

Microsoft Deployment Toolkit
- installieren und konfigurieren
von Holger Voges



© 2017 by Holger Voges, Netz-Weise IT Training

Version 1.0

Freundallee 13 a
30173 Hannover
www.netz-weise.de

Inhalt

| | |
|---|----|
| Einführung | 4 |
| Installation des MDT..... | 4 |
| Deployment Shares | 11 |
| Hinzufügen von Betriebssystemen, Treibern und Anwendungen | 15 |
| Betriebssysteme | 15 |
| Treiber | 19 |
| Anwendungen und Konfigurationskripte | 25 |
| Erstellen einer einfachen Task-Sequence..... | 29 |
| Ein bootfähiges Windows PE erstellen..... | 37 |
| Multicast-Unterstützung | 38 |
| Berechtigungen vergeben | 44 |
| Windows PE per WDS aus dem Netzwerk booten | 45 |
| Installieren eines Clients..... | 52 |
| Anpassen von Deploymentshare und Windows PE | 59 |
| Bootstrap.ini | 59 |
| CustomSettings.ini..... | 60 |
| Software und Konfigurationskripte bereitstellen | 61 |
| Über den Autor..... | 66 |

Einführung

Das Microsoft Deployment Toolkit, kurz MDT, ist ein einfaches, aber sehr wirkungsvolles Werkzeug, um Windows-Computer (Clients und Server) zu installieren. Man spricht bei MDT von einem Lite-Touch-Deployment (LTD), da die Installation nicht 100% automatisiert ist, sondern der Installationsvorgang im Gegensatz z.B. zum System Center Configuration Manager (SCCM) immer manuell gestartet werden muss.

Im Prinzip besteht das MDT aus zwei Komponenten, nämlich einer Freigabe, dem sogenannten Deployment Share, in dem sich die zu installierende Software befindet, und einem bootfähigen Client auf Basis von Windows PE, der die neuen Computer bootet und den Installationsprozess startet. Es gibt beim MDT keine Serverkomponente! Das MDT kann also auch auf einem Windows Client installiert werden, und das Deploymentshare kann sogar auf einen Linux-Rechner verschoben werden.

Der Installationsprozess wird dabei durch Visual Basic Script automatisiert. Mit Hilfe von sogenannten Task-Sequences wird vorgegeben, in welcher Reihenfolge die Skripte gestartet und welche Optionen den Skripten mitgegeben werden. Die Task-Sequence wird vor der Installation mit Hilfe der Deployment Workbench, dem Verwaltungswerkzeug des MDT, erstellt.

Um den Installationsprozess komfortabler zu gestalten, kann man den Windows PE Client aus dem Netzwerk per PXE booten, anstatt auf Boot-DVDs oder USB-Sticks zurückzugreifen. Will man diese Option nutzen, benötigt man zusätzlich einen DHCP-Server und einen WDS (Windows Deployment-Service). Beide Dienste benötigen einen Windows Server.

Das MDT ist kostenlos und wird seit 2004 bei Microsoft ständig weiterentwickelt.

Installation des MDT

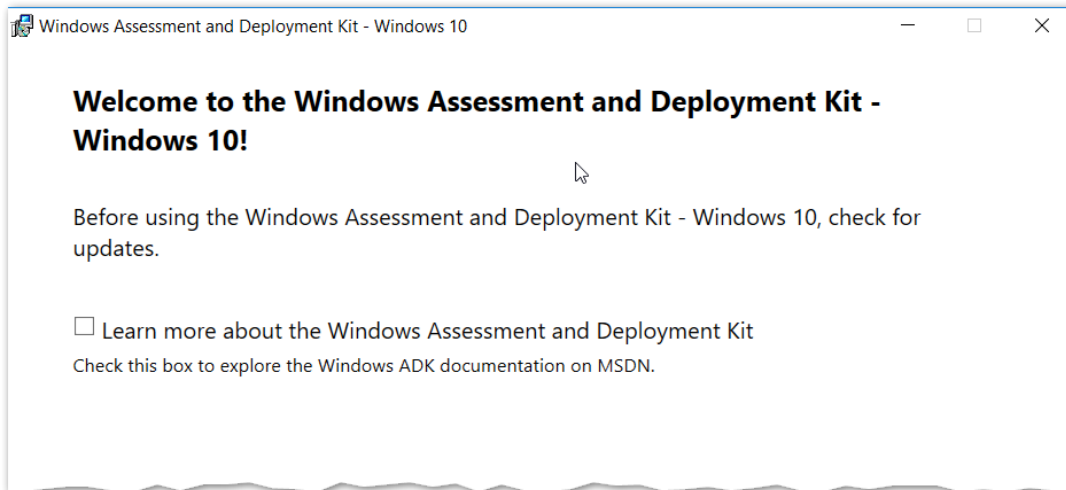
Bevor das MDT (derzeit MDT 2013, Update 2, Version 8443, Stand: Juni 2017) installiert werden kann, muss zuerst das Windows Assessment and Deployment Kit (ADK) installiert sein, das man ebenfalls kostenlos bei Microsoft herunterladen kann. Das ADK besteht aus einer ganzen Reihe von Tools, von denen MDT allerdings nur drei benötigt:

- Die Windows Deployment Tools (zur Bearbeitung von WIM-Images)
- Windows PE (zum Starten der Clients)
- das User State Migration Tool (USMT) für Migrationsszenarien

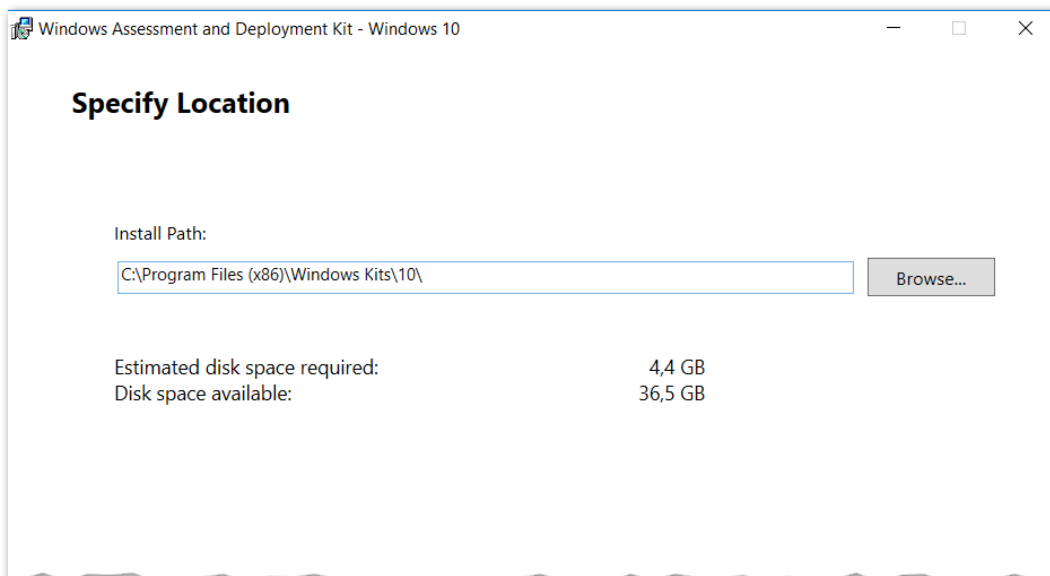
Achten Sie beim Herunterladen darauf, dass Sie jeweils das aktuellste ADK verwenden, da es mit jedem Feature-Release von Windows 10 ebenfalls aktualisiert wird. Sie finden es derzeit unter

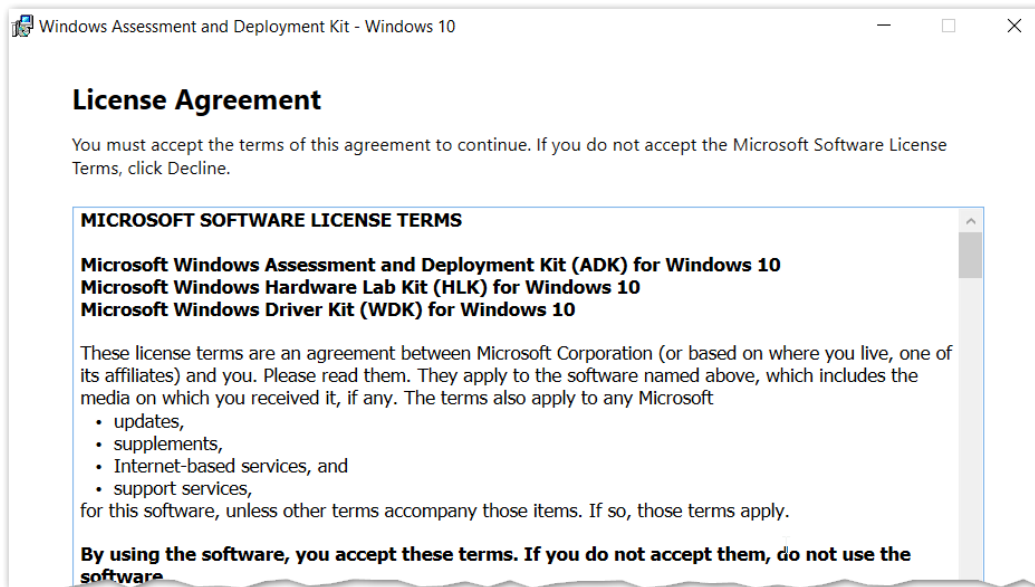
<https://developer.microsoft.com/de-de/windows/hardware/windows-assessment-deployment-kit>

Wenn Sie das Setup herunterladen, bekommen Sie nur eine kleine Setup-Datei. Diese kann die ADK-Tools dann nach Bedarf aus dem Internet herunterladen, oder das komplette ADK (ca. 4GB) lokal speichern. Im folgenden Beispiel wurde das ADK zuerst lokal heruntergeladen.

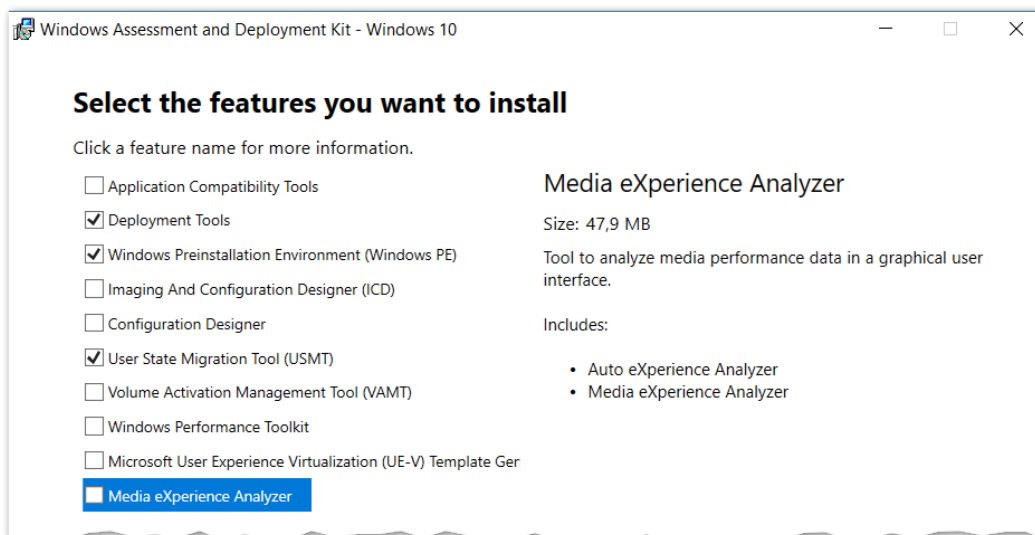


Geben Sie einen Zielpfad ein. Das ADK wird im %ProgramFiles (x86)%\Windows-Kits\ Ordner bereitgestellt.

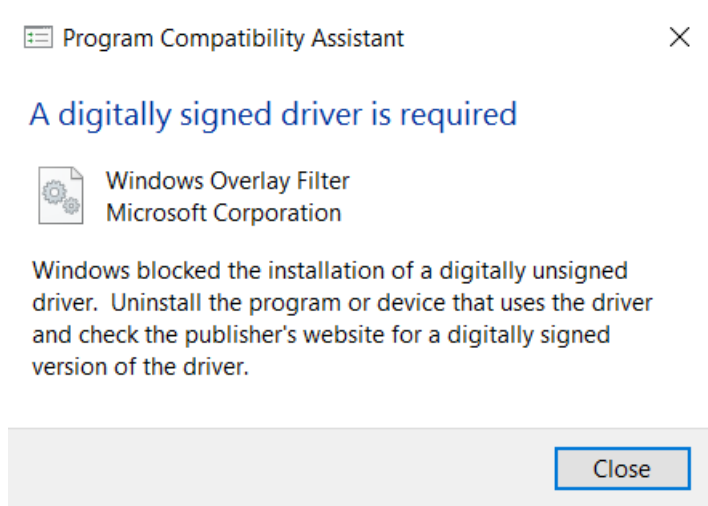




Wählen Sie in der Featureliste „Deployment Tools“, „Windows Preinstallation Environment“ und „User State Migration Tool“ aus.



Wenn Sie das ADK 1703 verwenden, erhalten Sie vermutlich unten abgebildete Fehlermeldung. Der Fehler kann nachträglich behoben werden, indem Sie statt des leider von Microsoft falsch signierten Treibers den Original-Betriebssystemtreiber verwenden.



Das Problem rührt daher, dass Microsoft das ADK mit einem nicht korrekt signierten Treiber ausgeliefert hat, der vom Betriebssystem aber in einer voll funktionsfähigen Version eh mitgeliefert wird. Laden Sie einfach den Windows-Treiber, indem Sie den Treiber in der Registry mit folgendem Powershell-Kommando nachladen:

Führen Sie dazu einfach folgendes Script aus (Der Code besteht aus einer Zeile):

```
Set-ItemProperty -Path  
Registry::HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WIMMount -Name  
ImagePath -Value system32\drivers\wimmount.sys
```

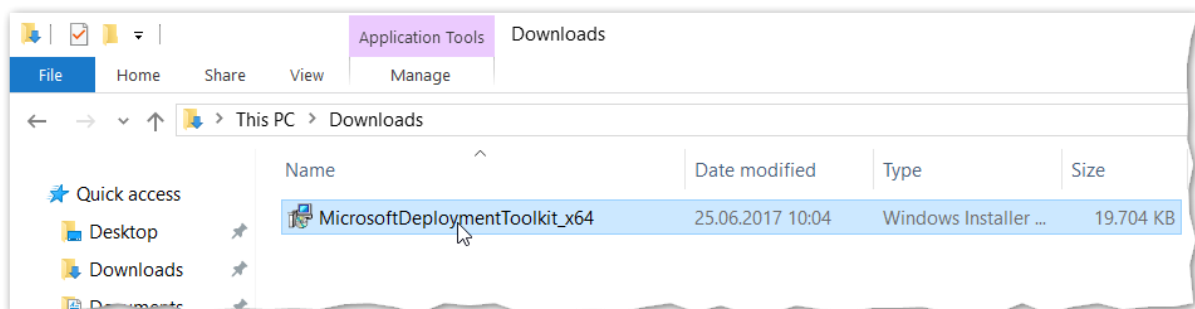
Die Beschreibung des Fehlers finden Sie im Michael Niehaus Blog:

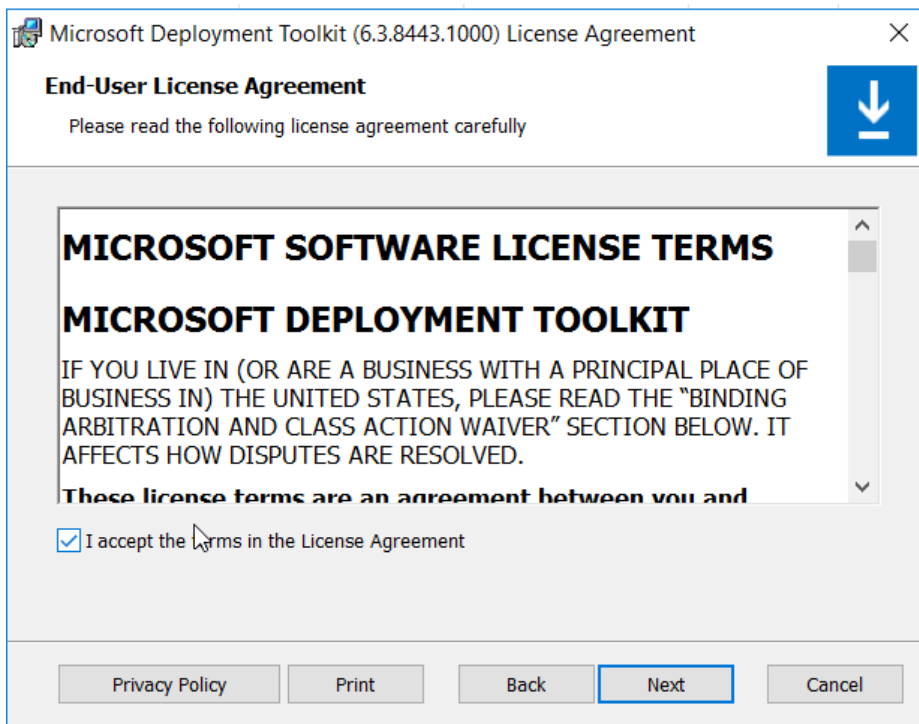
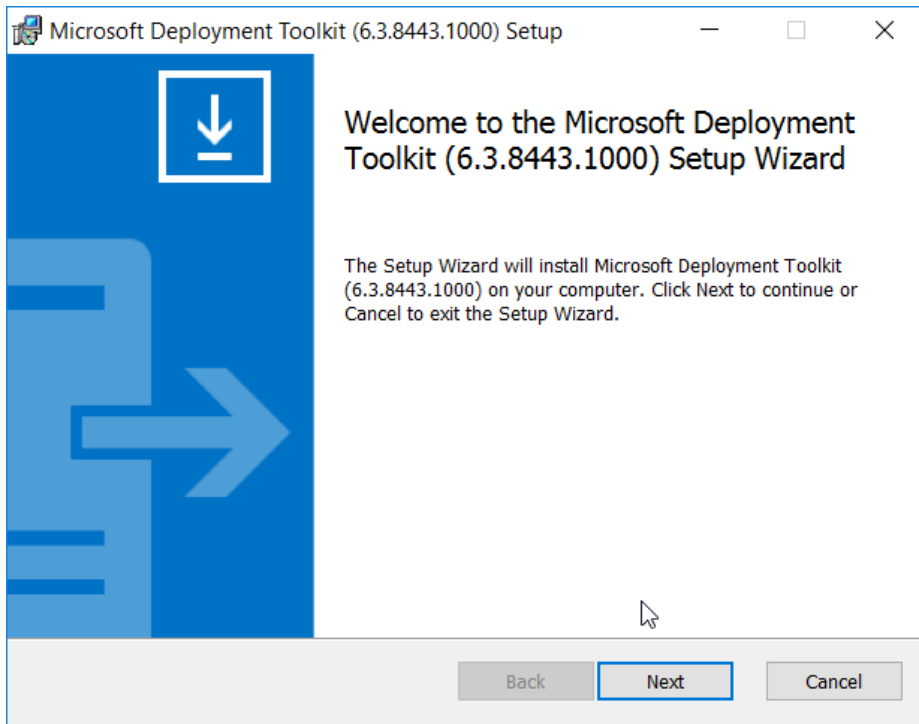
<https://blogs.technet.microsoft.com/mniehaus/2017/05/16/quick-workaround-for-adk-1703-issue/>

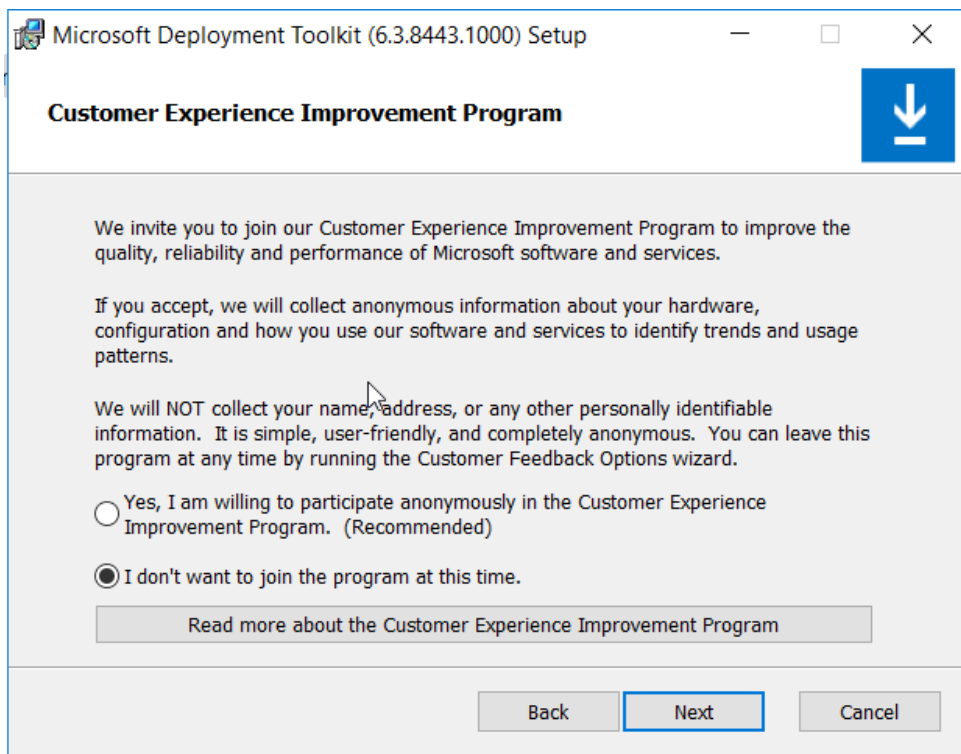
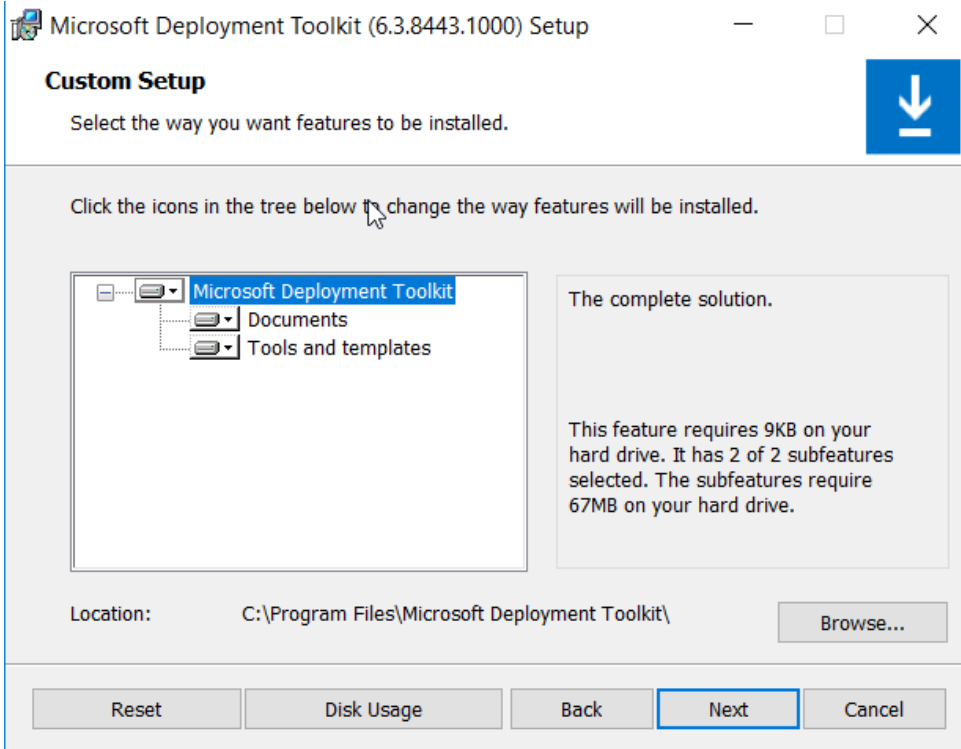
Nach Abschluss der Installation des ADK kann das MDT installiert werden. Auch hier gibt es regelmäßig Updates. Die neuste Version finden Sie, wenn Sie im MDT Team Blog nachschauen.

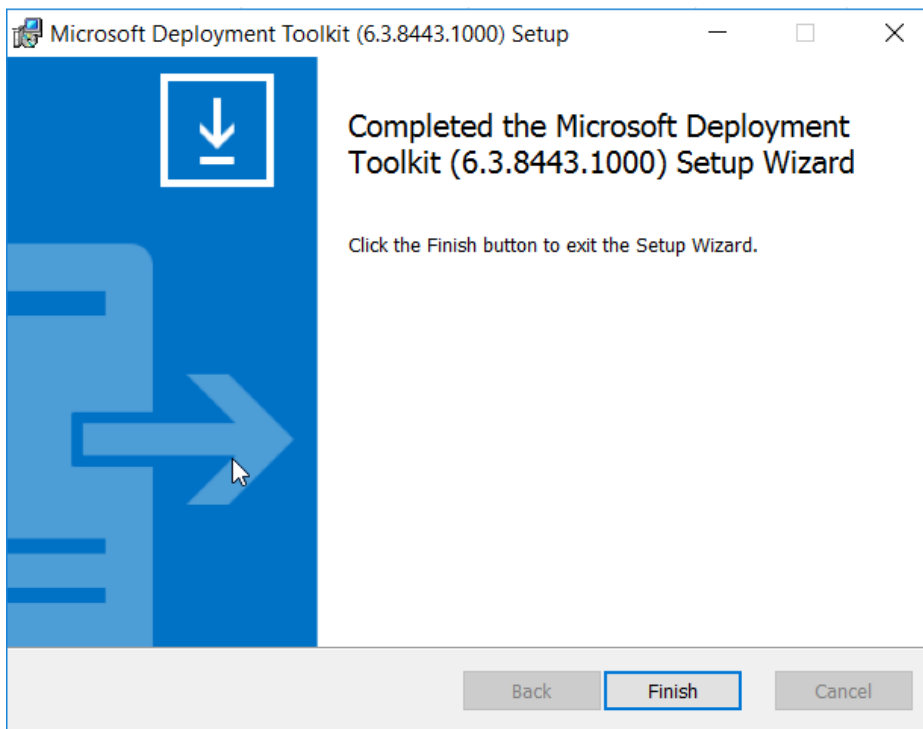
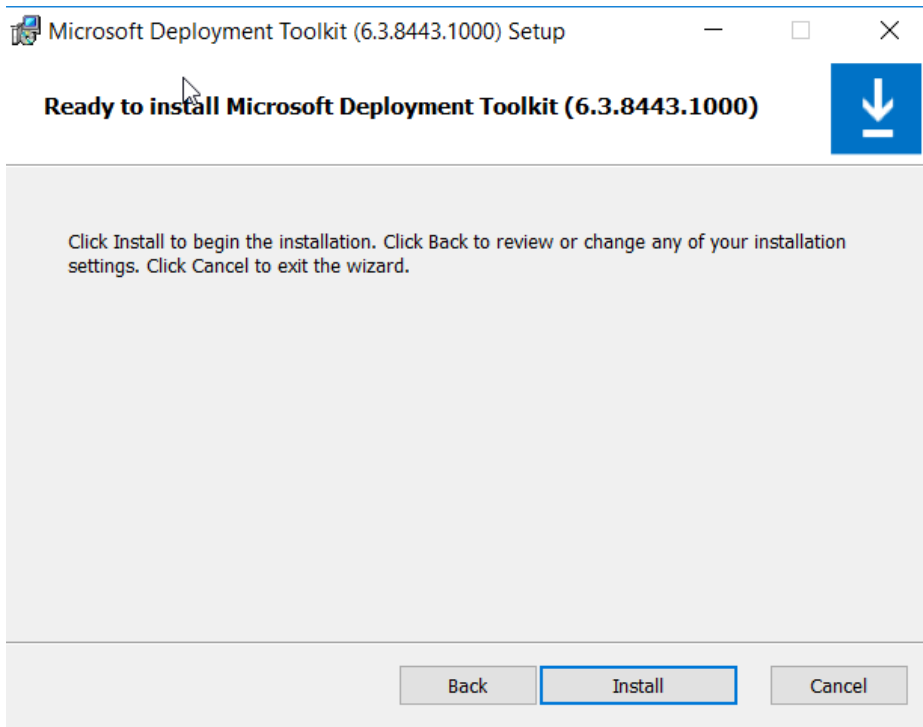
<https://blogs.technet.microsoft.com/msdeployment/>

Verwenden Sie auf einem 64-Bit System auch die 64-Bit Version. Die Installation ist sehr simpel. Klicken Sie sich einfach durch die Installation, indem Sie alles bestätigen.





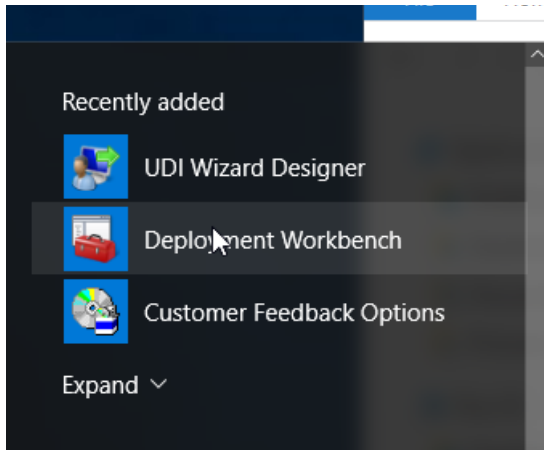




Damit ist die Installation des MDT abgeschlossen.

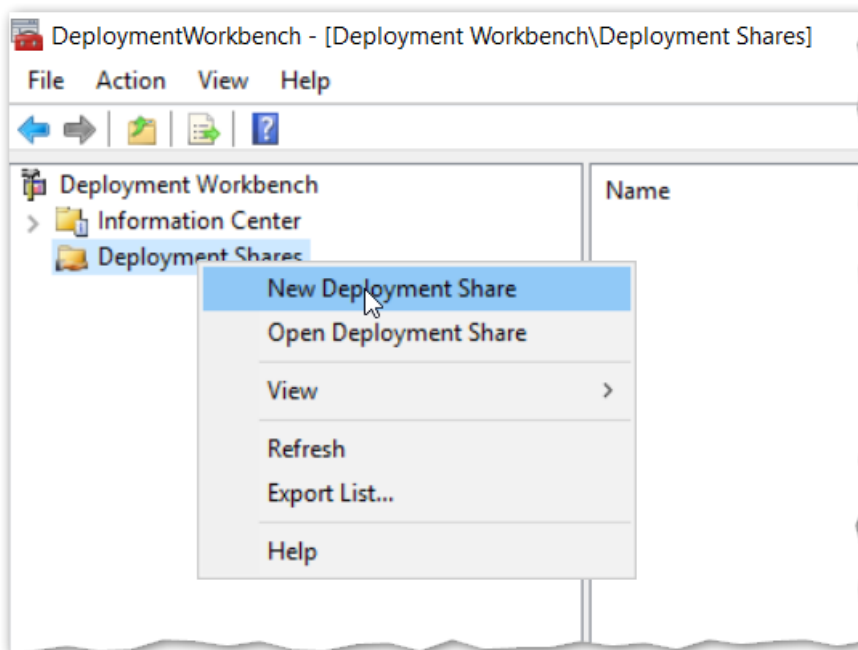
Deployment Shares

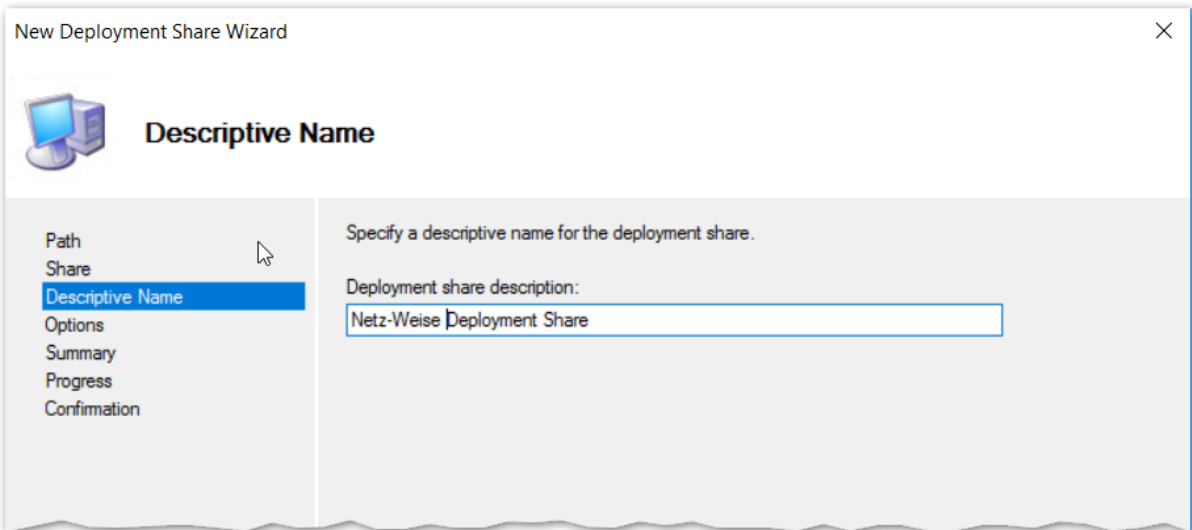
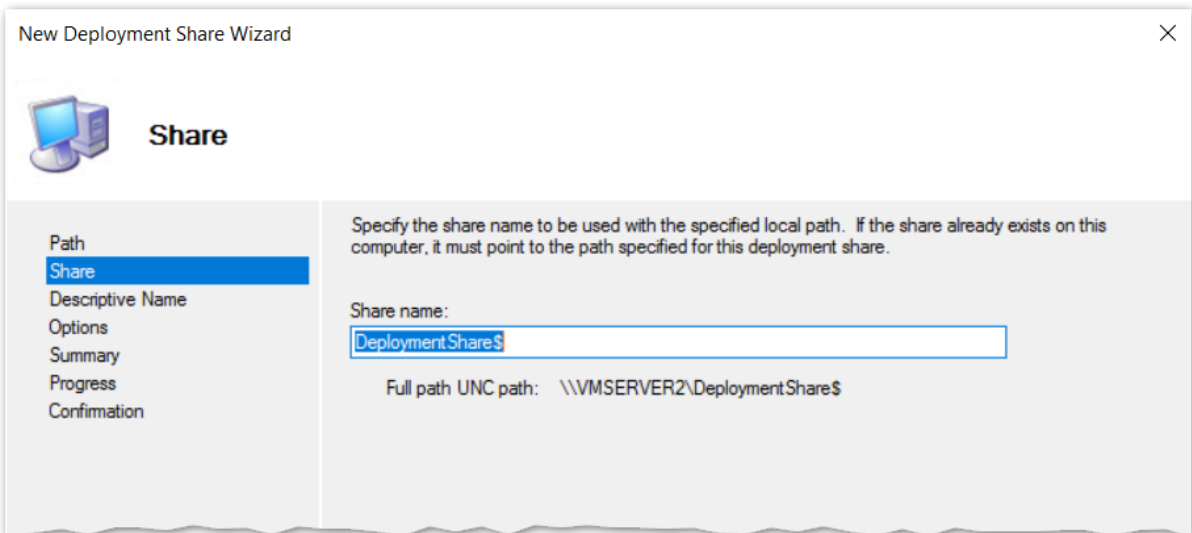
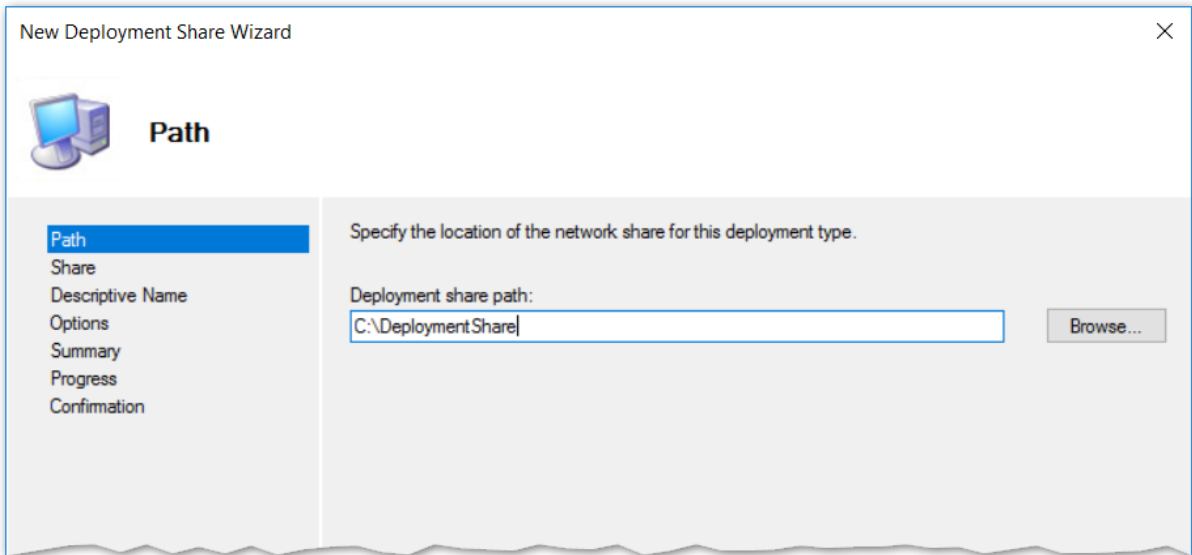
Als DeploymentShare wird ein Ordner bezeichnet, der alle zur Installation notwendigen Daten zur Verfügung stellt. Er besitzt eine bestimmte Dateistruktur und beinhaltet alle zur Installation notwendigen Skripte. Der Ordner ist standardmäßig unter dem Namen Deploymentshare\$ freigegeben. Um ein neues Deploymentshare anzulegen, verwenden Sie die Deployment-Workbench. Sie ist Hauptbestandteil des MDT.



