Microsoft Endpoint Manager Zero to Hero – Lab Guide

Volume 1

By Microsoft MVP's:

Peter Daalmans and Émile Cabot

PUBLISHED BY

MVPDays Publishing
http://www.mvpdays.com

ISBN: TBD

Warning and Disclaimer

Every effort has been made to make this manual as complete and as accurate as possible, but no warranty or fitness is implied. The information provided is on an “as is” basis. The authors 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 book.

Feedback Information

We’d like to hear from you! If you have any comments about how we could improve the quality of this book, please don’t hesitate to contact us by visiting www.checkyourlogs.net or sending an email to dave@mvpdays.com.

About the Authors

Émile Cabot – Microsoft MVP

Émile is a seven-time Microsoft Most Valuable Professional (MVP) who started in the industry during the mid-90s working at an ISP and designing web sites for celebrities. He has a strong background specializing in datacenter and deployment solutions, and has spent many years performing infrastructure analyses and solution implementations for organizations ranging from 20 to over 200,000 employees.

Émile organizes the Calgary Microsoft User Group, blogs on CheckYourLogs.net, and has presented at several conferences, including Ignite, VeeamOn, TechMentor, TechReady, and MVPDays.

In 2019, he tied for 1st place at the Microsoft Ignite Conference in Orlando with over 1,800 speakers in attendance.

BLOG: www.checkyourlogs.net

Twitter: @ECabot

Peter Daalmans – Microsoft MVP

Peter is a Principal Workplace Consultant / Architect and Microsoft Certified Trainer at Daalmans Consulting with a primary focus on the Microsoft Endpoint Manager Configuration Manager and Enterprise Mobility. Peter is awarded every year as a Microsoft Enterprise Mobility MVP (Configuration Manager/Microsoft Intune/Enterprise Mobility Suite) since 2012.

Peter writes blogs and sharing his knowledge on his blog ConfigMgrBlog.com. Peter is also one of the founders and leads of the Workplace Ninja User Group Netherlands.

Author of Mastering System Center 2012 Configuration Manager, Mastering System Center 2012 R2 Configuration Manager and the just in 2017 released Mastering System Center Configuration Manager, books from Sybex and a book called: Microsoft Enterprise Mobility Suite, Microsoft Enterprise Mobility Suite: Planning and Implementation. Nowadays Peter co-authors a book called Microsoft 365 Security for ITPros which is updated every month.

Peter speaks at local and international group meetings, conferences like Microsoft Ignite, Microsoft TechEd (Australia / New Zealand), IT/Dev Connections, Techmentor, Techorama Belgium, Midwest Management Summit (MMS), BriForum (London, Denver and Boston), TechDays Netherlands and ExpertsLive Netherlands. More about me here: https://ref.ms/aboutme

BLOG: http://www.ConfigMgrBlog.com

Twitter: @PDaalmans

Contents

Introduction

MVPDays Online

The purpose of this lab guide is to showcase the fantastic expertise of our guest speakers of MVPDays Online. They have so much passion, expertise, and expert knowledge that it only seemed fitting to write it down in a book.

MVPDays was founded by Cristal and Dave Kawula back in 2013. It started as a simple idea; "There's got to be a good way for Microsoft MVPs to reach the IT community and share their vast knowledge and experience in a fun and engaging way" I mean, what is the point in recognizing these bright and inspiring individuals, and not leveraging them to inspire the community that they are a part of.

We often get asked the question, "Who should attend MVPDays"?

Anyone that has an interest in technology is eager to learn and wants to meet other like-minded individuals. This Roadshow is not just for Microsoft MVP's, it is for anyone in the IT Community.

Make sure you check out the MVPDays website at www.mvpdays.com. You never know, maybe the roadshow will be coming to a city near you.

The goal of this particular lab guide is to show you how to setup and use Microsoft Endpoint Manager to manage the Windows 11 lifecycle for on-premises and cloud-connected workstations following a Work from Anywhere model.

Sample Files

All sample files for this book can be downloaded from www.checkyourlogs.net and https://github.com/PDaalmans/MEM-Win11-LabFiles/

Additional Resources

In addition to all the tips and tricks provided in this book, you can find extra resources like articles and video recordings on our blogs http://www.ConfigMgrBlog.com and http://www.checkyourlogs.net

Chapter 1 – Introduction to your MEM Lab

1.1 Downloading an electronic copy of the Lab Guide

Follow the steps below to download a copy of the lab guide from https://leanpub.com/memzerotoherolab

1.2 Logging into LabOnDemand

We will be using LabOnDemand by learnondemandsystems.com to provision all of the student lab environments for this Hands-On-Lab Experience. Follow the steps below to sign into your Lab machines.

1.3 Lab Server Names

The following table describes the required Virtual Machines built by this lab. This lab utilizes Microsoft Azure to host the resources in a multi-tenant configuration by LabOnDemand.

When launching your lab environment you will have your unique environment where you are able to perform the exercises.

Hostname	Role	Operating System
DC01	Primary Domain Controller running Active Directory Certificate Services as an Enterprise Root	Windows Server 2016
CM01	Configuration Manager Primary Site Server	Windows Server 2016
CL01	Admin Workstation	Windows 10
CL02	User Workstation	Windows 10
CL03	User Workstation	Windows 10

Lab Overview

Learn on Demand Systems has graciously provided their solution for configuring your lab environment. Their product, LabOnDemand, has drastically reduced the amount of effort involved in presenting Azure resources to multiple students. We invite you to visit their site and learn more about how they can remove effort and cost of managing your enterprise lab environments.

This lab has been built using a combination of cloud slices and VM templates to hydrate the environment. To eliminate multiple hours of base configuration, we have taken the liberty of configuring the core infrastructure and performing the initial OS installation and patch cycle. We've also automated the creation of an Active Directory Domain and an Azure tenant. This should replicate the starting point for most MEM admins.

We present you with a functioning domain and domain controller, with a member server, couple of workstations and some privileged accounts. We are using a Single Primary MEM-CM environment that has been already installed. The entire process was already automated for us thanks to some community-created scripts. Our first exercise will walk everyone through the steps we completed to prepare the environment, and what types of organizations this design is suitable for.

All the automation capabilities and tools you will learn during this lab are publicly available and regularly accepted for use in production.

Scenario

The creation of the lab environments were automated to ensure a consistent experience for all attendees, however it is important that your enterprise lab environment be created this way as well. Not only does it decrease the effort involved in its creation, but it allows you the freedom to easily create multiple lab environments in isolation. Furthermore, exporting configurations from your production environment will allow you to call those configurations during lab build, creating a more accurate representation of your production environment.

The following exercises will guide you through the initial requirements for building your own on-premises lab environment that can extend to a Test/Dev tenant in Azure.

Exercise 1 – Managing User Identity

In this exercise we will configure the correct UPN for AD so that the on-premises identities in the corp.viamonstra.com domain can be used to login to cloud resources.

Configuring the secondary UPN need to be done in two steps, first the domain name need to be added to the Active Directory and secondly the user objects need to be configured with the right domain name. Both can be done via PowerShell or via the tooling in Windows Server. In this exercise we will use PowerShell to configure the right domain name for Active Directory and the UPN of the users.

Ofcourse our Global Admin account(s) always need to be protected with MFA.

Instructions	Screenshot (if applicable)
1. First, download the ISO with the scripts from Github.	
2. Connect to the **DC01** VM and login with **VIAMONSTRA\Administrator**.	
3. On **DC01**, open the **Internet Explorer** and browse to https://github.com/PDaalmans/MEM-Win11-LabFiles/and download the file called **Student.iso**.	
4. Click **Open** to mount it as a drive.	
5. Copy all scripts to C:\LABS	

Before you are able to use the PowerShell scripts start the following command in **PowerShell** in **Administrative mode!**

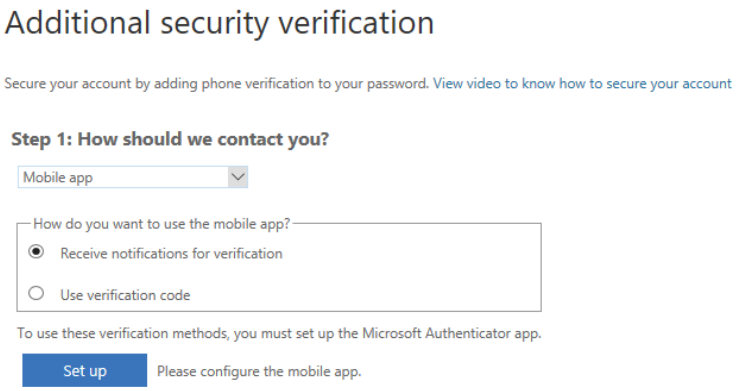
```
Set-ExecutionPolicy -ExecutionPolicy Unrestricted -Force
```

Protect Azure Admin with MFA

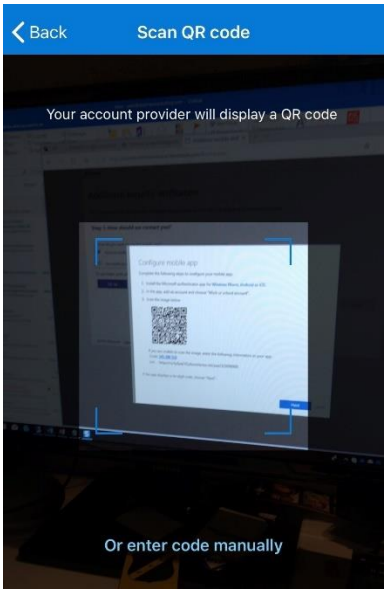
One of the most important steps is to protect your accounts by adding MFA to the authentication process.

Instructions	Screenshot (if applicable)
1. Connect to the **CL01** VM and login with **VIAMONSTRA\Administrator**.	
2. In a web browser on your PC, go to **https://aka.ms/MFASetup**	
3. Login with your Global Admin as the Username (eg, AzureAdmin@M365xXXXXXX.onmicrosoft.com) and the password and click **Next**.	

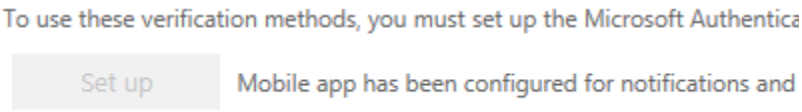
4. In Step 1: How should we contact you? select **Mobile App** and select **Receive notification for verification**.
5. Click **Set up**.



6. Download on your favorite mobile phone the **Microsoft Authenticator** app and open the app.
7. In the app, tap the **+** (plus) sign to add an account, and a **Work or school account**. Scan the QR code and click **Next** in the activation screen on your PC.



8. After the activation status is checked and the following message is visible ” Mobile app has been configured for notifications and verification codes”, click **Next**



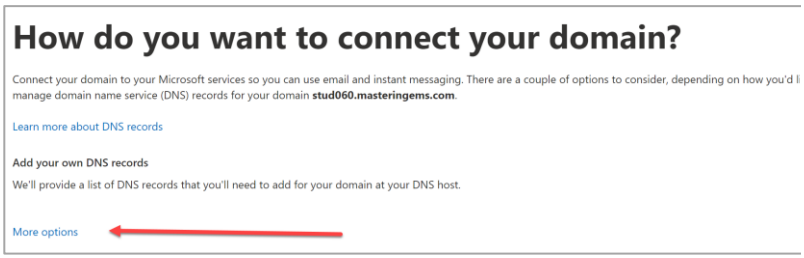
9. Approve the sign-in on your mobile phone for the user AzureAdmin@TMStudXXX.onmicrosoft.com. 10. Add optionally your phone number and click **Finished**.	

