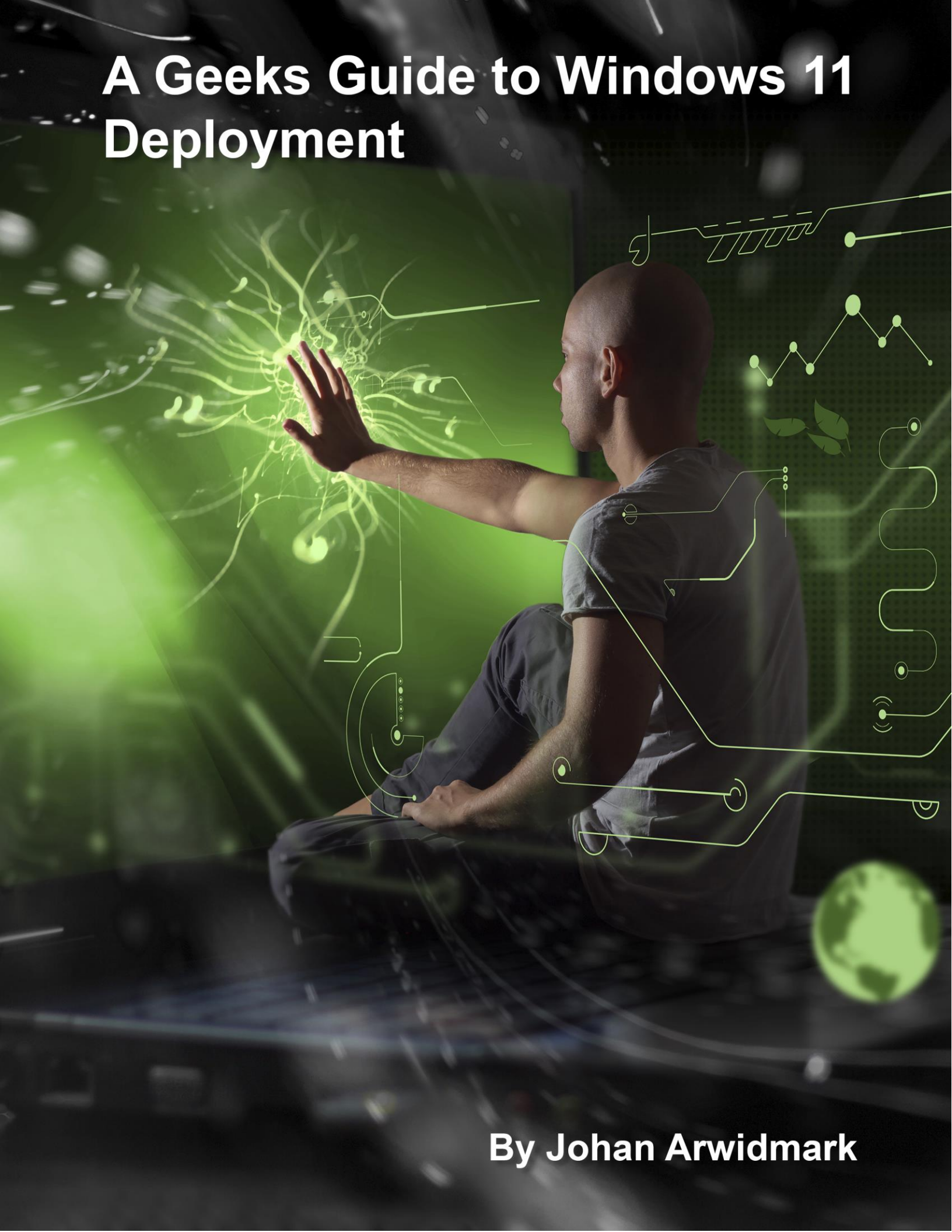
A Geeks Guide to Windows 11 Deployment



By Johan Arwidmark

PUBLISHED BY

Deployment Artist
https://deploymentartist.com

Copyright © 2022 by Deployment Artist

All rights reserved. No part of this guide may be reproduced or transmitted in any form or by any means without the publisher's prior written permission. Deployment Artist is a brand under the Mirolus corporation, https://mirolus.com.

Warning and Disclaimer

Every effort has been made to make this guide as complete and accurate as possible, but no warranty or fitness is implied. The information provided is on an "as is" basis. The author and the publisher shall have neither liability nor responsibility to any person or entity concerning any loss or damages arising from the information contained in this manual.

Feedback Information

We'd like to hear from you! If you have any comments about how we could improve the quality of this guide, please don't hesitate to contact us by visiting https://deploymentartist.com or emailing feedback@deploymentartist.com.

Module 1 ..... **1**

A Geeks Guide to Windows 11 Deployment **1**

Welcome to this guide 1

Lab Environment 2

Module 2 ..... **6**

Planning for Windows 11 Deployment **6**

Windows 11 Deployment 6

Module 3 ..... **9**

Using PowerShell to export Windows 11 Enterprise Edition **9**

What you need for this module 9

Step 1 – Exporting Windows 11 Enterprise Edition 10

Module 4 ..... **13**

Optional - Creating Reference Images **13**

What you need for this module 13

Why using MDT Lite Touch? 14

Step 1 – Creating the MDT server structure 14

Step 2 – Add Windows 11 installation files 18

Step 3 – Add Applications 20

Step 4 – Create the MDT Task Sequence 22

Step 5 – Configure the deployment share 24

Step 6 – Create Windows Reference Images 27

Module 5 ..... **30**

Setup MDT Lite Touch for OSD **30**

What you need for this module 30

Step 1 – Prepare for Production Deployment 31

Step 2 – Add Operating System Images 35

Step 3 – Add Drivers 37

Step 4 – Create the Production Task Sequences 41

Step 5 – Configure the Deployment Share 43
Step 6 – Install and Configure WDS 46
Step 7 – Deploy Windows using PXE 49
Module 6 51
Setup ConfigMgr for OSD 51
What you need for this module 51
Step 1 – Prepare for ConfigMgr OSD 52
Step 2 – Add operating system images 65
Step 3 – Add Drivers 67
Step 4 – Create Task Sequences 74
Step 5 – Deploy Windows 11 using PXE 84
Beyond the eBook 86
Live Presentations 86
Live Instructor-led Academy Courses 86
Live Instructor-led Masterclasses 86
Twitter and LinkedIn 86
Facebook 86

Module 1

A Geeks Guide to Windows 11 Deployment

Welcome to this guide

Thanks for downloading this guide to Windows 11 Deployment. While this guide does cover some generic Windows 11 deployment topics, the focus is intended for sysadmins using either Microsoft Endpoint Configuration Manager (ConfigMgr/SCCM/MEMCM) or Microsoft Deployment Toolkit (MDT) Lite Touch, wanting to deploy Windows 11.

For the ConfigMgr module of this guide, I used ConfigMgr Current Branch 2203, and for the MDT Lite Touch modules, I used MDT 8456.

Real World Note: The instructions in this guide are made for Windows 11 Enterprise, but you can follow the same instructions for deploying other editions of Windows 11, and also for deploying Windows 10.

Step-by-Step Modules

This guide contains a crash course in planning for Windows 11 deployments followed by a series of step-by-step modules to perform the following Windows 11 deployment tasks.

- Using PowerShell to export Windows 11 Enterprise Edition from the Windows 11 ISO file
- Optional - Setup MDT Lite Touch to build and capture a Windows 11 reference image
- Setup MDT Lite Touch for bare metal deployment of Windows 11
- Setup ConfigMgr OSD for bare metal deployment of Windows 11

Download the sample files

The sample files for this guide are available for download at the below link:

- https://github.com/DeploymentResearch/DRFiles/tree/master/GeeksGuideW11

© Copyright 2022 Deployment Artist

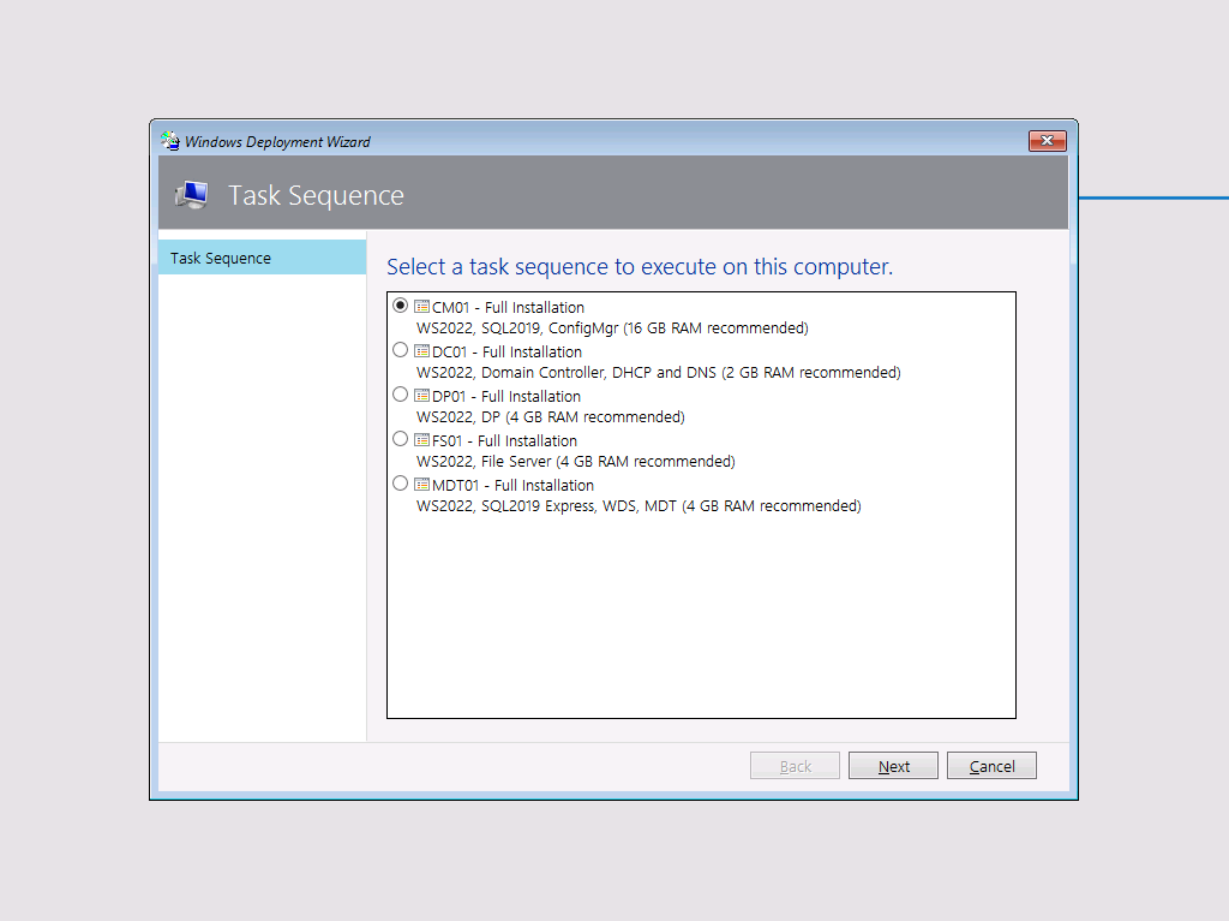
Lab Environment

In this guide I'm using a set of virtual machines, which are all part of the corp.viamonstra.com domain. If you want a quick path to setup a lab environment for this, I recommend using the Hydration Kit available here: https://www.deploymentresearch.com/hydration-kit-for-windows-server-2022-sql-server-2019-and-configmgr-current-branch/

In my lab I'm using Windows Server 2022 as a server operating system, but all step-by-step guides also work with Windows Server 2019, and even good old Windows Server 2016 (which is still supported, even while only on extended support until Jan 12, 2027).

In the various modules in this guide, I'm using the following list of virtual machines.

- **DC01.** A Windows Server 2022 configured as DNS, DHCP, and Domain Controller.
- **MDT01.** A Windows Server 2022 configured as a File Server
- **CM01.** A Windows Server 2022 or configured with SQL Server and ConfigMgr (SCCM)



Using the Hydration Kit to deploy a lab environment for ConfigMgr or MDT Lite Touch.

© Copyright 2022 Deployment Artist

© Copyright 2022 Deployment Artist

Virtual Machines - Details

To build a replica of the infrastructure used for this guide, you can run all virtual machines on a single Hyper-V or VMware host with 32 GB of RAM and at least 500 GB free disk space. I recommend using a 1TB NVMe SSD for your VMs to give you some extra headroom.

As mentioned in the introduction, you can obviously use your own custom lab environment when working through the guides in this book, but if you want all guides and scripts to match exactly, I do recommend that you use the automated lab setup (hydration).

Note: If you only want to work with ConfigMgr OSD, and not use MDT Lite Touch for any deployments or for creating reference images, you can safely skip deploying the MDT01 server.

- **DC01.** A **Windows Server 2022** VM, fully patched with the latest security updates, and configured as Active Directory Domain Controller, DNS Server, and DHCP Server in the **corp.viamonstra.com** domain.
    - Server name: **DC01**
    - IP Address: **192.168.1.200**
    - Roles: **DNS**, **DHCP**, and **Domain Controller**
    - Disk Volumes: **C: (100 GB dynamic disk)**
    - vCPU count: **2**
    - RAM: **2 GB**
- **MDT01.** A **Windows Server 2022** VM, fully patched with the latest security updates, and configured as a member server in the **corp.viamonstra.com** domain.
    - Server name: **MDT01**
    - IP Address: **192.168.1.210**
    - Roles: **File Server**
    - Software: **Windows ADK 11**, and **MDT 8456**
    - Disk Volumes: **C: (100 GB dynamic disk),** and **E: (300 GB dynamic disk),**
    - vCPU count: **2**
    - RAM: **4 GB**

- **CM01.** A **Windows Server 2022** VM, fully patched with the latest security updates, and configured as a member server in the **corp.viamonstra.com** domain.
    - Server name: **CM01**

© Copyright 2022 Deployment Artist

- IP Address: **192.168.1.214**
- Roles: **File Server** and **WDS**
- Software: **SQL Server 2017**, **Windows ADK 11**, and **ConfigMgr 2203**
- Disk Volumes: **C: (100 GB dynamic disk),** and **E: (300 GB dynamic disk),**
- vCPU count: **4**
- RAM: **16 GB**

Internet Access

In terms of providing Internet access to your VMs, I highly recommend putting all the VMs on an isolated (internal) virtual network/switch, and then use the NAT feature in either Hyper-V or VMware to have the machines accessing Internet. Learn more on the links below:

- How to enable Hyper-V NAT: https://bit.ly/Hyper-VNAT

For more advanced networking you can use virtual routers with two or more network adapters. For example, running either Windows Server, or pfSense.

- How to setup a pfSense virtual router: https://bit.ly/pfSenseVirtualRouter
- How to setup a Windows Server virtual router: https://bit.ly/WSVirtualRouter

© Copyright 2022 Deployment Artist

Module 2

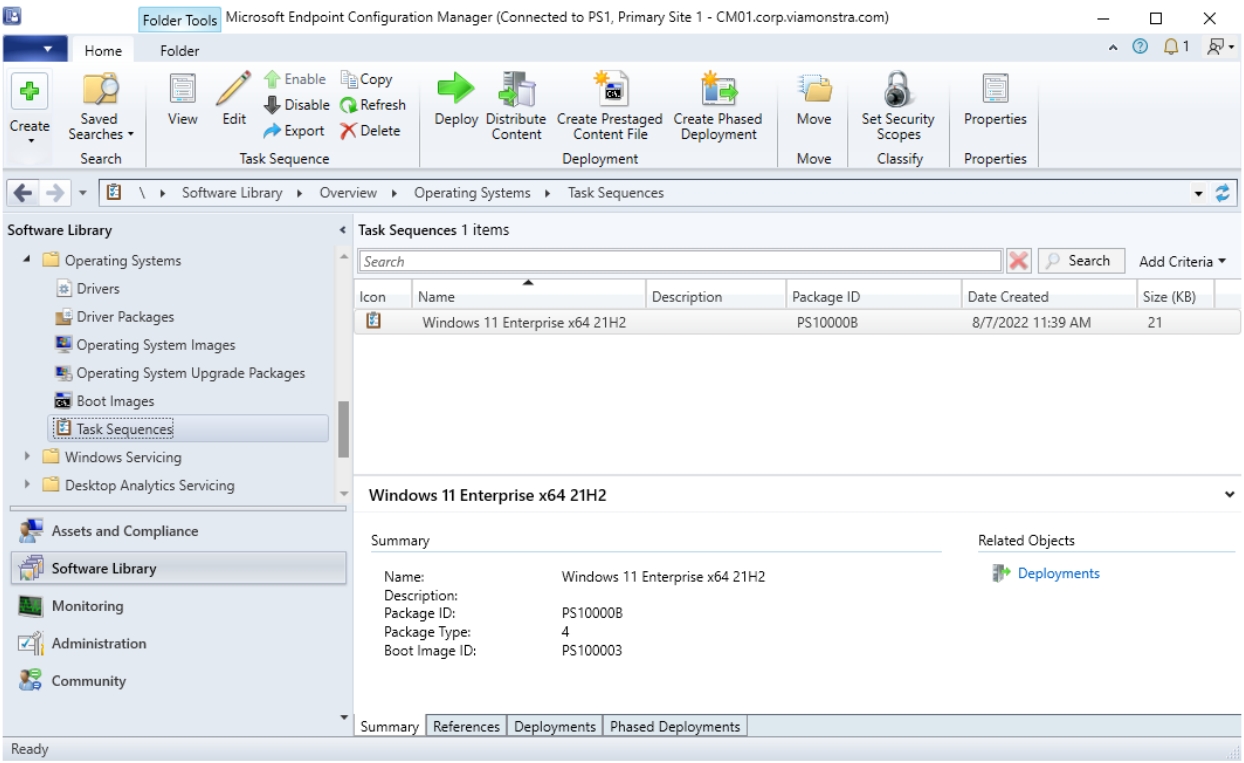
Planning for Windows 11 Deployment

Windows 11 Deployment

In general, if you know how to deploy Windows 10, you are in a great position for deploying Windows 11. From a core setup point of view there are not too many changes.

The two most popular platform for deploying Windows 11 are ConfigMgr and MDT Lite Touch, where MDT Lite Touch is primarily used for organizations not having ConfigMgr, but still need to do Windows 11 deployment.

Real World Note: MDT can also be integrated into ConfigMgr, providing extra optional OSD features. Due to the legacy MDT code base (VBScript), and the lack of updates from Microsoft in recent years, more and more organizations are moving away from the original MDT integration. Replacing commonly used MDT features with PowerShell scripts.



The ConfigMgr console showing an OSD task sequence.

© Copyright 2022 Deployment Artist

© Copyright 2022 Deployment Artist

Deployment Solutions for Windows 11

As you learned in the introduction, Microsoft offers two different solutions for OS Deployment. MDT Lite Touch, which is a standalone deployment solution, and ConfigMgr OSD which is part of the Software Distribution feature in ConfigMgr (also known as MEMCM or SCCM).

In the guide you learn to setup both solutions, but organizations typically pick on or the other even though exceptions apply. For example, it's not uncommon that organizations are using ConfigMgr OSD for Windows Client deployments, while using MDT Lite Touch for Windows Server deployments.

Assessment and Readiness Tools

Historically Microsoft have offered on-premises planning tools for Windows deployment, but they are long gone now, and have been replaced with cloud services like Endpoint Analytics and Update Compliance.

Windows ADK for Windows 11 contains the core deployment tools that MDT and ConfigMgr are using in the background. This kit is mostly based on command line tools but also provides interfaces for developers to create their own tools. In addition to Windows ADK, you also need the Windows ADK 11 WinPE Addon, which is used for all types of Windows 11 deployments except upgrades / servicing.

Note: As of this writing the new Windows ADK for Windows 11 22H2 is not supported by either ConfigMgr or MDT. Please use Windows ADK for Windows 11 21H2 instead.

Deployment Scenarios

ConfigMgr supports the following deployment scenarios for Windows 11, and you'll learn how to setup one of them in this guide (New Computer):

- **New Computer.** This scenario is also known as bare metal deployment, and is used when you have a blank machine you need to deploy, or an existing machine you want to wipe and redeploy without caring about any existing data. The setup usually starts from a boot media via USB or PXE, but also supports standalone media for deployments in location with poor network connectivity.
- **Computer Refresh.** Sometimes called wipe-and-load, and is primarily used to fix broken machines, but also to "upgrade/service" machines that cannot be serviced via the Inplace upgrade process. This guide does not cover this scenario.
- **Computer Replace.** A computer replace is quite much like the refresh scenario. But since you are replacing the machine, this scenario is divided into two main tasks: Backup of the old client, and bare metal deployment of the new client. As with the refresh scenario, user data and settings are backed up and restored. This guide does not cover this scenario.
- **Inplace Upgrade.** These days, Inplace upgrade scenarios are primarily used to upgrade Windows 11 from either Windows 10, or from other Windows 11 versions to the next.

© Copyright 2022 Deployment Artist

Module 3

Using PowerShell to export Windows 11 Enterprise Edition

In this module you learn to quickly export a Windows 11 Enterprise image from the Windows 11 ISO file that you can downloaded from either the Microsoft Volume License Site, from your Visual Studio subscription or from the Microsoft Evaluation center if you don't have access to the real media.

Once the Windows 11 Enterprise image has been exported, you now have a WIM image you can add to ConfigMgr for production deployment which is covered in *Module 5 – Setup ConfigMgr OSD*. This type of image, extracted directly from the ISO, is often referred to as having a thin image.

Real World Note: While Windows 11 images certainly can have additional applications in them, often called thick images, or master images, I see more and more organizations moving towards using a single thin image, and have the task sequence install needed application etc. during deployment.

What you need for this module

For the guides in this module, you need the following:

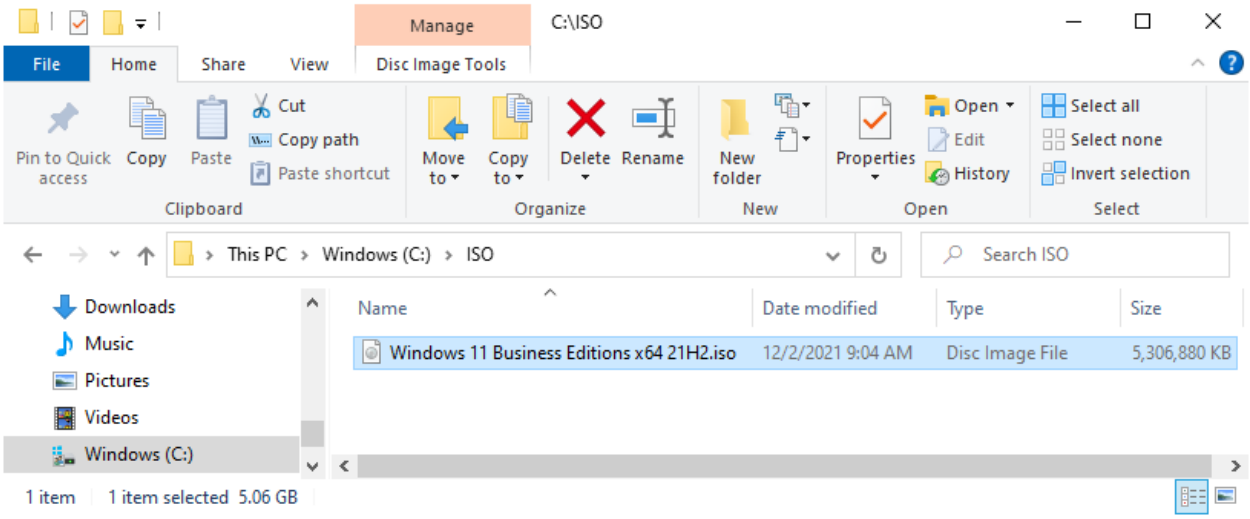
- A Windows 11 21H2 ISO.
- The sample files for this guide: https://github.com/DeploymentResearch/DRFiles/tree/master/GeeksGuideW11

© Copyright 2022 Deployment Artist

Step 1 – Exporting Windows 11 Enterprise Edition

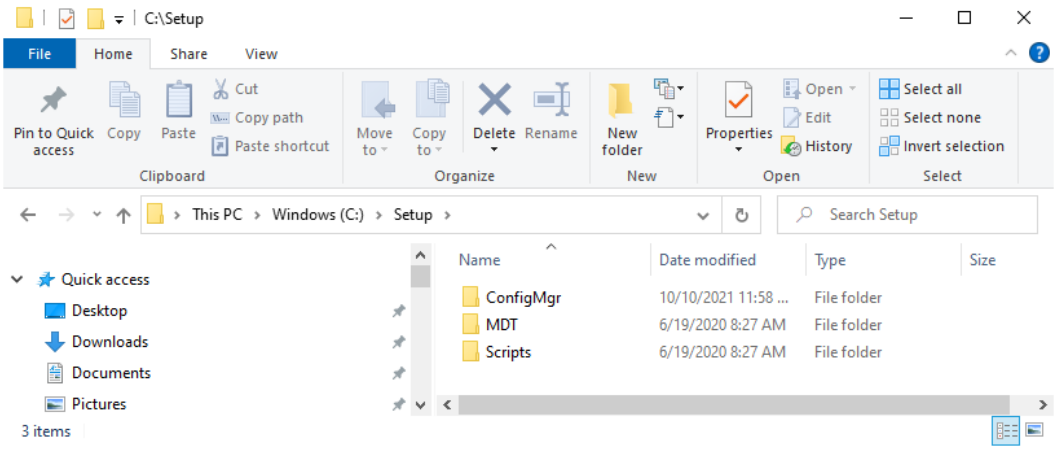
In this guide I assume that you have downloaded the sample files as well as a Windows 11 ISO.

1. On your Windows PC, create the following folders:
    - C:\ISO
    - C:\Setup
    - C:\WIM
2. Copy the Windows 11 ISO to the C:\ISO folder. In my example I named the file Windows 11 Business Editions x64 21H2.iso.