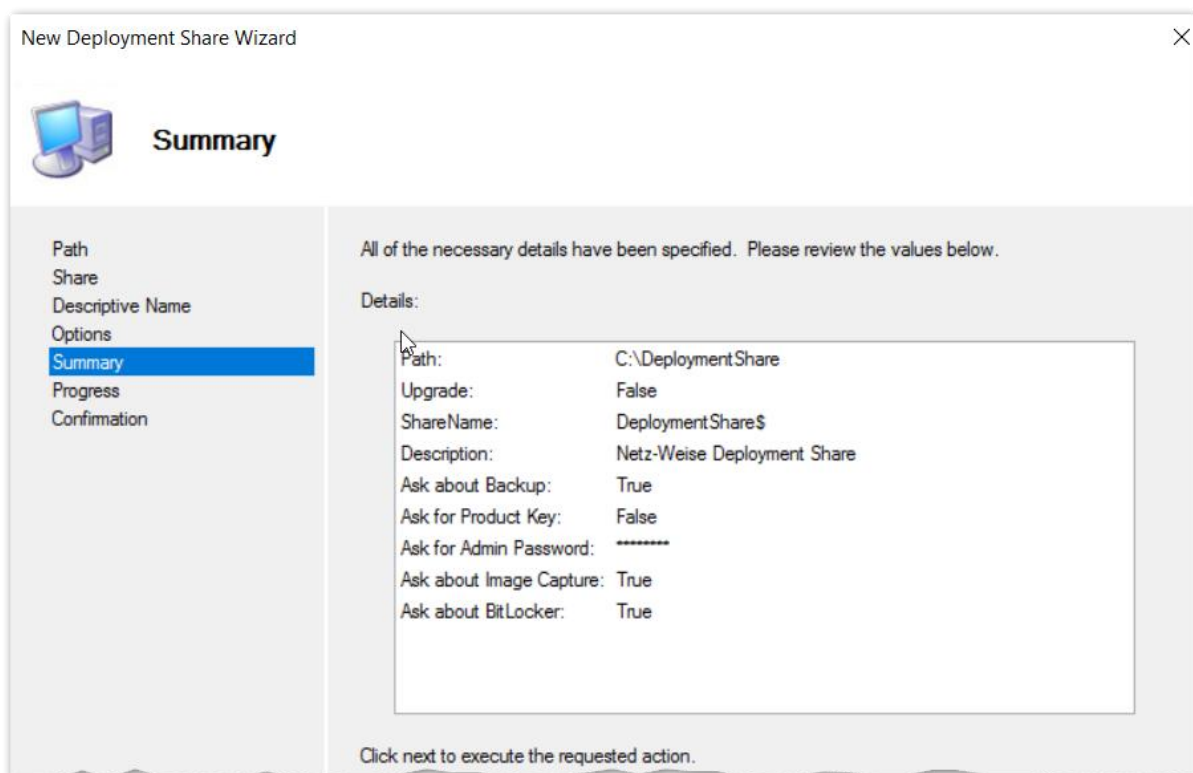
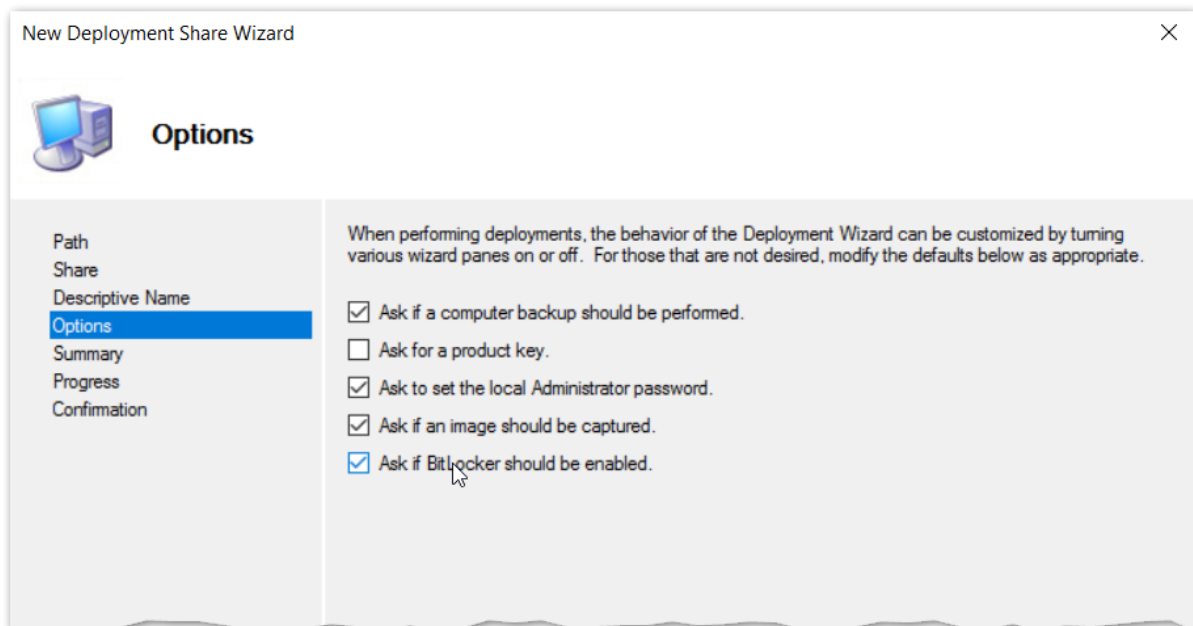
Wenn Sie die Deployment-Workbench zum ersten Mal starten, finden Sie nur zwei Knoten, Information-Center mit Internet-Links zur Dokumentation und „Deployment Shares“. Legen Sie über das Kontextmenü von „Deployment Shares“ ein neues Deployment-Share an.

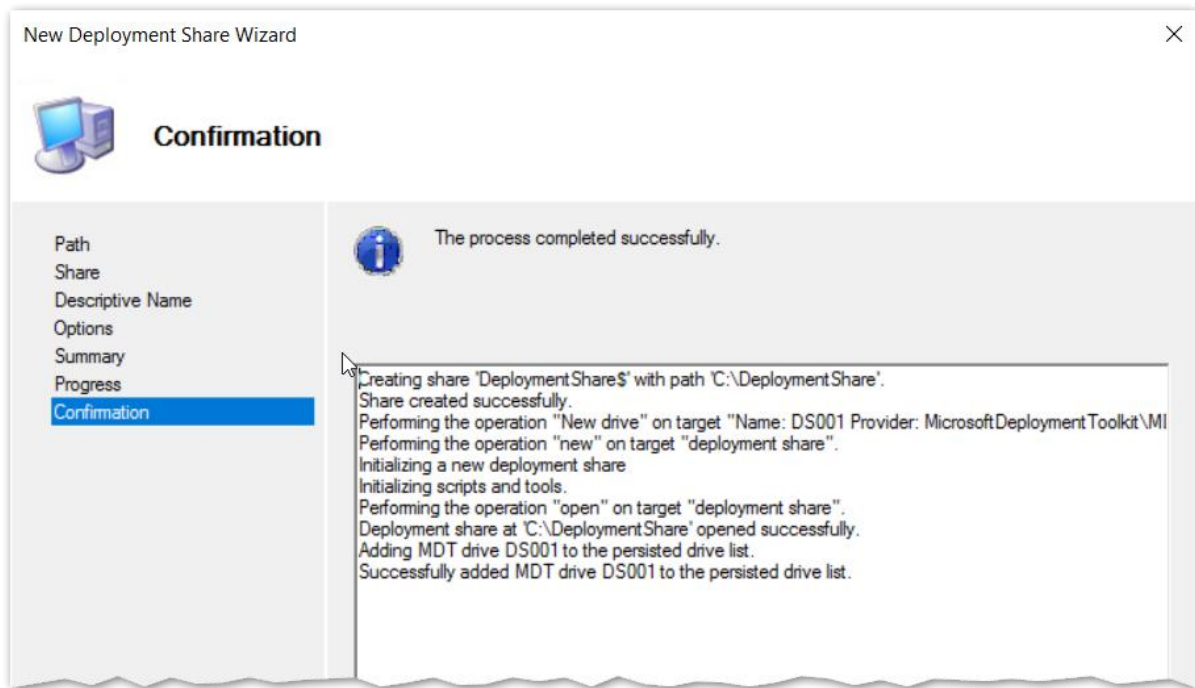
Geben Sie nun den Pfad zum Deployment-Share und den Namen an. Das \$-Symbol hinter dem Freigabenamen sorgt dafür, dass die Freigabe im Netzwerk nicht angezeigt wird.





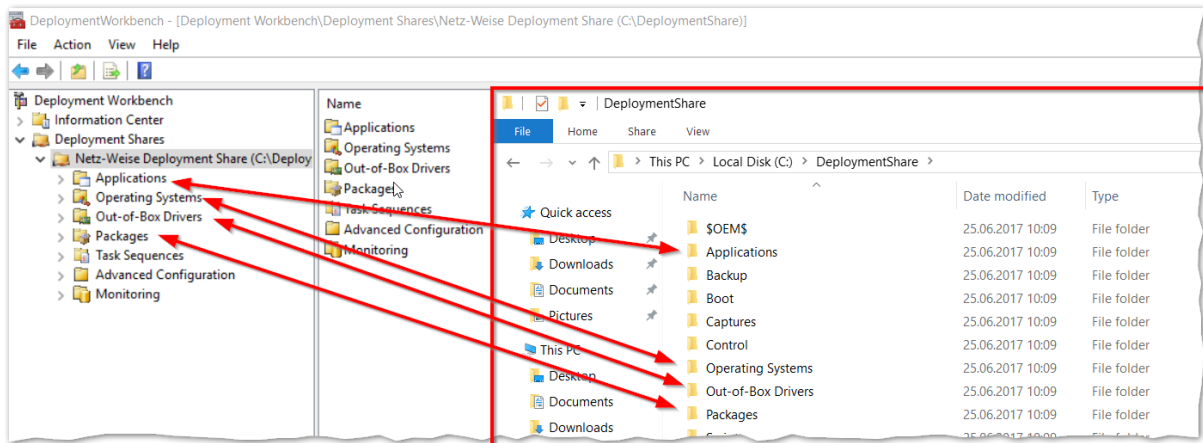
Die Konfigurationen, die Sie unter Options vornehmen, legen fest, welche Abfrage-Fenster während der Client-Installationen angezeigt werden sollen. Sie können diese Auswahl später jederzeit ändern. Sie können die Auswirkungen dieser Einstellungen sofort sehen, wenn Sie in das Kapitel Client-Installation schauen.





Nachdem Sie das Deployment-Share erstellt haben, wird unterhalb des Menü-Eintrags Deployment-Shares in der Workbench eine neue Ordnerstruktur angezeigt.

Die Ordnerstruktur aus der Workbench finden Sie in ähnlicher Form auch im Dateisystem wieder.

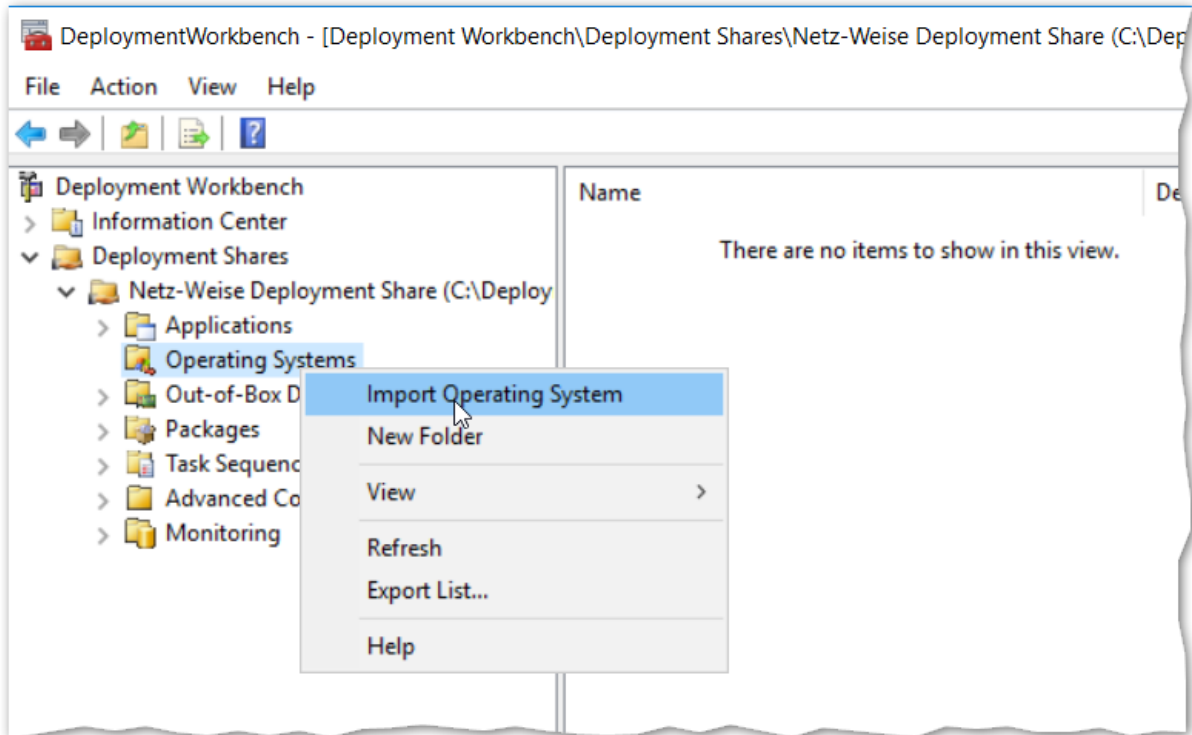


Hinzufügen von Betriebssystemen, Treibern und Anwendungen

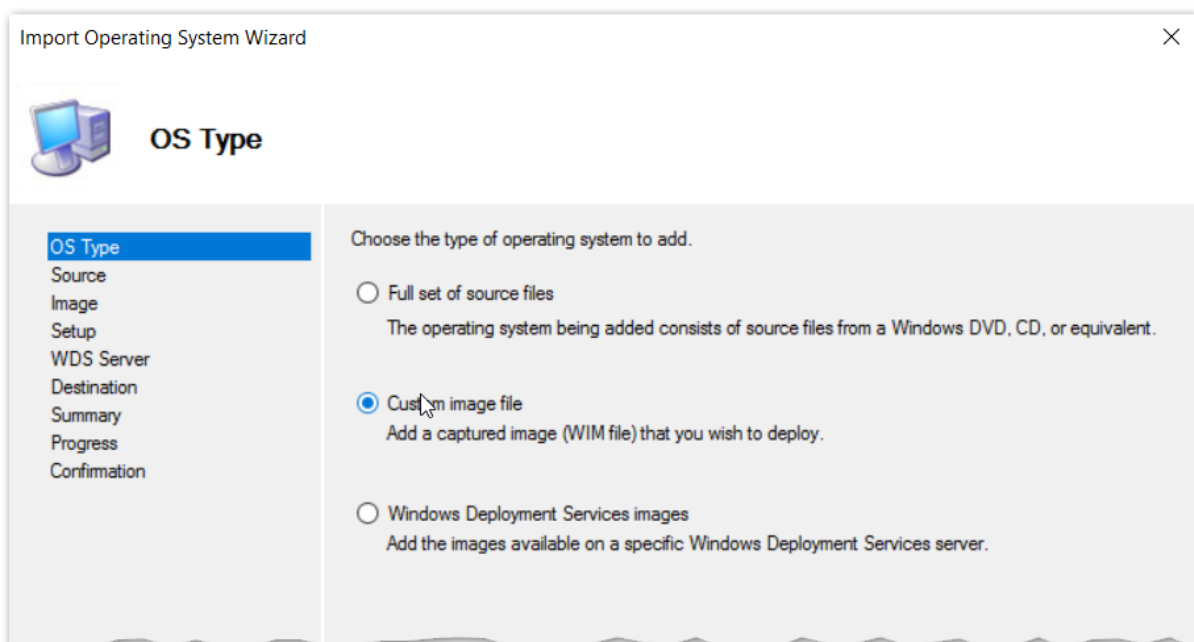
Als nächstes müssen die Installationsdaten dem Deployment-Share hinzugefügt werden.

Betriebssysteme

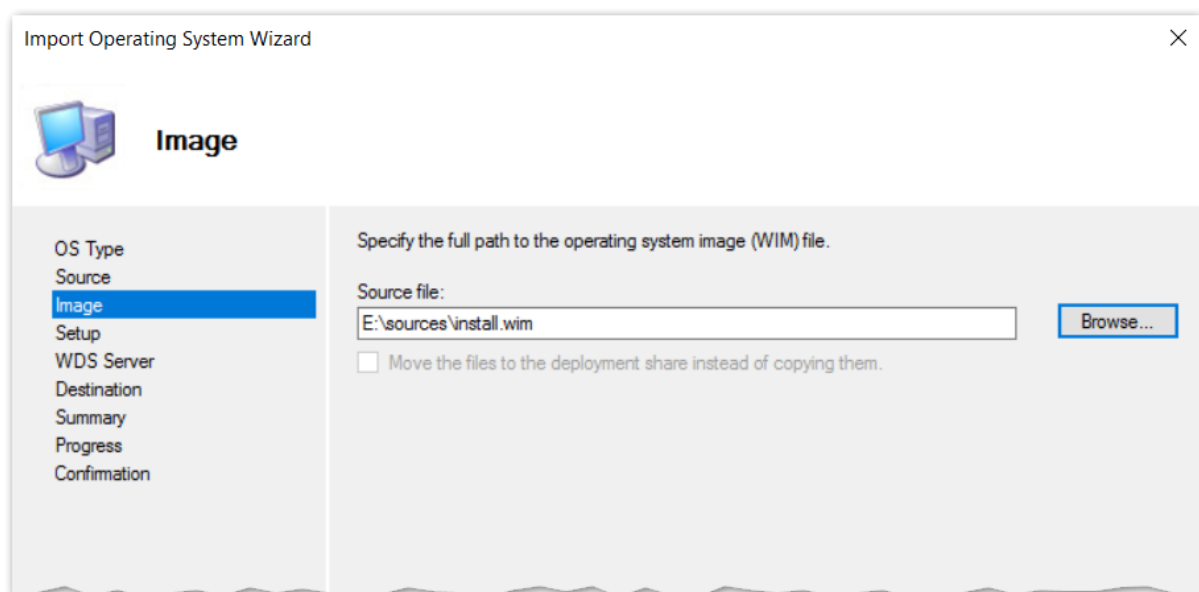
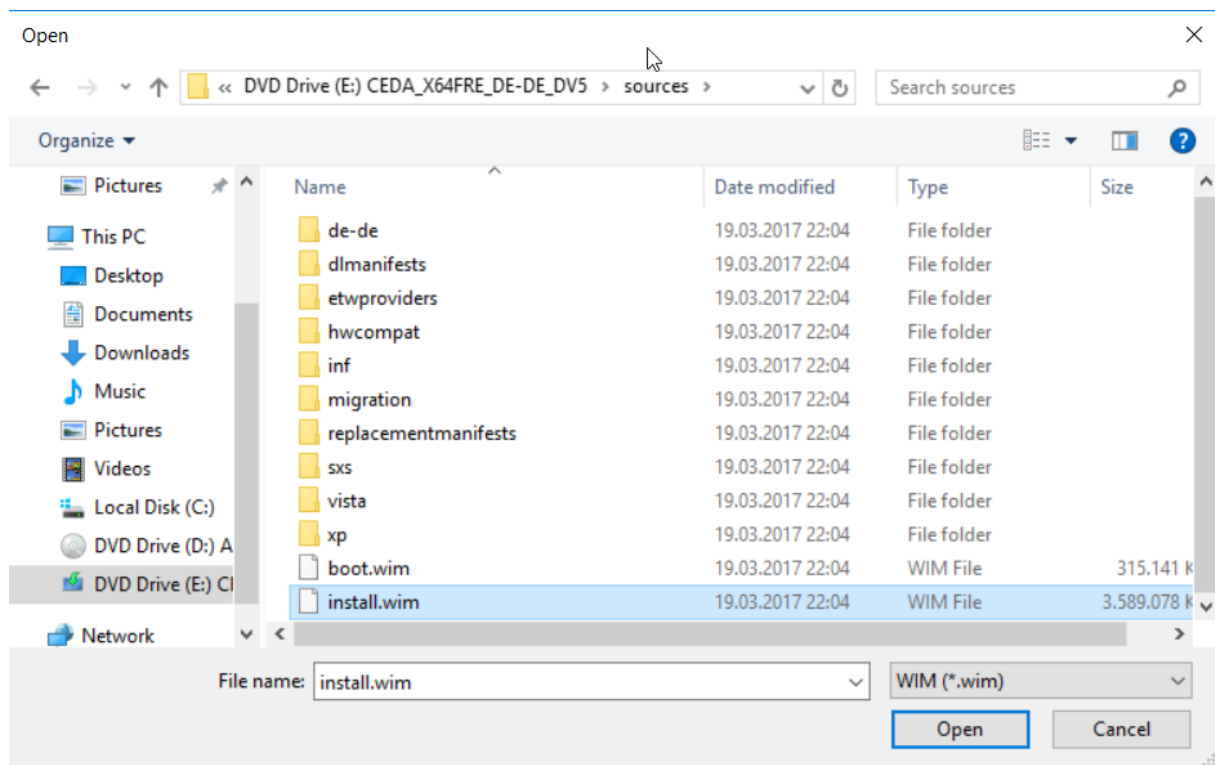
Beginnen Sie mit dem Betriebssystem, indem Sie den Eintrag Operating Systems auswählen und „Import Operating System aus dem Kontextmenü wählen.



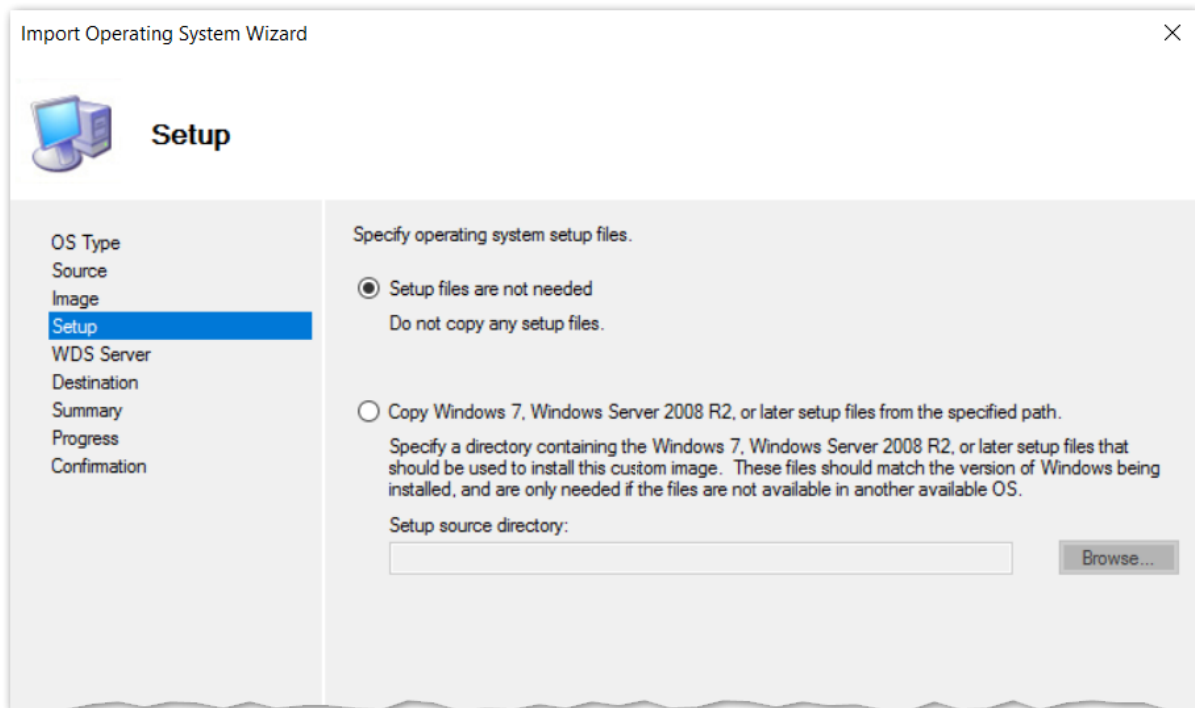
Sie können eine komplette DVD importieren, oder nur ein WIM (Windows-Image)-File. Da das MDT für die Installation nur ein WIM-File benötigt, können Sie auch dann Custom Image File wählen, wenn Sie eine DVD haben. Auch der Eintrag „Windows Deployment Services Image“ importiert nur ein WIM-File, in diesem Fall dann aber direkt aus einem laufenden WDS-Server.



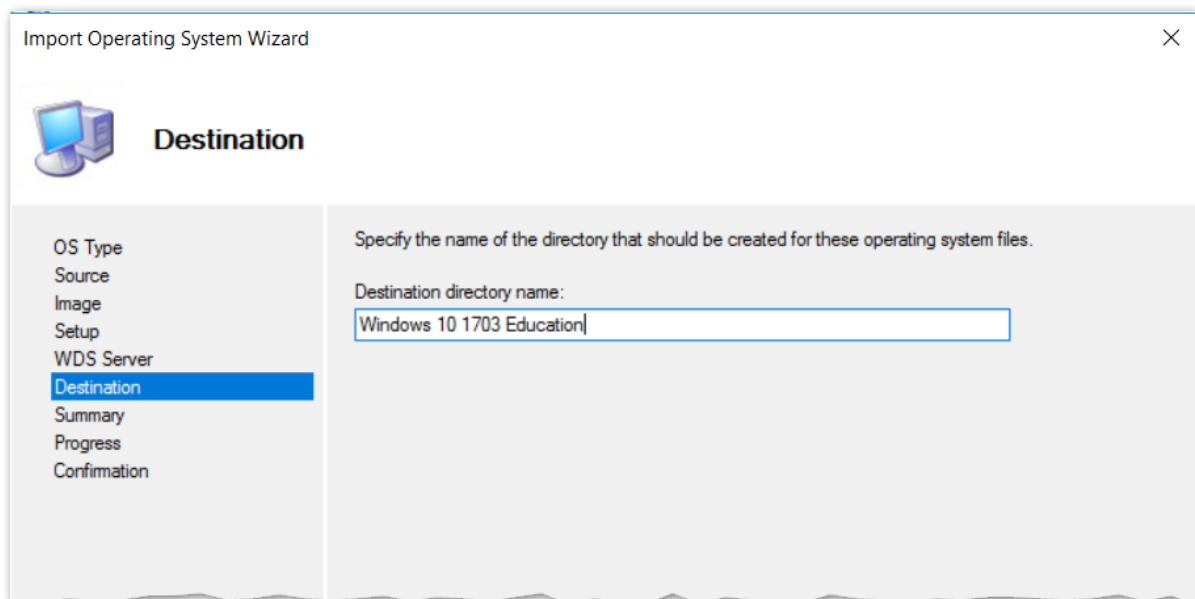
Geben Sie nun den Pfad zur Install.wim auf Ihrem Installationsmedium an, oder importieren Sie ein angepasstes WIM-File.



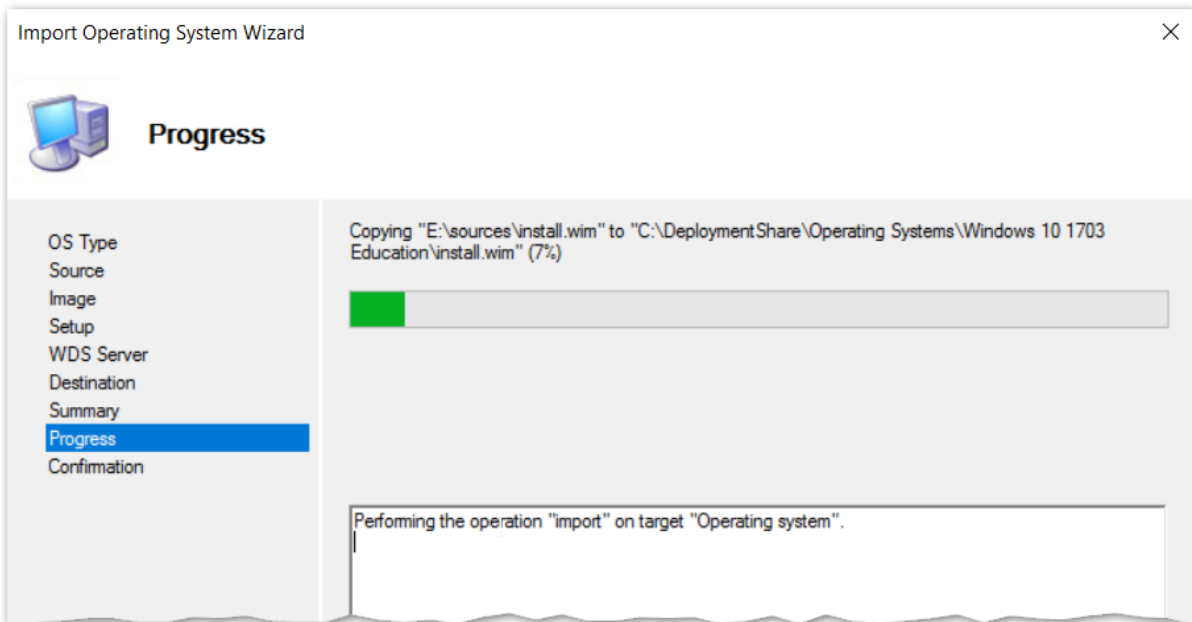
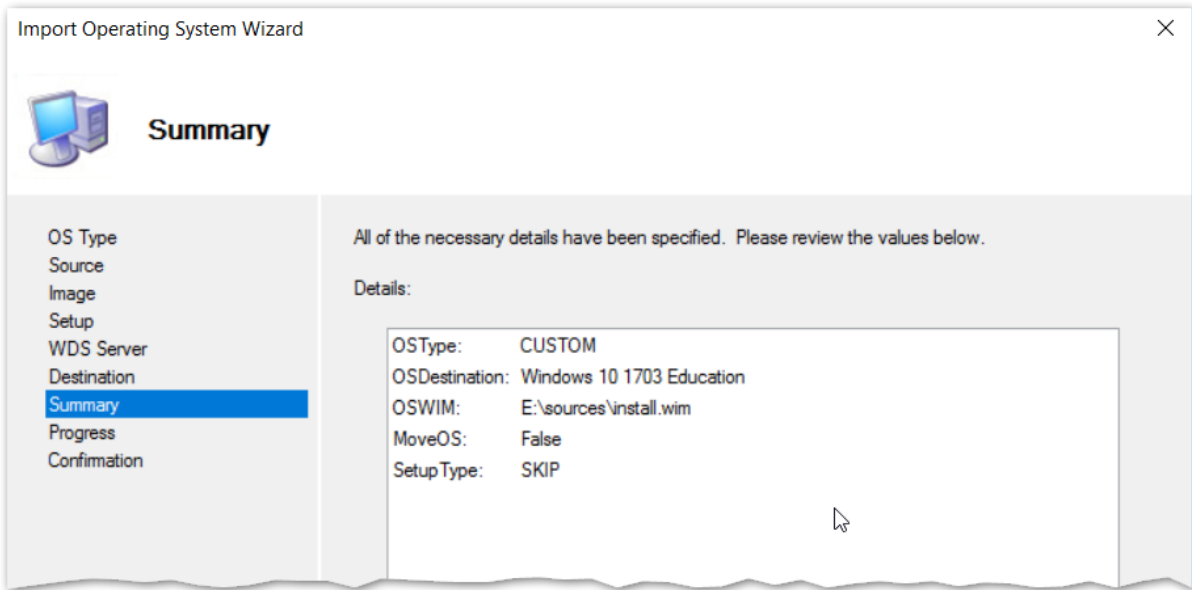
Sie benötigen keine Setup-Files, da Windows PE die Einrichtung der Datenträger für Sie vornimmt und dann das WIM-File entpackt. Sie können daher immer „setup Files not needed“ auswählen.



