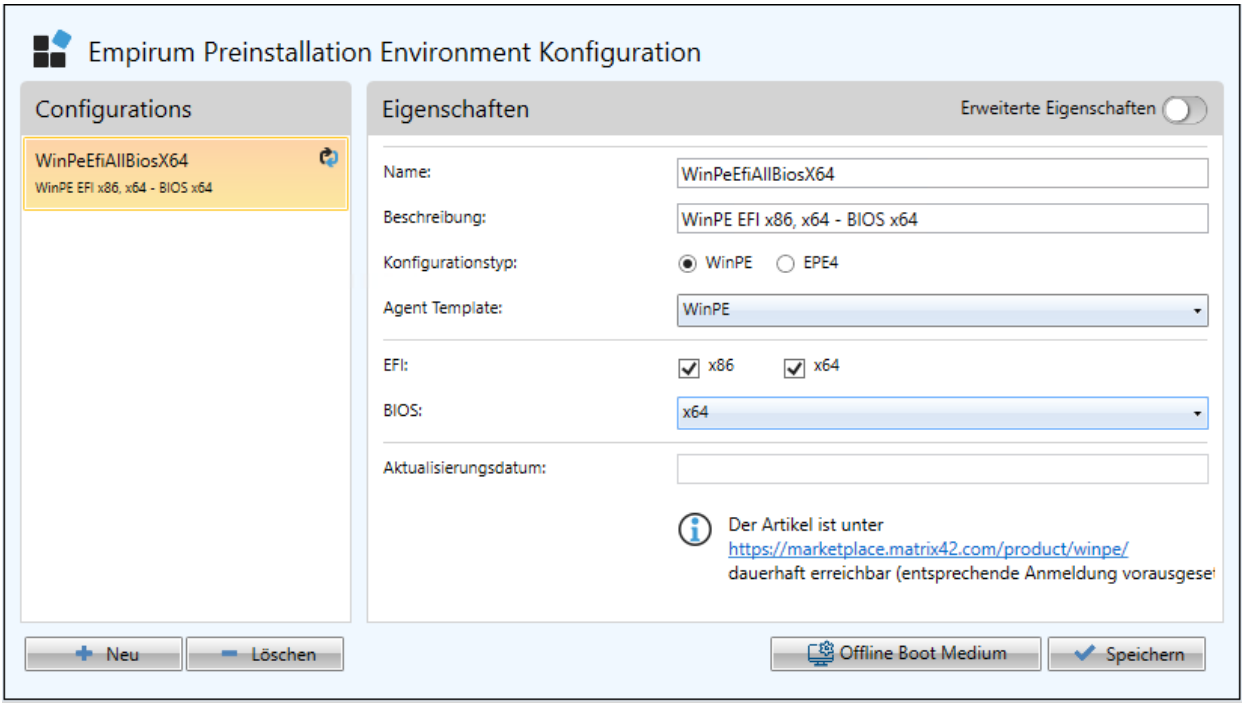
Symboldefinition:

- Eine oder mehrere kritische Angaben fehlen oder sind fehlerhaft.
- Angaben bzw. Daten wurden geändert, aber noch nicht gespeichert.
- Der Auftrag ist in der Warteschlange.
- Die Boot Konfiguration wird gerade im Hintergrund erstellt.
- Die Boot Konfiguration wurde erfolgreich erstellt.
- Ein Fehler ist bei der Erstellung der Boot Konfiguration aufgetreten.
- Die Konfiguration wurde im Hintergrund geändert und kann nicht mehr gespeichert werden. Eine Aktualisierung ist notwendig, um die Änderungen zu laden.
- Die Boot-Konfiguration wurde im Hintergrund gelöscht und kann nicht mehr gespeichert werden. Eine Aktualisierung ist erforderlich, um die Konfiguration aus der Liste zu entfernen.

3 Passen Sie die Felder *Name* und *Beschreibung* gemäß Ihren Anforderungen an.

> **Hinweise:**
> Für den Namen sind nur alphanumerische Zeichen (a-z, A-Z und 0-9) zugelassen. Namen müssen eindeutig sein.
> Die Verwendung von reservierten Namen wird unterbunden. Darunter fallen auch Namen, die bereits in der Bootdiskettenkonfiguration (EPE 3) verwendet werden.
> Das Symbol zeigt an, dass die Eingabe nicht zulässig ist.

4 Wählen Sie als Konfigurationstyp *WinPE* aus, um ein WinPE basiertes PXE-Boot-Image zu erstellen - falls nicht bereits ausgewählt.



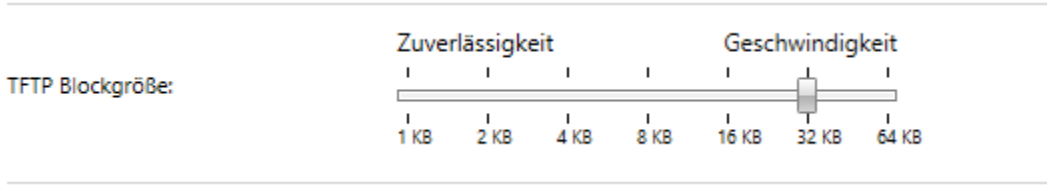
> **Hinweis:** Falls Sie als Konfigurationstyp WinPE auswählen, werden die Auswahl der Empirum PE Quelle und der dynamischen Servererkennung ausgeblendet und stehen in der Konfiguration nicht zur Verfügung. Diese Eigenschaften stehen nur unter der EPE4 Konfiguration zur Verfügung und sind im Fall von einem WinPE Preinstallation Environment nicht notwendig. Standard-mäßig ist WinPE bei neuen Konfigurationen ausgewählt.

5 Wählen Sie aus dem Drop-Down Feld Agent Template das gewünschte *Agent-Template*.

> **Hinweise:**
> Die Angaben im ausgewählten Agent Template legen fest, wie und mit welchem Server versucht wird, beim OS Deployment sich zu verbinden.
> Falls in der Auswahl der Agent Templates kein Eintrag angezeigt wird, erstellen Sie zunächst ein Empirum Agent Template über Konfiguration > Software Management > Empirum Agent. Neben Benutzername, Passwort und Servername werden auch die Einstellungen für die DHCP-Optionen in das PXE Boot Image übernommen, sofern es im Agent Template konfiguriert wurde. Im Anschluss kann über die Schaltfläche Aktualisieren die Auswahl aktualisiert werden. Wurden mehrere Empirum Agent Templates erstellt, wird - Alphabetisch sortiert - immer das erste direkt angezeigt.
> Die Übersicht Links wird in Echtzeit aktualisiert. Änderungen werden aber erst nach dem Bestätigen mit Speichern dauerhaft übernommen.

6 Um auszuwählen, welche Plattformen unterstützt werden sollen, müssen Sie entweder *EFI x86 oder EFI x64* anhaken oder *eine der Plattformen aus dem BIOS-Drop-Down-Feld* auswählen. Mehrere Plattformen können gleichzeitig ausgewählt werden. Für *BIOS* kann allerdings *nur eine der Plattformen gleichzeitig* ausgewählt werden - entweder 32 Bit oder 64 Bit. Um eine Konfiguration zu erstellen, muss *mindestens eine der Plattformen* ausgewählt werden. Als Standard ist EFI x64 (64 Bit) ausgewählt.

7 Über *Erweiterte Eigenschaften* können Sie die Einstellung für *TFTP Blockgröße*, *Self Provisioning* und *Treiberverzeichnisse* vornehmen. Klicken Sie auf die Schaltfache Erweiterte Eigenschaften, um die Felder einzublenden.



8 Mit Hilfe der *TFTP Blockgrößeneinstellung* kann die Übertragung des WinPE Bootimages angepasst werden, um diese entweder *stabiler* oder *schneller* zu gestalten.

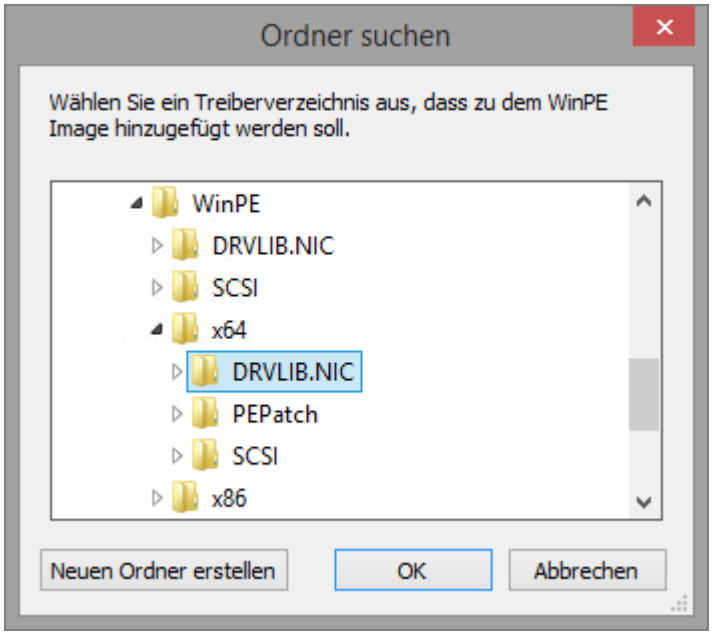
> **Hinweis:** Ein höherer TFTP Blockgrößenwert führt üblicherweise zu einer schnelleren Übertragung des Bootimages. Allerdings kann eine größere Blockgröße auch zu Übertragungsabbrüchen führen. Ein optimaler Wert hängt von der vorhanden Netzwerkinfrastruktur und deren Auslastung ab. Bei einer neu erstellten Bootkonfiguration liegt der Standardwert der TFTP Blockgröße bei 4 KB.

9 *Self Provisioning aktivieren* wird ausführlich in Kapitel 7 beschrieben.

10 Möchten Sie *zusätzliche Treiber* im WinPE verwenden, dann gehen Sie wie folgt vor:

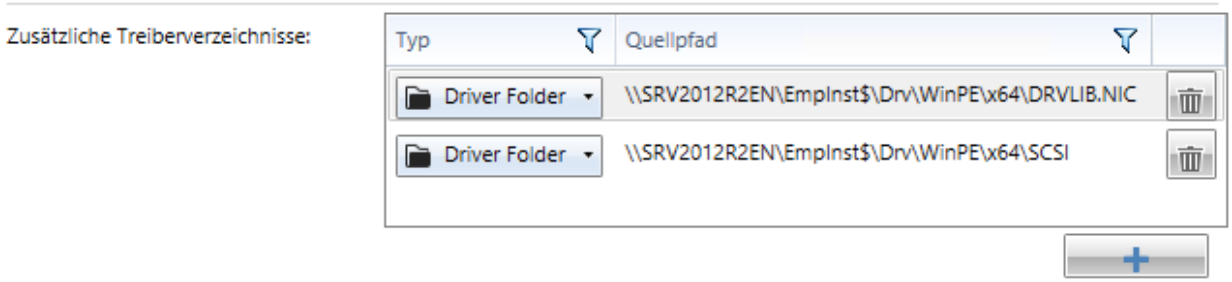
> **Hinweise:**
> Eine detaillierte Schritt für Schritt Anleitung zur Handhabung von Treibern ist im Dokument "Matrix42 Treiberintegration in WinPE - HowTo - DE.pdf" beschrieben.
> Bei der Erstellung des WinPE PXE Images werden die Inhalte der zusätzlichen Treiberverzeichnisse in das Image kopiert und stehen dann zur Laufzeit zur Verfügung.
> Es können sowohl 32-bit als auch 64-bit-Treiber hinzugefügt werden.

11 Klicken Sie auf die *Schaltfläche* unterhalb der Liste der zusätzlichen Treiberverzeichnisse, um ein weiteres *Verzeichnis* der Liste hinzuzufügen.



Es öffnet sich das Ordner suchen Fenster mit dessen Hilfe ein Verzeichnis ausgewählt werden kann.

- Bestätigen Sie die Auswahl mit *OK*.
- Das ausgewählte *Verzeichnis* wird in die *Liste der zusätzlichen Treiberverzeichnisse* aufgenommen.



- Falls Sie ein Treiberverzeichnis wieder *entfernen* möchten, dann klicken Sie auf die Schaltfläche rechts neben dem Eintrag des Treiberverzeichnisses.

12 Wurden alle Einstellungen zu Ihrer Zufriedenheit vorgenommen, bestätigen Sie mit *Speichern* und anschließend beantworten Sie die Sicherheitsfrage mit *Ja*.

Wie bei der EPE 4 Konfiguration wird jetzt nicht nur die Konfiguration in der Datenbank gespeichert, sondern auch direkt ein PXE-Boot-Image erstellt.

Nach erfolgreichem Erstellen des PXE-Boot-Images wird dieses mit dem angegebenen Namen in Management > Administration unter PXE-Boot-Images angezeigt und kann wie gewohnt verwendet werden.

2.2.2 Generierungsprozess des PXE Images

Nach dem Abspeichern der Empirum Preinstallation Environment Konfiguration wird die automatische Erstellung des PXE Images durch eine neue Backend Task Queue Erweiterung übernommen. Im Anschluss kann dann wie gewohnt in der Administration das PXE-Boot-Image zugewiesen werden.

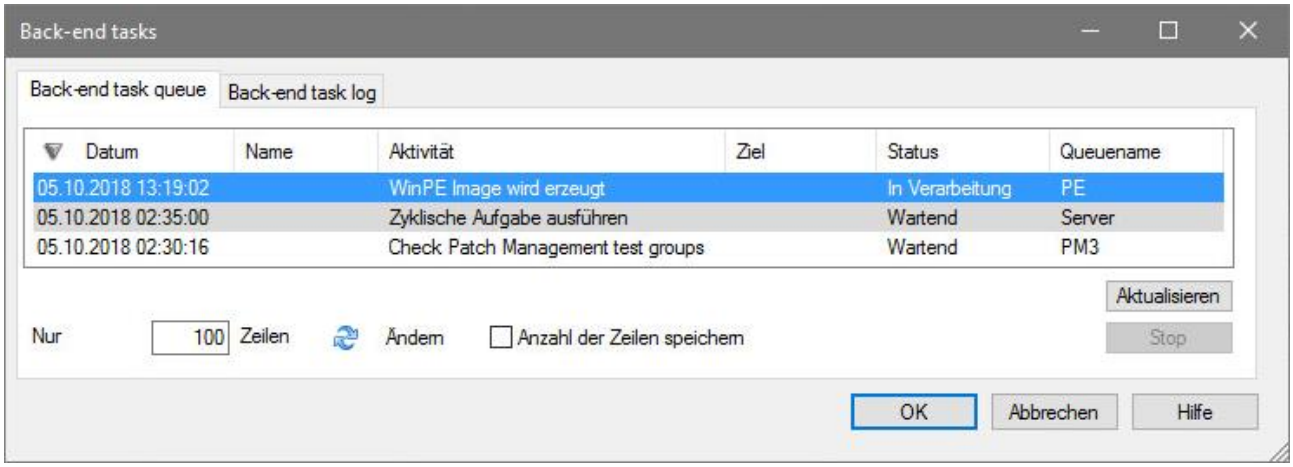
2.2.2.1 Back-end Task Queue

Die aktuellen Aufträge in der Backend Task Queue können über den folgenden Dialog in der Matrix42 Management Console (EMC) geprüft werden:

- *Matrix42 Management Console > Management > Administration > Menü Datei > Info > Back-end tasks*

Die Queue Einträge mit dem Namen PE (= Preinstallation Environment) sind die Aufträge, die zur Erstellung des PXE Images von Interesse sind.

Die Liste zeigt, welche Aufträge gerade von der Queue abgearbeitet werden.



Hinweis: Der BTQH-Dienst muss unter einem Benutzer laufen, der über administrative Rechte verfügt.

2.2.2.2 Back-end Task Log

Im Register Back-end Task log kann der Status der bereits abgearbeiteten Aufträge betrachtet werden.

In der Spalte Ergebnis ist der Erfolg des Auftrages zu erkennen.

Bei Misserfolg ist in der Spalte Hinweis eine detaillierte Information hinterlegt, die Auskunft über den Fehler gibt.

Hinweis: Detaillierte Informationen erhält man ebenfalls über die Log-Datei der Backend Task Queue - diese liegt unter: `C:\ProgramData\Matrix42\Logs\BackendTaskQueueHost64\BackendTaskQueueHost64.log`

2.3 Zuweisung in der Administration

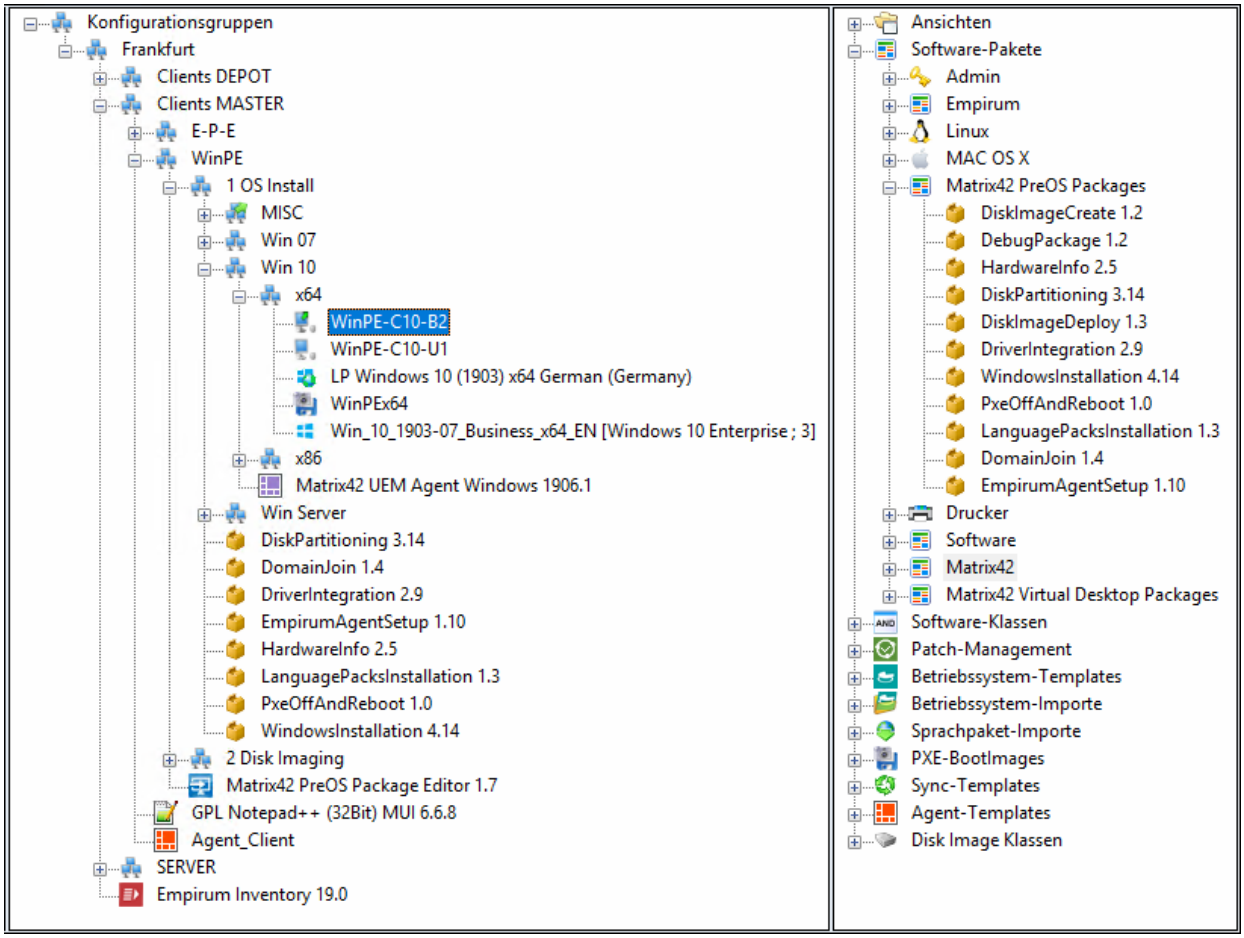
Damit ein Computer nun das OS Deployment über WinPE ausführt, sind jetzt noch die Einstellungen in der Administration vorzunehmen.

- Dem Computer muss das erstellte PXE-Boot-Image ("WinPEBoot") zugewiesen werden.
- Dem Computer muss die gewünschte Edition des Betriebssystem-Imports zugewiesen werden.
- Dem Computer können die gewünschten Sprachpaketimporte zugewiesen werden.
- Dem zu installierenden Computer müssen die Matrix42 PreOS Pakete DiskPartitioning, WindowsInstallation, PxeOffAndReboot, DomainJoin und EmpirumAgentSetup zugewiesen werden. Bei Verwendung von Sprachpaketimporten muss auch das LanguagePacksInstallation Paket zugewiesen sein. Wenn zusätzlich noch Treiber in das Betriebssystem integriert werden sollen, muss auch das DriverIntegration-Paket zugewiesen werden.
- Die von DiskPartitioning, WindowsInstallation, PxeOffAndReboot, DomainJoin und EmpirumAgentSetup benötigten Computervariablen sind für das Deployment zu setzen.
- Der Computer muss aktiviert werden und kann dann das Deployment ausführen.

Hinweis: Generell ist es auch möglich, mehrere Matrix42 PreOS-Pakete zuzuweisen, die dann beim Booten des WinPE basierenden PXE Images nacheinander ausgeführt werden. Die **Ausführungsreihenfolge** dieser Pakete kann, wie auch bei anderen Softwarepaketen, global über die Reihenfolge der Pakete im Depot gesteuert werden.

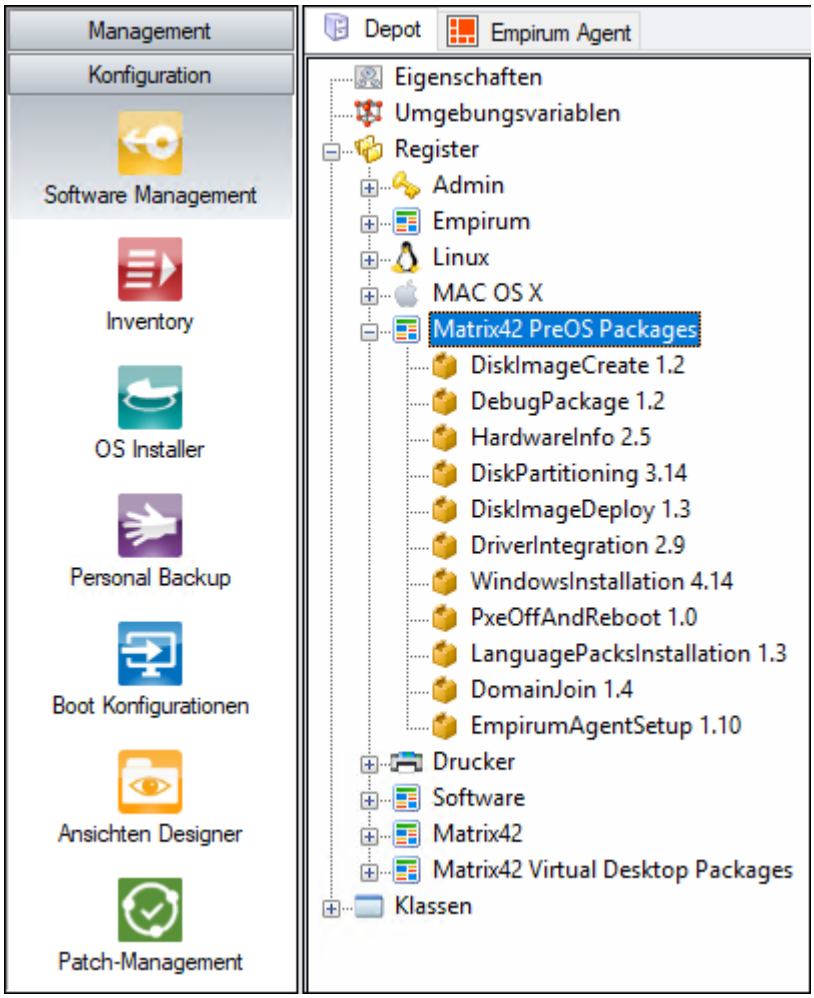
Die einzelnen Schritte werden im Folgenden nun genauer ausgeführt:

1 Weisen Sie die Pakete *DiskPartitioning*, *WindowsInstallation*, *PxeOffAndReboot*, *DomainJoin*, *EmpirumAgentSetup* und das *WinPE basierende PXE-Boot-Image* (in diesem Beispiel WinPE) dem Computer zu.



2.3.1 Paketreihenfolge in der Depotkonfiguration

Soll diese Variante des Windows-Deployments verwendet werden, muss sichergestellt sein, dass die Matrix42 PreOS Pakete auch in der **richtigen Reihenfolge** ausgeführt werden. Hierzu kann die Reihenfolge in der Depotkonfiguration angepasst werden.



Die Reihenfolge muss somit wie folgt aussehen:

- DiskPartitioning
- DriverIntegration (siehe Kapitel zur Treiberintegration)
- WindowsInstallation
- PxeOffAndReboot
- LanguagePacksInstallation
- DomainJoin
- EmpirumAgentSetup

Die Änderungen der Reihenfolge in der Depotkonfiguration müssen abgespeichert werden.

2.3.2 Globale Variablen

2.3.2.1 TimeZone

Mit der *TimeZone* Variable unter *OS_RegionalSettings* stellt man die Zeitzone sowohl unter *WinPE* als auch im zu installierenden *Betriebssystem* ein (z.B. (UTC+1:00) Amsterdam, Berlin, Bern, Rome, Stockholm, Viena).

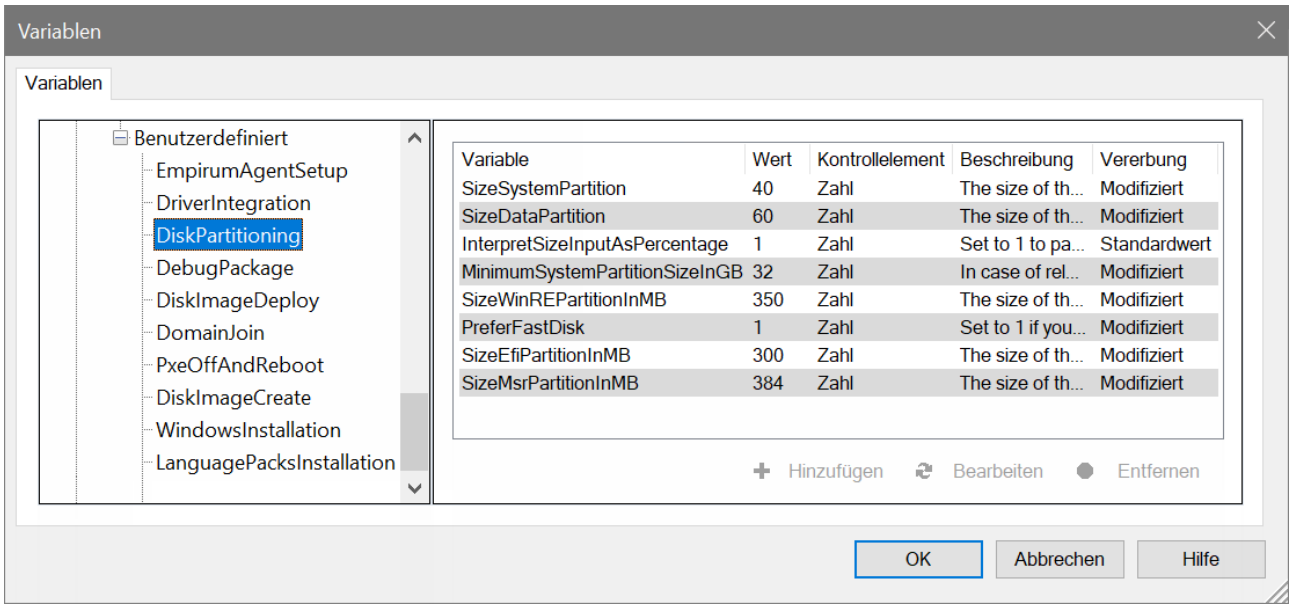
WindowsInstallation und *SimpleOsDeployment* Pakete verwenden diese Variable, um die Zeitzone unter Windows einzustellen.

Der PE Agent verwendet diese Variable, um die Zeitzone unter WinPE zur Laufzeit einzustellen.

2.3.3 DiskPartitioning Paket

Das erste Paket in der Reihe ist das DiskPartitioning Paket, das die Partitionierung der Festplatte durchführt. Die Konfiguration der Partitionierung wird über die Paketvariablen gesteuert.

> **Hinweis:** Das Ausführen des DiskPartitioning Paketes schaltet den Paketstatus aller vorher installierten Pakete auf GELB, auch wenn sie kurz vorher noch auf GRÜN waren. Da hier eine komplette Löschung der Festplatte erfolgt, kann der zuvor ermittelte Status jetzt nicht mehr stimmen.



Für die Partitionierung stehen zwei Varianten (absolute oder prozentuale Partitionierung) zur Verfügung, die über die Variable *DiskPartitioning.InterpretSizeInputAsPercentage* gesteuert wird. Ist der Wert auf *1* gesetzt, dann werden die Angaben in *DiskPartitioning.SizeDataPartition*, *DiskPartitioning.SizeSystemPartition* als prozentuale Angaben interpretiert.

2.3.3.1 SizeSystemPartition (Größe der Systempartition)

Der Wert SizeSystemPartition gibt bei einer absoluten Partitionierungsvariante die Partitionsgröße der Systempartition in Gigabyte an.

Beispiel:

```
DiskPartitioning.SizeSystemPartition = 150
```

Hier wird dann vom DiskPartitioning Paket versucht, eine 150 GB große Systempartition zu erstellen.

2.3.3.2 SizeDataPartition (Größe der Datenpartition)

Der Wert SizeDataPartition gibt bei einer absoluten Partitionierungsvariante die Partitionsgröße der Datenpartition in Gigabyte an.

Beispiel:

```
DiskPartitioning.SizeDataPartition = 400
```

Hier wird dann vom DiskPartitioning Paket versucht, eine zweite 400 GB Datenpartition hinter der Systempartition zu erstellen.

2.3.3.3 InterpretSizeInputAsPercentage (Prozentuale Partitionierung)

Sollen für die Partitionierung relative Angaben verwendet werden, dann ist der folgende Variablenwert zu setzen.

```
DiskPartitioning.InterpretSizeInputAsPercentage = 1
```

Die Werte in SizeSystemPartition und SizeDataPartition werden in diesem Fall als prozentuale Werte interpretiert.

Beispiel:

```
DiskPartitioning.SizeSystemPartition = 100
DiskPartitioning.SizeDataPartition = 0
```

Hier wird vom DiskPartitioning Paket nur eine Systempartition erstellt, die den gesamten freien Platz der Festplatte einnimmt.

```
DiskPartitioning.SizeSystemPartition = 50
DiskPartitioning.SizeDataPartition = 50
```

Hier wird vom DiskPartitioning Paket eine Systempartition und Datenpartition erstellt, die jeweils die gleiche Größe auf der Festplatte einnehmen.

Sollen für die Partitionierung feste Gigabyte Werte verwendet werden, dann ist der folgende Variablenwert zu setzen.

```
DiskPartitioning.InterpretSizeInputAsPercentage = 0
```

> **Hinweis:** Das DiskPartitioning Paket wird mit einem Fehler abbrechen, falls die Festplatte zu klein ist, um die angegebenen Partitionsgrößen anzulegen.

2.3.3.4 MinimumSystemPartitionSizeInGB (Minimale System-Partitionsgröße)

Wenn eine relative/prozentuale Partitionierungsmethode (InterpretSizeInputAsPercentage=1) gewählt wurde, kann mit der Variable MinimumSystemPartitionSizeInGB eine minimale System-Partitionsgröße vorgegeben werden, die bei der Partitionierung nicht unterschritten wird.

2.3.3.5 SizeWinREPartitionInMB (Größe der Wiederherstellungspartition)

Soll während der Partitionionierung auch eine Windows Recovery Partition angelegt werden, lässt sich die Größe über die Variable SizeWinREPartitionInMB in MB definieren. Wenn diese Variable leer oder 0 ist, wird keine Windows Recovery Partition angelegt. Die Standardgröße der Wiederherstellungspartition beträgt 350 MB.

2.3.3.6 PreferFastDisk (Bevorzugtes Installations-Ziellaufwerk)

Häufig befinden sich im Client mehrere Festplattenlaufwerke unterschiedlicher Bauweise. Zumeist würde man es bevorzugen das Betriebssystem auf dem schnellten Festplattenlaufwerke zu installieren. Das DiskPartitioning Paket bietet hierzu die Möglichkeit das bevorzugte Installations-Ziellaufwerk über die Variable

```
DiskPartitioning.PreferFastDisk= 1
```

im Vorfeld zu definieren, basierend auf der Speichertechnologie des Laufwerkes.

Wird diese Variable auf 1 gesetzt, sucht das DiskPartitioning Paket nach der ersten (laut BIOS) NVME Disk und wird diese für die Betriebssysteminstallation vorbereiten.

> **Hinweis:** Falls auf dem zu installierenden Gerät mehrere Festplatten und bereits eine Windows-Installation vorhanden sind und als System-Festplatte nun eine andere ausgewählt wurde - etwa durch die Variable "PreferFastDisk", dann kann es bei der Ausführung des Windows-Setup zu Problemen kommen. Man kann hierbei entweder manuell über DISKPART oder über die versteckte Variable "ClearAllDisks" alle Festplatten beim Ausführen des DiskPartitioning Paketes bereinigen. Hierzu muss man unter "Extras > Variablendefinitionen…" in der EMC die Variable "ClearAllDisks" als Zahl zur Variablen DiskPartitioning hinzufügen und diese auf 1 setzen.

2.3.3.7 SizeEfiPartitionInMB (Größe der EFI Partition)

Die Variable *SizeEfiPartitionInMB* definiert in EFI-basierten Clientszenarien die Größe der zu erstellenden EFI Partition in MB.

Standardmäßig wird diese mit 100 MB angelegt. Sie darf für EFI-basierte Bespielungen auch größer, aber nicht kleiner definiert werden.

Bei älteren MBR-basierten Clientszenarien wird diese Variable ignoriert.

2.3.3.8 SizeMsrPartitionInMB (Größe der Microsoft Reserved Partition)

Die Variable *SizeMsrPartitionInMB* definiert in EFI-basierten Clientszenarien die Größe der zu erstellenden "Microsoft Reserved Partition" in MB.

Standardmäßig wird diese mit 128 MB angelegt. Sie darf für EFI-basierte Bespielungen auch größer, aber nicht kleiner definiert werden.

Bei älteren MBR-basierten Clientszenarien wird diese Variable ignoriert.

2.3.4 DriverIntegration Paket

Wenn Treiber in das WinPE integriert wurden, kann man mit hoher Wahrscheinlichkeit davon ausgehen, dass auch Treiber für die Windows Installation integriert werden müssen.

Für mehr Informationen darüber, lesen Sie bitte erst das Kapitel 4 Treiberintegration für die Betriebssysteminstallation ab Seite 39, wenn weitere Treiber integriert werden müssen.

2.3.5 WindowsInstallation Paket

Das WindowsInstallation Paket ist wie folgt zu konfigurieren.

1 Setzen Sie nun die vom WindowsInstallation-Paket benötigten Computervariablen, die unterhalb der WindowsInstallation-Variablen strukturiert sind:

- *LocalUserName* (Anmeldename für das lokale Konto)
- *LocalUserPassword* (Passwort für das lokale Konto)
- *LocalUserDisplayName* (Anzeigename des lokalen Kontos)
- *SetupUILanguage* (Definiert die Sprache, die in Windows Setup und Windows Deployment Services verwendet werden soll. z.B. en-US. Die Language Pack-Installation findet erst nach der Betriebssystem-Installation statt d.h.: es kann hier nur eine Sprache ausgewählt werden, die im Betriebssystem-Import vorhanden ist.)
- *InputLocale* (Gibt die Eingabesprache und Eingabemethode für Eingabegeräte an, z. B. das Tastaturlayout. z.B. de-DE)
- *SystemLocale* (Gibt die Standardsprache an, die für Nicht-Unicode-Programme verwendet werden soll. z.B. en-US)
- *UILanguage* (Gibt die zu installierende Sprache an, die als Standardsystemsprache für die Anzeige von Elementen der Benutzeroberfläche (z. B. Menüs, Dialogfelder und Hilfedateien) verwendet wird. z.B. en-US)
- *UserLocale* (Spezifiziert die benutzerbezogenen Einstellungen für die Formatierung von Datum, Uhrzeit, Währung und Zahlen in einer Windows-Installation. z.B. de-DE)
- *ProductKey* Variable gibt den Product Key an, der in der Windows-Setup-Phase verwendet werden soll. Der Product Key kann ein KMS Client Setup Key sein (siehe https://docs.microsoft.com/de-de/windows-server/get-started/kmsclientkeys). Für MAK-Aktivierungsszenarien ist die Konfiguration (ProductKey=KMS Setup Key) / (ActivationKey=MAK Activation Key siehe unten) notwendig. In VL/KMS-Konfiguration besteht aktuell keine Notwendigkeit die Variablen ProductKey oder ActivationKey zu konfigurieren; hier übernimmt der KMS-Server alle notwendigen Konfigurationen.
- *UnattendXmlFile* Variable kann verwendet werden, um auf eine andere unattend.xml Vorlage zu verweisen, die dann bei der Installation von Windows verwendet wird. Die Pfadangabe ist relativ zur EmpInst$-Freigabe anzugeben.