Complete Domain Setup

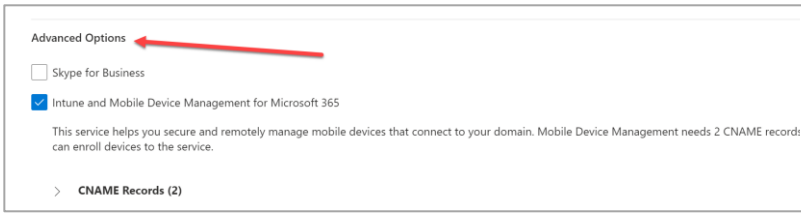
As part of the hands on lab we will be using our own custom domain name, we have prepared some of the steps but as part of this exercise the completion of adding a custom domain name need to be done.

Instructions	Screenshot (if applicable)
1. Connect to the **CL01** VM and login with **VIAMONSTRA\Administrator**.	
2. Start the Microsoft Edge and browse to **https://admin.microsoft.com** 3. Log in with your Azure Global Admin (AzureAdmin@tmstudXXX.onmicrosoft.com) and go to **Settings** and **Domains**. 4.	

5. Click **studXXX.dc.learning** to complete the setup of the custom domain.
6. Click **Start Setup** and click **Continue**, leave **Add a TXT record to the domain's DNS records** selected since this is already prepared for you**.**
7. Click **Verify** and click **More options**



8. Leave **Add your own DNS records** selected and click **Continue.**
9. Click **Advanced options**.
10. Select **Intune and Mobile Device Management for Microsoft 365** and click **Continue.**
11. Click **Done.**



Add Microsoft 365 licenses

As part of the provisioning Office 365 licenses are added to your tenant. To make sure that the Microsoft Endpoint Manager and Azure AD services can be used extra licenses need to be requested.

Instructions	Screenshot (if applicable)

1. Connect to the **CL01** VM and login with **VIAMONSTRA\Administrator**. Start Microsoft Edge and browse to **https://admin.microsoft.com.**
2. Log in with your Azure Global Admin (AzureAdmin@tmstudxxx.onmicrosoft.com) and go to **Billing** and **Purchase Services**.

3. Search for and click on **Microsoft 365 E5**, and choose **Get free trail**
4. Provide your phone number to prove you're not a robot and choose either **Text me** or **Call me**.
5. Provide the verification code and click **Start your free trail.**
6. Click **Try Now** and **Continue.**

Add secondary UPN to (local) Active Directory via PowerShell

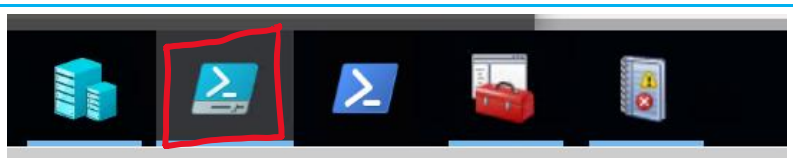
Instructions	Screenshot (if applicable)

7. Connect to the **DC01** VM and login with **VIAMONSTRA\Administrator**.	
8. On the taskbar, click **Windows PowerShell ISE** as **Administrator**	
9. Open **C:\LABS\1-AddUPN.ps1**	

10. Change XXX for the number you got from your instructor.	`1 Set-ExecutionPolicy -ExecutionPolicy unRestricted`
	`2 Get-ADForest | fl UPNSuffixes`
	`3 Get-ADForest | Set-ADForest -UPNSuffixes @{add="studXXX.dc.training"}`
	`4`

Add secondary UPN to all user accounts via PowerShell

To allow the users to use the new UPN, all user objects need to be changed to support the new domain name.

Instructions	Screenshot (if applicable)
11. Connect to the **DC01** VM and login with **VIAMONSTRA\Administrator**.	
12. On the taskbar, click **Windows PowerShell ISE**	

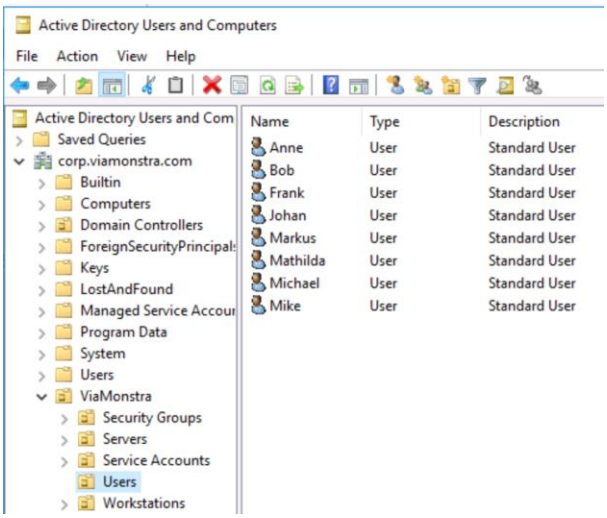
as **Administrator**

13. Open **C:\Labs\2-ChangeUPN.ps1**

14. Change XXX for the number you got from your instructor.

```
1  Import-Module ActiveDirectory
2
3  $newSuffix = "studXXX.dc.training"
4  $ou = "OU=Users,OU=ViaMonstra,DC=corp,DC=viamonstra,DC=com"
5
6  Get-ADUser -Filter * -SearchBase $ou -Properties userPrincipalName | foreach {
7  Set-ADUser $_ -UserPrincipalName "$($_.samaccountname)@$newSuffix"}
```

15. Run the script.
16. Still in DC01, open **Active Directory Users and Computers** and check the UPN of your users under corp.viamonstra.com > Viamonstra > Users
17. Open the properties of a user and review the **Account tab** of the user objects.



Exercise 2 – Reviewing the AD Configuration

As you can see in the previous exercise,

In this exercise, we will set up the on-premises accounts we need to complete the labs. To ensure the account creation process is quick, repeatable, and consistent, we're using scripts to do this instead of dsa.msc.

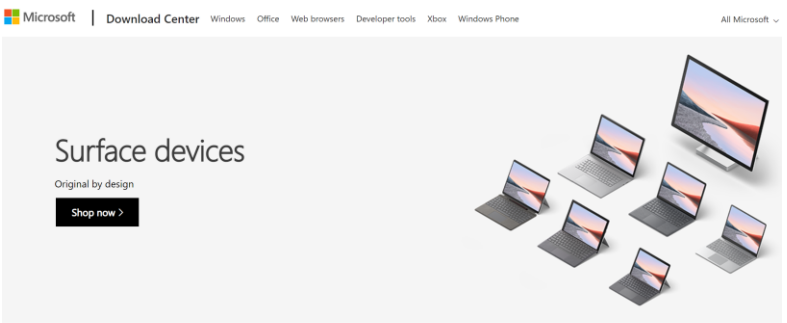
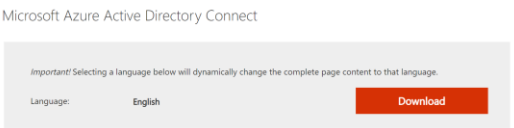
IMPORTANT NOTE: ***THESE COMMANDS HAVE ALREADY EXECUTED IN YOUR DOMAIN. YOU DO NOT NEED TO OPEN or DO ANYTHING FOR THIS EXERCISE. JUST REVIEW THE EXERCISE 2 IN THIS LABGUIDE. DO NOT PERFORM ANY OF THE STEPS BELOW. WE ARE SHOWING THEM JUST TO SHOW HOW EXERCISE 2 WAS SETUP.***

Instructions	Screenshot (if applicable)
1. Logon to the Lab Host as Administrator	
2. On the taskbar, click **Windows PowerShell ISE** as **Administrator**	
3. Open **C:\LABS\Lab1-Exercise1-Setup.PS1**	

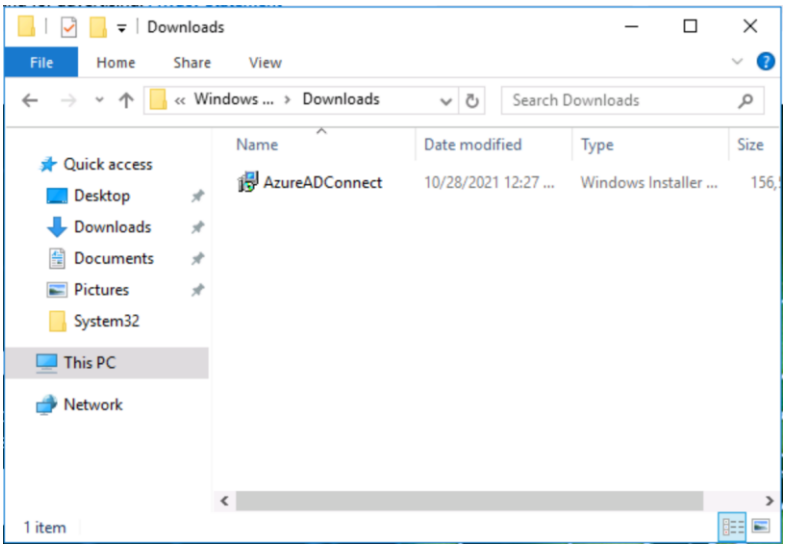
Exercise 3 – Connecting Active Directory to Azure

In this exercise, you will install and configure Azure Active Directory Connect, the tool that synchronizes the on-premises and Azure copies of Active Directory. This is also where we

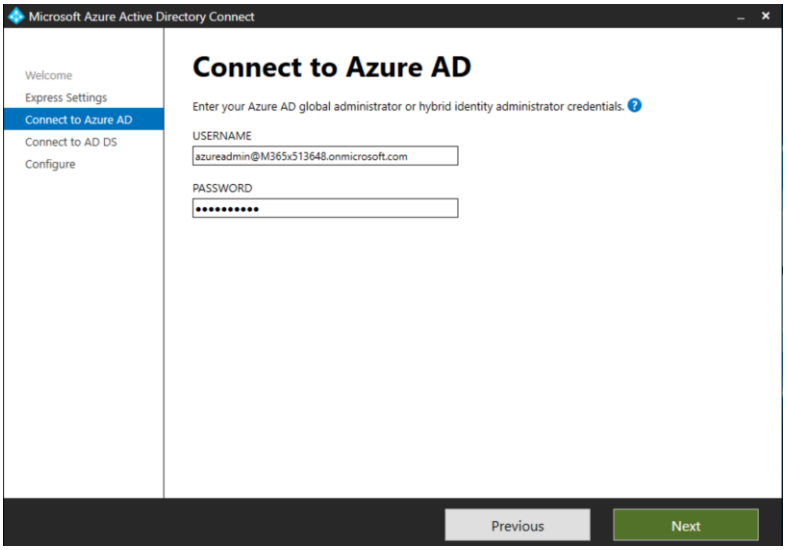
configure features like Single Sign-On, Workstation Hybrid-Join, and enable users to reset their AD passwords from office.com. Since we will be using this tool a couple of times in a row, please make sure not to directly start synchronizing the identities, since by default all identities and groups will be synced. We want to be selective.