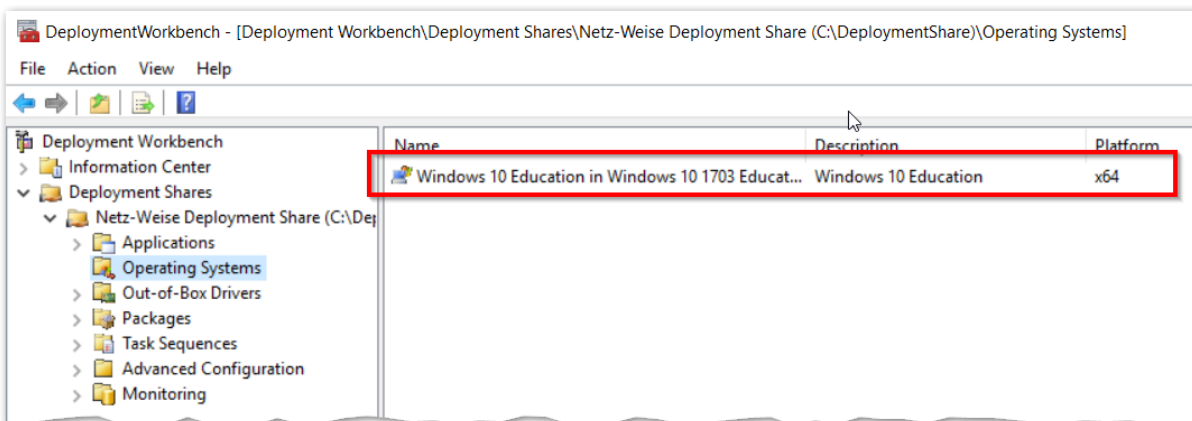
Geben Sie einen Namen für das zu importierende Betriebssystem ein. Unter diesem Namen wird im Dateisystem ein Ordner angelegt, in den die Install.wim kopiert wird.



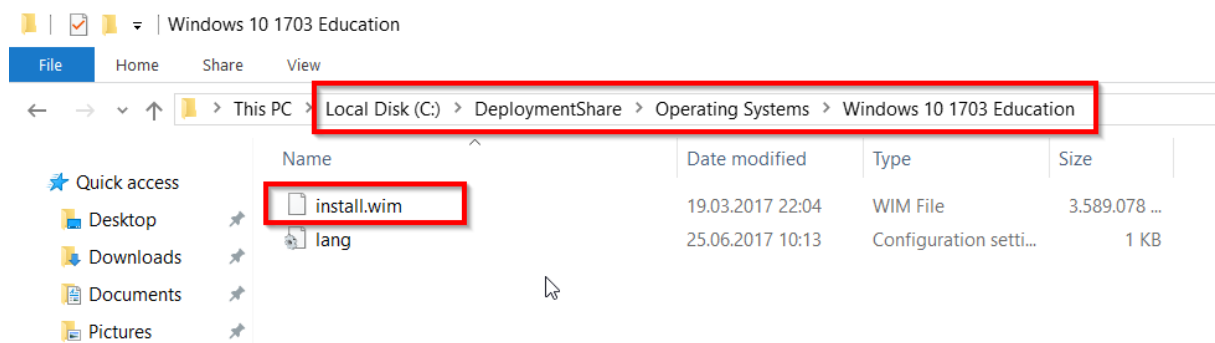
Bestätigen Sie die Zusammenfassung. Die Deployment Workbench importiert dann das Image-File.



Nachdem der Import abgeschlossen ist, werden alle im WIM-File enthaltenen Betriebssysteme angezeigt – im Beispiel war es nur Windows 10 Version 1703 Education.



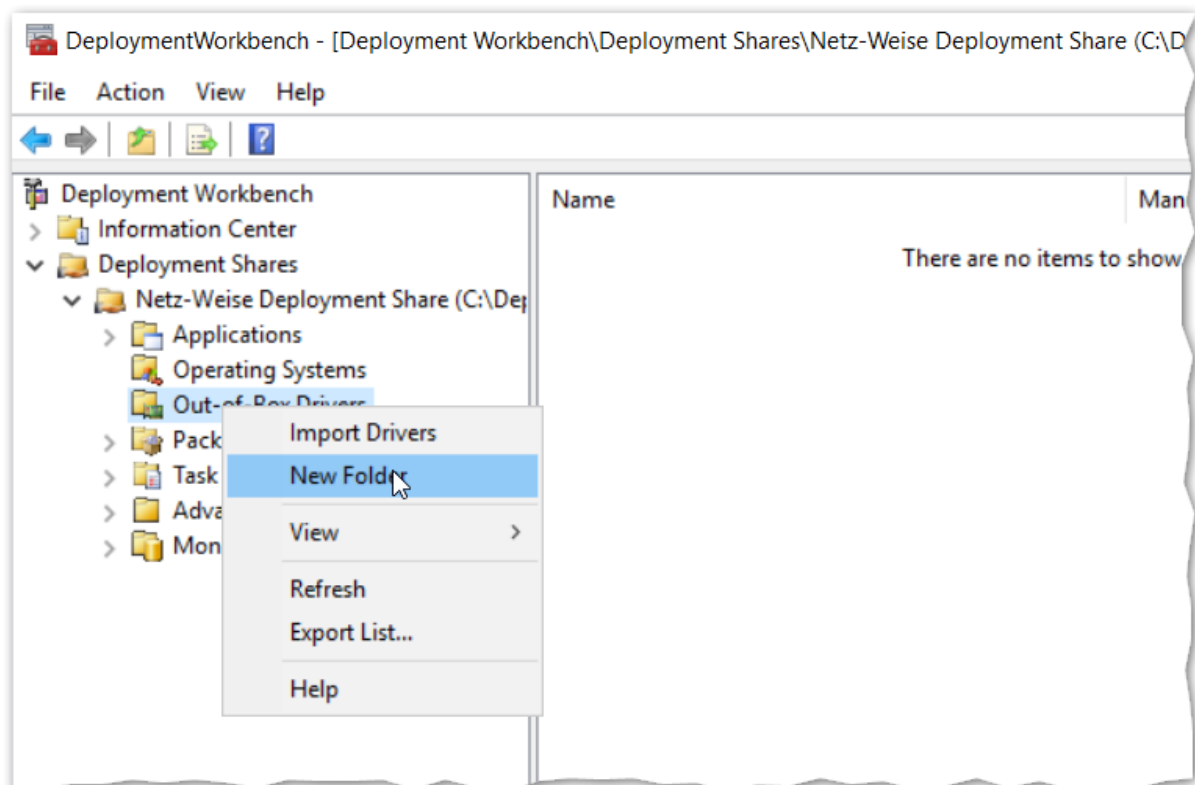
Das WIM-File wurde in die Dateisystemstruktur des MDT kopiert, und zwar in den Ordernamen, den wir oben angegeben haben.



Treiber


Als nächstes importieren Sie alle Treiber, die Sie benötigen. Hierfür wird eine Ordner-Struktur angelegt, die es erlaubt, anhand des Modells nur die jeweils notwendigen Treiber bei der Installation hinzuzufügen. Anschließend wird ein Selection-Profile erstellt, das für die Auswahl der notwendigen Treiber benötigt wird.

Erstellen Sie zuerst in „out-of-box-driver“ einen neuen Ordner. Das können Sie zur Sortierung von Daten auch für das Betriebssystem, Packages und Applications machen – dieser Schritt wurde beim Betriebssystem (s.o.) jedoch ausgelassen. Ordner können auch nachträglich jederzeit angelegt werden.



Geben Sie dem Ordner einen Namen. Je nachdem, wie viele unterschiedliche Geräte Sie unterstützen, macht es Sinn, eine mehrstufige Hierarchie zu erstellen, z.B. nach Hersteller und dann nach Modell. Im Beispiel wird nur ein Modellordner erzeugt.

New Folder ×

 **General Settings**


General Settings
Summary
Progress
Confirmation

Specify general information about this folder.

Folder name:

Folder comments:

New Folder ×

 **Summary**


General Settings
Summary
Progress
Confirmation

All of the necessary details have been specified. Please review the values below.


Details:

Folder Name: Lenovo M93p
Folder Comments: Lenovo M93p Drivers

New Folder ×

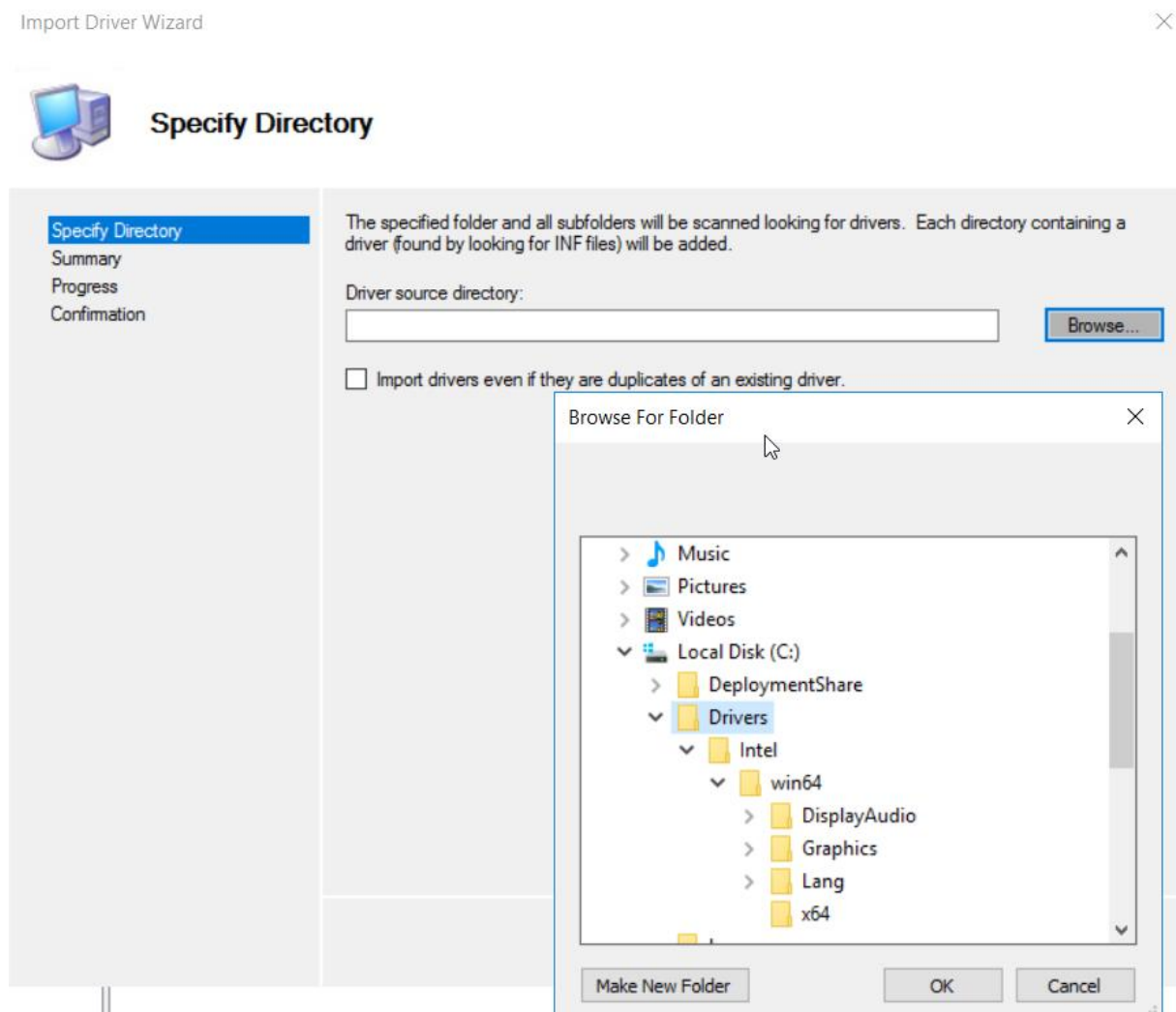
 **Confirmation**

General Settings
Summary
Progress
Confirmation

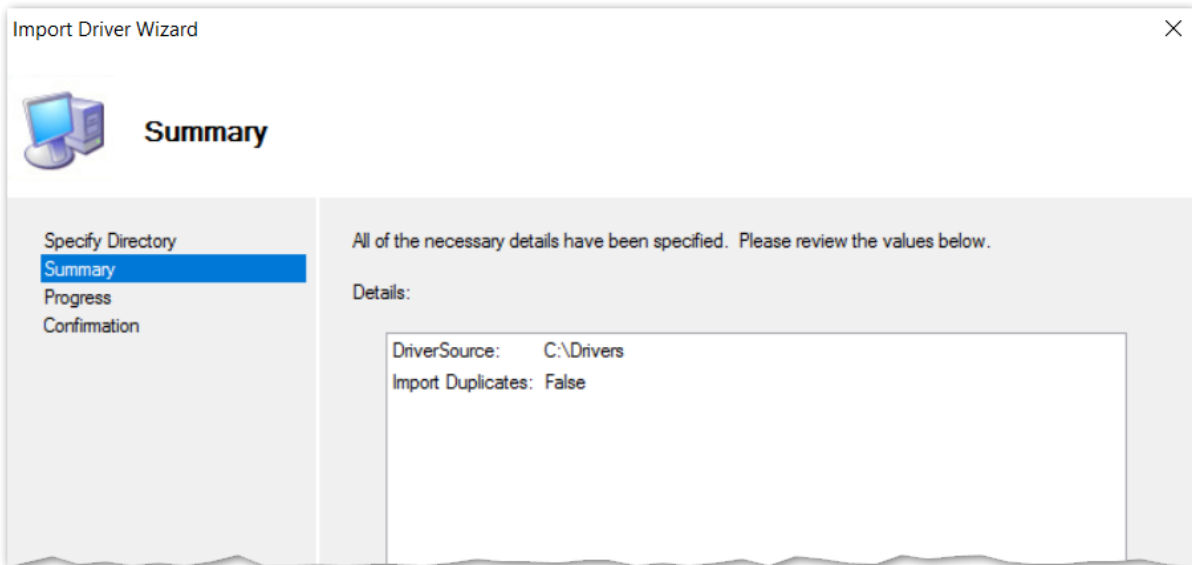
 The process completed successfully.

Performing the operation "new item" on target "C:\DeploymentShare\Out-of-Box Drivers\Lenovo M93p".

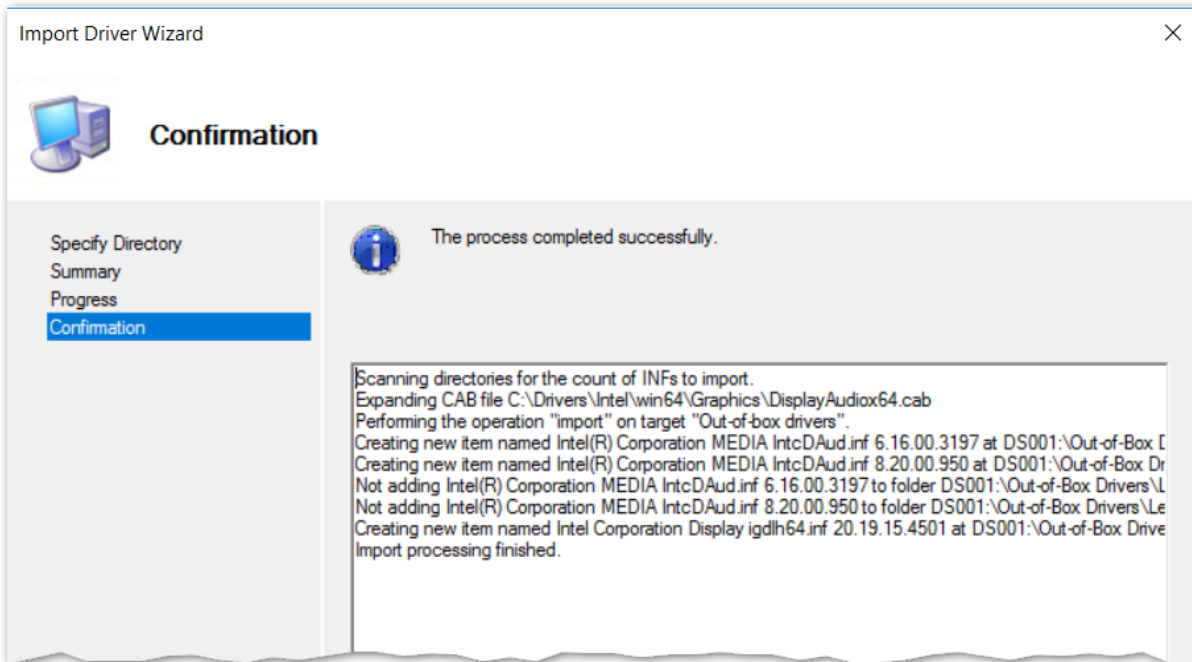
Wählen Sie nun den Ordner „Out-of-box-Driver“ erneut aus und wählen Sie „Import Drivers“. Anschließend wählen Sie den Ordner, in dem sich die Modell-Treiber befinden.



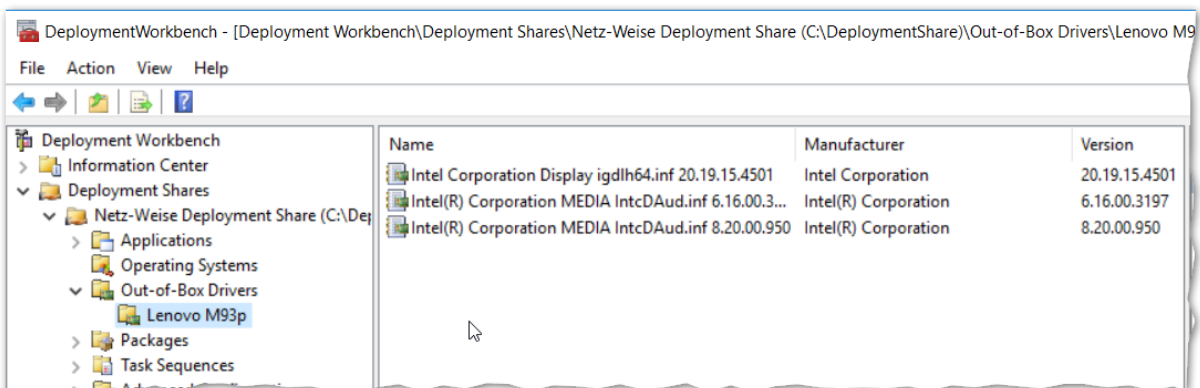
MDT kann keine Setup.exe ausführen, sondern benötigt immer die Dateien mit den Endungen .inf, .cat und .sys. Wenn sich mehrere Treiber in dem angegebenen Ordner befinden, importiert das MDT alle Treiber, die es findet. Für Business-PC bietet eigentlich jeder Hersteller ein Download-Tool an, das es erlaubt, die Treiber für sämtliche Modelle herunterzuladen. Bei Lenovo heißt es Thinkvantage System Update. Sie finden einen Support-Artikel und den Download-Link unter <https://support.lenovo.com/de/de/solutions/ht003029>.



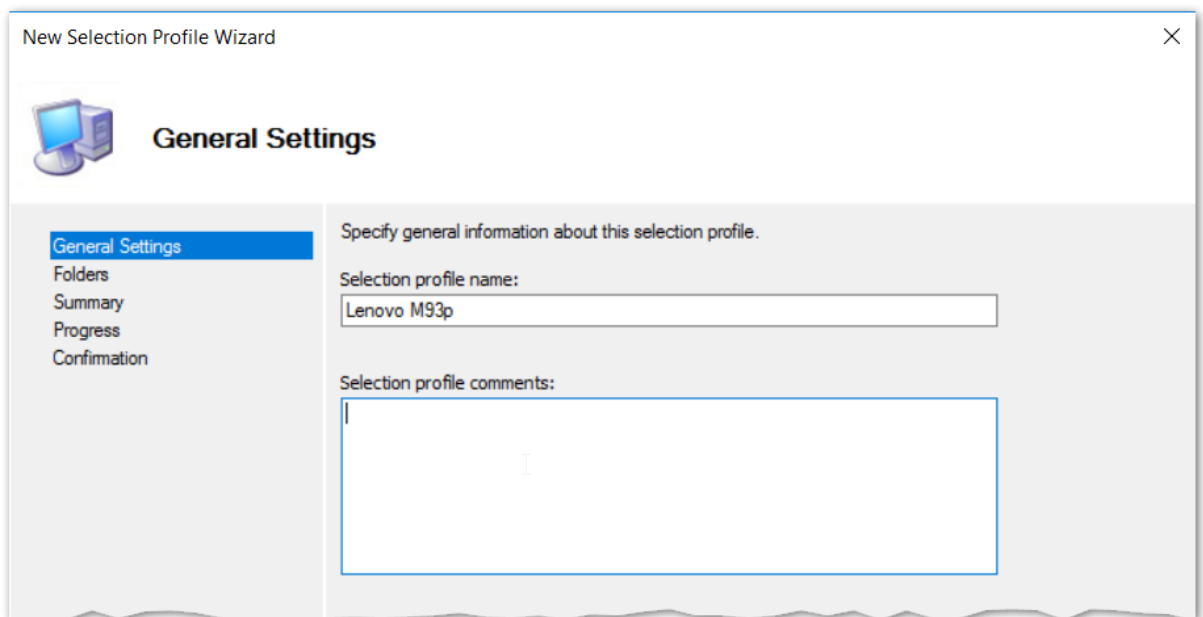
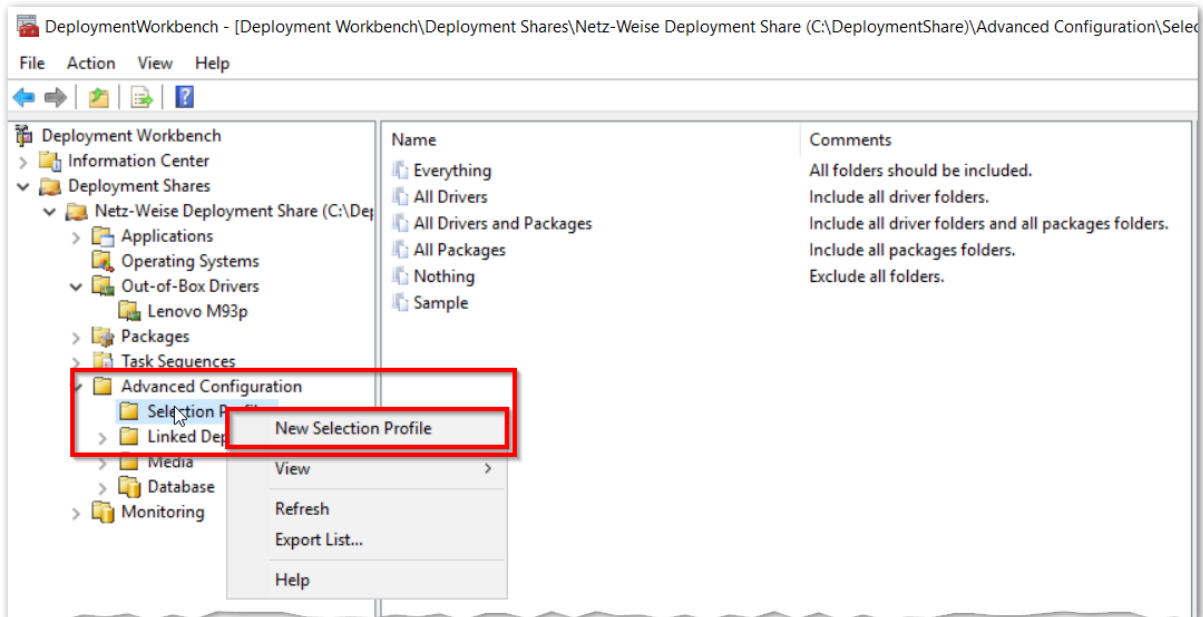
MDT kopiert alle gefundenen Treiber in das Deployment-Share.



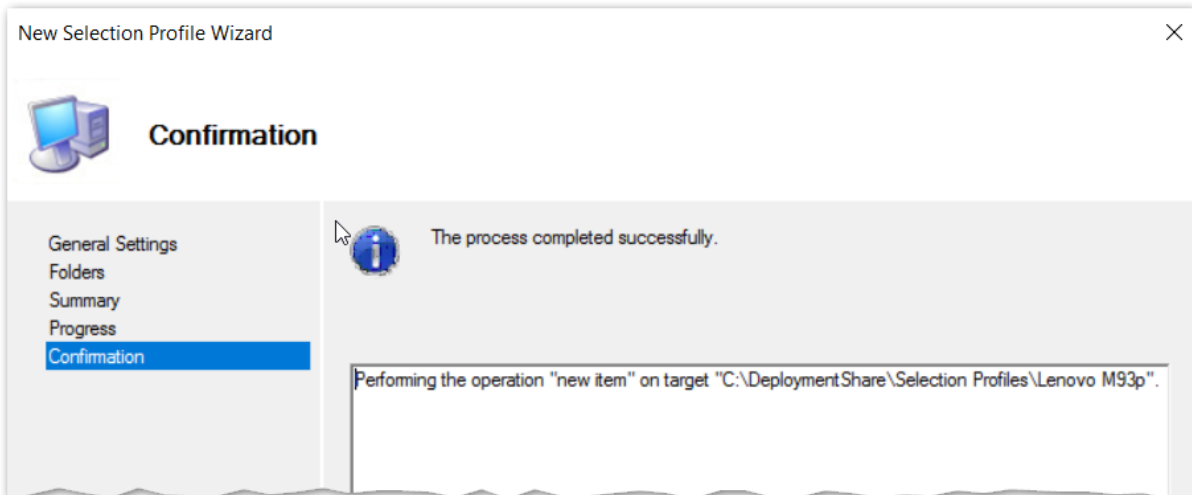
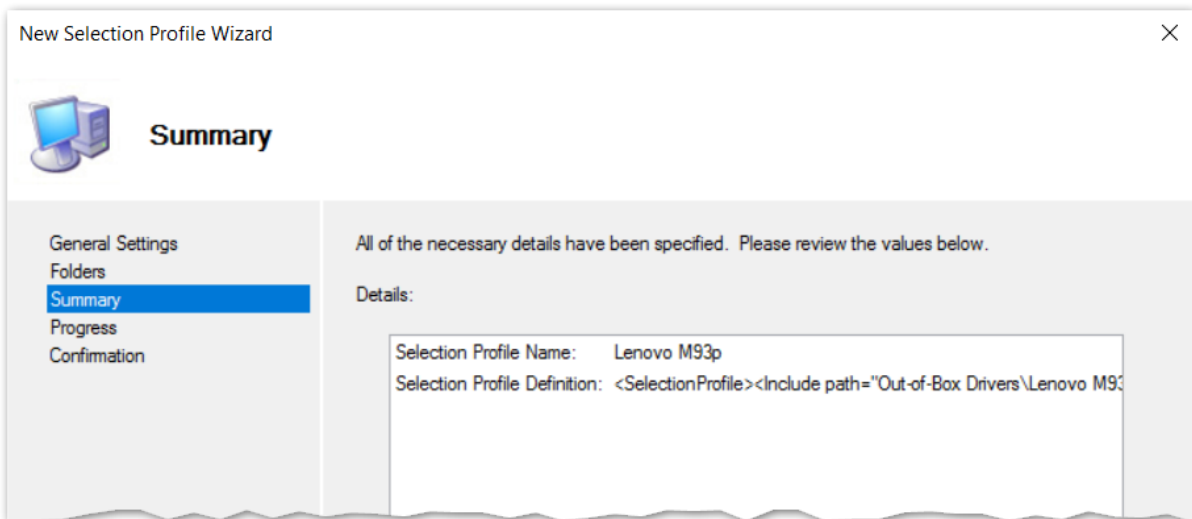
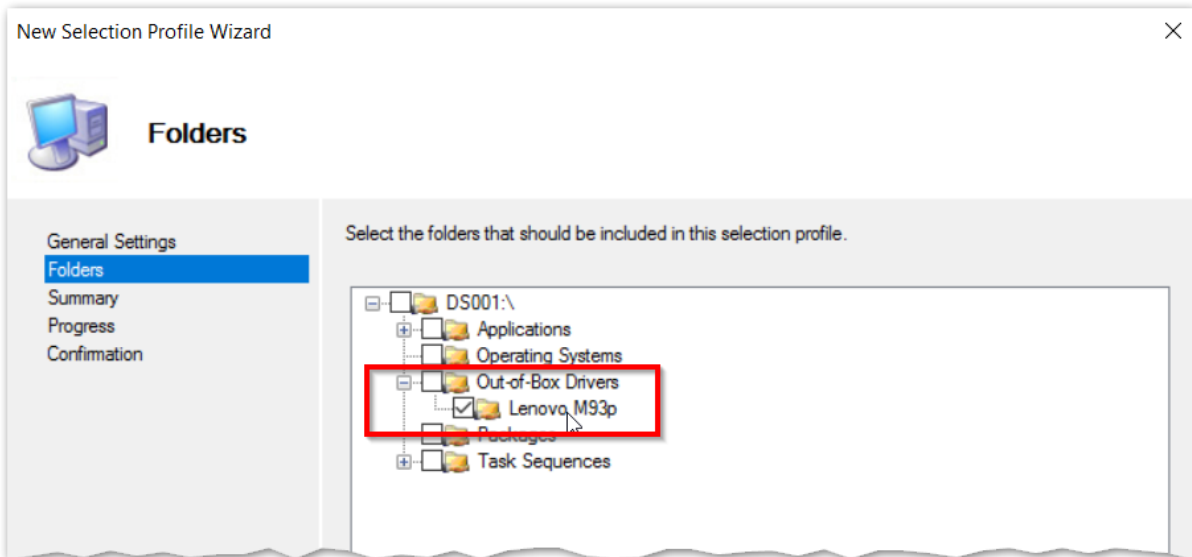
Wenn der Import abgeschlossen ist, werden alle kopierten Treiber im Modell-Ordner angezeigt.



Als nächstes legen Sie über den Eintrag „Advanced Configuration > Selection Profile“ ein neues Selection-Profil an.



Nachdem Sie dem Selection-Profil einen Namen gegeben haben, legen Sie die Ordner fest, die im Selection-Profil vorhanden sein sollen. Das Selection-Profil kann auch zur Erstellung von Offline Installations-DVDs verwendet werden, daher kann man hier auch Betriebssysteme und Anwendungen auswählen, die auf die DVD kopiert werden sollen. Das hier erstellte Profil soll aber nur die Treiber für den Lenovo M93p enthalten.

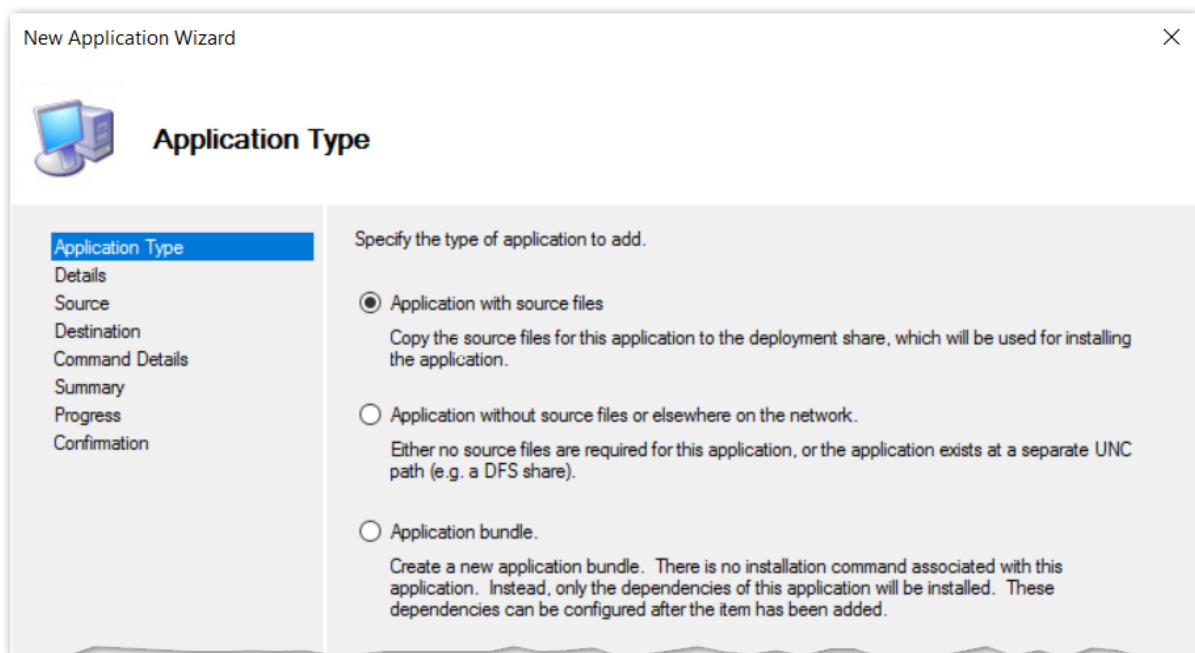


Das Selection-Profile kann z.B. verwendet werden, um Treiber in das Windows-PE Image mit aufzunehmen oder um eine Tasksequenz anzupassen. Anwendungsszenarien finden Sie in den Abschnitten [Erstellen einer einfachen Task-Sequence](#) und [Anpassen von Deploymentshare und Windows PE](#).