  Beispiel:

  ```
  WindowsInstallation.UnattendXmlFile = Sys\unattend.xml
  ```

  Entspricht:

  ```
  \\%EmpirumServer%\EmpInst$\Sys\unattend.xml
  ```

  Wird die Variable UnattendXmlFile nicht gesetzt, wird die unattend.xml Vorlage aus dem Paketverzeichnis verwendet.
- *ActivationNow* Variable (0/1) definiert, dass beim Ausführen des später folgenden DomainJoin-Paketes eine Windows-Aktivierung stattfinden soll. Wird ein ActivationKey gesetzt, ist auch diese Variable ActivationNow auf "1" zu setzen. Das Resultat der Aktivierung wird in der Datei `C:\ProgramData\Matrix42\Logs\UAF\WinActivation.log` protokolliert.
- *ActivationKey* Variable gibt den Key an, mit dem Windows aktiviert werden soll. Die Definition des Keys ist nicht notwendig in VL/KMS-Konfiguration, findet aber in MAK-Aktivierungsszenarien Verwendung.

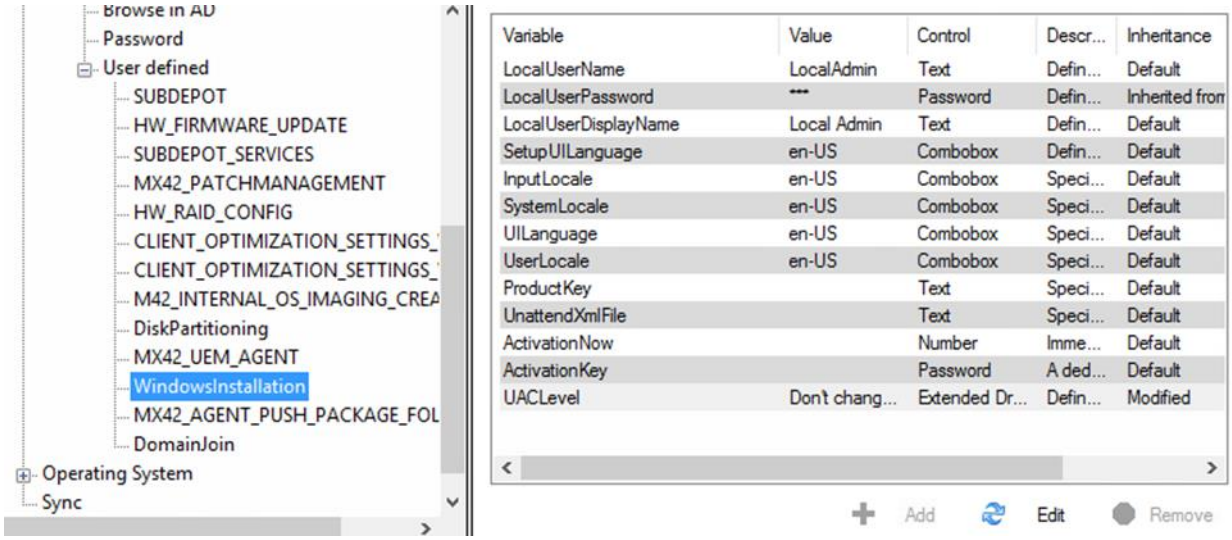
- UACLevel Variable definiert die von Ihnen gewünschte Einstellung der Windows Benutzerkontensteuerung, welche in der Dropdown Box ausgewählt werden kann. Diese Einstellung wird vom Client im Verlauf der Installation übernommen.

  Die Auswahl "Don't change UAC settings" resultiert aus der "default" Einstellung der Benutzerkontensteuerung. Diese Einstellung findet vor allem dann Anwendung, wenn Sie bereits auf anderem Wege die Benutzerkontensteuerung regulieren und ihre Lösung weiterhin nutzen möchten.
- *BuiltinAdministratorActive* (Aktivieren oder deaktivieren des integrierten Administrator-Kontos)
- *BuiltinAdministratorPassword* (Passwort für das integrierte Administrator-Konto)

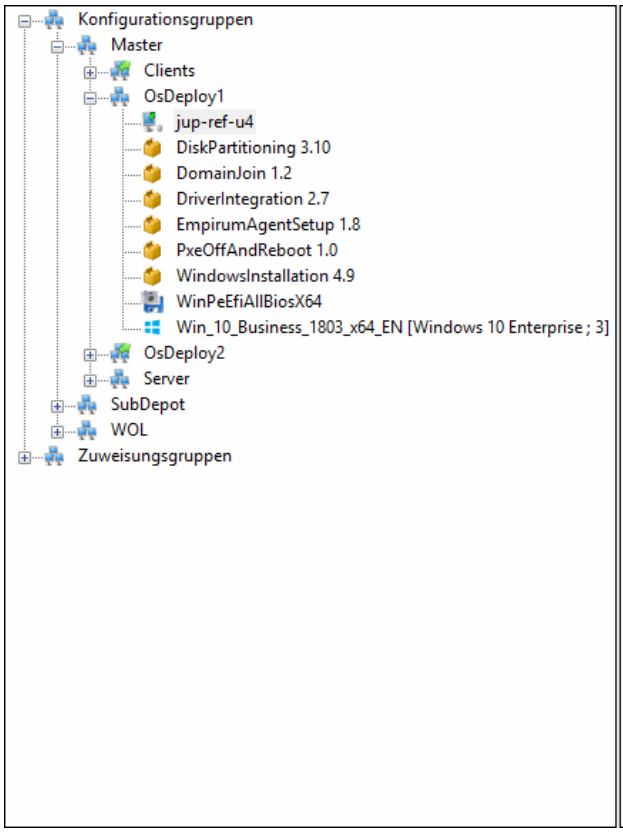
  Standardmäßig - falls kein Passwort konfiguriert wird - wird das Passwort *Matrix42!* gesetzt.
- *ForceDotNetInstallation* Variable (Yes/No). Erzwingt die Installation von *.Net 4.7*, wenn der Wert der Variable auf *Yes* eingestellt wird - z.B. notwendig für die Installation von *Windows 10 2016 LTSB*.

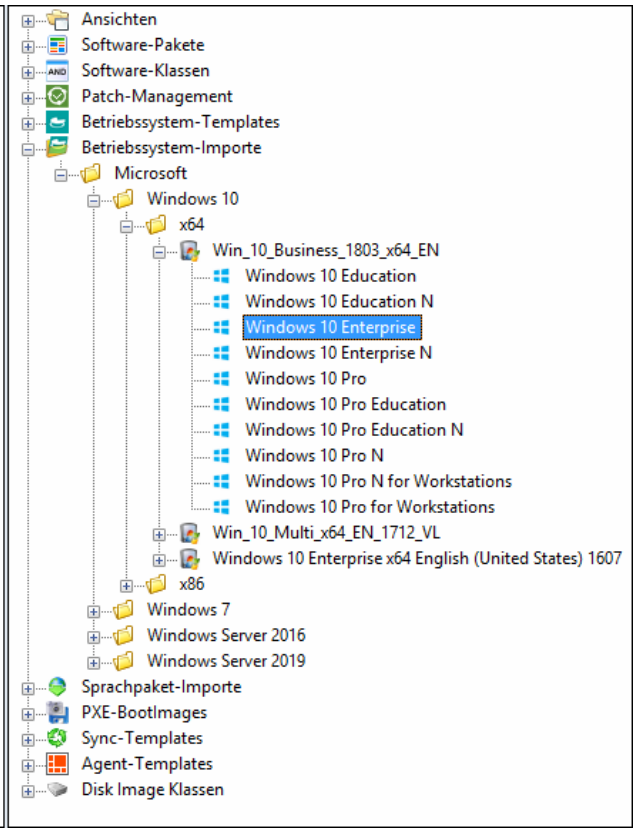
> **Hinweis:** Die Variablen für den Domain Join sind ab der WindowsInstallation Paketversion 4.0 nicht mehr vorhanden, da das Paket den Client nicht mehr für den Domain Join vorbereitet. Diese Funktionalität ist in das separate Paket DomainJoin gewandert. Die entsprechenden Variablen müssen in diesem neuen Paket gesetzt werden, wie im Kapitel 2.3.9 DomainJoin-Paket
>
> auf Seite 30 genauer beschrieben wird.

2 Bestätigen Sie die *Variableneinstellung* erneut mit *OK*.
Hier die für das Beispiel verwendeten Werte:



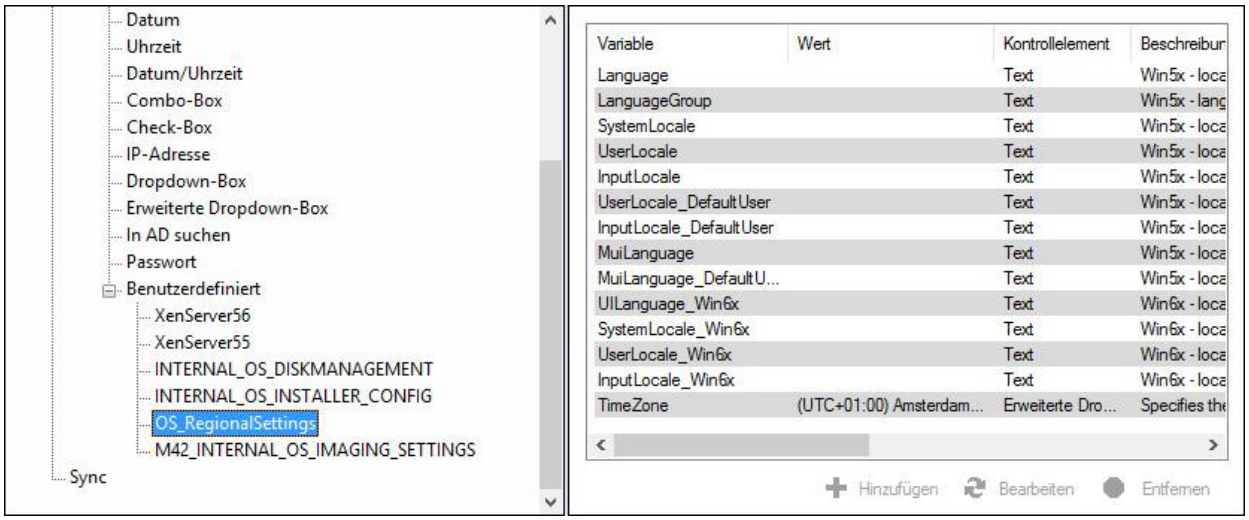
3 Wählen Sie die zu verwendende *Betriebssystemquelle* aus, indem Sie in die Betriebssystem-Importe im rechten Baum wechseln. Dort werden alle importierten Windows 7, Windows 10, Windows Server 2016 und Windows Server 2019 Betriebssysteme in der Ordnerstruktur angezeigt, wie sie auch beim Betriebssystemimport verwendet wird. Bei jedem Betriebssystemimport werden die entsprechenden Editionen des Betriebssystems angezeigt. Weisen Sie die gewünschte *Edition* dem *Computer* im *mittleren Baum* zu.





Hinweise:
Die Konfiguration der zu verwenden Betriebssystemquelle kann auch noch immer über die Betriebssystemvariable OS_PACKAGE_SOURCES durchgeführt werden. Dabei ist zu beachten, dass kein Betriebssystemimport über die Administration dem Computer zugewiesen werden darf, da sonst diese Zuweisung bevorzugt verwendet wird. Bei der Verwendung der Betriebssystemvariable OS_PACKAGE_SOURCES ist ebenfalls zu beachten, dass die zu verwendende Edition des Betriebssystems nicht festgelegt werden kann.
Bei älteren Empirum Versionen (vor 18.0.2) muss die Betriebssystemvariable OS_PACKAGE_SOURCES gesetzt werden.
Ab Empirum v19.0 können neben Windows 10 auch Windows 7, Windows Server 2016 und Windows Server 2019 Betriebssystemimporte zugewiesen werden.

4 Setzen Sie die *TimeZone* Variable unter *OS_RegionalSettings*, um die Zeitzone unter WinPE und im zu installierenden Betriebssystem einzustellen. (z.B. (UTC+1:00) Amsterdam, Berlin, Bern, Rome, Stockholm, Viena).



2.3.6 LocalExperiancePacksInstallation-Paket

1 Weisen Sie das *LocalExperiancePacksInstallation*-Paket der entsprechenden *Konfiguration* zu. Dieses Paket installiert zugeordnete LocalExperiancePacks (LXP/APPX) Sprachpakete, nachdem das WindowsInstallation-Paket das Windows Betriebssystem aufgesetzt hat.

> **Achtung:** Achten Sie darauf, dass im Software-Depot die Reihenfolge der Pakete eingehalten wird: LocalExperiancePacksInstallation muss nach dem WindowsInstallation und vor dem PxeOffAndReboot Paket angeordnet sein.

2 Wählen Sie die zu verwendenden LXP/APPX *Sprachpakete* aus. Hierzu wechseln Sie in die Sprachpaket-Importe im rechten Baum. Dort werden alle importierten Windows 10 Sprachpakete in der Ordnerstruktur angezeigt. Weisen Sie die gewünschten Sprachpakete dem Computer im mittleren Baum zu.

> **Hinweis:** LocalExperiancePacks werden erst ab Windows 10 Version 1809 aufwärts unterstützt. Es werden dabei nur Installationen mit einem englischen (EN-US) Basissystem unterstützt. Alle anderen Basissprachen werden nur experimentell unterstützt. Das Setzen der UILanguage/Desktopsprache kann benutzerspezifisch über den Windows-Dialog "Regionen und Sprache" erfolgen.

2.3.7 PxeOffAndReboot-Paket

Nachdem WindowsInstallation-Paket muss das PxeOffAndReboot-Paket ausgeführt werden. PXE wird für diesen Client deaktiviert und der Client wird im Anschluss neu gebootet. Damit wird beim nächsten Reboot nicht mehr vom Netzwerk gestartet und die unbeaufsichtigte Betriebssysteminstallation kann auf dem Client durchgeführt werden.

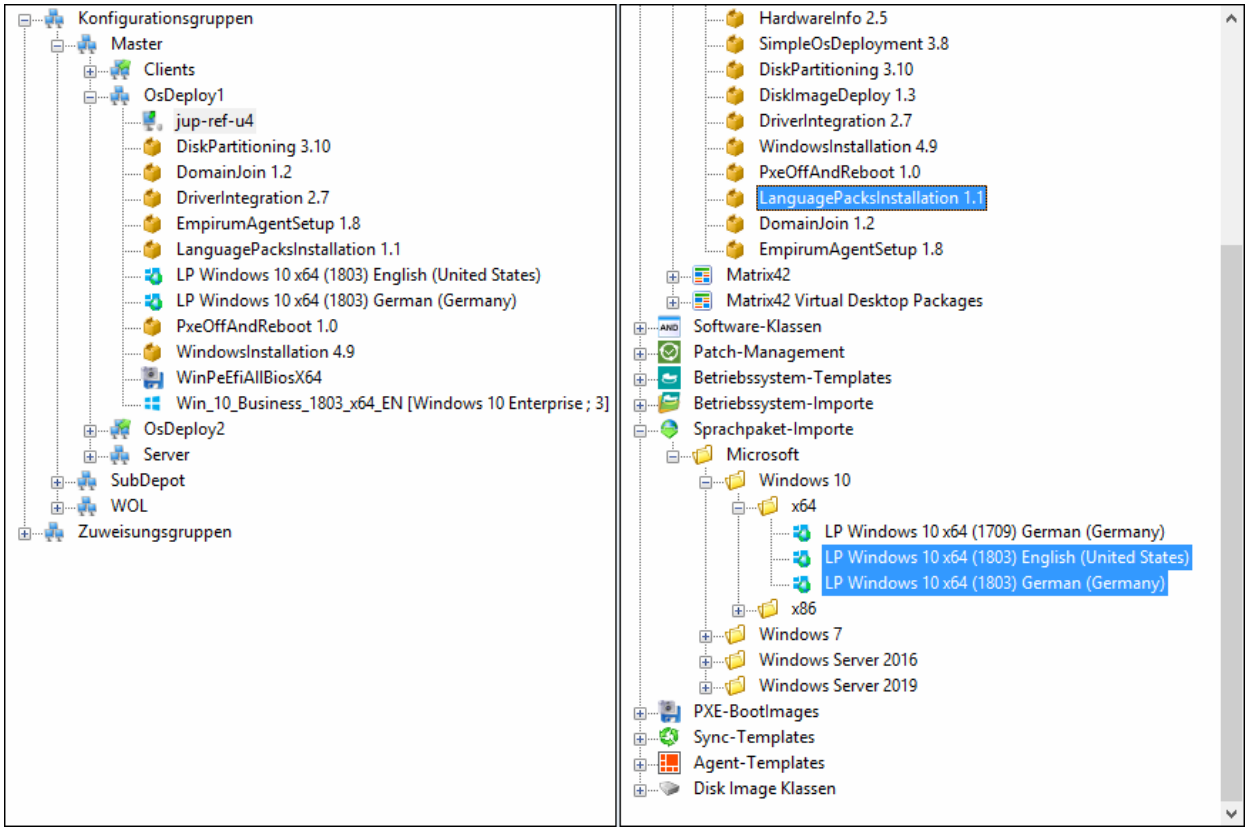
Nach der Betriebssysteminstallation wird der PE Agent auch unter Windows gestartet, um die letzten PreOS-Pakete auszuführen, die dafür vorgesehen wurden, unter Windows ausgeführt zu werden (Das sind insbesondere LanguagePacksInstallation, DomainJoin, EmpirumAgentSetup).

2.3.8 LanguagePacksInstallation-Paket

1 Weisen sie das *LanguagePacksInstallation*-Paket der entsprechenden *Konfiguration* zu. Dieses Paket installiert die zugeordneten Sprachpakete nachdem das WindowsInstallation-Paket das Windows Betriebssystem aufgesetzt hat.

> **Achtung:** Achten Sie darauf, dass im Software-Depot die Reihenfolge der Pakete eingehalten wird: LanguagePacksInstallation muss nach WindowsInstallation und PxeOffAndReboot Paketen angeordnet sein. Beachten Sie, dass die eigentliche Installation der Language Packs einige Zeit in Anspruch nehmen wird.

2 Wählen Sie die zu verwendenden *Sprachpakete* aus. Hierzu wechseln Sie in die Sprachpaket-Importe im rechten Baum. Dort werden alle importierten Windows 10 Sprachpakete in der Ordnerstruktur angezeigt. Weisen Sie die gewünschten Sprachpakete dem Computer im mittleren Baum zu.



> **Hinweis:** Nach den WindowsInstallation und PxeOffAndReboot-Paketen wird unter Windows 10 das Paket LanguagePacksInstallation laufen und die zugewiesenen Sprachpakete auf dem Client installieren.

2.3.8.1 Anzeige-Sprache einstellen

Das WindowsInstallation-Paket führt die Variable "UILanguage" um die Anzeigesprache zu konfigurieren.

Falls das Betriebssystem-Image bereits über die konfigurierte Sprache verfügt, wird diese beim Windows Setup über die Ausführung des WindowsInstallation-Paketes gesetzt.

Falls die Sprache erst mit der Installation von Language Packs durch das LanguagePacksInstallation-Paket hinzukommt, dann wird die Anzeigesprache auch über das LanguagePacksInstallation-Paket gesetzt. Hierfür reicht WindowsInstallation-Paket die zu setzende Sprache an das LanguagePacksInstallation-Paket weiter.

Die Einstellung der Anzeigesprache wird also im WindowsInstallation-Paket über die Variable "UILanguage" definiert.

2.3.9 DomainJoin-Paket

Das DomainJoin-Paket wird ausgeführt, um den Computer zu einer Domäne hinzuzufügen.

Dafür müssen die Variablen DomainJoinCredentialsUser (Domäne\Benutzername), DomainJoinCredentialsPassword, DomainJoinErrorAction und DomainJoinAuthority gesetzt werden. Die Variablen FQDN und ORGANIZATIONAL_UNIT, die über den Computereigenschaftendialog konfiguriert werden können, werden von diesem Paket verwendet.

- Variablen, die die Zugangsdaten für den Domain-Join festlegen:
- *DomainJoinCredentialsUser*:

  Name mit Domain des zu verwendenden Domänen Benutzers.
- *DomainJoinCredentialsPassword*:

  Passwort des zu verwendenden Benutzers.
- *DomainJoinErrorAction*:

  Fehlerbehandlung während des Anmeldeprozesses.
  - *Warning*:

    Bei einem Fehler während des Anmeldeprozesses in eine andere OU erfolgt eine Warnmeldung im Log, dass der Client nicht in der angegebenen OU angemeldet werden konnte. Der Installationsprozess wird weiter durchgeführt, wenn möglich.
  - *Error*:

    Bei einem Fehler während des Anmeldeprozesses in eine andere OU erfolgt eine Fehlermeldung im Log, der Installationsprozess wird abgebrochen.
- *DomainJoinAuthority:*

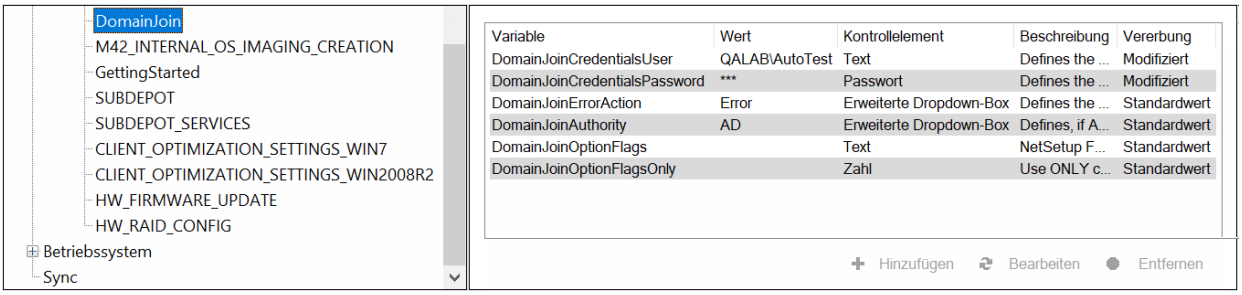
  Definiert, ob die AD oder Empirum den Anmeldeprozess regelt. Nur wenn hier *Empirum* ausgewählt wurde, kann ein bereits vorhandener Client in eine andere OU verschoben werden.
- *DomainJoinOptionFlags:*

  Definiert eine kundenspezifische NETSETUP OptionFlag Bitmaske, mit der der Join Prozess vor oder anstatt dem Empirum DomainJoin ausgeführt werden kann. Erlaubt sind hier leere Werte (Standard), dezimale Werte, oder hexadezimale Werte im Format "0x0000". Weitere Informationen:

  https://docs.microsoft.com/en-us/windows/win32/api/lmjoin/nf-lmjoin-netjoindomain
- *DomainJoinOptionFlagsOnly:*

  Wenn DomainJoinOptionFlags definiert ist, kann über den Wert DomainJoinOptionFlagsOnly bestimmt werden, ob nur über OptionFlags ein DomainJoin

ausgeführt werden soll (Wert "1") oder beim Misserfolg noch ein Standard Empirum Domainjoin ausgeführt werden soll (Wert "0" oder leer).

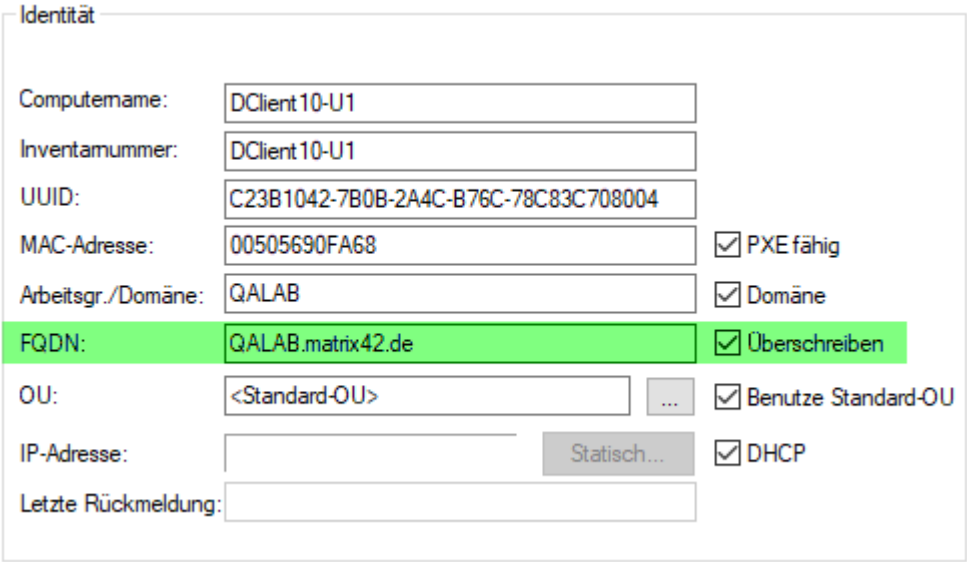


Achtung: Die Funktion "Verschieben in eine andere OU" ist erst ab der OS-Installation von Windows 10 ab Build-Version 1809 verfügbar!

1 Stellen Sie sicher, dass in den *Eigenschaften des Computers* der *FQDN* Wert gesetzt ist. Dieser Wert muss für den Domain-Join angegeben werden.

Hinweis: Ist der FQDN Wert nicht gesetzt, dann wird auch kein Domain-Join durchgeführt. Der Computer wird in der "Arbeitsgruppe" verbleiben.
Soll der Computer in eine *Arbeitsgruppe* integriert werden, muss das Häkchen bei *Domäne* entfernt und die *Arbeitsgruppe* eingetragen werden. Es darf dann **kein** *DomainJoin Paket* zugeordnet sein!

2 Öffnen Sie die *Computereigenschaften* über das Kontextmenü des Computers. Der FQDN Wert kann dann gesetzt werden, wenn die *Überschreiben* Eigenschaft aktiviert ist (Häkchen gesetzt).



3 Falls gewünscht, kann auch die *OU* (*Organizational Unit im Active Directory*) angepasst werden.

4 Bestätigen Sie die Änderungen mit *OK*.

2.3.10 EmpirumAgentSetup-Paket

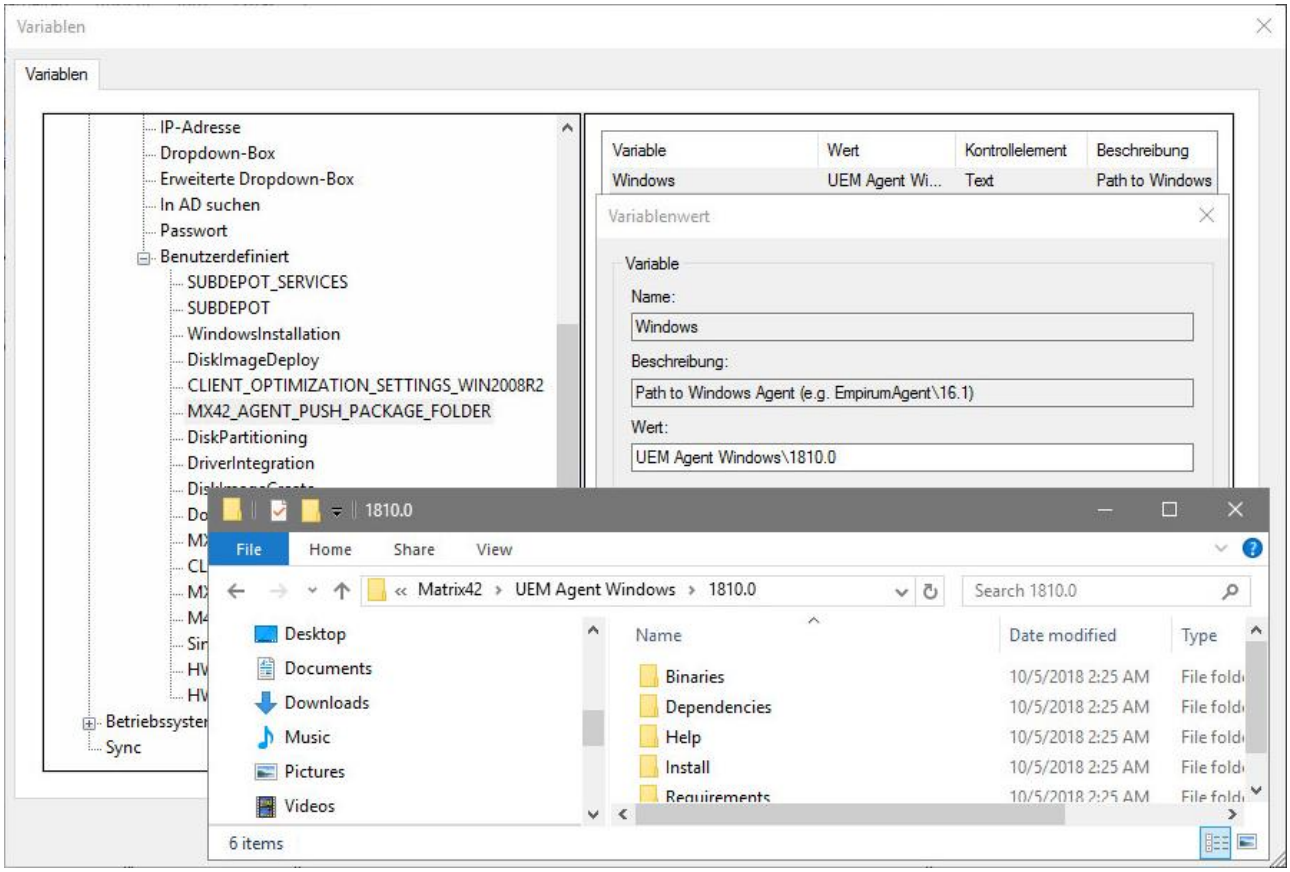
Das EmpirumAgentSetup-Paket wird ausgeführt, um den Empirum Agent auf dem Client zu installieren.

Die Variable *M42_AGENT_PUSH_PACKAGE_FOLDER.Windows* wird dafür verwendet, die Version und die Variante (Matrix42 Advanced Agent oder Matrix42 UEM Agent) des Empirum Agenten festzulegen.

Der Wert dieser Variable ist ein relativer Pfad zum gewünschten Agent-Verzeichnis beginnend mit dem Ordner, der sich unter "`Configurator$\Packages\Matrix42\`" befindet. Zum Beispiel:

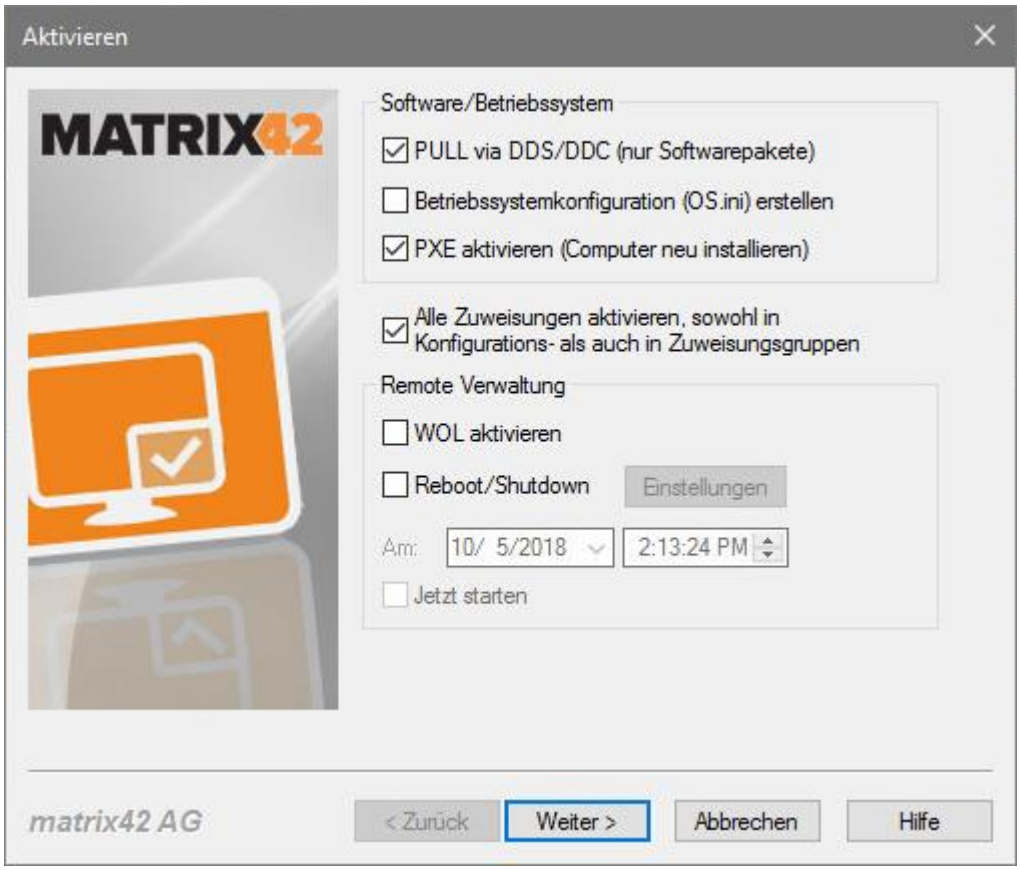
- UEM Agent Windows\2004.0
- EmpirumAgent\19.0

> **Hinweis:** EmpirumAgentSetup 2.2 und höher ermittelt und verwendet die neueste zur Installation freigegebene Matrix42 UEM Agent Version, falls kein Wert gesetzt ist.
>
> Ältere Versionen von EmpirumAgentSetup verwenden standardmäßig den Matrix42 Advanced Agent *EmpirumAgent\19.0*, oder *EmpirumAgent\18.0*.



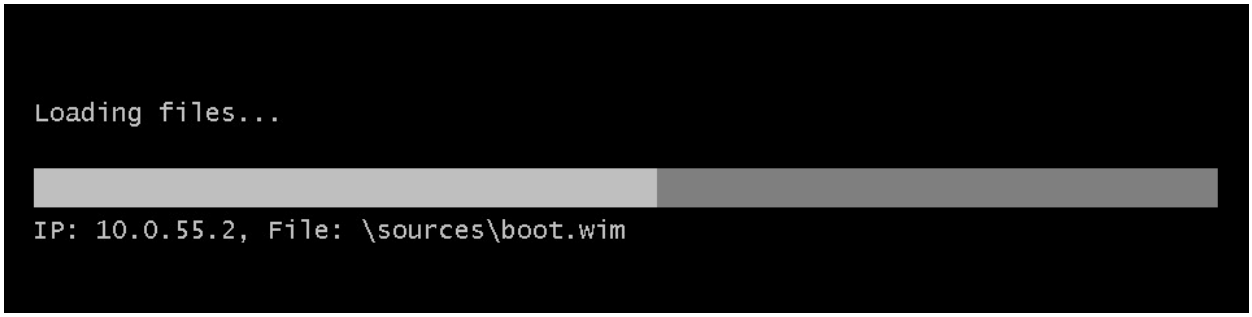
1 Sind alle *Zuweisungen* vorgenommen, kann der Computer über das *Kontextmenü aktiviert* werden.

2 Im sich öffnenden Aktivierungsassistenten ist die Aktivierung *PULL via DDS/DDC (nur Softwarepakete)* und *PXE aktivieren (Computer neu installieren)* anzuhaken.