The Windows 11 Business Editions x64 21H2.iso file

3. Extract the content of the sample files to the C:\Setup folder.



© Copyright 2022 Deployment Artist

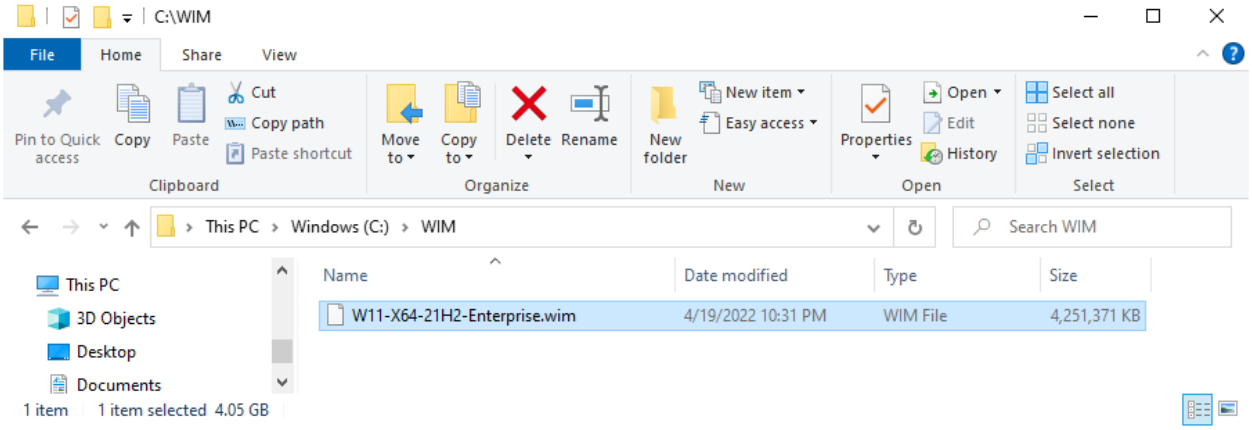
The sample files extracted to C:\Setup

© Copyright 2022 Deployment Artist

4. Start either PowerShell ISE or Visual Studio Code elevated, navigate to the C:\Setup\Scripts folder, and then open the Export-Windows11EnterpriseWIMFromISO.ps1 script

5. If needed, update line 5-8 in the script to reflect your environment.

6. Then run the Export-Windows11EnterpriseWIMFromISO.ps1 script, and review the result in the C:\WIM folder.



The Windows 11 Enterprise WIM image.

© Copyright 2022 Deployment Artist

Module 4

Optional - Creating Reference Images

In this optional module you learn to create a thick image using the MDT Lite Touch build and capture process. The resulting image will later be imported into ConfigMgr for the standard deployment scenarios only (bare metal, computer refresh, and computer replace). Most organizations don't need a thick image these days, but if you are working for one that still does, this module is for you. If you don't need a thick reference image in your environment, you can safely skip to Module 5 instead.

Please note that a thick image, meaning an image with applications that has been built and captured, cannot be used for Windows 11 upgrades or servicing. It must be an image that has been either serviced offline, or simply the native media that Microsoft provides.

Real World Note: To prevent issues with Sysprep failing due to application being updated in the background, make sure the VM you are building does not have Internet access, or implement the Setup State changes (INI file and Registry value) in this blog post for a fix: https://bit.ly/w1121h2-refimage

What you need for this module

For the guides in this module, you need the following:

- A Domain Controller running Windows Server 2016 or later. In my lab I used a domain controller named DC01 running Windows Server 2022 that manages the corp.viamonstra.com domain (VIAMONSTRA is the NetBIOS name).
- A Member Server running Windows Server 2016 or later. In my lab the Member Server is named MDT01, its running Windows Server 2022, and it's a member of the corp.viamonstra.com domain.
- A Windows 11 21H2 ISO file.
- Application(s) that you want to add to your image during build and capture. For simplicity I added Adobe Reader DC as an example.
- The sample files for this guide: https://github.com/DeploymentResearch/DRFiles/tree/master/GeeksGuideW11

© Copyright 2022 Deployment Artist

Why using MDT Lite Touch?

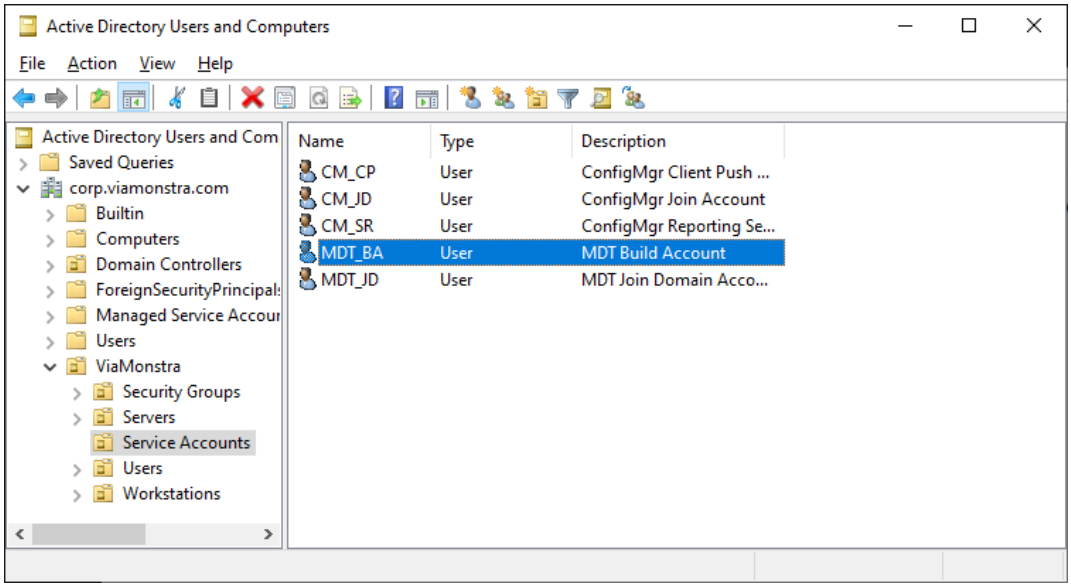
A great question to ask is the following: Since we are going to deploy the image with ConfigMgr, why don't we use ConfigMgr to also create the reference image? Well, it boils down to this:

- Using MDT Lite Touch is simply faster, and easier to automate. This means less hours spent (in total) creating and maintaining the image.
- You can use a Suspend action that allow for reboots, useful when needing to installing/configure something manually, or just want to check the reference image before it's automatically captured.

Step 1 – Creating the MDT server structure

Review the service account for MDT builds

1. On **DC01**, using **Active Directory Users and Computers**, make sure you have created a user account named **MDT_BA**. In my lab I added that account to the **ViaMonstra / Service Accounts** OU, and I assigned a password of **P@ssw0rd**.



The Service Accounts OU in my lab.

Install MDT 8456

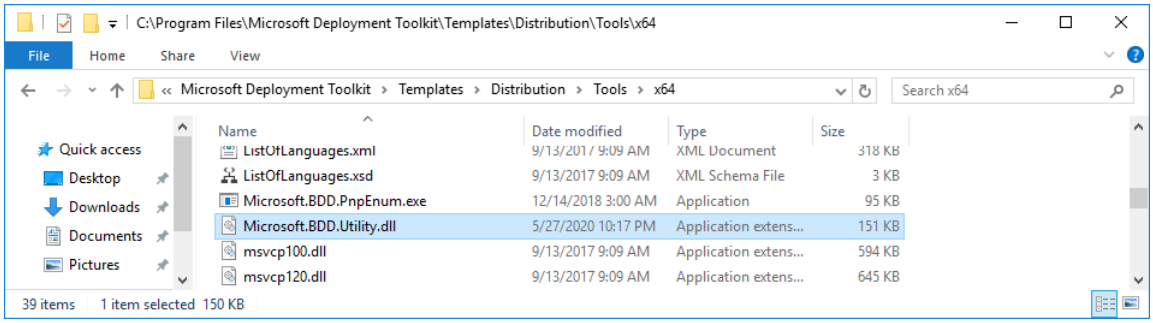
1. On **MDT01**, log on as an administrator.
2. Download **MDT 8456** from https://aka.ms/mdtdownload and install it using the default settings.

© Copyright 2022 Deployment Artist

Install the MDT 8456 Hotfix

Due to a code change in Windows ADK 11, and Windows 11, you need to download some updated files for MDT 8456. You can download these files here: https://bit.ly/MDT8456HF

1. On **MDT01**, download the **MDT 8456 hotfix** (MDT_KB4564442.exe), and extract it to a folder named **C:\Setup\MDT 8456 Update** (create the folder).
2. Copy the x86 version of the new **Microsoft.BDD.Utility.dll** from **C:\Setup\MDT 8456 Update\x86** to **C:\Program Files\Microsoft Deployment Toolkit\Templates\Distribution\Tools\x86**. Replace the existing file.
3. Copy the x64 version of the new **Microsoft.BDD.Utility.dll** from **C:\Setup\MDT 8456 Update\x** to C**:\Program Files\Microsoft Deployment Toolkit\Templates\Distribution\Tools\x64**. Replace the existing file.



The updated Microsoft.BDD.Utility.dll added to the MDT installation folder.

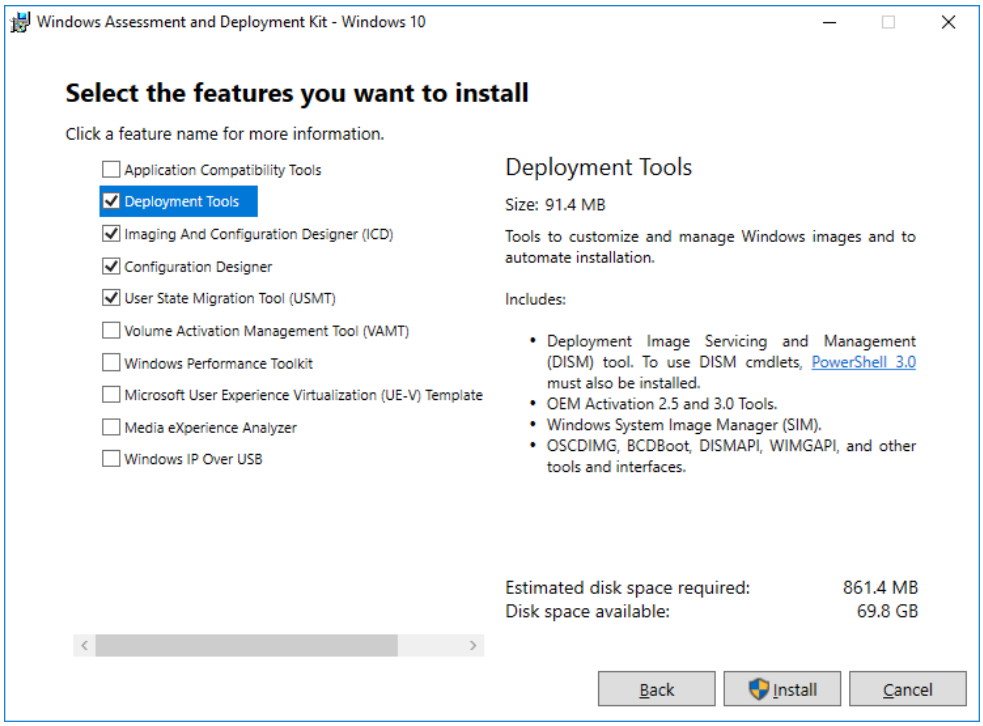
Install Windows ADK for Windows 11 21H2 and WinPE Addon

1. On **MDT01**, download and install **Windows ADK for Windows 11 21H2** from https://aka.ms/adk

Note: As of this writing the new Windows ADK for Windows 11 22H2 is not supported by either ConfigMgr or MDT. Please use Windows ADK for Windows 11 21H2 instead.

2. Select the following features during setup:
    - Deployment Tools
    - Imaging and Configuration Designer (ICD)
    - Configuration Designer
    - User State Migration Tool (USMT)

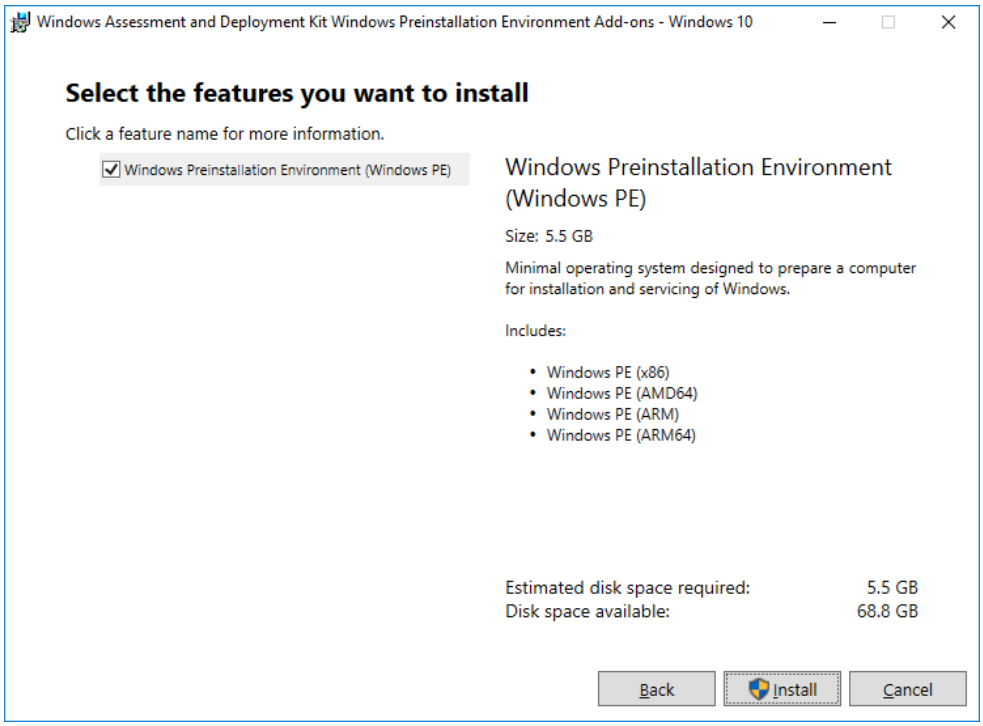
© Copyright 2022 Deployment Artist



The Windows ADK for Windows 11 21H2 setup.

1. Download and install **Windows ADK for Windows 11 21H2 WinPE Addon** from https://aka.ms/adk

2. Use the default settings during setup:



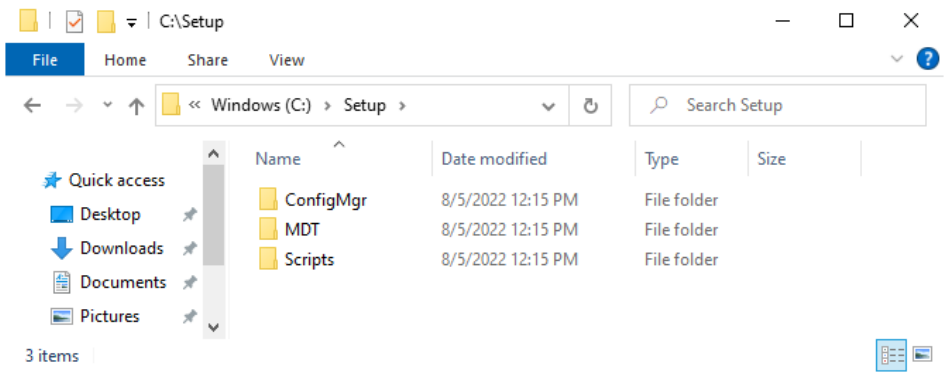
© Copyright 2022 Deployment Artist

Installing the WinPE Addon for Windows ADK for Windows 11 21H2.

Copy the Sample Files to MDT01

In this section I assume you have downloaded the sample files for this guide.

1. On **MDT01**, create the **C:\Setup** folder.
2. Download, and extract the sample files to **C:\Setup**
3. Review the content in the **MDT**, and **Script** folders (the ConfigMgr folder is for module 6).



The C:\Setup folder after extracting the sample files.

Create the MDT Build Lab Deployment Share

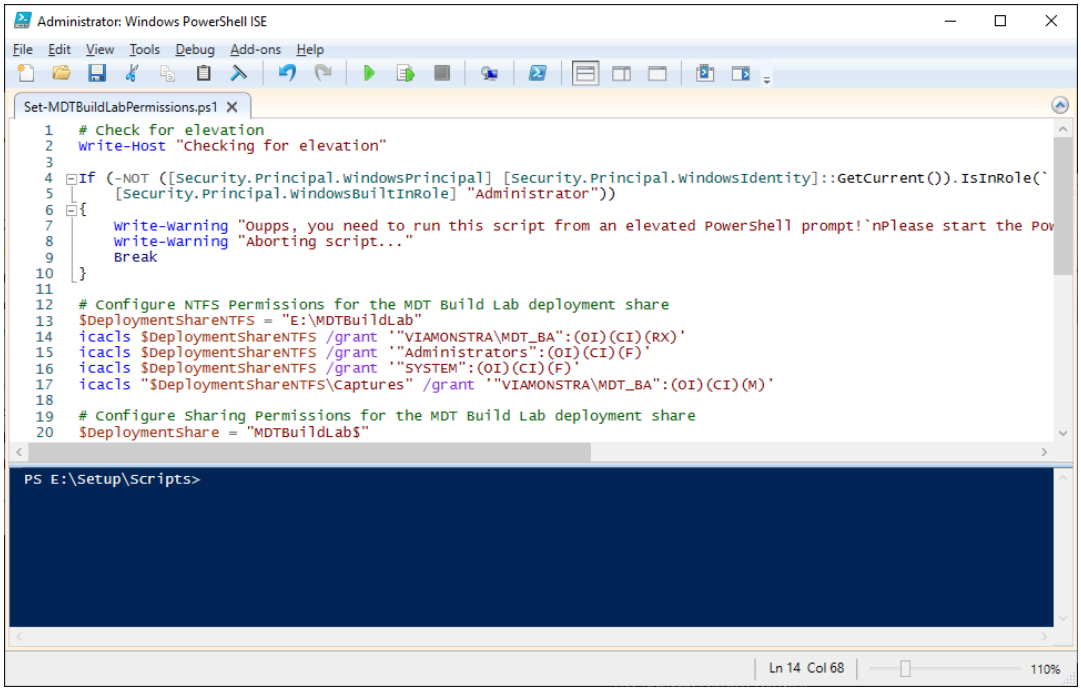
1. On **MDT01**, using the **Deployment Workbench** (available on the start menu), right-click **Deployment Shares** and select **New Deployment Share**. Use the following settings for the **New Deployment Share Wizard**.
    a. Deployment share path: **E:\MDTBuildLab**
    b. Share name: **MDTBuildLab$**
    c. Deployment share description: **MDT Build Lab**
    d. Options: <**default settings>**
2. Using **File Explorer**, open the **E:\MDTBuildLab** folder, and accept any permissions questions.

Configure Permissions for the MDT Build Lab deployment share

MDT by default locks down the deployment share so that only administrators can access it, which is a bit too harsh.

1. On **MDT01**, using PowerShell ISE or Visual Studio Code, review the **C:\Setup\Scripts\Set-MDTBuildLabPermissions.ps1** script. Modify line 17 if needed to match your domain name and MDT Build Account.

© Copyright 2022 Deployment Artist



The C:\Setup\Scripts\Set-MDTBuildLabPermissions.ps1 script.

2. Start an elevated **PowerShell** prompt (run as administrator).
3. Allow the **MDT_BA** account **Modify** permissions (NTFS Permissions) to the **E:\MDTBuildLab\Captures** folder by running the following script:

   ```
   C:\Setup\Scripts\Set-MDTBuildLabPermissions.ps1
   ```

4. Verify that you can access **E:\MDTBuildLab** folder, and the **\\MDT01\MDTBuildLab$** share.

Step 2 – Add Windows 11 installation files

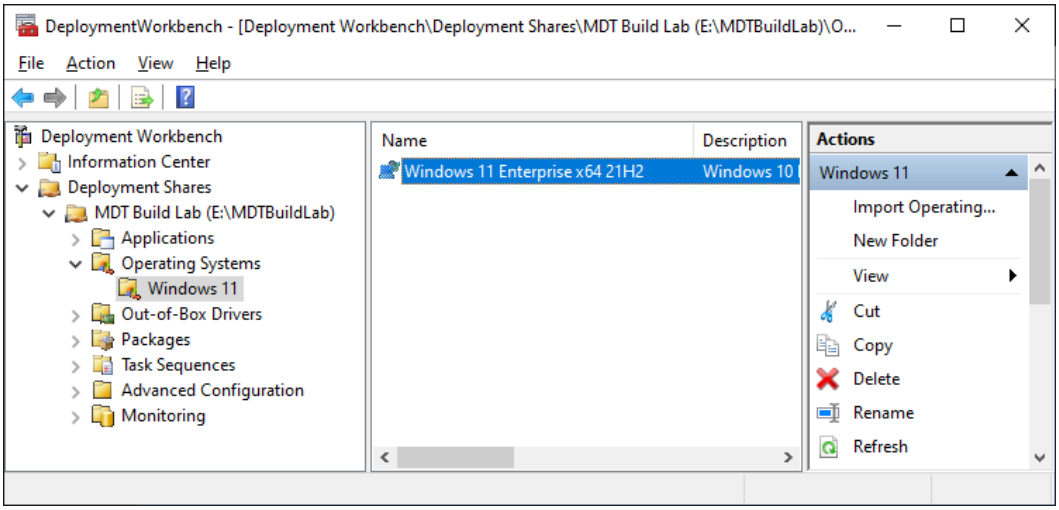
When adding operating system images, always try to use a media that is already updated by Microsoft, or by you using the OSD Builder community tool. This way you don't need to install software updates during build and capture, which will save some time.

Import the Windows 11 operating system

1. On **MDT01**, download (or copy) the **Windows 11 Business Editions x64 21H2** media to **E:\ISO**. In my lab I renamed the download file to **Windows 11 Business Editions x64 21H2.iso**.
2. Mount the **E:\ISO\Windows 11 Business Editions x64 21H2.iso** media.
3. Using the **Deployment Workbench**, expand the **Deployment Shares** node, expand **MDT Build Lab**, select the **Operating Systems** node and create a folder named **Windows 11**.

© Copyright 2022 Deployment Artist

4. Right-click the **Windows 11** node, and select **Import Operating System**. Use the following settings for the **Import Operating System Wizard**.
    a. **Full set of source files**
    b. Source directory: **D:** (or whatever drive letter the ISO mounted as)
    c. Destination directory name: **W11X64-21H2**
5. After adding the operating system, using the **Deployment Workbench**, in the **Windows 11** node, delete all images except for **Windows 11 Enterprise in W11X64-21H2 install.wim**.
6. Rename the remaining operating system to **Windows 11 Enterprise x64 21H2**.



The Windows 11 node after renaming the label for the imported operating system.

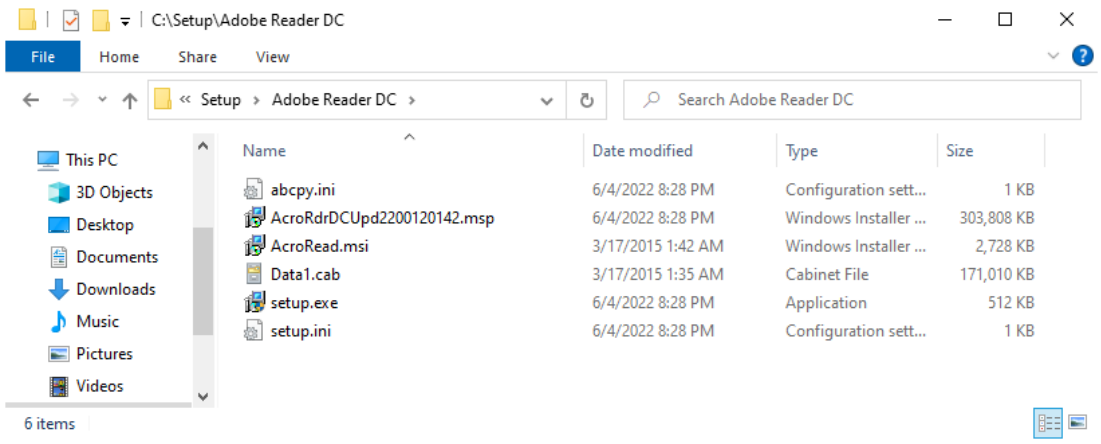
© Copyright 2022 Deployment Artist

Step 3 – Add Applications

In this step you are using Adobe Reader DC as example of adding an application to MDT. You can download this from: https://get.adobe.com/reader/enterprise/

Add Adobe Reader DC

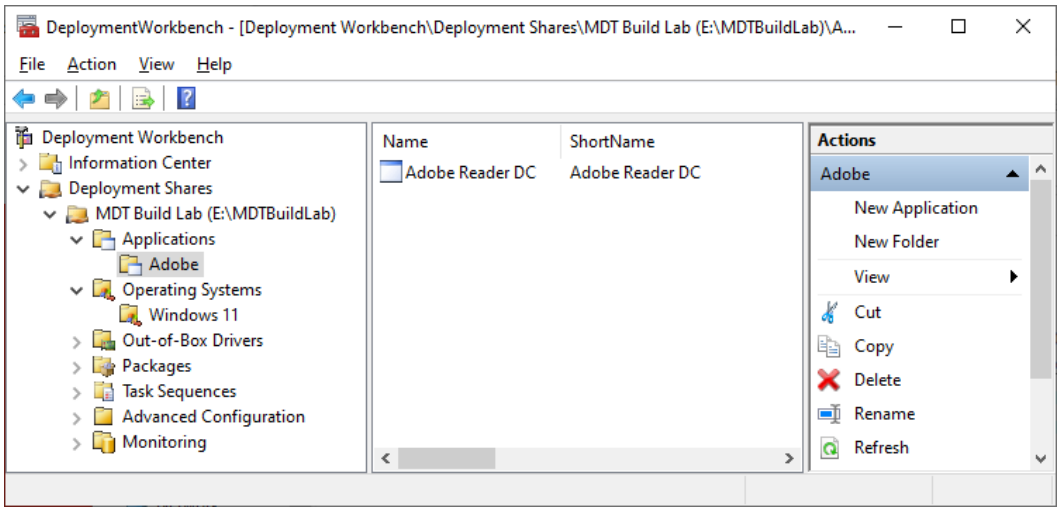
1. On **MDT01**, download and install **7-Zip** from https://www.7-zip.org
2. Download **Adobe Reader DC** from https://get.adobe.com/reader/enterprise/ and extract it using **7-Zip** to the **C:\Setup\Adobe Reader DC** folder (create the folder).



Adobe Reader DC extracted using 7-Zip to the C:\Setup\Adobe Reader DC folder.

3. Using the **Deployment Workbench**, expand the **Deployment Shares** node, expand **MDT Build Lab**, select the **Applications** node and create a folder named **Adobe**.
4. Expand the **Applications** node, right-click the **Adobe** folder, and select **New Application**, Use the following settings for the **New Application Wizard**.
    a. Application with source files
    b. Publisher: **<blank>**
    c. Application name: **Adobe Reader DC**
    d. Version: **<blank>**
    e. Source Directory: **C:\Setup\Adobe Reader DC**
    f. Specify the name of the directory that should be created: **Adobe Reader DC**
    g. Command Line: **Setup.exe /sALL /rs /msi EULA_ACCEPT=YES**
    h. Working directory: **.\Applications\Adobe Reader DC**

© Copyright 2022 Deployment Artist



The Adobe Reader DC application added.