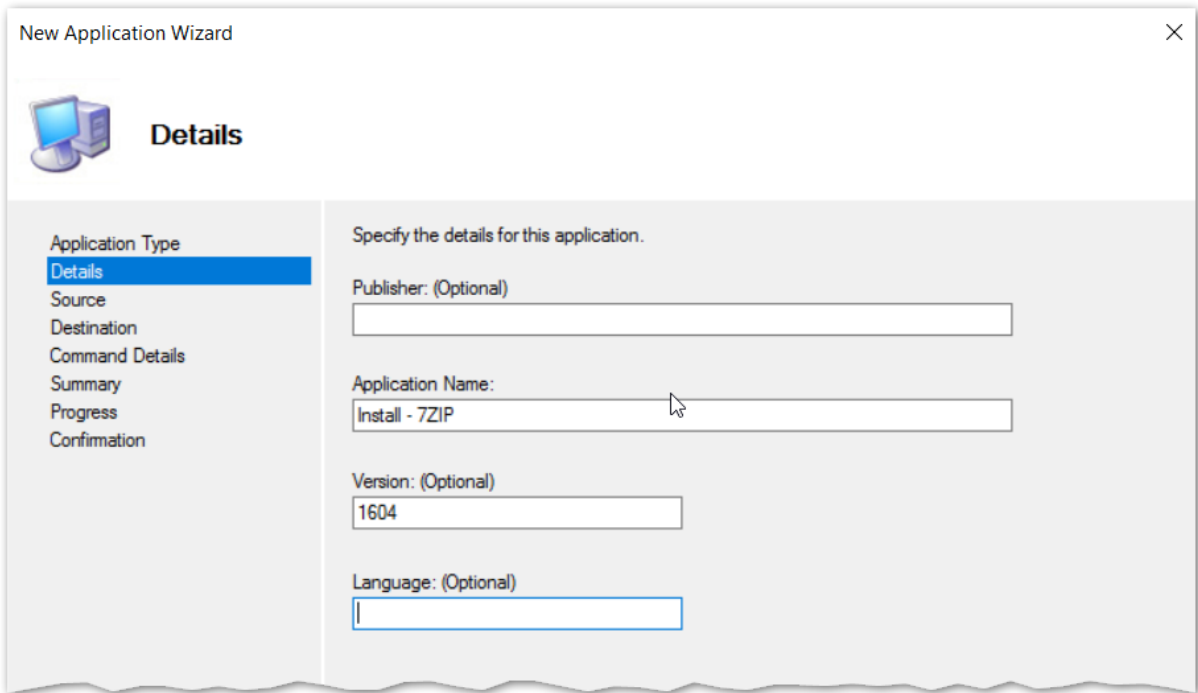
Anwendungen und Konfigurationsskripte

Im letzten Schritt werden Anwendungen und Konfigurationsskripte hinzugefügt. Da das MDT die Installation nicht selbständig automatisieren kann, müssen Sie zuerst herausfinden, wie das Setup für Ihre Software unbeaufsichtigt installiert werden kann. Eine ausführliche Beschreibung hierzu finden Sie im Abschnitt [Software und Konfigurationsskripte bereitstellen](#).

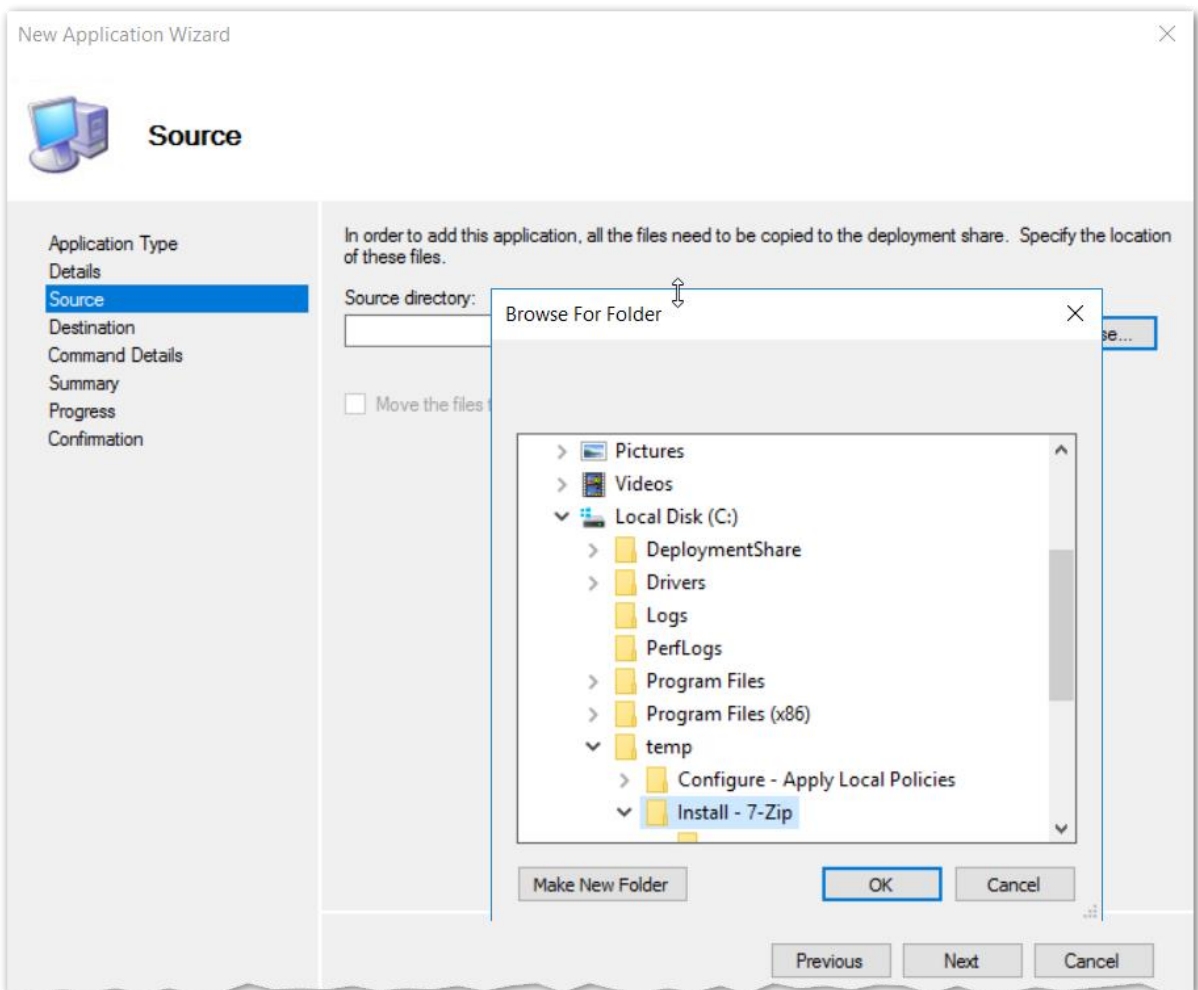
Wählen Sie aus dem Abschnitt „Applications“ New Application aus. Im folgenden Assistenten müssen Sie auswählen, ob Sie die Anwendung mit den Quelldateien in das Deployment-Share kopieren wollen (Application with source files), oder ob die Installationsquellen auf einem Netzwerk-Share liegen und nicht in das Deployment-Share kopiert werden sollen. Application Bundle können Abhängigkeiten zwischen Anwendungen auflösen. Eine genaue Beschreibung finden Sie unter <http://www.vaughnemiller.com/2011/12/21/application-bundles-in-mdt-2010/>.



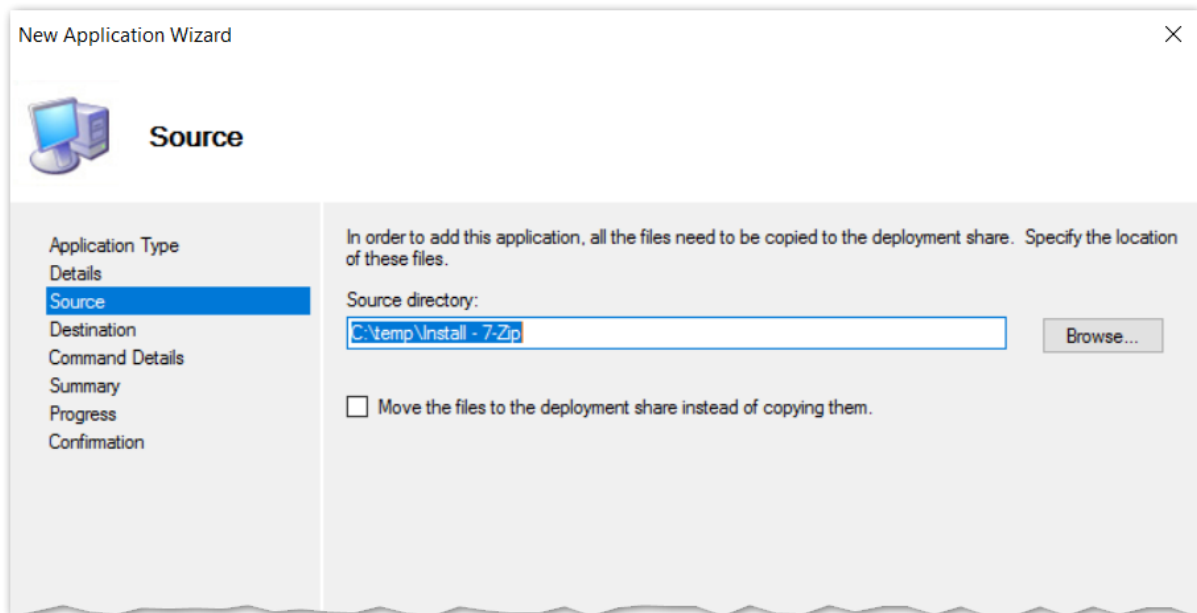
Geben Sie nun einen Anwendungsnamen ein. Im Beispiel wird der Name aus „Install-“ und dem Programmnamen zusammengesetzt. Dadurch wird es möglich, Anwendungen und Konfigurationsskripte zu unterscheiden. Sie können außerdem die Version, Sprache und den Herausgeber angeben.



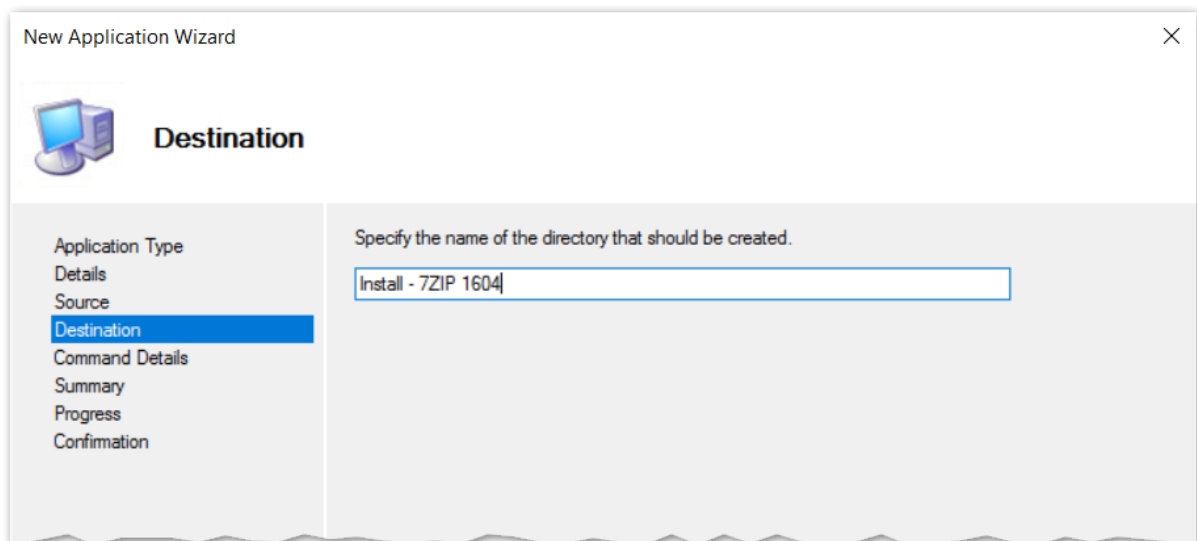
Wählen Sie nun den Ordner aus, in dem sich die Anwendung befindet. Achten Sie darauf, dass der gesamte Ordner kopiert wird. Auch eine einzelne Setupdatei muss in einem eigenen Ordner liegen!



Wenn Sie das Quellverzeichnis nicht mehr benötigen, wählen Sie „Move the files to the deployment Share“ aus.

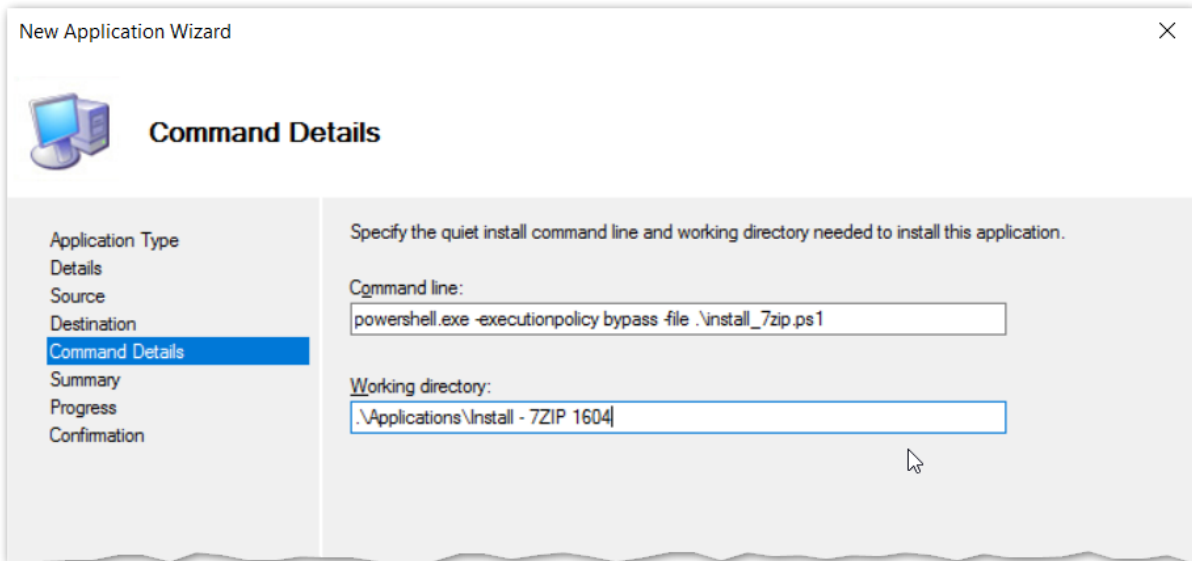


Geben Sie einen Namen für das Verzeichnis an, in den die Anwendung kopiert wird. Dies ist der Name des Ordners im Dateisystem.

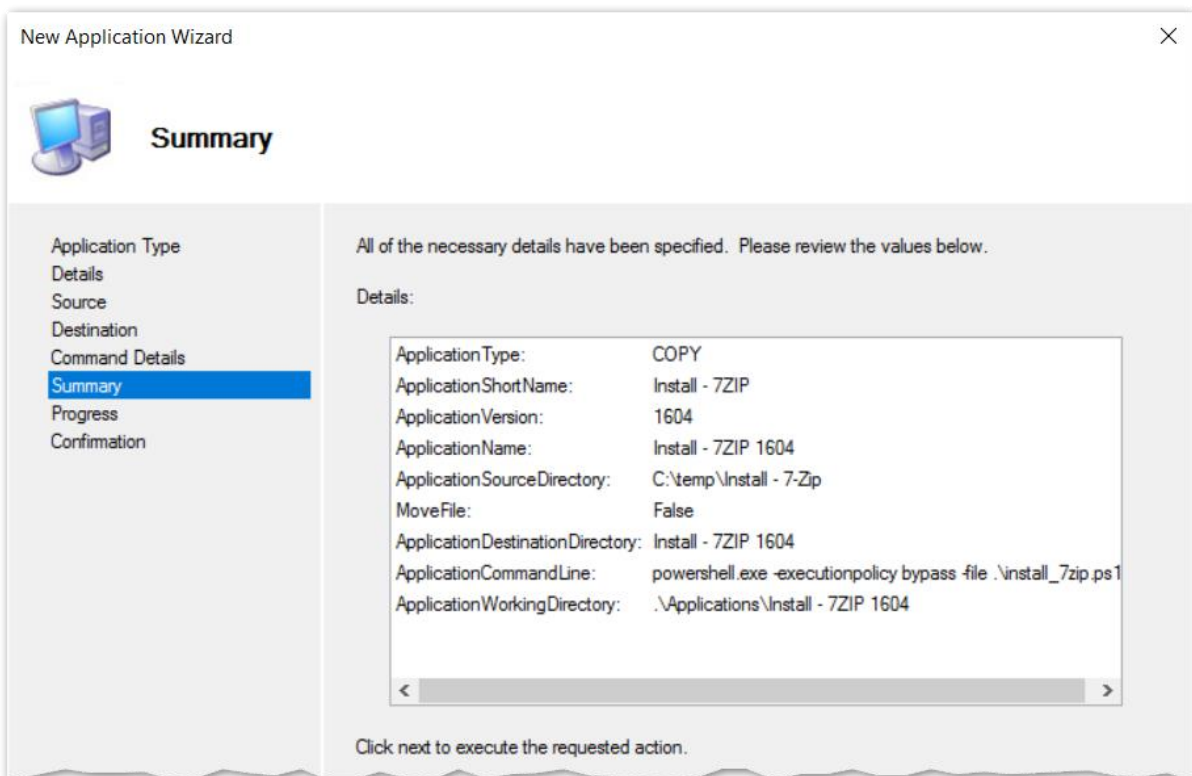


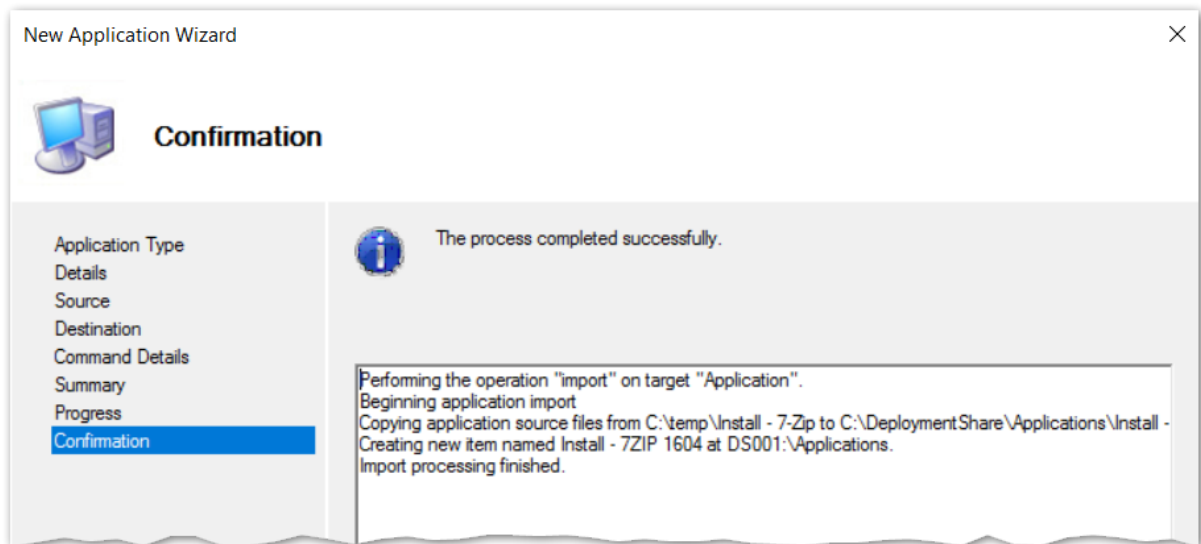
Als letztes geben Sie die Kommandozeile an, die aufgerufen werden soll, um die Anwendung zu installieren. Es ist empfehlenswert, für die Installation immer ein Installationskript zu verwenden, da man dieses flexibler verwenden kann. Sie können im Skript z.B. direkt das MDT-Logging ansprechen, Registry-Keys setzen oder Dateien nachträglich bearbeiten. Wenn Sie Powershell für Ihre Installationskripte nutzen, sollten Sie beim Aufruf der Powershell die Ausführungsrichtlinie auf Bypass setzen. Bypass führt alle Skripte aus und deaktiviert alle Sicherheitsfunktionen wie die Prüfung von Zertifikaten. Der Parameter -file zeigt auf das Skript, das ausgeführt werden soll. „.\“ vor dem Skriptnamen gibt an, dass sich das Skript im aktuellen Ordner befindet.

```
Powershell.exe -executionPolicy bypass -file .\Install_7zip.ps1
```

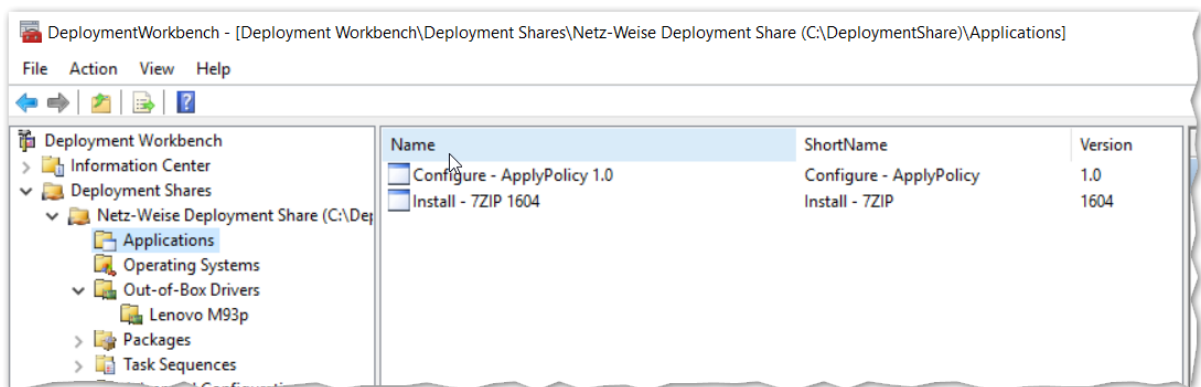


Sie erhalten nun noch einmal eine Zusammenfassung der eingegebenen Daten, bevor das Erstellen der Anwendung beginnt.





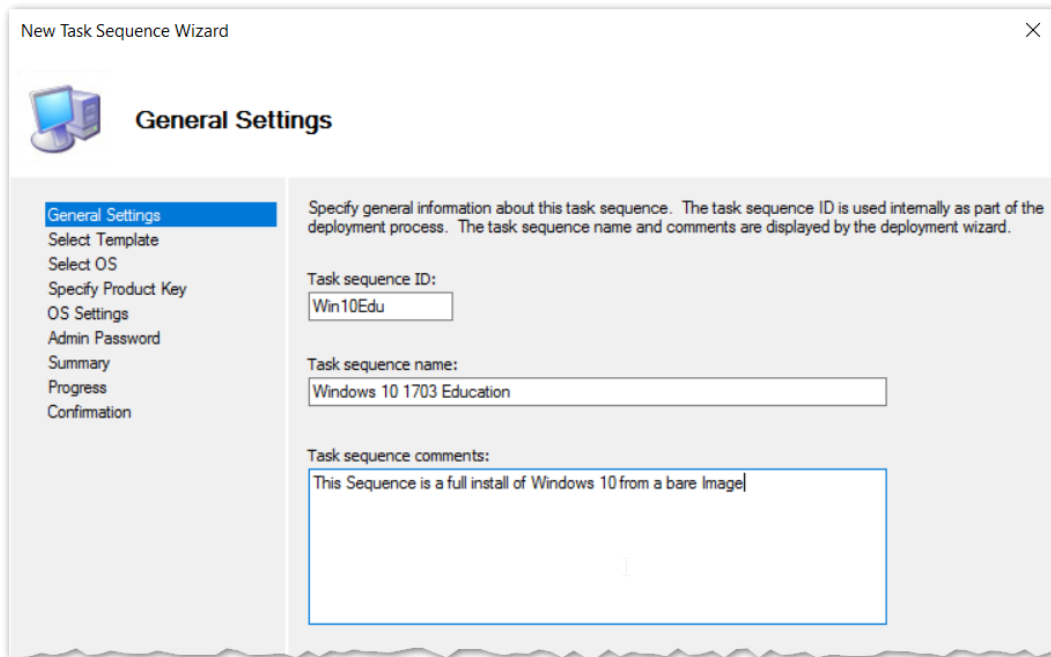
In der Abbildung ist eine weitere Applikation zu sehen. Tatsächlich handelt es sich nicht um eine Anwendung, sondern um ein Konfigurationsskript. Da eine „Application“ ja tatsächlich einfach nur ein Programm in der Kommandozeile aufruft, kann man so auch beliebige Skripte während der Installation aufrufen.



Erstellen einer einfachen Task-Sequence

Task-Sequenzen entsprechen der Installationsanleitung. Eine Task-Sequenz legt fest, in welcher Reihenfolge die Installationskripte des MDT aufgerufen und welche Parameter angegeben werden müssen. Glücklicherweise müssen Sie Tasksequenzen nicht selber erstellen, was ziemlich aufwändig ist. Stattdessen gibt es eine ganze Reihe von Vorlagen (Templates), die Sie nur noch anpassen müssen.

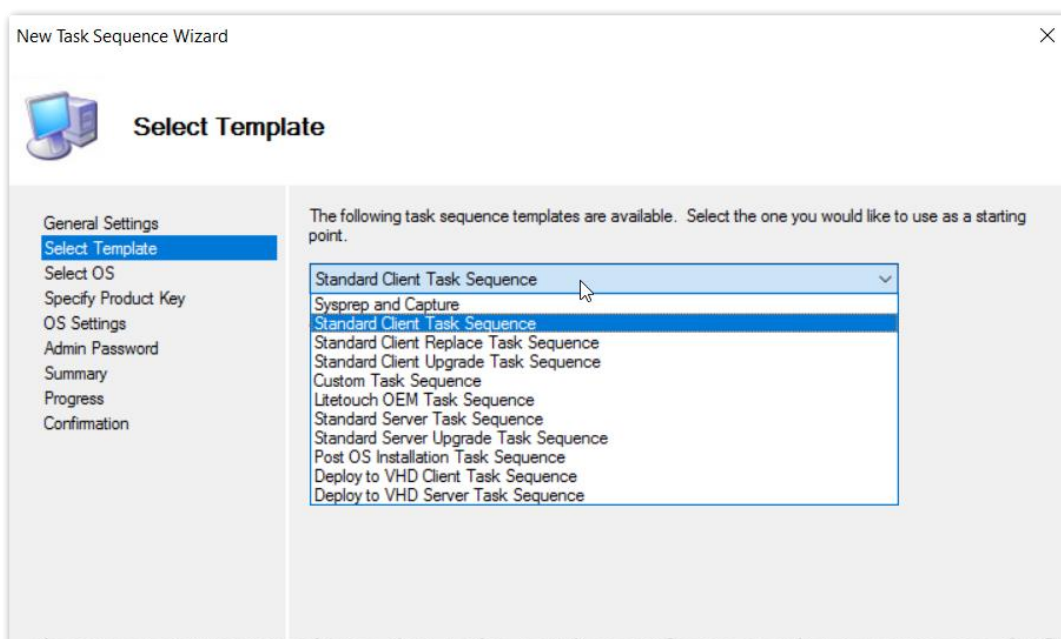
Um eine Tasksequenz anzulegen, wählen Sie im Deployment-Share aus dem Kontextmenü des Eintrags „Task Sequences“ den Menüpunkt „New Task Sequence“ aus. Im folgenden Menü geben Sie dann einen Namen für die Task-Sequence ein. Außerdem benötigen Sie eine eindeutige ID. Optional können Sie einen Kommentar eingeben. Es ist empfehlenswert, hier einen eindeutigen Namen und eine sinnvolle Beschreibung einzutragen, da der Benutzer, der das Setup startet, als erstes die Task Sequence auswählen muss und genau diese Daten angezeigt werden. Die Anzeige können Sie im Abschnitt [Installieren eines Clients](#) sehen.



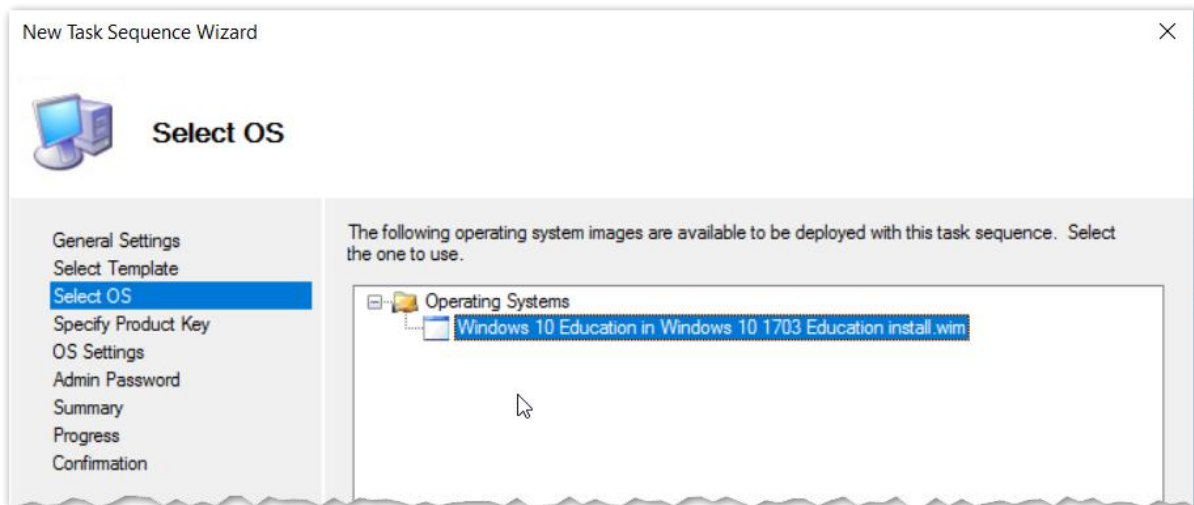
Sie haben eine ganze Reihe von möglichen Vorlagen verfügbar.

| Template | Beschreibung |
|-----------------------------------|--|
| Sysprep and Capture | Erzeugt aus einem laufenden Windows ein wim-File |
| Standard Client TS | Installiert einen Client |
| Standard Client Replace TS | Sichert die Benutzerdaten und installiert den Client neu |
| Standard Client Upgrad TS | Aktualisiert auf der gleichen Hardware das Betriebssystem |
| Custom TS | Erstellt eine leere TS, die Sie frei anpassen können |
| Standard Server TS | Installiert einen Server |
| Post OS Installation TS | Kann alleinstehend von einem installierten System gestartet werden |
| Deploy to VHD | Die Installation wird auf eine VHD durchgeführt |

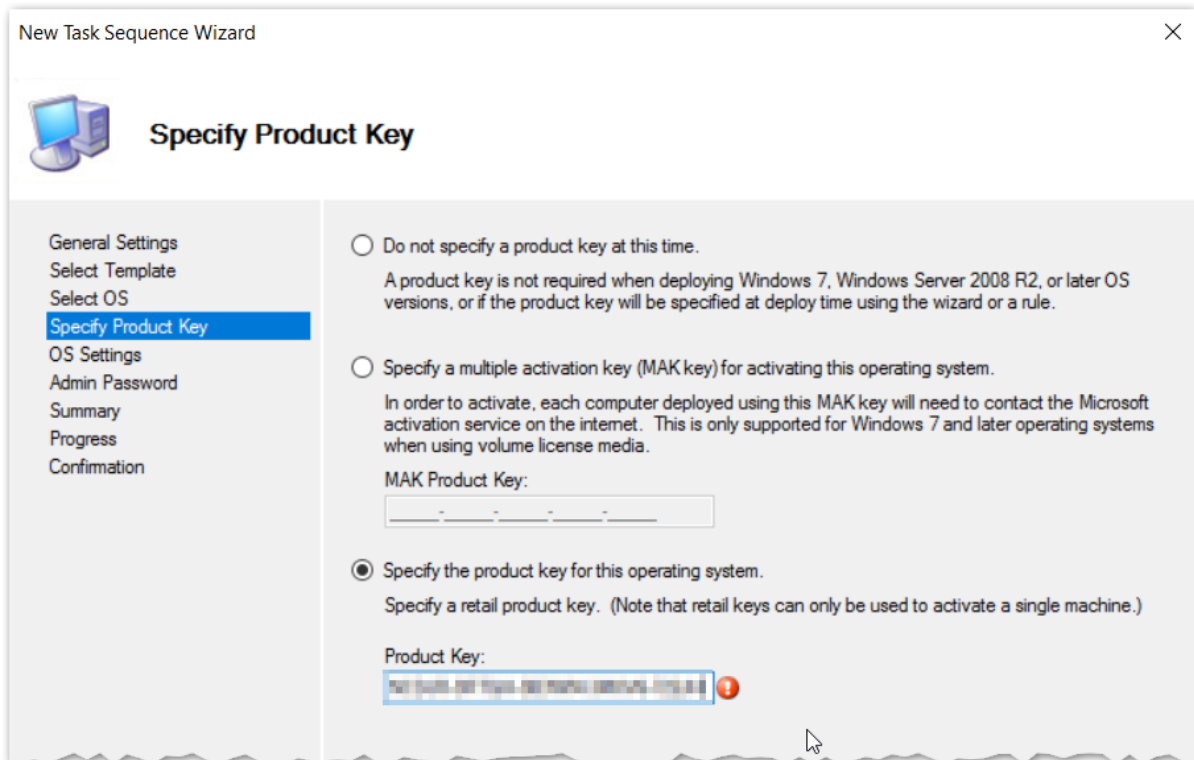
Wählen Sie für die Clientinstallation „Standard Client Task Sequence“

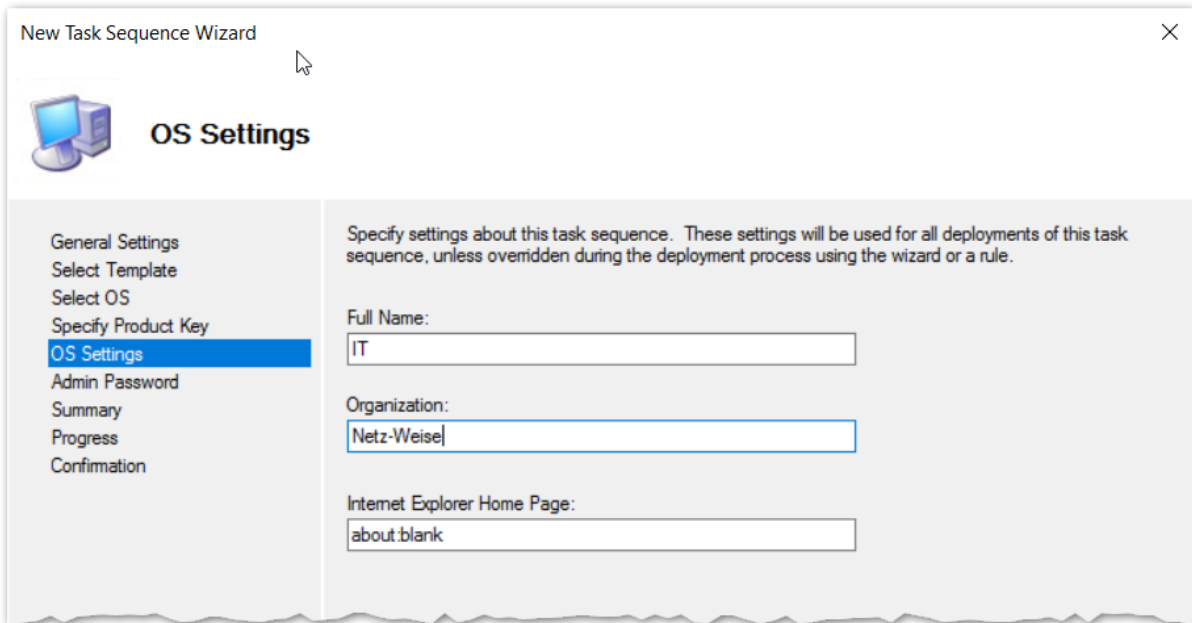


Nun müssen Sie das Betriebssystem auswählen, das installiert werden soll.