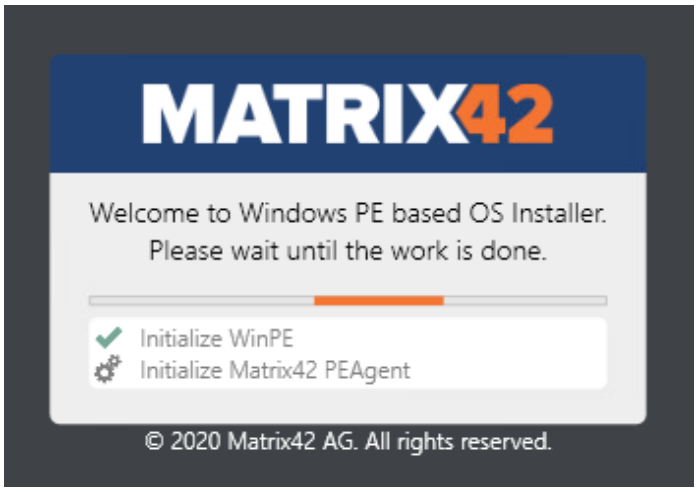
3 Klicken Sie auf *Weiter*. Der Aktivierungsassistent ist im Anschluss mit *Fertig stellen* zu bestätigen.

4 Der aktivierte Computer kann nun per *Netzwerk/PXE* gebootet werden und wird das erstellte WinPE basierte PXE-Boot-Image laden.

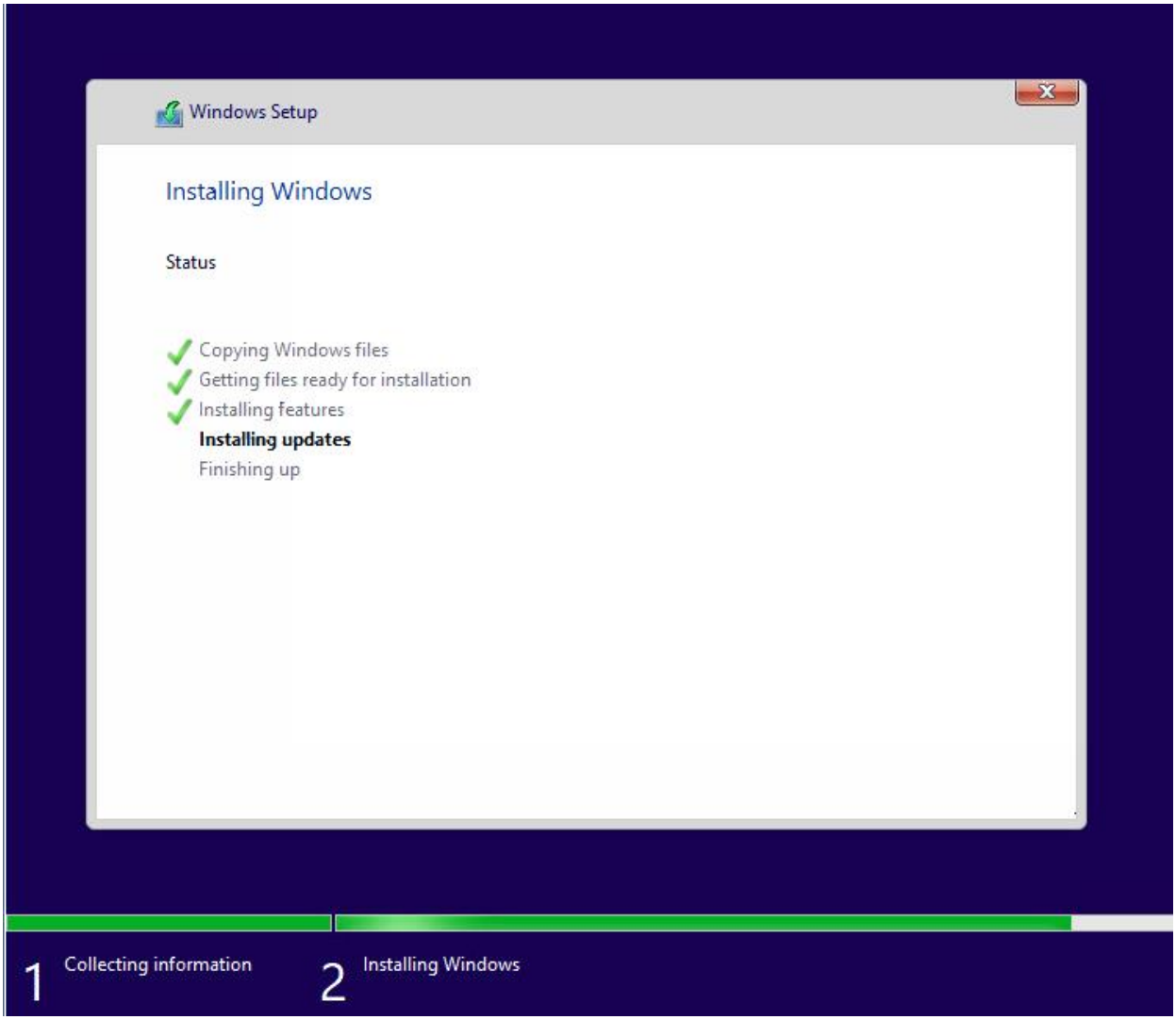


5 Wird das geladene WinPE auf dem Computer gestartet, startet automatisch das *Matrix42 Universal Agent Framework*, welches die zugewiesenen Pakete nacheinander ausführt.

Achtung:
In der aktuellen Version ist ein manuelles Eingreifen in die Ausführung des Matrix42 Universal Agent Framework möglich.
Dies soll Ihnen ermöglichen, die Abläufe möglichst einfach zu analysieren und ggf. korrigieren zu können.
Ein manueller Eingriff kann ggf. zu einem Abbruch der Installation führen!



Bei der Ausführung des *WindowsInstallation Paketes* wird dann auch die *Windows Installation* ausgeführt.

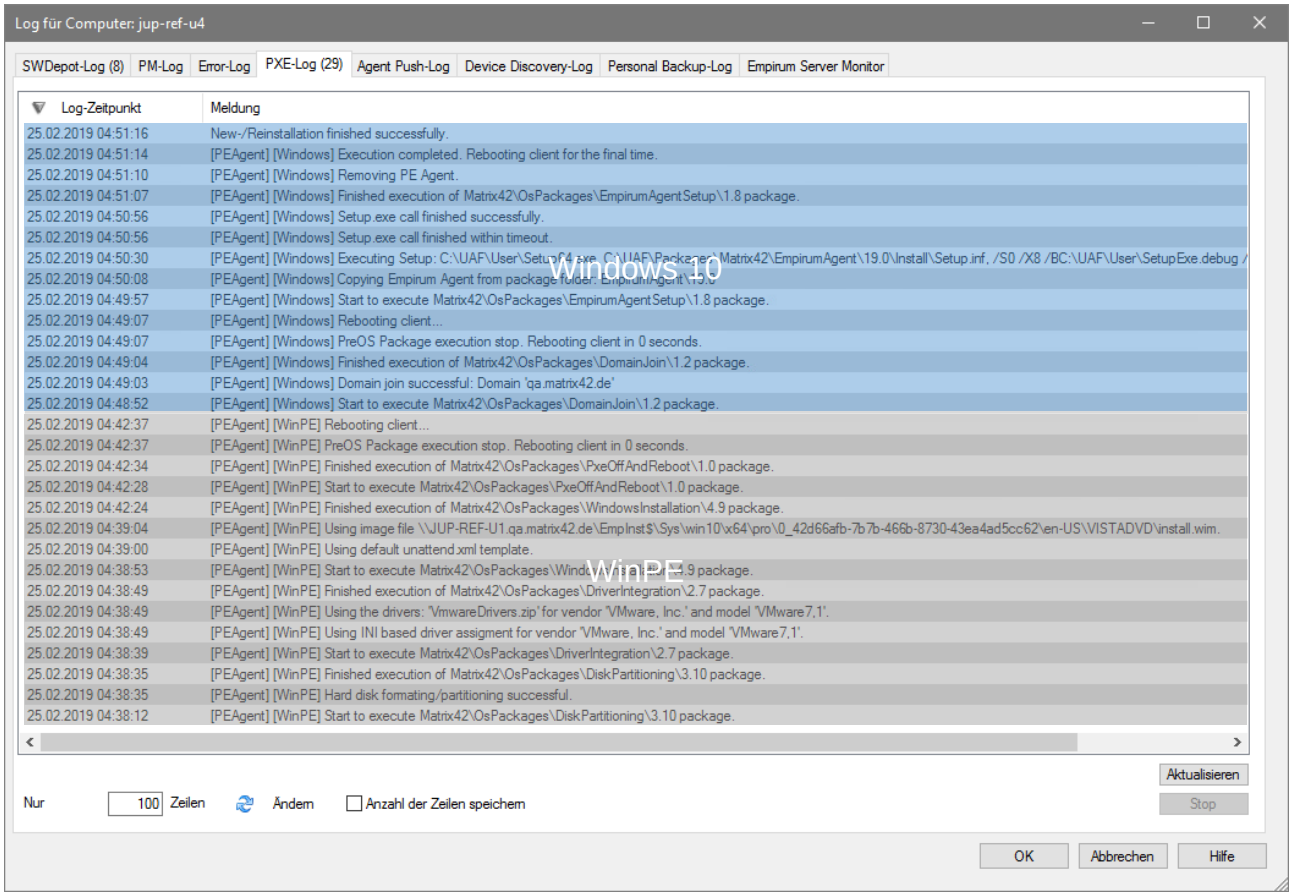


Nach dem *Durchlaufen der verschiedenen Installationsphasen* ist das Betriebssystem auf dem Computer installiert.

2.4 PXE-Log

Während eines Deployments mit diesen Paketen finden mehrere Neustarts statt. Das Betriebssystem wird dabei mehrmals ausgeführt. Während dessen führt der PE-Agent die zugeordneten Pakete aus: LanguagePacksInstallation, DomainJoin und EmpirumAgentSetup. Der PE-Agent selbst wird mit dem ersten booten von Windows in der Phase des ersten Log-ons installiert und entfernt sich selbst nach erfolgreicher Ausführung aller Pakete.

Um die Situation zu verstehen oder zu verfolgen, empfiehlt es sich, in das PXE-Log des entsprechenden Clients reinzuschauen.



3 Matrix42 OS Deployment mit WinPE im Empirum Depot Server-Szenario

Die Verwendung vom Matrix42 OS Deployment mit WinPE im Empirum Depot Server-Szenario ist möglich. Hierzu sind allerdings folgende Punkte zu berücksichtigen:

- Die Verwendung des richtigen Agenten Template in der Bootkonfiguration.
- Die Verwendung eines zusätzliches Empirum Sync-Template auf dem Empirum Depot.

Die Punkte werden in den folgenden Kapiteln genauer ausgeführt.

3.1 Agent-Template im Empirum Depot Server-Szenario

Das Matrix42 Universal Agent Framework im WinPE basierten PXE Image entnimmt die Informationen aus dem Agent-Template, wie es sich mit dem Empirum-Server verbinden soll.

Soll ein Computer über ein Empirum Depot aufgesetzt werden, dann muss in der verwendeten Boot-Konfiguration ein Agent Template ausgewählt werden, bei dem dieses Empirum-Depot als Ausfall-Server angegeben ist.

3.1.1 DHCP-Optionen

Eine weitere Variante ist die Verwendung von DHCP-Optionen im Depot-Szenario. Über die DHCP-Optionen können die folgenden Werte aus dem Agent Template übersteuert werden:

- Empirum Server (Option EmpirumServer).
- Benutzer mit dem die Verbindung zum Empirum Server hergestellt werden soll (Option EmpUser).
- Passwort für den Benutzer mit dem die Verbindung zum Empirum Server hergestellt werden soll (Option EmpPW).

> **Hinweis:** Das Password kann Sync-, AES256-verschlüsselt oder unverschlüsselt angegeben werden.

> **Hinweis:** Sind nicht alle drei DHCP Optionen (EmpirumServer, EmpUser, EmpPW) auf dem DHCP-Server konfiguriert, dann werden nur die konfigurierten Werte verwendet und übersteuern die entsprechenden Agent Template Werte. Somit ist es beispielsweise möglich, nur den EmpirumServer Wert des Agent Templates zu übersteuern, wenn der Benutzername und das Passwort gleichbleiben.

Vorteil dieser Variante ist, dass eine Bootkonfiguration für mehrere Depots verwendet werden kann.

Um die DHCP-Optionen während des Deployment mit WinPE verwenden zu können, sind folgende Schritte notwendig:

1 *DHCP-Optionen* in *Matrix42 DBUtil* einrichten.

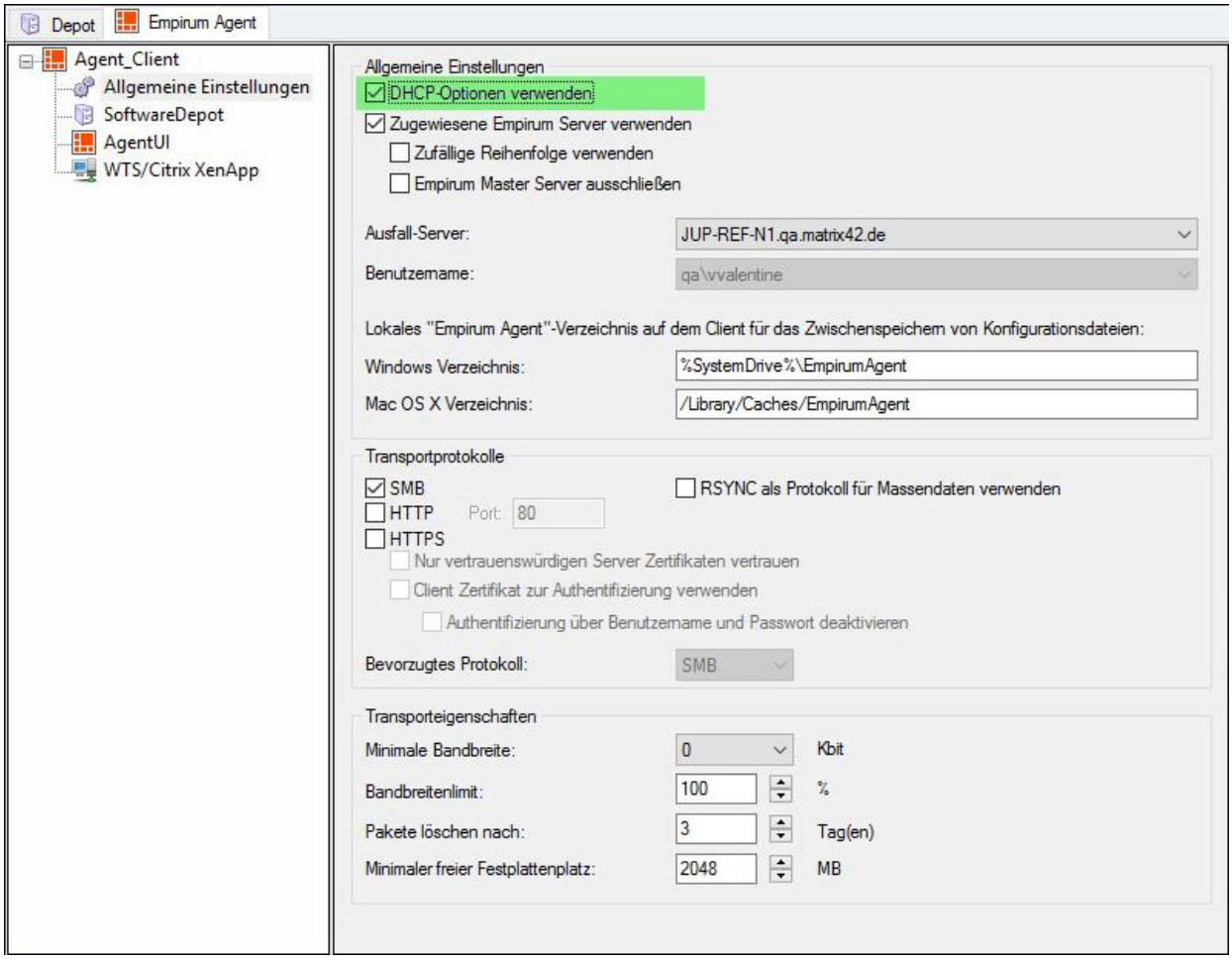
> **Hinweis:** In der Online-Hilfe DHCP-Optionen einrichten wird beschrieben wie die DHCP-Optionen mit Matrix42 DBUtil eingerichtet werden können. Der PE-Agent berücksichtigt die folgenden Optionen für den Verbindungsaufbau zum Depot-Server:

- EmpirumServer
- EmpUser
- EmpPW

2 Einrichten der *DHCP-Optionen* auf dem *DHCP-Server*.

> **Hinweis:** Soll die DHCP-Option *EmpPW* verwendet werden, dann kann dieser Wert auf dem DHCP-Server Sync-, AES256-verschlüsselt oder unverschlüsselt angegeben werden.

3 Im *Agent-Template*, das von der Bootkonfiguration verwendet wird, muss die allgemeine Einstellung *DHCP-Optionen verwenden* angehakt sein.



4 Die *Bootkonfigurationen*, die das *Agent-Template* verwenden, müssen im Anschluss *neu gespeichert* werden, damit die Änderungen des Agent-Templates im Image landen.

3.2 Zusätzliches Empirum Sync-Template

Das Matrix42 Universal Agent Framework im WinPE basierten PXE Image verwendet eine Konfigurationsdatei DeviceMapping.xml die sich auf dem Empirum-Share verbindet, um den Namen des zu bespielenden Computers anhand seiner MAC-Adresse zu bestimmen. Diese Datei wird vom einer Backend Task Queue Erweiterung auf dem Empirum-Share des Master Servers geschrieben.

Wenn Empirum Depots beim dem WinPE basierten OS Deployment verwendet werden, muss das Sync-Template ESubdepot_DeviceMapping dem verwendeten Depot zugewiesen werden und die Änderung muss durch die Aktivierung des Depots mitgeteilt werden.

4 Treiberintegration für die Betriebssysteminstallation

Benötigt ein Gerät Treiber, die in Windows noch nicht enthalten sind, kann dies über das zusätzliche Matrix42 PreOS Paket *DriverIntegration* umgesetzt werden.

> **Hinweis:** Eine detaillierte Schritt für Schritt Anleitung zur Handhabung der Treiber ist im Dokument "Matrix42 Treiberintegration in WinPE - HowTo - DE.pdf" beschrieben.
> Für die Dateiablage und alle Pfadangaben gilt eine maximale Pfadlänge von **256** Zeichen. Die folgende Zeichen **~ " # % & * : < > ? / \ { | }** sind in Ordner, Datei und Pfadangaben nicht erlaubt.

Die folgenden Matrix42 PreOS Pakete kommen in diesem Zusammenhang zum Einsatz:

- DiskPartitioning
- DriverIntegration
- WindowsInstallation
- PxeOffAndReboot
- DomainJoin
- EmpirumAgentSetup

Das DiskPartitioning Paket löscht die Partitionen der ersten Festplatte des Computers und legt die notwendigen Partitionen an. Dabei kann über die Variablen des Paketes die Partitionierung konfiguriert werden.

Mit dem DriverIntegration Paket wird die Systempartition von Windows vorbereitet, damit Treiber für den Computer übernommen werden.

Mit dem WindowsInstallation Paket wird dann die eigentliche Windows Installation durchgeführt, die dann die vorbereiteten Treiber berücksichtigt.

> **Hinweis:** Das Deployment mit den vorgestellten Paketen ist auf die erste Festplatte des Computers eingeschränkt.

4.1 DriverIntegration Paket

Für die Integration der Betriebssystemtreiber ist das Matrix42 PreOS Paket *DriverIntegration* zuständig. Anhand des Herstellernamens, Modellnamens und des Betriebssystemtyps (Windows 7, Windows 10, Windows Server 2016 oder Windows Server 2019) des zu bespielenden Rechners wird mit Hilfe einer Steuerungsdatei (Drivers.json) bestimmt, welches Treiberarchiv oder Treiberverzeichnis für die Betriebssysteminstallation verwendet werden soll.

Die Steuerungsdatei finden Sie auf dem Empirum-Server unter:

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

Mit Hilfe des *WinPE Driver Assistant* kann die Steuerungsdatei (Drivers.json) angepasst werden.

4.2 Ermitteln des Herstellers und des Modells

Um die automatisierte Zuordnung von Treiberpaketen richtig ausführen zu können, benötigen wir die Hersteller und Modellbezeichnung, wie sie durch die Windows Management Instrumentation ermittelt wird.

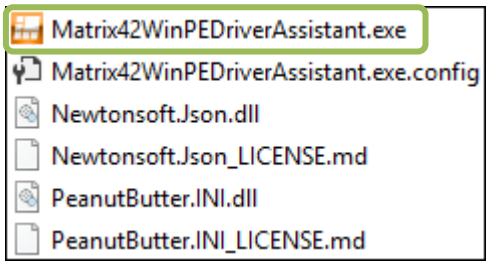
Hierfür stellen wir Ihnen das Matrix42 PreOS Paket *HardwareInfo* zur Verfügung, mit dem Sie alle notwendigen Informationen ermitteln, um sie dann mit Hilfe des *Matrix42 Treiberassistenten* in die Steuerdatei importieren zu können.

Weisen Sie dem Rechner nur das Matrix42 PreOS Paket *HardwareInfo und ein WinPE basiertes PXE Image zu.* Nach dem Aktivieren des Rechners kann der Rechner gebootet werden und das HardwareInfo Paket wird ausgeführt und hinterlegt im Mac8- bzw. UUID-Verzeichnis eine Ini-Datei. Diese Ini-Datei enthält die Information über den erkannten Herstellernamen und Modelnamen des Rechners und kann mit Hilfe des WinPE Driver Assistant in die Treibersteuerungsdatei importiert werden.

> **Hinweis:** Die Namen für Hersteller und Modell werden mit dem *HardwareInfo Paket* mit Hilfe der Windows Management Instrumentation (WMI) Klassen zur WinPE Laufzeit ausgelesen (gwmi win32_computersystem).

4.3 Starten des Matrix42 Treiberassistenten

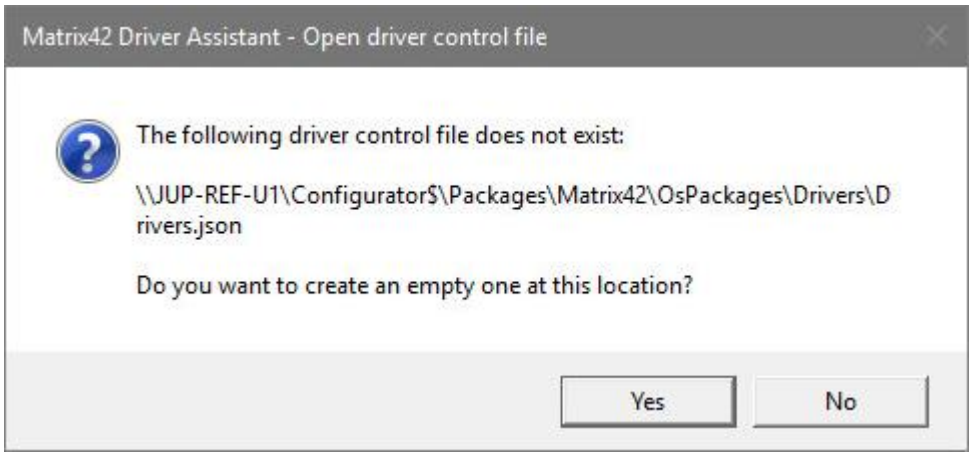
Um den Matrix42 Treiberassistenten zu starten, muss die Matrix42WinPEDriverAssistant.exe Anwendung in dem Verzeichnis `Empirum\AddOns\WinPEDriverAssistant` auf dem Empirum Master Server ausgeführt werden.



> **Hinweis:** Stellen Sie sicher, dass der Benutzer, mit dem der Assistent gestartet wird, ausreichende Berechtigungen besitzt, um die Datei Drivers.json auf dem Empirum-Share zu öffnen, zu editieren, neu zu erstellen und zu löschen. Er sollte ebenfalls die Berechtigungen besitzen, Dateien und Verzeichnisse dort hinzukopieren.

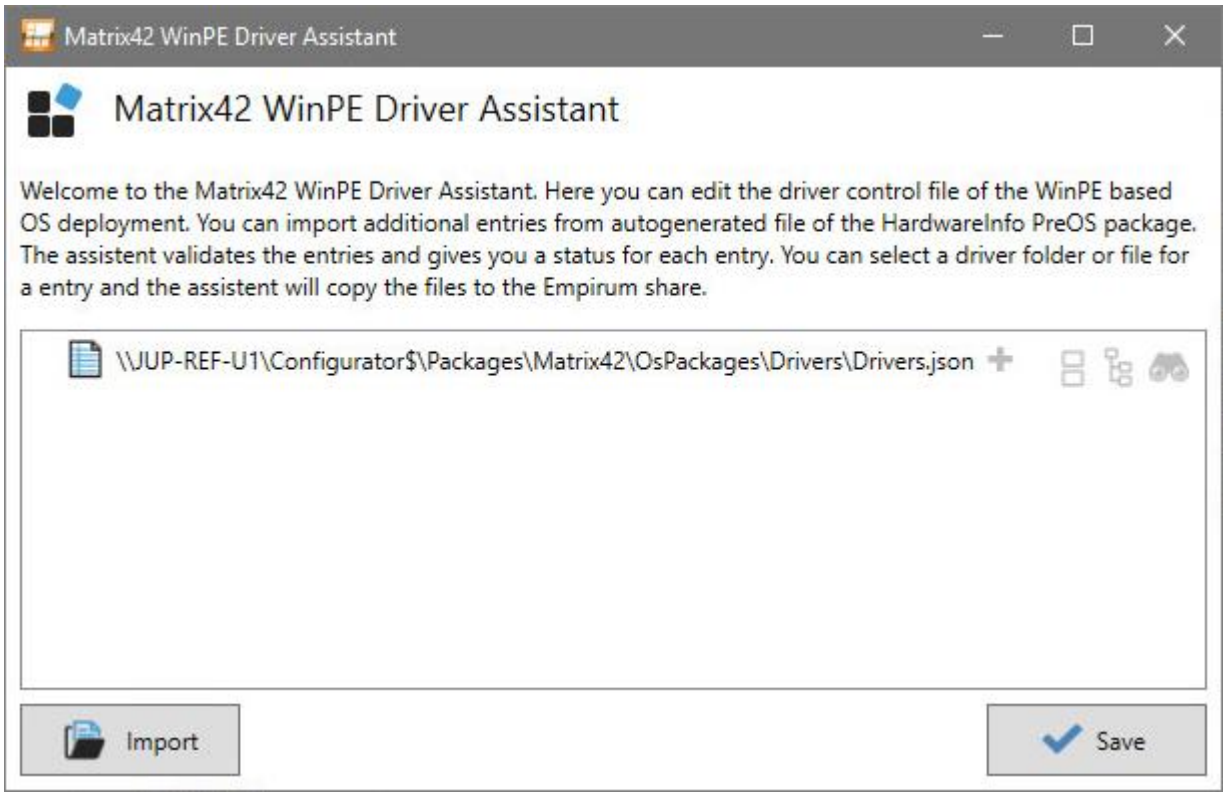
4.3.1 Erstellen der Datei Drivers.json beim ersten Start

Der Matrix42 Treiberassistent versucht beim Starten automatisch die Treiberkontrolldatei *Drivers.json* auf dem Empirum-Share zu öffnen. Ist diese noch nicht vorhanden, wird Ihnen angeboten, eine leere Treiberkontrolldatei erstellen lassen, indem Sie die folgende Meldung mit *Ja* bestätigen.



Nach dem Erstellen einer leeren Treiberkontrolldatei öffnet sich der Assistent und zeigt die geöffnete Datei an.

Als oberster Eintrag wird die Treiberkontrolldatei *Drivers.json* und ihr Dateipfad angezeigt.

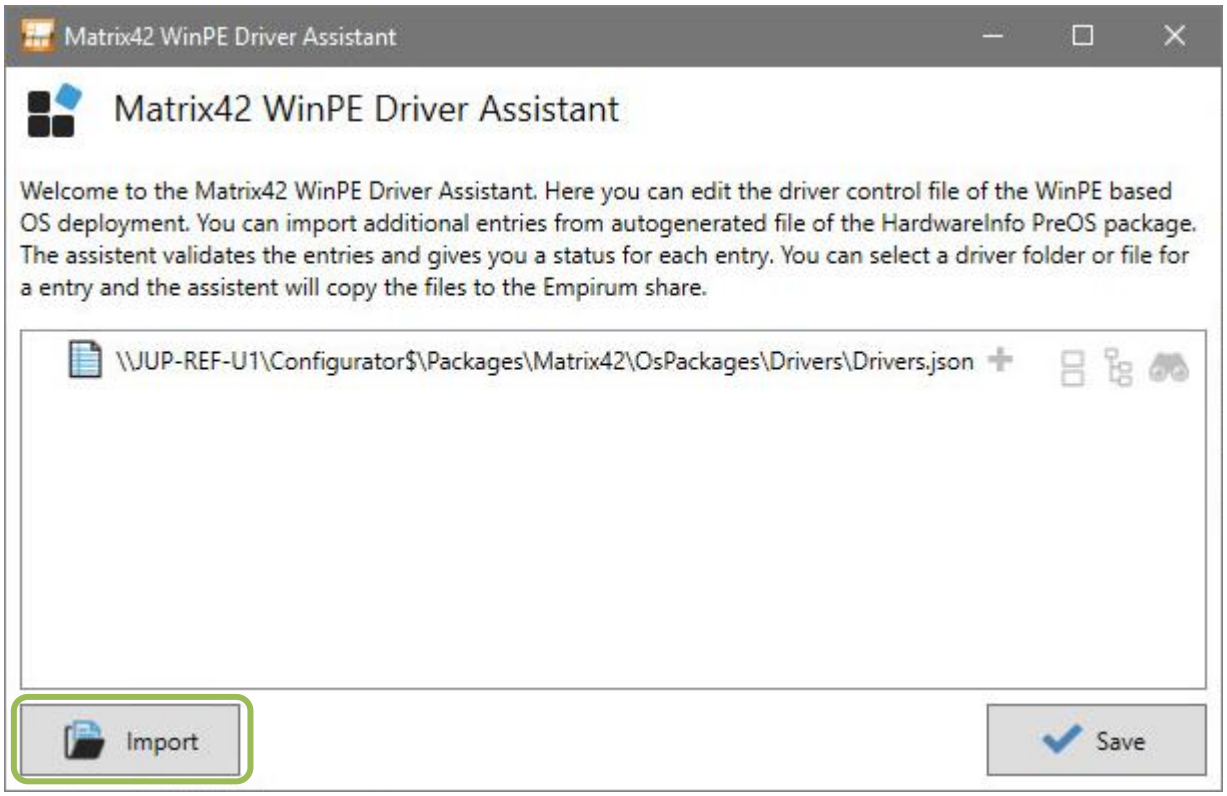


4.3.2 Import von HardwareInfo-Paket Ausgabedateien

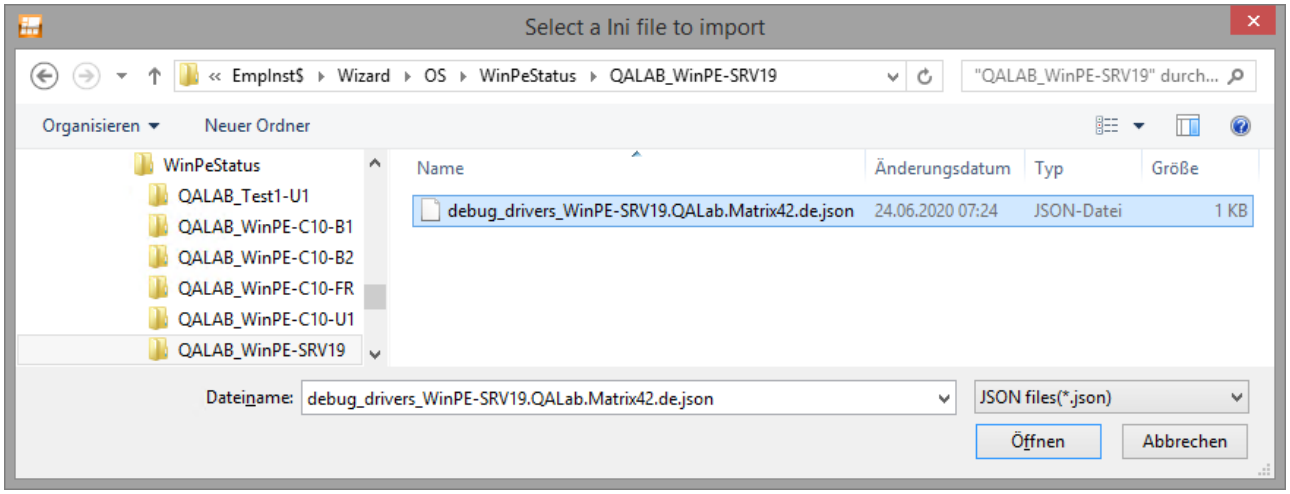
Die einfachste und sicherste Variante die Treiberkontrolldatei zu erweitern ist das Importieren von der Ausgabedatei des Matrix42 PreOS Pakets *HardwareInfo*, die das HardwareInfo-Paket automatisch im `EmpInst$\Wizard\OS\Auto` Unterverzeichnis des jeweiligen Clients erstellt. Im Kapitel 4.2 Ermitteln des Herstellers und des Modells auf Seite 40 wird genauer beschreiben, wie die diese Ausgabedatei mit Hilfe des HardwareInfo Pakets erstellt werden kann.

Diese Ausgabedatei enthält die Informationen über den Hersteller und das Model des verwendeten Clients.

Der Import lässt sich über den *Import* Schaltfläche starten.



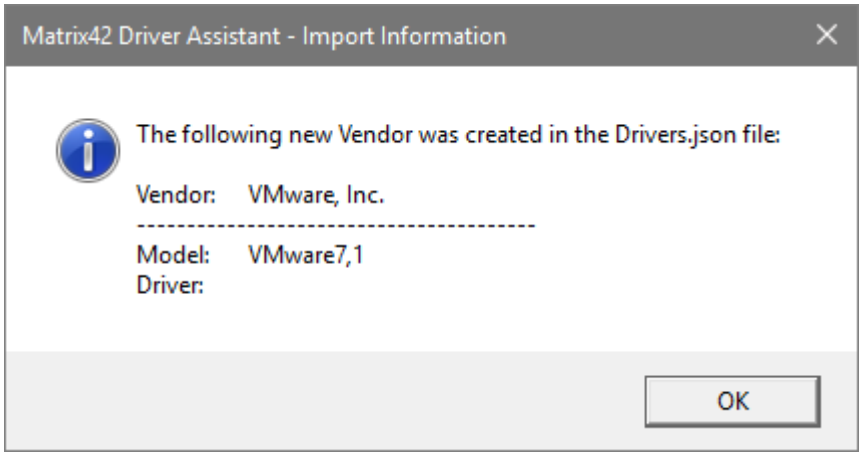
Es öffnet sich der Dateibrowser in dem `EmpInst$\Wizard\OS\WinPeStatus` Verzeichnis, mit dessen Hilfe die Ausgabedatei des HardwareInfo-Pakets geöffnet werden kann.



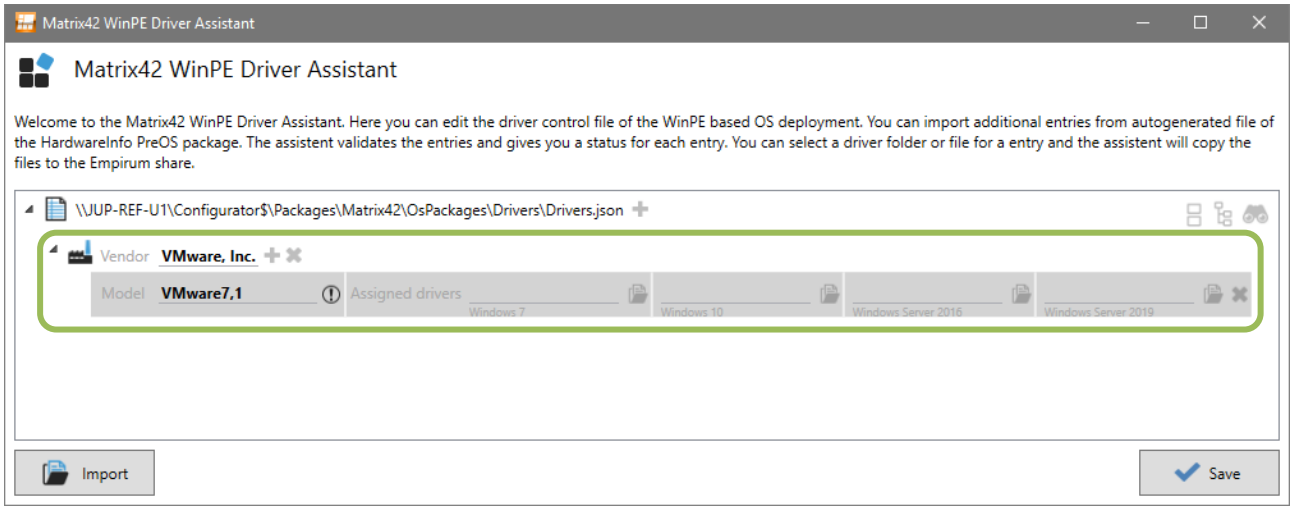
Die Ausgabedatei des HardwareInfo-Pakets liegt in dem jeweiligen Unterverzeichnis des Clients.