Instructions	Screenshot (if applicable)
1. Logon to the DC01 as Administrator	
2. Open the Internet Explorer and go to **https://ref.ms/aadconnect** and click **download** to get the latest version of AADConnect. Save the file to C:\Downloads\	 

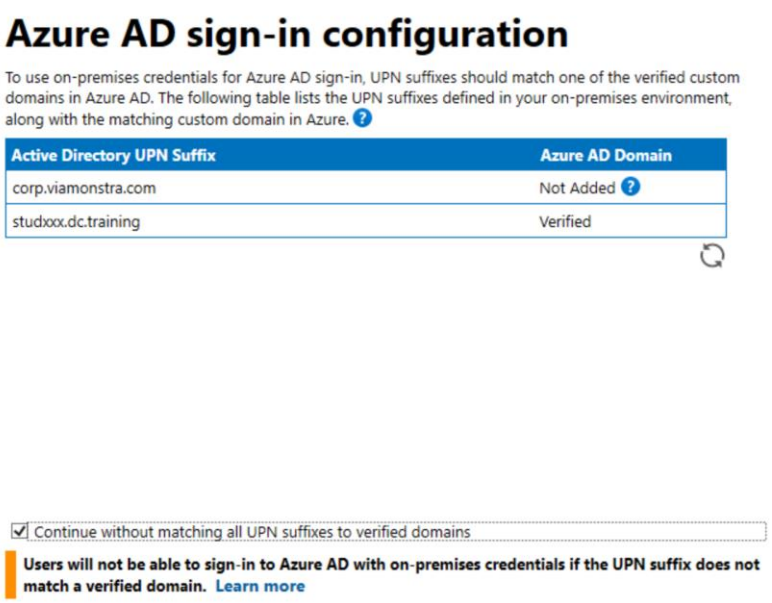
3. Go to C:\Downloads and double click **AzureADConnect.msi** and click **Yes** twice, to confirm that this app to make changes to your device in the User Account Control dialogs.
4. Agree to the license terms and privacy notice by checking the **checkbox**. Click **Continue**.



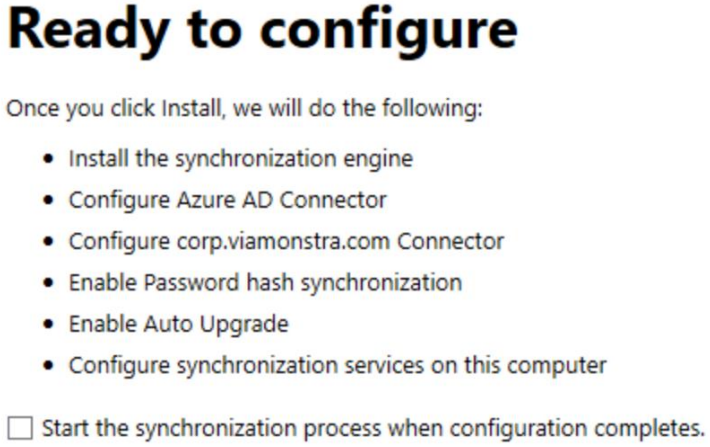
5. Click **Use express settings.**
6. Supply the Global Admin as the Username (eg, AzureAdmin@TMStudXXX.onmicrosoft.com) and the password and click **Next**.



7. Supply the **Domain Administrator** as the **Username** (eg VIAMONSTRA\Administrator) and the password and click **Next**.
8. Review that the custom domain name is verified.
9. Check **Continue without matching all UPN suffixes to verified domains**
10. Click **Next.**



11. **Uncheck** Start the synchronization process when configuring completes.
12. Click **Install.**

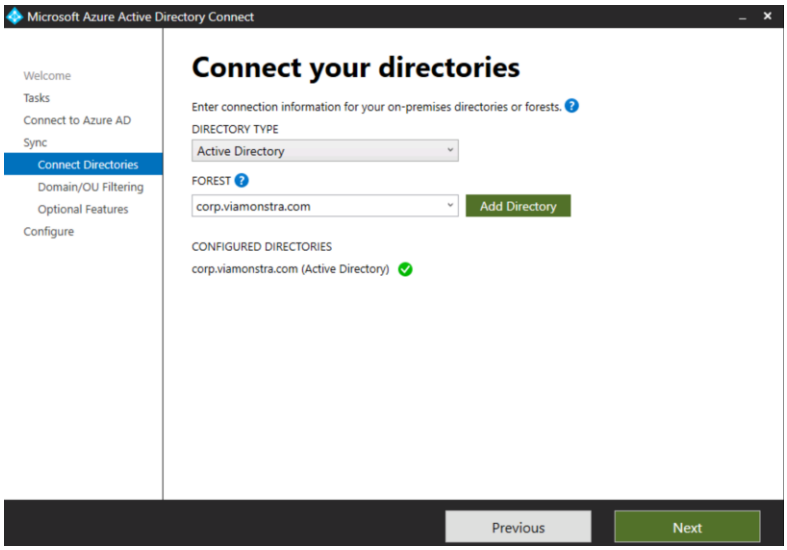


13. Review the results and click **Exit**.

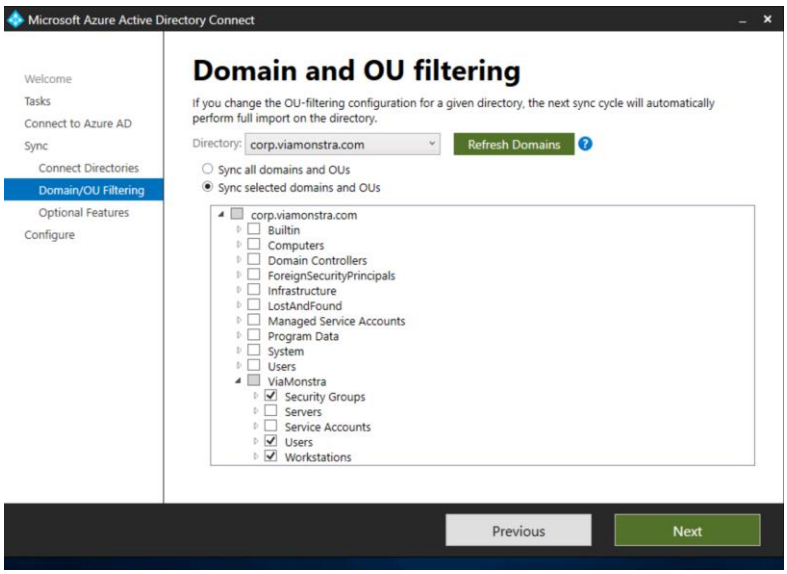
14. Still on **DC01**, double click **Azure AD Connect** located on the Desktop**.**

15. Click **Configure**, and select **Customize synchronization options**. Click **Next.**

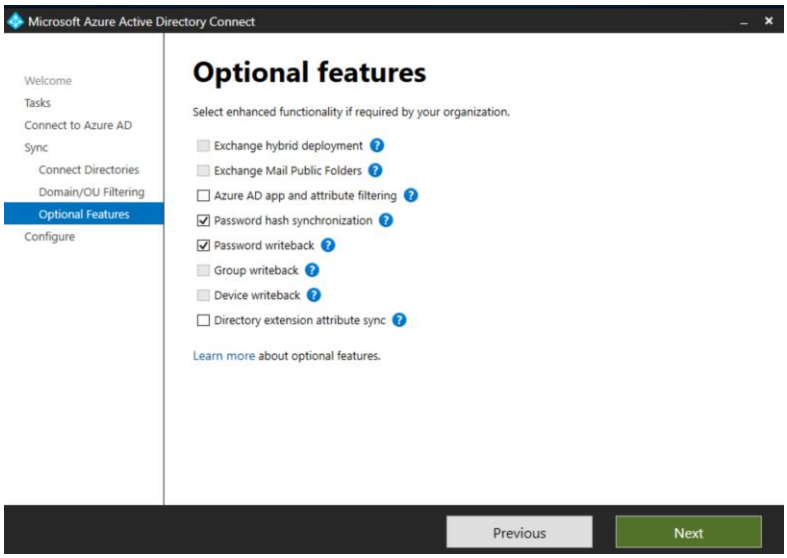
16. Supply the Global Admin as the Username (eg, AzureAdmin@TMStudXXX.onmicrosoft.com) and the password and click **Next**.
17. In the Connect your directories screen, click **Next**.



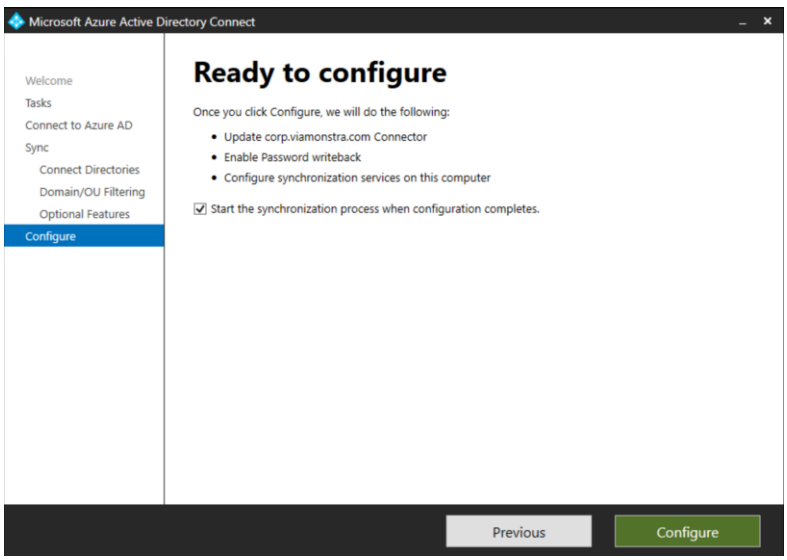
18. In the Domain and OU filtering Screen, select **Sync selected domains and OUs** and deselect **corp.viamonstra.com**
19. Expand **corp.viamonstra.com, viamonstra,** and select **security groups**, **Users** and **Workstations**.
20. Click **Next**



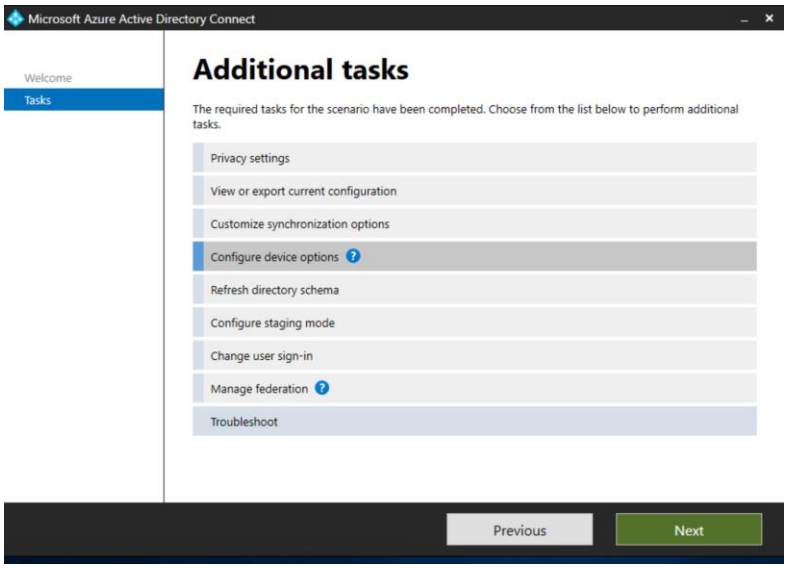
21. Make sure that both **Password hash synchronization** and **Password writeback** are selected and click **Next**.