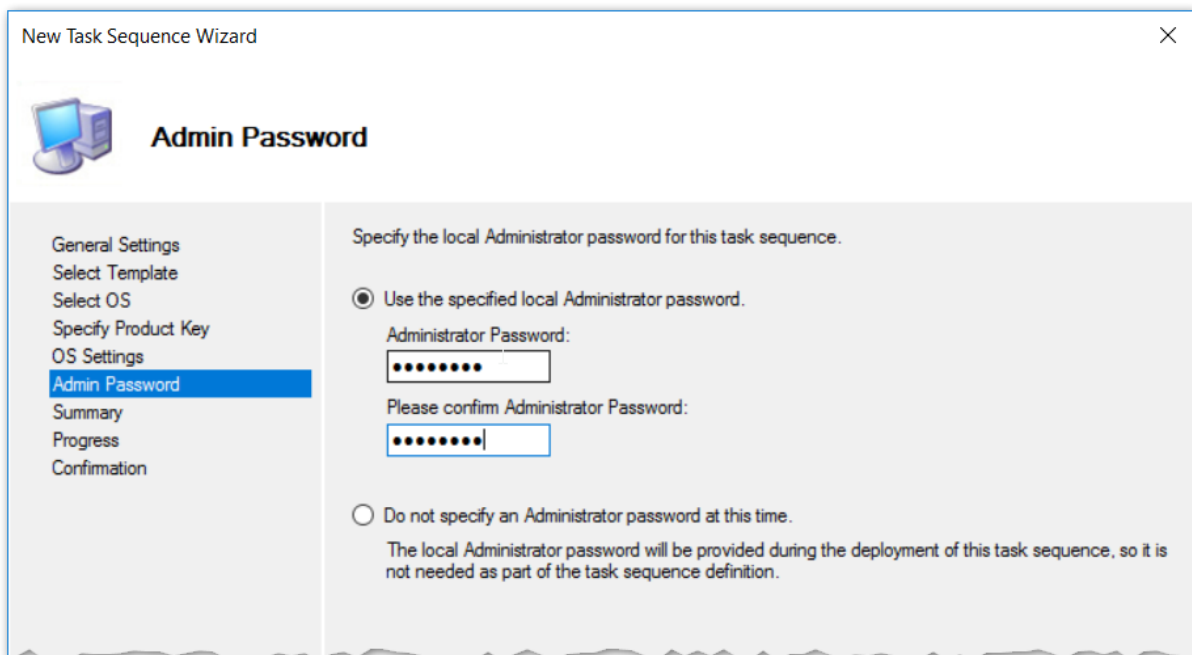


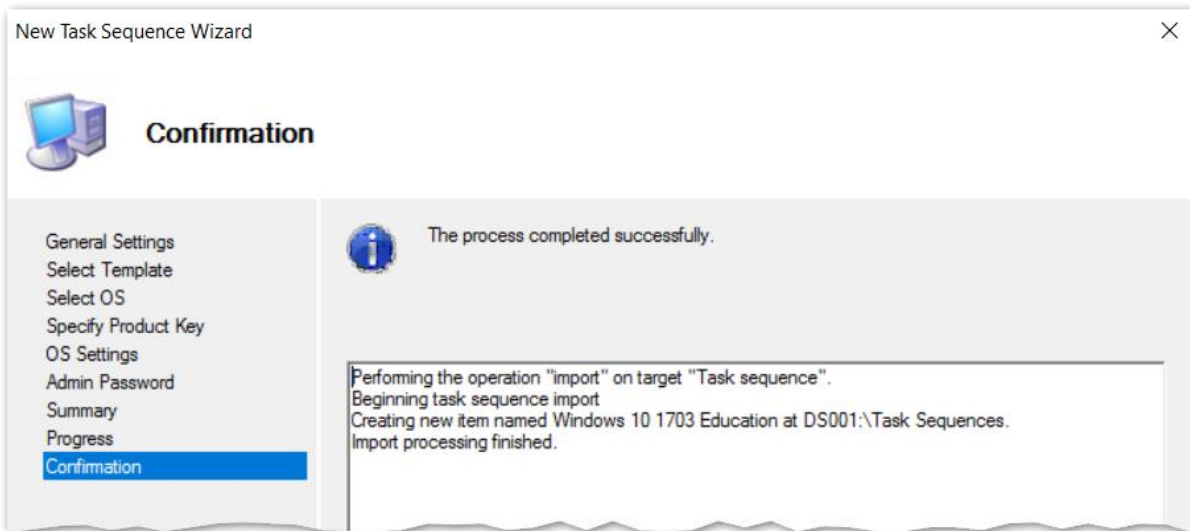
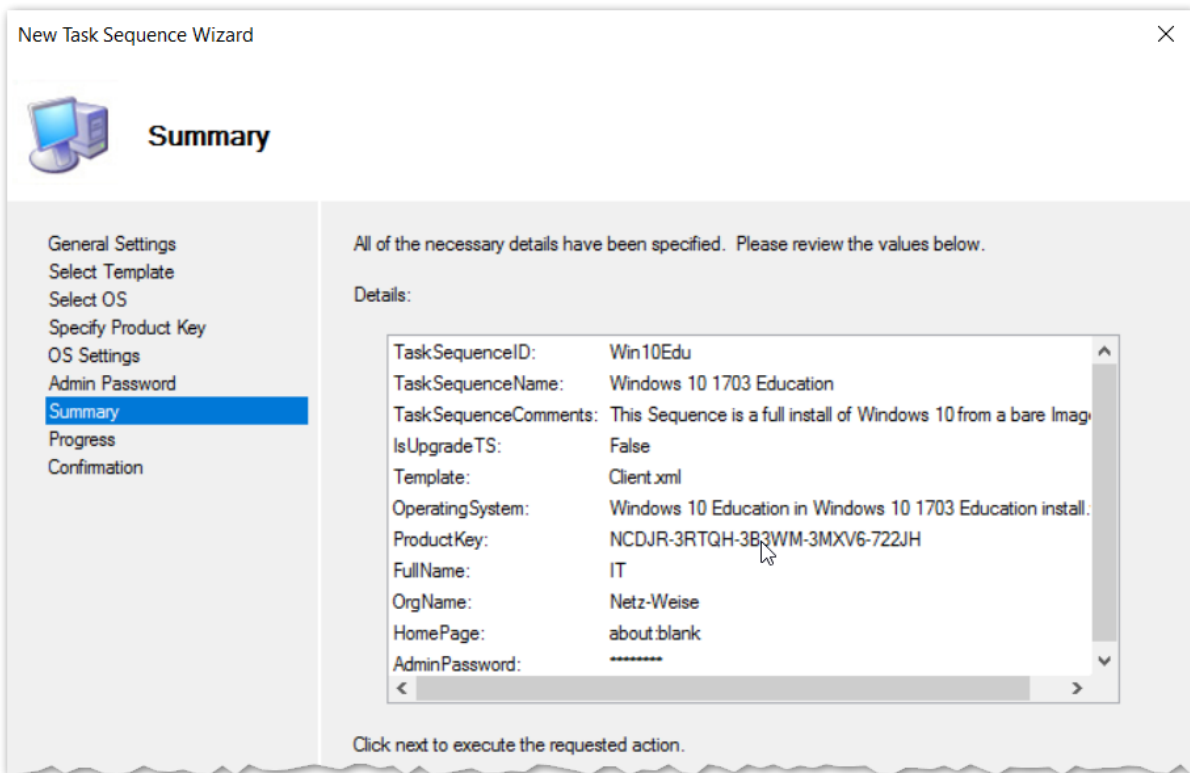
Nun können Sie den Produktschlüssel angeben. Wenn Sie „Do not specify a product key at this time“ auswählen, und Sie den Produktschlüssel auch beim Starten der Installation nicht angeben, wird die automatische Installation von Windows an der Stelle unterbrochen, an der der Produktschlüssel eingegeben werden muss.



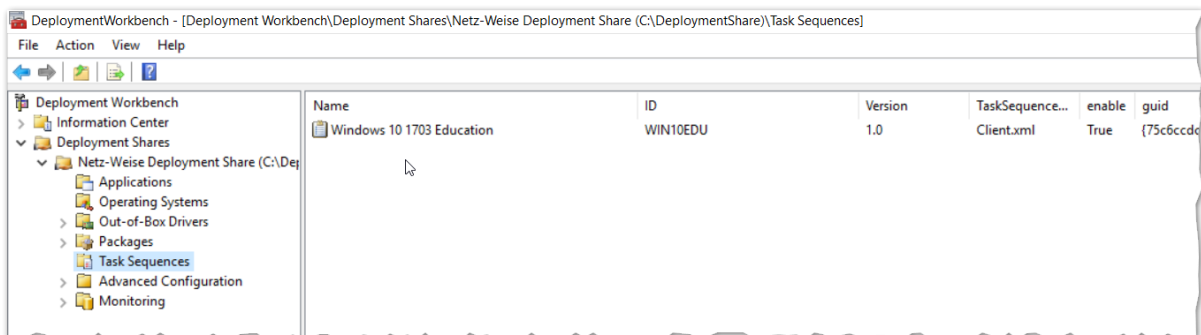


Legen Sie nun das Administrator-Kennwort des lokalen Administrators fest. Sie können das Kennwort allerdings auch beim Starten der Installation festlegen.





Nach Abschluss des Assistenten wird die Task-Sequence angezeigt. Sie können die Task-Sequence anpassen, indem Sie die Ihre Eigenschaften aus dem Kontextmenü aufrufen.



Die Eigenschaften können hier bearbeitet werden, aber auch auf eine bestimmte Plattenform festgelegt, deaktiviert oder im Deployment-Wizard ausgeblendet werden. Dadurch ist die Task Sequence für die Installation nicht mehr verfügbar.

Windows 10 1703 Education Properties

General Task Sequence OS Info

Task sequence ID: WIN10EDU

Task sequence name: Windows 10 1703 Education

Comments: This Sequence is a full install of Windows 10 from a bare Image

Task sequence version: 1.0

This can run on any platform.

This can run only on the specified client platforms:

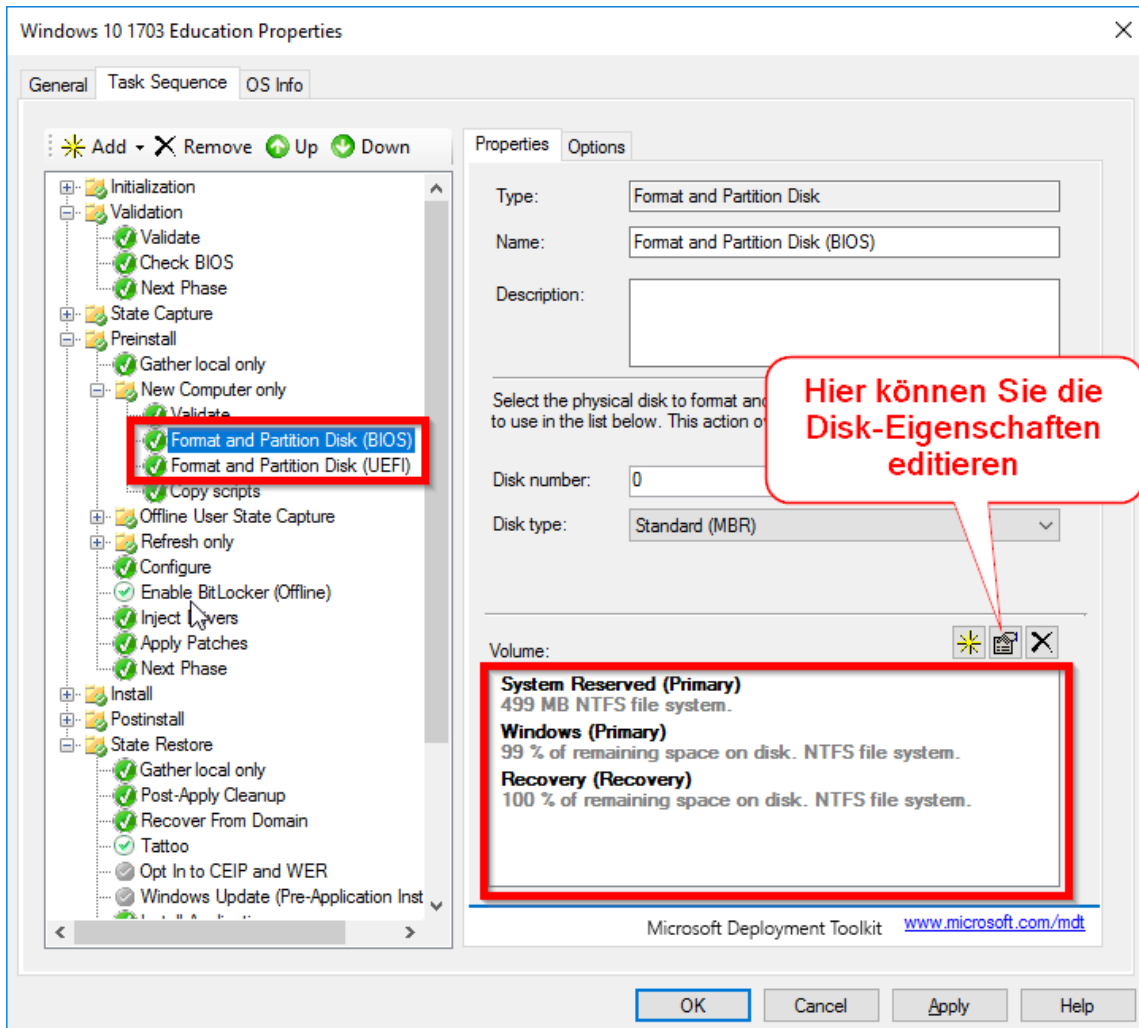
- Windows PE
- All x86 NT
- All x86 Windows 7 Client
- All x86 Windows 8 Client
- All x86 Windows 8.1 Client
- All x86 Windows 10 Client
- All x64 Windows 7 Client
- All x64 Windows Server 2008 R2

Hide this task sequence in the Deployment Wizard

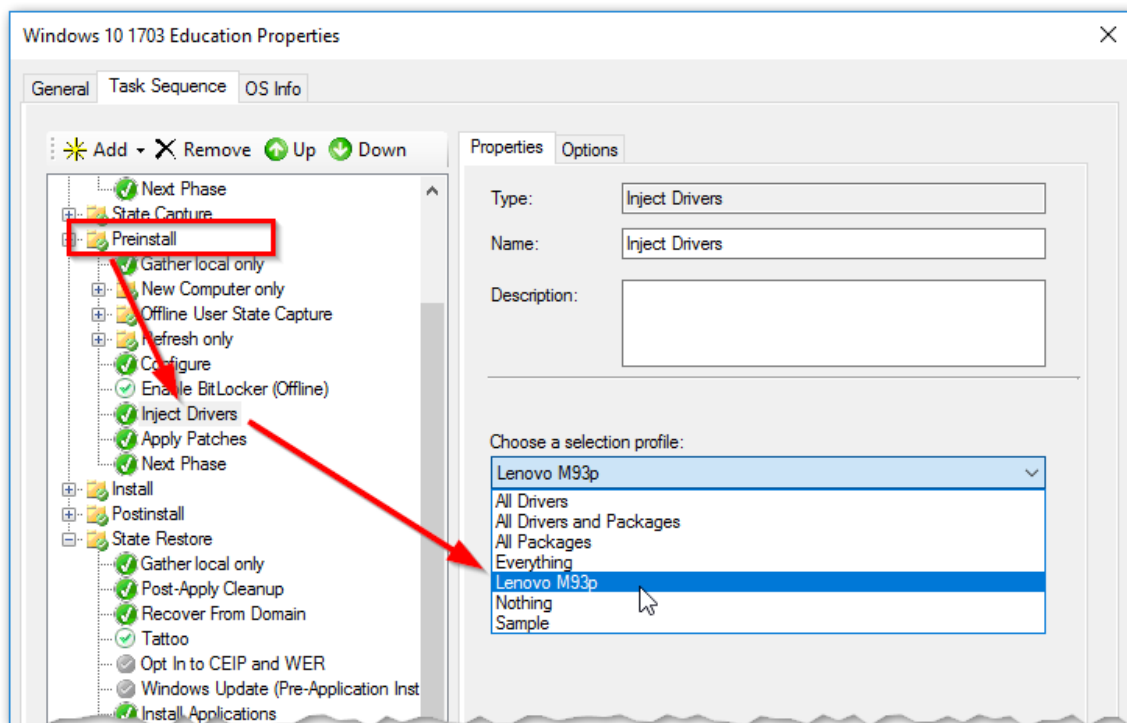
Enable this task sequence

Auf der nächsten Registerkarte mit den Namen „Task Sequence“ finden Sie die Reihenfolge der einzelnen Installationsschritte. Jeder Schritt der Liste entspricht einem Skript, das bei der Installation abgearbeitet wird. Die Task-Sequence wird dabei von unten nach oben abgearbeitet.

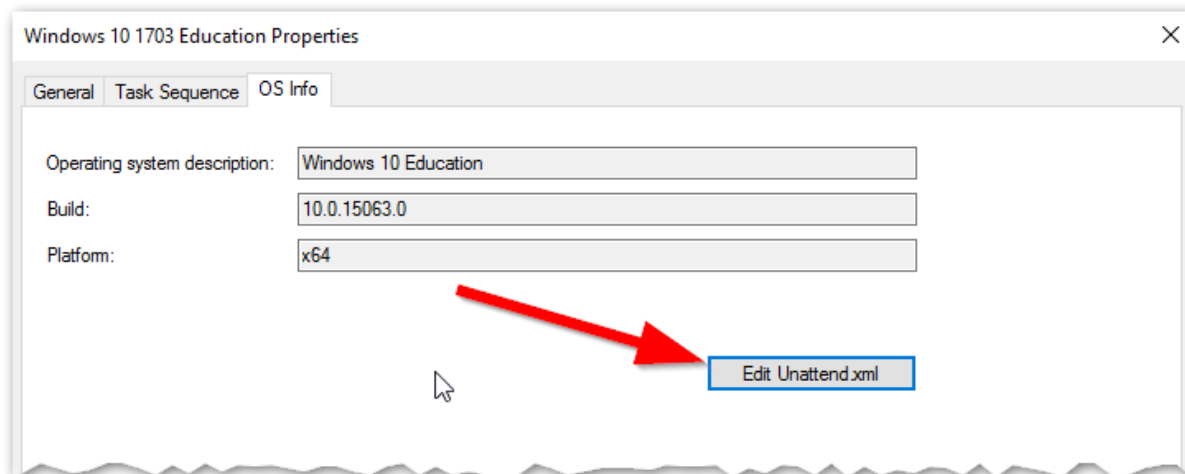
Die Task-Sequence kann angepasst werden, indem Sie entweder die Parameter auf der rechten Seite anpassen, wie z.B. die Partitionsgrößen, es können aber auch komplett neue Aktionen über den Button „Add“ oben links in der Liste hinzugefügt oder entfernt werden. Mit den Up- und Down-Buttons können Sie auch die Reihenfolge der Verarbeitung anpassen.



Hier können Sie auch zu installierende Treiber anhand eines Selection Profiles festlegen.

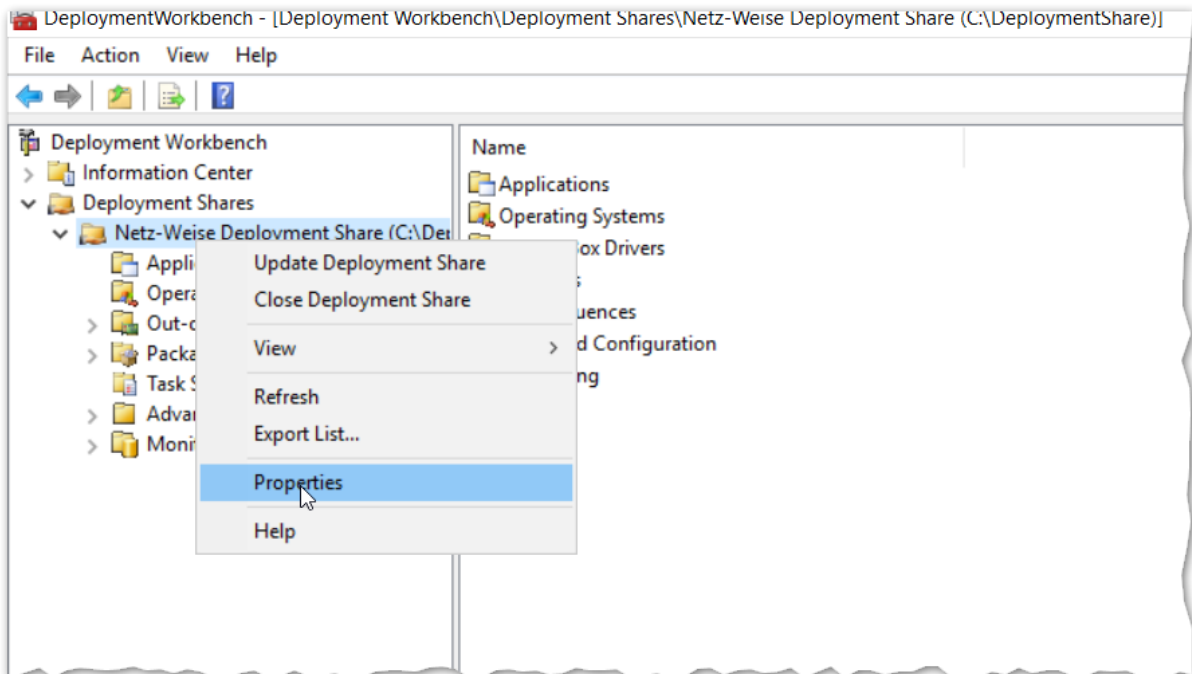


Im letzten Fester OS-Info finden Sie schließlich den Link zur Unattend.xml, die Sie auch manuell bearbeiten können. Dies kann allerdings eine Weile dauern, wenn Sie die Datei zum ersten Mal aufrufen, da erst ein Katalogfile erstellt werden muss.

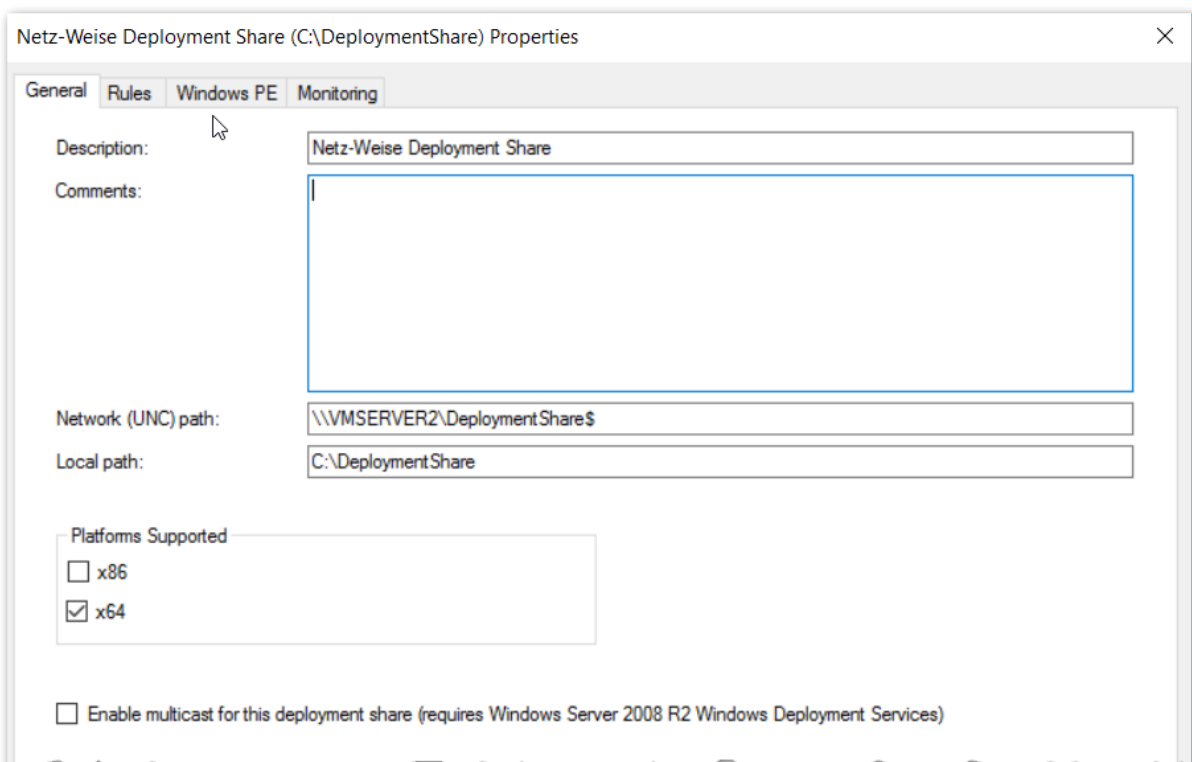


Ein bootfähiges Windows PE erstellen

Um die Installation zu starten, benötigen Sie zuerst ein bootfähiges Windows PE in Form eines ISO- oder WIM-Files. Das MDT übernimmt die Erstellung für Sie. Sie können das Windows PE allerdings vorher an Ihre Bedürfnisse anpassen. Wählen Sie hierfür Ihr Deploymentshare aus (Im Beispiel ist das der Eintrag „Netz-Weise Deployment Share“) und wählen Sie aus dem Kontextmenü „Properties“.



Im ersten Fenster können Sie festlegen, ob Sie ein 32-Bit oder ein 64-Bit Windows PE erstellen wollen. Das hat nur Auswirkungen auf von Ihnen zusätzlich gewählte Programme zum WinPE, also normalerweise gar keine. Wählen Sie nur eine Plattform aus, um die Erstellung zu beschleunigen.



Multicast-Unterstützung

Multicast bedeutet, dass der Server das zu installierende Datenpaket an mehrere Clients gleichzeitig verteilen kann, ohne die Daten mehrfach verschicken zu müssen. Mit Multicast können Sie, wenn Sie mehrere Clients gleichzeitig installieren wollen, den Installationsvorgang deutlich beschleunigen und dabei die Festplatten- und Netzwerklast des Servers deutlich entlasten. Das funktioniert, indem alle Clients die gleiche (Multicast-) IP-Adresse verwenden. Multicast-IP-Adressen können zusätzlich zur Unicast-IP verwendet werden.

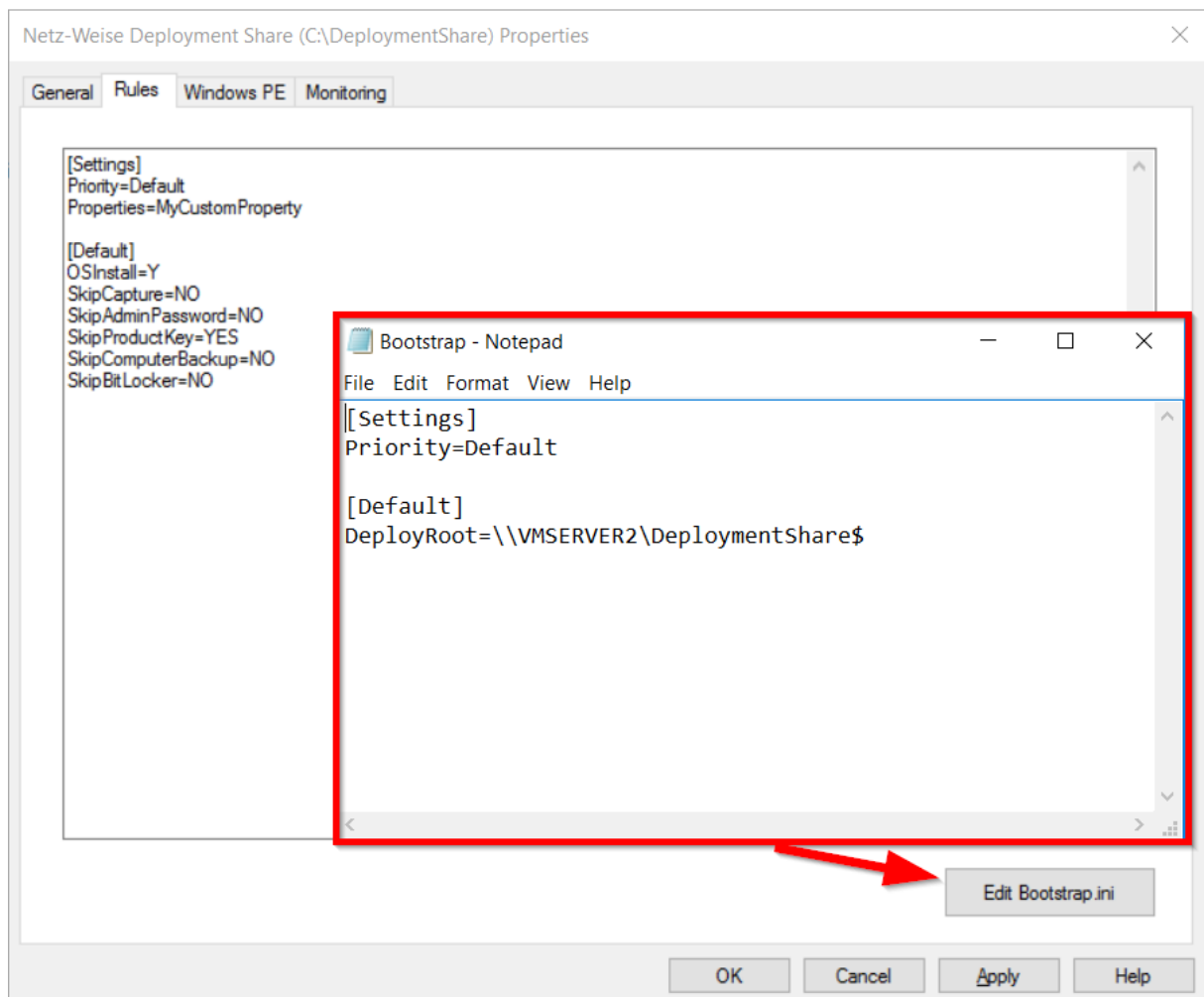
Der Server fängt an, das WIM-File an die Multicast-Adresse zu schicken, sobald der erste Client online ist. Sobald weitere Clients das Setup starten, bekommen auch Sie die Datenpakete geliefert. Wenn der Server die Installationsdaten fertig übertragen hat, beginnt er die Übertragung von neuem, bis der letzte Client alle Daten erhalten hat und sich vom Installationsserver abmeldet.

Wenn Sie Multicastunterstützung benötigen, aktivieren Sie den Haken „Enable Multicast for this Deployment Share“. Sie benötigen dann aber auf jeden Fall einen WDS-Server für die Installation. Sofern Sie einen WDS-Server besitzen, brauchen Sie für die Aktivierung von Multicast nichts weiter zu machen als den Haken zu setzen. Die Multicast-Unterstützung wird dann im Windows PE automatisch eingeschaltet.

Auf der Registerkarte Rules wird der Inhalt der Customsettings.ini angezeigt, die der Client beim Starten aus dem Deploymentshare lädt. Sie ist im Ordner Control abgelegt und definiert u.a., welche Eingabefenster während des Startvorgangs des Clients abgefragt werden (s. [Installieren eines Clients](#)). Sie kann jederzeit angepasst werden. Da der Client die Datei nach dem Starten einliest, werden Änderungen, die Sie hier vornehmen, sofort gültig!

Wenn Sie den Button „Edit Bootstrap.ini“ auswählen, wird im Notepad die Bootstrap.ini geöffnet, die ebenfalls im Ordner „Control“ liegt. In der Bootstrap.ini werden Daten festgelegt, die Windows PE vor dem Herstellen einer Verbindung mit dem Netzwerk auswertet. In der Bootstrap.ini können neben dem Pfad zum Deployment-Share auch z.B. die Standard-Keyboardeinstellungen für Windows PE hinterlegt werden, oder Benutzername und Kennwort für den MDT-User, mit dem eine Verbindung zum Deployment-Share hergestellt wird.