Step 4 – Create the MDT Task Sequence

Create and configure a Task Sequence

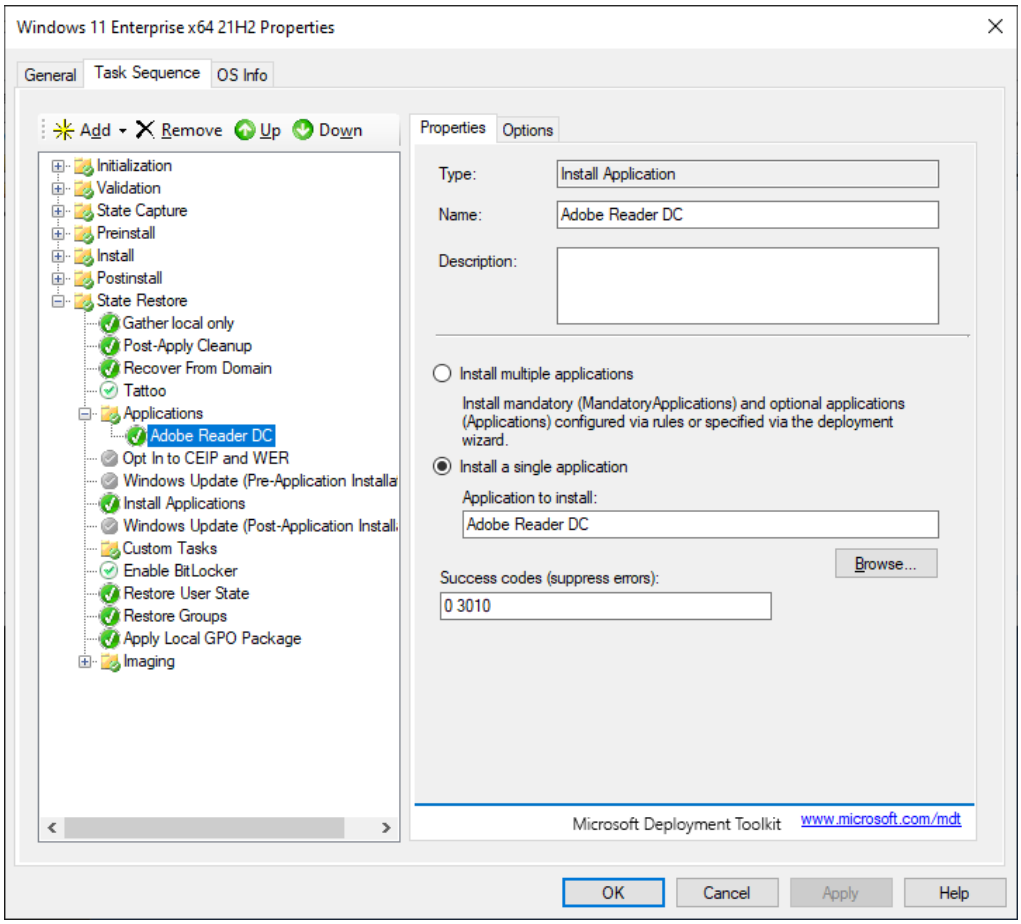
1. On **MDT01**, using the **Deployment Workbench**, in the **MDT Build Lab** deployment share, select the **Task Sequences** node, and create a folder named **Windows 11**.
2. Expand the **Task Sequences** node, right-click on the **Windows 11** node, and select **New Task Sequence**. Use the following settings for the **New Task Sequence Wizard**.
    a. Task sequence ID: **REFW11-X64-001**
    b. Task sequence name: **Windows 11 Enterprise x64 21H2**
    c. Template: **Standard Client Task Sequence**
    d. Select OS: **Windows 11 Enterprise x64 21H2**
    e. Specify Product Key: **Do not specify a product key at this time**
    f. Full Name: **ViaMonstra**
    g. Organization: **ViaMonstra**
    h. Internet Explorer home page: **about:blank**
    i. **Do not specify an Administrator password at this time**

Edit the Task Sequence

1. In the **Task Sequences / Windows 11** folder, right-click the **Windows 11 Enterprise x64 21H2** task sequence, and select **Properties**.
2. On the **Task Sequence** tab, configure the **Windows 11 Enterprise x64 21H2** task sequence with the following settings:
    a. State Restore. After the **Tattoo** action, add a **new Group** action with the following setting:
        Name: **Applications**
    b. State Restore / Applications. Add a new **Install Application** action with the following settings:
        - Name: **Adobe Reader DC**
        - Install a Single Application: **Adobe Reader DC**
3. Click **OK**.

© Copyright 2022 Deployment Artist

Real World Note: If your imported OS media is not already updated, you may also want to enable the two Windows Update actions in the MDT task sequence. They are disabled by default.



The task sequence with an application added, and the Windows Update actions enabled.

© Copyright 2022 Deployment Artist

Step 5 – Configure the deployment share

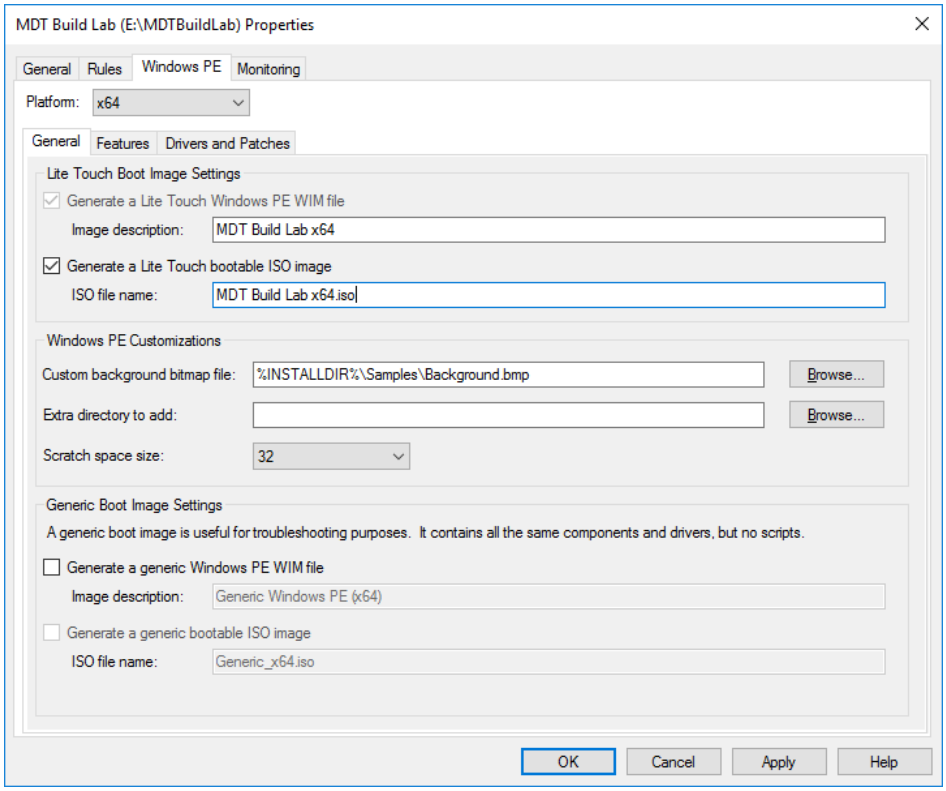
Prepare the deployment share rules

1. On **MDT01**, navigate to the **C:\Setup\MDT\MDTBuildLab** folder.
2. Copy the following files to **E:\MDTBuildLab\Control** (replace the existing files).
    - **Bootstrap.ini**
    - **CustomSettings.ini**
3. Review the **E:\MDTBuildLab\Control\Bootstrap.ini** file, note the **DeployRoot** value.
4. Review the **E:\MDTBuildLab\Control\CustomSettings.ini** file.

Configure the Deployment Share

1. Using the **Deployment Workbench**, right-click the **MDT Build Lab** deployment share and select **Properties**.
2. In the **General** tab, in the **Platforms Supported** area, clear the **x86** checkbox.
3. In the **Windows PE** tab, in the **Platform** dropdown list, make sure **x64** is selected.
4. In the **Lite Touch Boot Image Settings** area, configure the following settings
    c. Image description: **MDT Build Lab x64**
    d. ISO file name: **MDT Build Lab x64.iso**
5. Click **OK**.

© Copyright 2022 Deployment Artist



Properties of the MDT Build Lab deployment share, Windows PE x64 settings.

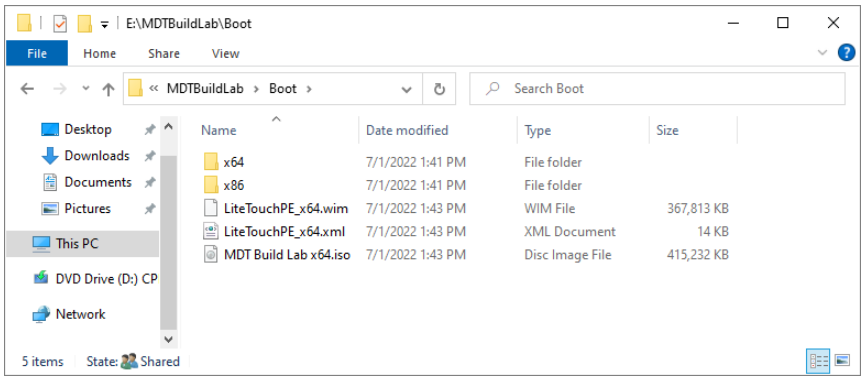
Update the Deployment Share

1. On **MDT01**, right-click the **MDT Build Lab** Deployment Share and select **Update Deployment Share**.

2. Use the default Options for the **Update Deployment Share** wizard.

Note: The update process will take a few minutes.

3. Review the contents of the **E:\MDTBuildLab\Boot** folder.



© Copyright 2022 Deployment Artist

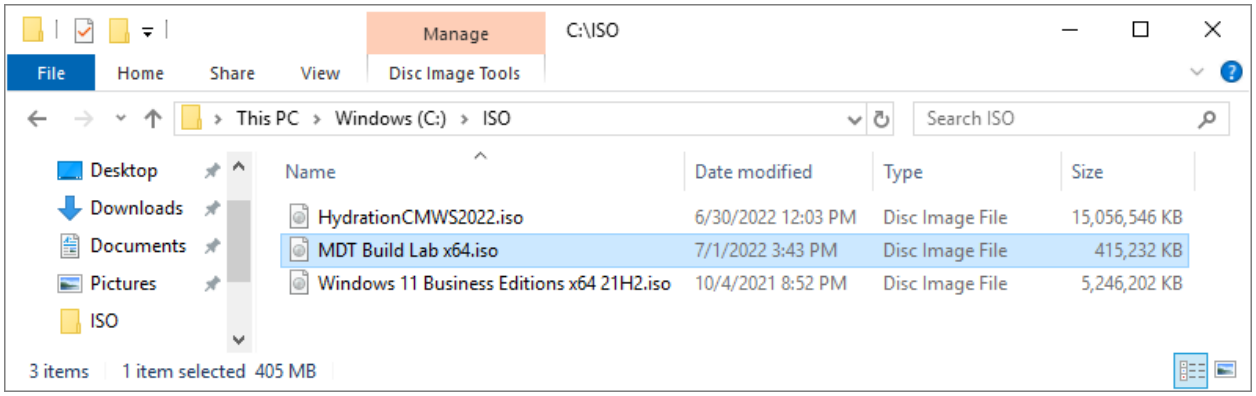
The contents of the E:\MDTBuildLab\Boot folder after updating the deployment share.

© Copyright 2022 Deployment Artist

Step 6 – Create Windows Reference Images

Create a Windows 11 Reference WIM Image, fully automated

1. On **MDT01**, copy the **E:\MDTBuildLab\Boot\MDT Build Lab x64.iso** file to **C:\ISO** on your Hyper-V or VMware PC.



The MDT Build Lab x64.iso file copied to C:\ISO.

2. On your Hyper-V or VMware PC, create a virtual machine named **REF001**. Assign at least 2 vCPU's and 4 GB RAM to the virtual machine, and make sure to disable any dynamic memory. Then mount the **MDT Build Lab x64.iso** media (located in the C:\ISO folder on the Hyper-V or VMware PC).

3. Start the **REF001** virtual machine, allow it to boot, and complete the **Deployment Wizard** using the below settings:

    a. Select a task sequence to execute on this computer:

        Windows 11 Enterprise x64 21H2

    b. Specify whether to capture an image:

        i. **Capture an image of this reference computer**

        ii. Location: **<default>**

        iii. File name: **<default>**

4. The deployment process will do the following:

    a. Install the Windows 11 Enterprise operating system.

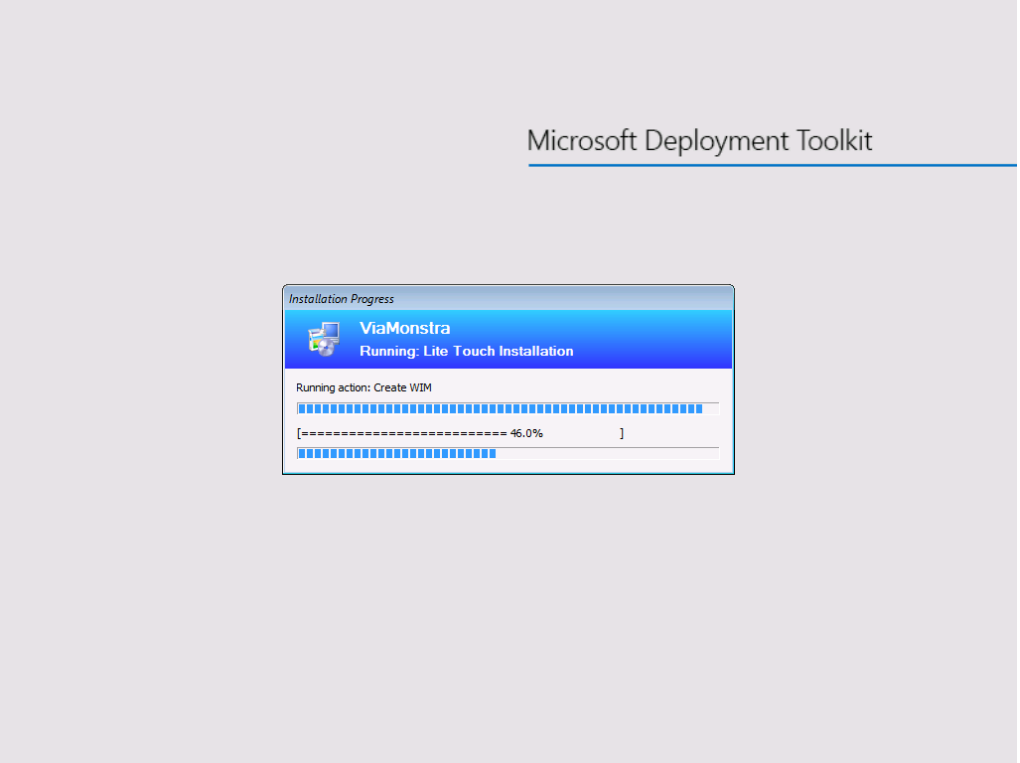
    b. Install the added Adobe Reader DC applications

    c. Run Sysprep and reboot into WinPE.

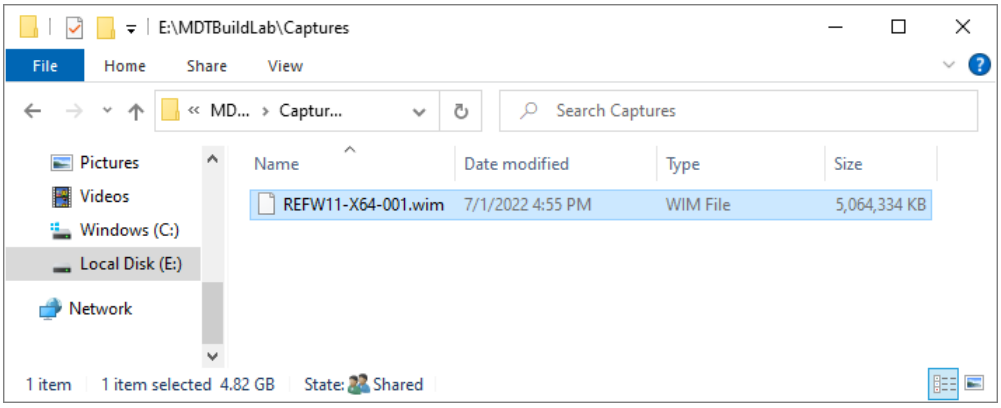
© Copyright 2022 Deployment Artist

d. In WinPE, capture the installation to a WIM file (saved in the E:\MDTBuildLab\Captures folder on MDT01).

Real World Note: Again, if you are having issues with Sysprep failing due to application being updated in the background, make sure the VM you are building either does not have Internet access, or implement the Setup State changes (INI file and Registry value) in this blog post for a fix: https://bit.ly/w1121h2-refimage



Windows 11 WIM image capture in progress.



© Copyright 2022 Deployment Artist

The Windows 11 WIM image on the Captures folder.

Module 5

Setup MDT Lite Touch for OSD

What you need for this module

For the guides in this module, you need the following:

- A Domain Controller running Windows Server 2016 or later. In my lab I used a domain controller named DC01 running Windows Server 2022 that manages the corp.viamonstra.com domain (VIAMONSTRA is the NetBIOS name).
- A Member Server running Windows Server 2016 or later with Windows ADK for Windows 11 21H2 and MDT 8456 installed. In my lab I was using Windows Server 2022 named MDT01.

Note: For detailed steps on installing Windows ADK for Windows 11 21H2 and MDT, check out the Step 1 – Creating the MDT server structure section in Module 4. Obviously, if you already perform the steps in module 4 you are all good from a setup point of view.

- A Windows 11 Enterprise 21H2 WIM image created via the steps in ether Module 3 or Module 4.
- The sample files for this guide: https://github.com/DeploymentResearch/DRFiles/tree/master/GeeksGuideW11

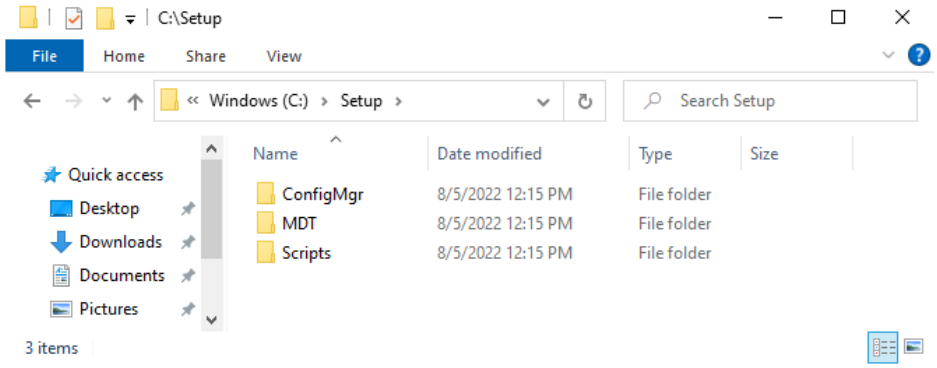
© Copyright 2022 Deployment Artist

Step 1 – Prepare for Production Deployment

Copy the Sample Files to MDT01

In this section I assume you have downloaded the sample files for this guide.

1. On **MDT01**, create the **C:\Setup** folder.
2. Extract the sample files to **C:\Setup**

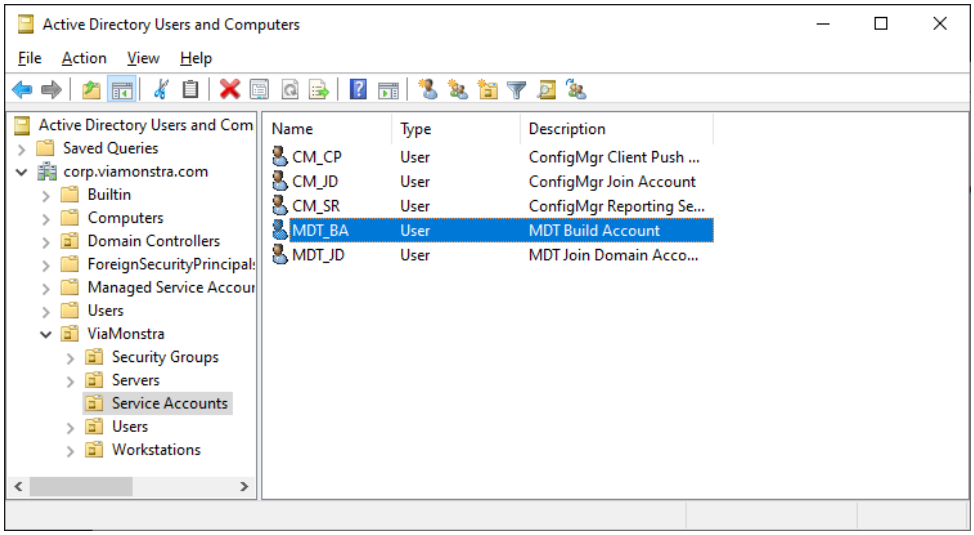


The C:\Setup folder.

Review the Service Account for MDT Lite Touch

1. On **DC01**, using **Active Directory Users and Computers**, make sure you have created the following user accounts:
    a. **MDT_BA**
    b. **MDT_JD**
2. In my lab I added the service account to the **ViaMonstra / Service Accounts** OU, and I assigned a password of **P@ssw0rd**.

© Copyright 2022 Deployment Artist



The Service Accounts OU in my lab.

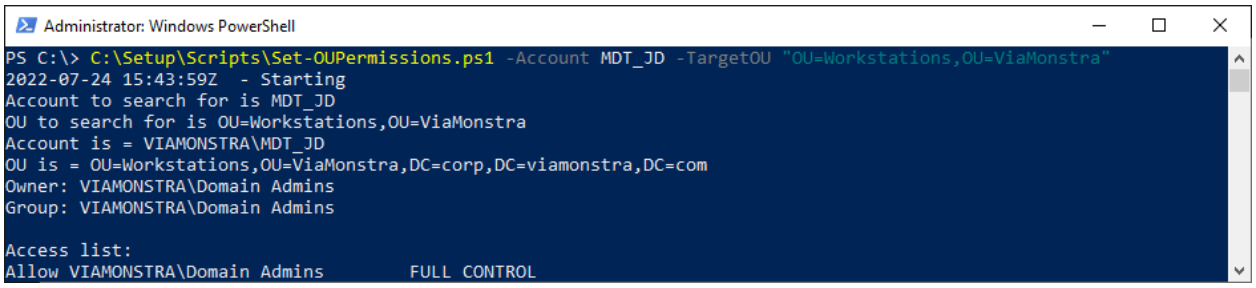
Configure Active Directory Permissions

For the ConfigMgr Task Sequence to join machines into your domain, the domain join account needs to have permissions to both create and update computer objects (depending on the computer object already existing or not). I recommend using a script to set the minimal permissions needed which is what you will do in this guide. In this section I assume you have downloaded the sample files for this guide.

1. On **DC01**, create the **C:\Setup** folder.
2. Extract the sample files to **C:\Setup**.
3. In an elevated **PowerShell prompt** (run as Administrator), run the following commands, press **Enter** after each command (please note that the last command is wrapped):

```
Set-ExecutionPolicy -ExecutionPolicy Unrestricted -Force


C:\Setup\Scripts\Set-OUPermissions.ps1
-Account MDT_JD -TargetOU "OU=Workstations,OU=ViaMonstra"
```



Running the Set-OUPermissions.ps1 script.

© Copyright 2022 Deployment Artist

4. The preceding script allows the MDT_JD user account minimal permissions to manage computer accounts in the ViaMonstra / Workstations OU.

Create and share the Logs folder

1. On **MDT01**, start an elevated **PowerShell** prompt (run as administrator).
2. Review the **C:\Setup\Scripts\Create-MDT01LogsFolder.ps1** script.
3. Create and share the **E:\Logs** folder by running the following command from an elevated PowerShell prompt:

   ```
   C:\Setup\Scripts\Create-MDT01LogsFolder.ps1
   ```

4. Verify that the **Logs$** share was created by running the following command:

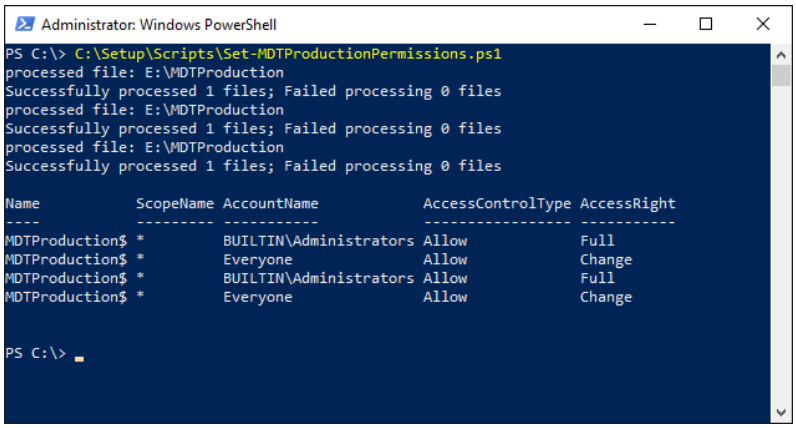
   ```
   Get-SmbShare
   ```

© Copyright 2022 Deployment Artist

Create the MDT Production Deployment Share

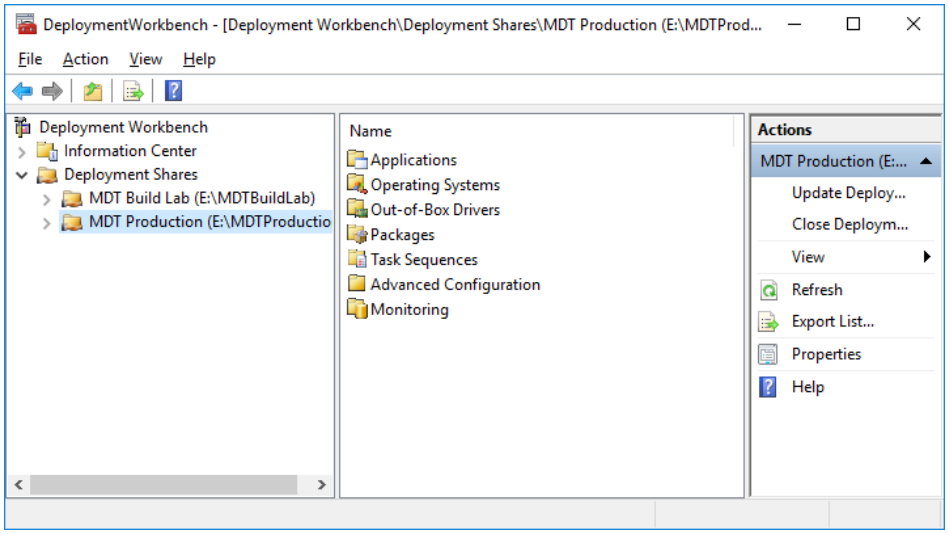
1. On **MDT01**, using the **Deployment Workbench**, right-click **Deployment Shares** and select **New Deployment Share**.
2. Use the following settings for the **New Deployment Share Wizard**.
    a. Deployment share path: **E:\MDTProduction**
    b. Share name: **MDTProduction$**
    c. Deployment share description: **MDT Production**
    d. Options: **<default settings>**
3. Using **File Explorer**, open the **E:\MDTProduction** folder, and accept any permissions questions.
4. Allow the **MDT_BA** account **Read** permissions (NTFS Permissions) to the **E:\MDTProduction** folder by running the following command in an elevated PowerShell prompt (modify the script if needed to match your domain and service account):

    ```
    C:\Setup\Scripts\Set-MDTProductionPermissions.ps1
    ```



Modifying the E:\MDTproduction permissions using PowerShell.