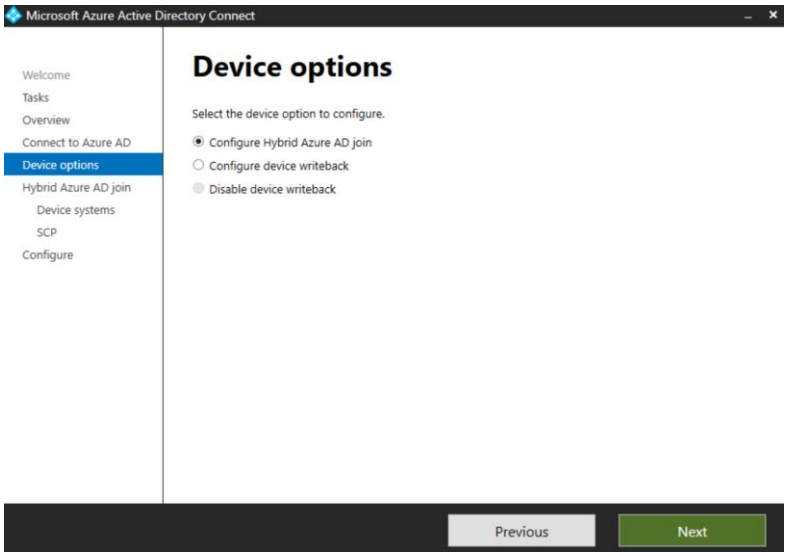
22. **Enable** Start the synchronization process when configuration completes.
23. Click **Configure** and click **Exit** when the configuration is finished.



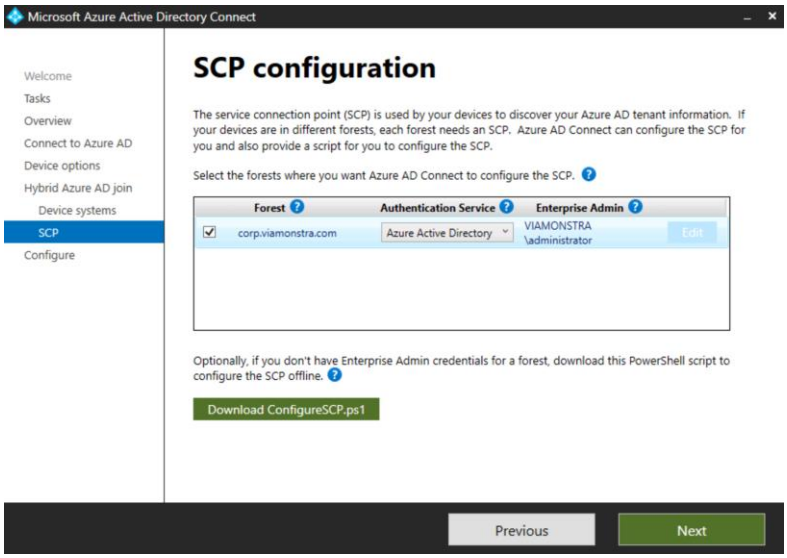
24. The last feature we want to configure is the ability to Hybrid Azure AD join the Windows 10 devices. Still on **DC01**, double click **Azure AD Connect** located on the Desktop**.**
25. Click **Configure**, and select **Customize synchronization options**. Click **Next.**
26. Select **Configure device options** and click **Next twice.**



27. Supply the Global Admin as the Username (eg, AzureAdmin@TMStudXXX.onmicrosoft.com) and the password and click **Next**.
28. Leave **Configure Hybrid Azure AD join** selected and click **Next**.



29. Select **Windows 10 or later domain-joined devices** and click **Next**.
30. Select **corp.viamonstra.com** and change the Authentication Service to **Azure Active Directory** click **Add** to add the VIAMONSTRA\Administrator account.
31. Click **Next**.
32. Click **Configure**.
33. Click **Exit.**



Chapter 2 – Building the right Solution Design

Scenario

Microsoft provides sizing information for Microsoft Endpoint Manager Servers and Site Roles that allow you to properly scale the configuration for your environment. Maximum thresholds for various configurations are well defined and there is a clear outline for the physical placement of each role.

Unfortunately, the information Microsoft provides on sizing and performance guidelines are only listed for 25k-700k client environments, and do not provide help on exact requirements for anyone smaller. Subsequently, infrastructure requirements for various features, such as Operating System Deployment, have the potential to exponentially increase the implementation, network and management costs of the solution, even for a company with fewer than 5,000 clients. This failure to properly scale down requires a significantly higher capital expenditure for initial deployment, or investigate alternatives to address the deficiency.

While this may seem like an oversight, it is more due to how Microsoft Endpoint Manager is designed and the architectural limitations experienced with increasing client counts.

Based on these limitations, we can loosely categorize everyone in one of four groups: Group A is everyone that has less than 25,000 Windows devices, Group B is between 25,000 and 49,999, Group C organizations have between 50,000 and 150,000, and Group D enterprises have more than 150k Windows endpoints.

Topic 1 – Physical Hardware

No organization today, of any size, should be installing production workloads directly on physical servers. It increases the total cost of ownership of the workload by complicating the backup, restoration, configuration and availability of the operating system that supports the workload.

Virtual machines are portable, easily backed up and modified to increase/upgrade virtual hardware during the operational lifecycle of the product. Any resources lost to the hypervisor are easily compensated by the ease of management of the production workload. Furthermore, the hardware infrastructure hosting the virtual machine can be configured in highly-available designs without complicating the product's installation.

Azure? Of course. You can fully deploy Configuration Manager in your Azure environment and avoid your datacenter entirely. Outside of the monthly VM costs, you must also consider data egress, which is about 10 cents a GB.

In almost all circumstances, resources for your Configuration Manager infrastructure will be provisioned from an existing cluster.

Microsoft provides minimum hardare requirement guidelines for specific numbers of managed clients, but not always for >25k clients. It's also somewhat hard to decipher, as the hardware requirements are based on roles that are usually combined on the same virtual machine. After we've figured out what's gonna go where, we'll decide how much resources each VM will need.

Processor Cores and Memory

Configuration Manager runs many simultaneous processes, so needs a certain minimum number of CPU cores for various site sizes. These minimums are what Microsoft has listed on their requirements page. At some point, I think it was 1607, the core requirements for the primary site server doubled from previous versions. Since then, I have shied away from reducing CPU allocation on site servers.

While more cores can be presented to the VM than recommended, there are some very important points to be considered first:

1. If CPU is the performance bottleneck, moving to the same number of faster cores will provide significantly improvement than adding additional slower cores.
2. Configuration Manager threads utilize parallel processing and need to be incremented properly.
3. The relationship between cores and memory is important. You should have at least 3GB of RAM per core to prevent SQL performance issues.
4. These are virtual cores that are being allocated from physical processors on the hypervisor. Above all else, you must properly match the vCPU cores/processors with what is being presented by the host and be careful not to over-subscribe.

With RAM, more is almost always better. We want to ensure that the amount of allocated RAM does not over subscribe the host, and properly allocate memory to SQL on the database server. Microsoft (as of October, 2021) recommends 96GB RAM for a single server installation. For an

organization with 500 people this number seems inflated, and it is. This is based on 25,000 endpoints, and scaling down is up to us.

The Primary Site Server we use for this lab is configured with 16GB, and that is as small as you should ever go. Production environments with less than 300 endpoints and light use can get away with 24GB, and to provide basic support to 500 users you'd need at least 32GB.

In Hyper-V, there is an option to dynamically adjust memory allocation based on resource demand. This option should never be used in production, as it can lead to over-allocation of memory resources on the host and also uses CPU resources to calculate the memory required. This additional CPU requirement is not factored in sizing considerations when planning the design and difficult to calculate.

Let's look at how we'd spec out the site servers for our 4 groups. In the following table we will outline the processor requirements for each classification group. To recap:
Group A – > 25,000 clients
Group B – 25,001 – 49,999
Group C – 50,000 – 149,999
Group D – 150,000+ clients

Group	Site Server	Cores and Memory
A	Primary, SQL, SUP, MP, DP, SMP, RP, EP, EPP	16/96
B	Primary, SQL, MP, DP, EP, EPP	16/96
B	SUP, RP	12/48
B	MP, DP	6/18
C	Primary, SQL, MP, DP, EP, EPP	16/96
C	SUP, RP	16/72
C	MP, DP, SMP	6/18
C	MP, DP, SMP	6/18
D	CAS, SQL	20+/128+

D	Primary, SQL	16/96
D	Primary, SQL	16/96
D	MP, DP, SMP	6/18
D	SUP, RP	16+/96+

Disks, Databases, and IOPS

Disk performance is arguably the most overlooked component when implementing Configuration Manager, and it is the most important. Not only is SQL a hog for input/output operations, but any old-school SCCM guru will tell you that Configuration Manager relies heavily on scheduled tasks and file system operations to function. You need to pay attention to more than just the SQL temp and log file locations to ensure your CM environment will continue to purr when under load.

When consolidating roles on a single virtual server, and even across multiple VMs on a single host, you need to try and prevent simultaneous reads and writes occurring on the same drive. Even for all-flash arrays, this is still important for most sizes.

When we initially build our production Primary, we want to validate the disk performance before we start filling it up with stuff. The free way to do this is to use Diskspd. Diskspd is an executable provided by Microsoft that will monitor the IOPS on a specific volume.

Topic 2 – Designing the Hierarchy

Earlier, we classified all organizations into 4 groups. In the datacenter, companies of the same group should exhibit a similar ConfigMgr design. Once the entire network is considered, however, we will see a large disparity in the count and placement of site servers that directly communicate with endpoints, namely Management and Distribution points. More on those in a bit.

SQL

The SQL instance for Configuration Manager is considered high-transaction and drives/databases must be configured accordingly. Even with all flash storage arrays, spreading databases across multiple drives will often benefit operational performance.

A very handy database sizing calculator was created for SCCM 2012 by MVP Kent Agerlund, and has since been updated by Anthony Clendenen and Steve Thompson to align with 2016+. It can be downloaded from here or found in the lab source files.

More than 25,000 Windows devices?

WSUS, as mentioned earlier, needs to be split off from the Primary Site Server once we go above 25,000 clients. WSUS is a bit of an IIS hog, and pulling down virtually anything more than definition files will cause it to crash during any sync job. Though documented as a troubleshooting step and not in the core design, even the smallest of environments will need to tweak their IIS configuration and AppPools to allow the process to work as intended.

At 25k, the Management Point role will also reach its upper limit, requiring at least one more.

Forgetting Distribution Points for a while longer, this is the first time we're *required* to move a role away from the Primary, and it's the first time we even consider decentralizing roles. CM is very chatty, especially between the Primary, Management Points, and SQL, and keeping these roles as close together as possible is **always** the goal.

This rule, providing adequate hardware is provided to the Primary, can almost be applied across the board, regardless of operational use; perhaps the only situation that requires deviation is an organization that has 10+ people running multiple reports daily, which typically doesn't start to happen until the product has been fully integrated into IT operations. Around this point, the typical design will start to see performance degradation in the SQL server, and would require adding ram to the primary and additional disk to assign to the reporting instance. Offloading the Reporting Services Point role to another SQL server, perhaps a production cluster, would likely be more cost effective than adding Flash. As the RSP does not participate in CM operations, decentralizing it does not impact performance of the overall solution.

Fallback Status Point

Over the years, I've only ever seen the Fallback Status Point role implemented properly three times. This is a special role that is only used during client installation, as a place to transmit status and logs when the installation cannot connect to the Management Point. If you place the FSP on your Primary, and a client can't connect to the Primary…it won't connect the FSP either.

Proper placement of the FSP would see it in a segmented network location, such as a DMZ, or in a different forest on a multi-forest environment. It should never be co-located on a Site Server and expected to be of any use.

Central Administration Site

When the redesign of Configuration Manager was released in 2012, it included a documented client-size limitation of 100,000 endpoints to a Single Primary environment, after which point you'd need a new server called the Central Administration Site.

Arguably the most impactful decision of SCCM 2012 was the inability to implement a CAS at a later date, which left everyone near 100,000 clients, or hopelessly optimistic SMBs, with having to deploy a CAS type hierarchy out of the gate.