Die Bootstrap.ini wird vor dem Verbinden mit dem Deploymentshare ausgewertet und ist im Gegensatz zur Customstettings.ini im Windows PE Bootimage enthalten. **Alle Änderungen in der Bootstrap.ini erfordern daher, dass das Windows PE neu erstellt und verteilt wird, bevor die Änderungen gültig werden können!**

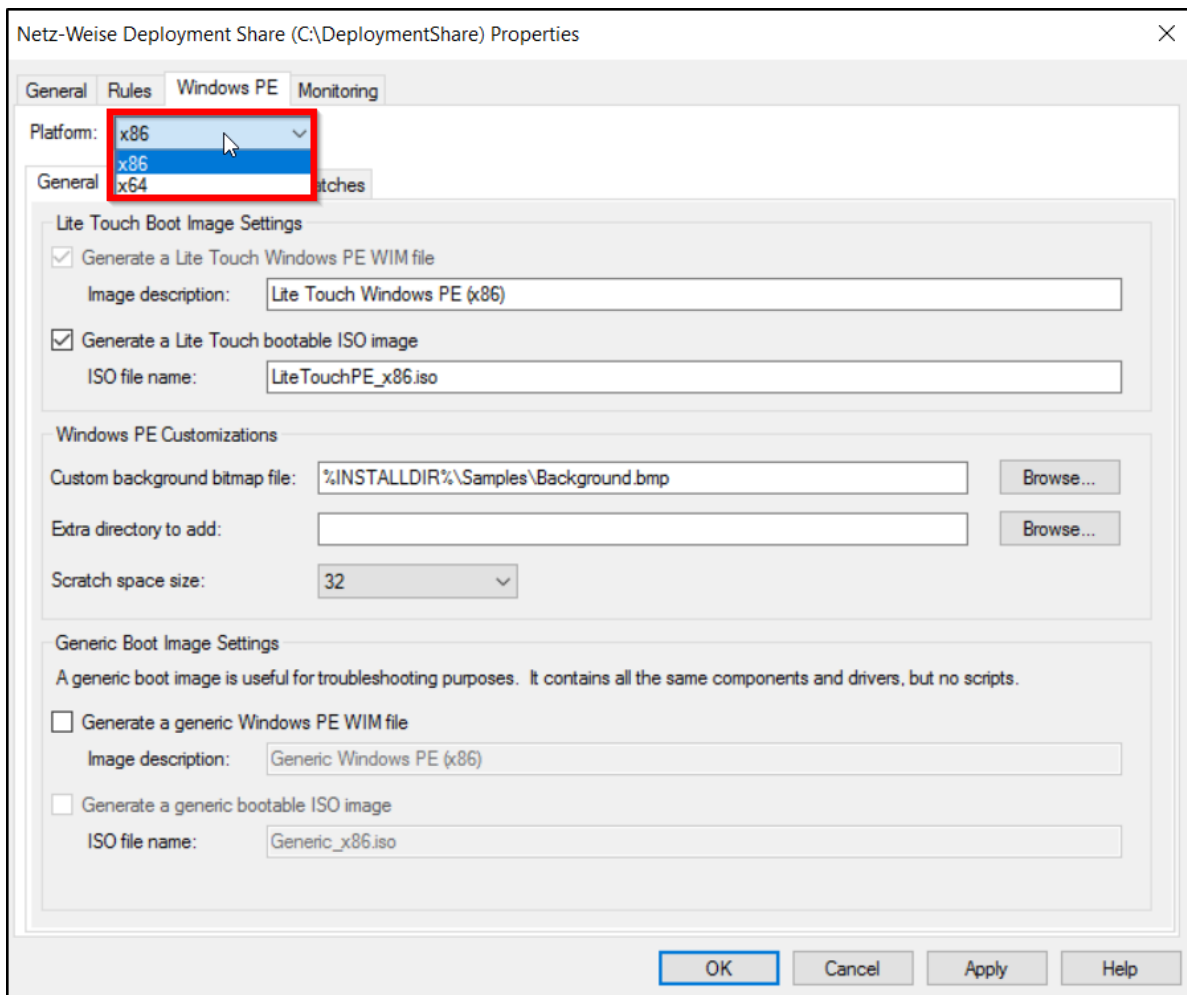


Wichtig! Sie haben oben auf der Registerkarte „Windows PE“ ein Drop-Down Fenster mit der Bezeichnung „Plattform“. Sie legen hier zwei unterschiedliche Konfigurationen fest, einmal für 32-Bit Systeme und einmal für 64-Bit Systeme. Es handelt sich hierbei um zwei unabhängige Konfigurationen. Wechseln Sie die Plattform, ändern sich auch die Einstellungen in der „Windows PE“ Registerkarte. Achten Sie darauf, dass Sie die Plattform bearbeiten, die Sie auch unter „General“ ausgewählt haben!

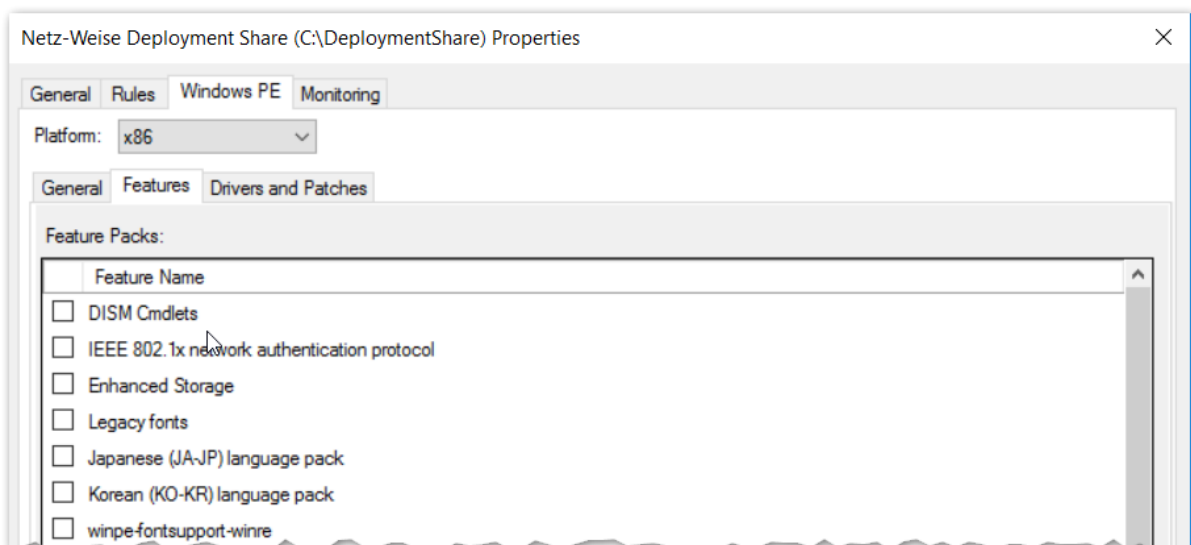
Sie können hier festlegen, welche Typen von Boot-Images ihnen das MDT erstellen soll. Sie können Boot-Images im Format ISO und WIM erstellen. Außerdem kann ein generisches (also nicht angepasstes) Windows PE für reine Wartungszwecke erstellt werden.

Wenn Sie nur vom Netzwerk aus booten, können Sie den Haken bei „Generate a Lite Touch bootable ISO image“ entfernen. Ein generisches Image ist zwar manchmal hilfreich, wird aber für das MDT nicht benötigt, sondern ist lediglich eine Zusatzfunktion.

Windows PE erstellt beim Starten eine RAM-Disk im Arbeitsspeicher und lädt sich dann in die RAM-Disk. Diese Disk hat den Laufwerksbuchstaben X: Wie groß die RAMDISK angelegt wird, hängt davon ab, wie groß Sie die Scratch Space Size wählen, die in MB angegeben ist. Wenn Sie Windows PE mit den Standardeinstellungen generieren, reichen 32 MB aus, je mehr zusätzliche Features Sie aber installieren, desto mehr Platz benötigt das Image. Das Maximum, das Sie für die Scratch Space Size einstellen können, ist 512 MB.



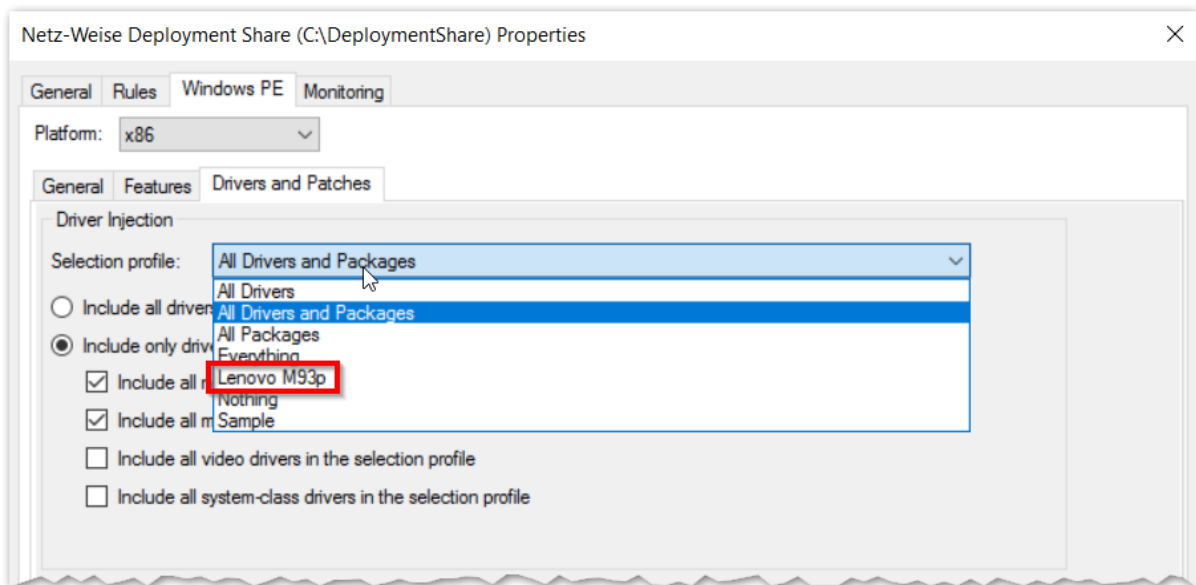
Unter Features können zusätzliche Funktionen aus dem Windows PE Paket in das Boot-PE aufgenommen werden. Für die Installation sind keine weiteren Features notwendig, aber eventuell für spätere Wartungsarbeiten oder wenn Sie ein generisches Windows PE erstellen lassen wollen.



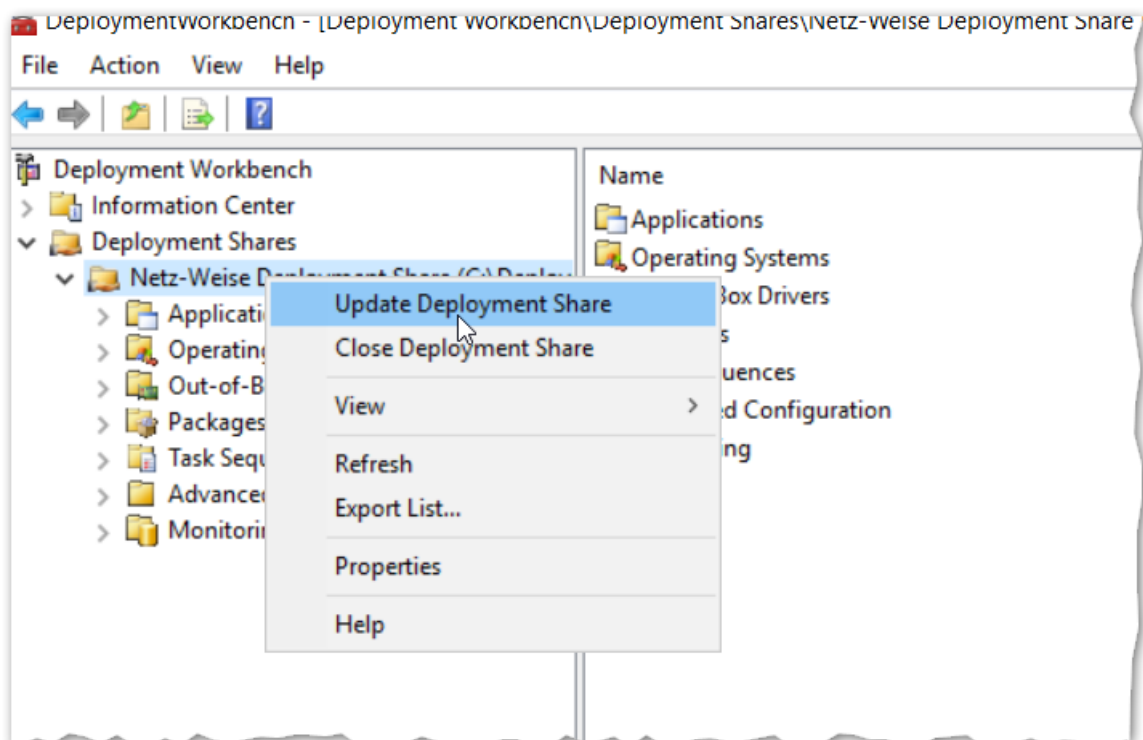
Auf dem letzten Eintrag „Drivers and Patches“ können Sie Treiber auswählen, die in das PE-Paket mit aufgenommen werden sollen. Das ist erfahrungsgemäß selten nötig. Wenn Sie jedoch über exotische

Netzwerkarten oder Festplatten verfügen, können Sie diese hier hinzufügen. Über das Selection profile können Sie festlegen, welche Treiber in das PE-Image aufgenommen werden sollen.

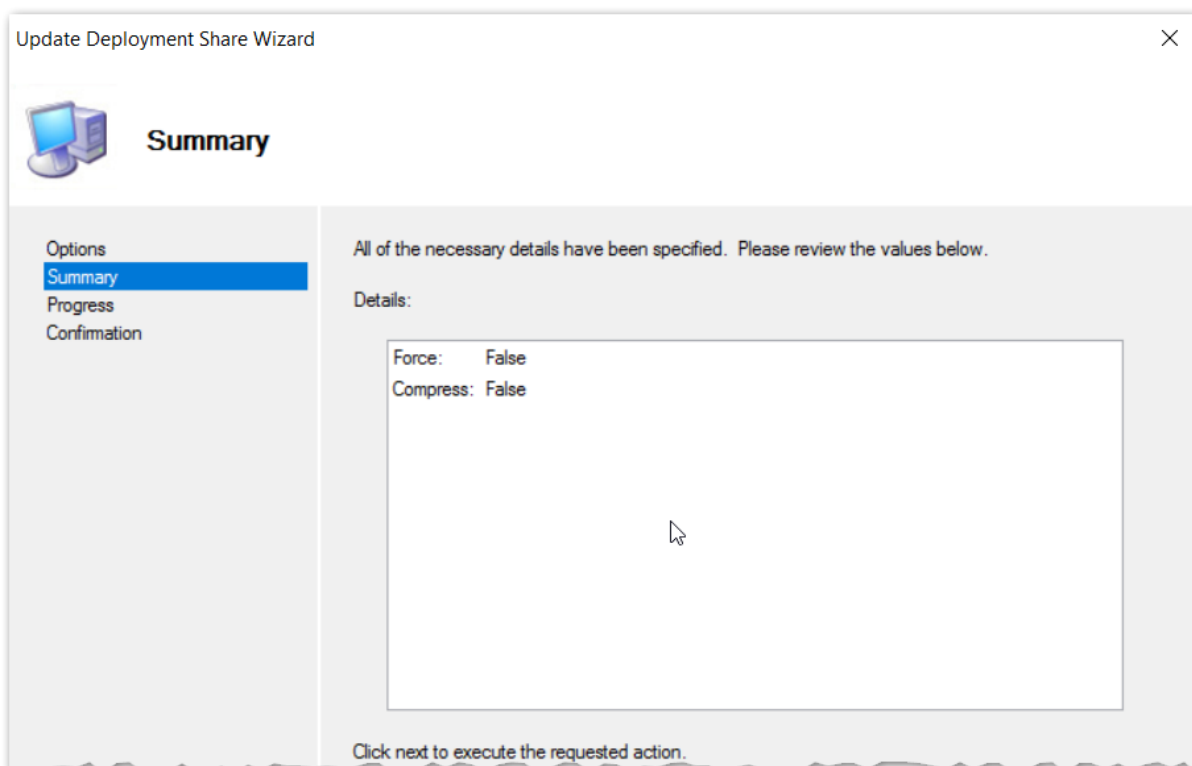
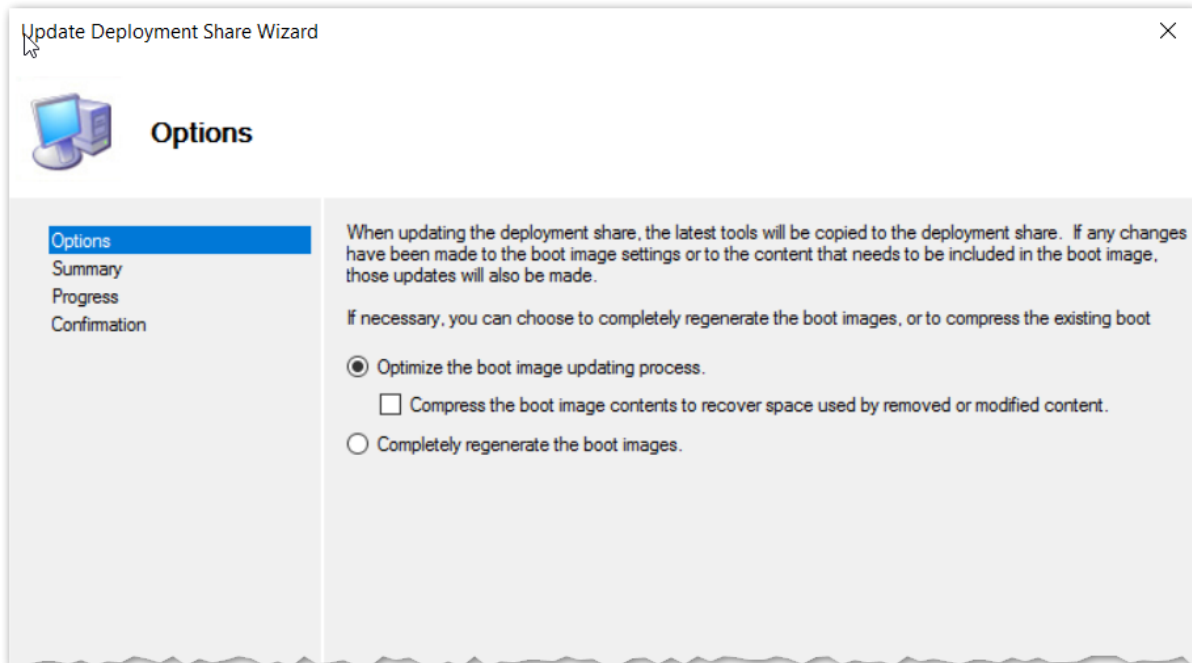
Die Standard-Profile sehen hier nur alles oder nichts vor und bedienen sich aller Treiber, die Sie dem Deployment-Share hinzugefügt haben. Das ist zwar nicht grundsätzlich schädlich, sorgt aber dafür, dass beim Erstellen des PE-Images alle Treiber aus dem Deployment-Share ins Windows PE injiziert werden. Das vergrößert das Image unnötig und dauert außerdem sehr lange. Erstellen Sie also am besten, wie oben bereits für die Lenovo M93p Geräte, ein eigenes Selection profile für Windows PE, dass nur die Treiber beinhaltet, die benötigt werden.



Wählen Sie nun aus dem Kontextmenü des Deployment-Shares den Eintrag „Update Deployment Share“. Korrekterweise müsste dieser Eintrag eigentlich „Create Windows PE“ heißen, denn das ist genau das, was passiert, wenn Sie diesen Menüpunkt aufrufen.



Sie können entweder ein bestehendes Image anpassen (Optimize the boot image updating Process), oder ein komplett neues Image erzeugen. Da noch kein Image existiert, wird MDT trotz der Auswahl „Optimize“ ein neues Image generieren. Ansonsten würde das bestehende Image gemounted und aktualisiert. Das geht, vor allem bei großen Images mit vielen installierten Features, deutlich schneller.



Nachdem das Image erzeugt wurde, finden Sie es im Ordner „Boot“ des Deployment-Shares.



Confirmation

- Options
- Summary
- Progress
- Confirmation**



The process completed successfully.

```
=== Making sure the deployment share has the latest x86 tools ===  
=== Processing LiteTouchPE (x86) boot image ===  
Building requested boot image profile.  
Determining if any changes have been made in the boot image configuration.  
No existing boot image profile found for platform x86 so a new image will be created.  
Calculating hashes for requested content.  
Changes have been made, boot image will be updated.  
Windows PE WIM C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows  
WIM file mounted.  
Set Windows PE system root.  
Set Windows PE scratch space.  
Added component winpe-hta  
Added component winpe-scripting  
Added component winpe-wmi  
Added component winpe-securestartup  
Added component winpe-fmapi  
Added component winpe-mdac  
Copy: C:\DeploymentShare\Control\Bootstrap.ini to C:\Users\ADMINI~1.NET\AppData\Local\Temp\
```

Save Output...

View Script

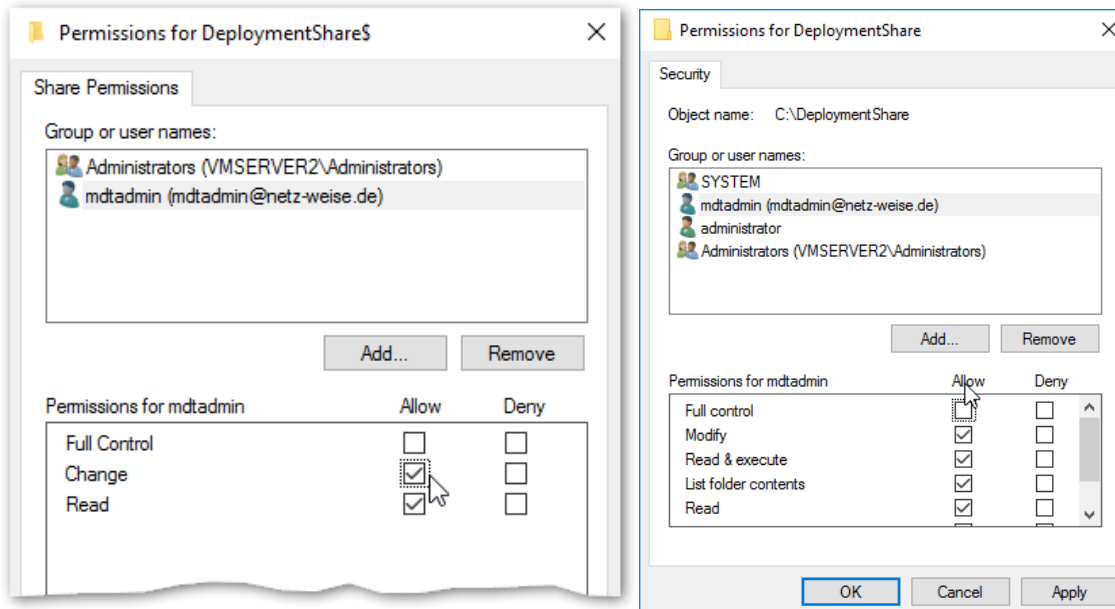
Previous

Finish

Cancel

Berechtigungen vergeben

Um zu steuern, wer auf das Deployment-Share zugreifen (und damit einen Rechner neu installieren darf), können einfach die Zugriffsberechtigungen des Deployment-Shares angepasst werden. Erstellen Sie hierfür am Besten einen Benutzer mdtadmin in Ihrer Domäne oder berechtigen Sie z.B. Ihre Unternehmensadministratoren. Passen Sie jetzt die Berechtigungen so an, dass der oder die Benutzer Schreib- und Leserechte auf der Freigabe „Deploymentshare\$“ und dem Ordner „Deploymentshare“ haben.



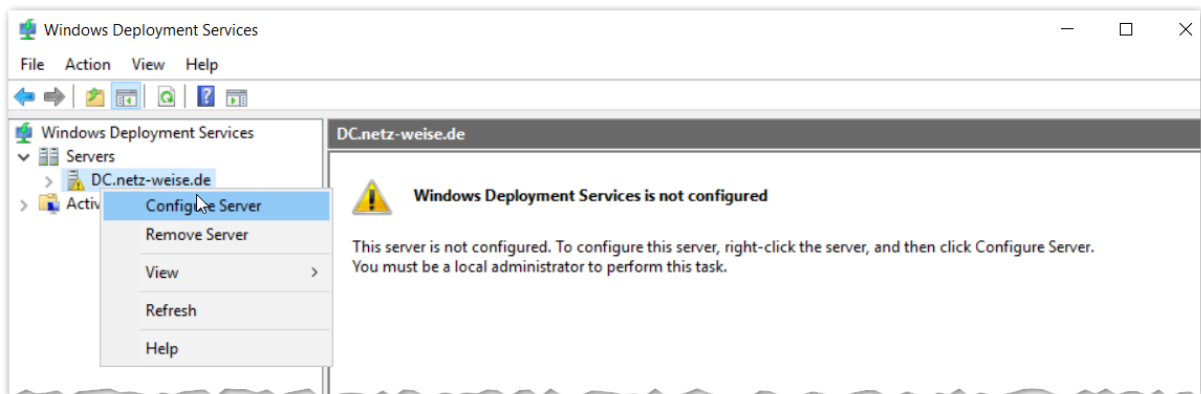
Windows PE per WDS aus dem Netzwerk booten

Um Windows PE über PXE (Netzwerkboot) starten zu können, wird ein WDS-Server benötigt. WDS (Windows Deployment Services) fungiert als DHCP-Proxy und TFTP-Server. Der DHCP-Proxy stellt dem PXE-Client beim Booten den Pfad zum Bootimage zur Verfügung, während der TFTP-Server das Boot-Image selber bereithält.

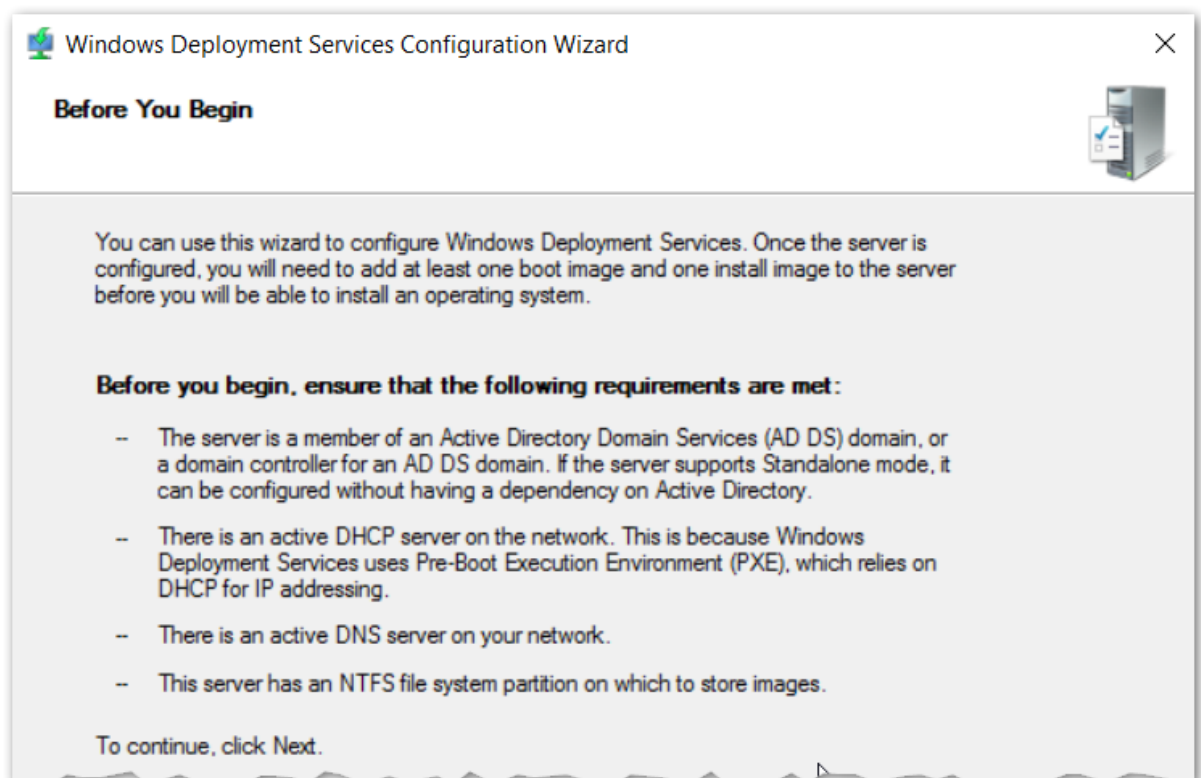
Installieren Sie zuerst den WDS-Dienst. Am schnellsten geht das über eine administrative Powershell-Konsole:

```
Add-WindowsFeature -Name WDS -IncludeAllSubFeature -IncludeManagementTools
```

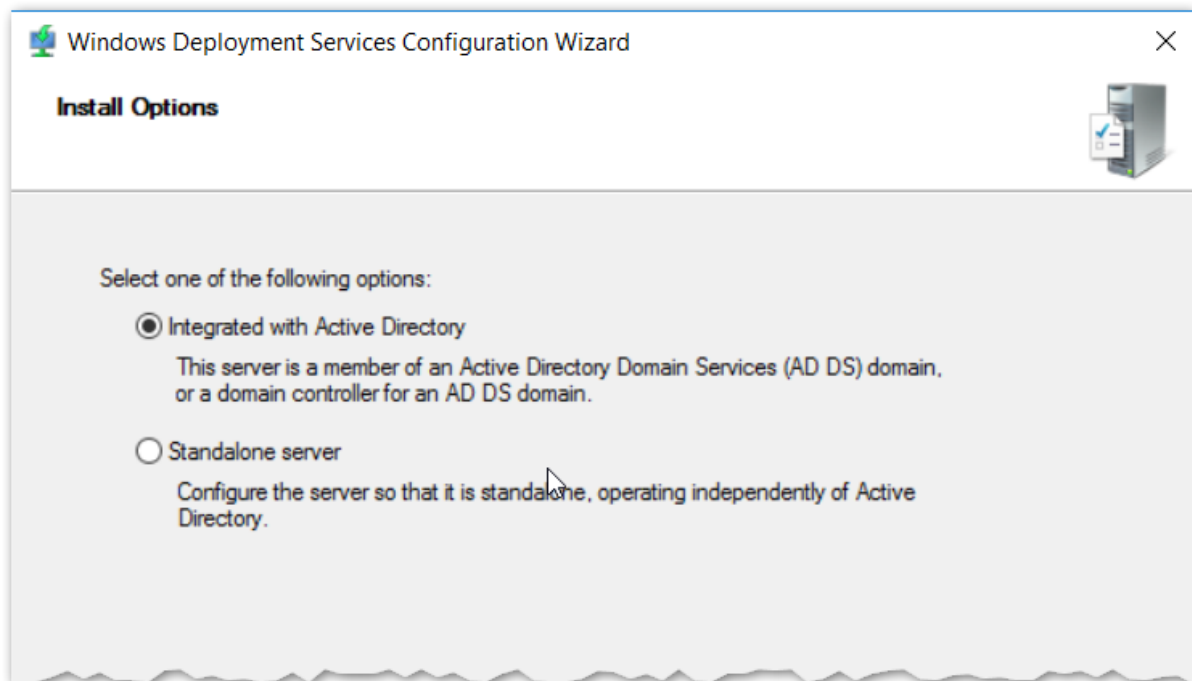
Nachdem die Installation abgeschlossen ist, öffnen Sie die Windows Deployment-Services Konsole und starten Sie die Konfiguration über „Configure Server“ im Kontextmenü des WDS-Servers.



Klicken Sie auf weiter, nachdem Sie überprüft haben, ob Sie einen DHCP-Server verfügbar haben. Am einfachsten ist die Konfiguration, wenn Sie den DHCP-Server und den WDS auf demselben Server installieren.



Sie können WDS AD-integriert oder Stand-Alone installieren. Der Unterschied zwischen den beiden Installationen besteht darin, wie Sie auf anfragende Clients reagieren können. Wenn der Server Mitglied einer Domäne ist, wählen Sie am besten immer „Integrated with Active Directory“.



Als nächstes müssen Sie angeben, wo WDS die Installationsdateien ablegen soll. Da der WDS nur Boot-Images von Windows PE verteilen soll, ist weder der Platzbedarf noch die Festplattenlast besonders groß. Daher können Sie die folgende Warnung, falls Sie auf dem Systemlaufwerk installieren, getrost ignorieren.

Remote Installation Folder Location

The remote installation folder will contain boot images, install images, PXE boot files, and the Windows Deployment Services management tools. Choose a partition that is large enough to hold all of the images that you will have. This partition must be an NTFS partition and should not be the system partition.

Enter the path to the remote installation folder.

Path:

System Volume Warning

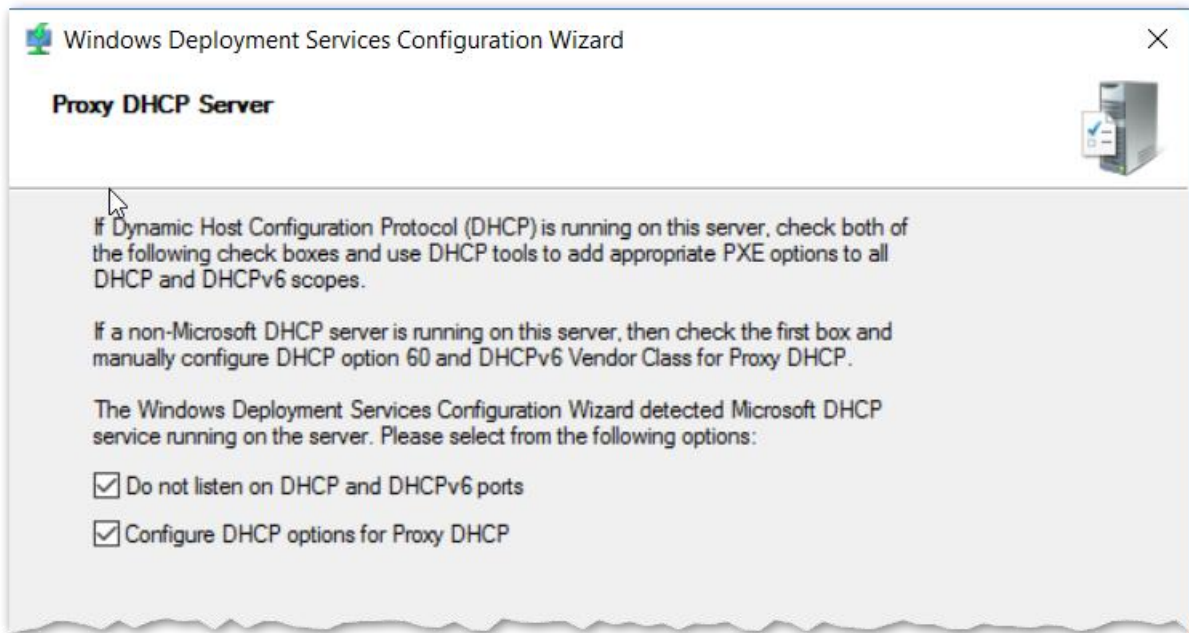


The volume selected is also the Windows system volume. For best performance and data reliability, the remote installation folder should be stored on a separate volume, and, where possible, on a separate disk from the system volume.
Do you want to continue?

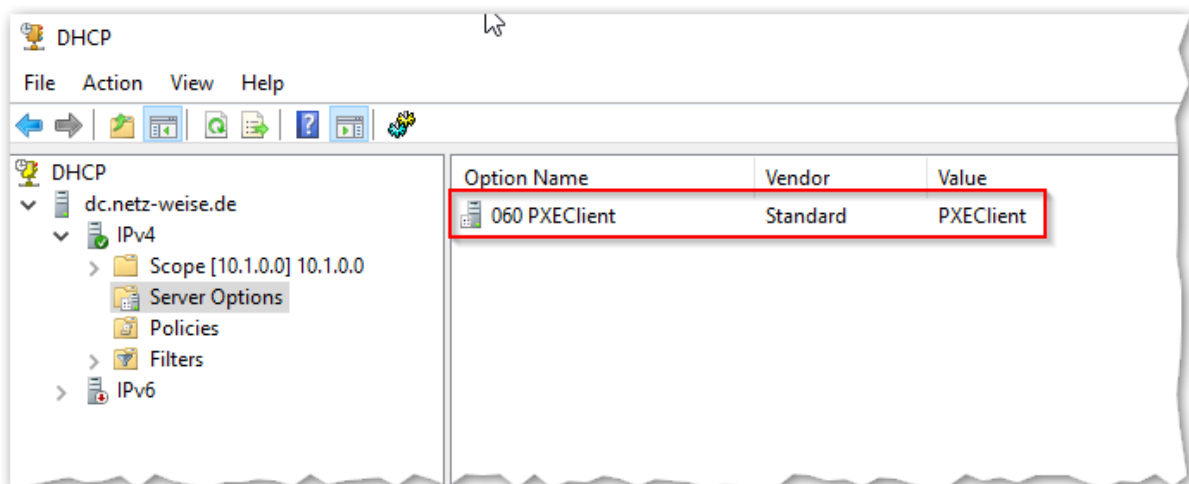
Die nun folgende Abfrage ist wichtig. Der WDS-Server fungiert als DHCP-Proxy. Das bedeutet, dass er wie ein DHCP-Server Daten an anfragende Clients ausgibt, allerdings keine IP-Adressen, sondern nur den Pfad zu einem Boot-Laufwerk. Da ein DHCP-Proxy normalerweise die gleichen Ports verwendet wie ein DHCP-Server, sorgt der Wizard automatisch dafür, dass WDS andere Ports verwendet, wenn er parallel zu einem DHCP-Server installiert wird.