© Copyright 2022 Deployment Artist

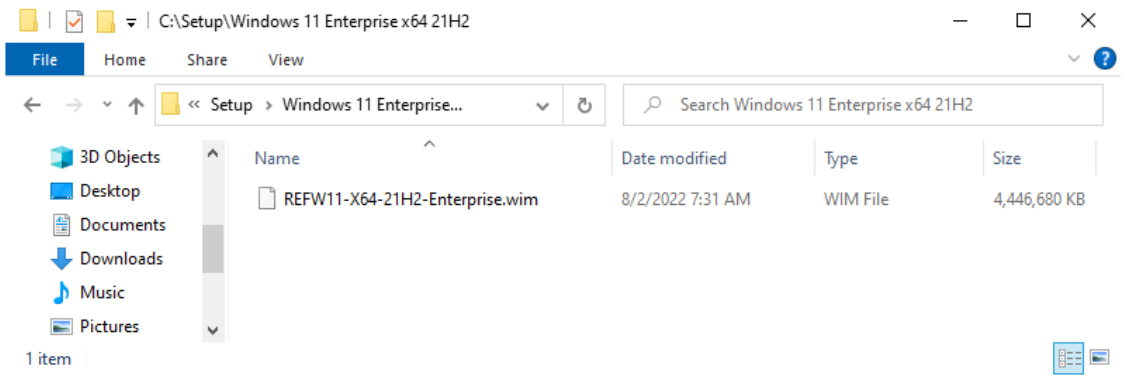


The new MDT Production Deployment Share.

Step 2 – Add Operating System Images

1. On **MDT01**, create the **C:\Setup\Windows 11 Enterprise x64 21H2** folder.
2. Copy your Windows 11 reference WIM image to the **C:\Setup\Windows 11 Enterprise x64 21H2** folder.

Note: In this example I copied the REFW11-X64-21H2-Enterprise.wim image from the steps in Module 3, but you can also copy the WIM image from MDT01 if you did a build and capture in Module 4.



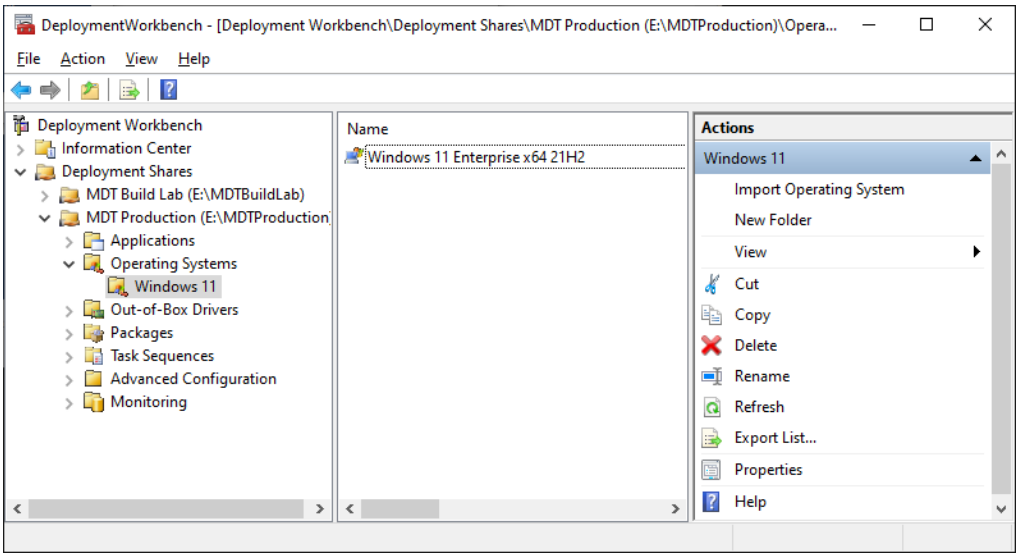
The Windows 11 image copied.

Import the Windows 11 image to MDT

1. On **MDT01**, using the **Deployment Workbench**, expand the **MDT Production** node, select the **Operating Systems** node, and create a folder named **Windows 11**.

© Copyright 2022 Deployment Artist

2. Expand the **Operating Systems** node, right-click on the **Windows 11** folder, and select **Import Operating System**. Use the following settings for the **Import Operating System Wizard**.
    a. **Custom image file**
    b. Source file: **C:\Setup\Windows 11 Enterprise x64 21H2\REFW11-X64-21H2-Enterprise.wim**
    c. **Setup files are not needed**
    d. Destination directory name: **W11-X64-21H2**
3. After adding the operating system, in the **Windows 11** node, change the Operating System name to **Windows 11 Enterprise x64 21H2**.



The MDT Production deployment share after adding the Windows 11 image.

© Copyright 2022 Deployment Artist

Step 3 – Add Drivers

To add support for new hardware you may need to add drivers to both WinPE, and to full Windows. In this example you add some WinPE and Windows drivers from HP.

The drivers referenced in this guide can be downloaded from these URLs:

WinPE x64 drivers
https://ftp.ext.hp.com/pub/caps-softpaq/cmit/softpaq/WinPE10.html

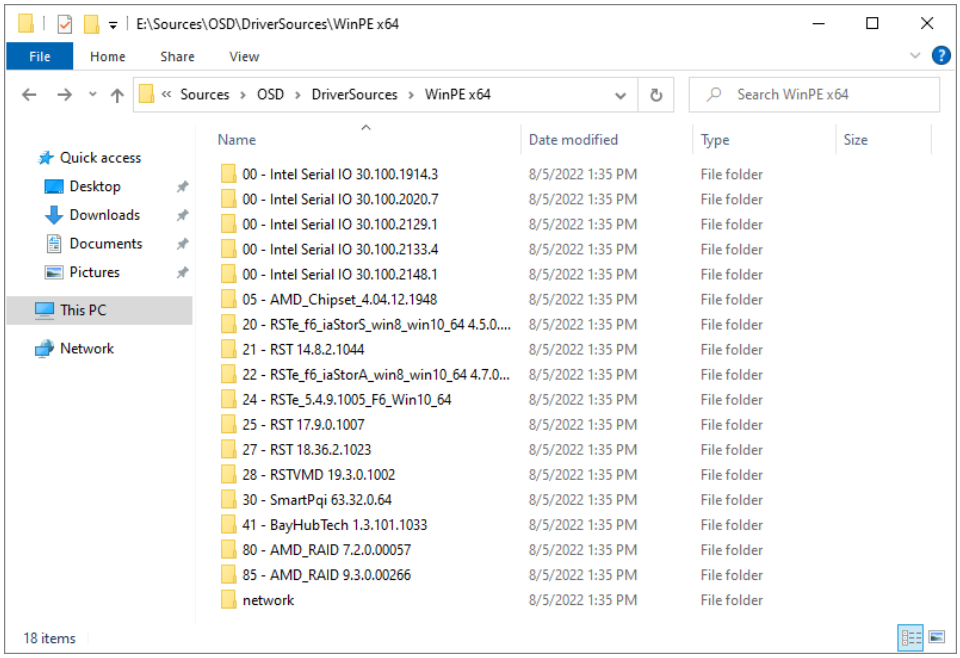
HP ZBook Firefly 14 G7 Mobile Workstation drivers for Windows 11 21H2
https://ftp.hp.com/pub/softpaq/sp140501-141000/sp140701.exe

Add Drivers for WinPE

1. On **MDT01**, download the **HP Client Windows PE 10 x64 Driver Pack**, and extract it to the **E:\Drivers\WinPE x64** folder (create the folder).

Real World Note: Even if you are using Windows ADK for Windows 11 21H2, you can still use the Windows PE 10 driver pack from HP. It's targeted for both WinPE 10 and WinPE 11.

Also, if you are wondering about the strange folder structure in the HP driver package for WinPE: Its simply there to guide you to the installation order, a requirement for some of the Intel RST drivers to work correctly. The challenge is that MDT, unlike ConfigMgr, does not inject the drivers from the source folder in alphabetic order. MDT create its own structure for drivers.



The HP WinPE drivers extracted to E:\Drivers\WinPE x64.

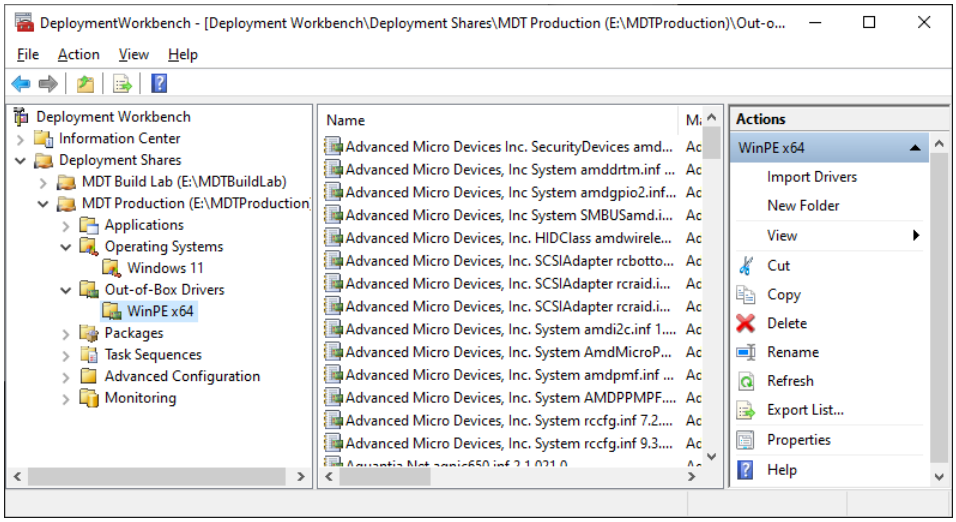
© Copyright 2022 Deployment Artist

2. From the **E:\Drivers\WinPE x64** folder, delete the **28 - RSTVMD 19.3.0.1002** subfolder. If you need support for this platform, you need to create an exit script to make sure the driver is installed after the non-VMD RST drivers. You can learn more about MDT Exit scripts here: https://www.deploymentresearch.com/understanding-the-mdt-lite-touch-exits-feature/

3. Using the **Deployment Workbench**, in the **MDT Production** node, select the **Out-Of-Box Drivers** node, and create a folder named **WinPE x64**.

4. Right-click the **WinPE x64** folder and select **Import Drivers**, use the following settings for the **Import Driver** wizard.

   Driver source directory: **E:\Drivers\WinPE x64**



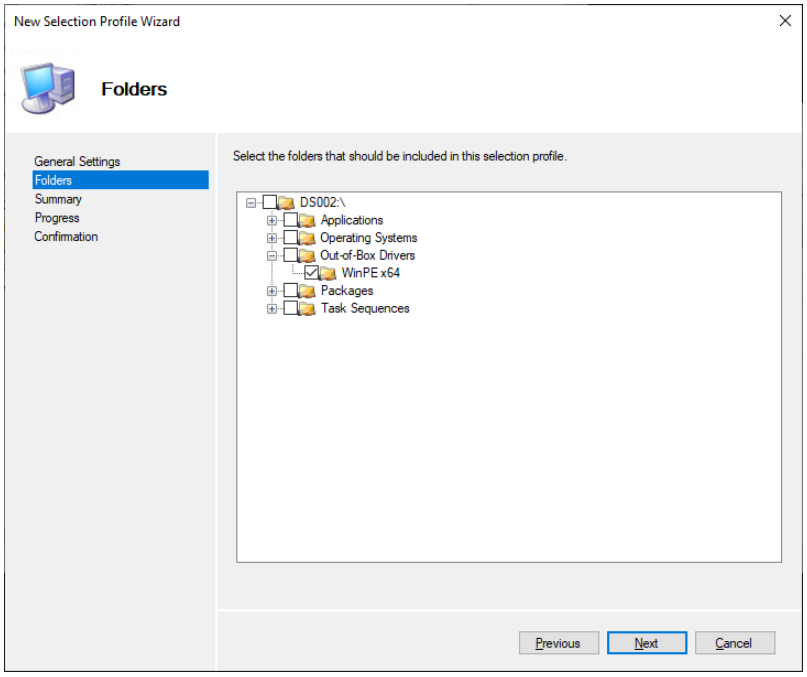
The HP WinPE drivers added.

5. Expand the **Advanced Configuration** node, right-click **Selection Profiles**, and select **New Selection Profile**. Use the following settings for **New Selection Profile** wizard.

   a. Selection profile name: **WinPE x64**

   b. Folders: select the **WinPE x64** folder in **Out-Of-Box Drivers**

© Copyright 2022 Deployment Artist



Creating the WinPE x64 selection profile.

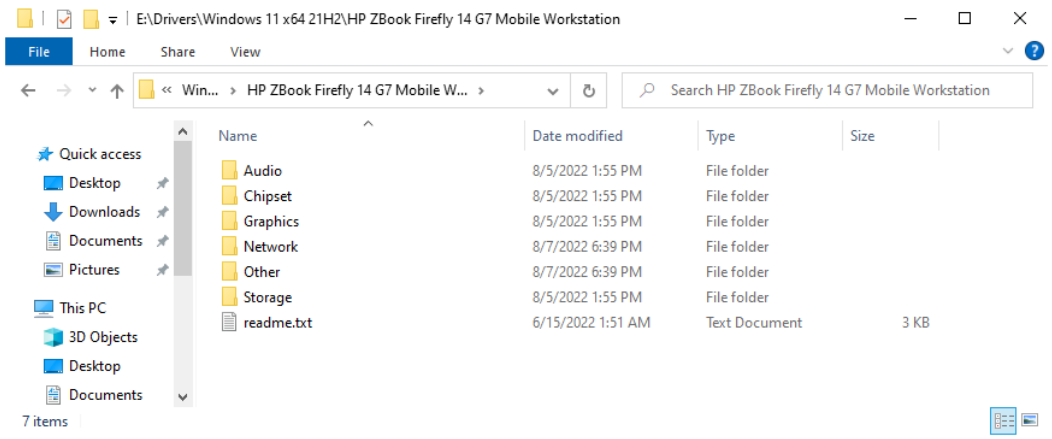
Add Windows 11 Drivers for HP ZBook Firefly 14 G7 Mobile Workstation

In this section you learn to add drivers for the HP ZBook Firefly 14 G7 Mobile Workstation model to MDT Lite Touch.

1. On **MDT01**, download the **HP ZBook Firefly 14 G7 Mobile Workstation** drivers, and extract them to the **E:\Drivers\Windows 11 x64 21H2\HP ZBook Firefly 14 G7 Mobile Workstation** folder (create the folder structure).

Real World Note: The x64 architecture in the Windows 11 folder is not really needed since Windows 11 is not available for the x86 architecture, but I recommend adding it to be consistent with any Windows 10 deployment you might still be doing.

© Copyright 2022 Deployment Artist

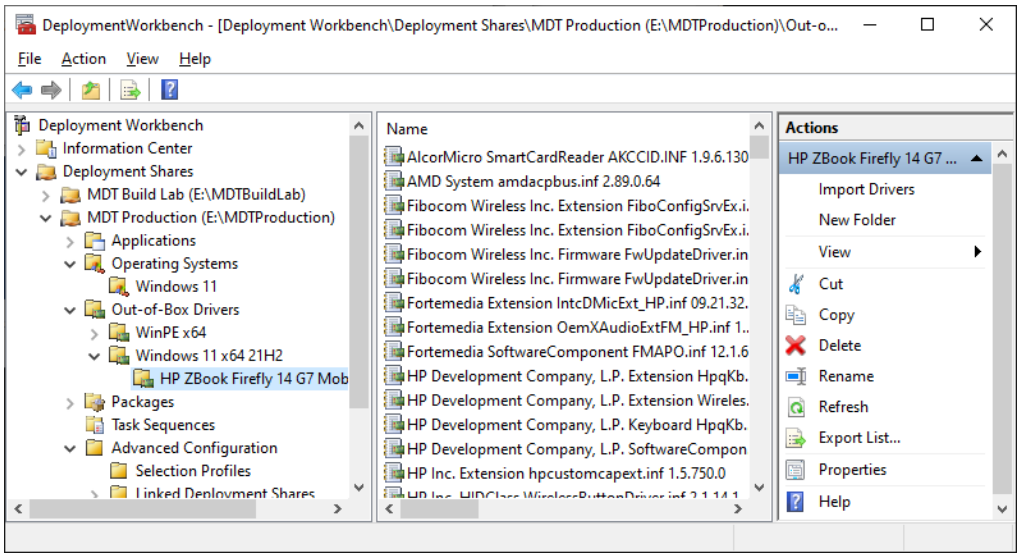


The drivers for HP ZBook Firefly 14 G7 Mobile Workstation copied.

2. In the **Out-Of-Box Drivers** node, create a folder named **Windows 11 x64 21H2**. Then, in the **Windows 11 x64 21H2** folder, create a subfolder named **HP ZBook Firefly 14 G7 Mobile Workstation**.

3. Right-click the **HP ZBook Firefly 14 G7 Mobile Workstation** folder and select **Import Drivers**, use the following settings for the **Import Driver** wizard.

   Driver source directory: **E:\Drivers\Windows 11 x64 21H2\HP ZBook Firefly 14 G7 Mobile Workstation**



MDT Production deployment share with imported drivers.

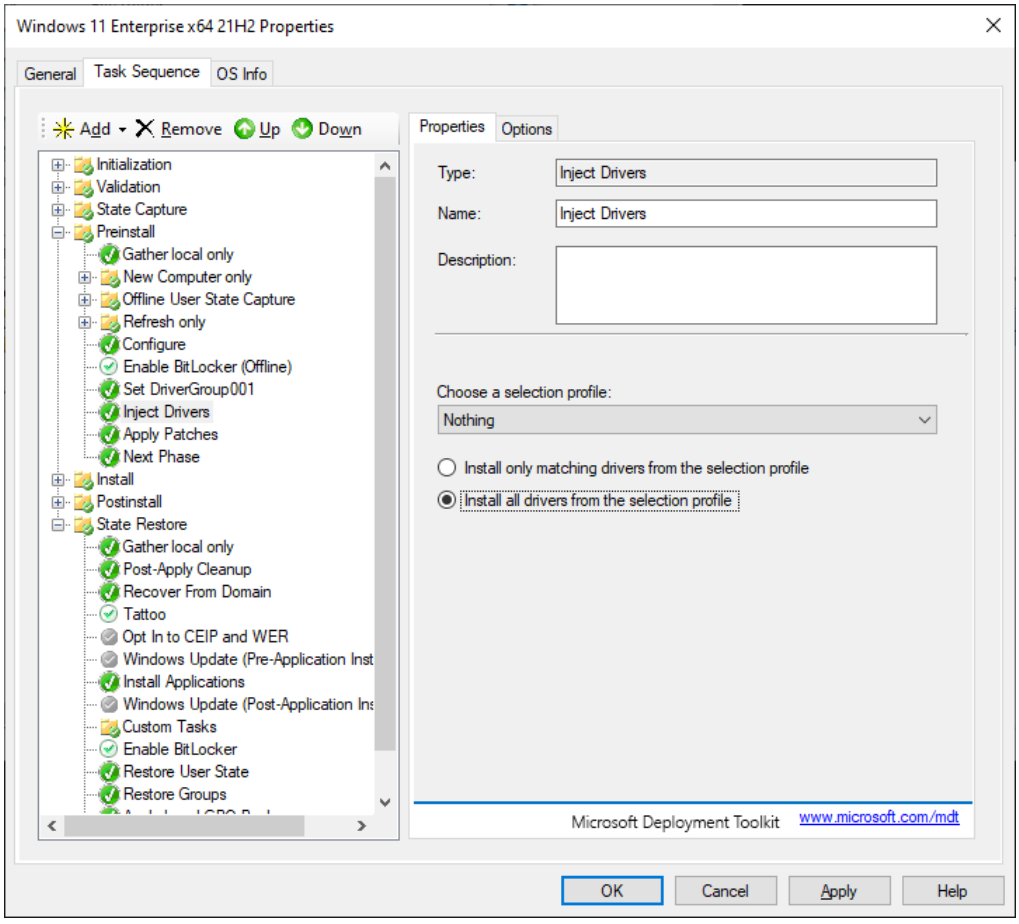
© Copyright 2022 Deployment Artist

Step 4 – Create the Production Task Sequences

Create and configure a Windows 11 Task Sequence

1. On **MDT01**, using the **Deployment Workbench**, in the **MDT Production** deployment share, select the **Task Sequences** node, and create a new folder named **Windows 11**.
2. Right-click on the **Windows 11** node, and select **New Task Sequence**. Use the following settings for the **New Task Sequence Wizard**.
    a. Task sequence ID: **W11-X64-001**
    b. Task sequence name: **Windows 11 Enterprise x64 21H2**
    c. Task sequence comments: **Production image**
    d. Template**: Standard Client Task Sequence**
    e. Select OS: **Windows 11 Enterprise x64 21H2**
    f. Specify Product Key: **Do not specify a product key at this time**
    g. FullName: **ViaMonstra**
    h. Organization: **ViaMonstra**
    i. Internet Explorer home page: **about:blank**
    j. **Do not specify an Administrator password at this time**
3. In the **Task Sequences / Windows 11** node, double-click the **Windows 11 Enterprise x64 21H2** task sequence, and click the **Task Sequence** tab.
4. Expand the **Preinstall** group. After the **Enable BitLocker (Offline)** action, add a new **Set Task Sequence Variable** action with the following settings:
    a. Name: **Set DriverGroup001**
    b. Task Sequence variable: **DriverGroup001**
    c. Value: **Windows 11 x64 21H2\%Model%**
5. Select the **Inject Drivers** action, and configure the following:
    a. Choose a selection profile: **Nothing**
    b. **Install all drivers from the selection profile**
6. Click **OK.**

© Copyright 2022 Deployment Artist



The task sequence configured for drivers, total control style.

© Copyright 2022 Deployment Artist

Step 5 – Configure the Deployment Share

Prepare the deployment share rules

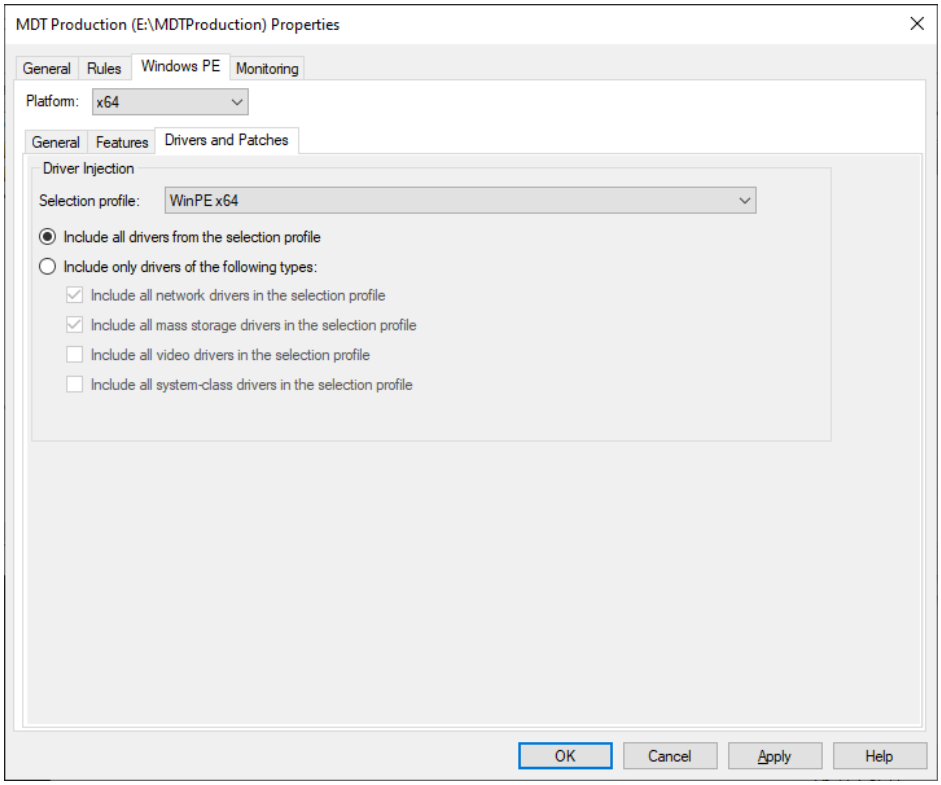
1. On **MDT01**, using **File Explorer**, navigate to the **C:\Setup\MDT\MDTProduction** folder.
2. Copy the following files to **E:\MDTProduction\Control** (replace the existing files).
    a. **Bootstrap.ini**
    b. **CustomSettings.ini**
3. Using **Deployment Workbench**, right-click the **MDT Production** Deployment Share and select **Properties**.
4. In the **Rules** tab, change the **HideShell=YES** line to **HideShell=NO**. Then click **OK**.