This was not intended, and created a substantial delay in daily operations. Shortly thereafter, Microsoft introduced the ability to deploy a CAS post-installation, and admins started lining up to rearchitect their environment.

Over the past ten years, Microsoft has continued to optimize the solution to maximize advancements in server hardware, and Primary Site Servers can now support up to 175,000 endpoints across Windows and MacOS operating systems without the need of a Central Administration Site.

Until you exceed that 175,000 client threshold (or 150,000 Windows devices), avoid including a CAS in the design.

Distribution Points

From a capital expenditure perspective on a new implementation, and yearly operational support of the infrastructure, Distribution Points are by far the most costly. Factors that contribute to this are hardware/VM allocation costs, OS licensing and server management, as well as the staggering amount of time admins have to spend managing content across dozens, hundreds, or even thousands of distribution points. Subsequently, as most DPs in a Microsoft design are remote, traffic between the DP and the Primary is not optimized. Often deployed over slow links, these connections need to be managed and shaped by the network team to prevent operational impact.

Looking at the current size and scale numbers, each distribution point can support connections for up to 4,000 endpoints. While this level of support may be scalable and reasonable from a server requirements perspective, the bigger issue is introduced with placement: A Distribution Point needs to be placed at every corporate location where workstations reside. With advancements in WAN technology, highly connected-high bandwidth locations may not require a local DP and will rely on a centralized DP instead. Taking a "try and see approach" to DP requirements, deploying the Office365 client or a dozen simultaneous Operating System deployments to a branch location would quickly tell you where your network will fall down. 😊

Topic 3 – 3rd Party Infrastructure Support

Looking at the hierarchical design, there is one glaring problem when we start planning out the location of all required site servers, and that focuses on Content Distribution. Most visibly when we're performing Operating System Deployments, clients need a good connection to the MP and DP to ensure communications and operations are successful. Geo-dispersed organizations, regardless of size, often find this requirement exponentially increases the overall cost of the solution, driving admins to seek a better alternative.

1E Nomad

1E has been providing augmented technologies for Configuration Manager since 1997. Nomad is a self-managing content distribution solution for Microsoft Endpoint Management. It removes the complexity and management overhead of three complicated technologies while ensuring a completely healthy endpoint estate. Nomad's proven peer-to-peer technology intelligently uses only spare bandwidth for all IT content distribution, so your employees' work is never impacted by MEMCM or Windows deployments.

With Nomad's Single Site Download feature, we're able to facilitate up to 4,000 sites with each distribution point and no longer require server hardware to be present in the branch infrastructure. This reduces the initial deployment cost of the solution, eliminates operational management of remote servers, no longer requires networking to shape ConfigMgr traffic, and above all ensures that content downloads never impact end user productivity at a location.

Topic 4 – 3rd Party Patching

Around 30 days after getting all your clients on Configuration Manager, you'll start looking into reports to see how the initial patch cycle completed. You'll notice pretty quickly that the only items that are being patched is what comes from Microsoft Update…any patches that were released for Adobe, for example, aren't included. If you want to include the rest of the patches that must be deployed, another product is required.

Microsoft provides a solution to enable 3rd party patching via Configuration Manager that's called System Center Updates Publisher. It's been around since Configuration Manager was a System Center product, is 100% free, and you get exactly what you pay for. It is cumbersome to configure and maintain, and we despise using it, though occasionally still rears its ugly head in K-12 education.

Since the de facto product is crap, we once again look to the industry for solutions, and there are several that have been created to fill this gap. These products can be loosely categorized into two groups, those that deploy patches through Configuration Manager, and those that handle their own content delivery.

For the purposes of this exercise, we will consider products that utilize ConfigMgr software distribution. The two products that we see most commonly are Ivanti Patch for MEM and PatchMyPC.

Ivanti Patch for MEM

Ivanti, formerly LANDesk, is a competing systems management product to Configuration Manager. Their patching solution is intuitive, and thankfully they've given us the opportunity to leverage it in Configuration Manager, through their tool Ivanti Patch for MEM.

Ivanti Patch integrates into the MEM console and provides an extensive regularly-updated catalog of 3rd party patches. You can download required patches and deploy them to Collections and Intune directly from the plugin's UI. It supports patch customization and full workflow automation.

PatchMyPC

PatchMyPC has been around for a number of years, and is the other tool we commonly see these days. There are two flavours, an automatic publishing service and the catalog only subscription. The latter is the original method and integrates with Updates Publisher to make it actually usable in a production environment. The solution provides an extensive update catalog where Updates Publisher can acquire the patch content and a plug-in to manage the extended functionality. Since it connects at the Updates Publisher level, software manufacturers and the available patches are listed in the native locations in the console, when configuring the Software Update Point and when looking at the All Software Updates node.

Topic 5 – 3rd Party Endpoint Detection and Remediation

Configuration Manager is database driven, and operations are executed based on the information contained in the database. Furthermore, it is a pull-based solution that relies on the endpoint agent to regularly "check in" to communicate status and check for new downloads.

This allows the solution to perform remarkably at scale and provides the administrator with a wealth of information and options for managing the device environment.

What this does not do, however, is give us the ability to take immediate action on a large number of devices or obtain real-time data about the connected environment. From a reporting perspective, this leads to inconsistent and stagnant data that requires hours' worth of effort to validate. In a modern world of zero day threats, we need the ability to quickly detect new vulnerabilities on our connected devices, remediate them before an intrusion is attempted, and isolate devices that have been impacted.

This very important topic is the basis behind our *Enforcing Compliant Configurations* book. The book is free and can be downloaded from https://leanpub.com/enforcingcompliantconfigurations.

Chapter 3 – Automating the Configuration

Scenario

We've designed the environment to adhere to Microsoft recommendations and maximized our use of the modern datacenter, submitted the designs through Change Control and have our VMs provisioned. Like the AD setup for our lab environment, we want to script the installation process for Configuration Manager as much as possible for production. This will ensure the installation documentation is 100% as-built, and can be repeated in lab environments to provide safe areas to test.

Large enterprises will often have 3 different testing labs a product must certify through before approval for production use, and consistent configurations through those environments ensure a smooth progression through Test, Development, Staging, and Production.

Topic 1 – ConfigMgr Prerequisites Tool

While some server teams will go the extra step and provide some additional configuration when provisioning a server, for the sake of instruction we will expect to receive a fully patched VM with Windows Server 2019 Datacenter installed.

Throughout these exercises, we will be using some free tools provided to the community by our MVP colleagues over at MSEndpointMgr.com. They're PowerShell driven, regularly updated, and widely accepted in the enterprise. We continually thank them for their contributions and the hours of time their tools have saved us on projects.

The ConfigMgr Prerequisites Tool is designed to help administrators prepare their infrastructure and systems when about to install System Center Configuration Manager.

Preparing your environment for a successful deployment of Configuration Manager should not be an obstacle or a time consuming task. Many of the issues that might occur during the initial installation process can often be resolved by making sure that the correct prerequisites are in place.

The ConfigMgr Prerequisites Tool will help you to successfully prepare your environment allowing you to install the required software and Windows features for the Central Administration Site, Primary Sites, and Secondary Sites. It supports prerequisites for the Management Point, Distribution Point, Software Update Point, State Migration Point, Application Catalog, Enrollment Point, Enrollment Proxy Point, Certificate Registration Point.

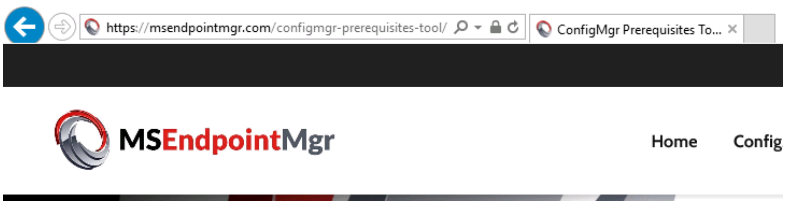
In addition to what's mentioned above, the tool will also allow you to perform:

- Configuration Manager configuration:
  - Download prerequisite files for Configuration Manager setup
  - Download and install Windows Assessment and Deployment Kit (ADK)
  - Create NO_SMS_ON_DRIVE.SMS files to prevent unwanted volumes to be used by Configuration Manager
- Active Directory configuration:
  - Extend Active Directory schema
  - Create System Management container in Active Directory
  - Configure permissions on System Management container
- SQL Server configuration:
  - Configure SQL Server memory usage settings
  - Validate SQL Server collation
  - Pre-create the Configuration Manager database
  - Configure SSRS database file size settings

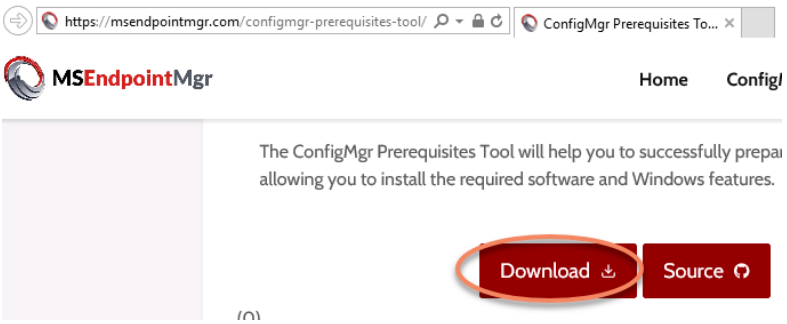
IMPORTANT NOTE: ***THESE COMMANDS HAVE ALREADY EXECUTED IN YOUR DOMAIN. YOU DO NOT NEED TO OPEN or DO ANYTHING FOR THIS EXERCISE. JUST REVIEW THE EXERCISE 2 IN THIS LABGUIDE. DO NOT PERFORM ANY OF THE STEPS BELOW. WE ARE SHOWING THEM JUST TO SHOW HOW EXERCISE 2 WAS SETUP.***

Instructions	Screenshot (if applicable)
1. Logon to CM01 as Administrator	
2. On the taskbar, click **Internet Explorer.**	

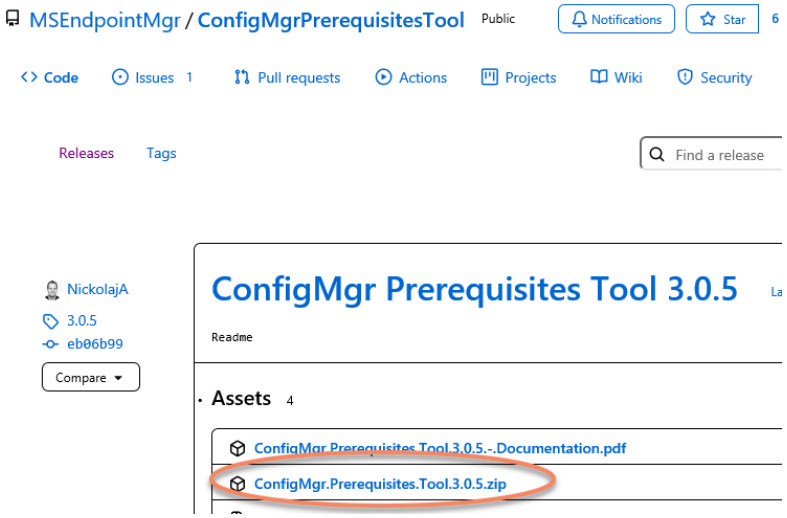
3. In the address bar, navigate to https://msendpointmgr.com/configmgr-prerequisites-tool/



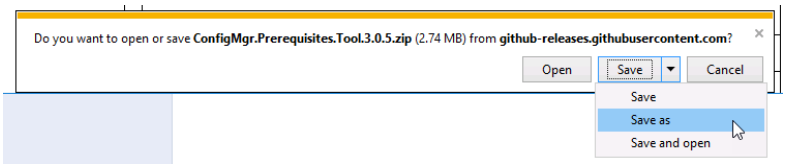
4. Scroll to the bottom of the page and click on the Download button.