Außerdem müssen die Clients instruiert werden, das angebotene Bootfile des DHCP-Proxys auch zu verwenden. Dies geschieht, indem der DHCP-Server die DHCP-Option 60 zusammen mit der IP-Adresse sendet. Wird der WDS auf dem DHCP-Server installiert, kann der Wizard das automatisch erledigen (Configure DHCP-Options for Proxy DHCP). Wird der WDS auf einem eigenständigen Server

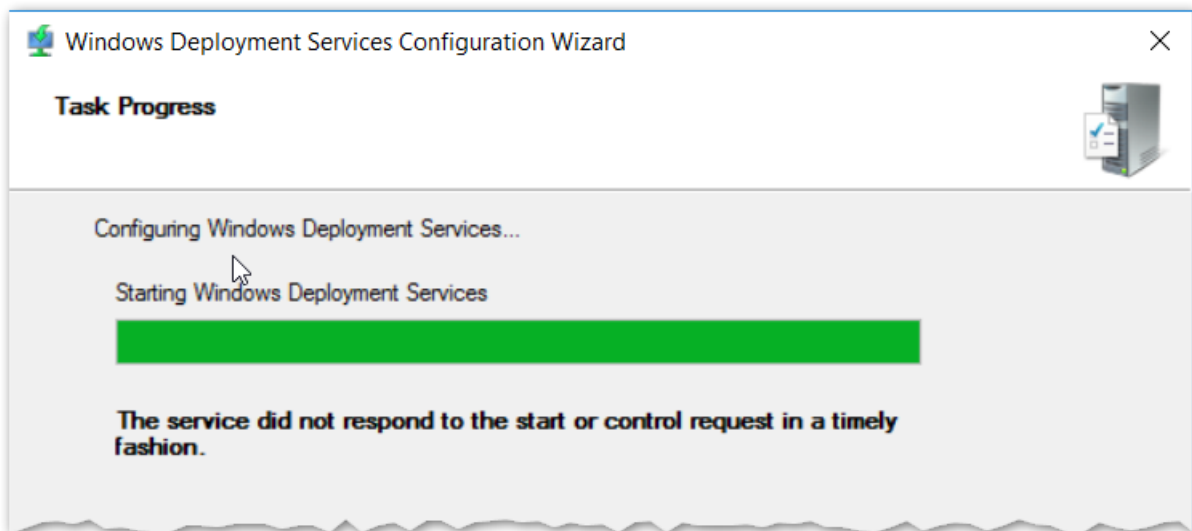
installiert, muß die Option 60 manuell gesetzt werden (s.u.).



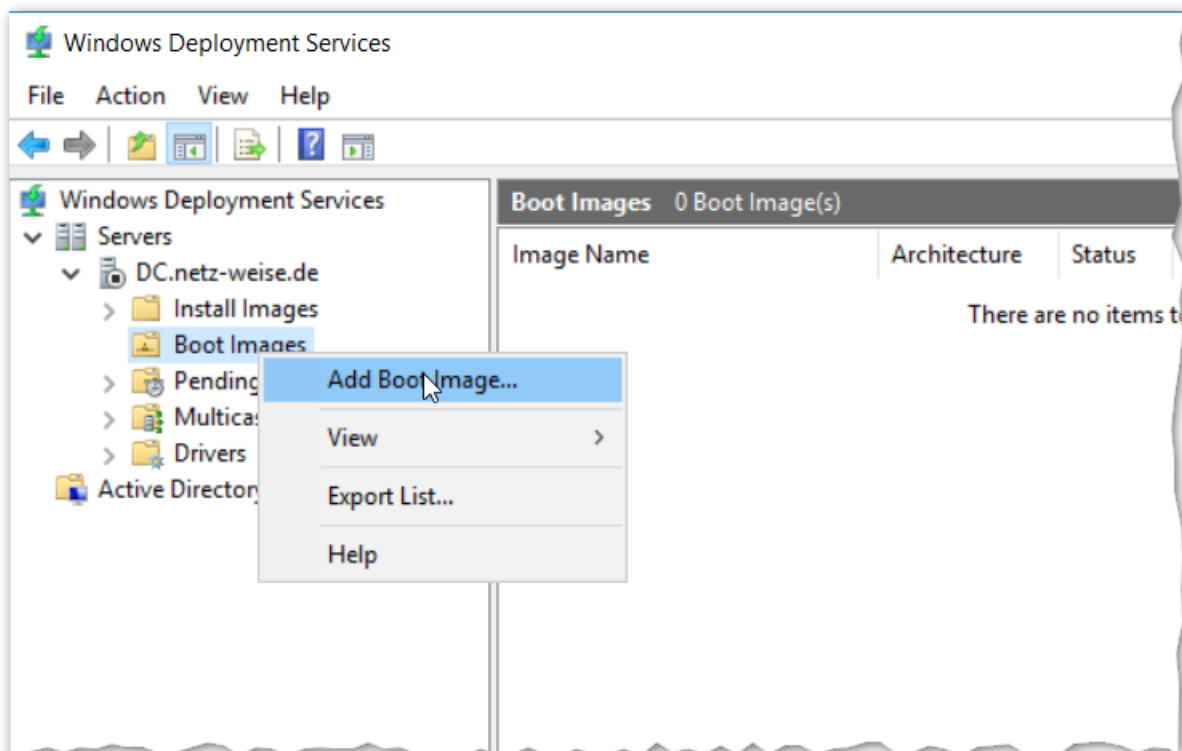
Die DHCP-Option 60 wird im Server-Scope angegeben.



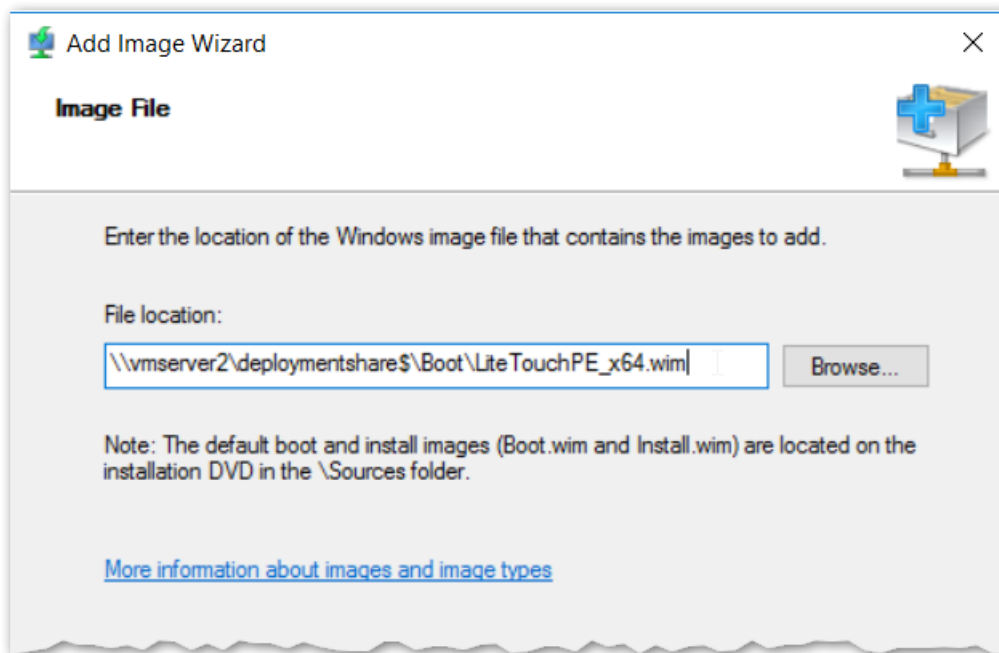
Beim ersten Start des WDS-Servers kommt es oft zu einer Fehlermeldung, dass der Dienst nicht geantwortet hat. Das können Sie ignorieren, der Dienst kann später ohne Probleme manuell gestartet werden.



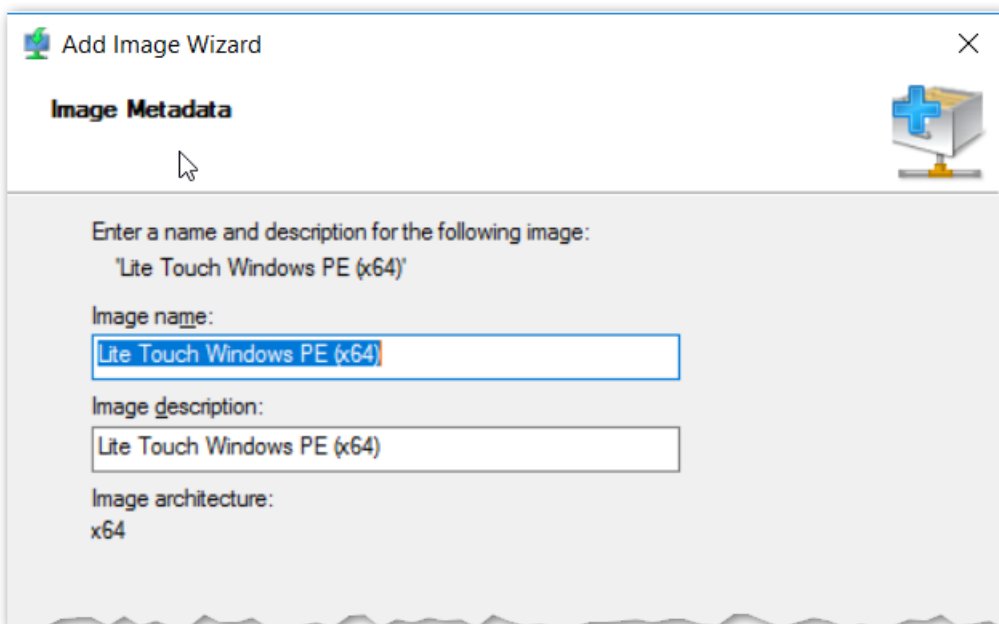
Nachdem der WDS installiert ist, muss nun das PE-Bootimage zum WDS hinzugefügt werden. Wählen Sie hierfür im Kontextmenü des Eintrags „Boot Images“ „Add Boot Image“ aus.

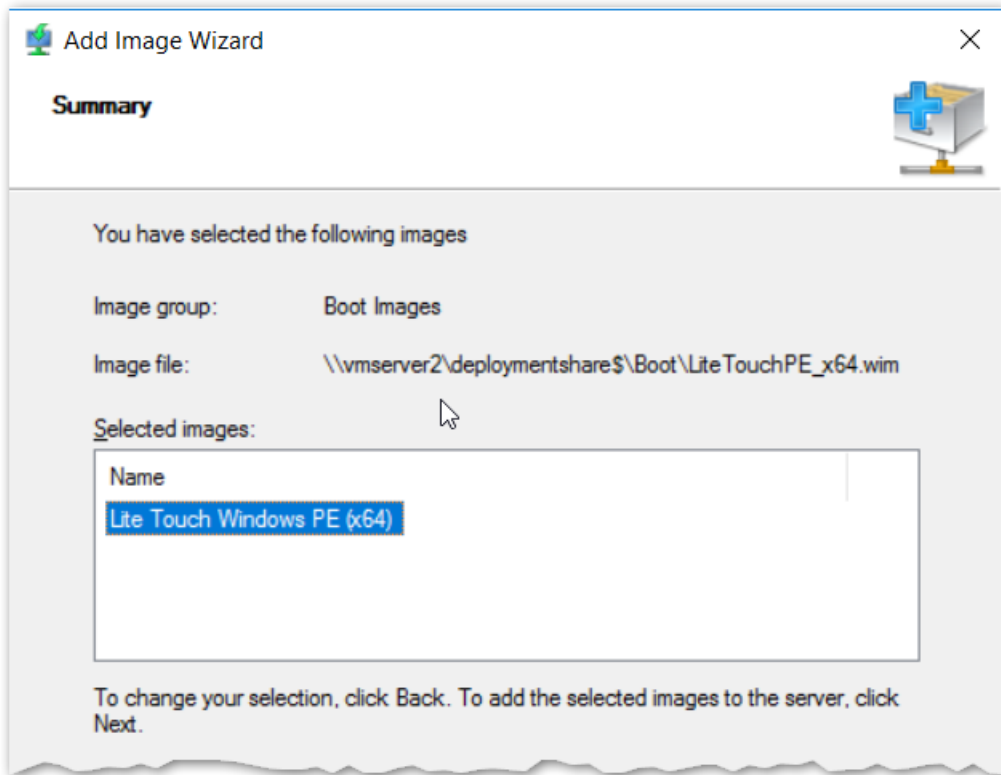


Das Boot-Image wurde von MDT bereits erstellt. Verbinden Sie sich über das Netzwerk mit der Deploymentshare-Freigabe und wählen Sie aus dem Ordner Boot das LitetouchPE-wim-File aus.

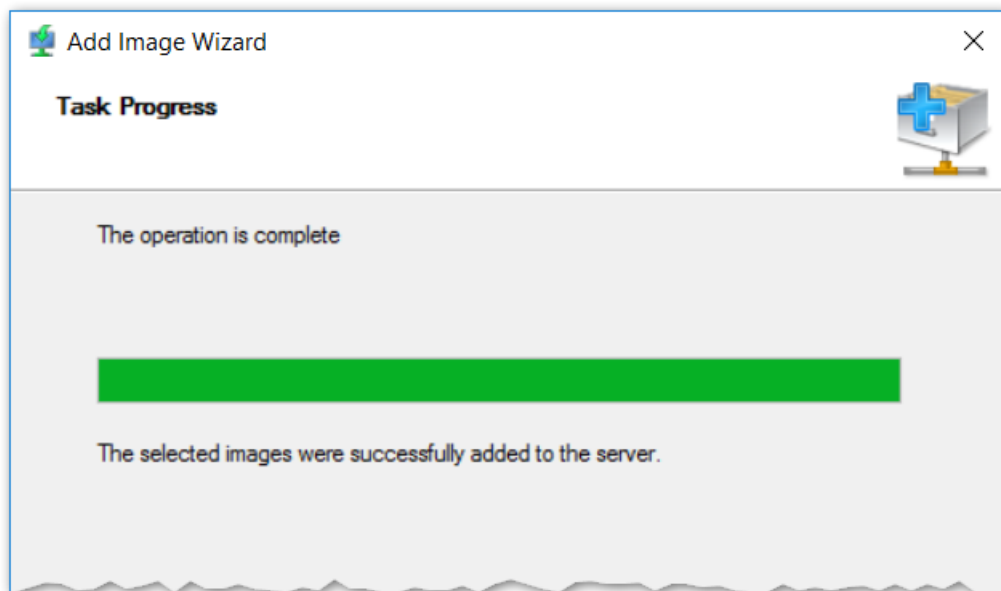


Geben Sie einen Namen für das Boot-Image an.

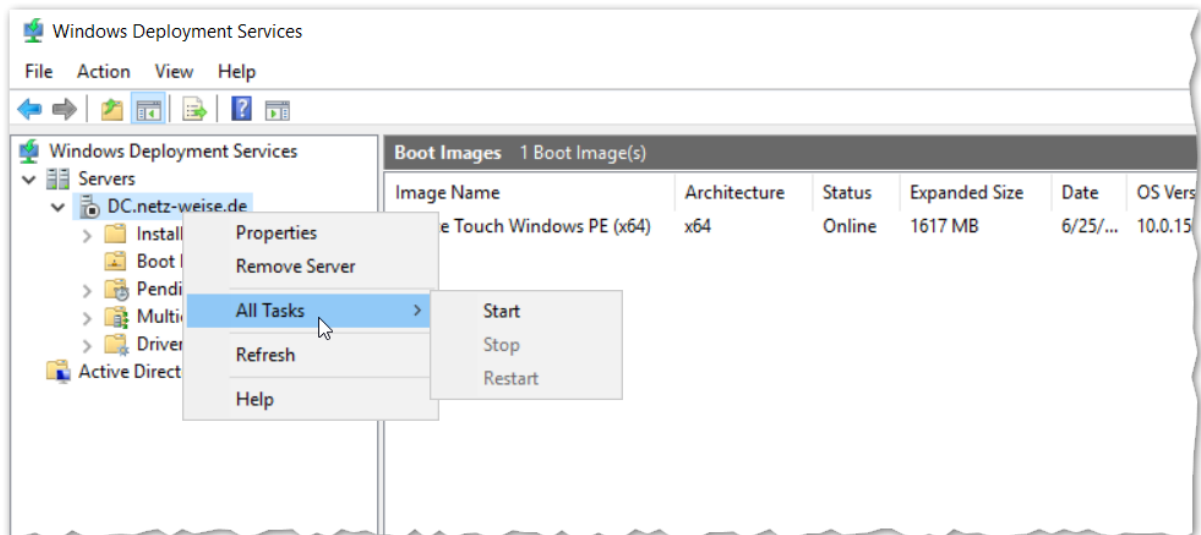




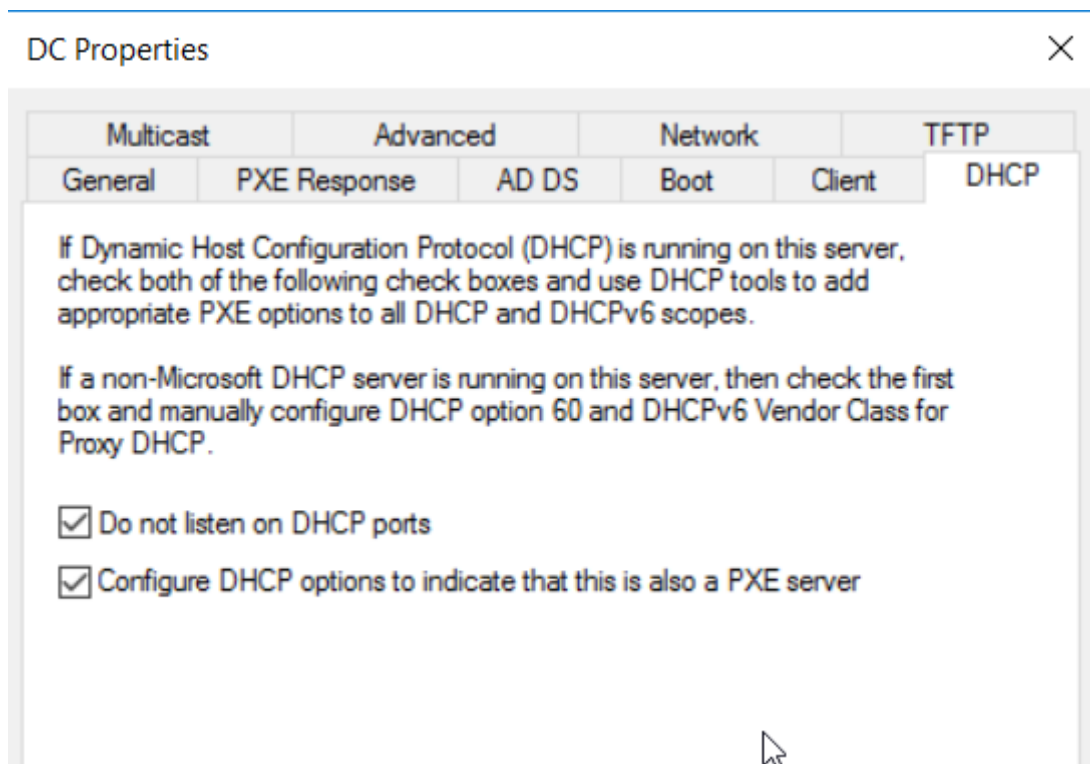
Beenden Sie den Assistenten. Das Image wird dem WDS nun hinzugefügt.



Starten Sie jetzt den WDS-Server.



In den Eigenschaften des WDS-Servers können Sie die DHCP-Integration einsehen und ändern.



Installieren eines Clients

Wenn Sie den WDS aktiviert und das Boot-Image bereitgestellt haben, können Sie mit der Installation eines Clients beginnen. Achten Sie darauf, dass im Bios des Clients das Booten vom Netzwerk aktiviert ist.

Starten Sie den Client. Je nach Bios-Version müssen Sie die Installation mit F12 oder der Enter-Taste initiieren. Wenn Sie die Eingabe-Aufforderung nicht erhalten, sollten Sie testen, ob Ihr DHCP-Server und Ihr WDS-Server aktiviert und erreichbar sind.

```
WDS Boot Manager version 0800
Client IP: 10.1.100.1
Server IP: 10.1.2.200
Server Name: DC.netz-weise.de

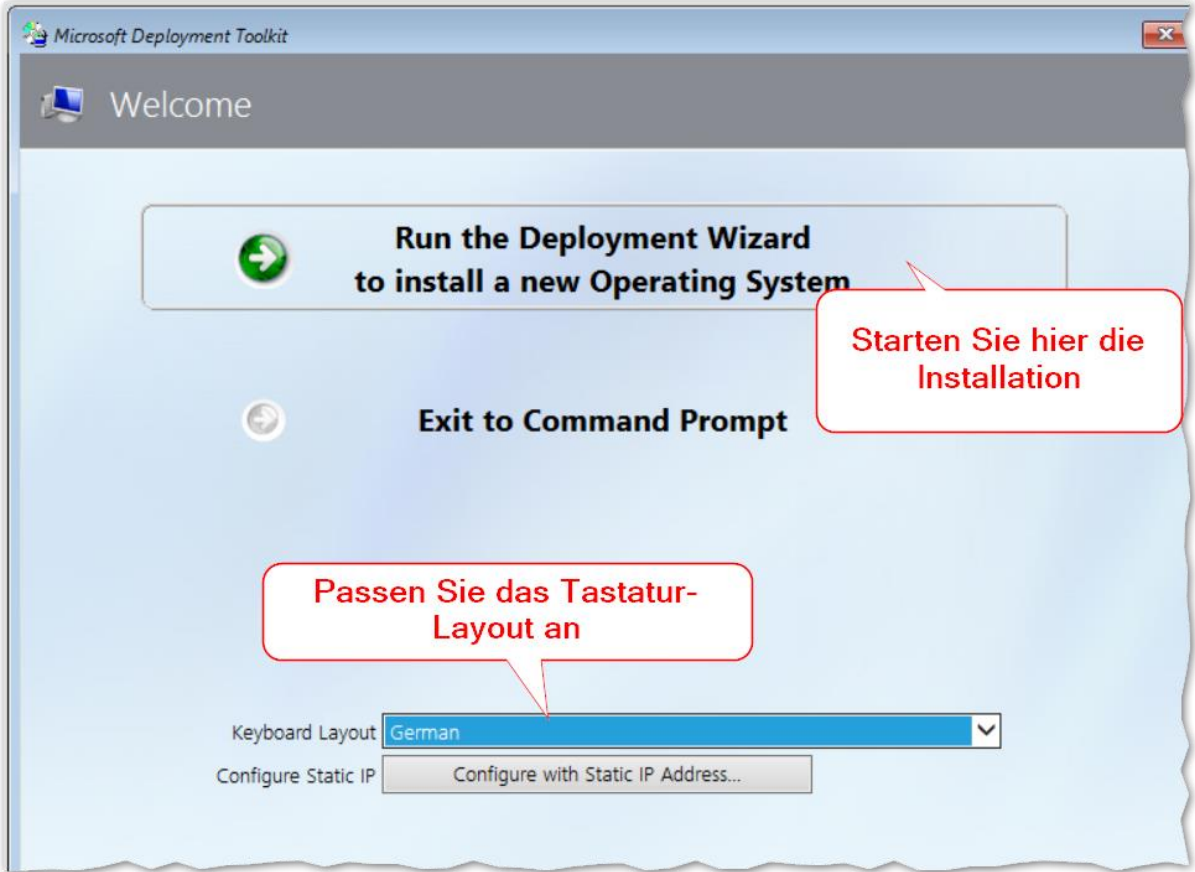
Press ENTER for network boot service.
```

Beim Laden wird der Pfad zum Boot-Image auf dem WDS-Server angezeigt.

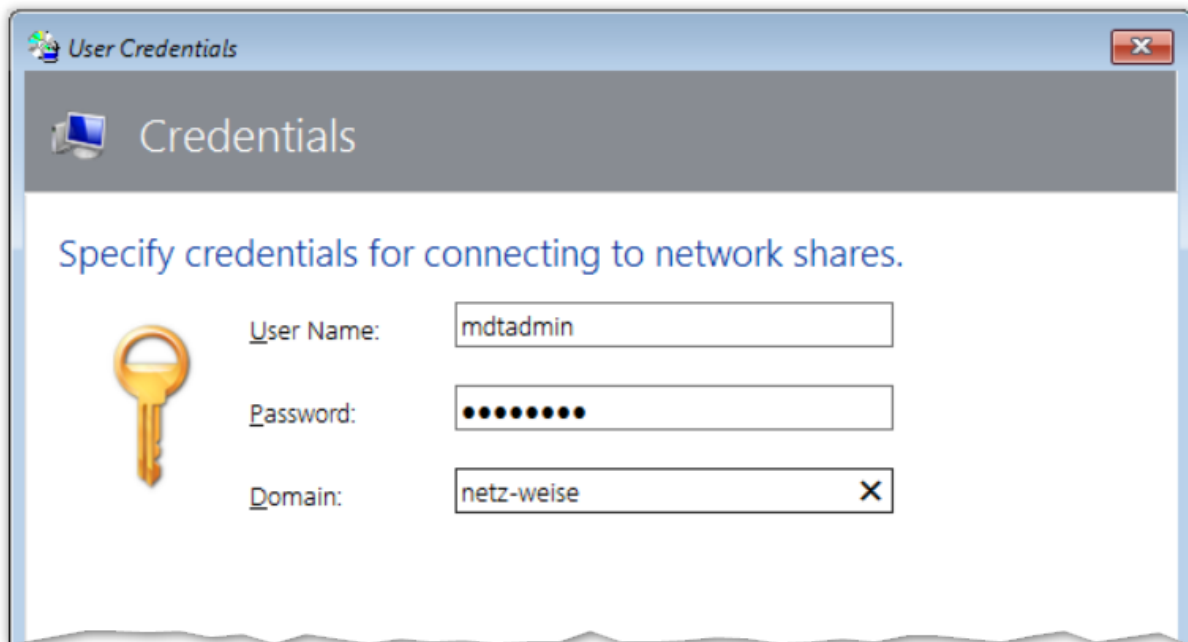
```
Loading files...

IP: 10.1.2.200, File: \Boot\x64\Images\LiteTouchPE_x64.wim
```

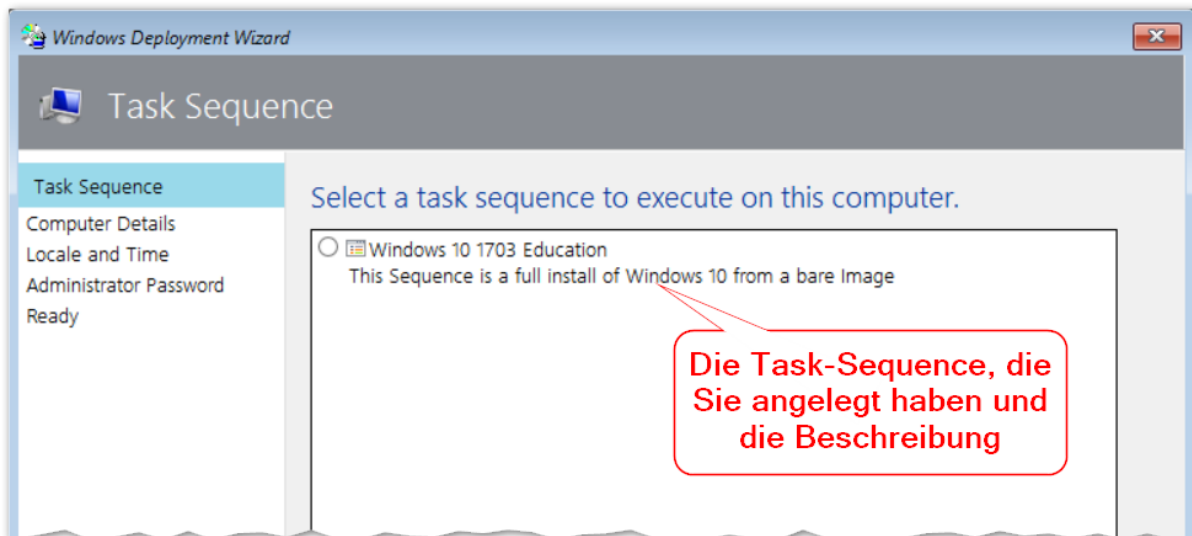
Passen Sie die Tastatur-Einstellungen an und starten Sie anschließend den Deployment-Wizard.



Nun müssen Sie sich am Deployment-Share anmelden. Der Zugriff wird hierbei sehr einfach gesteuert – jeder Benutzer mit Zugriffsrechten (Lesen und für das Erstellen neuer Images und das Hochladen von Logfiles auch Schreiben) auf das Deployment-Share kann eine Installation ausführen.

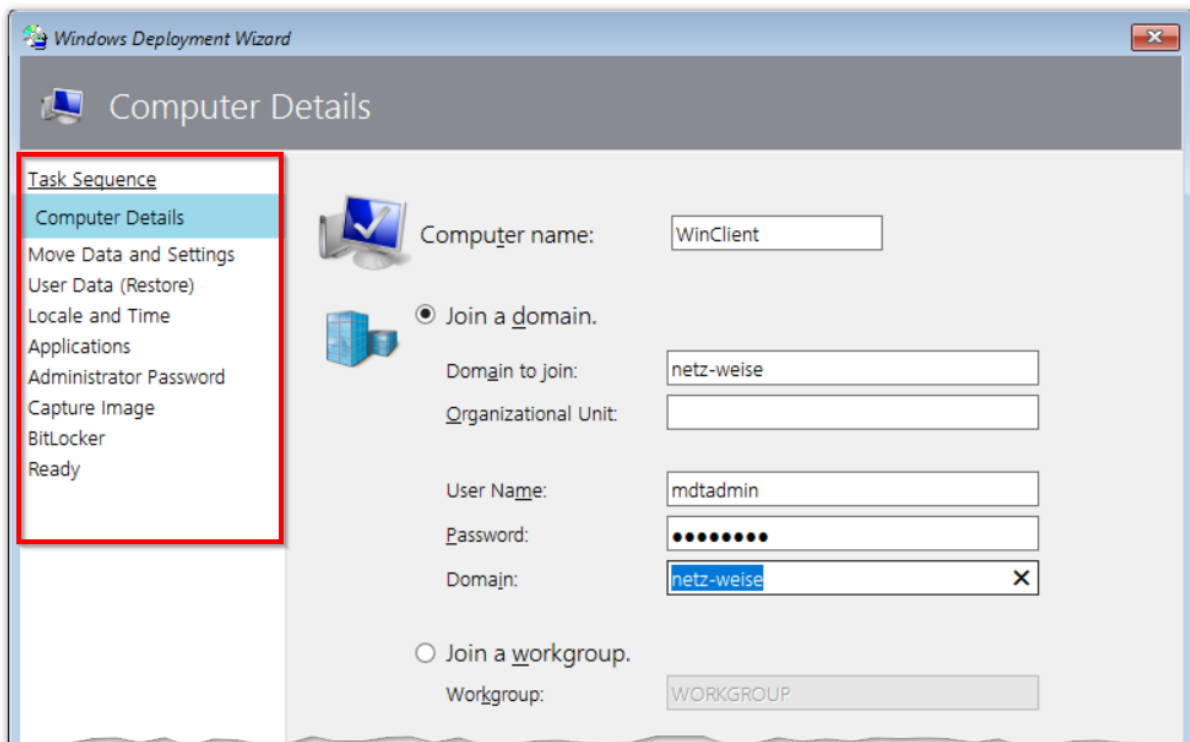


Nach der Anmeldung müssen Sie zuerst eine Task-Sequence auswählen. Hier sehen Sie die Liste aller Task-Sequences, die Sie angelegt haben.

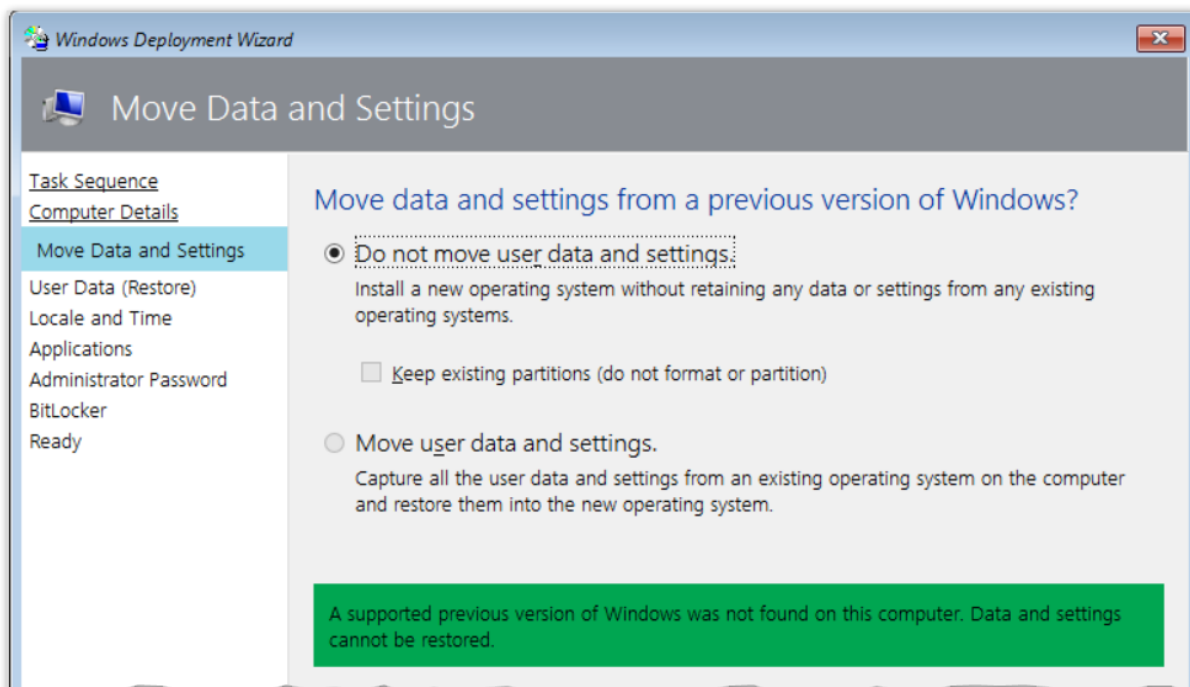


Nach der Auswahl der Task-Sequence müssen Sie eine Reihe von Informationen über den neuen Computer angeben. Welche Auswahlfenster Sie hier angezeigt bekommen (s. Kasten in der Abbildung), haben Sie bei der Installation des MDT festgelegt. Wie Sie diese Liste anpassen können, erfahren Sie im nächsten Abschnitt, Anpassen des Deployment-Shares.