> **Hinweis:** Sie können hier auch die Suchfunktion des Dateibrowsers verwenden und nach dem Client-Namen suchen. Die Ausgabedatei des HardwareInfo-Pakets enthält immer den Client-Namen und die Domäne des Clients.

Wurde die Ausgabedatei gefunden und ausgewählt, kann der Import per *Open* gestartet werden. Der Assistent gibt dann eine Rückmeldung welcher Hersteller und welches Model importiert wurde.



Nach der Bestätigung mit OK wird der neue Eintrag in die Kontrolldatei eingefügt.



Für den erkannten Hersteller *VMware, Inc.* wurde ein Eintrag erstellt. Untergeordnet wurde ein Eintrag für das erkannte *Model VMware7,1* erstellt.

Dem Model-Eintrag sind momentan noch keine Treiberdateien, oder Treiberverzeichnisse für die Betriebssystem *Windows7, Windows 10, Windows Server 2016* und *Windows Server 2019* zugewiesen. Aus diesem Grund wird der Eintrag hellgrau hinterlegt angezeigt und mit einem Ausrufezeichen markiert. Das zeigt an, dass der Eintrag noch nicht gültig ist, da noch kein Treiber angegeben für ein Betriebssystem angegeben ist, bzw. ein angegebener Treiber nicht auf dem Empirum-Share vorhanden ist.

4.3.3 Treiberdatei importieren

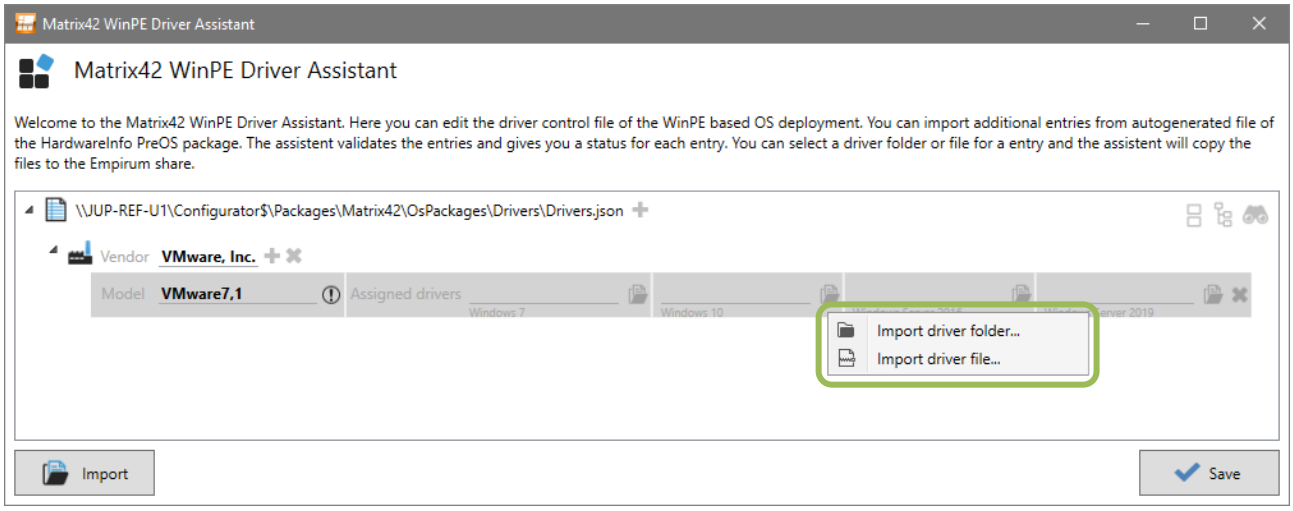
> **Hinweis:** Die meisten Hersteller bieten mittlerweile sogenannte SCCM Treiber Pakete an. Diese Treiberpakete enthalten in der Regel alle Treiber, die vom Hersteller für die Installation als Treiber Installationsdatei freigegeben wurden.

> Dies kann, muss aber nicht zwangsweise alle Gerätetreiber für das Hardwaremodell beinhalten. Häufig werden Treiber, die eine zusätzliche Software Komponente beinhalten, nicht in solchen Paketen erfasst. Hier bietet sich an, diese als separate Softwarepakete auf den Client auszubringen. In der Regel ist der Umfang der Treiber in dem SCCM Paket der Dokumentation des Herstellers zu entnehmen, zumeist werden auch die nicht enthaltenen Treiber explizit gelistet.

Das Matrix42 WinPE basierte OS Deployment unterstützt aktuell sowohl nicht gepackte Treiberverzeichnisse als auch komprimierte Archive vom Typ .cab oder .zip.

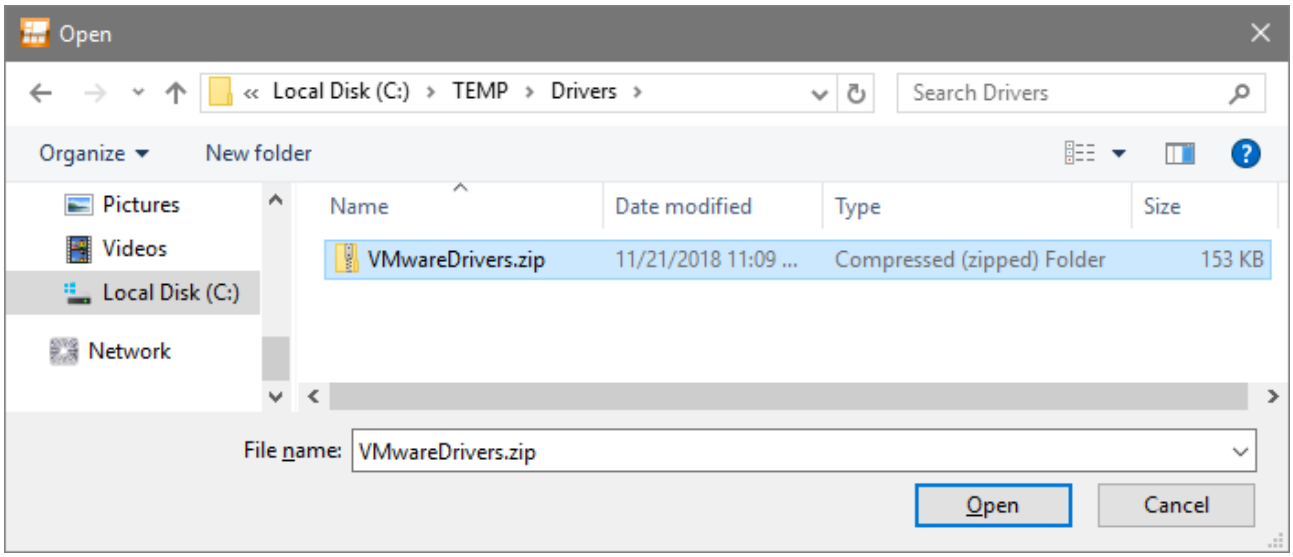
Sie können die Treiberimport Schaltfläche hinter dem jeweiligen Betriebssystemeintrag drücken, um eine Treiberdatei oder ein Treiberverzeichnis auszusuchen, dass dann auf den Empirum-Share kopiert wird.

In diesem Fall wird der Treiberimport Schaltfläche des Windows 10 Eintrages gewählt, um dem Model einen Treiber für Windows 10 zu importieren.

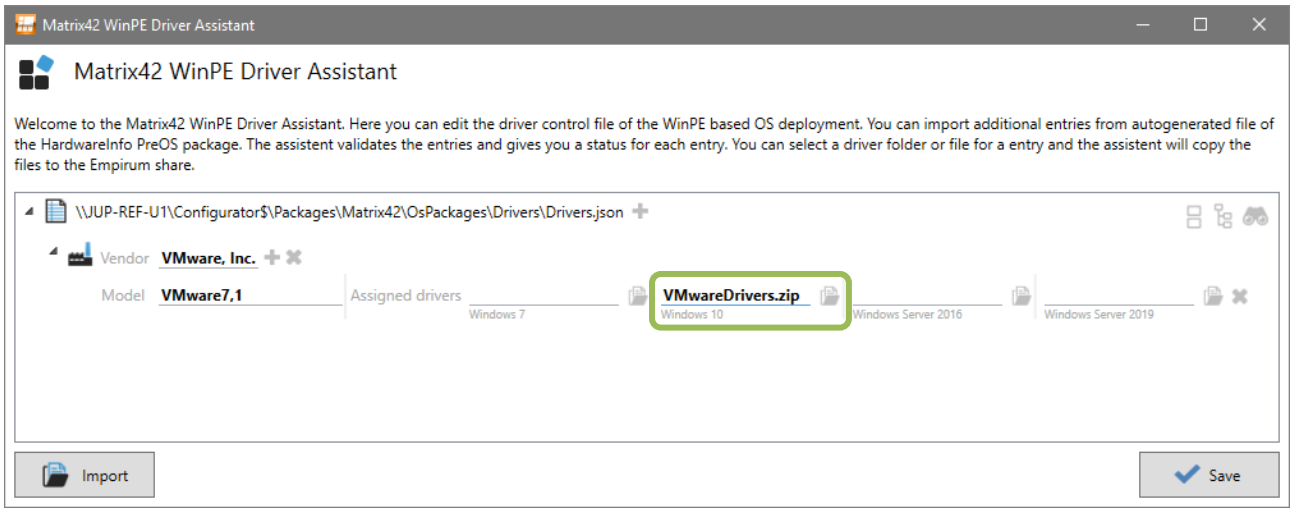


Es öffnet sich ein Kontextmenü mit dessen Hilfe entschieden werden kann, ob eine einzelne Treiberdatei, oder ein komplettes Treiberverzeichnis importiert werden soll. In diesem Fall soll ein einzelnes Zip-Archiv importiert werden, weswegen der *Import driver file* Eintrag ausgewählt wird.

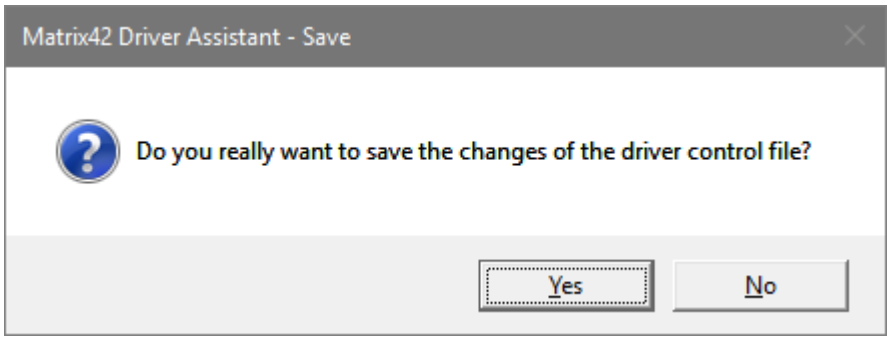
Es öffnet sich der Dateibrowser mit dessen Hilfe nun das Treiberarchiv ausgewählt werden kann.



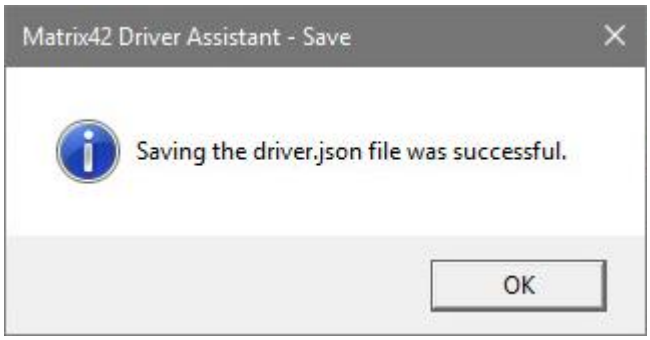
Wird die Auswahl mit *Open* bestätigt, wird die ausgewählte Treiberdatei auf den Empirum-Share kopiert und der *Assigned driver* Wert des Model-Eintrages wird angepasst.



Mit Hilfe des *Save* Schaltfläche können nun die Änderungen gespeichert werden. Es öffnet sich der Bestätigungsdialog.



Bestätigen Sie die Meldung mit *Ja*, so werden die Änderung in der Steuerungsdatei auf dem Empirum-Server abgespeichert.



Die Bestätigung des erfolgreichen Speichervorgangs erscheint, bestätigen Sie mit *OK*, dies bringt Sie zurück in die Ansicht des Treiberassistenten.

Damit ist die Steuerungsdatei für das *DriverIntegration* Paket vorbereitet und ein entsprechender Treiber ist für den Rechnertyp hinterlegt. Das Deployment kann somit durchgeführt werden.

5 SimpleOsDeployment Paket

Das Matrix42 OS Deployment mit WinPE bietet mit Hilfe des SimpleOsDeployment-Pakets eine einfache Variante, einen Computer mit Windows 10 zu bespielen. Es ist für Testzwecke sehr gut geeignet - für den produktiven Einsatz empfiehlt es sich, die Pakete DiskPartitioning, WindowsInstallation und ggfs. DriverIntegration zu verwenden.

Eine Erweiterung der von Windows vorgesehen Treiber wie sie im WindowsInstallation-Paket umgesetzt ist, ist dabei nicht vorgesehen.

Eine einfache Partitionierung der Festplatte ist Bestandteil des SimpleOsDeployment-Paketes und es setzt daher keine weiteren Pakete voraus.

Das SimpleOsDeployment Paket ist gleich zu konfigurieren wie das WindowsInstallation-Paket (siehe Kapitel über die Zuweisung in der Administration).

Hierzu sind drei Schritte notwendig:

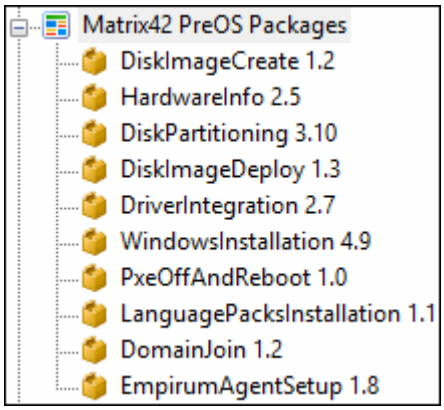
- Paketvariablen setzen
  - Variablen für die Erstellung eines Lokalen Administrator Kontos
    - LocalUserName (Anmeldename für das lokale Konto)
    - LocalUserPassword (Passwort für das lokale Konto)
    - LocalUserDisplayName (Anzeigename des lokalen Kontos)
  - Variablen, die die Sprache festlegen
    - SetupUILanguage (Definiert die Sprache, die in Windows Setup und Windows Deployment Services verwendet werden soll. z.B. en-US)
    - InputLocale (Gibt die Eingabesprache und Eingabemethode für Eingabegeräte an, z. B. das Tastaturlayout. z.B. en-US)
    - SystemLocale (Gibt die Standardsprache an, die für Nicht-Unicode-Programme verwendet werden soll. z.B. en-US)
    - UILanguage (Gibt die zu installierende Sprache an, die als Standardsystemsprache für die Anzeige von Elementen der Benutzeroberfläche (z. B. Menüs, Dialogfelder und Hilfedateien) verwendet wird. z.B. en-US)
    - UserLocale (Spezifiziert die benutzerbezogenen Einstellungen für die Formatierung von Datum, Uhrzeit, Währung und Zahlen in einer Windows-Installation. z.B. en-US)
  - Variablen, die die Credentials für den Domain Join festlegen
    - DomainJoinCredentialsUser (Name des zu verwendenden Benutzers)
    - DomainJoinCredentialsDomain (Domain-Angabe für den zu werdenden Benutzer)
    - DomainJoinCredentialsPassword (Passwort des zu verwendenden Benutzers)
- OS_PACKAGE_SOURCE Variable setzen, um die Betriebssystemquelle zu definieren.
- FQDN Eigenschaft am Computer setzen.
- TimeZone Variable unter OS_RegionalSettings setzen, um die Zeitzone unter WinPE und im zu installierenden Betriebssystem einzustellen.
  - z.B. (UTC+1:00) Amsterdam, Berlin, Bern, Rome, Stockholm, Viena

6 Disk Imaging

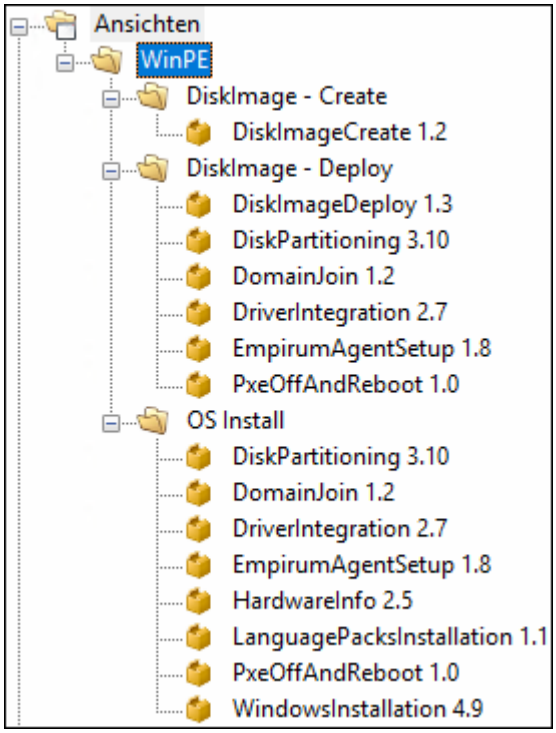
Wenn Sie bereits ein Windows 10 Betriebssystem wie im Kapitel *2.3.5 WindowsInstallation* Paket *ab Seite 24* beschrieben, installiert und konfiguriert haben, gibt es jetzt auch die Möglichkeit, diese komplette Betriebssystemkonfiguration per Disk Imaging über das Paket - *DiskImageCreate* - abzuziehen und auf andere Computer über das Paket - *DiskImageDeploy* - zu verteilen.

Das Disk Imaging per WinPE verhält sich analog zum Disk Imaging per EPE4. Auch hier werden die gleichen Disk Imaging Variablen verwendet. Siehe Disk Imaging - HowTo und VHD-Erstellung im Offline Szenario - HowTo.

Folgende PreOS Packages werden für das Abziehen und das Verteilen eines Disk Images benötigt. Diese müssen in der *Empirum Management Console (EMC)* unter *Konfiguration > Software Management > Depot > Register Matrix42 PreOS Packages* ***in dieser Reihenfolge vorliegen***:



Sehr hilfreich ist es, wenn Sie die benötigten PreOS Pakete für die jeweilige Aktion erst einmal über *Konfiguration > Ansichten Designer* konfigurieren. Danach können diese einfach per Drag & Drop einer Gruppe zugeordnet werden.

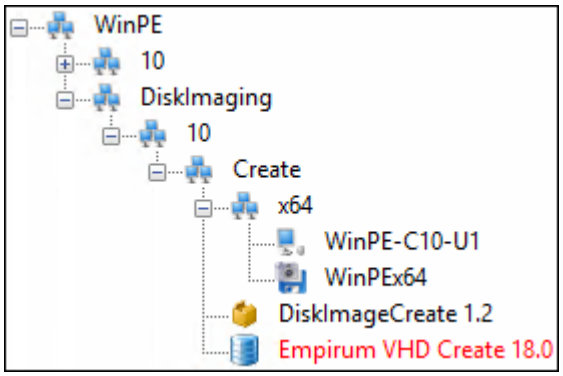


6.1 DiskImageCreate

Ausgangsbasis ist ein möglichst neu installiertes und konfiguriertes Windows 10 Betriebssystem. Das kann per EPE 4 oder auch per WinPE installiert worden sein. In Versionen vor Windows 10 (1803) gab es immer wieder Probleme beim Ausführen von Sysprep (verschiedene Apps mussten zuerst deinstalliert werden), das konnten wir mit der Version 1803 vom Juli 2018 nicht feststellen, aber mit der Version 1809 vom Februar 2019. Die Anmeldung an Windows sollte generell nur mit EINEM Benutzer (Installationsbenutzer) erfolgen. Wenn mehrere Benutzer angemeldet waren, müssen die zu deinstallierenden Apps für jeden angemeldeten Benutzer deinstalliert werden!

Hinweis: Es ist ratsam, von diesem fertig konfigurierten OS einen Snapshot zu erstellen, damit man, wenn noch etwas fehlt, an genau dieser Stelle wieder beginnen kann. Wurde einmal Sysprep ausgeführt, ist diese Konfiguration nicht mehr zu verwenden!

1 Erstellen Sie sich eine beliebige *Konfigurationsgruppe* nach folgendem Beispiel:



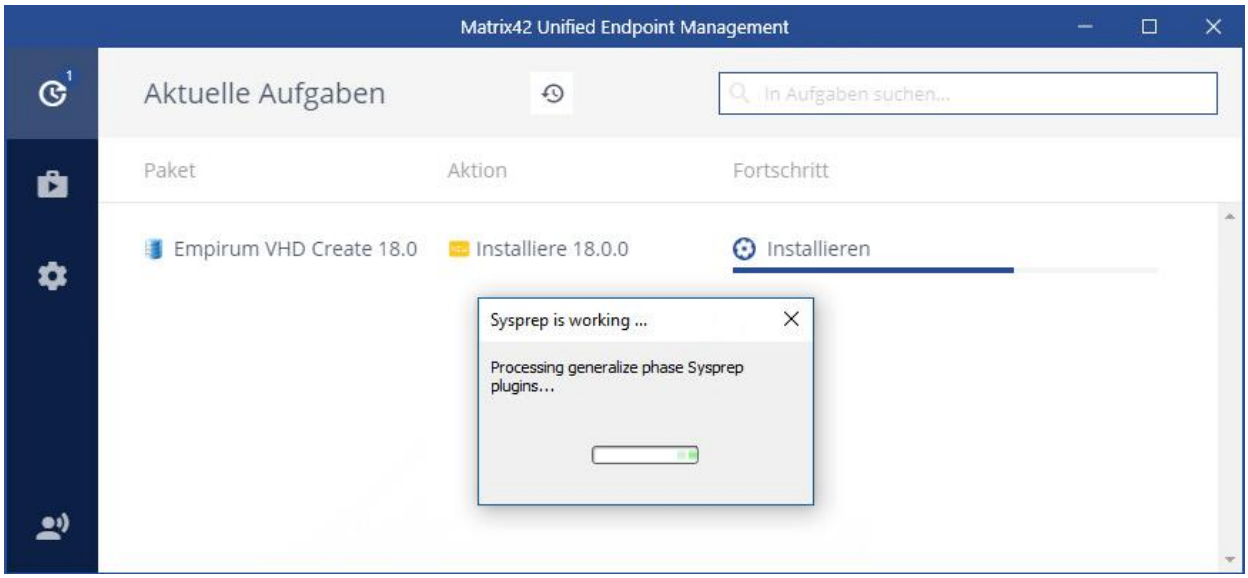
2 Tragen Sie auf *Gruppen-* oder *Computerebene* die von Ihnen bevorzugten *Werte* in die folgenden *Variablen* ein (das Beispiel enthält Standard- oder Demowerte).

- M42_INTERNAL_OS_IMAGING_CREATION.USERNAME = LAB\Test
- M42_INTERNAL_OS_IMAGING_CREATION.PASSWORD = Mx42
- M42_INTERNAL_OS_IMAGING_CREATION.SHARE = \\10.0.55.2\EmpInst$
- M42_INTERNAL_OS_IMAGING_CREATION.SOURCE = disk:0
- M42_INTERNAL_OS_IMAGING_CREATION.DESTINATION = \\10.0.55.2\EmpInst$\DiskImages\Win_10_1809-1902_PRO_x64_DE_WinPE.vhd
- M42_INTERNAL_OS_IMAGING_CREATION.DOSYSPREP = Yes

Hinweis: Das Software Paket "Empirum VHD Create" wurde per Verteilungsoptionen auf Deinstallieren gesetzt, um zu verhindern, dass das Paket ungewollt direkt installiert wird.

3 Aktivieren Sie für die *Gruppe* oder den *Computer* die Optionen "*PULL via DDS/DDC (nur Softwarepakete)*" und "*PXE aktivieren (Computer neu installieren)*", wenn alle Einstellungen auf dem Client vorgenommen und alle Variablen gefüllt wurden.

Wenn die Variable *M42_INTERNAL_OS_IMAGING_CREATION.DOSYSPREP = Yes* gesetzt ist, das Software Paket "*Empirum VHD Create <Version>*" per Verteilungsoptionen auf *Installieren*, *Erneuern* geändert wurde, dann wird das Paket jetzt ausgeführt. Dadurch wird der aktivierte Computer per Sysprep vorbereitet - es erfolgt kein Reboot.



Hinweis: Über das Software Paket "Empirum VHD Create" wird aktuell kein automatischer reboot initiiert. Wenn direkt nach Ausführen von Sysprep ein automatischer Reboot erfolgen soll, können Sie in der Setup.inf des Software Paketes "Empirum VHD Create" (liegt unter: `.\Empirum\Configurator\Packages\Matrix42\Empirum VHD Create\<Version>\Install)` den Aufruf von Sysprep.exe von "quit" auf "reboot" ändern.

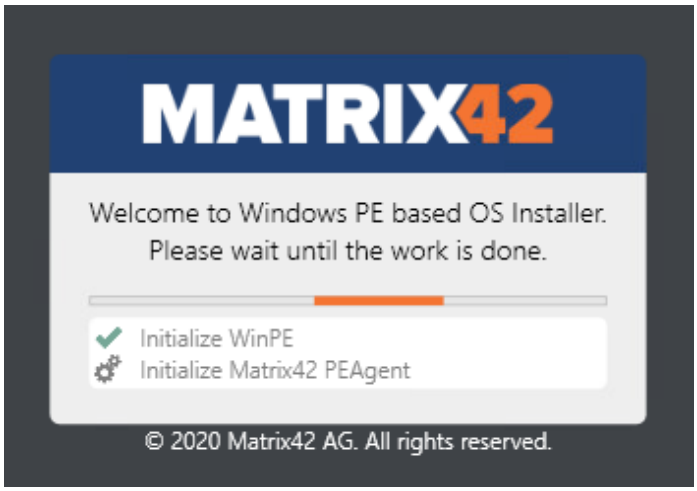
```
#!Log
Set LOGMSG=Call "%windir%\System32\sysprep\sysprep.exe" /generalize /oobe /reboot
#!Log
Call "%windir%\System32\sysprep\sysprep.exe" /generalize /oobe /reboot
```

4 Starten Sie den Computer neu.

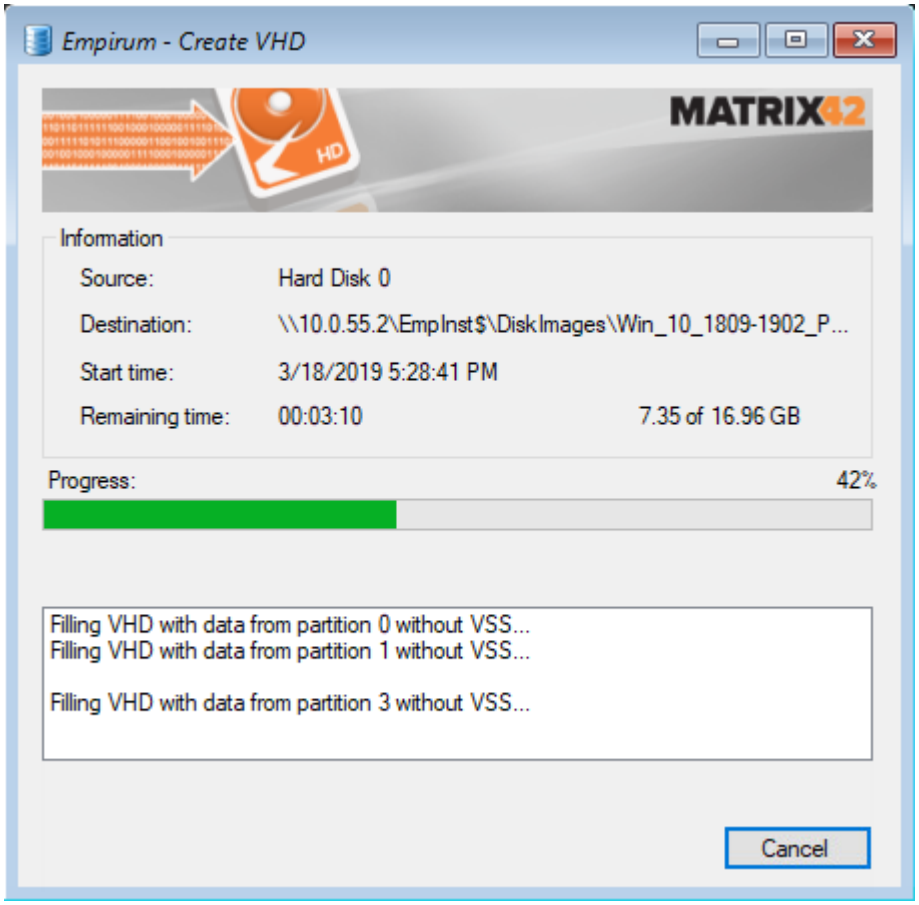
5 Die Vorgehensweise nach dem Reboot entspricht der Disk Imaging - VHD Offline Beschreibung ab *Punkt 5* in der *Online Dokumentation*.

Im Unterschied zum EPE4 muss hier kein WinPE ISO und daraus eine Boot-CD / DVD, oder ein bootbarer USB-Stick erstellt werden.

6 Nach dem Reboot startet über das PreOS Package *DiskImageCreate* das *WinPE* und bereitet den Client vor.



7 Nachdem der Client vorbereitet ist, startet die *Erstellung des Images* über das Paket *Empirum - Create VHD*.



Wenn dieser Prozess abgeschlossen ist, liegt in dem von Ihnen angegebenen Verzeichnis das Image - in diesem Beispiel auf dem angegebenen Empirum Server unter:

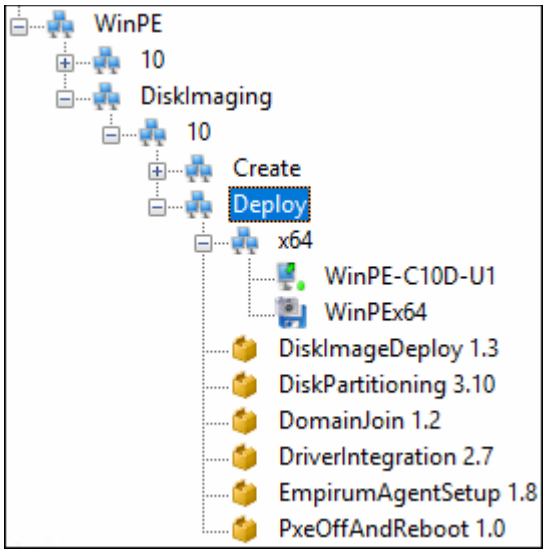
```
.\EmpInst$\DiskImages\Win_10_1809-1902_PRO_x64_DE_WinPE.vhd
```

Dieses Image kann jetzt auf andere Clients verteilt werden - siehe nächstes Kapitel.

6.2 DiskImageDeploy

Nachdem Sie ein Disk Image - wie im vorherigen Kapitel beschrieben - abgezogen haben, können Sie dieses wie im Folgenden beschrieben auf andere Computer verteilen.

1 Erstellen Sie sich eine beliebige *Konfigurationsgruppe* nach folgendem Beispiel:



2 Tragen Sie auf *Gruppen* oder *Computerebene* die von Ihnen bevorzugten *Werte* in die folgenden Variablen ein (das Beispiel enthält Standard- oder Demowerte).

- **Variablen mit Standardwerten - keine Änderungen notwendig:**
  - M42_INTERNAL_OS_IMAGING_SETTINGS.ADDRESS = 239.42.42.42
  - M42_INTERNAL_OS_IMAGING_SETTINGS.PORT = 9000
  - M42_INTERNAL_OS_IMAGING_SETTINGS.CLIENTTHRESHOLD = 0
  - M42_INTERNAL_OS_IMAGING_SETTINGS.STARTTIMEOUT = 300
  - M42_INTERNAL_OS_IMAGING_SETTINGS.JOINTIMEOUT = 180
  - M42_INTERNAL_OS_IMAGING_SETTINGS.TTL = 2
  - M42_INTERNAL_OS_IMAGING_SETTINGS.CLIENTTIMEOUT = 120
  - M42_INTERNAL_OS_IMAGING_SETTINGS.PACKAGESIZE = 65200
  - M42_INTERNAL_OS_IMAGING_SETTINGS.PAGESIZE = 1024
- **Variablen mit Demowerten - Änderungen notwendig:**
  - DiskImageDiploy.LocalUserName = Admin
  - DiskImageDiploy.LocalUserPassword = Mx42
  - DiskImageDiploy.LocalUserDisplayName = Admin
  - DiskImageDiploy.SetupUILanguage = en-US
  - DiskImageDiploy.InputLocale = de-DE
  - DiskImageDiploy.SystemLocale = en-US
  - DiskImageDiploy.UILanguage = de-DE
  - DiskImageDiploy.UserLocale = de-DE
  - M42_INTERNAL_OS_IMAGING_SETTINGS.IMAGEFILE_MBR = Win_10_1809-1903_ENT_x86_EN_MBR_WinPE.vhd

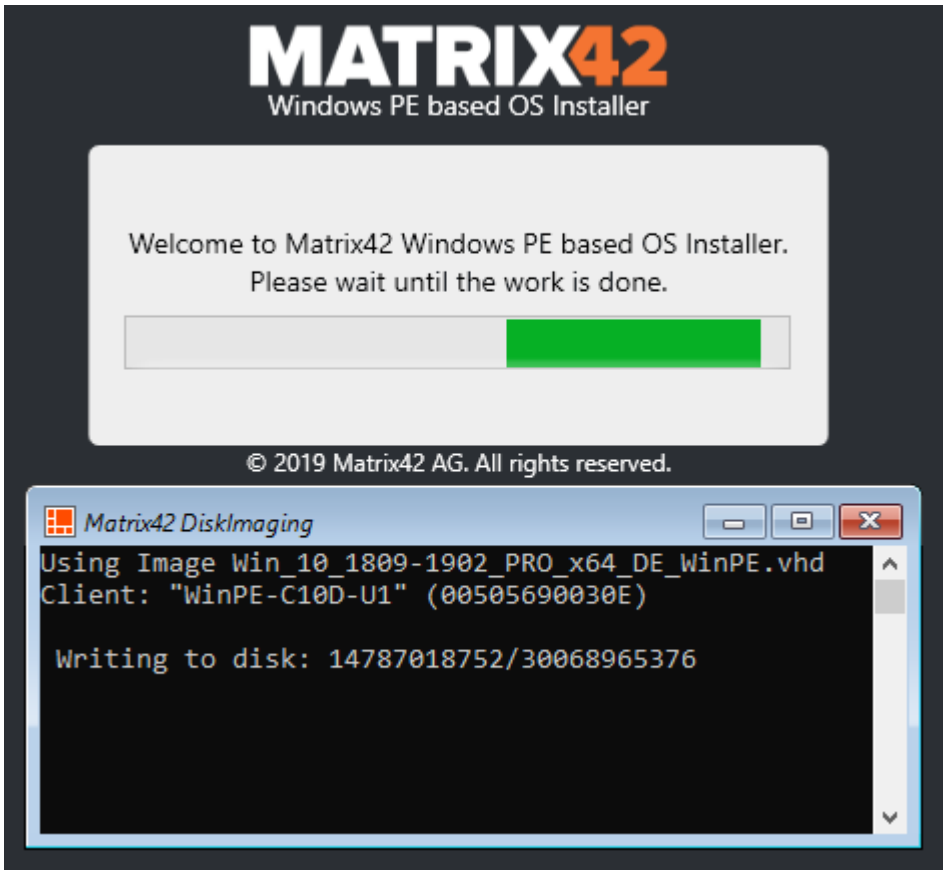
- M42_INTERNAL_OS_IMAGING_SETTINGS.IMAGEFILE_GPT = Win_10_1809-1902_PRO_x64_DE_GPT_WinPE.vhd
- M42_INTERNAL_OS_IMAGING_SETTINGS.DISKIMAGING_PASSWORD = Mx42
- DiskPartitioning.SizeSystemPartition = 60
- DiskPartitioning.SizeDataPartition = 40
- DiskPartitioning.InterpretSizeInputAsPercentage = 1
- DiskPartitioning.MinimumSystemPartitionSizeInGB = 28
- DiskPartitioning.SizeWinREPartitionInMB = 350
  Wenn der Client per **WinPE** installiert wurde, muss hier die Größe beim Abziehen eingetragen werden - in diesem Beispiel ist das **350**.
  Wurde der Client per **EPE4** installiert, muss hier **0** eingetragen werden.
- DiskPartitioning.PreferFastDisk = 0
  Wenn auf eine NVME oder SSD HD installiert werden soll, dann 1 eintragen.
- DiskPartitioning.ClearAllDisks = 0
  Wenn alle Festplatten zuerst komplett gelöscht werden sollen, dann hier 1 eintragen.
- DomainJoin.DomainJoinCredentialsUser = LAB\Test
- DomainJoin.DomainJoinCredentialsPassword = Mx42
- DriverIntegration.DriverArchiveFileName = VMware.zip
- MX42_AGENT_PUSH_PACKAGE_FOLDER.Windows = UEM Agent Windows\1903.0

3 Aktivieren Sie für die *Gruppe* oder den *Computer* die Optionen "*PULL via DDS/DDC (nur Softwarepakete)*" und "*PXE aktivieren (Computer neu installieren)*" und *starten* Sie anschließend den *zugeordneten Computer*.