Real World Note: While the Microsoft.BDD.Utility.dll - which is hosting the HideShell function - does run fine on Windows 11, there is a known timing issue with the first run animation feature in both Windows 10 and Windows 11. For now, I recommend disabling HideShell by setting its value to NO. You can learn more about the HideShell behavior here: https://techcommunity.microsoft.com/t5/windows-blog-archive/windows-10-mdt-2013-update-1-and-hideshell/ba-p/706592

Configure the Deployment Share Settings

1. Right-click the **MDT Production** Deployment Share and select **Properties**.
2. In the **General** tab, in the **Platforms Supported** area, clear the **x86** check box.
3. In the **Windows PE** tab, in the **Platform** dropdown list, make sure **x64** is selected.
4. In the **General** tab, in the **Lite Touch Boot Image Settings** area, configure the following settings
    c. Image description: **MDT Production x64**
    d. ISO file name: **MDT Production x64.iso**
5. In the **Drivers and Patches** tab, configure the following settings:
    a. Selection profile: **WinPE x64**
    b. **Include all drivers from the selection profile**
6. In the **Monitoring** tab, select the **Enable monitoring for this deployment share** check box, and then click **OK**.
7. Review the **E:\MDTProduction\Control\CustomSettings.ini** file and note the added **EventService=http://MDT01:9800** line in the end that configures the deployment to send monitoring information.

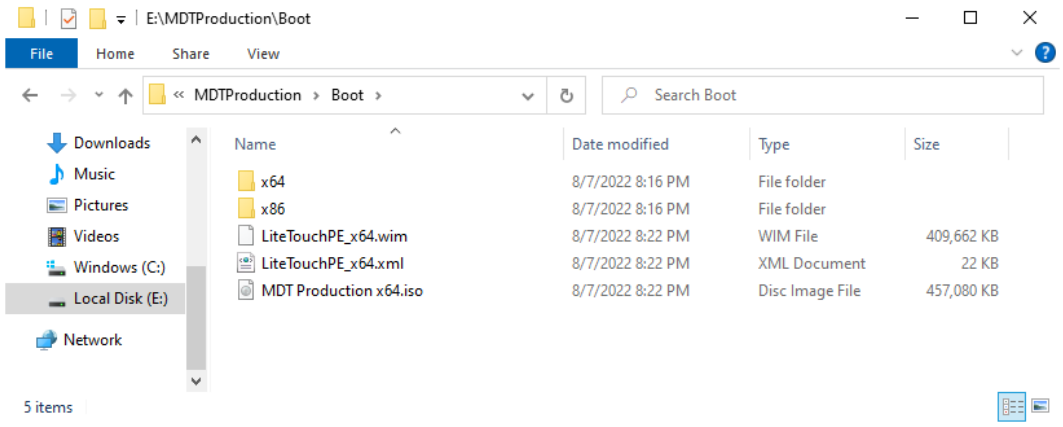
© Copyright 2022 Deployment Artist



Configuring the MDT Production deployment share properties.

Update the Deployment Share

1. Right-click the **MDT Production** Deployment Share and select **Update Deployment Share**.
2. Use the default options for the **Update Deployment Share Wizard**.
3. After the update, you should have the following files in the **E:\MDTProduction\Boot** folder:



E:\MDTProduction\Boot folder after deployment share update.

© Copyright 2022 Deployment Artist

© Copyright 2022 Deployment Artist

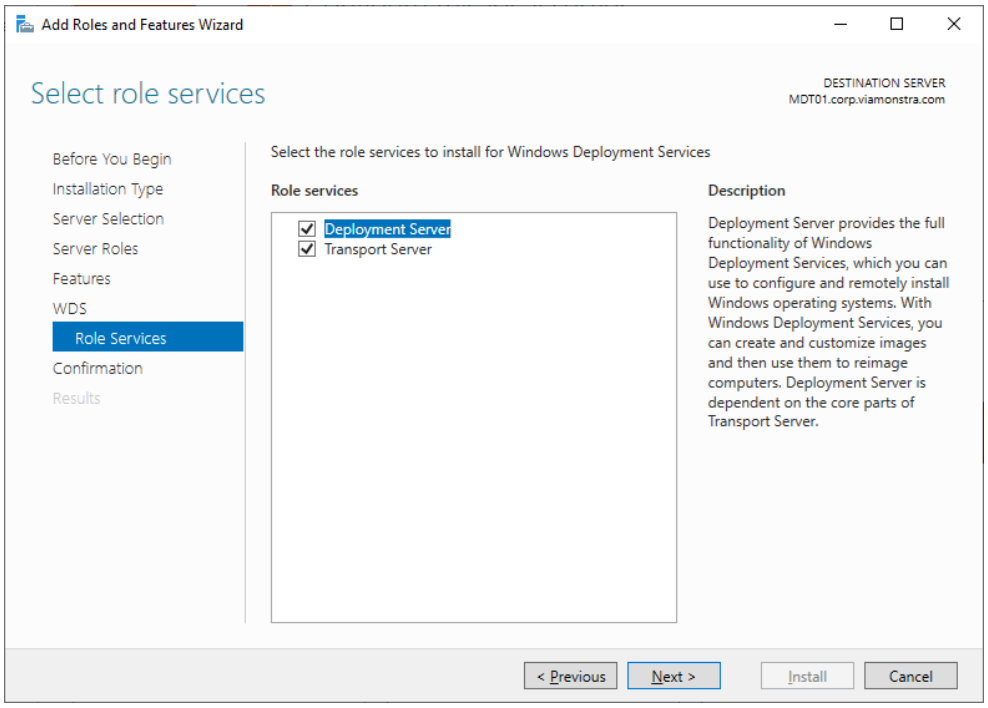
Step 6 – Install and Configure WDS

In this step you enable PXE boot for MDT Lite Touch via Windows Deployment Services (WDS).

Real Word Note: While Microsoft is deprecating the option of running Setup.exe via boot.wim in WDS running on Windows Server 2022. Regular PXE boot is not affected. You can continue to PXE boot Windows deployments via MDT Lite Touch, or any other solution needing to boot WinPE over the network. You can read more about these changes here: https://docs.microsoft.com/en-us/windows-server/get-started/removed-deprecated-features-windows-server-2022

Install WDS

1. On **MDT01**, using **Server Manager**, select **Add roles and features**
2. On the **Before you begin** page, click **Next**.
3. On the **Select installation type** page, accept the default settings, and click **Next**.
4. On the **Select destination server** page, click **Next**.
5. On the **Select server roles** page, select the **Windows Deployment Services** check box, click **Add Features**, and then click **Next**.
6. On the **Select features** page, accept the default settings, and click **Next**
7. On the **WDS** page, click **Next**
8. On the **Select role services** page, accept the default settings, and click **Next**.



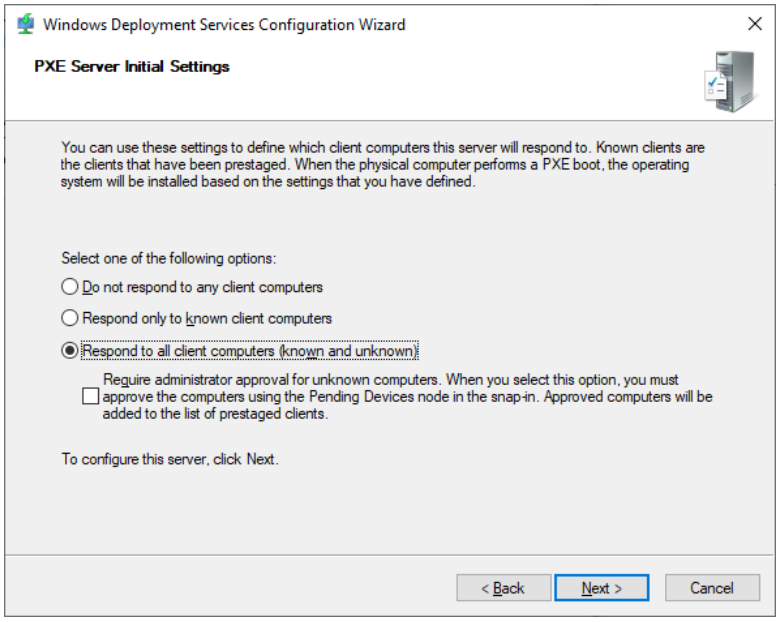
© Copyright 2022 Deployment Artist

Selecting the role services for WDS.

9. On the **Confirm installation selections** page, click **Install**, and once the installation is completed, click **Close**.

Configure WDS

1. In **Server Manager**, click **Tools**, and then select **Windows Deployment Services**.
2. Expand the **Servers** node, right-click **MDT01.corp.viamonstra.com**, and select **Configure Server**.
3. On the **Before you begin** page, click **Next**.
4. On the **Install options** page, select the **Integrated with Active Directory** option, and click **Next**.
5. On the **Remote Installation Folder Location** page, in the **Path** text box, type in **E:\RemoteInstall** and click **Next**.
6. On the **PXE Server Initial Settings** page, select the **Respond to all client computers (known and unknown)** option, and then click **Next**.



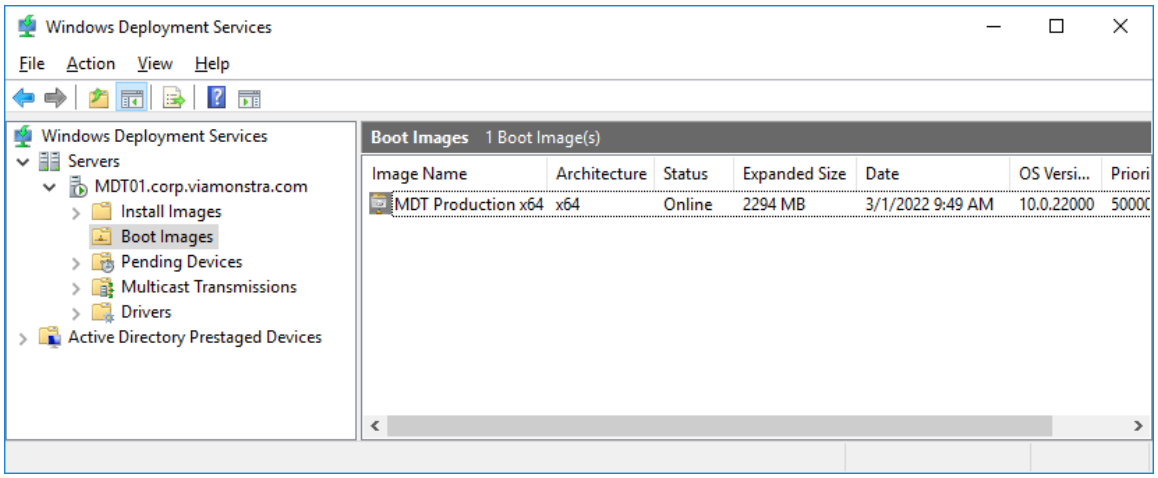
Configuring the WDS initial settings.

7. On the **Operation Complete** page, clear the **Add images to the server now** check box, and click **Finish**.

© Copyright 2022 Deployment Artist

Add the MDT Lite Touch Boot Image

1. In the **WDS Console**, in the **Servers** node, expand **MDT01.corp.viamonstra.com**, right-click **Boot Images** and select **Add Boot Image**
2. Browse to the **E:\MDTProduction\Boot\LiteTouchPE_x64.wim** file and add the image with the default settings.



The MDT Production x64 boot image added.

© Copyright 2022 Deployment Artist

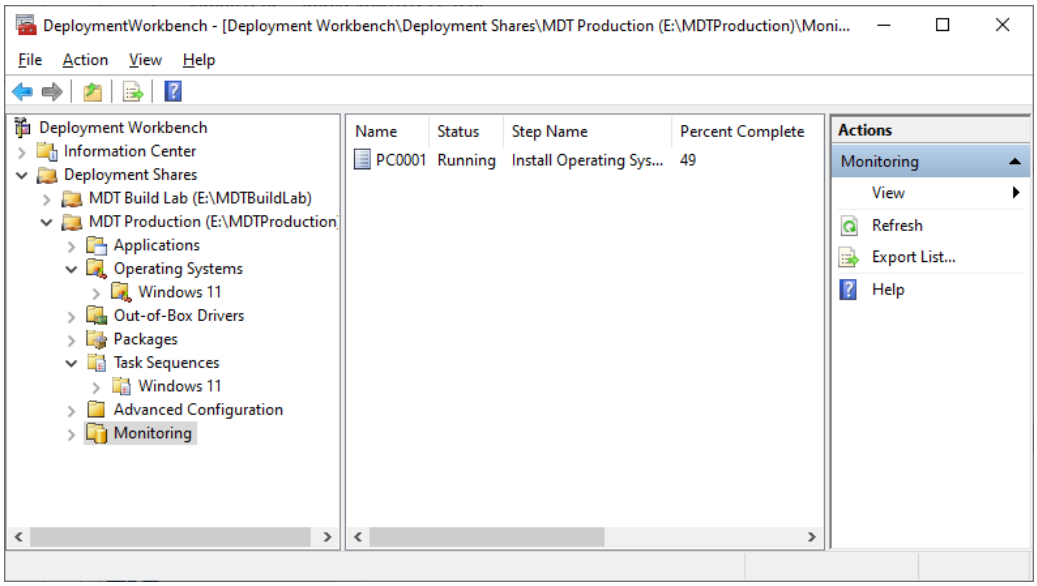
Step 7 – Deploy Windows using PXE

Deploy the Windows 11 Enterprise Image to an UEFI machine

1. On your Hyper-V or VMware PC, create a virtual machine named **PC0001**, with 2 vCPUs, 4 GB RAM, UEFI type, and a 250 GB hard drive (dynamic type so the VM won't allocate the full 250 GB). Also configure the VM to boot from network (PXE).

Real World Note: When testing bare metal deployments, I found it really useful to take a snapshot/checkpoint of the virtual machine before deploying it. That way you can easily revert back if something went wrong, fix the issue, and try again.

2. Start the **PC0001** virtual machine, and when the PXE menu is displayed, press **Enter** to PXE boot when prompted (you need to be quick).
3. After WinPE has been started, in the **Credentials** dialog box, type in a password of **P@ssw0rd**.
4. Complete the MDT deployment wizard using the following settings:
    a. Select a task sequence to execute on this computer:
    Windows 11 Enterprise x64 21H2
    b. Computer Name: **PC0001**
5. When the task sequence starts, on **MDT01**, using the **Deployment Workbench**, use the **Monitoring** node to view the deployment process (press F5 to refresh).



PC0001 in deployment.

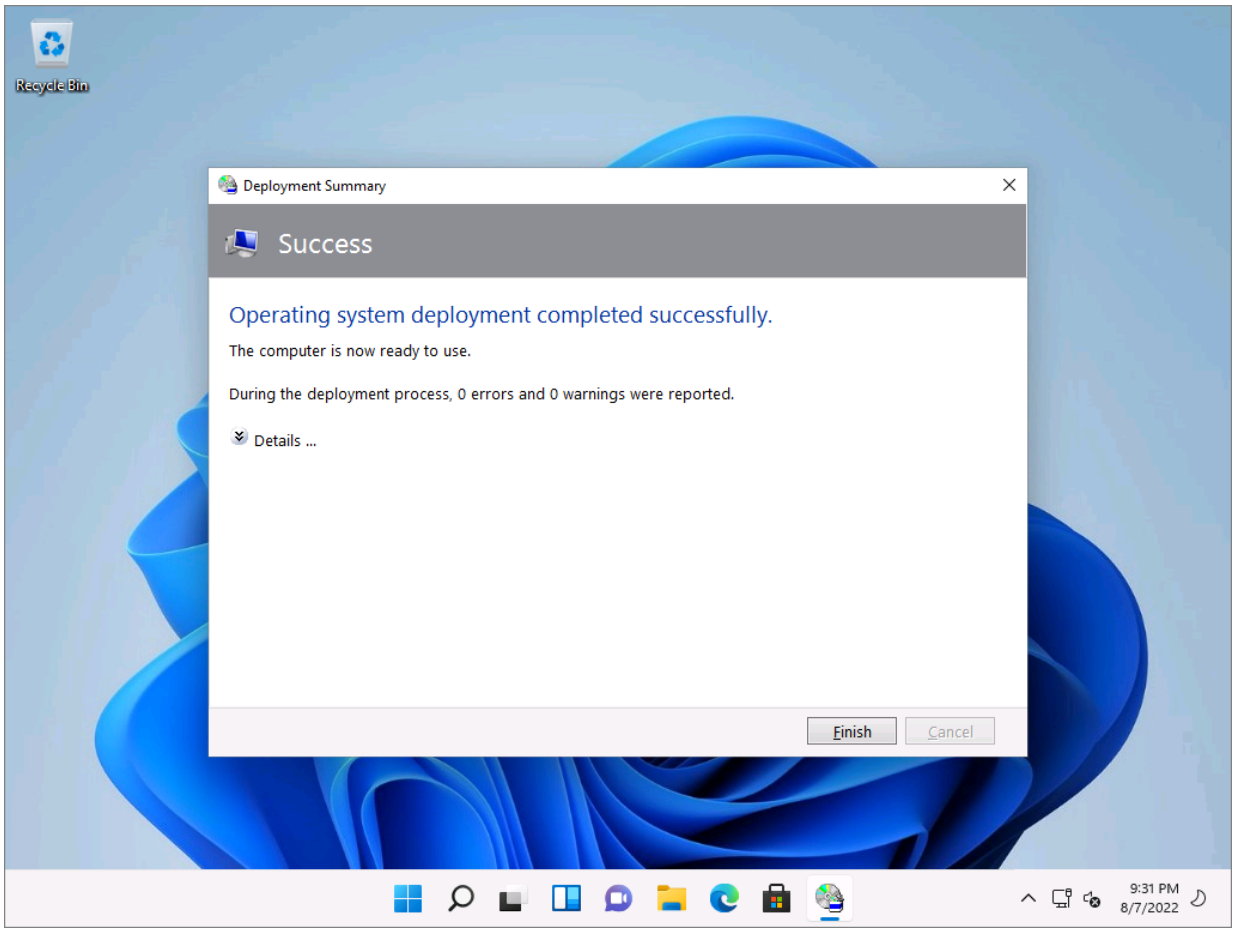
6. The Task Sequence will now run and do the following:

a. Install the Windows 11 operating system

b. Inject the drivers

c. Join the machine to the domain

7. Allow the installation to complete, on the **Deployment Summary** page, click **Finish**.



PC0001 Successfully deployed with Windows 11 Enterprise.

© Copyright 2022 Deployment Artist

Module 6

Setup ConfigMgr for OSD

In this module you learn to setup ConfigMgr current branch site server for deploying machines with Windows 11 Enterprise 21H2.

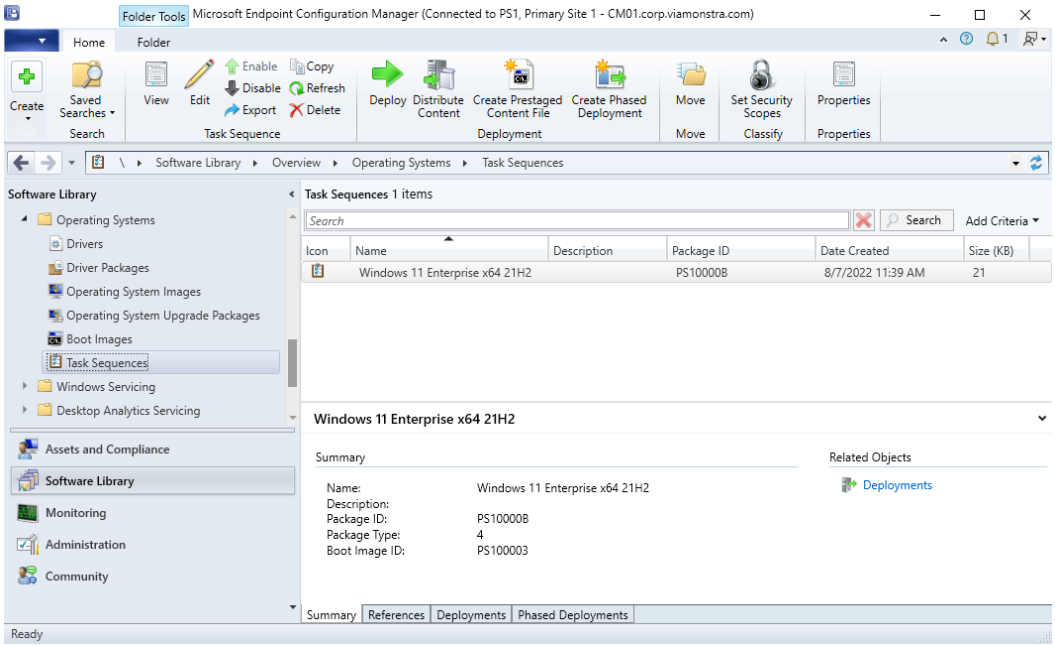
Real World Note: If still using ConfigMgr 2107, make sure you have installed KB4567007 which contains OSD fixes for UEFI-based machines. Or simply upgrade to ConfigMgr 2111 or later.

What you need for this module

For the guides in this module, you need the following:

- A Domain Controller running Windows Server 2016 or later. In my lab I used a domain controller named DC01 running Windows Server 2022 that manages the corp.viamonstra.com domain (The NetBIOS name for the domain is VIAMONSTRA).
- A Member Server running Windows Server 2016 or later with a ConfigMgr site server installed. In my lab I was using Windows Server 2022, with CM01 as a single site server for the PS1 primary site, having all the needed roles, like MP, DP, SMS provider, and site database server. The ConfigMgr version used was 2203 with the latest hotfixes.
- A Windows 11 Enterprise 21H2 WIM image created via the steps in ether Module 3 or Module 4.
- The sample files for this guide: https://github.com/DeploymentResearch/DRFiles/tree/master/GeeksGuideW11

© Copyright 2022 Deployment Artist



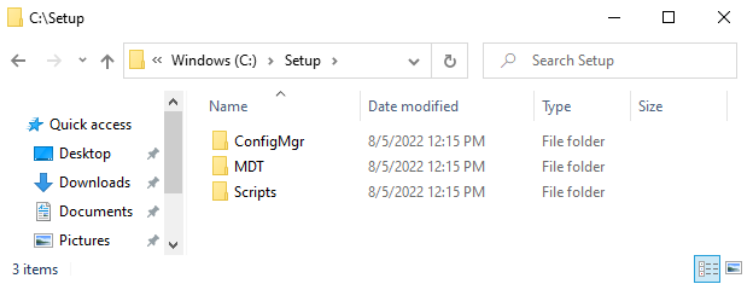
The admin console for ConfigMgr 2203.

Step 1 – Prepare for ConfigMgr OSD

Copy the Sample Files to CM01

In this section I assume you have downloaded the sample files for this guide.

1. On **CM01**, create the **C:\Setup** folder.
2. Extract the sample files to **C:\Setup**



The C:\Setup folder containing the sample files.

Review the Service Account for ConfigMgr OSD

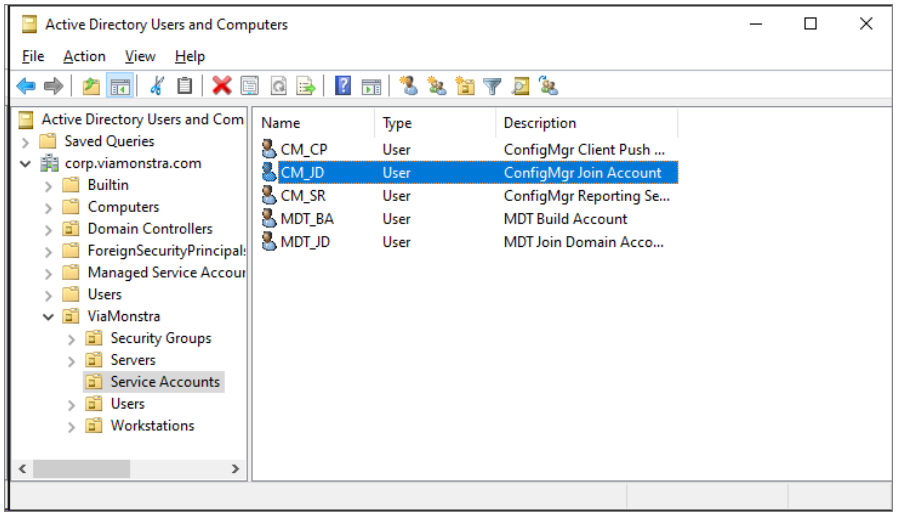
1. On **DC01**, using **Active Directory Users and Computers**, make sure you have created the following user account:

© Copyright 2022 Deployment Artist

a. **CM_JD**

Security Note: If you configure the ConfigMgr site for HTTPS or Enhanced HTTP, you no longer need to use a network access account for OSD. This greatly improves the security in your ConfigMgr environment and is highly recommended.

2. In my lab I added the service account to the **ViaMonstra / Service Accounts** OU, and I assigned a password of **P@ssw0rd**.



The Service Accounts OU in my lab.

Configure Active Directory Permissions

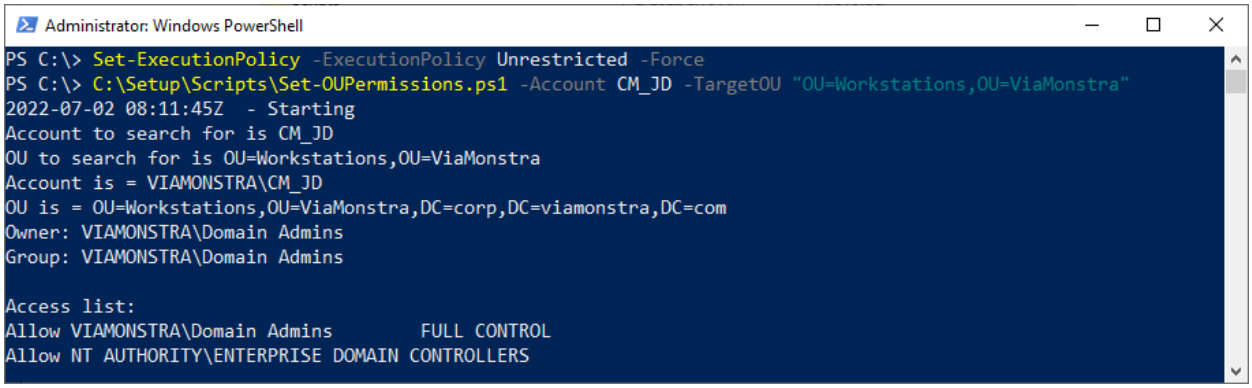
For the ConfigMgr Task Sequence to join machines into your domain, the domain join account needs to have permissions to both create and update computer objects (depending on the computer object already existing or not). I recommend using a script to set the minimal permissions needed which is what you will do in this guide. In this section I assume you have downloaded the sample files for this guide.

1. On **DC01**, create the **C:\Setup** folder.
2. Extract the sample files to **C:\Setup**.
3. In an elevated **PowerShell prompt** (run as Administrator), run the following commands, press **Enter** after each command (please note that the last command is wrapped):

```
Set-ExecutionPolicy -ExecutionPolicy Unrestricted -Force

C:\Setup\Scripts\Set-OUPermissions.ps1
-Account CM_JD -TargetOU "OU=Workstations,OU=ViaMonstra"
```

© Copyright 2022 Deployment Artist



Running the Set-OUPermissions.ps1 script.