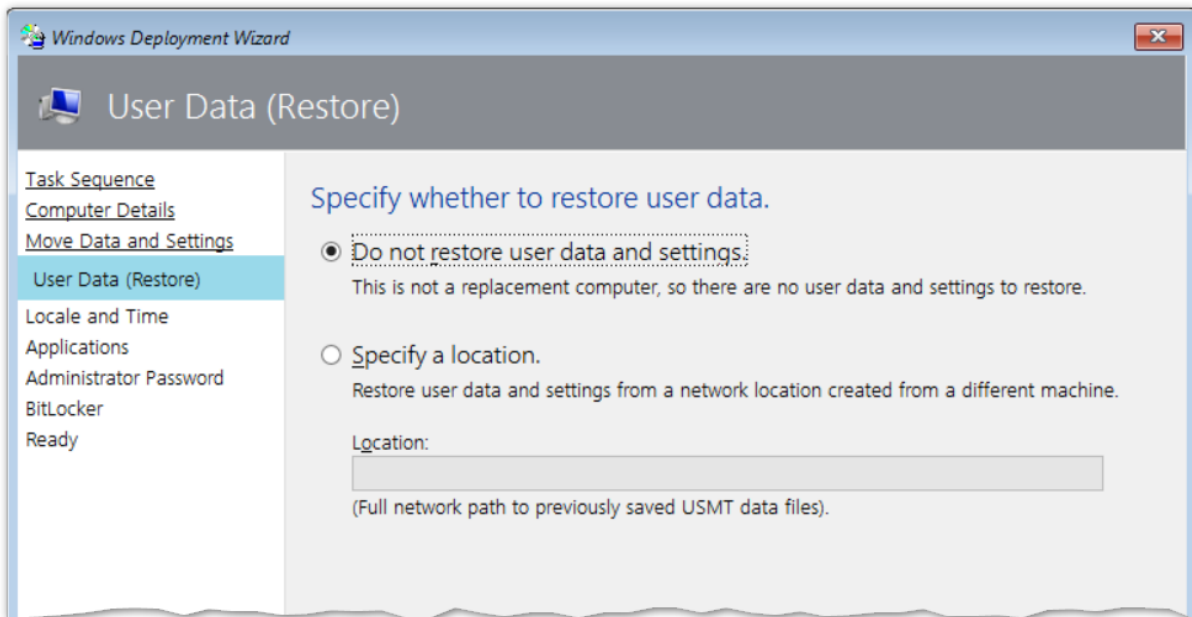
Geben Sie nun den Namen des neuen Clients ein. Außerdem können Sie den Computer nach der Installation einer Domäne beitreten lassen.



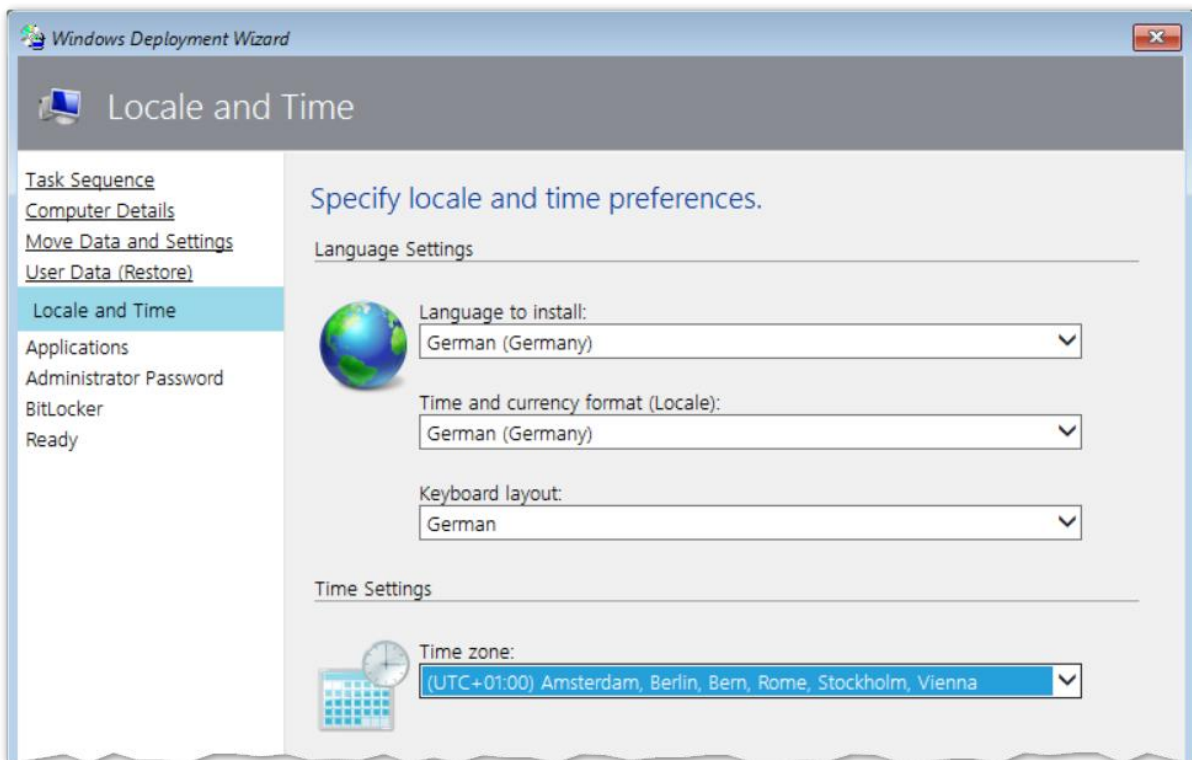
MDT unterstützt auch verschiedene Migrations-Szenarien. Move Data and Settings verwendet das USTM (User State Migration Tool) zum Sichern von Benutzerdaten. Da die im Beispiel verwendete Maschine eine leere Festplatte besitzt, kann der Assistent auch keine Daten migrieren. Ansonsten können Sie „Move User data and Settings“ auswählen und die Benutzerdaten auf einen Fileserver kopieren lassen, von wo aus MDT die Daten nach der Installation wieder importieren kann.



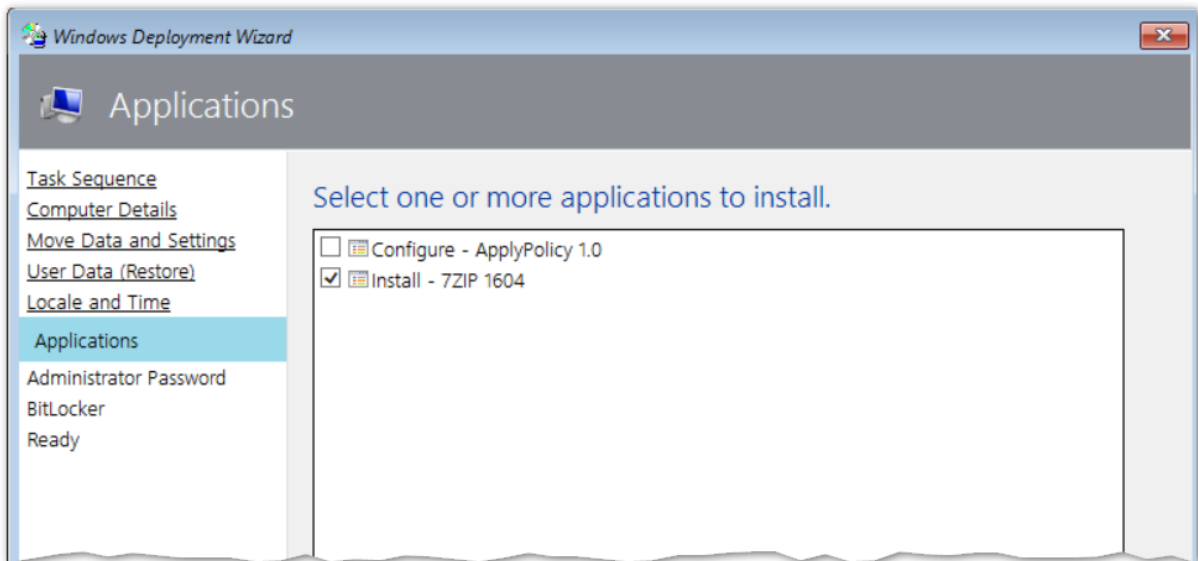
Wenn Sie die Daten von einem bestehenden Rechner bereits im Vorfeld durch den vorigen Vorgang gesichert hatten, können Sie jetzt die Freigabe angeben, von der Benutzerdaten migriert werden sollen.



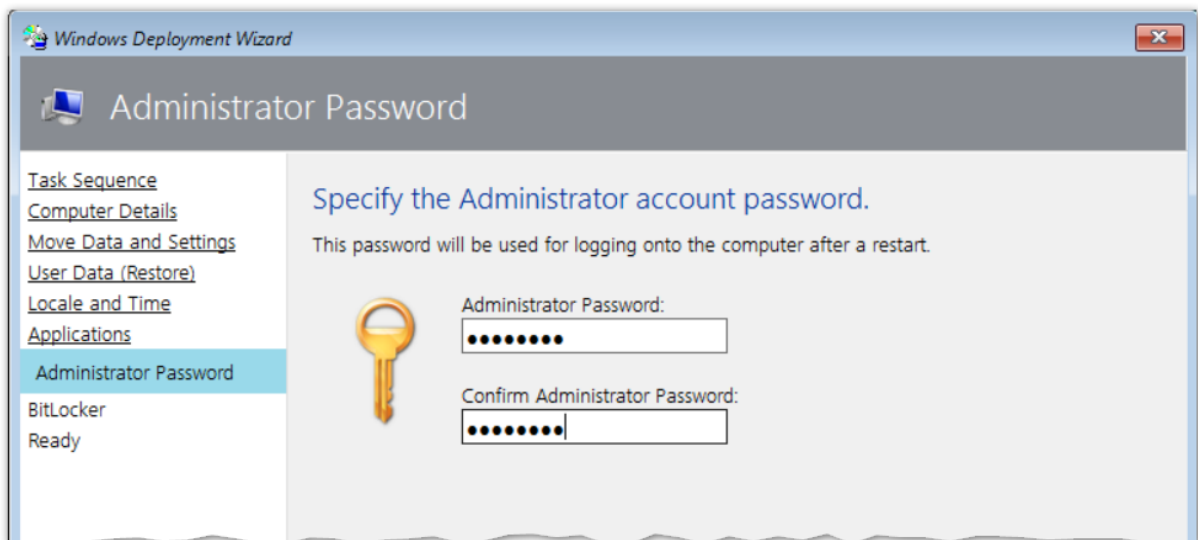
Passen Sie Sprache, Tastatur und Zeitzone des zu installierenden Clients an.



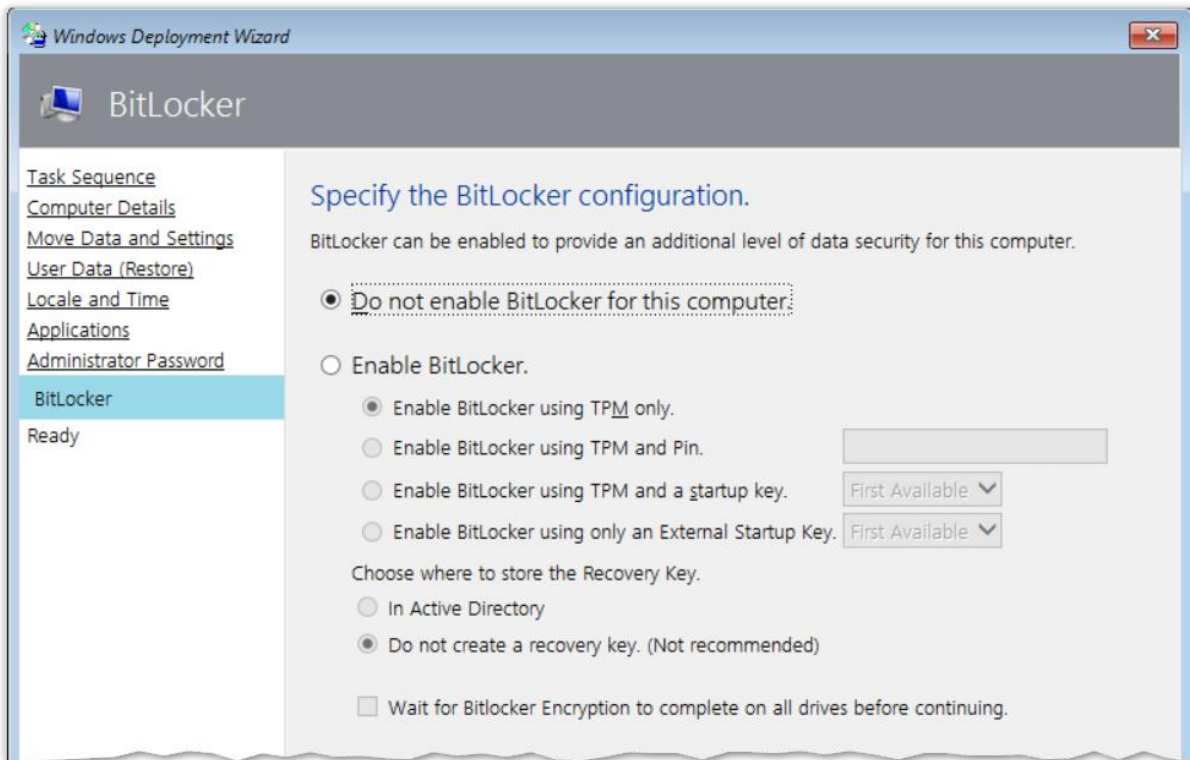
Nun werden die Anwendungen angezeigt, die Sie dem Deployment-Share hinzugefügt haben. Wählen Sie die Anwendungen aus, die nach der Betriebssysteminstallation installiert werden sollen. Da dieser Teil der Konfiguration einfach nur Programme angibt, die gestartet werden, können Sie hierüber auch z.B. Konfigurationsskripte starten. Im Beispiel wurde ein Skript zum Hinzufügen von Richtlinien angelegt. Dies macht die Anwendungsinstallation besonders flexibel. Hier sehen Sie, dass es wichtig ist, die Anwendungen systematisch zu benennen.



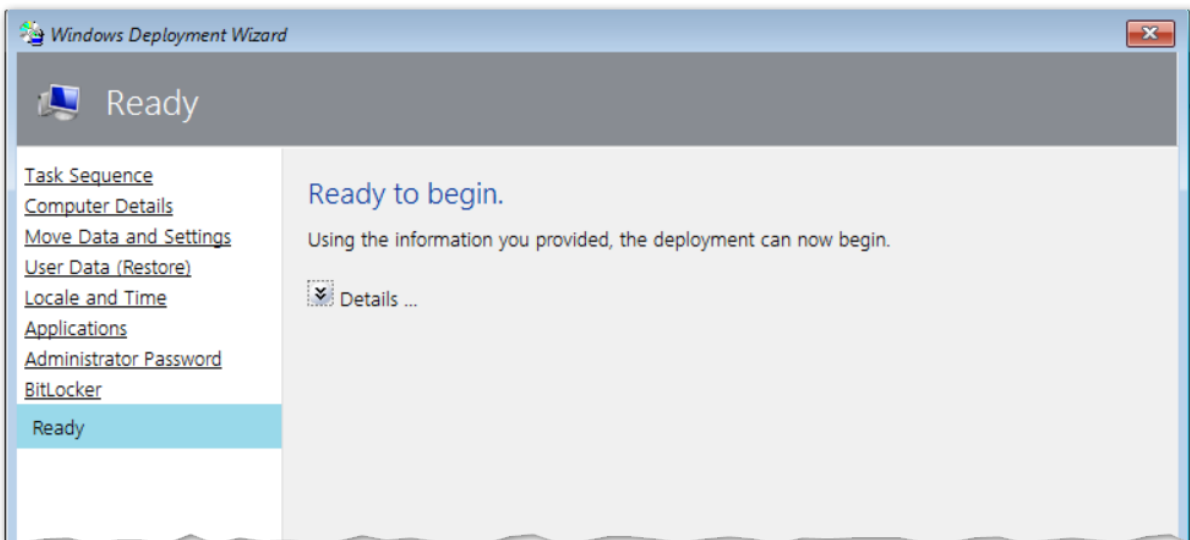
Geben Sie nun das Kennwort an, dass für den lokalen Administrator verwendet werden soll.



Wenn Sie möchten, können Sie Bitlocker, eine in Windows integrierte Festplattenverschlüsselung, nach der Installation aktivieren.

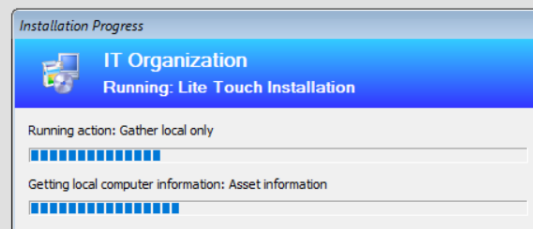


Starten Sie jetzt die Installation.



Der Assistent zeigt Ihnen an, was er während der Installation tut.

Microsoft Deployment Toolkit



Anpassen von Deploymentshare und Windows PE

Bisher haben Sie die Standard-Einstellungen des Deployment-Shares verwendet, um die Installation durchzuführen. Mit angepassten Einstellungen können Sie MDT aber noch deutlich flexibler gestalten.

Bootstrap.ini

Änderungen in der Bootstrap.ini legen die Konfiguration fest, die vor der Verbindung mit dem Deploymentshare gestartet wird. Änderungen, die Sie in der Bootstrap.ini vornehmen, erfordern einen Refresh des Windows PE-Files, den Sie über den Menüpunkt „Update Deployment Share“ aufrufen können.

Eine angepasste Bootstrap.ini könnte z.B. folgende Einträge beinhalten:

```
[Settings]
Priority= Default

[Default]
DeployRoot=\\MDTServer\DeploymentShare$
UserID=mdtAdmin
UserDomain=Netz-Weise
UserPassword=meinGeheimesPassw0rt
KeyboardLocalePE=0407:00000407
InputLocalePE=0407:00000407
SkipBDDWelcome=YES
```

DeployRoot legt den Pfad des Deploymentshares fest. Dieser Eintrag muss gesetzt sein.

UserID, Userdomain und UserPassword legen die Benutzerinformationen fest, mit denen die Anmeldung am Deploymentshare durchgeführt wird.

InputLocalePE und KeyboardLocalePE legen die Spracheinstellungen von Windows PE auf Deutsch fest.

SkipBDDWelcome=YES überspringt den ersten Anmeldebildschirm. Windows PE bootet damit direkt bis zur Tasksequenz durch.

CustomSettings.ini

Auch die CustomSettings.ini bietet einiges an Konfigurationspotential. CustomSettings.ini steuert den Windows PE Client nach der Anmeldung am Deploymentshare. Ein Update des Deploymentshares ist daher nicht notwendig, wenn in dieser Datei Änderungen vorgenommen werden.

```
[Settings]
Priority= Default

[Default]
SLSHARE=\\mdtserver\DeploymentShare$\Logs
_SMSTSORGNAME=Netz-Weise
SkipApplications=No
SkipAdminPassword=YES
SkipBDDWelcome=YES
SkipBitLocker=YES
SkipCapture=No
SkipComputerBackup=YES
SkipLocaleSelection=YES
SkipProductKey=YES
SkipRoles=YES
SkipTimeZone=YES
SkipUserData=YES
KeyboardLocale=de-DE
InputLocalePE=de-DE
UserLocale=de-DE
UILanguage=de-DE
TimeZoneName=W. Europe Standard Time
```

Die Einträge, die mit Skip anfangen, bezeichnen einzelne Fenster, die beim Konfigurieren der Installation nicht angezeigt werden sollen.

SLShare verweist auf eine Freigabe. Ist SLShare gesetzt, werden nach Abschluss der Installation alle Logfiles in einen Unterordner mit Namen des Clients hierhin hochgeladen.

TimeZoneName gibt die Zeitzone an. Wenn Sie eine andere Zeitzone konfigurieren wollen, können Sie den korrekten Namen der Zeitzone mit tzutil.exe /l auflisten lassen.

Eine vollständige Auflistung aller setzbaren Einträge finden Sie in der Toolkit-Referenz, die leider nicht besonders übersichtlich ist, unter <https://technet.microsoft.com/en-us/library/dn781091.aspx>.

Software und Konfigurationsskripte bereitstellen

Um mit MDT Anwendungen bereit zu stellen, müssen Sie zuerst herausfinden, wie Sie das Setup Ihrer Anwendung ohne Interaktion mit dem Benutzer (Silent) durchführen können. Kommt das Setup in Form einer MSI-Datei, ist das sehr simpel, denn MSI-Dateien beinhalten eine Reihe von Tabellen, die sämtliche Informationen enthalten, die für die Installation notwendig sind. Die Installation eines msi-Paketes können Sie von der Kommandozeile aus mit dem Tool `msiexec.exe` starten. `msiexec` benötigt für die unbeaufsichtigte Installation die Parameter `/i <msi-Datei> /qn`. Wollen Sie also z.B. die Datei `7zip.msi` installieren, lautet die Kommandozeile

```
msiexec.msi /i 7zip.msi /qn
```

`/qn` gibt dabei an, dass die Installation (q)uiet, also lautlos erfolgen soll, und `n` steht dafür, dass die Installation vollständig im Hintergrund ausgeführt wird. Für die Installation mit MDT ist es aber durchaus sinnvoll, wenn Sie sehen können, wie weit die Installation gerade fortgeschritten ist. Geben Sie `/qb` ein, so wird zumindest eine minimale Oberfläche angezeigt, die den Stand der Installation anzeigt. Eine vollständige Auflistung aller möglichen Parameter erhalten Sie, wenn Sie `msiexec` an der Kommandozeile ohne Parameter starten, oder im folgenden Kasten:

```
msiexec /Option <Erforderliche Parameter> [Optionale Parameter]

Installationsoptionen
  </package | /i> <Produkt.msi>
    Installiert oder konfiguriert ein Produkt.
  /a <Produkt.msi>
    Administrative Installation - Installiert ein Produkt im Netzwerk.
  /j<u|m> <Produkt.msi> [/t <Transformliste> [/g <Sprach-ID>]
    Kündigt ein Produkt an - m für alle Benutzer, u für den aktuellen
Benutzer
  </uninstall | /x> <Produkt.msi | Produktcode>
    Deinstalliert das Produkt.
Anzeigeoptionen
  /quiet
    Hintergrundmodus, keine Benutzerinteraktion
  /passive
    Unbeaufsichtigter Modus - Nur Statusleiste
  /q[n|b|r|f]
    Legt die Benutzeroberfläche fest.
    n - Keine Benutzeroberfläche
    b - Einfache Benutzeroberfläche
    r - Reduzierte Benutzeroberfläche
    f - Vollständige Benutzeroberfläche (Standard)
  /help
    Hilfeinformationen
Neustartoptionen
  /norestart
    Kein Neustart nach Abschluss der Installation
  /promptrestart
    Benutzereingabe, falls Neustart erforderlich ist
  /forcerestart
    Computer nach Abschluss der Installation immer neu starten

Protokollierungsoptionen
  /l[i|w|e|a|r|u|c|m|o|p|v|x|+|!|*] <Protokolldatei>
    i - Statusmeldungen
    w - Nicht schwerwiegende Warnungen
    e - Alle Fehlermeldungen
    a - Ausführung von Aktionen
    r - Aktionsspezifische Einträge
    u - Benutzeranforderungen
    c - Ursprüngliche Benutzeroberflächenparameter
```

```

    m - Informationen betreffend zuwenig Arbeitsspeicher oder
        schwerwiegendem Abbruch
    o - Meldungen betreffend zuwenig Speicherplatz
    p - Terminaleigenschaften
    v - Ausführliche Ausgabe
    x - Zusätzliche Debuginformationen
    + - An vorhandene Protokolldatei anhängen
    ! - Jede Zeile ins Protokoll aufnehmen
    * - Alle Informationen mit Ausnahme der Optionen v und x
        protokollieren
/log <Protokolldatei>
    Entspricht /l* <Protokolldatei>

Aktualisierungsoptionen
/update <Update1.msp>[;<Update2.msp>]
    Übernimmt Update(s).
/uninstall <Patchcode-GUID>[;<Update2.msp>] /package <Produkt.msi |
Produktcode>
    Entfernt Update(s) für ein Produkt.

Reparaturoptionen
/f[p|e|c|m|s|o|d|a|u|v] <Produkt.msi | Produktcode>
    Repariert ein Produkt.
    p - nur wenn eine Datei fehlt
    o - wenn eine Datei fehlt oder eine ältere Version installiert ist
        (Standard)
    e - wenn eine Datei fehlt oder eine gleiche bzw. ältere Version
        installiert ist
    d - wenn eine Datei fehlt oder eine andere Version installiert ist
    c - wenn eine Datei fehlt oder die Prüfsumme nicht mit dem berechneten
        Wert übereinstimmt
    a - Erzwingt die Neuinstallation aller Dateien
    u - Alle erforderlichen benutzerspezifischen Registrierungseinträge
        (Standard)
    m - Alle erforderlichen Computerspezifischen Registrierungseinträge
        (Standard)
    s - Alle vorhandenen Verknüpfungen (Standard)
    v - Führt von der Quelle aus und speichert das lokale Paket zwischen
        Öffentliche Eigenschaften festlegen
[PROPERTY=Eigenschaftswert]

```

Um msi-Pakete anzupassen, müssen Sie entweder ein mst (Microsoft Installer Transform) -Paket auf die msi-Datei anwenden, oder Sie editieren das msi-File direkt. Für Microsoft Office wird z.B. im Office Admin-Kit ein Assistent zur Verfügung gestellt, der das Erstellen der mst-Datei für Sie übernimmt. Zum Editieren eines msi-Files verwenden Sie am besten das Tool Superorca, das Sie kostenlos unter http://www.pantaray.com/msi_super_orca.html herunterladen können.

Steht Ihnen kein msi-File zur Verfügung, können Sie die Installation meist über die setup.exe unbeaufsichtigt ausführen. Eine über Jahre gewachsene Bibliothek mit vielen Tipps zu unterschiedlichen Programmen finden Sie unter <http://www.itninja.com/software/browse>. Die

Programme sind hier in alphabetischer Reihenfolge aufgelistet.

The screenshot shows the ITNinja Software Library page. At the top, there's a navigation bar with 'Software Tips', 'Questions', 'Blogs', 'Links', and 'Communities'. A search bar is located on the right. Below the navigation bar, there's a breadcrumb trail: 'Software'. The main heading is 'Software Library' with the subtitle 'Browse our extensive list of software.' There are two tabs: 'Software' (selected) and 'Companies'. Below the tabs is a navigation bar with letters A through Z and an 'ALL' button. A red arrow points to a search box labeled 'Keyword search' with a 'Go!' button. Below the search box are filters: 'Show only software:' with checkboxes for 'With Content', 'With Tips', and 'Trending', and a 'Platform' dropdown menu set to 'All'. An 'Update List' button is also present. The main content is a table with the following data:

| Versions | Software Name | Company | Content Count | Tip Count | Following |
|----------|--|-----------|---------------|-----------|-----------|
| 1 | 'PTC Places' Namespace Shell Extension | PTC | 0 | 0 | follow |
| 2 | .Net Components | Sage | 0 | 0 | follow |
| 3 | .Net Connector | SAP | 0 | 0 | follow |
| 1 | .NET Data Provider | Microsoft | 0 | 0 | follow |
| 6 | .NET Framework | Microsoft | 65 | 45 | follow |

Unter Keyword Search können Sie direkt das Programm eingeben, das Sie suchen. Geben Sie z.B. Irfanview ein, bekommen Sie eine ganze Reihe von Einträgen. Wählen Sie den ersten (derzeit: Irfan Skiljan Irfanview 4), landen Sie in der Anzeige „All Tips“. Der erste Eintrag beschreibt gleich alle Parameter, die an setup.exe übergeben werden können.

Nachdem Sie herausgefunden haben, wie Sie Ihr Programm automatisiert installieren können, erstellen Sie ein Skript-File. Prinzipiell können Sie alle verfügbaren Skriptsprachen verwenden, aber ab Windows 7 ist es empfehlenswert, Powershell zu nutzen.

Um die integrierten Logging-Möglichkeiten des MDT zu nutzen, sollten Sie ein neues TSEnvironment-Objekt erzeugen. Es gibt Ihnen Zugriff auf die Logging-Pfade des MDT. Außerdem ist es sinnvoll, die Dateiprüfung zu deaktivieren, die die Installation von aus dem Internet heruntergeladenen Programmen verhindert. Die machen Sie, in dem Sie die Umgebungsvariable %SEE_MASK_NOZONECHECKS% auf den Wert 1 setzen.

```
$tsenv = New-Object -COMObject Microsoft.SMS.TSEnvironment
$logPath = $tsenv.Value('LogPath')
$logFile = join-path -path $logPath -childpath ( $MyInvocation.MyCommand.Name +
'.log' )
$env:SEE_MASK_NOZONECHECKS=1
```

Das Logging kann jetzt z.B. über Start-Transcript gestartet werden. Dieser Powershell-Befehl schreibt die komplette Ausgabe der folgenden Befehle in eine Log-Datei.

```
Start-Transcript $logFile
"Logging to $logFile`r"
```

Die Installation kann z.B. mit dem Befehl Start-Process gestartet werden. Start-Process kennt zwei wichtige Parameter, -Filepath <Setup-Datei> und -Argumentlist <Argumente> für die Argumente, die beim Aufruf der Setup-Datei übergeben werden sollen. Um den laufenden Prozess überwachen zu können, geben Sie als weiteren Parameter -Wait an und speichern die Ausgabe des Start-Process in

einer Variablen. Dadurch wird das Powershell-Skript erst weiter ausgeführt, wenn das Setup beendet wurde. Im folgenden Beispiel wird eine msi-Datei mit Hilfe von msiexec installiert.

```
$msiArguments = "/i `"{0}\sourcen\7z1604-x64.msi`" /qn" -f
$myInvocation.MyCommand.path
write-Output "Starting command: msiexec.exe $msiArguments`r"
$Process = Start-Process -FilePath "msiexec.exe" -ArgumentList $arguments -
WorkingDirectory $env:temp -verb RunAs -Wait -PassThru
```

Über den Exitcode der Anwendung können Sie nun überprüfen, ob die Ausführung des Setups erfolgreich war.

```
if ( $Process.ExitCode -ne 0 -or $Process.ExitCode -ne 3010 )
{
    write-Output "Installation failed`r"
    $error[0]
}
else
{
    Write-Verbose -Message 'Installation successfull'
}
```

Anschließend müssen Sie die Sicherheitsüberprüfung von Dateien wieder aktivieren und das Logging beenden.

```
if ( $Process.ExitCode -ne 0 -or $Process.ExitCode -ne 3010 )
{
    write-Output "Installation failed`r"
    $error[0]
}
else
{
    Write-Verbose -Message 'Installation successfull'
}
```

Das vollständige Skript sieht dann so aus:

```
$tsenv = New-Object -COMObject Microsoft.SMS.TSEnvironment
$logPath = $tsenv.Value("LogPath")
$logFile = join-path -path $logPath -childpath ( $MyInvocation.MyCommand.Name +
".log" )
$env:SEE_MASK_NOZONECHECKS=1

Start-Transcript $logFile
"Logging to $logFile`r"

$msiArguments = "/i `"{0}\sourcen\7z1604-x64.msi`" /qn" -f
$myInvocation.MyCommand.path
write-Output "Starting command: msiexec.exe $msiArguments`r"
$Process = Start-Process -FilePath "msiexec.exe" -ArgumentList $arguments -
WorkingDirectory $env:temp -verb RunAs -Wait -PassThru
if ( $Process.ExitCode -ne 0 -or $Process.ExitCode -ne 3010 )
{
    write-Output "Installation failed`r"
    $error[0]
}
else {write-host "7-Zip erfolgreich installiert`r"}

$env:SEE_MASK_NOZONECHECKS=0
Stop-Transcript
```

Die Funktionalität ist durchaus noch verbesserbar. Ein sehr schönes Beispiel für ein Setup, das sowohl msi-Dateien als auch normale Setup-Dateien installieren kann und das über XML-Dateien anpassbar ist, finden Sie unter <https://syscramblings.wordpress.com/2015/03/25/powershell->

[application-wrapper-for-standalonemdt-and-sccm/](#). Wie Sie das Script mit Powershell starten können, finden Sie im Abschnitt [Anwendungen und Konfigurationsskripte](#).



Über den Autor

Holger Voges ist IT-Trainer und Consultant. Seine IT-Karriere begann mit einem Atari ST 512 Mitte der 80er Jahre. Seine ersten Erfahrungen mit großen Netzwerken hat er im Systembetrieb der Volkswagen Financial Services 1999 gewonnen. Ab dem Jahr 2000 war er dann als freiberuflicher IT-Trainer für verschiedene Schulungsunternehmen im Bereich Braunschweig und Hannover tätig, bis er 2002 mit 2 Mitstreitern sein erstes Schulungsunternehmen LayerDrei in Braunschweig gründete. Nach seinem Ausstieg bei LayerDrei war er dann mehrere Jahre als freiberuflicher Consultant vor allem im

SQL-Server Umfeld u.a. für T-Home Entertain, e.on und Hewlett-Packard unterwegs. 2012 gründete er dann das Schulungsunternehmen Netz-Weise IT-Training.

Im Dezember 2016 erschien sein Buch „Gruppenrichtlinien in Windows Server 2016, 2012 und 2008 R2“, das er in der 3. Auflage übernommen hat. Außerdem ist er regelmäßiger Sprecher z.B. auf der europäischen Powershell-Konferenz und auf verschiedenen anderen Veranstaltungen.

Netz-Weise IT-Training hat sich auf Firmenschulungen im professionellen IT-Umfeld spezialisiert und bietet Schulungen u.a. im Bereich Microsoft, VMware, Linux und Oracle an.