Das WinPE startet, die Festplatte wird über das PreOS Package *DiskPartitioning* vorbereitet, anschließend startet das PreOS Package *DiskImageDeploy*.

In dieser Abbildung unter Verwendung des ***WADK 10 (1803)***.

In dieser Abbildung unter Verwendung des ***WADK 10 (1809)***.



Nachdem das Disk Image übertragen wurde, erfolgt ein Reboot.

4 Anschließend werden die PreOS Pakete *DriverIntegration, DomainJoin* und *EmpirumAgentSetup* ausgeführt.

Wurden diese Pakete erfolgreich ausgeführt, ist der Computer zur weiteren Verwendung einsatzfähig.

7 WinPE-basiertes Computer Self Provisioning

Das WinPE-basierte OS-Deployment unterstützt ab Matrix42 Client Management v19.0.1 das *Computer Self Provisioning*, mit dessen Hilfe Computer vollständig automatisiert mit Betriebssystem und Software durch Mitarbeiter ohne Empirum-Kenntnisse aufgesetzt werden können.

Im Vergleich zum Empirum-PE-basiertem Computer Self Provisioning (EPE 4) wurden zwei konzeptuelle Änderungen vorgenommen. Zum einen wurde das Konzept der Computervorlagen eingeführt, das im Folgenden genauer erklärt wird. Außerdem wird zur Kommunikation mit dem Empirum-Server während der Computer Self Provisioning-Ausführung die Empirum-API verwendet. Somit muss auch auf dem Empirum Server der Empirum-API-Dienst installiert und konfiguriert sein.

Achtung:
WinPE-basiertes Self Provisioning ist im aktuellen Zustand nur für Installation-Straßen vorgesehen und sollte nicht in produktiven Umgebungen eingesetzt werden, da dort die Arbeitsgeräte nicht lokal booten werden können, solange diese nicht erst einmal mit einer Empirum-Version ab 19.0.1 und WinPE Support-Version ab 1.6.1 aufgesetzt wurden.

Das WinPE-basierte Self Provisioning kann auf allen (EFI) Computern verwendet werden, der lokale Boot (um Self Provisioning überspringen zu können) funktioniert jedoch nur auf Computern, die zuvor bereits mit Empirum 19.0.1 / WinPE Support 1.6.1 aufgesetzt wurden. Alternativ kann man das Self Provisioning auch über einen USB-Stick durchführen, den man sich unter *Boot Konfigurationen* in Empirum erstellen kann. Beim Ausführen einer Self Provisioning-Boot-Konfiguration werden im aktuellen Stand zur Konfiguration hinzugefügte Treiber nicht installiert.

Das WinPE-basierte Self Provisioning funktioniert nur mit EFI Computern - BIOS Computer werden nicht unterstützt. Und daher ist es wichtig darauf zu achten, dass die für Self Provisioning vorgesehene WinPE-basierte Boot-Konfiguration **keine BIOS**-Option enthält.

Ein Deployment über die Empirum Subdepots ist zurzeit nur möglich, wenn der zu bespielende Computer eine Verbindung direkt zum Empirum-API-Dienst des Empirum Master Servers aufbauen kann.

Das WinPE-basierte Self Provisioning setzt mindestens die Empirum Version 19.0.1 (und WinPE Support 1.6.1) voraus, in der es möglich ist, die neuen Computervorlagen zu erstellen. In älteren Versionen von Empirum sind die Computervorlagen nicht vorhanden.

Wird ein USB Stick auf einem *Server 2012 R2* oder *Server 2016* bzw. *Windows Server vor Version 1703* erstellt, können offline Quellen nicht hinzugefügt werden, da diese Windows Server Versionen keine zweite NTFS Partition erstellen können. Unterstützt wird diese Funktion erst ab Server 2019. Alternativ kann auch eine aktuelle Windows 10 Version mit installiertem ADK 10 (1903) und installierter Matrix42 Management Console verwendet werden.

Achtung: Wenn Sie ein WinPE PXE-Image als Default-Image definieren, sind über diesen PXE-Server keine EPE-basierten OS-Installationen mehr möglich. Ein Mischbetrieb von WinPE-PXE Default/SP-Images und EPE-PXE OS-Installationen wird nicht unterstützt.

Hinweis: Ab der WinPE PreBoot Support Version 1.8.3 ist es möglich, Self Provisioning auch per http(s) durchzuführen.

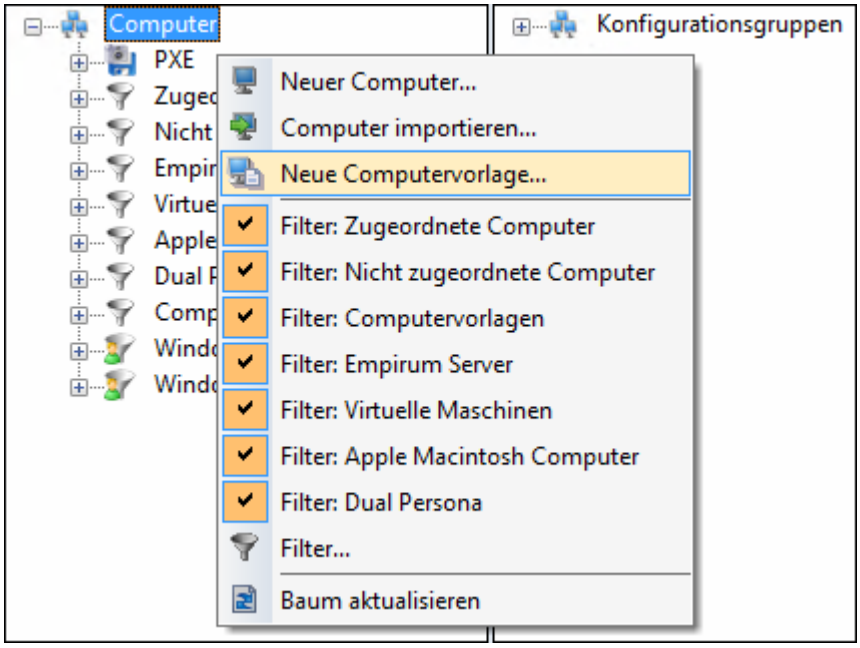
7.1 Computervorlagen

Mit dem WinPE-basierten Computer Self Provisioning wird das Konzept der *Computervorlagen* eingeführt. Dieses löst das Konzept der *Self Provisioning Gruppen* ab. Computervorlagen sind Platzhalter in Empirum und können wie normale Computer-Objekte behandelt werden. Sie können wie normale Computer den Konfigurations- und Zuweisungsgruppen zugewiesen werden. Somit lässt sich der Zielzustand der neuen Computer festlegen.

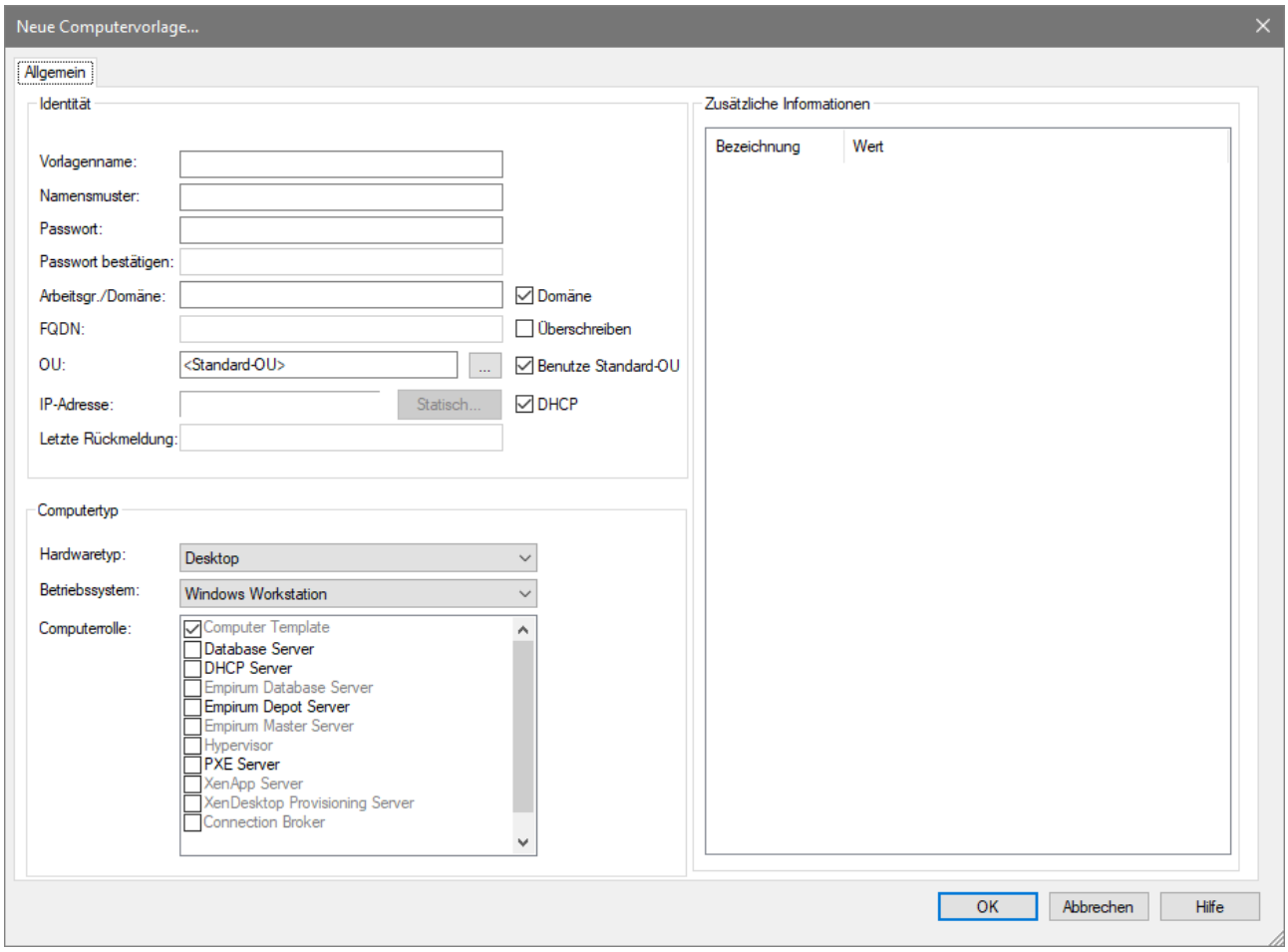
Während der Durchführung eines Computer Self Provisioning wird eine Computervorlage ausgewählt, die als Vorlage für den neu-zu-erstellenden Computer dient. Dieser wird den Konfigurations- und Zuweisungsgruppen zugewiesen, denen auch die Computervorlage zugewiesen ist. Der neue Computer wird aktiviert und das Deployment wird im Anschluss durchgeführt.

7.1.1 Erstellen von Computervorlagen

Computervorlagen können in der *Matrix42 Management Console* unter *Administration* über das *Kontextmenü* im linken Baum erstellt werden. Öffnen Sie hierzu beispielsweise das Kontextmenü im linken Baum am obersten Eintrag *Computer*.



Wird der Kontextmenüeintrag *Neue Computervorlage* ausgewählt, öffnet sich der Dialog, mit dessen Hilfe eine neue Computervorlage erstellt werden kann.



Der *Neue Computervorlage*-Dialog entspricht dem Dialog zum Erstellen eines neuen Computers.

Wird bei *Namensmuster* ein Name vorgegeben, wird dieser Wert im Selfprovisioning-Prozess genommen und ***der Benutzer wird nicht mehr nach dem Namen gefragt***. Der Wert wird automatisch als Computername für den zu erstellenden Client verwendet. Die EmpirumAPI wird gegebenenfalls ein "*_1, _2, ...*" anhängen, falls es bereits einen entsprechenden Client gibt. Es sind nur maximal 12 der Zeichen ***A-Z, a-z, 0-9, -,*** _ erlaubt.

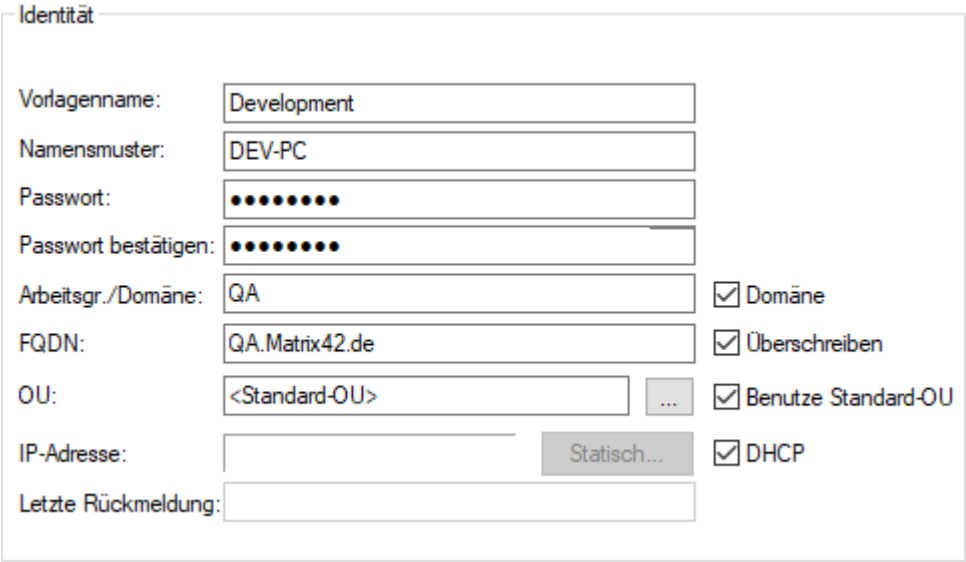
Ist der Wert bei *Namensmuster* leer, wird auf jeden Fall der Benutzer (während des Selfprovisioning Prozesses) nach dem Computernamen gefragt. Maximal 12 Zeichen (plus der 3 Zeichen, die angehängt werden, wenn es den Computernamen bereits gibt!).

Besonderheit hier ist, dass die Eigenschaften UUID und MAC-Adresse bei Computervorlagen nicht angegeben werden. Diese Angaben können erst zur Laufzeit des Computer Self Provisioning vom jeweiligen Computer festgelegt werden. Diese Eigenschaften sind deswegen ausgegraut.

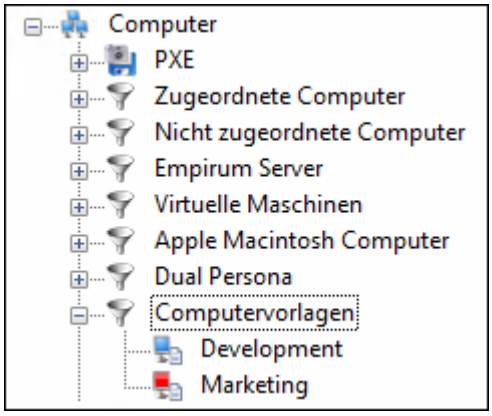
Wird unter *Passwort* und *Passwort bestätigen* ein Passwort eingetragen, muss direkt nach der Anzeige/Auswahl des Computer-SP-Templates das hier hinterlegte Passwort angegeben werden, um mit der Installation zu beginnen. Ist kein Passwort eingetragen, erfolgt auch keine Abfrage.

Die folgenden Angaben sind bei der Erstellung einer Computervorlage notwendig:

- Computername
- Arbeitsgruppe/Domäne



Wurden die Angaben gemacht, kann die neue Computervorlagen mit *OK* erstellt werden und erscheint in der Liste unter dem Filter *Computervorlagen* im linken Baum (außerdem wird die Computervorlage auch in der Liste der nicht zugeordneten Computer angezeigt).



Der Filter *Computervorlagen (Computer Templates)* listet alle Computer mit der zugeordneten Rolle *Computer Template* auf.

7.1.2 Konfigurieren und zuweisen von Computervorlagen

Ist die Computervorlage erstellt, kann diese einer bestehenden Konfigurationsgruppe zugewiesen werden, die bereits für das WinPE-basierte OS-Deployment vorbereitet wurde. Wie in *Kapitel 2.3 Zuweisung in der Administration* ab Seite 19 beschrieben, muss die Konfigurationsgruppe vorbereitet sein. Es müssen somit folgende Einstellungen vorgenommen werden:

- **Zuweisung eines WinPE-basierten PXE-Image**
- Zuweisung der notwendigen PreOS-Pakete (DiskPartitioning, WindowsInstallation, PxeOffAndReboot, DomainJoin, EmpirumAgentSetup, …)
- Konfiguration von Deployment-relevanten Computervariablen
- Zuweisung der Betriebssystemedition
- Zuweisung von Software-Paketen, die nach der Betriebssysteminstallation installiert werden sollen

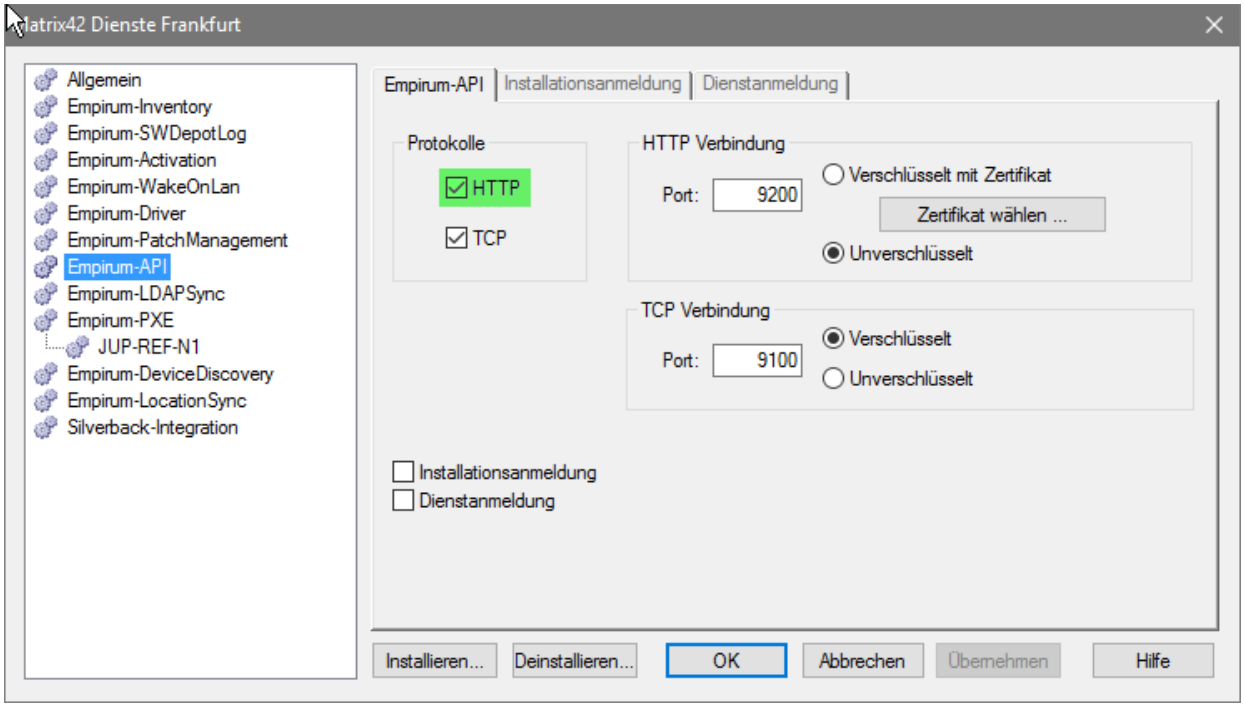
Hinweis: Zur Konfiguration von Computervorlagen können Zuweisungsgruppen ebenfalls verwendet werden. Computervorlagen können somit auch mehreren Zuweisungsgruppen zugewiesen werden. Bei der Durchführung eines Computer Self Provisioning wird der neu erstellte Client allen Gruppen zugewiesen, denen auch die Computervorlage zugewiesen ist.

7.2 Empirum-API Dienst konfigurieren und installieren

Wie bereits beschrieben findet die Kommunikation beim Computer Self Provisioning mit dem Empirum Server über die Empirum-API statt. Hierfür muss über Matrix42 DBUtil der Empirum-API-Dienst auf dem Empirum Master Server konfiguriert und installiert werden.

Hinweis: Ausführlichere Informationen zur Dienste Installation über Matrix42 DBUtil finden Sie in der Matrix42 Online-Hilfe.

1 Starten Sie hierfür *Matrix42 DBUtil*.
2 Melden Sie sich mit dem entsprechenden *Benutzer* an.
3 Wählen Sie den *Standort* aus.
4 Öffnen Sie die Dienste Konfiguration über Menü > *Aktionen* > *Dienste installieren/konfigurieren*.
5 Wählen Sie das Protokoll *HTTP* aus und konfigurieren Sie den *Port* für die HTTP Verbindung (Standardwert ist 9200).



Achtung: Aktuell wird für das WinPE-basierte Computer Self Provisioning vorausgesetzt, dass der Empirum-API-Dienst mit dem Protokoll HTTP konfiguriert ist. Dieser kann entweder unverschlüsselt (HTTP) oder mit einem Zertifikat verschlüsselt (HTTPS) konfiguriert sein.

6 Übernehmen Sie die Änderungen mit *Übernehmen*.

7 Installieren Sie den *Empirum-API Dienst neu* über das *Kontextmenü* des Empirum-API Eintrages in der Liste.

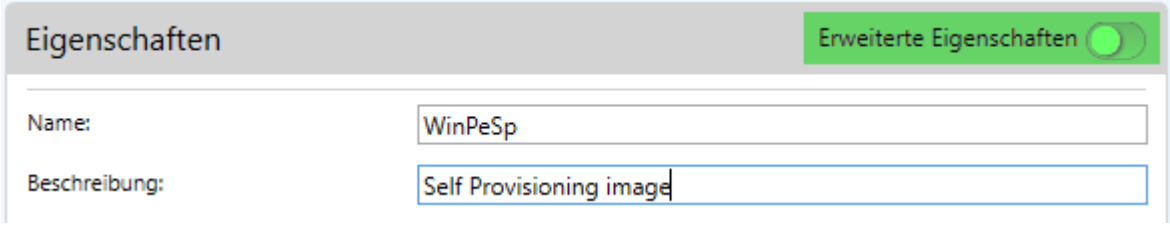
8 Nach der erfolgreichen Installation des Dienstes kann Matrix42 DBUtil wieder geschlossen werden.

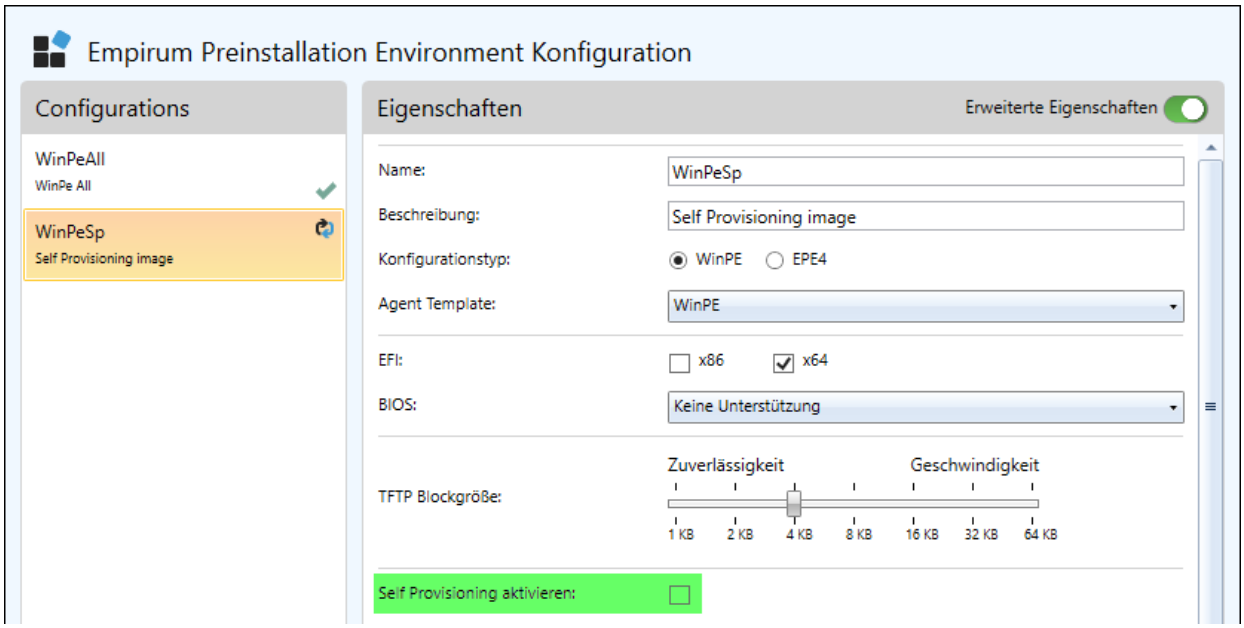
Der Empirum-API Dienst ist somit installiert und steht für das WinPE basierte Computer Self Provisioning zur Verfügung. Als nächstes kann nun mit der Erstellung einer Bootkonfiguration für das Computer Self Provisioning fortgefahren werden.

7.3 Bootkonfigurationen beim Computer Self Provisioning

Damit beim Bootvorgang das Computer Self Provisioning durchgeführt werden kann, muss eine Bootkonfiguration erstellt werden, die diese Funktionalität enthält.

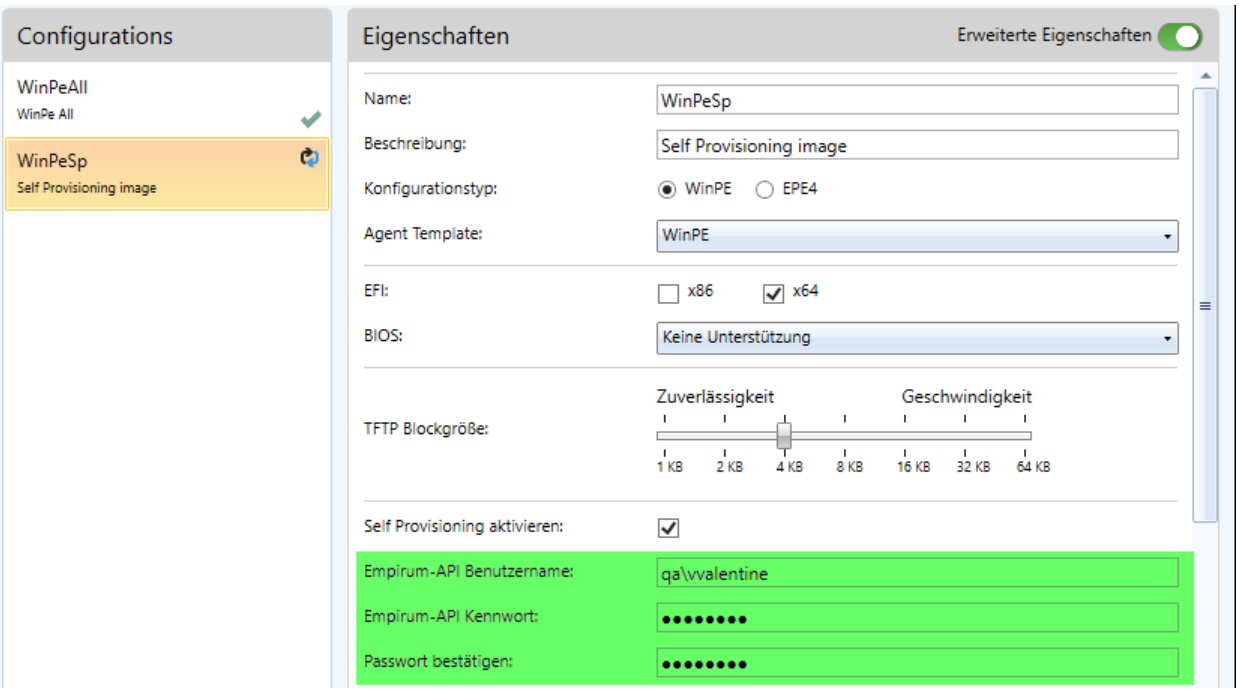
1 In einer WinPE-basierten Bootkonfiguration ist dafür die *Self Provisioning aktivieren* Eigenschaft enthalten, welche ausgewählt werden muss. Um alle Eigenschaften zu sehen muss zuerst der Schalter *Erweiterte Eigenschaften* aktiviert werden.





2 Nach dem Auswählen der Self Provisioning Eigenschaft erscheinen zusätzlich die Eigenschaften für die Anmeldeinformationen. Diese müssen ebenfalls angegeben werden.

> **Hinweis:** Verwenden Sie einen Benutzer der auf der Datenbank ausreichenden Rechte besitzt. Der Benutzer muss mindestens die Rollen *EMP_M_COMPUTER* und *EMP_M_COMP_ROLE* besitzen, die dem Benutzer über Matrix42 DBUtil zugeordnet werden können. SQL Server- oder Windows-Benutzer sind möglich.



3 Die Bootkonfiguration muss im Anschluss *abgespeichert* werden, damit die Änderungen am PXE-Boot-Image umgesetzt werden.

Wenn die PXE-Boot-Image-Erstellung dann erfolgreich durchgeführt wurde, kann mit der Aktivierung des Computer Self Provisioning auf dem Empirum Server fortgefahren werden.

7.3.1 Zusätzliche Empirum-API Verbindungsdaten

Um diese Kommunikation zur WinPE-Laufzeit aufbauen zu können werden beim Bauen des PXE-Boot-Images weitere Verbindungsdaten aus der Datenbank entnommen und im PXE-Boot-Image abgelegt. Die folgenden Angaben werden übernommen:

- Empirum-API Dienst Servername
- Empirum-API Dienst Angaben zum HTTP Protokoll:
  - Port
  - Verschlüsselung

Hinweis: Falls der Empirum-API Dienst noch nicht über Matrix42 DBUtil konfiguriert und installiert wurde, wird es beim Bauen des Self Provisioning aktivierten PXE-Boot-Image zu einem Fehler kommen, da die Verbindungsangaben zum Empirum-API Dienst notwendig sind.

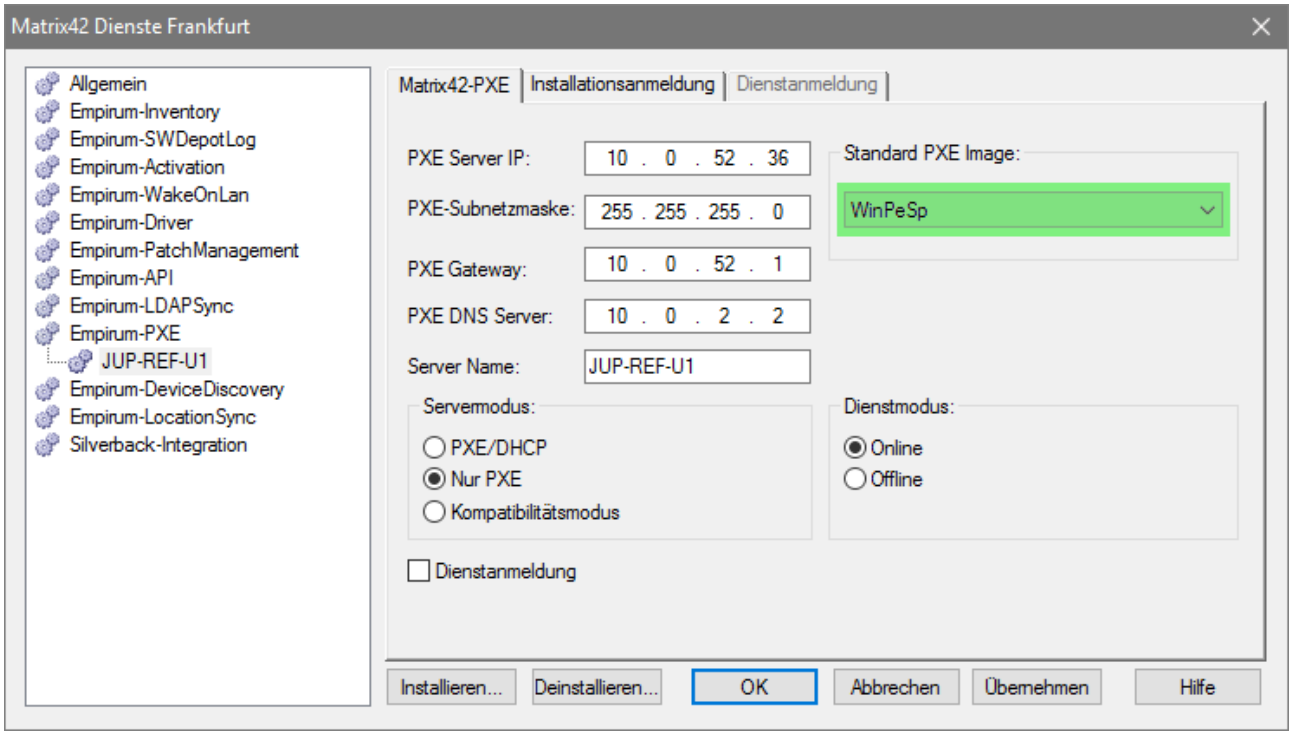
Achtung: Aktuell wird nur das HTTP-Protokoll unterstützt, das entweder unverschlüsselt (HTTP) oder mit einem Zertifikat verschlüsselt (HTTPS) konfiguriert sein kann.

7.4 Computer Self Provisioning auf dem Empirum Server aktivieren

Achtung: WinPE-basiertes Self-Provisioning ist im aktuellen Zustand nur für Installation-Straßen vorgesehen und sollte nicht in produktiven Umgebungen eingesetzt werden, da dort die Arbeitsgeräte nicht lokal booten werden können, solange diese nicht vorher einmal mit der WinPE Support-Version (1.6.3) oder EPE Support-Version (4.7.11) aufgesetzt wurden.

Mit Hilfe der *Offline Boot Medium*-Erstellung kann unter *Boot Konfiguration* in der Empirum Konsole ein USB-Stick erstellt werden, mit dessen Hilfe das Computer Self Provisioning direkt am neuen Computer über das Booten des USB-Sticks ausgeführt werden kann.

Soll das Computer Self Provisioning über einen PXE Boot ermöglicht werden, dann muss in Matrix42 DBUtil das Standard PXE Image auf dem PXE Server-Eintrag des Empirum Master Servers auf das erzeugte WinPE basierte Self Provisioning Image gesetzt werden. Im Anschluss muss die Änderung mit *Übernehmen* abgespeichert und der PXE-Dienst neu installiert werden.