4. The preceding script allows the CM_JD user account minimal permissions to manage computer accounts in the ViaMonstra / Workstations OU.

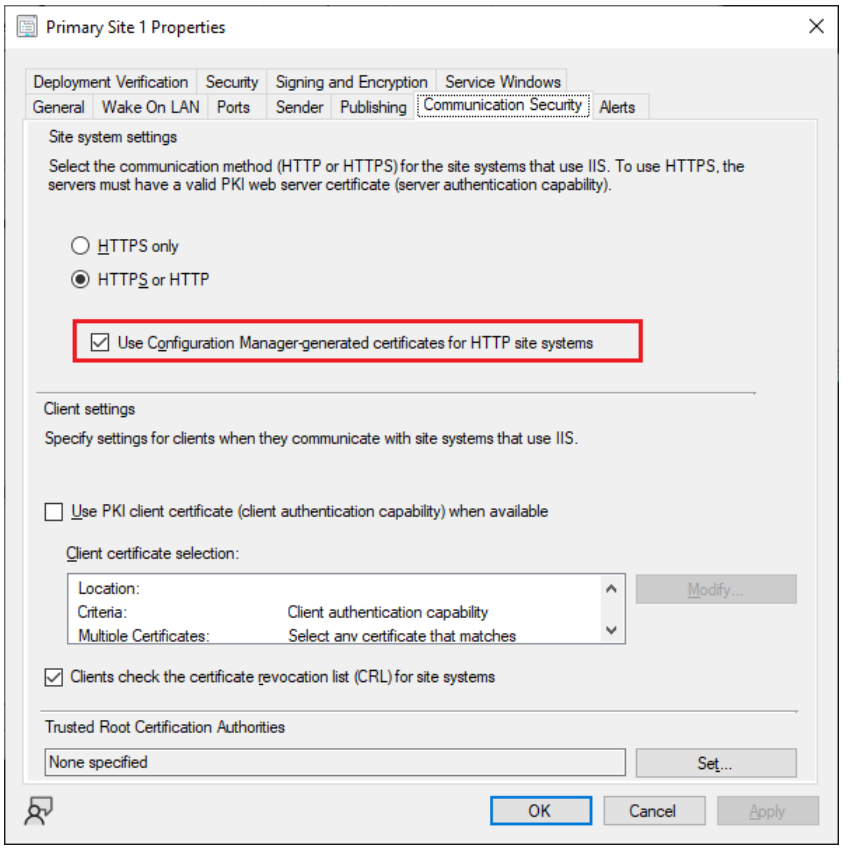
© Copyright 2022 Deployment Artist

Enable Enhanced HTTP

To enable imaging support without the need for a network access account, you should either setup ConfigMgr for a full PKI (HTTPS) setup, or enable Enhanced HTTP. In this guide you'll do the latter.

Real World Note: While using a network access account is still supported in ConfigMgr it's not recommended for security reasons. Any user on a Windows machine deployed with a network access account can retrieve the username and password for the network access account with publicly available scripts and tools.

1. On **CM01**, using the **Configuration Manager Console**, in the **Administration** workspace, in **Site Configuration**, select **Sites**, and then select your site in the right pane.
2. On the Ribbon bar, select **Properties**, and then click the **Communication Security** tab.
3. Select the **Use Configuration Manager-generated certificates for HTTP site systems** check box.



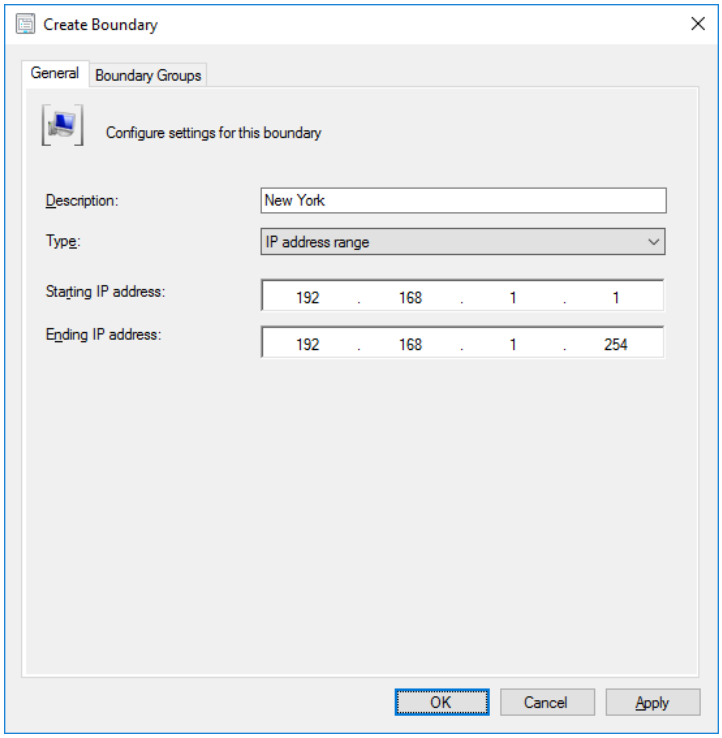
Configuring the site for Enhanced HTTP.

© Copyright 2022 Deployment Artist

Create a boundary

In ConfigMgr, boundaries and boundary groups are used for clients to locate management points, software update points, and content on distribution points. I prefer to use IP ranges, and your lab you should at least one of these. In this example you create an IP range for the 192.168.1.0/24 subnet.

1. On **CM01**, using the **Configuration Manager Console**, in the **Administration** workspace, in **Hierarchy Configuration**, select **Boundaries**.
2. Create a boundary using the following settings:
    a. Description: **New York**
    b. Type: **IP address range**
    c. Starting IP address: **192.168.1.1**
    d. Ending IP address: **192.168.1.254**



Creating an IP range boundary.

Create a Boundary Group

1. Using the **Configuration Manager Console**, in the **Administration** workspace, in **Hierarchy Configuration**, select **Boundary Groups**.
2. Create a boundary group using the following settings:

© Copyright 2022 Deployment Artist

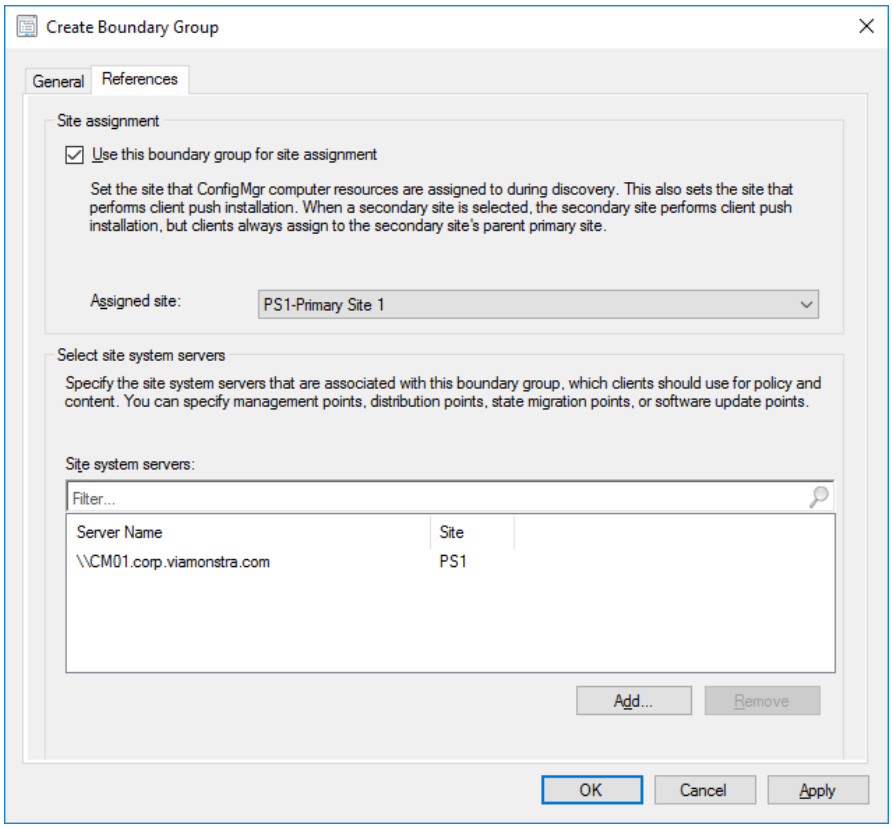
a. In the **General** tab

- Name: **HQ Assignment**
- Boundaries: Add the **192.168.1.1 – 192.168.1.254** boundary.

© Copyright 2022 Deployment Artist

b. In the **References** tab

- Site assignment area: Select the **Use the boundary group for site assignment** check box.
- **Site system servers:** area: Add the **CM01** server.



Creating the HQ Assignment boundary group.

Create a Distribution Point Group

To simplify distribution of content in ConfigMgr I recommend using distribution point groups. A nice feature of a distribution point group is that it will automatically distribute content to new distribution points that you add to the group.

1. Using the **Configuration Manager Console**, in the **Administration** workspace, create a distribution point group named **HQ DPs**.
2. In the **Members** tab, add the **CM01** distribution point.

Configure the Client Settings

While configuring client settings for production environments is out of scope for this guide, you should at least set some basic settings for OSD to work properly.

© Copyright 2022 Deployment Artist

1. On **CM01**, using the **Configuration Manager Console**, in the **Administration** workspace, select **Client Settings**.
2. Right-click **Default Client Settings**, and select **Properties**.

© Copyright 2022 Deployment Artist

3. In the **Client Cache Settings** node, configure the following settings

    a. Configure client cache size: **Yes**

    b. Maximum cache size (MB): **25600**

    c. Maximum cache size (percentage of disk): **25**

Note: With the above settings the client cache size can expand to either the maximum size in MB (25600), or the percentage of the disk (25), whichever is less.

4. In the **Computer Agent** node, configure the following settings

    a. Organization name displayed in Software Center: **ViaMonstra**

Note: The preceding setting is still need for the organization name to show during OS deployment.

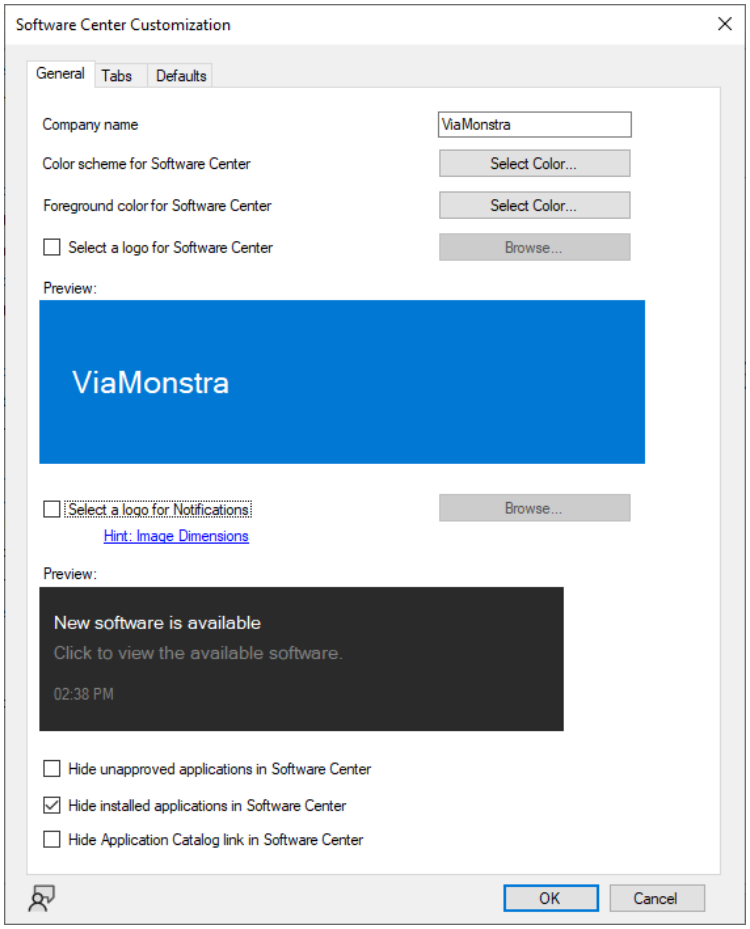
5. In the **Software Center** node, configure the following settings

    a. Select these new settings to specify company information: **Yes**

    b. Click **Customize**, and In the **Company name** text box, type in **ViaMonstra**.

6. Click **OK** twice.

© Copyright 2022 Deployment Artist



Configure default client settings.

Create the Sources folder structure

1. On **CM01**, in an elevated **PowerShell prompt** (run as Administrator), run the following command (note that command is wrapped):

   ```
   C:\Setup\Scripts\Create-ConfigMgrSourceFolders.ps1
   ```

2. The above script creates the following folder structure, and shares the **E:\Sources** folder as **Sources**.

   E:\Sources

   E:\Sources\OSD

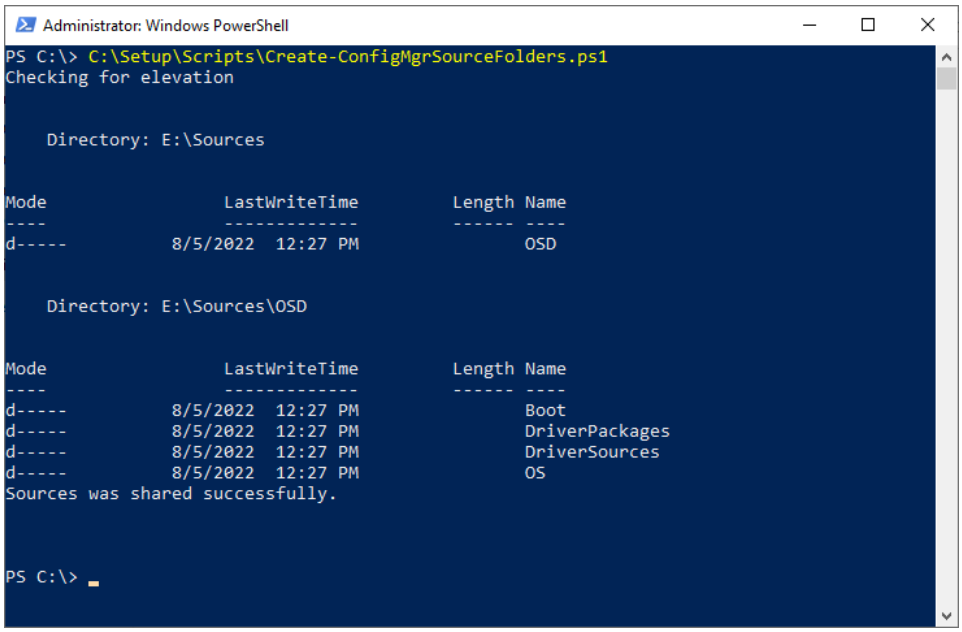
   E:\Sources\OSD\Boot

   E:\Sources\OSD\DriverPackages

   E:\Sources\OSD\DriverSources

   E:\Sources\OSD\OS

© Copyright 2022 Deployment Artist



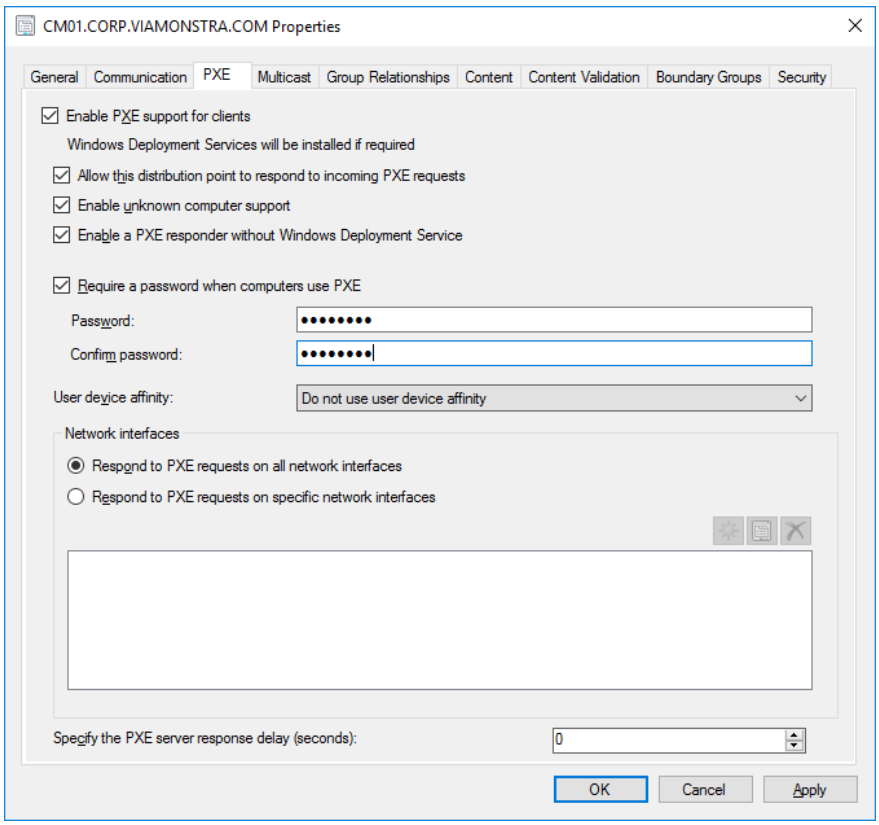
Running the Create-ConfigMgrSourceFolders.ps1 script.

© Copyright 2022 Deployment Artist

Enable PXE on the CM01 Distribution Point

1. In the **Configuration Manager Console**, in the **Administration** workspace, select **Distribution Points**.
2. Right-click the **\\CM01.CORP.VIAMONSTRA.COM** DP and select **Properties**.
3. In the **PXE** tab, enable the following settings:
    a. Enable PXE support for clients
    b. Allow this distribution point to respond to incoming PXE requests
    c. Enable unknown computer support
    d. Enable a PXE responder without Windows Deployment Service
    e. Require a password when computers use PXE
    f. Password and Confirm password: **P@ssw0rd**

Real World Note: The "Enable a PXE responder without Windows Deployment Service" option is a new PXE responder for ConfigMgr, and since it's not depending on WDS, it can be installed on DPs running on Windows 10 or Windows 11 as well.



Configure the CM01 DP for PXE.

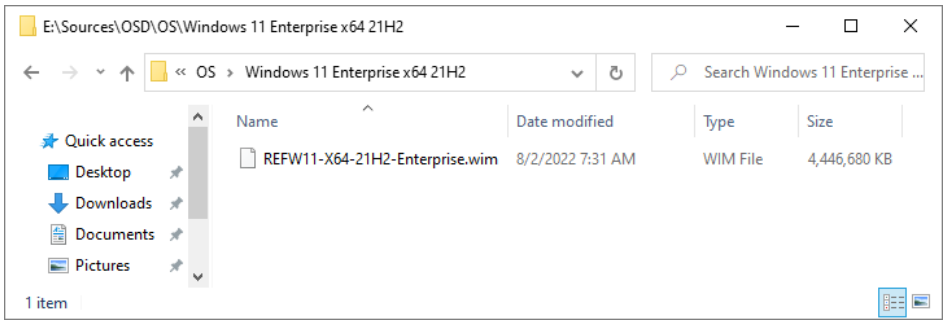
© Copyright 2022 Deployment Artist

Step 2 – Add operating system images

Create the Windows 11 Operating System Image Package

1. On **CM01**, create the **E:\Sources\OSD\OS\Windows 11 Enterprise x64 21H2** folder.
2. Copy your Windows 11 WIM image to the **E:\Sources\OSD\OS\Windows 11 Enterprise x64 21H2** folder.

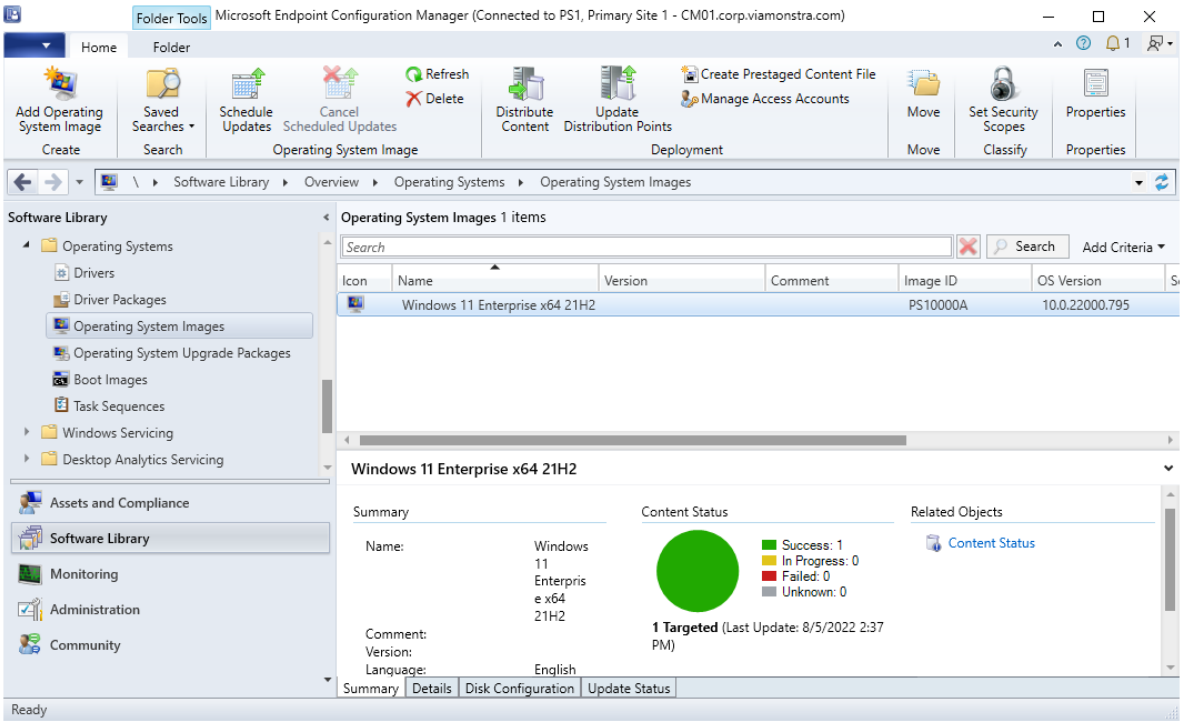
Note: In this example I copied the REFW11-X64-21H2-Enterprise.wim image that I extracted from the download Windows 11 ISO described in Module 3, but you can also copy the WIM image from MDT01 if you did a build and capture as described in Module 4.



The Windows 11 image copied.

3. Using **Configuration Manager Console**, in the **Software Library** workspace, expand the **Operating Systems** node, and then right-click **Operating System Images** and select **Add Operating System Image**. Use the following settings:
    a. Data Source
        a. Path (click Browse): \\cm01\Sources\OSD\OS\Windows 11 Enterprise x64 21H2\REFW11-X64-21H2-Enterprise.wim
        b. Select the Windows license (EULA) check box
    b. Pre-cache settings:
        a. Leave at default, you will not pre-cache a bare metal deployment package.
    c. General
        a. Name: **Windows 11 Enterprise x64 21H2**
4. After adding the operating system image, distribute the **Windows 11 Enterprise x64 21H2** package to the **HQ DPs** distribution point group.

© Copyright 2022 Deployment Artist



The Operating System image added.

© Copyright 2022 Deployment Artist

Step 3 – Add Drivers

To add support for new hardware you may need to add drivers to both WinPE, and to full Windows. In this example you add some WinPE and Windows drivers from HP.

Real World Note: Except for WinPE drivers, there is no need to import any other drivers into the ConfigMgr database. It's much better to create standard packages than to create native driver packages in ConfigMgr. Creating standard packages is much faster, doesn't have distribution issues (hash value is not correct type errors), allows for more flexibility during deployment, and it doesn't bloat the ConfigMgr database with info you never will use.

The drivers referenced in this guide can be downloaded from these URLs:

WinPE x64 drivers
https://ftp.ext.hp.com/pub/caps-softpaq/cmit/softpaq/WinPE10.html

HP ZBook Firefly 14 G7 Mobile Workstation drivers for Windows 11 21H2
https://ftp.hp.com/pub/softpaq/sp140501-141000/sp140701.exe

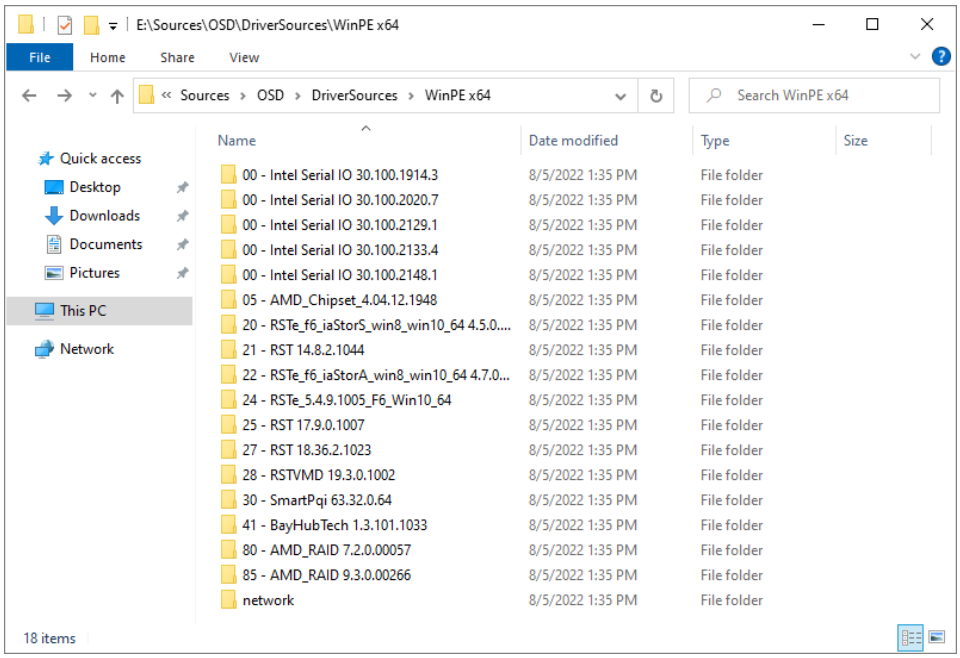
Add Drivers for WinPE

1. On **CM01**, download the **HP Client Windows PE 10 x64 Driver Pack**, and extract it to the **E:\Sources\OSD\DriverSources\WinPE x64** folder (create the folder).