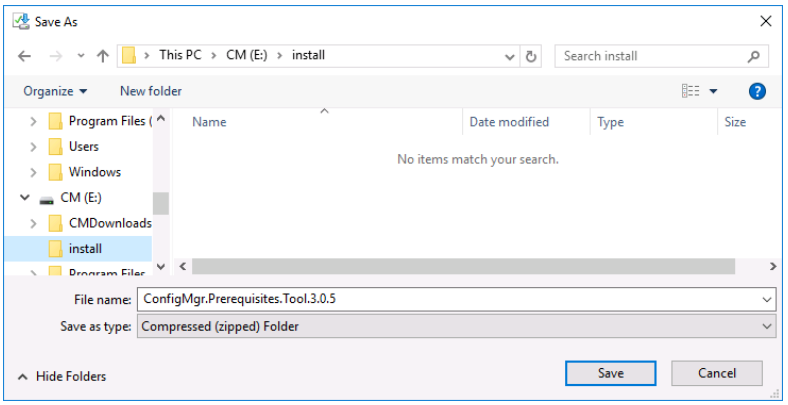
5. This will take you to the MSEndpointMgr Github repository. Click the ConfigMgr.Prerequisites.Tool.3.0.5.zip (or latest version)



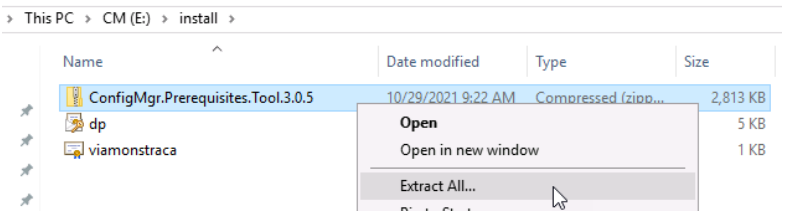
6. Click Save As in the pop-up window, and choose **Save As**.



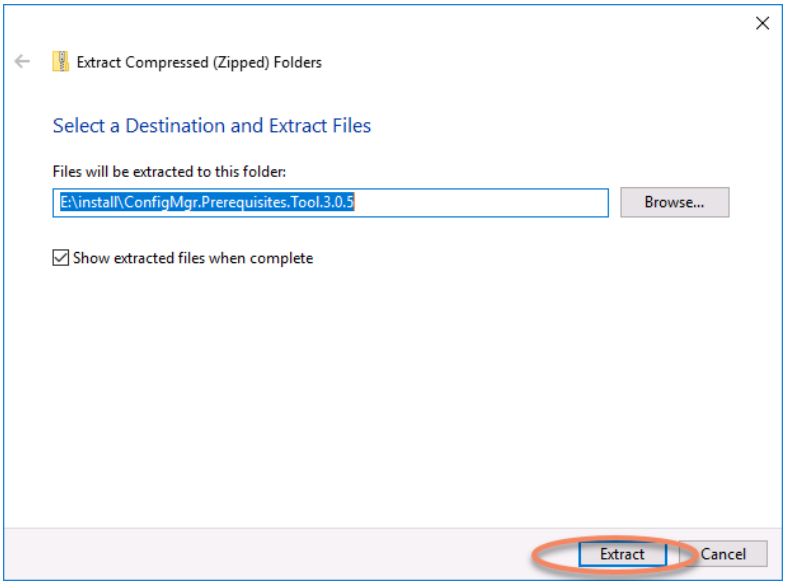
7. Save the file to E:\Install



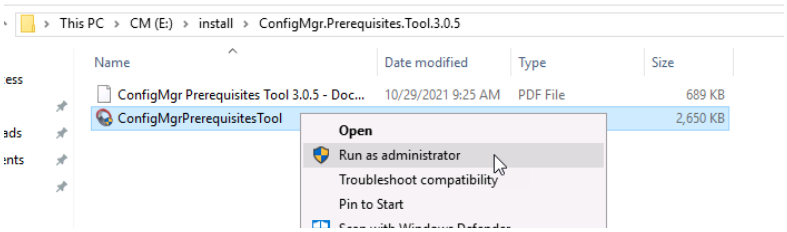
8. Right-click the file and choose **Extract All**.



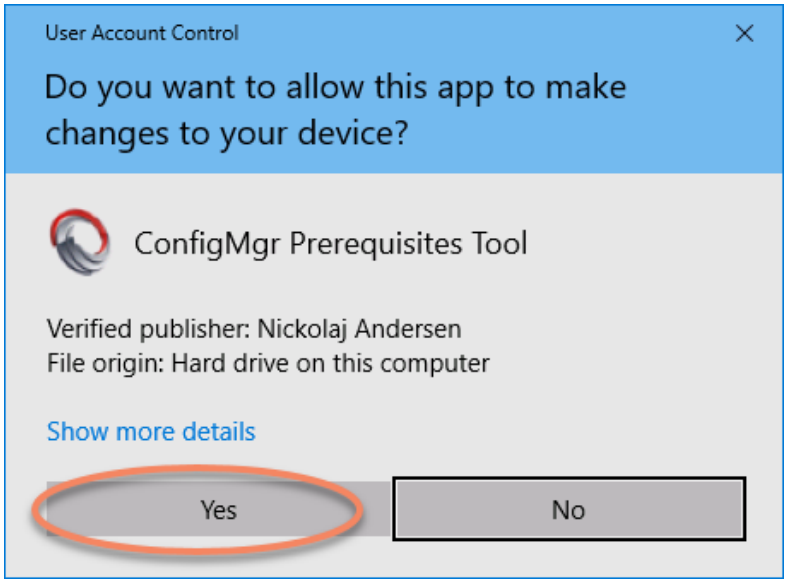
9. In the Extract Compressed (Zipped) Folders wizard, click Extract.



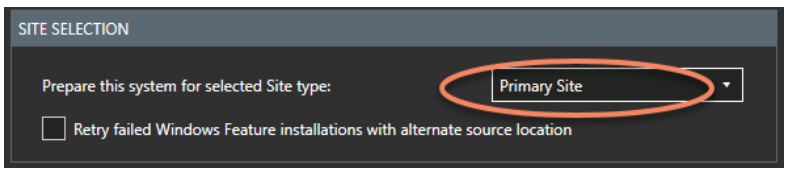
10. Right-click on the ConfigMgrPrerequisitesTool file, and choose **Run as Administator**.



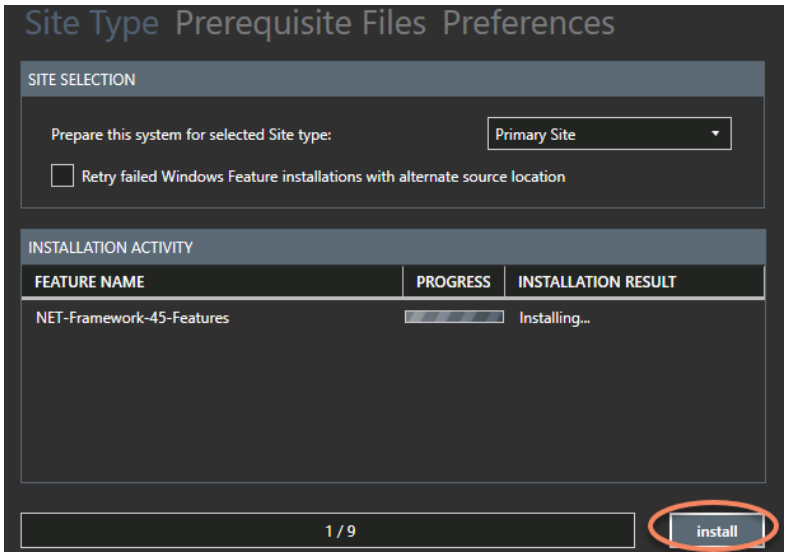
11. Select **Yes** on the User Account Control Prompt.



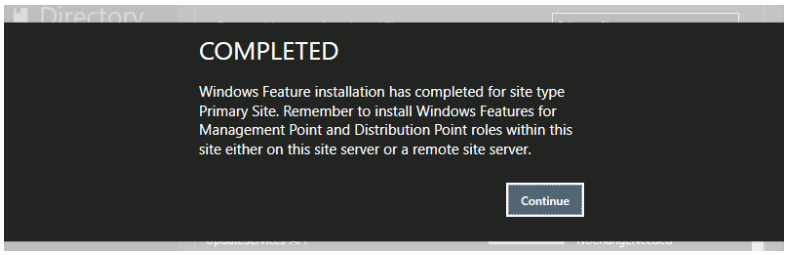
12. When the tool launches, ensure that **Primary Site** is selected in the Site Selection section.



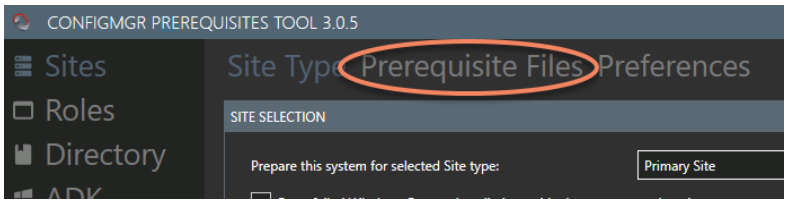
13. Click **install** to configure the roles and features required to support the site server.



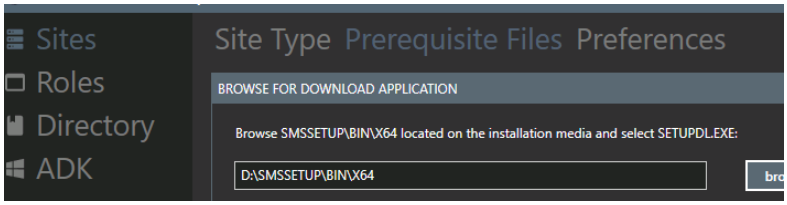
14. Completed! Click **Continue**.



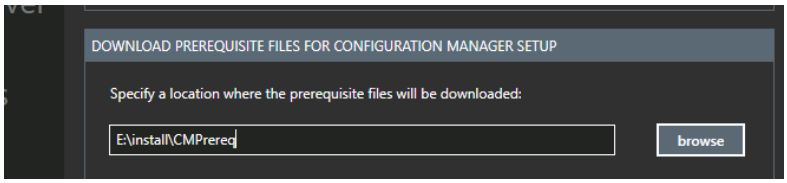
15. In the navigation row at the top, select **Prerequisite Files**.



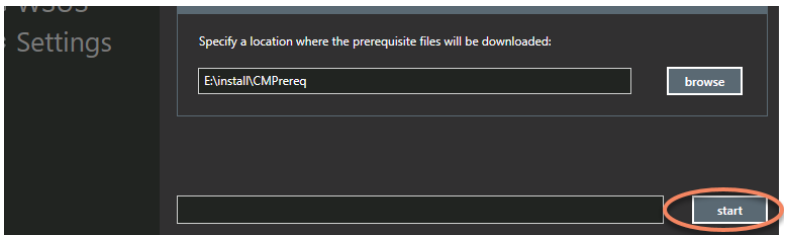
16. Browse to the correct path of the SMSSetup\bin\x64\SetupDL.exe file located on the ConfigMgr ISO file. (you will need to mount the ISO first by double-clicking it.