7.4.1 Registry-Anpassungen für den Empirum PXE-Dienst

Wie beim EmpirumPE 4-basierten Computer Self Provisioning ist auch beim WinPE-basierten Computer Self Provisioning zu beachten, dass der folgende Registry-Wert auf dem Empirum Master Server (auf dem der PXE-Dienst läuft) gesetzt ist:

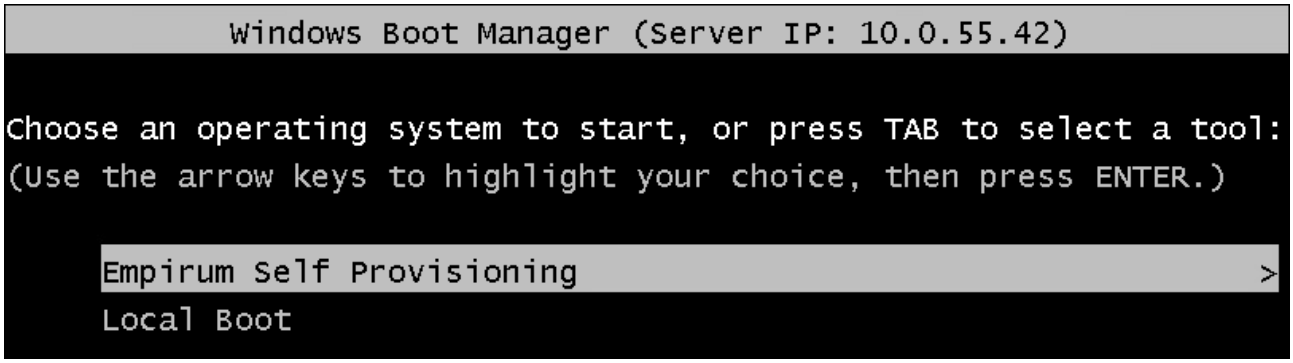
```
HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\matrix42\PXE\EPE4SP
```

muss auf 1 gesetzt werden, bzw. angelegt werden (Typ = DWORD; Wert = 1).

7.5 Computer Self Provisioning am Computer ausführen

Wurden die Vorbereitungen getroffen, kann nun das Computer Self Provisioning mit einem neuen Computer ausgeführt werden. Das Booten kann entweder über den erstellten USB-Stick oder über PXE ausgeführt werden.

Wenn der Computer bootet, wird der Windows Boot Manager angezeigt. Mit standard-mäßig ausgewählten *Local Boot* Eintrag. Möchte man das Self Provisioning ausführen, muss man den Menü-Eintrag Empirum Self Provisioning auswählen.



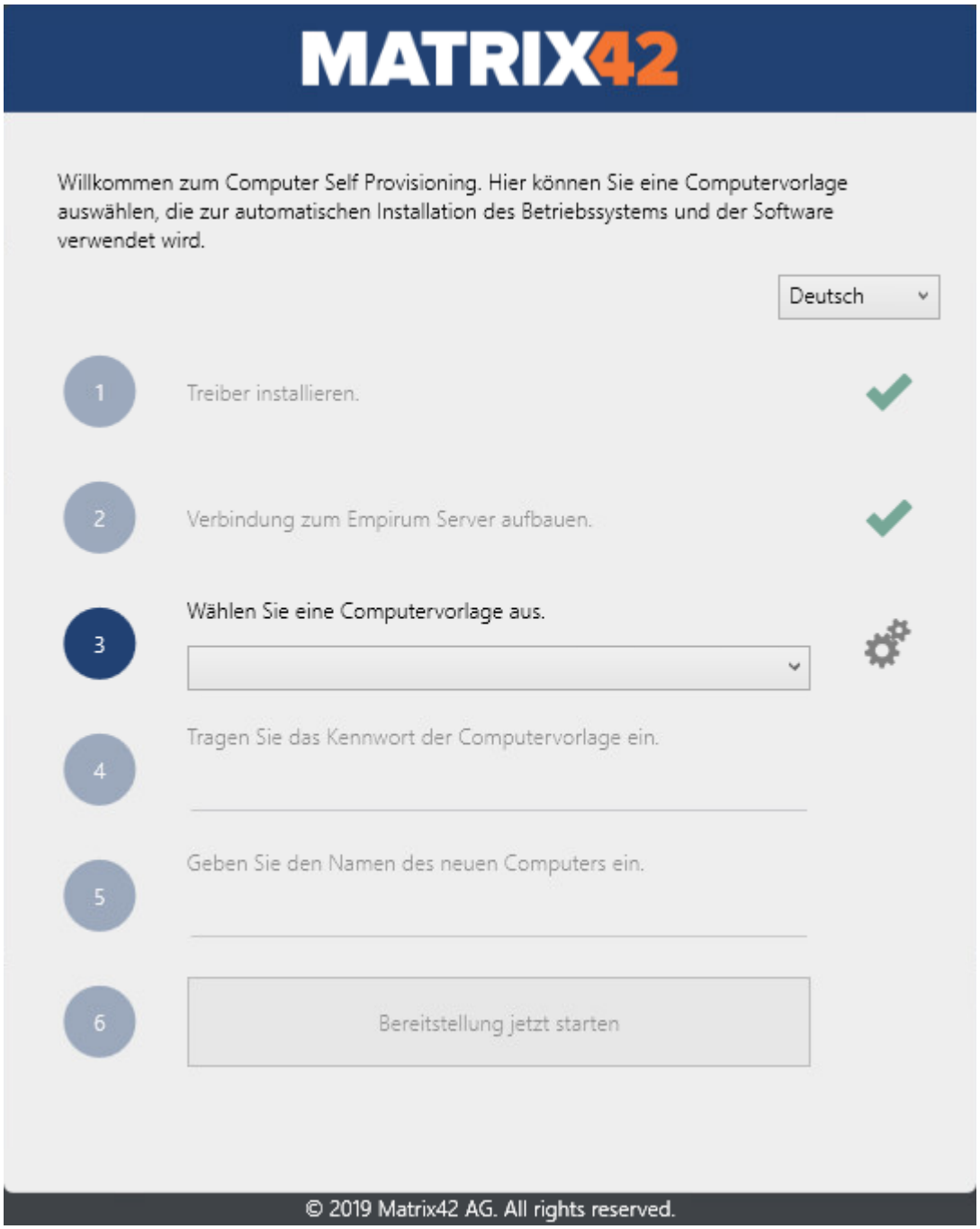
Nach der Auswahl des Empirum Self Provisioning wird automatisch die WinPE-basierte Umgebung gestartet und der Anwender wird durch das Computer Self Provisioning geleitet. Hierzu sind folgende Angaben notwendig:

- Auswahl der Computervorlage
- Angabe des hinterlegten Passwortes (wenn konfiguriert)
- Angabe des neuen Computernamens (wenn kein Namensmuster angegeben wurde)

Nach dem Start der Self Provisioning Benutzeroberfläche, erfolgt die Installation der notwendigen Treiber **(1)**, anschließend wird eine Verbindung zum Empirum Server hergestellt **(2)**. Wenn die Aktion erfolgreich war, wird dies jeweils mit einem grünen Häkchen quittiert.

(3) Computervorlage auswählen

Wählen Sie eine der verfügbaren Computervorlagen aus.

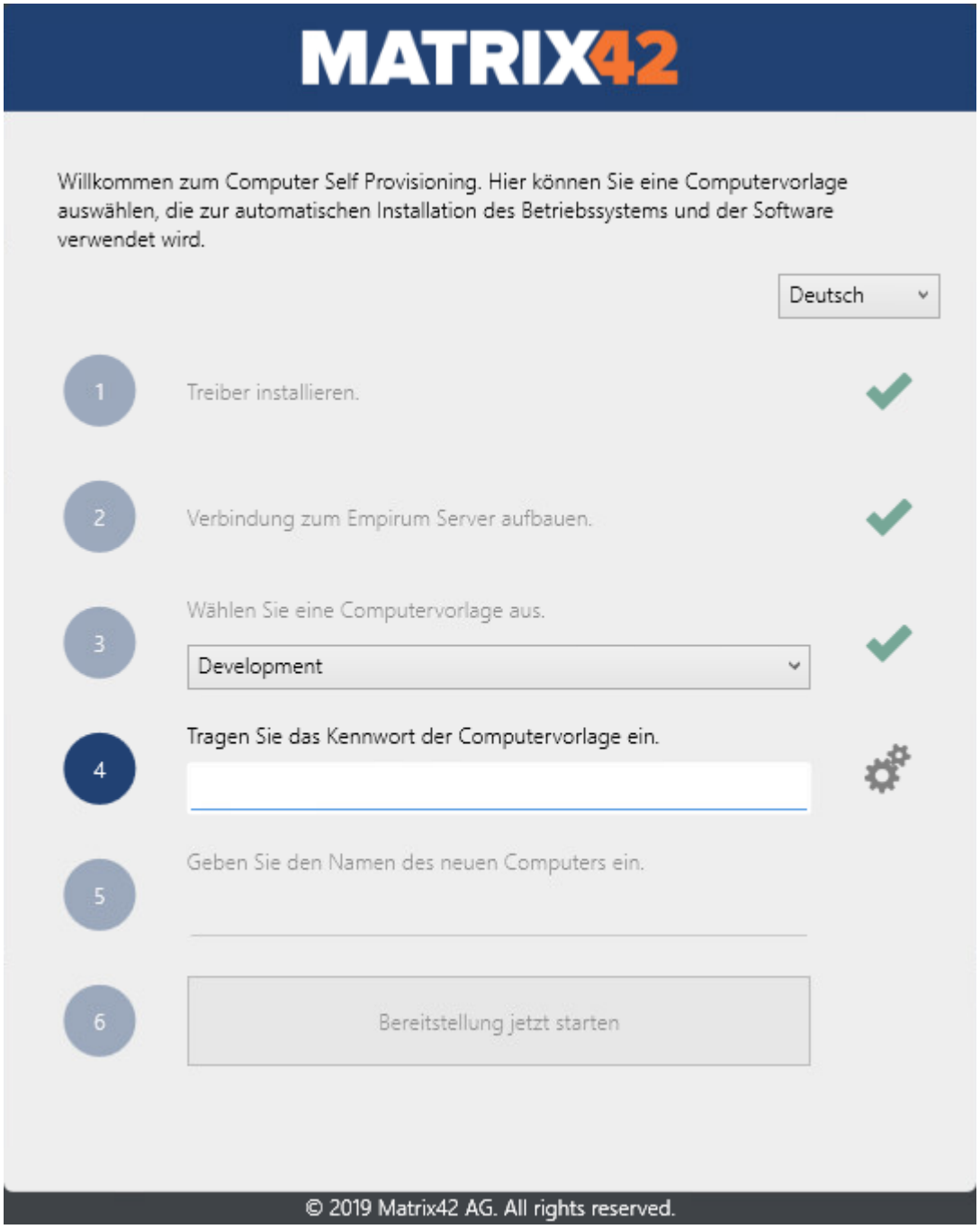


Hinweis: Hier werden ausschließlich zugewiesene Computervorlagen zur Auswahl angezeigt.

Nach Auswahl der Computervorlage gelangen Sie direkt zur Eingabe für das Kennwort der Computervorlage **(4)**. Wurde für die ausgewählte Computervorlage kein Kennwort vergeben, geht es direkt mit **(5)** weiter.

(4) Kennwort der Computervorlage eintragen

Wurde in der Computervorlage ein *Kennwort* eingetragen, muss dieses hier *bestätigt* werden, um den Computer per Self Provisioning neu zu installieren.



Nach Eingabe des Kennwortes betätigen Sie die [ENTER] oder [TAB] Taste.

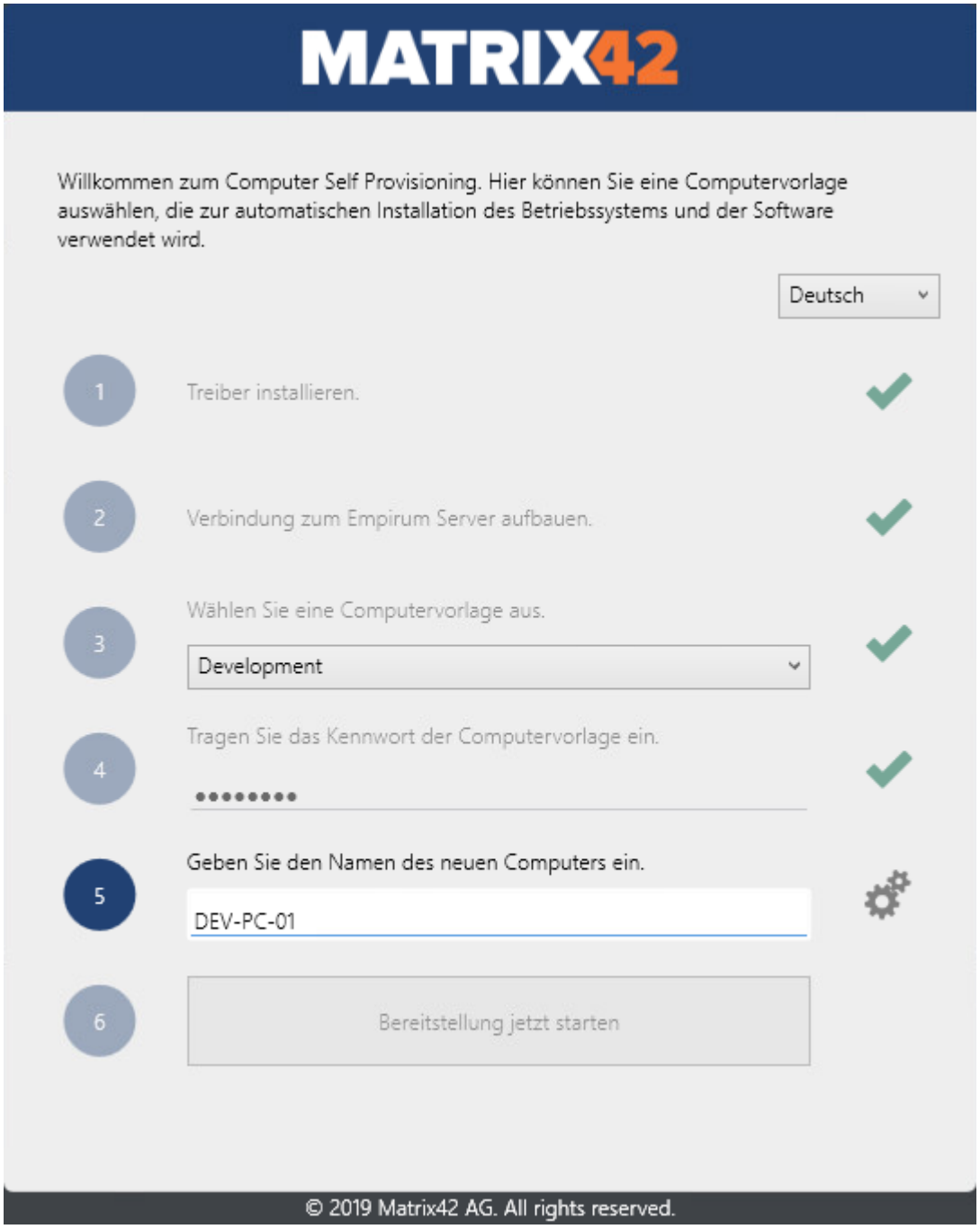
Das Kennwort der Computervorlage wird überprüft. Wurde das Kennwort korrekt eingegeben, geht es mit **(5)** weiter, wenn *kein Namensmuster* definiert wurde.

Wurde ein Namensmuster definiert, geht es direkt mit **(6)** weiter.

> **Hinweis:** Wurde für die ausgewählte Computervorlage ein falsches Kennwort angegeben, können Sie es noch einmal eingeben, oder Sie wählen eine andere Computervorlage **(3)** aus.

(5) Neuen Computernamen eintragen

Wurde in der Computervorlage kein Namensmuster definiert, muss jetzt ein Computername angegeben werden.



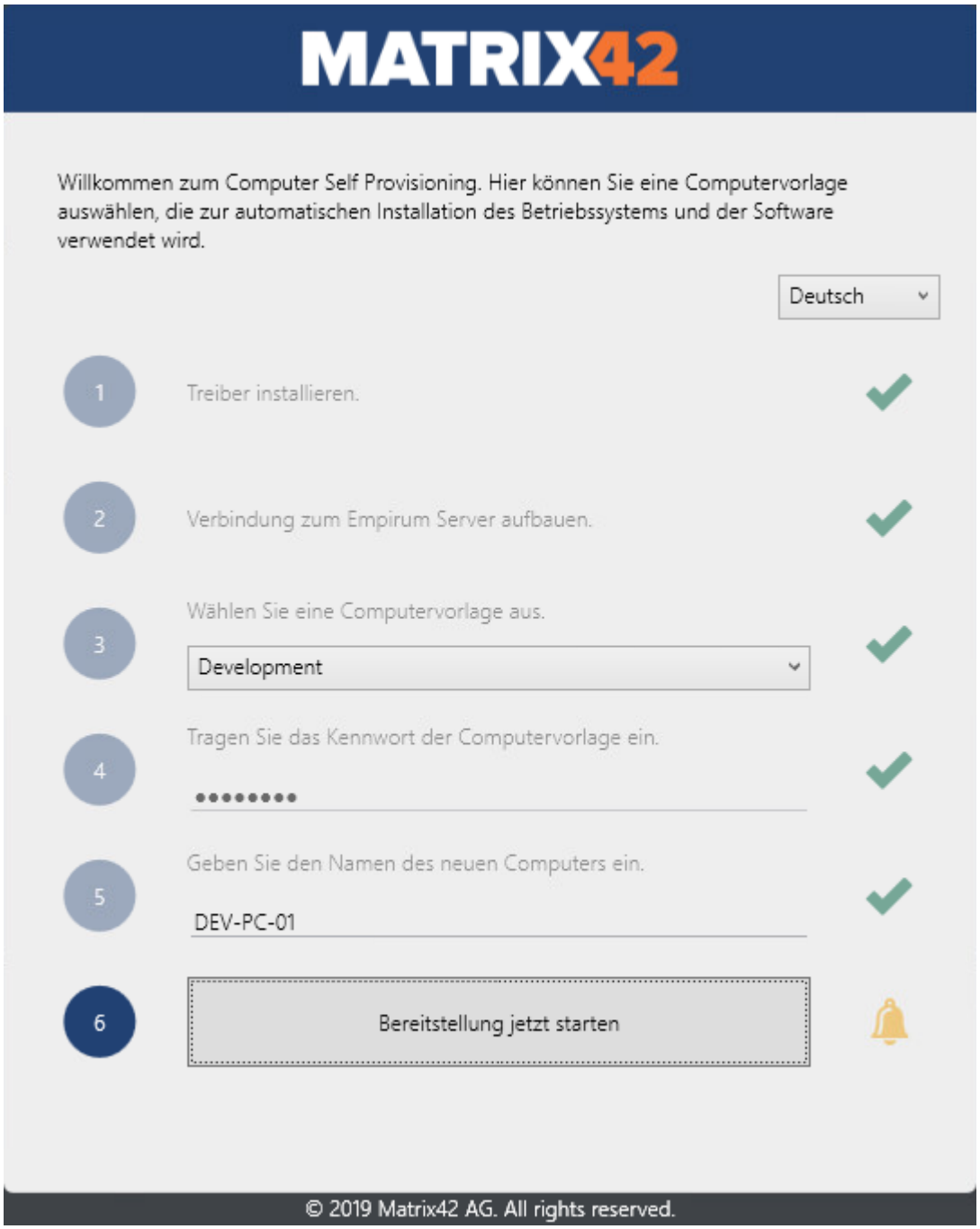
Nach Eingabe des Computernamens betätigen Sie die [ENTER] oder [TAB] Taste, Sie gelangen zum Punkt **(6)**.

Es sind maximal 12 der Zeichen ***A-Z, a-z, 0-9, -,*** _ erlaubt. Bei falsch eingetragenem Computernamen wird der Eingabebereich rot. Wenn Sie mit der Maus über den rot umrandeten Eingabebereich gehen, wird eine entsprechende Fehlermeldung angezeigt.

Wurde eine Computervorlage mit vordefiniertem Namensmuster ausgewählt, ist dieses Namensmuster als Computername bereits eingetragen.

(6) Bereitstellung jetzt starten

Sie haben jetzt die Gelegenheit, alle gemachten Angaben zu überprüfen.



Mit einer beliebigen Taste [ENTER] wird der Vorgang abgeschlossen, so dass das Deployment für den Computer direkt im Anschluss durchgeführt werden kann.

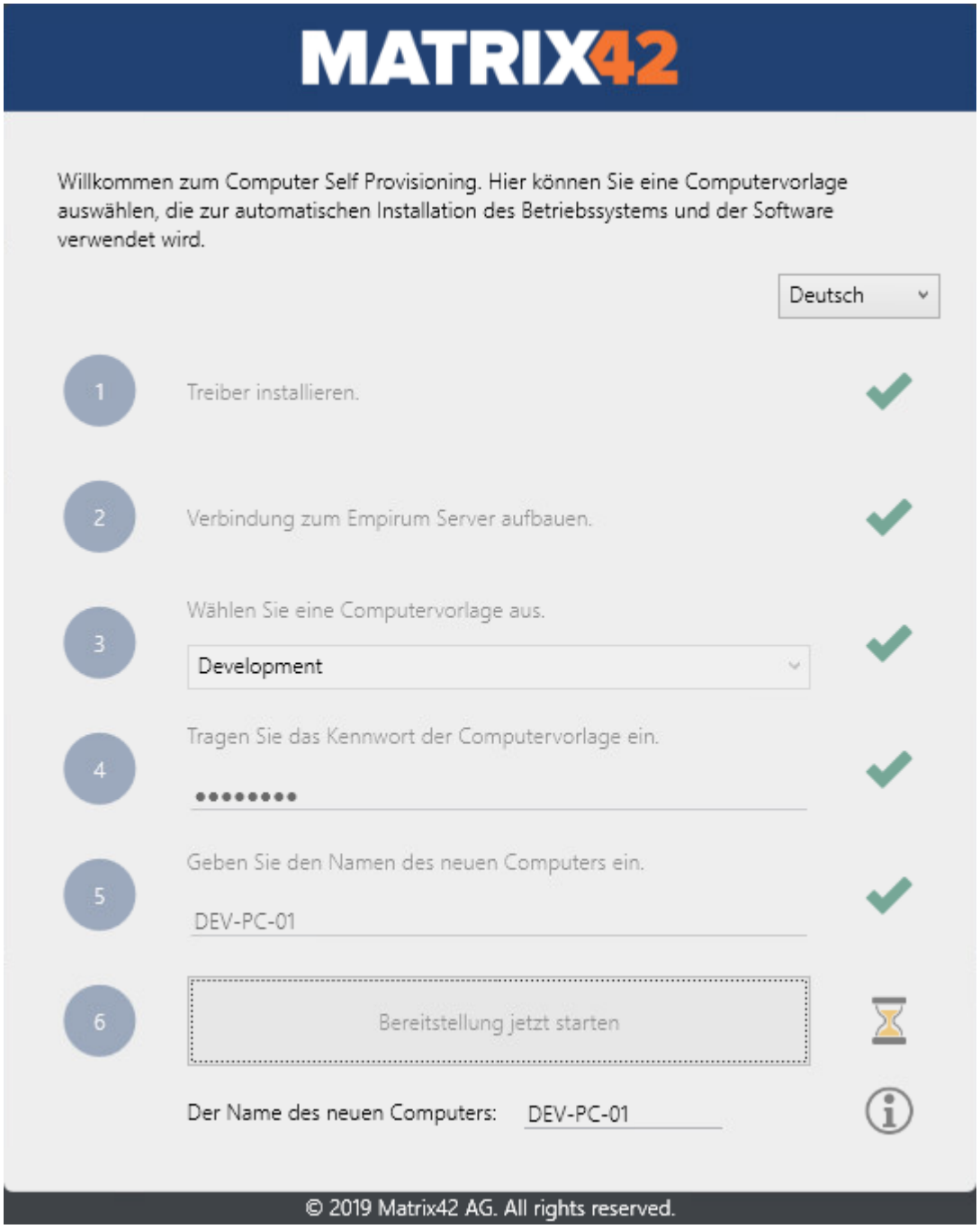
Es erfolgt eine Überprüfung, ob dieser Computername bereits existiert - wenn ja, wird ein "_01", "_02 etc." angehängt und der Computer mit diesem Namen in Empirum registriert.

> **Hinweis:** Aktuell ist dieses Schema der Namenserweiterung ("_01", "_02 etc.") fest vorgegeben und kann nicht geändert werden!

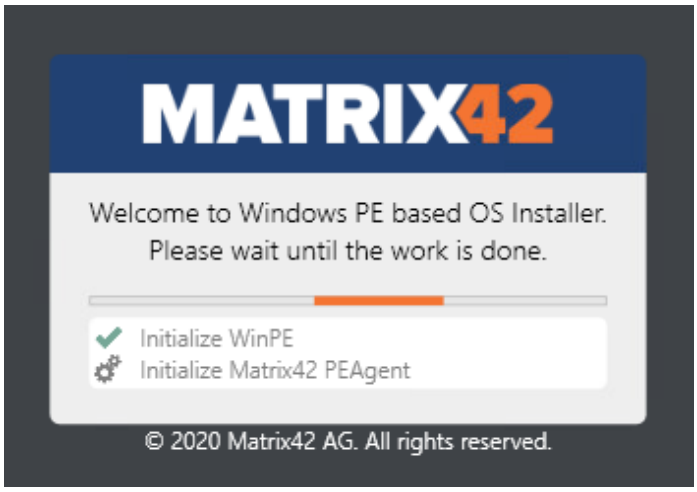
Im unteren Bereich wird jetzt der tatsächliche Computername angezeigt, so wie er in Empirum - in der Gruppe mit der ausgewählten Computervorlage - per API erstellt wurde.

Der neue Computer wird nun mit dem angegeben Namen in Empirum angelegt und entsprechend der ausgewählten Computervorlage den Konfigurations- und Zuweisungsgruppen zugewiesen und aktiviert.

Hinweis: Falls der Computer schon in Empirum bekannt ist, wird er vor erneuter Erstellung zuerst aus allen Gruppen und aus Empirum vollständig entfernt.



Dieser Self Provisioning Bildschirm bleibt so lange sichtbar (Sanduhr), bis die neuen Verteilungsaufträge in die DDC-Datei geschrieben wurden, anschließend wird der Startbildschirm angezeigt und das gewohnte OS Deployment läuft durch.



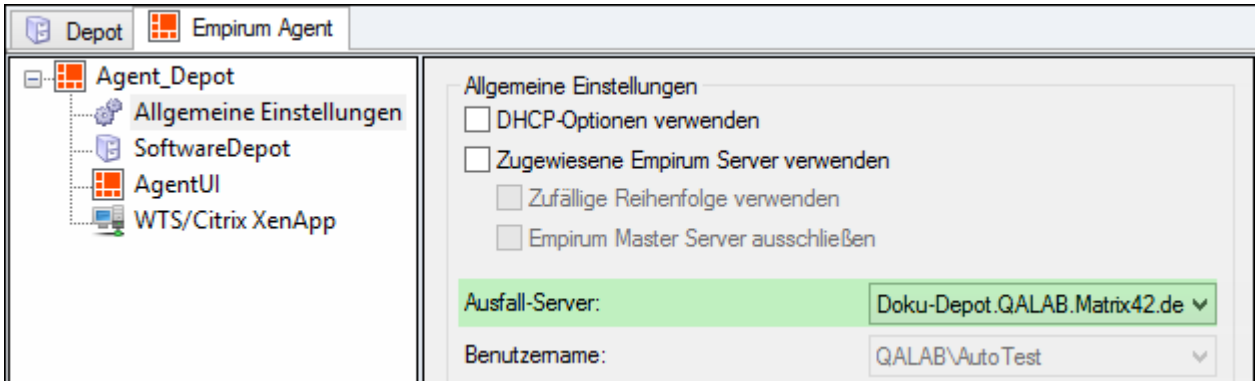
7.6 Self Provisioning über Depot Server (Offline)

Damit das Self Provisioning auch über einen Depot Server (Offline) durchgeführt werden kann, müssen folgende Voraussetzungen erfüllt sein:

- Angepasstes Empirum *Agent Template*.
- Angepasstes *WinPE Bootimage - Standard* und *Self Provisioning*.
- Eigene *Empirum Gruppe* mit entsprechend ausgewähltem *Empirum Server* (Depot).
- Die Variable *FQDN* muss mit einem korrekten Wert gefüllt sein.
- Das Sync-Template *ESubdepot_DeviceMapping* muss dem Depot Server (Offline) zugeordnet sein.

Empirum Agent Template

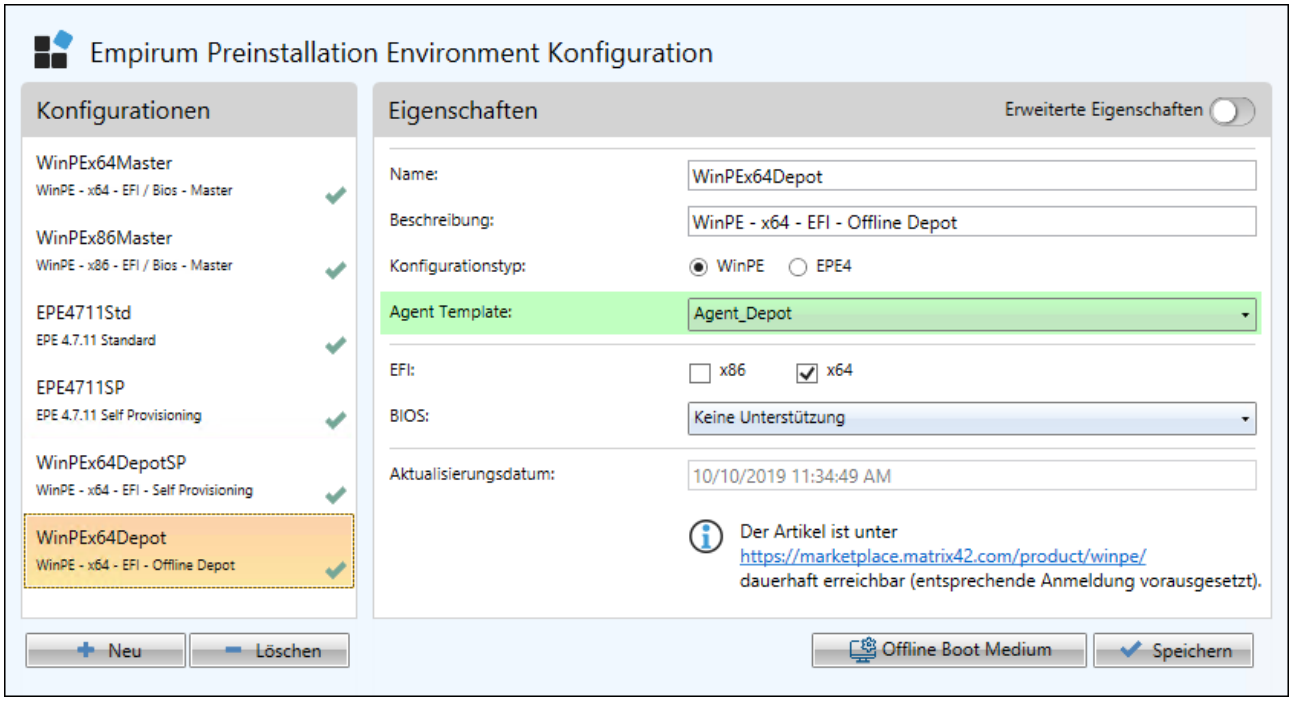
Am besten ist es, für den Depot Server (Offline) ein eigenes Empirum Agent Template zu erstellen (in diesem Beispiel "Agent_Depot").

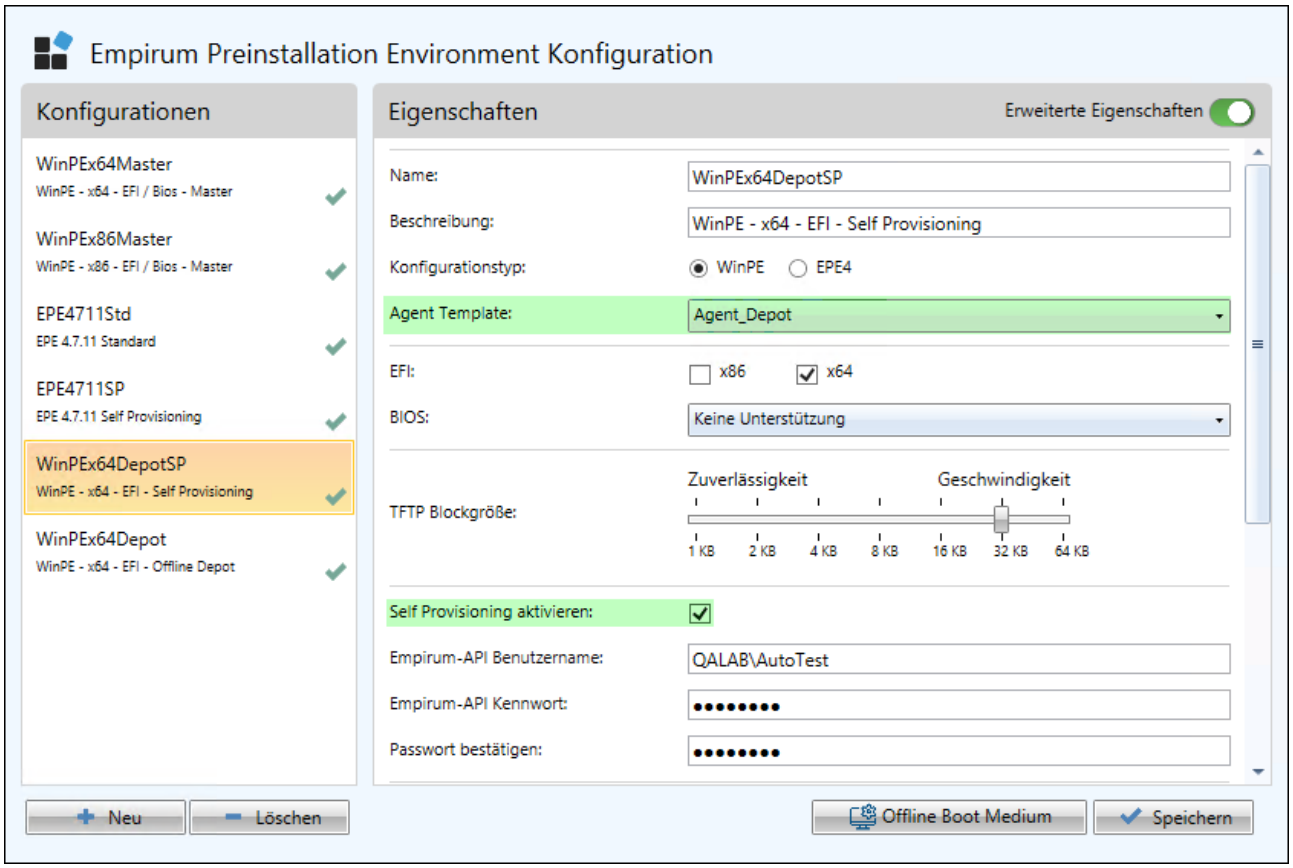


Wichtig ist, dass der Depot Server (Offline) hier als "***Ausfall-Server***" ausgewählt wurde (in diesem Beispiel "Doku-Depot.QALAB.Matrix42.de").

WinPE Bootimages

Für das Standard WinPE Boot Image und für das Self Provisioning Boot Image muss jeweils das zuvor erstellte Agent Template ausgewählt sein (in diesem Beispiel "Agent_Depot").