Real World Note: Even if you are using Windows ADK for Windows 11 21H2, you can still use the Windows PE 10 driver pack from HP. It's targeted for both WinPE 10 and WinPE 11.

Also, if you are wondering about the strange folder structure in the HP driver package for WinPE: Its simply to control the installation order, a requirement for some of the Intel RST drivers to work correctly. A little-known fact is that ConfigMgr injects the drivers from the source folder in alphabetic order.

© Copyright 2022 Deployment Artist



The HP WinPE drivers extracted to E:\Sources\OSD\DriverSources\WinPE x64.

2. Using the **Configuration Manager Console**, in the **Software Library** workspace, right-click the **Drivers** folder and select **Import Driver**. Use the following settings for the **Import New Driver Wizard**.

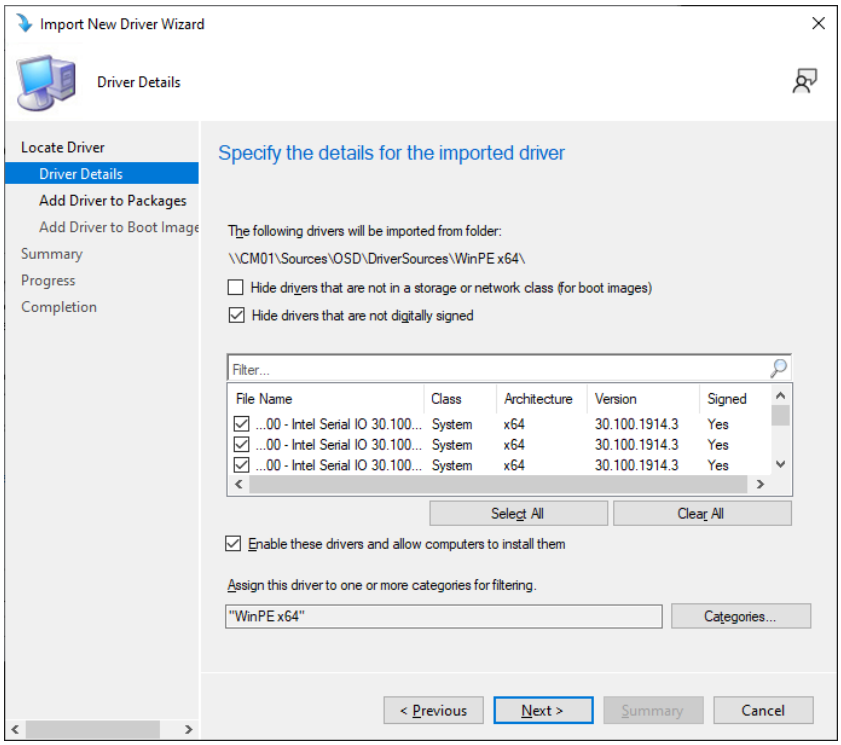
a. Locate Driver

Source folder: **\\CM01\Sources\OSD\DriverSources\WinPE x64**

b. Driver Details (Note the new driver information and filters being available).

Click **Categories**, and create a category named **WinPE x64**

© Copyright 2022 Deployment Artist



Adding HP drivers for WinPE x64.

c. Add Driver to Packages

<default>

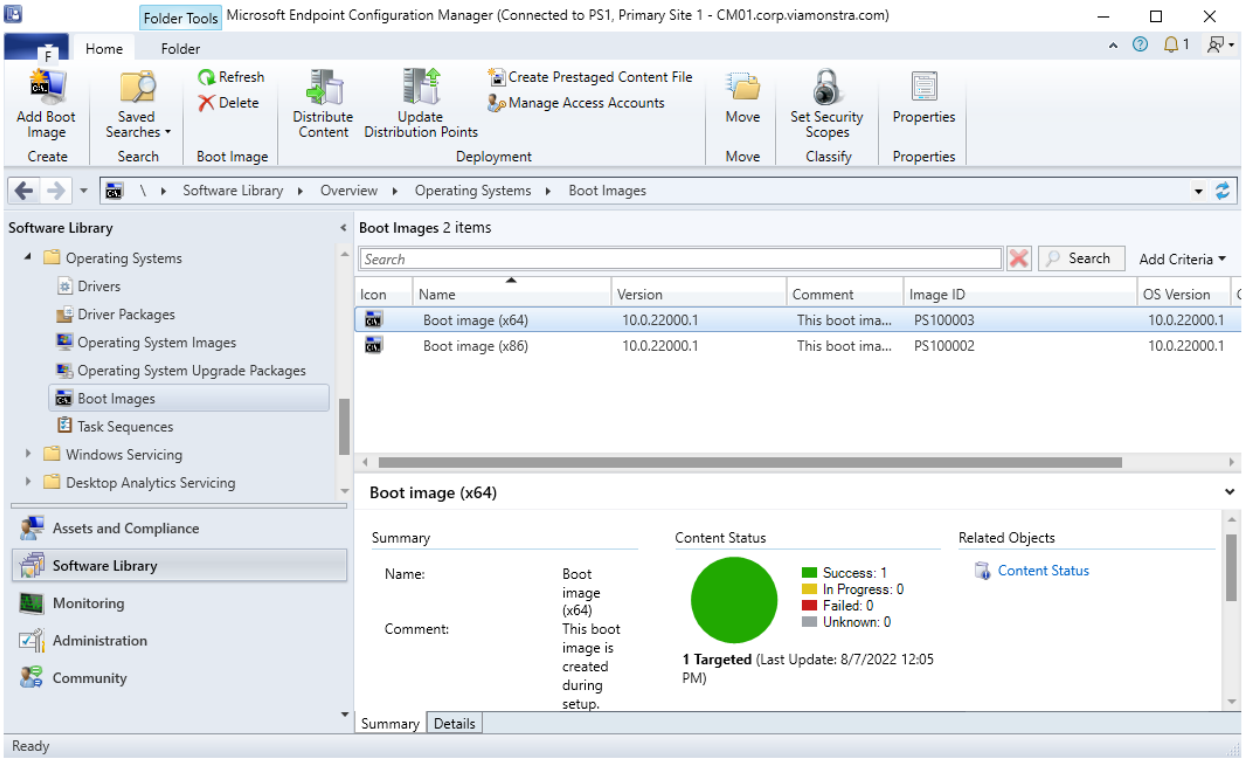
d. Add Driver to Boot Images

i. Select the **Boot Image (x64)** boot image, and click **Next**.

ii. Click **Yes** twice, and then **Next**, to start the update distribution process.

3. After adding the drivers, distribute the **Boot image (x64)** package to the **HQ DPs** distribution point group.

© Copyright 2022 Deployment Artist



The Boot image (x64) package successfully distributed.

© Copyright 2022 Deployment Artist

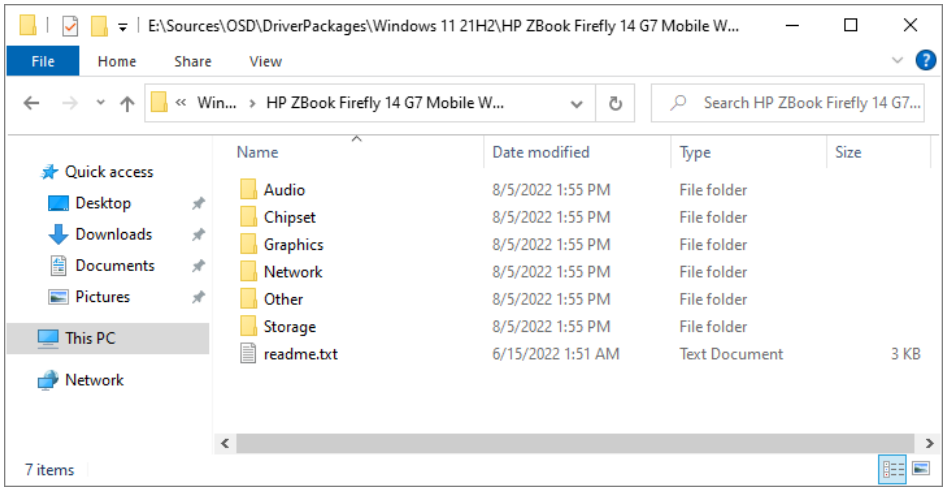
Add Windows 11 Drivers for HP ZBook Firefly 14 G7 Mobile Workstation

In this section you learn to add drivers for the HP ZBook Firefly 14 G7 Mobile Workstation model using a standard ConfigMgr package.

Real World Note: While these steps will work just fine from a technically point of view, I highly recommend looking into using the driver automation tool from Maurice Daly and Nickolaj Andersen. That tool - which is part of the free modern driver management solution - is hands down amazing and simplifies driver management a lot in ConfigMgr. You can read more about that solution here: https://msendpointmgr.com/modern-driver-management/

1. On **CM01**, download the **HP ZBook Firefly 14 G7 Mobile Workstation** drivers, and extract them to the **E:\Sources\OSD\DriverPackages\Windows 11 x64 21H2\HP ZBook Firefly 14 G7 Mobile Workstation** folder (create the folder structure).

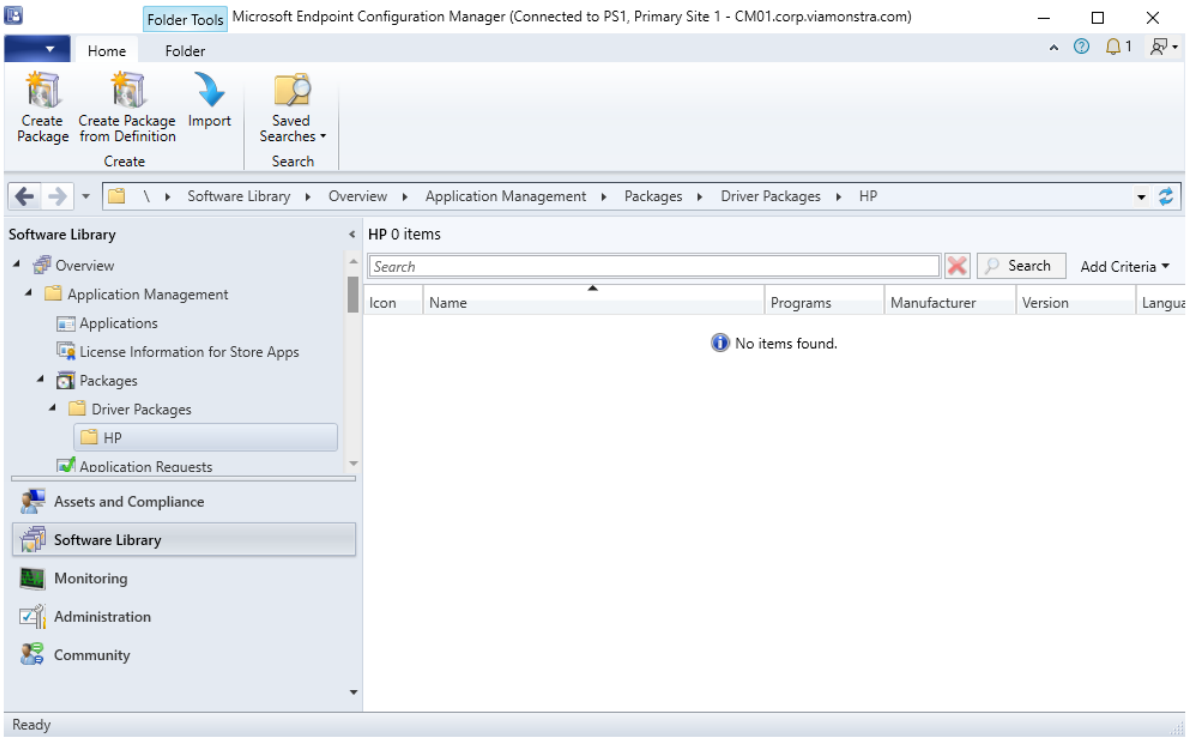
Real World Note: The x64 architecture in the Windows 11 folder is not really needed since Windows 11 is not available for the x86 architecture, but I recommend adding it to be consistent with any Windows 10 deployment you might still be doing.



The drivers for HP ZBook Firefly 14 G7 Mobile Workstation copied.

1. Using the **Configuration Manager Console**, in the **Software Library** workspace, right-click the **Packages** node, select **Folder / Create Folder**, and create the **Driver Packages** folder. Then repeat the process to create the **HP** subfolder in **Driver Packages**.

© Copyright 2022 Deployment Artist



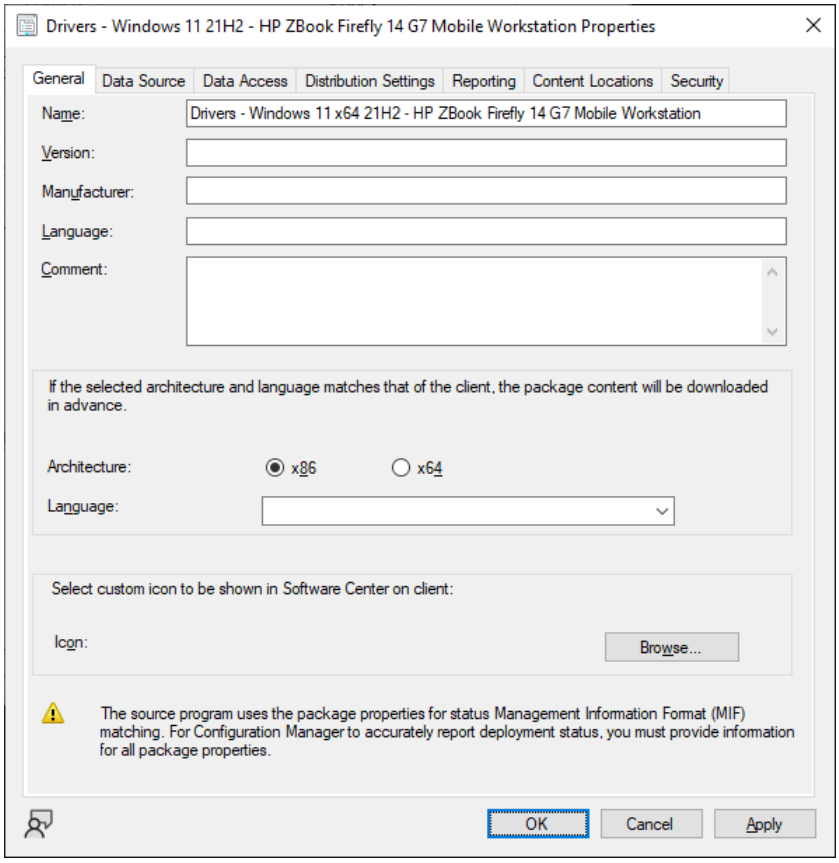
The Driver Packages / HP folder structure created.

2. Right-click the **Driver Packages / HP** folder, and select **Create Package**.
3. Use the following settings for the **Create Package and Program Wizard**.
    a. Name: **Windows 11 x64 21H2 - HP ZBook Firefly 14 G7**
    b. This package contains source files
        - Source folder (click Browse): \\CM01\Sources\OSD\DriverPackages\Windows 11 x64 **21H2\HP ZBook Firefly 14 G7 Mobile Workstation**
    c. Do not create a program
4. After creating the **Windows 11 x64 21H2 - HP ZBook Firefly 14** G7 package, right-click the package, and select **Properties**.
5. In the Name text box, assign the full model name, and a drivers prefix, by changing the name to: **Drivers - Windows 11 x64 21H2 - HP ZBook Firefly 14 G7 Mobile Workstation**

Real World Note: The Create Package and Program Wizard is limited to 50 characters, but you can change that to a longer name after the package has been created. Or by creating the packages via PowerShell. The real limit for a package name is 250 characters.

6. After changing the package name, distribute the **Drivers - Windows 11 x64 21H2 - HP ZBook Firefly 14 G7 Mobile Workstation** package to the **HQ DPs** distribution point group.

© Copyright 2022 Deployment Artist



Changing the name for the driver package.

© Copyright 2022 Deployment Artist

Step 4 – Create Task Sequences

In this section you create a native ConfigMgr task sequence for bare metal deployment of Windows 11.

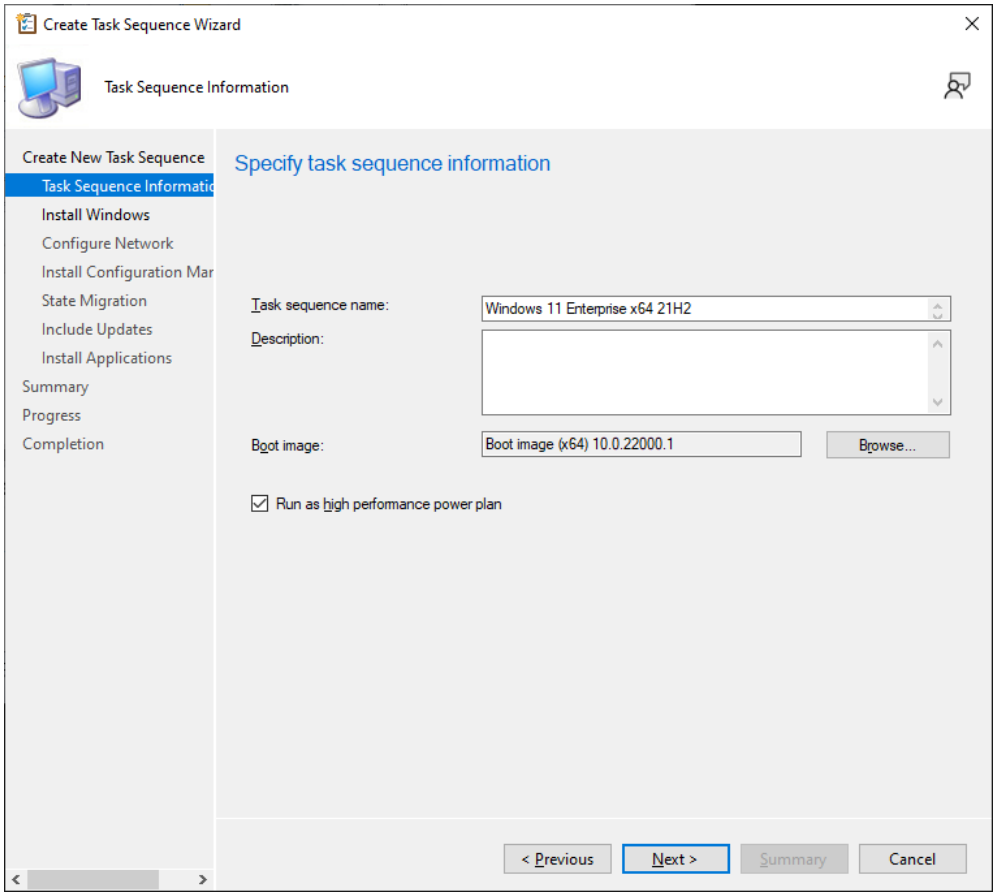
Create a ConfigMgr Task Sequence

1. Using **Configuration Manager Console**, in the **Software Library** workspace, expand **Operating Systems**, right-click **Task Sequences**, and select **Create Task Sequence**.
2. Use the following settings for the **Create Task Sequence** wizard.
    a. Create a new task sequence

        Install an existing image package

    b. Task Sequence Information

        - Task sequence name: **Windows 11 Enterprise x64 21H2**
        - Boot image: **Boot image (x64) 10.0.22000.1**
        - Select the **Run as high performance power plan** check box



© Copyright 2022 Deployment Artist

Specifying the task sequence information.

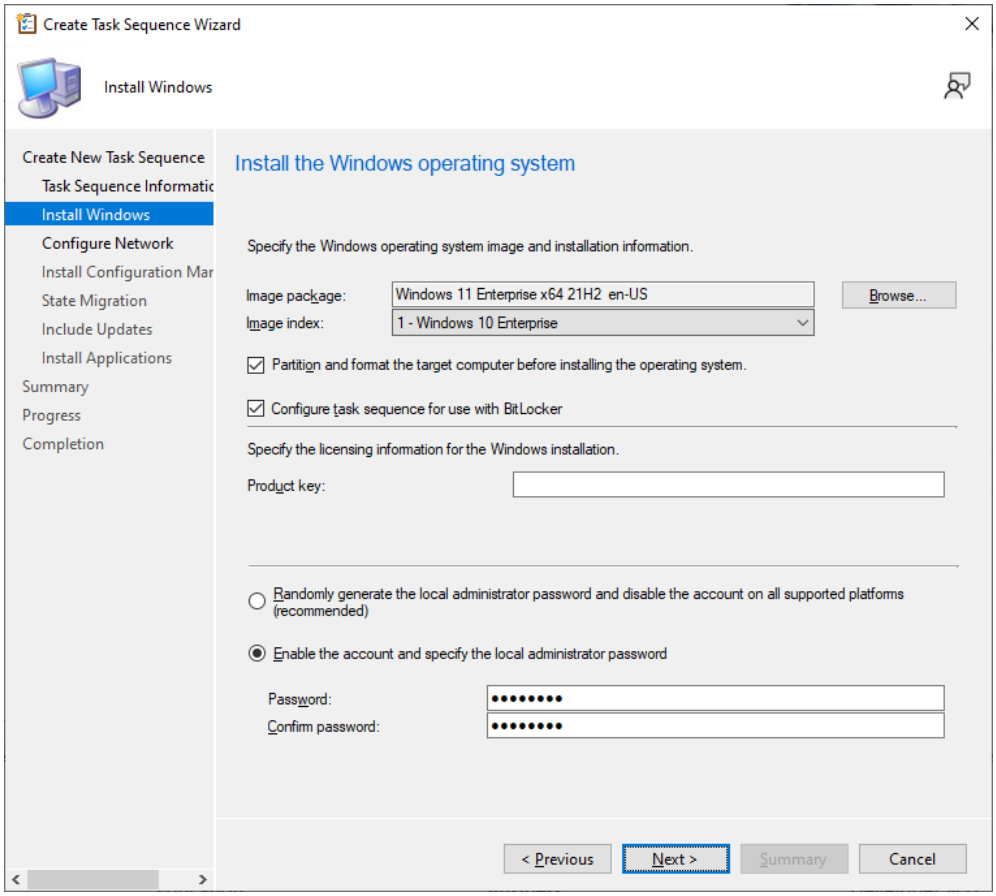
© Copyright 2022 Deployment Artist

c. Install Windows

- Image Package: **Windows 11 Enterprise x64 21H2 en-US**
- Image index: **1- Windows 10 Enterprise**

Real World Note: The image index name for Windows 11 21H2 is still Windows 10 Enterprise.

- Enable the account and specify the local administrator password
  - Password and Confirm password: **P@ssw0rd**



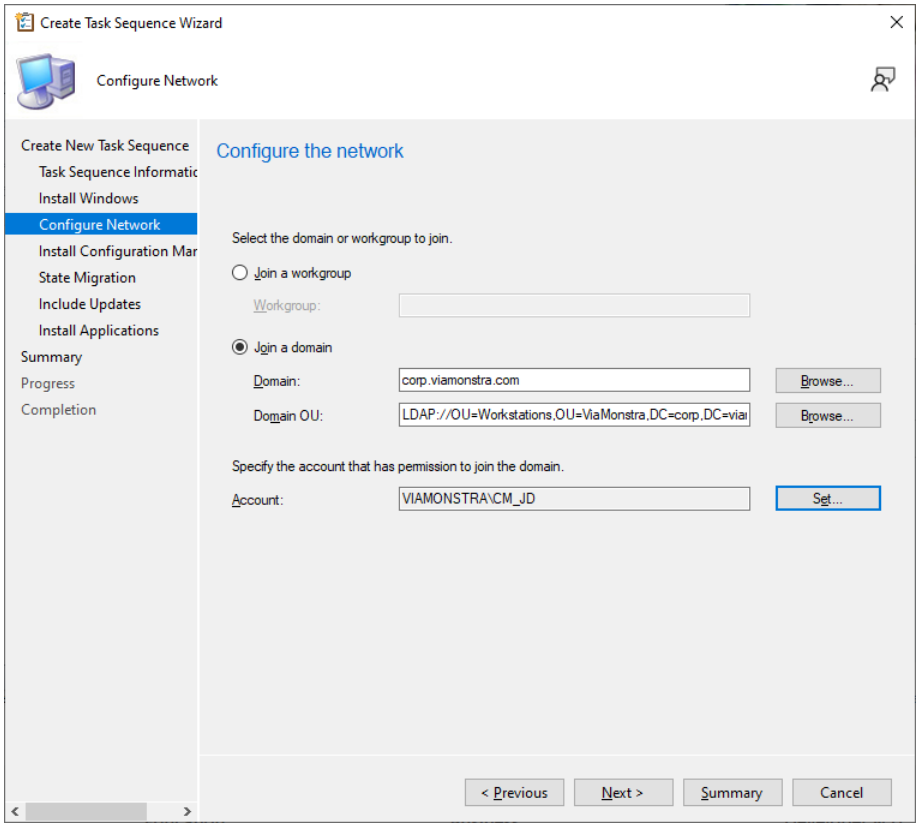
Configuring the Install Windows page.

Real World Note: Assigning a local administrator account/password is quite useful in a lab in the event you need to do some troubleshooting, but for production deployments its recommended having ConfigMgr disable it.

© Copyright 2022 Deployment Artist

d. Configure Network

- **Join a Domain**
- Domain: **corp.viamonstra.com**
- Domain OU: **LDAP://OU=Workstations,OU=ViaMonstra, DC=corp,DC=viamonstra,DC=com**
- Account: **VIAMONSTRA\CM_JD**
- Password: **P@ssw0rd**



Configure the network settings.

e. Install Configuration Manager Client

- Accept the default Configuration Manager Client Package

f. State Migration

- Clear all check boxes

g. Include Updates

- Select the **Do not install any software updates** option

© Copyright 2022 Deployment Artist

Real World Note: The ConfigMgr feature of installing software updates in a task sequence during deployment have been plagued with bugs over the years, and even when working, it adds extra time to the deployment too. I recommend having your image fully patched, and then use regular software updates in ConfigMgr to keep the client updated.

h. Install Applications

- Select any applications you want the task sequence to install, or skip for now. Can always be added later.