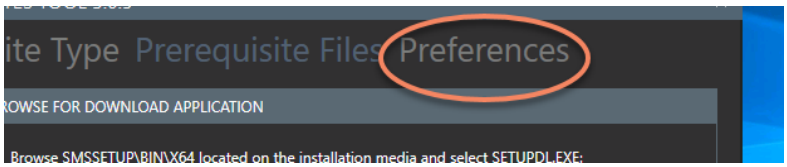
17. Below, specify an empty folder location where the downloaded preqrequisite files can be stored.



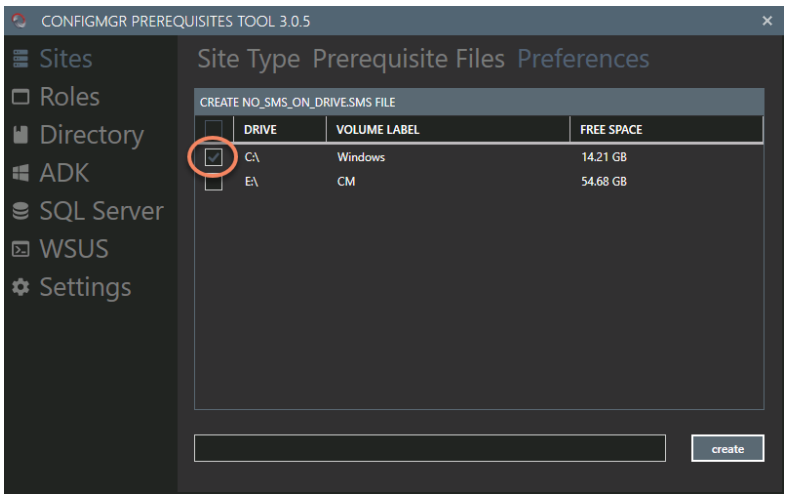
18. At the bottom of the window, click **Start** to initiate the download.



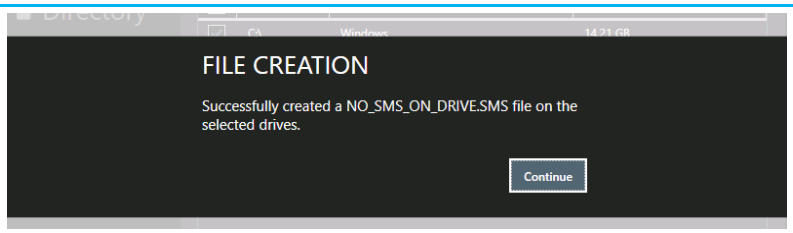
19. On the top navigation row, choose **Preferences**.



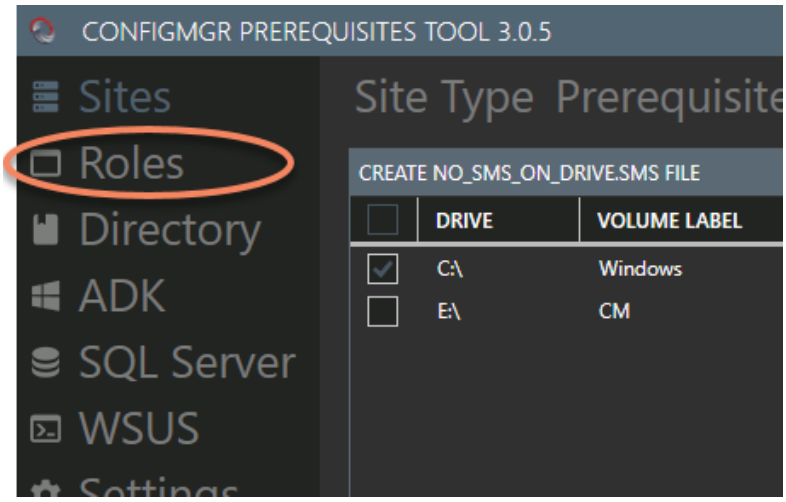
20. In Preferences, check the box for the C:\ drive to prevent CM from creating a content library on the OS drive, and click **Create**.



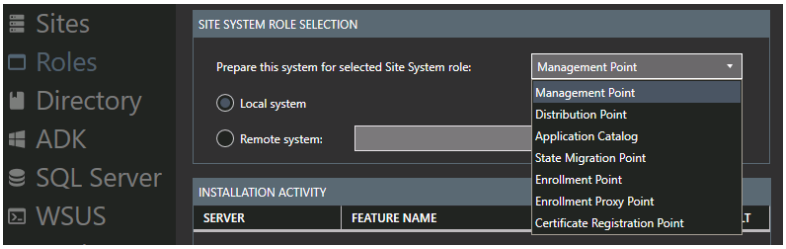
21. Click Continue when it finishes.



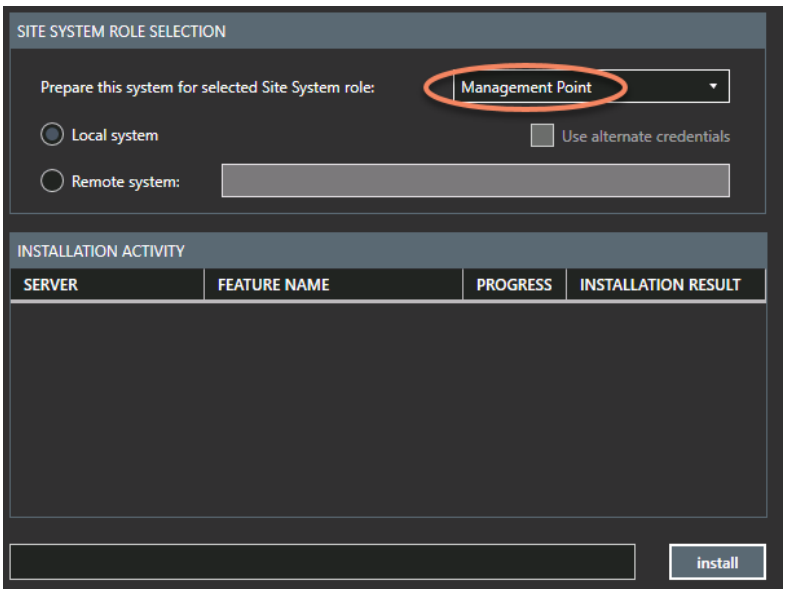
22. In the left hand navigation bar, select **Roles**.



23. The Role Selection section allows us to install preqrequisites for our required roles.

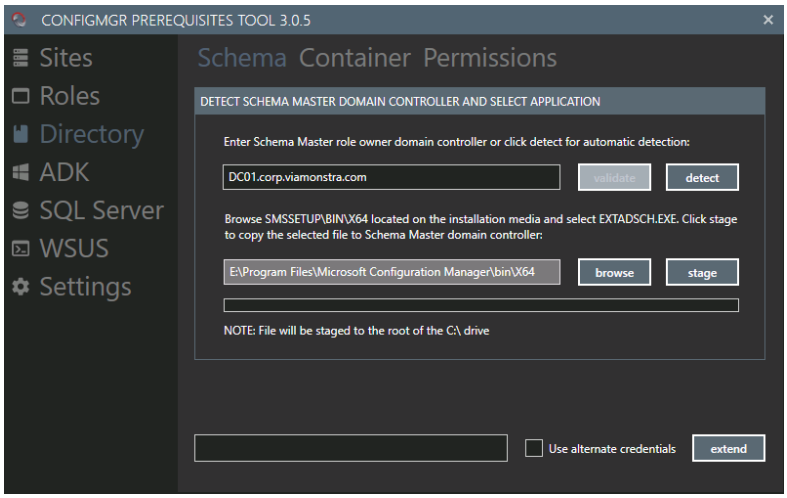


24. Select **Management Point**, then click the **install** button at the bottom.



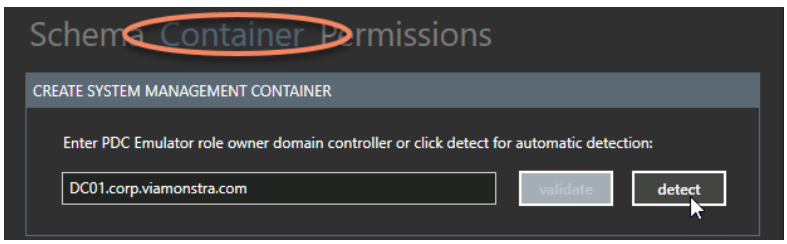
25. Once completed, repeat the process for Distribution Point, Application Catalog, State Migration Point, Enrollment Point, and Certificate Registration Point.

26. Back to the left-hand navigation pane, choose Directory. Click Detect to find the Schema Master, and locate EXTADSCH.exe in E:\Program Files\Microsoft Configuration Manager\bin\x64



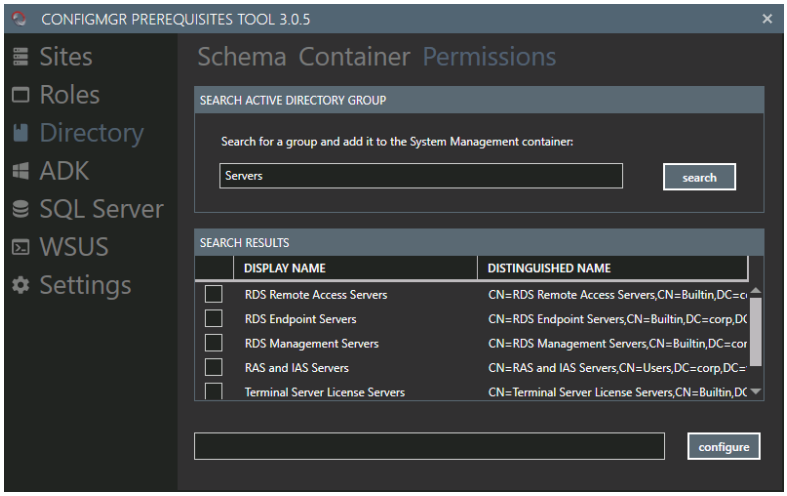
27. Click the **Extend** button to extend the AD Schema. **Note you need to be an enterprise admin or schema admin to complete this action.

28. At the top navigation row, click **Container** followed by the **detect** button to find the PDC Emulator.

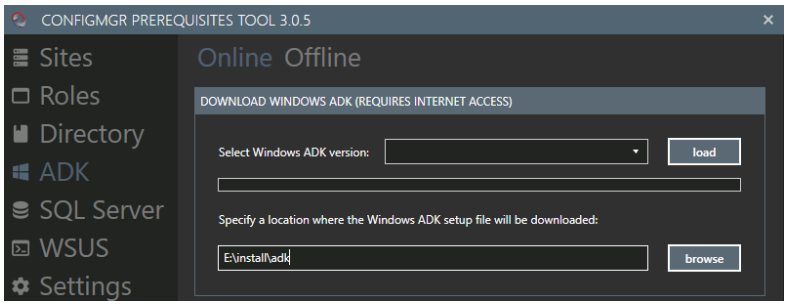


29. Click **create** at the bottom of the window to create the System Management container in Active Directory.

30. At the top navigation row, click **Permissions**. Here we can search for a group (ie: SMSSiteServers) and grant it permissions to publish site information to the Systems Management Container. Once you've selected the right group, click **configure**.