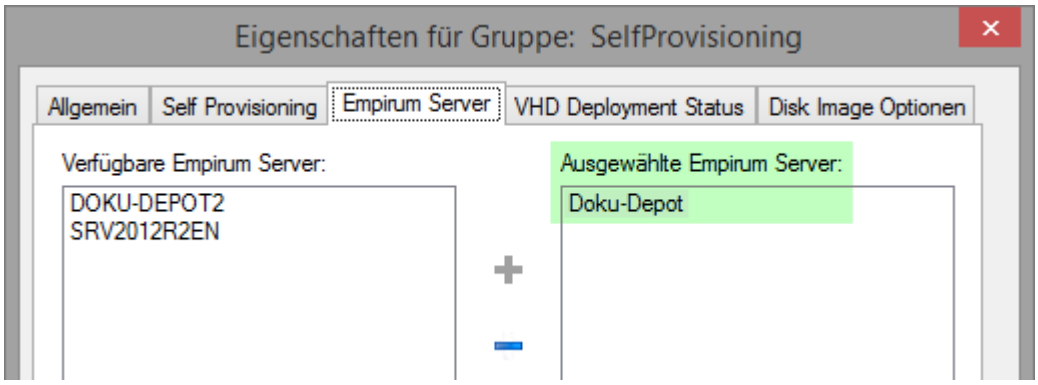




Empirum Gruppe

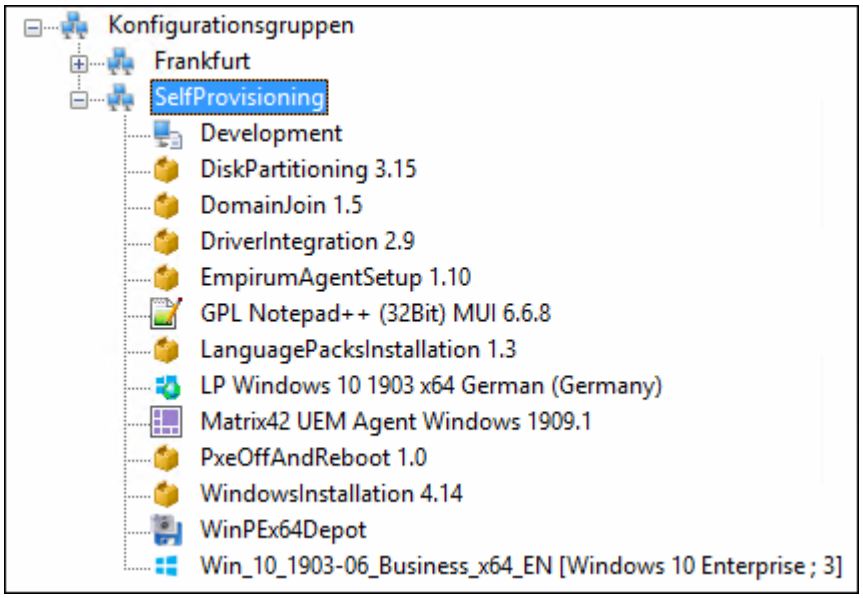
Sinnvoll ist es, hier eine eigene Konfigurationsgruppe zu erstellen (in diesem Beispiel "SelfProvisioning"). Damit die Konfigurationsdateien zum richtigen Depot Server übertragen (Sync) werden, müssen Sie folgende Einstellung vornehmen.

Klicken Sie mit der *rechten Maustaste* auf diese *Gruppe* und wählen Sie *Eigenschaften*. Wechseln Sie in das Register *Empirum Server*. Wählen Sie unter *Verfügbare Empirum Server* Ihren *Depot Server* (Offline) aus und fügen Sie ihn über die *Plus* Schaltfläche unter ***Ausgewählte Empirum Server*** hinzu.



Für einen funktionsfähigen Durchgang müssen dieser Konfigurationsgruppe die von Ihnen benötigten *PreOSPakete*, ein *Betriebssystem-Import*, ein *PXE-BootImage* (Standard, kein Self Provisioning) und eine *Computervorlage* zugeordnet sein. Optional können *Sprachpakete*, der *UEM Agent* und weitere *Software Pakete* zugeordnet werden.

Auch alle hier zur Verwendung kommenden *Variablen* müssen auf dieser Konfigurationsgruppe gesetzt sein, insbesondere die *FQDN* Variable.



Hinweis: Das Standard Boot Image wird benötigt, falls weitere Reboots während der Betriebssysteminstallation notwendig sind.

8 WinPE-basiertes OS Deployment via http(s)

Mit der WinPE PreBoot Version 1.8.0 wird der Verbindungsaufbau über http(s) unterstützt. Das folgende Kapitel beschreibt die notwendigen Anpassungen und die aktuellen Einschränkungen der Umsetzung.

Achtung: Das Deployment via http(s) setzt die Verwendung der aktuellen Matrix42 PreOS Pakete voraus, die mit der WinPE PreBoot Version 1.8.0 veröffentlicht wurden:

- *HardwareInfo 3.0* oder höher

- *WindowsInstallation 5.0* oder höher

- *EmpirumAgentSetup 2.0* oder höher

Stellen Sie sicher, dass die verwendete Boot Konfiguration ebenfalls mit dem aktuellen Stand der WinPE PreBoot Version 1.8.0 gebaut wurde.

8.1 Aktuelle Einschränkungen

Mit der aktuellen Umsetzung sind folgende Einschränkungen zu berücksichtigen:

- Disk Imaging unterstützt noch keine http(s) Übertragung.
- Der Zeitstempel der Swdepot-Log Einträge für die PreOS Pakete können im http(s) Fall abweichen.

8.1.1 http(s) Unterstützung bei selbsterstellten PreOS Paketen

Eventuell sind selbst erstellte PreOS Pakete nicht sofort für den http(s) Transport geeignet. Der Grund dafür ist meistens, dass in den PowerShell-Skripten auf Dateien oder Verzeichnisse auf den Empirum-Share im SMB Fall zugegriffen werden, die nicht Teil des PreOS Paketes selbst sind.

Um die Anpassungen zu erleichtern soll an dieser Stelle auf bestimmte Eigenheiten beim http(s) Transfer eingegangen werden.

- Generell wird im http(s) Fall mit lokalem Cache gearbeitet. Dateien und Verzeichnisse werden von der Empirum-Freigabe über http(s) in lokale Verzeichnisse transferiert. Es werden nicht alle Freigaben automatisch transferiert, sondern nur Verzeichnisse und Dateien, die angefragt werden.
- Im http(s) Fall wird vor der Ausführung des PreOS Paketes, der Inhalt des Paketes (z.B. wird im Fall des WindowsInstallation PreOS Paketes das Verzeichnis *Configurator$\Packages\Matrix42\OsPackages\WindowsInstallation\5.0)* zunächst lokal in ein Cache-Verzeichnis übertragen und im Anschluss das *install.ps1* Skript lokal ausgeführt. Relative Zugriffe auf Dateien, die sich im Paket befinden, sind somit abgedeckt.
- Muss in einem PreOS Paket auf Dateien zugegriffen werden, die nicht in dem Paketverzeichnis vorhanden sind, muss dies über das Cmdlet *Get-EmpirumPackagePath* stattfinden. Die Methode wurde bereits in der Vergangenheit verwendet, um den Pfad auf die Empirum-Freigabe (oder das Offline Medium) zu bekommen. Diese Methode muss im http(s) Fall die angegebene Datei oder das Verzeichnis zunächst lokal in ein Cache-Verzeichnis übertragen und im Anschluss den

lokalen Pfad zurückgeben. Das PreOS Paket kann somit mit dem lokalen Pfad weiterarbeiten und auf die Datei oder das Verzeichnis zugreifen.

- Das Cmdlet `Get-EmpirumAgentSetting -PeAgentConfig RemoteLogFolder` kann im http(s) nicht verwendet werden, um Log-Dateien auf der Empirum-Freigabe zu ändern. Für diese Zwecke wird das neue Cmdlet `Get-EmpirumTransfer` zur Verfügung gestellt. Mit dem Aufruf `Get-EmpirumTransfer LogFolder -Type RemoteLogFolder`, kann vergleichbar dem *Get-EmpirumPackagePath* Aufruf, das EmpInst$\Wizard\OS\WinPeStatus Unterverzeichnis lokal transferiert werden. Die Methode gibt ein Objekt zurück, mit dessen Hilfe über die *Path*-Eigenschaft auf den Pfad zum lokalen Cache zugegriffen werden kann. Das PreOS Skript kann nach den Anpassungen der Dateien im Cache das Synchronisieren der lokalen Dateien zum Server über die Methode *Sync* veranlassen. Dabei werden die lokalen Änderungen auf den Server übertragen.

Hier ein Beispiel aus dem HardwareInfo PreOS Paket:

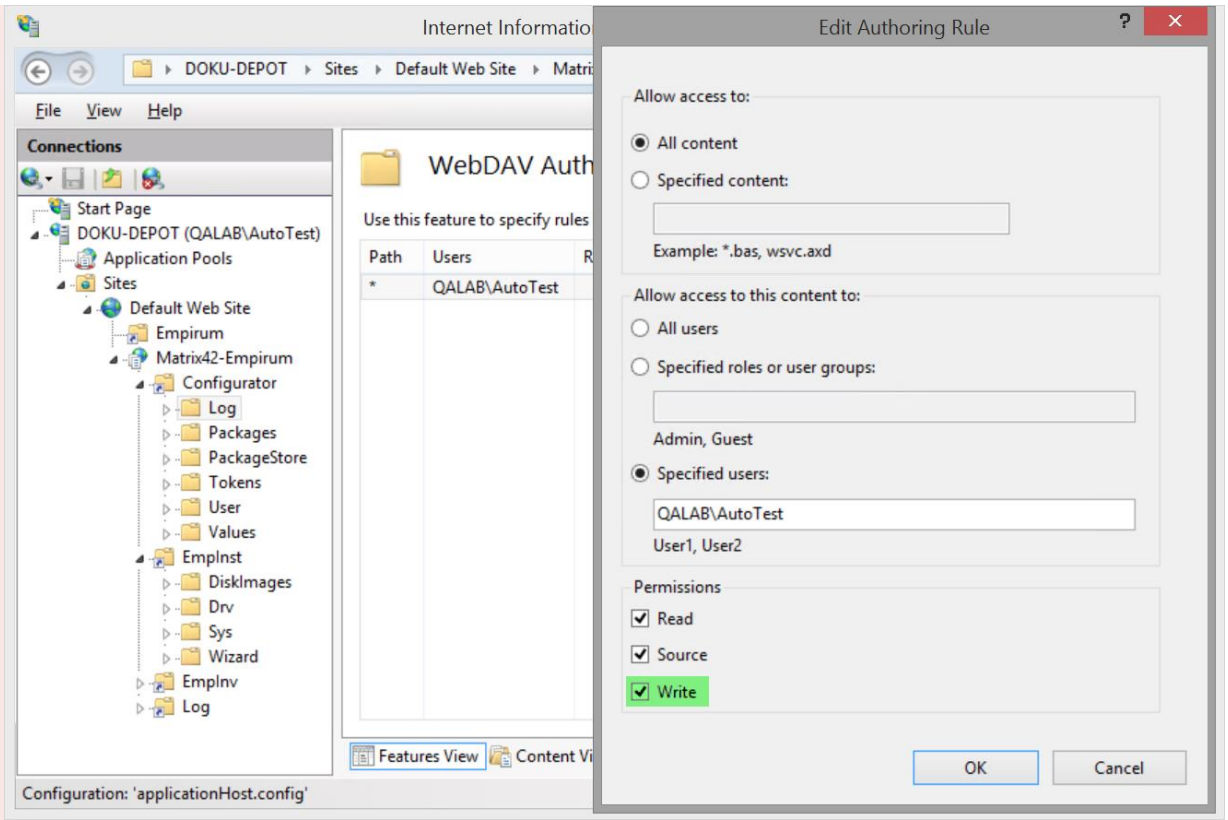
```
$RemoteFolderObject = Get-EmpirumTransfer LogFolder -Type RemoteLogFolder -Verbose;

…
CreateAndCopyDriverJson $RemoteFolderObject.Path;

$RemoteFolderObject.Sync();
```

8.2 WinPE basiertes Deployment via http(s) einrichten

Die folgenden Schritte beschreiben die Anpassungen, die vorgenommen werden müssen, damit beim WinPE basierten Deployment der Datentransfer per http(s) durchgeführt und nicht mehr über den Empirum-Share direkt über SMB zugegriffen wird.

1 Konfigurieren und installieren Sie das *Empirum Subdepot Webservice Configuration* Paket auf einem Subdepot über das das Deployment via http(s) durchgeführt werden soll, wie es in der Online-Hilfe beschrieben ist Empirum Subdepot Webservice Configuration Paket.

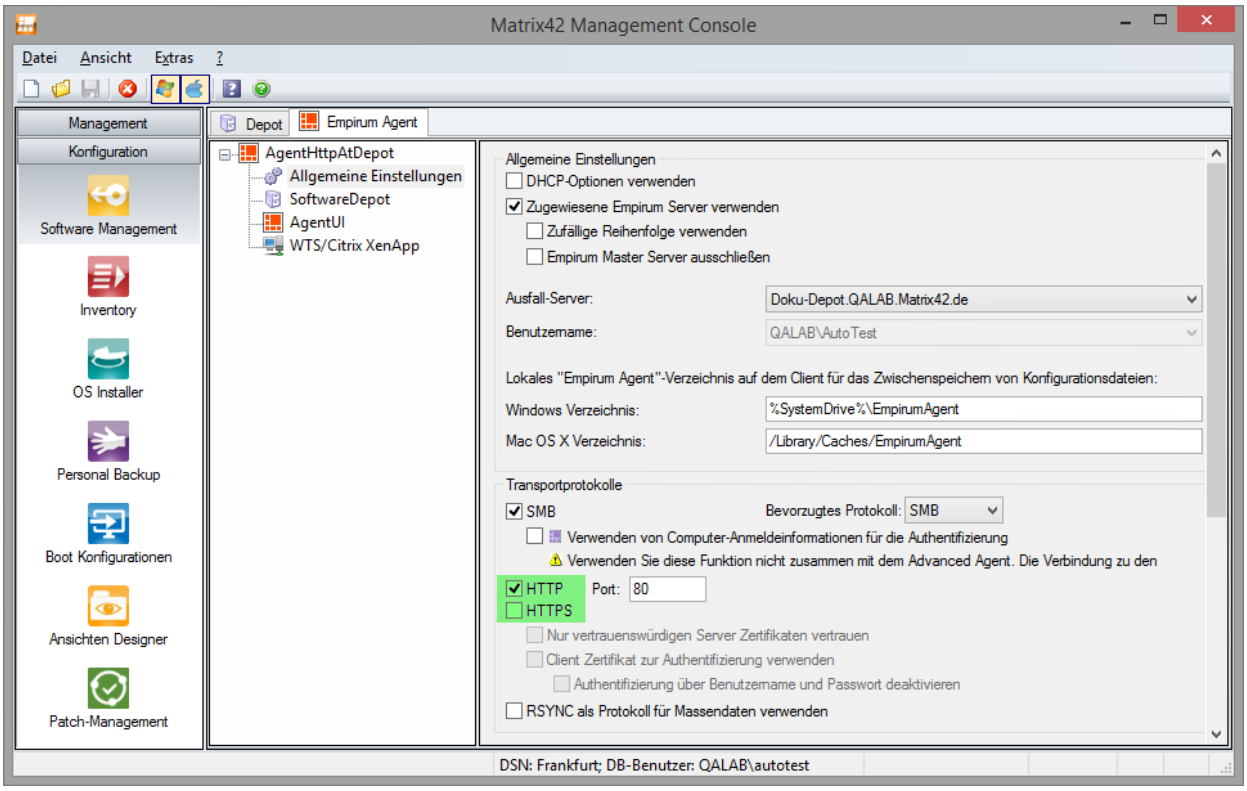
> **Achtung:** Bei Verwendung von Empirum bis einschließlich Version 20.0.0 müssen noch Schreibrechte für den Benutzer auf dem IIS vergeben werden, der für den Verbindungsaufbau verwendet wird. Welcher Benutzer verwendet wird, entnehmen Sie dem verwendeten Empirum Agent Template unter *Allgemeine Einstellungen > Benutzername*.
> Wechseln Sie im *IIS Manager* im linken Baum auf "*Sites\Default Web Site\Matrix42-Empirum\Configurator\Log*", klicken Sie im mittleren Fenster doppelt auf "*WebDAV Authoring Rules*" und anschließend auf den von Ihnen verwendeten *Benutzer*. Setzen Sie unten bei *Permissions* das Häkchen für die Option *Write* und bestätigen Sie mit *OK*.



Wechseln Sie im *IIS Manager* im linken Baum auf "*Sites\Default Web Site\Matrix42-Empirum/EmpInst/Wizard/OS/WinPeStatus*", klicken Sie auch hier im mittleren Fenster doppelt auf "*WebDAV Authoring Rules*" und anschließend auf den von Ihnen verwendeten *Benutzer*. Setzen Sie unten bei *Permissions* das Häkchen für die Option *Write* und bestätigen Sie mit *OK*.

Hinweis:
Seit Empirum v20.0.1 ist das manuelle Setzen von Schreibrechten für den Benutzer auf dem IIS nicht mehr notwendig.

2 Erstellen Sie ein *Empirum Agent* Template, bei dem die Option *HTTP* oder *HTTPS* im Bereich der Transportprotokolle angehakt ist. Stellen Sie sicher, dass das im ersten Schritt vorbereitete Subdepot als *Ausfall-Server* ausgewählt ist.



> **Hinweis:**
> Sobald die Option *HTTP* oder *HTTPS* im Agenten Template ausgewählt ist, wird das SMB Protokoll nicht verwendet, auch wenn es aktiviert ist.
> Bei Auswahl des HTTPS Protokolls muss sichergestellt werden, dass der hier ausgewählte *Ausfall-Server* auch von "*außen*" unter diesem Namen erreichbar ist.

3 Wechseln Sie in die *Boot Konfiguration* und führen Sie über das Menü *Ansicht* > *Aktualisieren* aus, damit die aktuellen Agenten Templates und deren Werte erneut aus der Datenbank gelesen werden.

4 Wählen Sie das erstellte *Empirum Agent Template* in einer *Boot Konfiguration* aus.

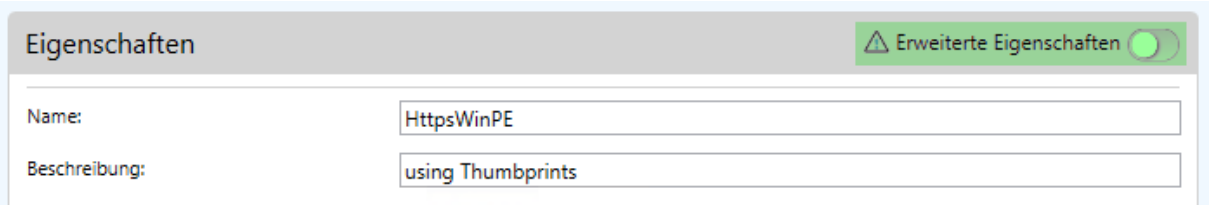
5 Wird die Option "*Nur vertrauenswürdigen Server Zertifikaten vertrauen*" im *HTTPS* Fall in dem verwendeten Agenten Template verwendet, müssen Sie die *Fingerabdrücke* der Serverzertifikate hinterlegen, denen vertraut werden soll. Zur WinPE-Laufzeit funktioniert die übliche Validierung über den Zertifikatsspeicher und die Zertifikatsketten nicht.

> **Hinweis:**
>
> In älteren Empirum Versionen (vor v20.0.2) können die Fingerabdrücke noch nicht über die Bootkonfiguration hinterlegt werden.
>
> Der Fingerabdruck, oder eine kommaseparierte Liste der Fingerabdrücke (ohne Leerzeichen), muss in diesen Fällen vor dem Abspeichern der Bootkonfiguration in der *Matrix42.Empirum.PeAgent.dll.config* Datei auf der Empirum-Freigabe im Verzeichnis *EmpInst$\Sys\Images\WinPE\binaries\UAF* eingetragen werden.

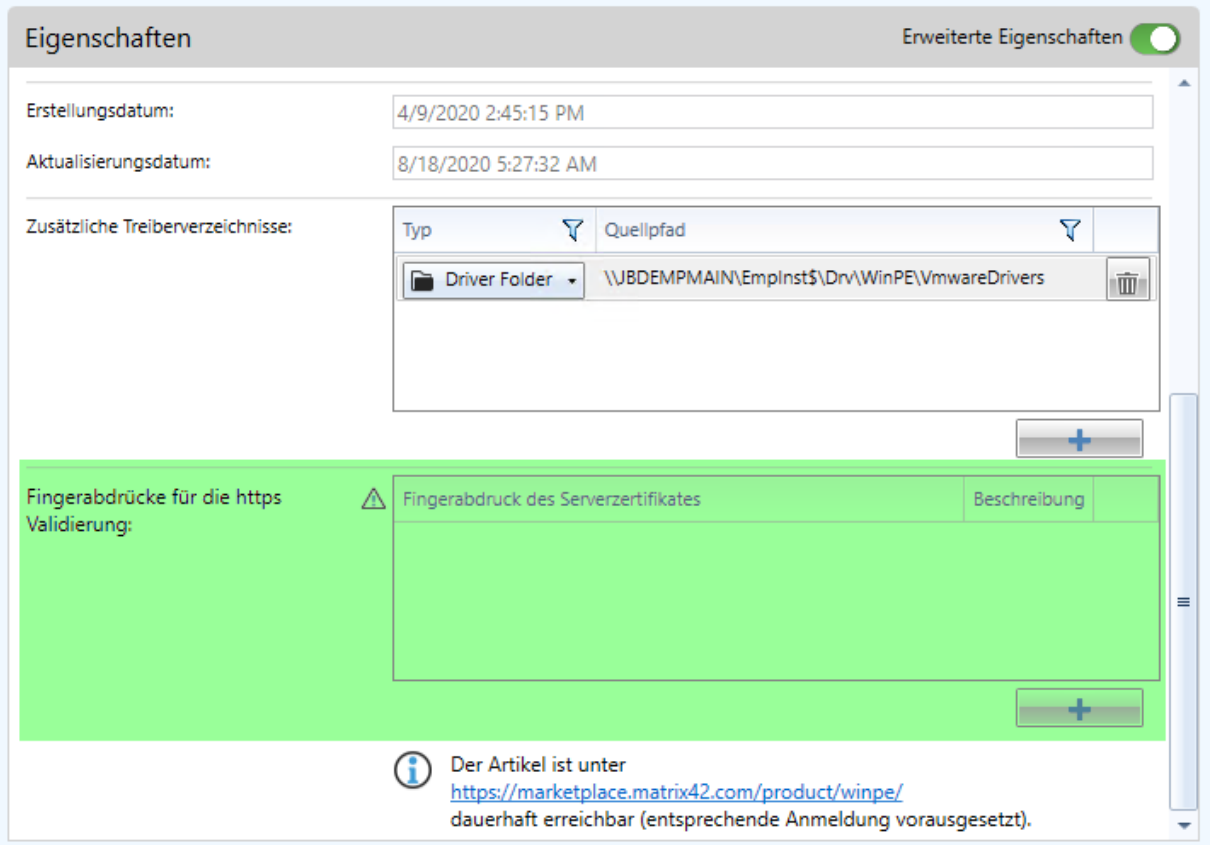
```xml
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>
  <appSettings>
    <add key="DelayOnErrorInSeconds" value="300" />
    <add key="DelayRebootInSeconds" value="0" />
    <add key="EnableReboot" value="True" />
    <add key="UsePXEServerAsEmpirumServer" value="False" />
    <add key="PowerShellTimeoutInSeconds" value="3600" />
    <add key="UafHostWindowsServiceName" value="Matrix42 UAF Host for PE Agent" />
    <add key="OpenConnectionRetries" value="10" />
    <add key="WaitAfterInitialize" value="False" />
    <add key="ThumbprintsAtHttpsCase" value = "28fcdc3b9d4ba0b82a6d06ac885bd88676e7d5b1" />
  </appSettings>
  <runtime>
</configuration>
```

Die folgenden Schritte sind somit in den älteren Empirum Versionen nicht durchzuführen.

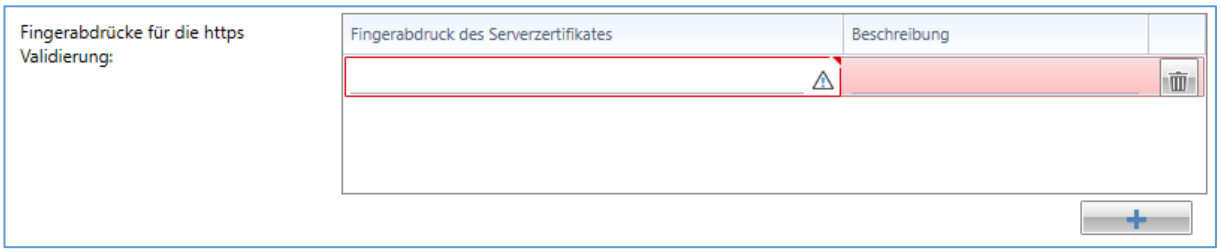
1 Stellen Sie sicher, dass die erweiterten Eigenschaften der Bootkonfiguration angezeigt werden, in dem Sie den Schalter *Erweiterte Eigenschaften* aktiviert.



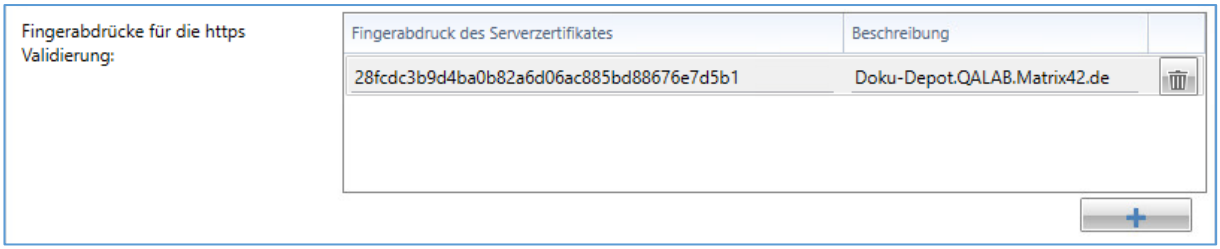
2 Im unteren Teil des Eigenschaftenbereichs wird einen List für die Fingerabdrücke angezeigt, die aktuell noch leer ist.



3 Fügen Sie mit der + Schaltfläche einen neuen Fingerabdruck hinzu.

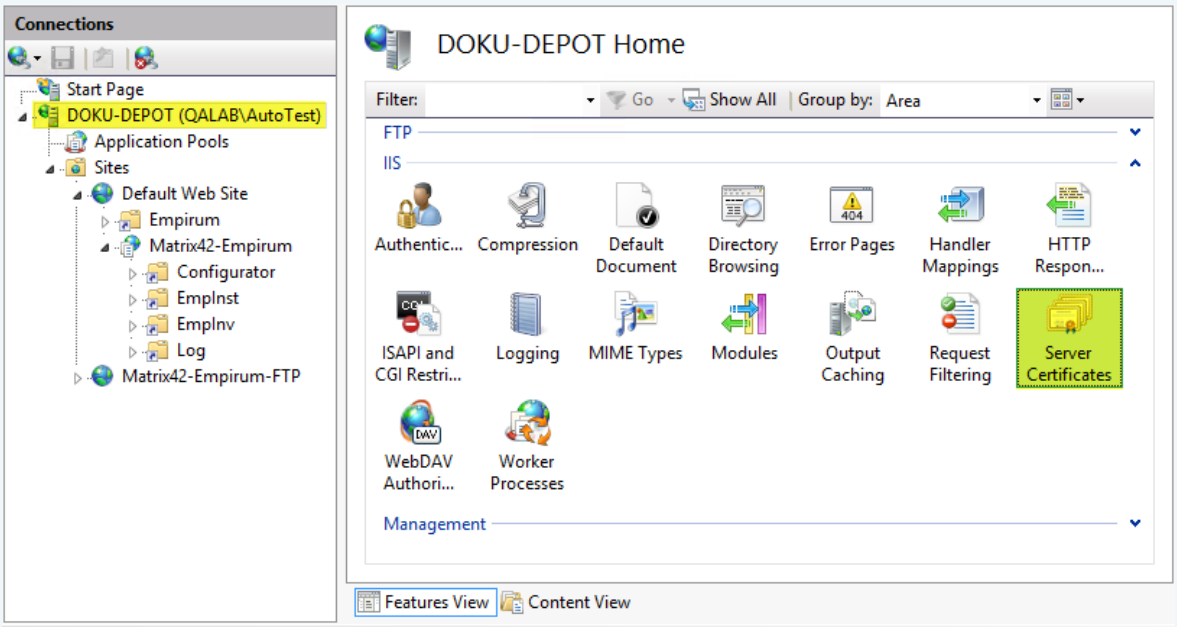


4 Übertragen Sie nun den Fingerabdruckwert in die Spalte *Fingerabdruck des Serverzertifikates* und fügen Sie noch eine Beschreibung hinzu.

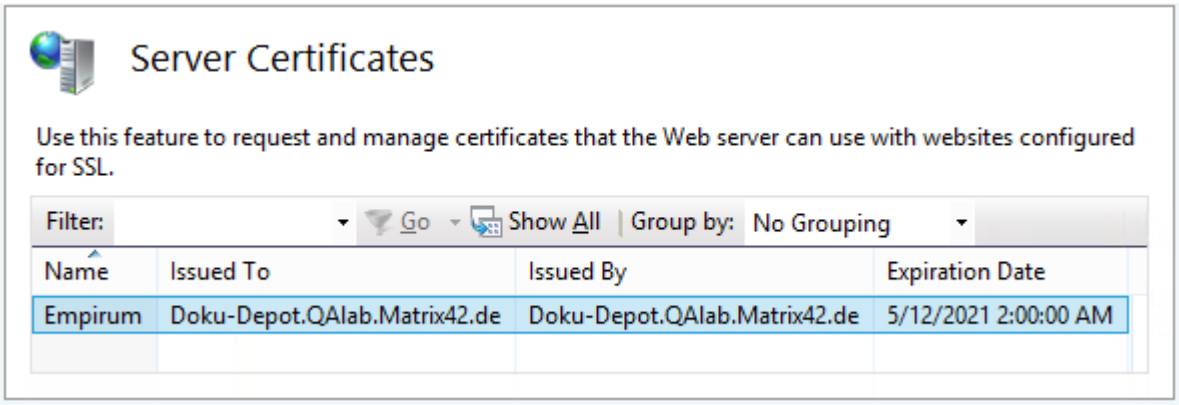


Hinweis: Die vom IIS verwendeten Zertifikate können im *Internet Information Service (IIS) Manager* überprüft und deren Fingerabdrücke ausgelesen werden.

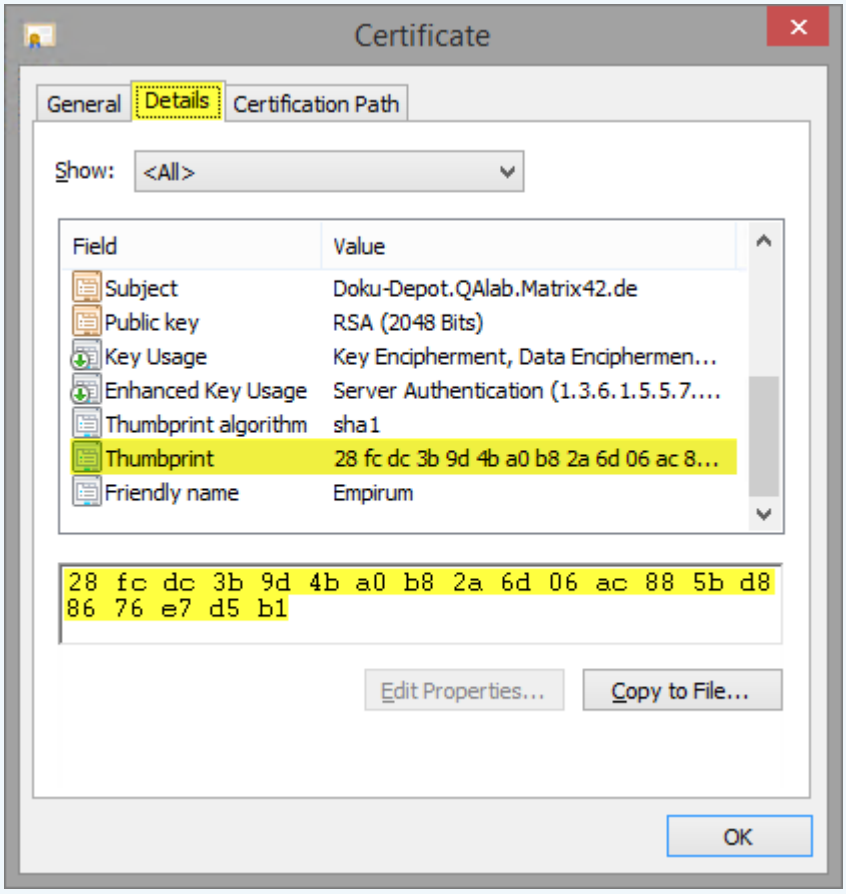
Wechseln Sie im *IIS Manager* im linken Baum auf die oberste Ebene auf den Eintrag des Servernamens, klicken Sie im mittleren Fenster doppelt auf "*Server Certificates*".



Im mittleren Fenster werden dann die Zertifikate angezeigt. Klicken Sie das verwendete Zertifikat doppelt an, um sich die *Eigenschaften* des Zertifikats anzeigen zu lassen.



Im Reiter "*Details*" kann nun das Feld "*Thumbprint*" ausgewählt werden.



5 Falls die Bootkonfiguration für mehrere Depots verwendet werden soll, wiederholen Sie den Vorgang für jedes Depot und tragen Sie die entsprechenden Fingerabdrücke des Serverzertifikates ein.

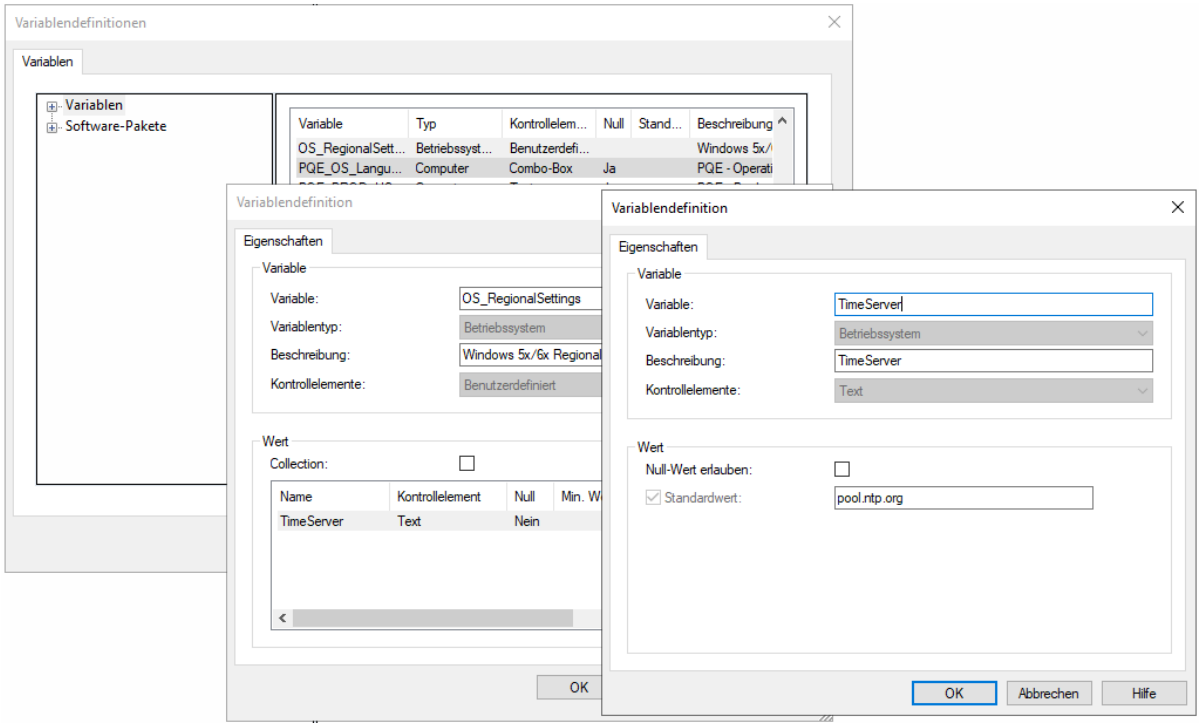
6 *Speichern* Sie die Boot Konfiguration erneut ab, damit die Änderungen übernommen werden.

7 Soll ein anderer Zeitserver als *'pool.ntp.org'* verwendet werden, kann dies über die Variable *OS_RegionalSettings.TimeServer* konfiguriert werden. Gehen Sie dazu wie folgt vor:

> **Hinweis:** Als Standard wird 'pool.ntp.org' als Zeitserver verwendet. Soll dies nicht abgeändert werden, brauchen Sie diese Schritte nicht durchführen.