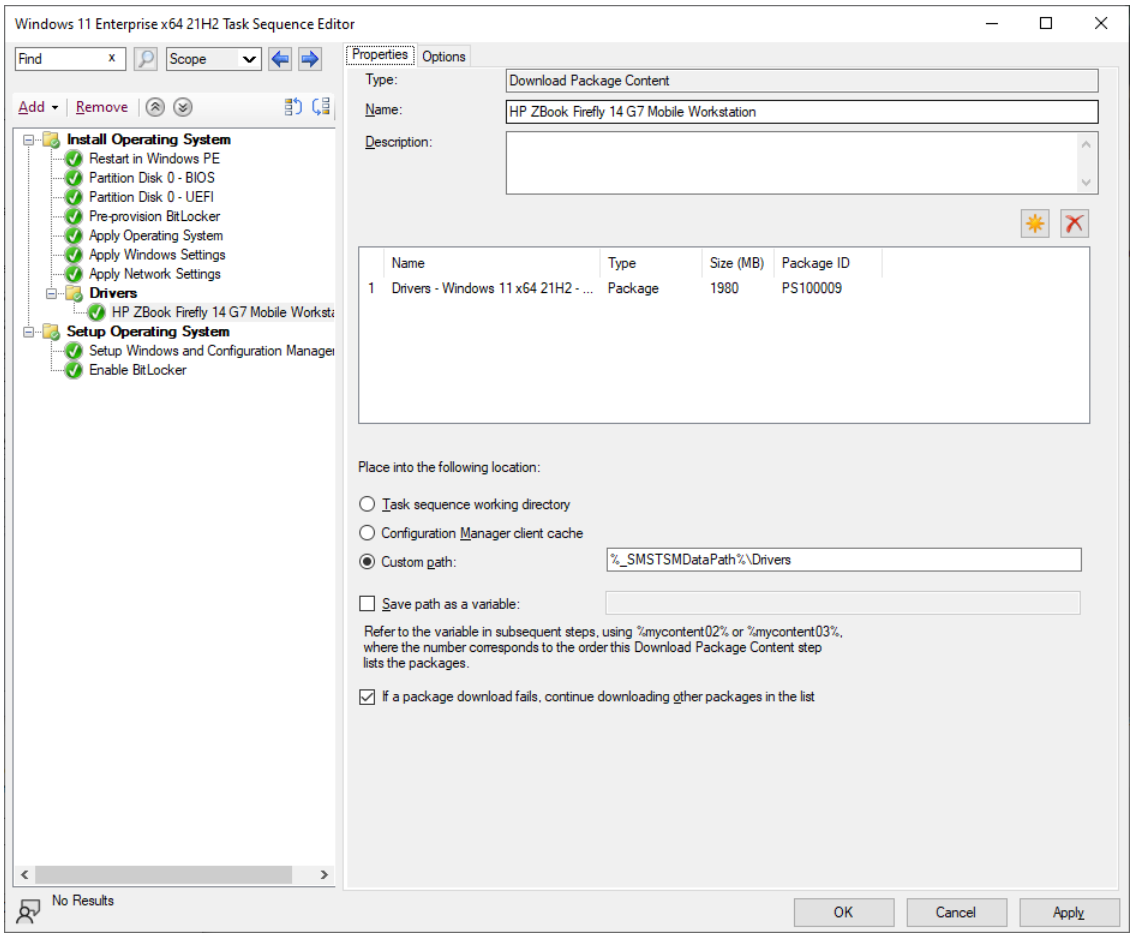
Edit the ConfigMgr Task Sequence

Once the ConfigMgr task sequence is created, you need to add drivers to it.

Real World Note: Again, while these steps will work just fine from a technically point of view, I highly recommend looking into using the modern driver management solution from Maurice Daly and Nickolaj Andersen. You can read more about that solution here: https://msendpointmgr.com/modern-driver-management/

1. Using the **Configuration Manager Console**, select **Task Sequences**, right-click the **Windows 11 Enterprise x64 21H2** task sequence and select **Edit**.
2. Remove the **Capture Files and Settings** group (not needed for bare metal deployments)
3. In the **Install Operating System** group, remove the **Auto Apply Drivers** action
4. After the disabled **Install Operating System / Apply Network Settings** action, add a new group named **Drivers**.
5. After the **Install Operating System / Drivers** group, add a **Download Package Content** action with the following settings:
    a. Name: **HP ZBook Firefly 14 G7 Mobile Workstation**
    b. Add package: **Drivers - Windows 11 x64 21H2 - HP ZBook Firefly 14 G7 Mobile Workstation**
    c. Custom path: **%_SMSTSMDataPath%\Drivers**

© Copyright 2022 Deployment Artist

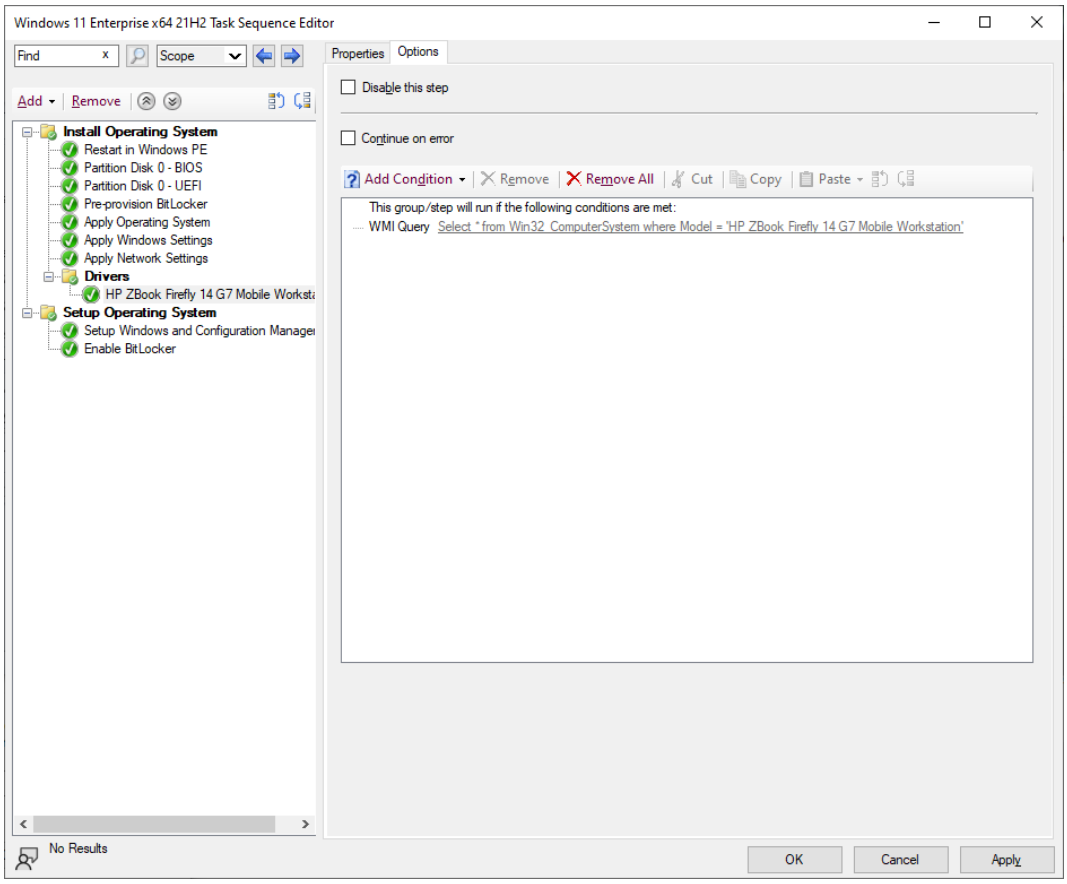


Adding drivers as standard packages.

d. In **Options**, add a **Query WMI c**ondition with the following WQL query:

- **Select * from Win32_ComputerSystem where Model = 'HP ZBook Firefly 14 G7 Mobile Workstation '**

© Copyright 2022 Deployment Artist



Setting the condition for the driver package.

6. After the **HP ZBook Firefly 14 G7 Mobile Workstation** Download Package Content action, add a Run Command Line with the following settings:

    a. Name: **Install Drivers via DISM**

    b. Command line: **DISM.exe /Image:%OSDTargetSystemDrive%\ /Add-Driver /Driver:%_SMSTSMDataPath%\Drivers\ /Recurse /logpath:%_SMSTSLogPath%\dism.log**

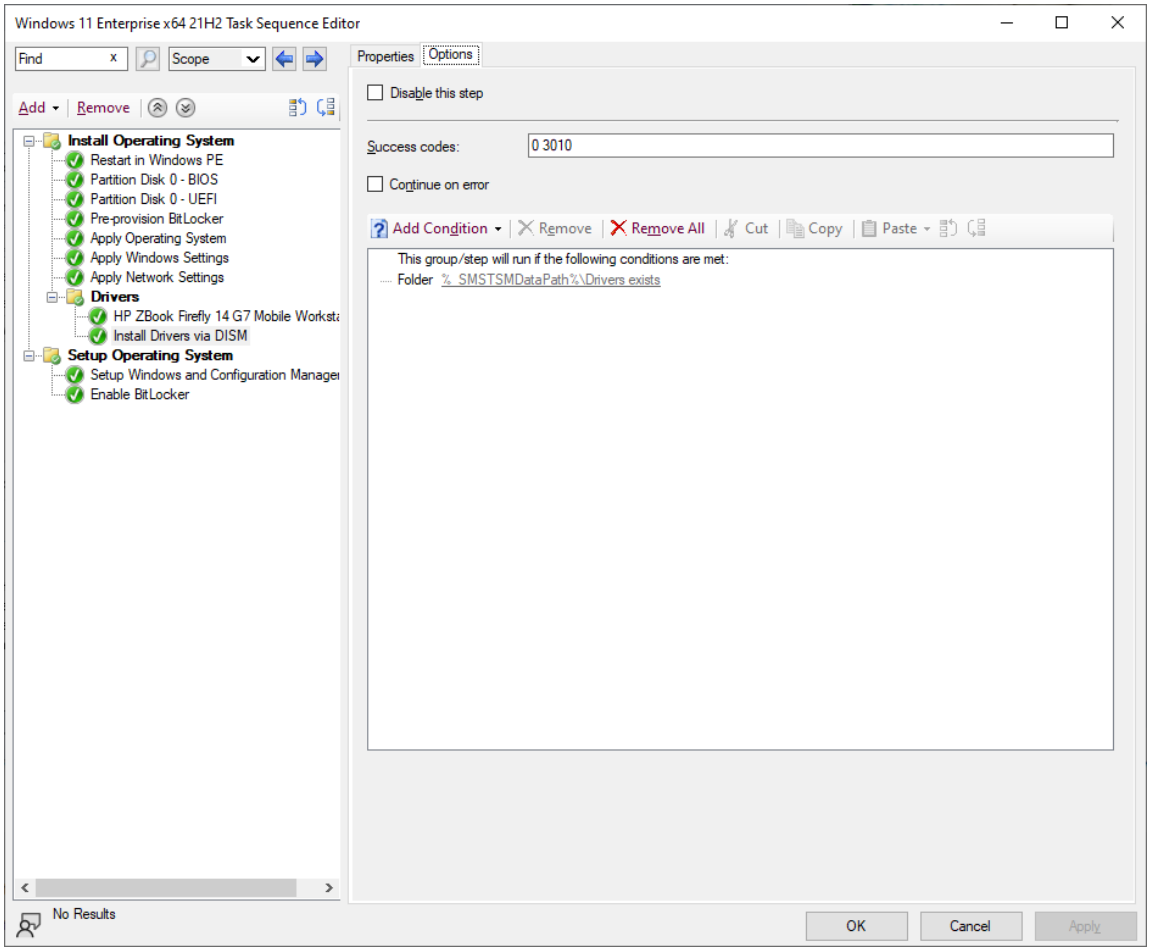
    c. In **Options**, add a **Folder Properties c**ondition with the following settings:

        i. **%_SMSTSMDataPath%\Drivers exists**

Note: Again, while this method of adding/installing driver packages works great from technical point of view, I highly recommend adding the community developed Modern Driver Management solution available on https://msendpointmgr.com/modern-driver-management/

When using the Modern Driver Management solution, you don't have to edit your task sequence(s) when adding support for additional models. Instead, a PowerShell script is called to figure out what package that is needed for a given model. The solution also adds support for archiving driver packages into WIM files which greatly reduces download time.

© Copyright 2022 Deployment Artist

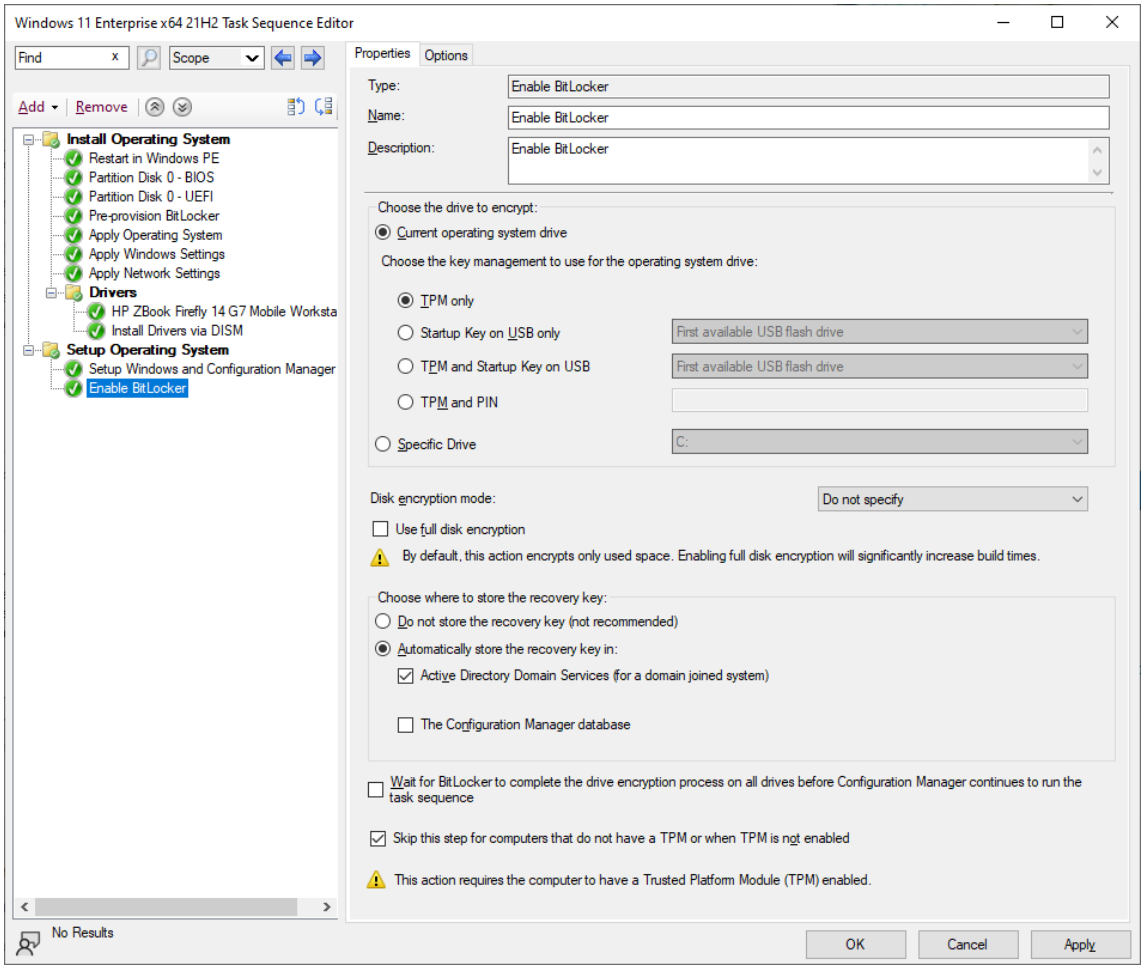


Setting the condition on the Install Drivers via DISM action.

7. Select the Enable BitLocker action, and select the **Skip this step for computers that do not have a TPM or when TPM** is not enabled checkbox.

Note: The preceding step is only needed in a lab with virtual machines where you may not have enabled all the VMs for a virtual TPM. That being said, both Hyper-V and VMware does support that you enabled a virtual TPM cryptoprocessor, so you can indeed test your BitLocker setting with VMs too.

© Copyright 2022 Deployment Artist



Enable BitLocker action configured.

Create a deployment for the Task Sequence

1. Using **Configuration Manager Console**, select **Task Sequences**, right-click **Windows 11 Enterprise x64 21H2**, and then select **Deploy**.

2. Use the following settings for the **Deploy Software Wizard**:

    a. General

        Collection: **All Unknown Computers**

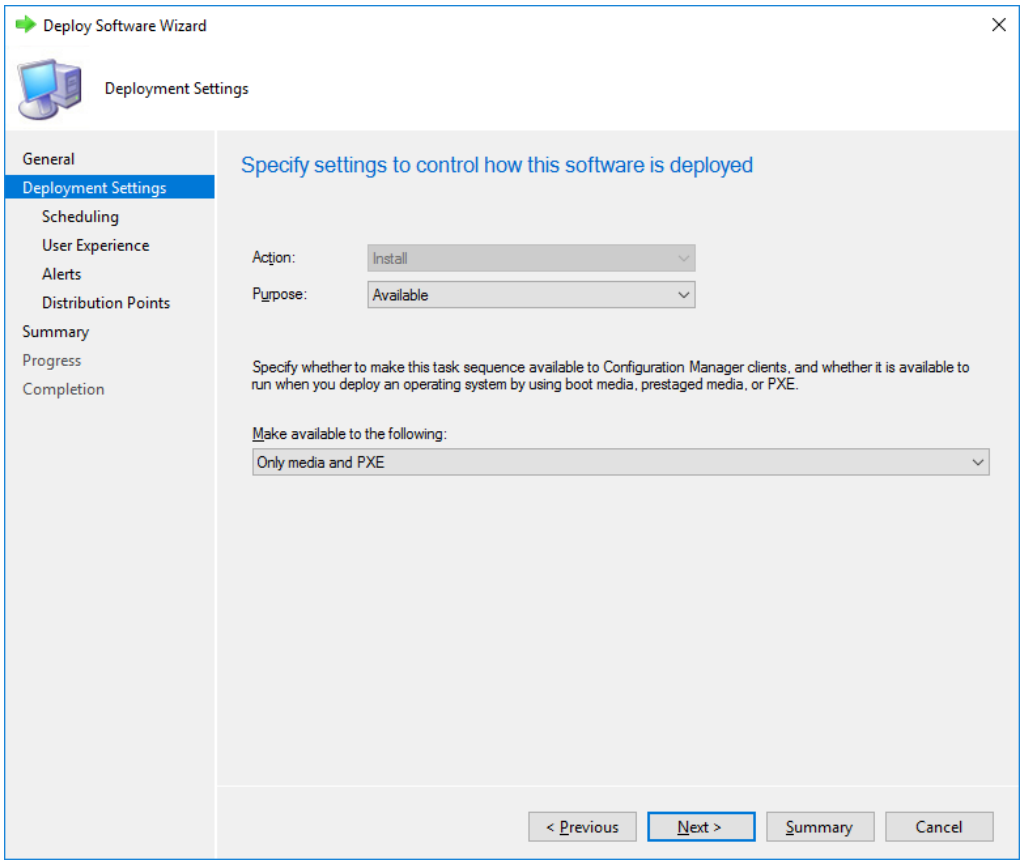
Note: Review the high risk deployment dialog box that is displayed when browsing for collections. This feature was added already back in ConfigMgr 2012 R2 SP1 and can be configured on the site properties.

    b. Deployment Settings

        - Purpose: **Available**
        - Make available to the following: **Only media and PXE**

© Copyright 2022 Deployment Artist

c. Use the default settings for the remaining wizard pages.



Configuring the deployment settings.

© Copyright 2022 Deployment Artist

Step 5 – Deploy Windows 11 using PXE

Deploy the Windows 11 Image

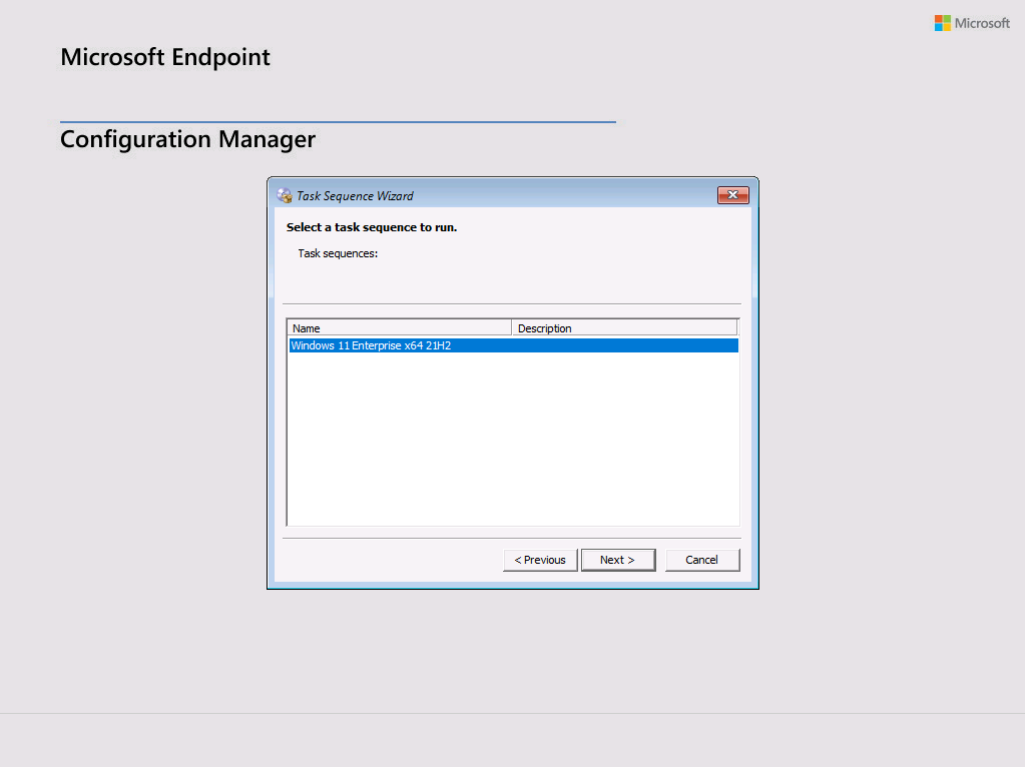
3. On your Hyper-V or VMware PC, create a virtual machine named **PC0002**, with 2 vCPUs, 4 GB RAM, UEFI type, and a 250 GB hard drive (dynamic type so the VM won't allocate the full 250 GB)

Real World Note: When testing bare metal deployments, I found it really useful to take a snapshot/checkpoint of the virtual machine before deploying it. That way you can easily revert back if something went wrong, fix the issue, and try again.

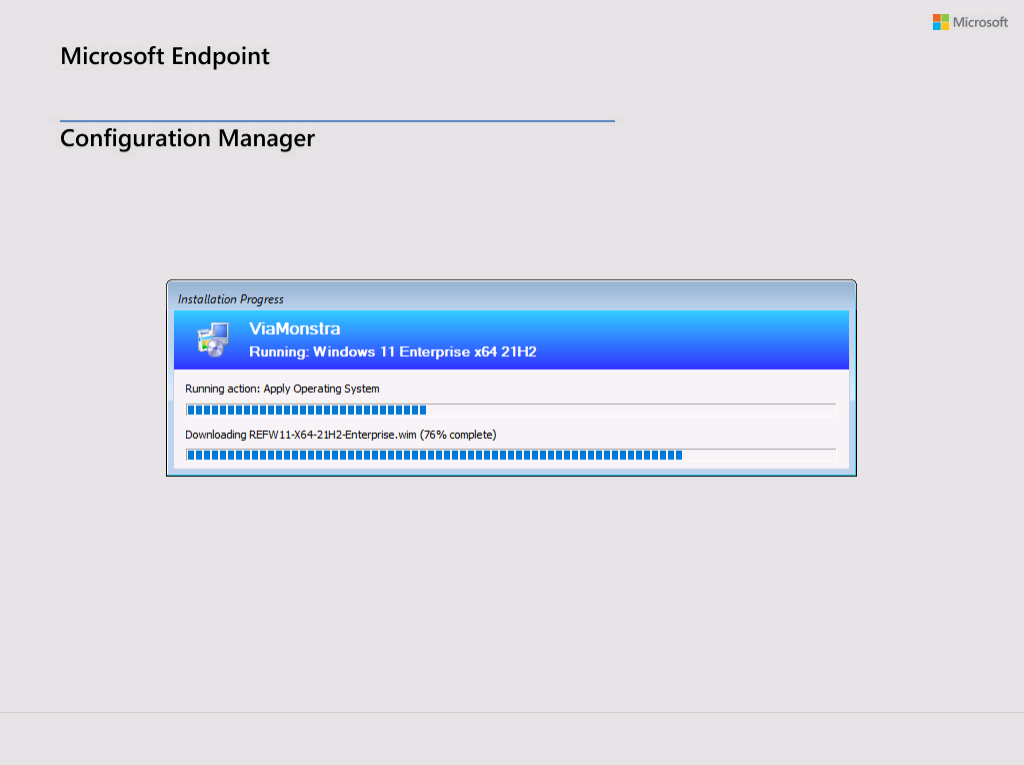
4. Start the **PC0002** virtual machine, and when the PXE menu is displayed, press **Enter** to PXE boot when prompted (you need to be quick).
5. After WinPE has been started, complete the deployment wizard using the following settings:
    a. Password: **P@ssw0rd**
    b. Select a task sequence to execute on this computer:
    Windows 11 Enterprise x64 21H2
6. The Task Sequence will now run and do the following:
    c. Install the Windows 11 operating system
    d. Inject the driver package
    e. Install the ConfigMgr Client
    f. Join the machine to the domain
7. After **PC0002** has been deployed, login and change the computer name to **PC0002**.

Note: Since you didn't add any computer naming options, or pre-staged the machine prior to deployment, ConfigMgr by default assigns a random MININT-XXXXXX type name to the machine.

© Copyright 2022 Deployment Artist



Selecting the Windows 11 task sequence while deploying PC0001.



Downloading the OS image.

© Copyright 2022 Deployment Artist

Beyond the eBook

If you liked this eBook, I bet you will like any of the events and trainings I'm part of.

Live Presentations

I frequently speak at tech conferences around the world, such as Midwest Management Summit (MMS), AppManagEvent, Nordic Infrastructure Conference (NIC), and Microsoft Ignite. For current tour dates and presentations, see my blog:

- https://deploymentresearch.com

Live Instructor-led Academy Courses

For Live Academy Courses, see the following site:

- https://academy.viamonstra.com

Live Instructor-led Masterclasses

For Live Masterclasses, see the following site:

- https://academy.viamonstra.com/pages/masterclasses

Twitter and LinkedIn

I would love to connect with you on Twitter and/or LinkedIn, so please reach out:

- Twitter: @jarwidmark
- LinkedIn: https://www.linkedin.com/in/jarwidmark

Facebook

And finally, yes, the Deployment Research blog have a Facebook page as well.

- https://www.facebook.com/deploymentresearch

© Copyright 2022 Deployment Artist