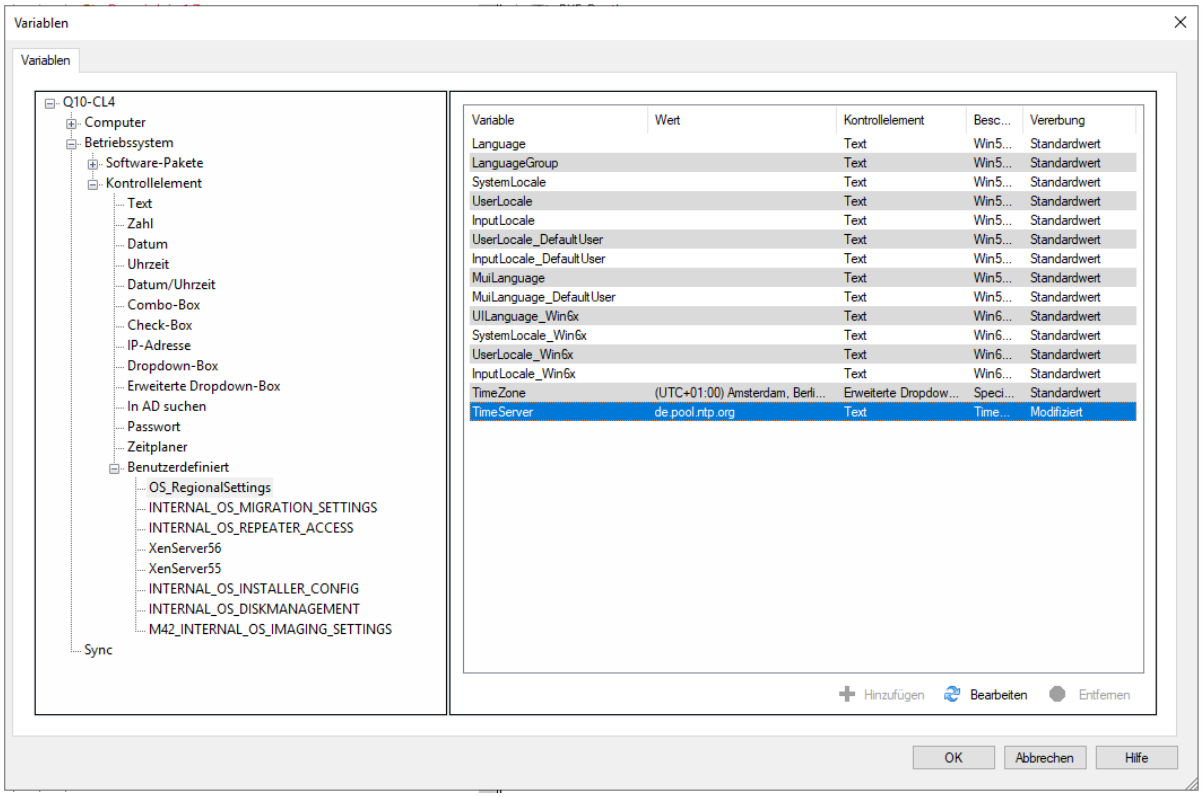
1 Wechseln Sie in der *Administration* zu *Menü Extras > Variablendefinitionen.*

2 Erzeugen Sie die Variable *TimeServer* vom Typ *Betriebssystem* und dem Kontrollelement *Text* (s. Screenshot).



3 Betätigen Sie mit *OK* zum Speichern und zum Schließen der Dialoge.

4 Wählen Sie anschließend den Client oder die jeweilige Konfigurationsgruppe aus und tragen den von Ihnen gewünschten Zeitserver ein, bspw. 'de.pool.ntp.org' (s. Screenshot).



8 Weisen Sie die erstellte *Boot Konfiguration* in der Administration der jeweiligen *Konfigurationsgruppe* zu und *aktivieren* Sie das Deployment.

9 End Of Life (EOL) per WinPE

Mit der *WinPE Preboot Version 1.8.3* wird *End Of Life* unterstützt.

End Of Life löscht alle Massenspeicher (nicht USB) des angeschlossenen Clients.

- Jede Partition wird mit verschiedenen Dateisystemen formatiert.
- Danach werden die entsprechenden Partitionen gelöscht.
- Dann werden Zufallsdaten mit vordefinierter Anzahl (Standard 10 GB) auf der Platte verteilt.
- Zum Schluss wird die Platte in einen definierten "Clean" Zustand versetzt.

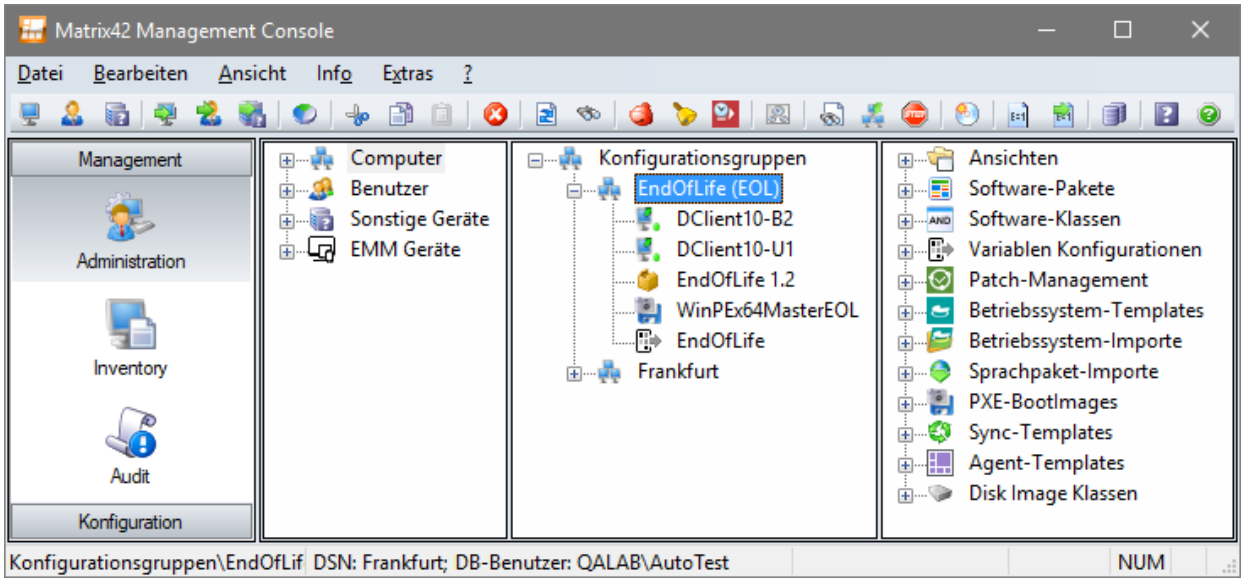
Hinweis: Wird eine NVME Platte erkannt, wird ein sicheres Löschen über "NVME-Format" durchgeführt, das gilt ab der WinPE PreBoot Version 1.8.5 und ist Bestandteil ab WinPE EndOfLife 1.1.
Für alle anderen Platten ist eine "NVME-Format" Funktion noch nicht Bestandteil von WinPE EndOfLife.

9.1 End Of Life konfigurieren (EOL)

Nachdem Sie die *WinPE_PreBoot_Support* Datei ab Version *1.8.3* aus dem Marketplace heruntergeladen und laut Beschreibung integriert haben, müssen Sie das neue EndOfLife-Paket importieren.

Erstellen Sie eine Konfigurationsgruppe möglichst weit oben im Baum, um zu verhindern, dass PreOS Pakete oder Variablen vererbt werden. Das kann dazu führen, dass das EndOfLife-Paket mit einer Fehlermeldung beendet wird, obwohl die Platten erfolgreich gelöscht wurden.

1 Erstellen Sie eine dedizierte *Konfigurationsgruppe* (im Beispiel EndOfLife (EOL)) im mittleren Baum in der *Matrix42 Management Console*.

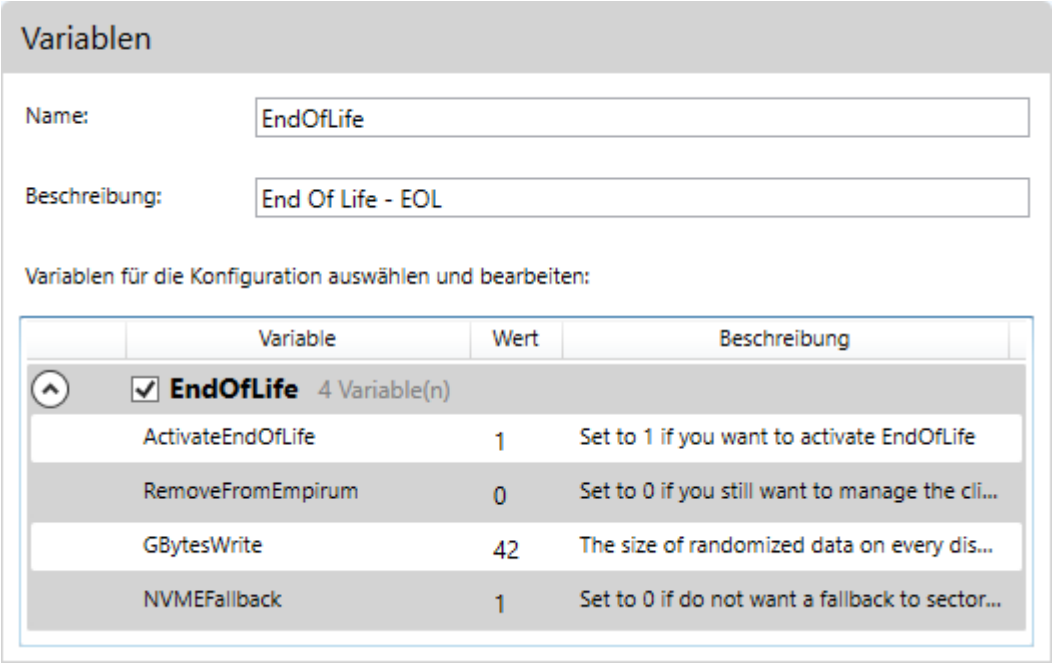


2 Ordnen Sie (nur) das PreOS Paket *EndOfLife* dieser Gruppe zu.
Kontrollieren Sie, dass diese Gruppe keine weiteren WinPE Pakete oder Variablen erbt.

3 Weisen Sie ein dediziertes und *aktuelles WinPE-BootImage* zu, dass (wenn Sie die Platte(n) komplett überschreiben wollen) mit einem *höheren Timeout Wert* erstellt wurde.

Hier ist der Standard 3600 Sekunden (1 Stunde) - beim kompletten Überschreiben werden 36000 Sekunden (10 Stunden) empfohlen.

4 Erstellen Sie eine *Variablen Konfiguration* mit den Variablen des *EndOfLife-Paketes*.



GBytesWrite

Gibt die Menge an Zufallsdaten an (Standard 10 GB), die auf jede Platte geschrieben werden. Kann in GB-Schritten eingestellt werden. Der Wert "*0*" überschreibt die komplette Platte einmal mit Zufallsdaten. Je nach Anzahl und Größe der Platten kann dieser Vorgang einige Stunden dauern. Hier muss gegebenenfalls der Time Out Wert des WinPE Bootimages angepasst werden.

Diesen Wert können Sie in der Datei *Matrix42.Empirum.PeAgent.dll.config* im Verzeichnis "*.\Empirum\EmpInst\Sys\Images\WinPE\binaries\UAF*" ändern.

Matrix42.Empirum.PeAgent.dll.config

```
1   <?xml version="1.0" encoding="utf-8"?>
2   <configuration>
3     <configSections>
8     <appSettings>
9       <add key="DelayOnErrorInSeconds" value="300" />
10      <add key="DelayRebootInSeconds" value="0" />
11      <add key="EnableReboot" value="True" />
12      <add key="UsePXEServerAsEmpirumServer" value="False" />
13      <add key="PowerShellTimeoutInSeconds" value="36000" />
14      <add key="UafHostWindowsServiceName" value="Matrix42 UAF Host for PE Agent" />
15      <add key="OpenConnectionRetries" value="10" />
16      <add key="WaitAfterInitialize" value="False" />
17    </appSettings>
18    <runtime>
26  </configuration>
```

Wenn dieser Wert geändert wird, muss das WinPE Boot Image neu erstellt werden, damit die Änderung auch übernommen wird!

RemoveFromEmpirum

Steuert das clientspezifische Verhalten nach einer End Of Life Prozedur.

Sie können über die Vergabe des Wertes "*0*" bestimmen, dass der Client nach der End Of Life Prozedur in der EMC und Empirum als gemanagtes Objekt verbleibt.

Wenn diese Variable auf "*1*" (Standard) steht, wird der Client nach der End Of Life Prozedur aus der EMC und Empirum entfernt.

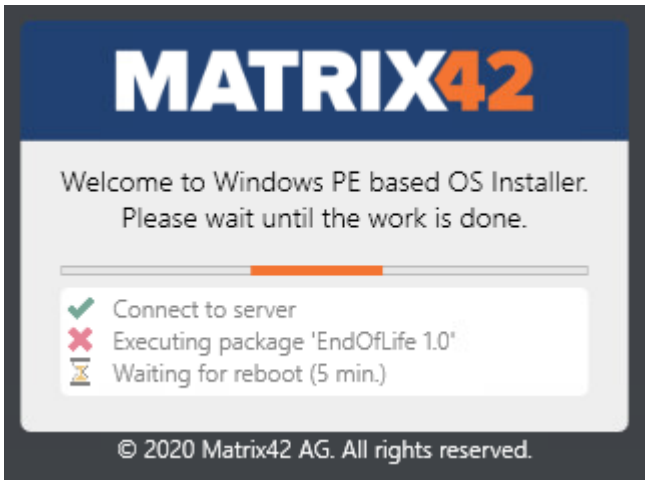
End Of Life hat keinen Einfluss auf die AD-Eigenschaften des Clients.

- **NVMEFallback**
  Steuert das Verhalten bei NVME-Format Fehlern. Standard: tritt ein Fehler beim NVME-Format auf, wird die Platte danach klassisch sektorweise gelöscht (NVMEFallback="1"). Sie können über die Vergabe des Wertes "0" bestimmen, dass NVME-Format Fehler zum Abbruch führen und die Platte ungelöscht verbleibt.

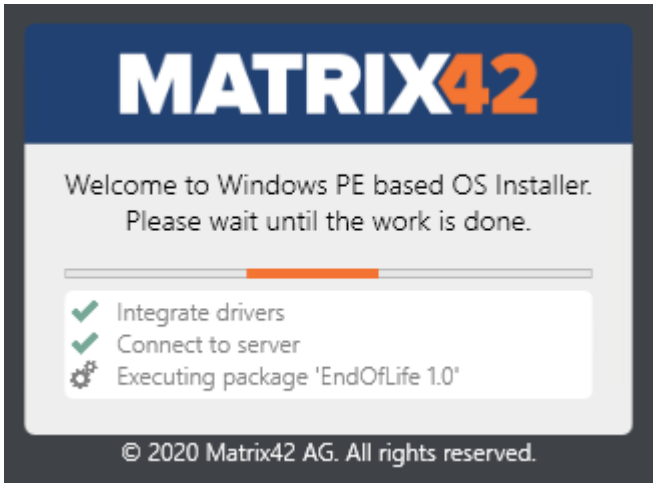
- **ActivateEndOfLife**

  Diese Variable ist eine *Sicherheitsfunktion* und muss manuell auf *1* gesetzt werden, damit End Of Life startet.

  Hat die Variable den Wert *0*, wird die Ausführung abgebrochen, im Log gibt es eine entsprechende Fehlermeldung.



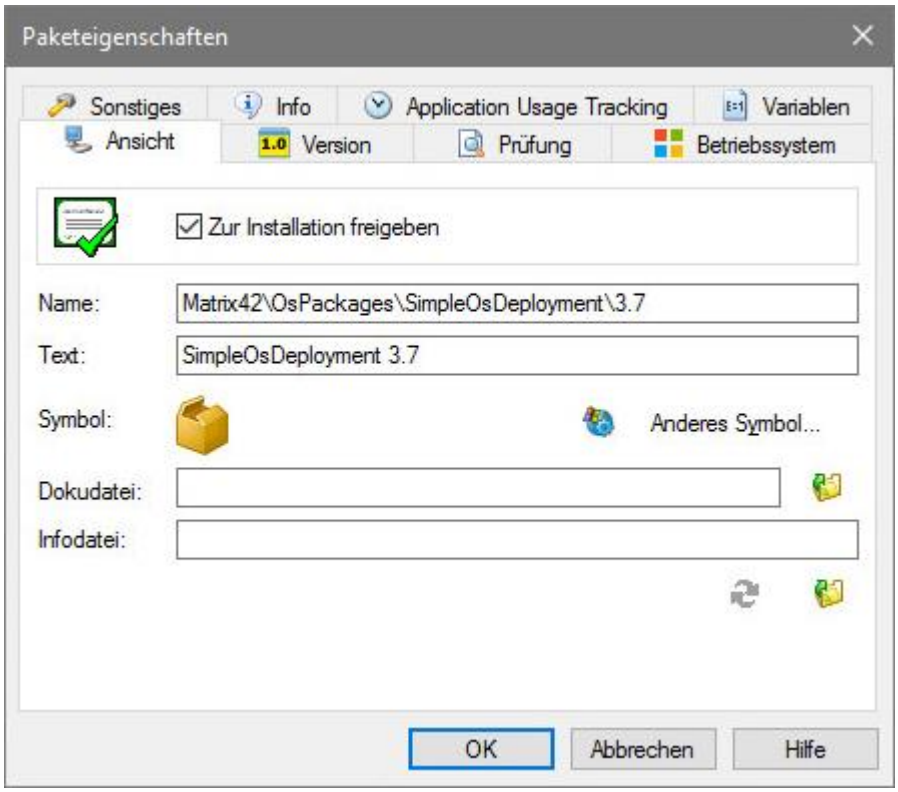
5 Weisen Sie die zu *löschenden Clients* zu und *aktivieren* Sie diese (PULL via DDS/DDC und PXE). End Of Life wird beim nächsten Boot ausgeführt.



10 FAQ

Frage: Ich habe nach dem erfolgreichen Import und der erfolgreichen Variablendeklaration das SimpleOsDeployment Paket (oder ein beliebig anderes Paket) zugewiesen, und den Client aktiviert, jedoch startet das Paket nicht und das Betriebssystem wird nicht installiert.

Antwort: Stellen Sie sicher, dass das SimpleOsDeployment Paket (oder ein beliebiges anderes Paket) auch zur Installation freigegeben wurde.



Frage: Der Client scheint Fehler beim Ausführen zu haben und startet die ganze Zeit neu. Wie kann ich verhindern, dass er neu startet, damit ich nachschauen kann, was fehlschlägt?

Antwort: Auf dem Client wird der PE-Agenten ausgeführt, welcher die PreOS-Pakete ausführt. Bei einem Fehler wird immer ein Reboot durchgeführt, sodass ein fehlerhaftes Paket immer wieder geladen wird und der Client immer wieder neu startet. Der PE-Agent verfügt über eine eigene Konfigurationsdatei, mit der man das Rebooten deaktivieren kann, um das Problem zu debuggen (Matrix42.Empirum.PeAgent.dll.config). Den Schalter "EnableReboot" muss man auf "False" setzen, um Neustarts zu deaktivieren. Mit den Schaltern "DelayOnErrorInSeconds" und "DelayRebootInSeconds" kann man auch die Ausführung im Fehlerfall oder vor Neustarts verzögern (Angabe in Sekunden).

```
9        <add key="DelayOnErrorInSeconds" value="300"/>
10       <add key="DelayRebootInSeconds" value="0"/>
11       <add key="EnableReboot" value="True"/>
12     </appSettings>
13   </configuration>
```

Eine einfachere Variante ist das Verwenden des Matrix42 PreOS Paket *DebugPackage*. Das Paket muss dem Client zugewiesen, der Client muss aktiviert werden und erneut booten. Das *DebugPackage* Paket deaktiviert den Reboot vom PE-Agenten, so dass nach Ausführung (erfolgreich oder fehlerhaft) der zugewiesenen Pakete der PE-Agent den Client nicht neu bootet.

> **Hinweis:** Das Matrix42 PreOS Paket *DebugPackage* benötigt eine PE-Agent Version 1.3.0 oder höher.

Frage: Beim Import eines Matrix42 PreOS-Paketes landet das Paket nicht in dem Register, von dem aus ich den Import gestartet habe. Was kann ich machen?

Antwort: Matrix42 PreOS-Pakete werden immer in das Register importiert, das in der Konfigurationsdatei (EmpirumPackageData.xml) des Paketes angegeben ist. Dieses Verhalten unterscheidet sich zu dem Verhalten beim Import von Softwarepaketen. Startet man dort den Import über ein spezielles Register (beispielsweise Software), dann werden die importieren Softwarepakete auch diesem Register hinzugefügt. Möchte man das Register ändern, in dem das Matrix42 PreOS-Paket landen soll, so sind Änderungen an der Datei EmpirumPackageData.xml im XML-Knoten SoftwareDepotRegister notwendig.

Frage: Der Domain Join funktioniert mit dem neuen WindowsInstallation 4.0-Paket nicht mehr. Was kann ich machen?

Antwort: Das WindowsInstallation-Paket enthält ab der Version 4.0 nicht mehr die Funktionalität des Domain Joins. Der Domain Join ist in ein eigenes PreOS-Paket mit dem Namen "DomainJoin" ausgelagert worden, dass nun zusätzlich zugewiesen und konfiguriert werden muss. Der Vorteil liegt nun darin, dass der Status des Domain Joins besser überprüft werden kann. Der Status des neuen Domain Join-Pakets soll über den Status des Domain Join Informationen geben.

Frage: Domain Join schlägt fehl: "2224 - The account already exists." Was kann ich machen?

Antwort: Das DomainJoin-Paket ist nicht dafür ausgelegt einen Computer aus einer bereits zugeordneten OU zu verschieben. Das heißt, dass der Computer bereits in einer OU zugeordnet war und diese sich von der aktuell konfigurierten unterscheidet. Entweder überprüft man die ORGANIZATIONAL_UNIT-Variable und stellt sicher, dass die OU korrekt konfiguriert worden ist oder man entfernt den Computer aus der OU in Active Directory (Domain Controller).

Frage: Der Empirum Agent ist nicht mehr nach der Verwendung des neuen WindowsInstallation 4.0-Paket installiert. Was kann ich machen?

Antwort: Das WindowsInstallation-Paket enthält ab der Version 4.0 nicht mehr die Funktionalität, dass der Empirum Agent installiert wird. Die Installation des Empirum Agenten ist in ein eigenes PreOS-Paket mit dem Namen "EmpirumAgentSetup" ausgelagert worden, dass nun zusätzlich zugewiesen und konfiguriert werden muss. Der Vorteil liegt nun darin, dass der Status der Empirum Agenteninstallation besser überprüft werden kann. Das neue Paket interpretiert ebenfalls die Variable

MX42_AGENT_PUSH_PACKAGE_FOLDER.Windows, um festzustellen welche Empirum Agent-Version installiert werden soll.

Frage: Beim Import eines Matrix42 PreOS-Paketes kommt es zu einem Fehler. Was kann ich machen?

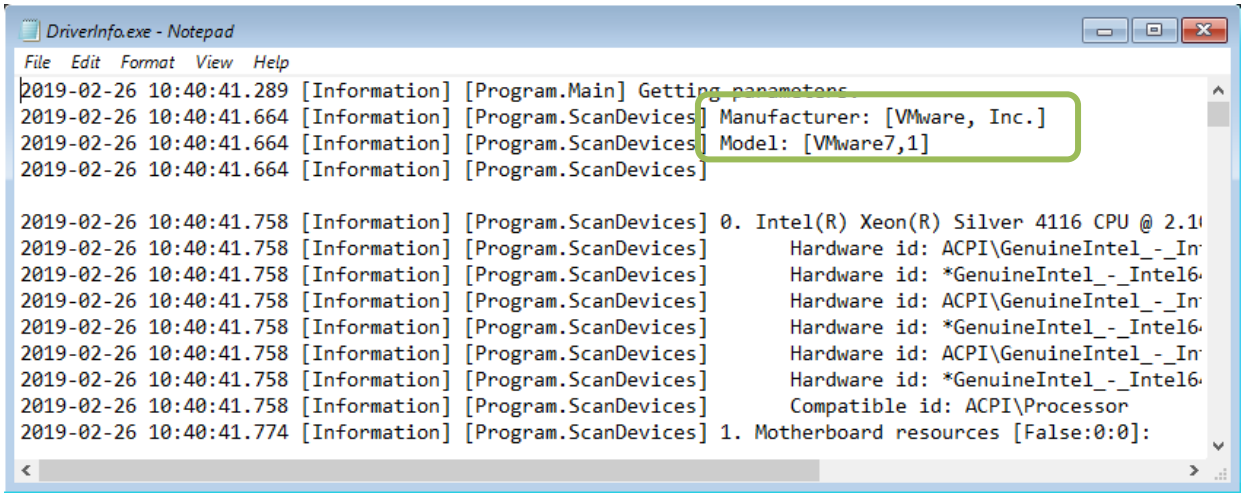
Antwort: Ältere Versionen eines Paketes können eventuell inkompatible Variablendefinitionen enthalten, die während des Imports nicht aktualisiert werden können. Löschen Sie zunächst alle älteren Versionen des Paketes, womit auch die inkompatiblen Variablendefinitionen gelöscht werden, und versuchen Sie den Import erneut zu starten. Beachten Sie bitte, dass die Variablenwerte dieses Paketes neu gesetzt werden müssen.

Frage: Beim Versuch einen Rechner per WinPE basierten Deployment aufzusetzen, bricht der PXE Boot des WinPE Images regelmäßig ab. Gibt es eine Möglichkeit die Übertragung zu stabilisieren?

Antwort: Mit Hilfe der TFTP Blockgrößeneinstellung kann in der Bootkonfiguration versucht werden die Übertragung stabiler zu gestalten. Eine kleine Blockgröße reduziert das Risiko, dass die Übertragung der einzelnen Blöcke abgebrochen wird und führt allerdings in der Regel zu einer verlangsamten Übertragung des Images.

Frage: Beim Versuch Hersteller und Model eines Rechners per HardwareInfo Paket zu bestimmen, habe ich das Problem, dass der Rechner keine Verbindung zum Empirum-Server aufbauen kann und deshalb auch das HardwareInfo Paket nicht ausführen kann. Gibt es eine Möglichkeit, die Informationen, ohne das HardwareInfo Paket zu bekommen?

Antwort: Ja. Jedes WinPE Image enthält eine Tool *DriverInfo.exe* im UAF-Verzeichnis, das nach dem Booten auf dem Rechner auch ohne Netzwerkverbindung ausgeführt werden kann. Das Tool gibt Informationen über die Geräte und deren Treiberversorgung auf dem Computer und schreibt die Informationen in eine Logdatei *DriverInfo.exe.log* im UAF-Verzeichnis. In den ersten Zeilen dieses Logs wird der Herstellername und der Modelname ausgegeben.



Frage: Beim Ausführen des Windows-Setup durch das WindowsInstallation-Paket kommt es zum Fehler der in etwa "unable to locate an existing system partition" lautet! Was kann ich hier unternehmen?

Antwort: Falls es auf dem zu installierenden Gerät mehrere Festplatten und bereits eine Windows-Installation vorhanden ist und als System-Festplatte nun eine andere ausgewählt wurde - etwa durch den Einsatz der "PreferFastDisk"-Variable, dann kann es bei der Ausführung des Windows-Setup zu Problemen führen. Man kann hierbei entweder manuell über DISKPART oder über die versteckte Variable "ClearAllDisks" alle Festplatten beim Ausführen des DiskPartitioning-Paketes bereinigen. Hierzu muss man unter "Extras > Variablendefinitionen…" in EMC die Variable "ClearAllDisks" als Zahl zur Variablen "DiskPartitioning" hinzufügen und diese auf 1 setzen.

Frage: Wenn das Erstellen des WinPE Images fehlschlägt, und im Log des BTQH Dienstes eine Fehlermeldung zu finden ist, die ungefähr so lautet "Starten Sie Windows PowerShell mit der Option "Als Administrator ausführen", und versuchen Sie noch einmal, das Skript auszuführen" (Englisch: "Start Windows PowerShell with the Run as administrator option, and then try running the script again"). Was kann ich machen?

Antwort: Das Benutzerkonto, unter dem der BTQH-Dienst läuft, braucht administrative Rechte. Es ist sicherzustellen, dass dieser Benutzer Mitglied der Administratorengruppe ist.

Frage: Beim Erstellen eines Computer Self Provisioning fähigen WinPE Images gibt es einen Fehler. Im Log des BTQH Dienstes ist die Fehlermeldung "Error at validating Empirum API server settings" zu finden. Was kann ich machen?

Antwort: Die Fehlermeldung deutet darauf hin, dass die Einstellungen des Empirum-API-Dienstes nicht ausgelesen werden können. Stellen Sie sicher, dass der Empirum-API Dienst über Matrix42 DBUtil konfiguriert und installiert ist. Stellen Sie auch sicher, dass das HTTP Protokoll verwendet wird, denn nur diese Variante wird aktuell von Computer Self Provisioning unterstützt.

Frage: Wie kann ich die Standardsprache der WinPE Self Provisioning Benutzeroberfläche anpassen?

Antwort: In der Datei "*Matrix42SelfProvisioningUI.exe.config*" unter *.\Empirum\EmpInst\Sys\Images\WinPE\binaries\UAF* gibt es ganz unten den Wert <add key="DefaultLanguage" value="Deutsch"/>. Dieser Wert kann auf die gewünschte Sprache "Deutsch / English / Francais" eingestellt werden.

```
<appSettings>
  <add key="EmpirumApiServer" value="RWR-Master.qalab.matrix42.de"/>
  <add key="EmpirumApiPort" value="9200"/>
  <add key="EmpirumApiProtocol" value="Http"/>
  <add key="EmpirumApiSecured" value="False"/>
  <add key="EmpirumApiUsername" value="sa"/>
  <add key="EmpirumApiPassword" value="XA9B58E60F21E4211E9EF67F81323A2
  <add key="OpenConnectionRetries" value="10"/>
  <add key="DefaultLanguage" value="Deutsch"/>
</appSettings>
</configuration>
```

Frage: Kann ich den neu erstellten WinPE USB Stick manuell mit Daten erweitern?

Antwort: Da ab der WinPE Version 1.6.5 zwei Partitionen auf dem USB Stick erstellt werden, kann die zweite NTFS Partition manuell mit den Daten - Betriebssystemdateien, Sprachpakete und Treiber - aus dem Empirum Share befüllt werden.

> **WICHTIG**: Auf dem USB Stick befindet sich die gleiche Struktur wie in dem Installationsverzeichnis. Die hinzugefügten Daten müssen ebenfalls an der gleichen Stelle auf dem USB Stick hinterlegt werden, wie sie auf dem Installationsverzeichnis hinterlegt sind.

Frage: In der Boot-Konfiguration sind alle Kontrollelemente ausgegraut. Man kann keine Boot-Konfiguration erstellen oder ändern. Woran liegt das?

Antwort: Um Boot-Konfigurationen erstellen oder ändern zu können, benötigt der angemeldete Benutzer die Rolle *EMP_I_DISK_CONFIG*. Besitzt er sie nicht, sind die Kontrollelemente ausgegraut und er kann sich nur die bestehenden Konfigurationen anschauen. Um die Boot-Konfiguration ändern zu können, melden Sie sich mit einem anderen Benutzer an der Matrix42 Management Console an oder benutzen Sie die Benutzerverwaltung in Matrix42 DBUtil, um den angemeldeten Benutzer die Rolle *EMP_I_DISK_CONFIG* zuzuweisen.

Frage: Die Betriebssysteminstallation per HTTP(S) schlägt fehl, es wird der Fehler 403 angezeigt. Was kann ich machen?

Antwort: Wenn diese Fehlermeldung angezeigt wird, hat der angegebene Benutzer auf Verzeichnisstrukturen im IIS keine schreibende Rechte.

Siehe auch Kapitel *8.2 WinPE basiertes Deployment via http(s) einrichten ab Seite 74*.

Frage: Die Betriebssysteminstallation per HTTP(S) läuft ohne Probleme durch, allerdings sind im Matrix42.Platform.Service.Host.log mehrere Fehler 404 zu sehen. Woran liegt das?

Antwort: Aktuell erscheinen die Fehler 404 auch in der Log-Datei einer erfolgreichen Bespielung. Grund dafür ist ein generischer Ansatz zur Übertragung von Verzeichnissen und einzelnen Dateien. Soll eine einzelne Datei übertragen werden, wird diese zunächst wie ein Verzeichnis behandelt. Der Transport versucht zunächst den Inhalt dieses Verzeichnisses zu bestimmt. Dies führt zu dem Fehler 404, da ein entsprechendes Verzeichnis nicht vorhanden ist. Die Fehlermeldung ist dann folgende:

```
[Error] [JobProcessorBase.Execute] The remote server returned an error: (404) Not Found.
```

Auf diese Fehlermeldung folgt dann nochmals eine Warnung:

```
[Warning] [HttpTransfer.RunGenerateList] Issue at running list job. Message: The remote server returned an error: (404) Not Found.
```

Im Anschluss wird darauf hingewiesen, dass nun die Übertragung mit einer einzelnen Datei versucht wird:

```
[Information] [HttpTransfer.TransferFromServer] Could not list content of 'http://EmpirumServer.Matrix42.de/Matrix42-
```

```
Empirum/Configurator/Values\DeviceMapping.xml'.    =>    Try    to
transfer single file.
```

Diese Übertragung wird dann erfolgreich durchgeführt.

```
[Information]    [HttpTransfer.StartTransfer]    Final    status:
'Finished'
```

Frage: Die Betriebssysteminstallation per HTTP(S) schlägt beim Übertragen der DeviceMapping.xml fehl. Die Netzwerkkarte ist noch nicht ansprechbar. Gibt es eine Möglichkeit die Anzahl der Verbindungsversuche zu konfigurieren?

Antwort: Die Anzahl der Verbindungsversuche kann in der Konfigurationsdatei *Matrix42.Empirum.PeAgent.dll.config* des PE Agent festgelegt werden. Die Datei liegt im *EmpInst$\Sys\Images\WinPE\binaries\UAF*-Verzeichnis.

Der *OpenConnectionRetries*-Wert gibt die Anzahl der Verbindungsversuche an und steht ursprünglich bei 10 Versuchen.

```
<appSettings>
  <add key="DelayOnErrorInSeconds" value="300" />
  <add key="DelayRebootInSeconds" value="0" />
  <add key="EnableReboot" value="True" />
  <add key="UsePXEServerAsEmpirumServer" value="False" />
  <add key="PowerShellTimeoutInSeconds" value="3600" />
  <add key="UafHostWindowsServiceName" value="Matrix42 UAF Host for PE Agent" />
  <add key="OpenConnectionRetries" value="10" />
  <add key="WaitAfterInitialize" value="False" />
</appSettings>
```

Wird der Wert in der *Matrix42.Empirum.PeAgent.dll.config* geändert muss im Anschluss die verwendete Bootkonfiguration neu gespeichert werden, damit die Änderung in das PXE-Image übernommen wird.

Frage: Die Betriebssysteminstallation per HTTPS bricht mit der Fehlermeldung "Thumbprint list is empty at Https case." ab. Woran liegt das, und was kann ich machen?

Antwort: Der PE Agent hat versucht, eine Übertragung per Https aufzubauen. In dem verwendeten Agent Template ist die Option "*Nur vertrauenswürdigen Server Zertifikaten vertrauen*" ausgewählt. Da unter WinPE die übliche Validierung über den Zertifikatsspeicher und die Zertifikatsketten nicht funktioniert, wird der PE Agent die Validierung anhand der definierten Zertifikat-Thumbprints durchführen. Die Liste der Thumbprints die in der Konfigurationsdatei *Matrix42.Empirum.PeAgent.dll.config* (im *EmpInst$\Sys\Images\WinPE\binaries\UAF*-Verzeichnis) des PE Agent festgelegt ist, ist allerdings leer. Eine gesicherte Verbindung konnte somit nicht aufgebaut werden.

In Kapitel *8.2 WinPE basiertes Deployment via http(s) einrichten* ab Seite *74* wird genauer beschrieben, wie man die Thumbprint-Information mit Hilfe des *IIS Manger* erhält und wie man sie in die Konfigurationsdatei eintragen muss. Im Anschluss muss die verwendete Bootkonfiguration erneut abgespeichert werden, damit die Änderung in das PXE-Image übernommen wird.

Ist eine Überprüfung des Server-Thumbprints nicht gewünscht, dann muss im verwendeten Agent Template die Option "*Nur vertrauenswürdigen Server Zertifikaten vertrauen*" abgewählt werden. Eine Verbindungsaufbau kann dann auch ohne Thumbprints stattfinden.

Matrix42 AG
Elbinger Straße 7
60487 Frankfurt am Main
Deutschland
Tel.: +49 (0)69 6677 38 380
Fax: +49 (0)69 6677 88 657
E-Mail: info@matrix42.de
Web: https://www.matrix42.com/de/