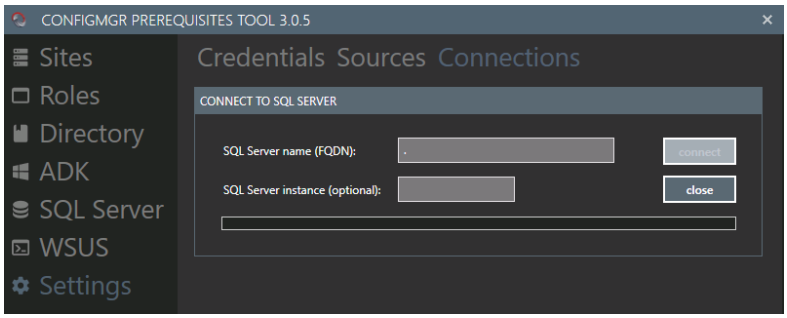


31. Next, click **ADK** on the left. There are two tabs here depending on whether we've already downloaded the ADK or not. Click the **load** button to populate the list of available ADK's

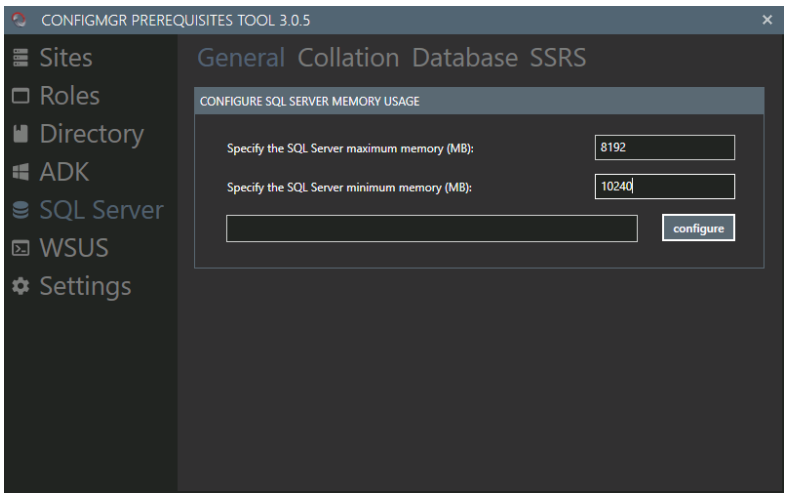


32. Change the download location from C:\Temp to **E:\install\adk**. Select the ADK for Windows 11 and click **install**.

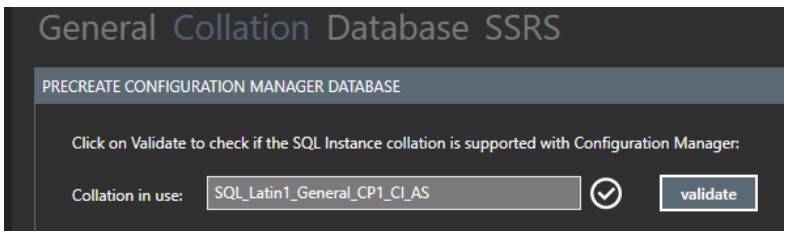
33. At this point we need to go to **Settings** at the bottom left and go to the **Connections** tab to connect to SQL. For a local install and default instance, we only need to place a period in the servername field.



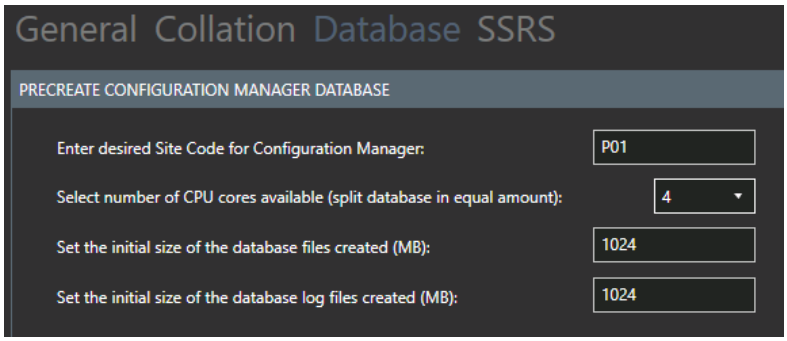
34. Next, to the SQL Server section, and on the General tab we want to specify the minimum and maximum settings for database memory. This is typically 65-80% of total memory on the Primary. Click **configure** when the right limits have been entered.



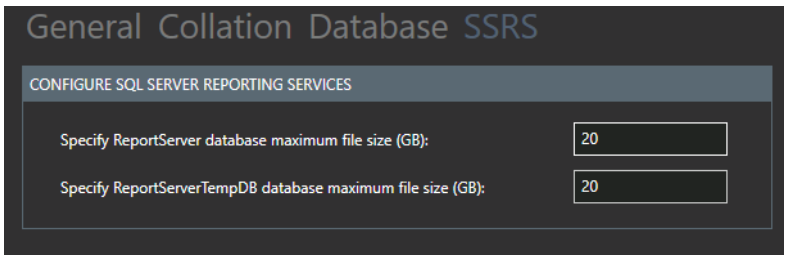
35. On SQL Server's Collation tab we can validate whether the instance has been configured with the proper collation.



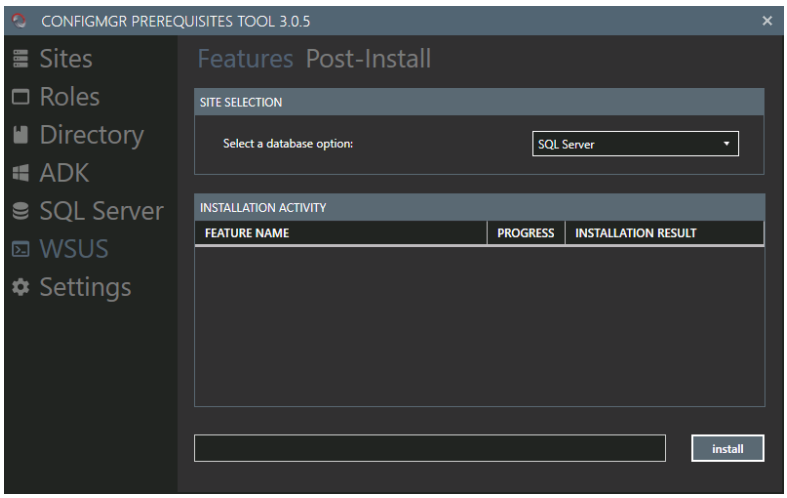
36. On the Database tab, we have the option to pre-create the database before installation. This allows us to specify the initial database and log sizes. The **create** button at the bottom will complete the process.



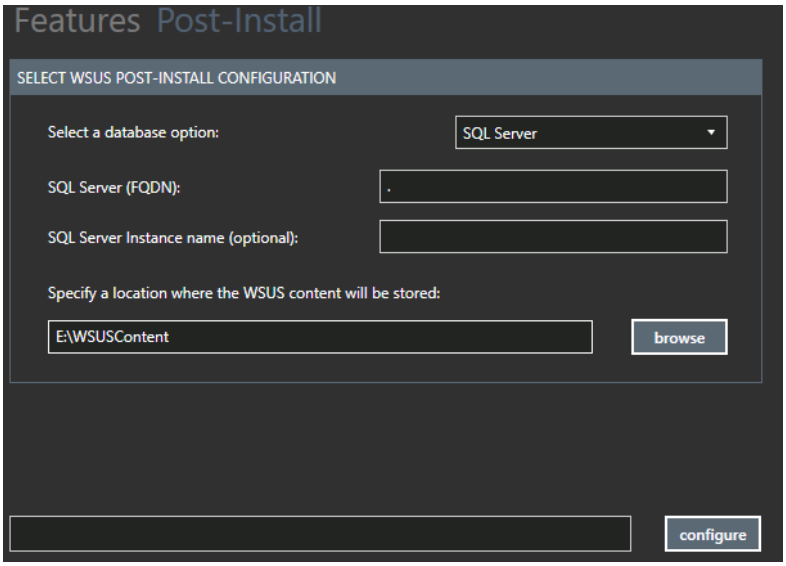
37. The SSRS tab will similarly allow us to pre-create the ReportServer and ReportServerTempDB databases and limits. You will need to click **configure** again to create the databases.



38. Finally, we have our last prerequisite section…preparing WSUS. On the Features tab we need to ensure **SQL Server** is selected for the database option, then we can click **install** to enable the feature



39. Click over to the Post-Install tab where we can automate WSUS's post-config. Make sure to change the WSUSContent folder to the E:\ drive before clicking **configure**.



Topic 2 – Automated MEM-CM Installation

Automating the installation of Configuration Manager has been capable and fully functional through its installer since 2012. After completing the installation wizard, all the information provided is saved to an information file in the temp directory, and you can find it at %temp%\ConfigMgrAutoSave.ini. This file contains all the required information, and can be modified and reused to make the process repeatable. Once the changes are made, save the file as **ConfigMgrUnattend.ini**.

IMPORTANT NOTE: ***THESE COMMANDS HAVE ALREADY EXECUTED IN YOUR DOMAIN. YOU DO NOT NEED TO OPEN or DO ANYTHING FOR THIS EXERCISE. JUST REVIEW THE EXERCISE 2 IN THIS LABGUIDE. DO NOT PERFORM ANY OF THE STEPS BELOW. WE ARE SHOWING THEM JUST TO SHOW HOW EXERCISE 2 WAS SETUP.***

We then launch the Configuration Manager setup wizard (from the SMSSetup\Bin\x64 directory of the installation media) with the following command:

SetupWpf.exe /Script E:\install\ConfigMgrUnattend.ini /NoUserInput

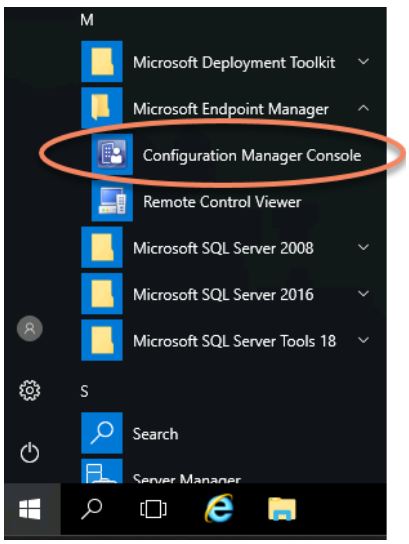
Anyone in Group A would use the same command we used, updating the values in the ini file for their respective environment. Here's the ini file for this lab:

```
[Identification]
Action=InstallPrimarySite

[Options]
ProductID=
SiteCode=PS1
SiteName=MEMCM Primary Site
SMSInstallDir=E:\Program Files\Microsoft Configuration Manager
SDKServer=CM01.corp.viamonstra.com
RoleCommunicationProtocol=EnforceHTTPS
ClientsUsePKICertificate=1
PrerequisiteComp=0
PrerequisitePath=E:\install\CMPrereq
MobileDeviceLanguage=0
ManagementPoint=CM01.corp.viamonstra.com
ManagementPointProtocol=HTTPS
DistributionPoint=CM01.corp.viamonstra.com
DistributionPointProtocol=HTTPS
DistributionPointInstallIIS=1
AdminConsole=1
JoinCEIP=0

[SQLConfigOptions]
SQLServerName=CM01.corp.viamonstra.com
SQLServerPort=1433
DatabaseName=CM_PS1
SQLSSBPort=4022
SQLDataFilePath=E:\SQLDB\
SQLLogFilePath=E:\SQLLOG\

[CloudConnectorOptions]
CloudConnector=1
CloudConnectorServer=CM01.corp.viamonstra.com
UseProxy=0
ProxyName=
ProxyPort=

[SystemCenterOptions]
SysCenterId=+hZORwDQlGhC0tYMQ93Cd26/IMSKaR5wVkssYl12WLM=
```

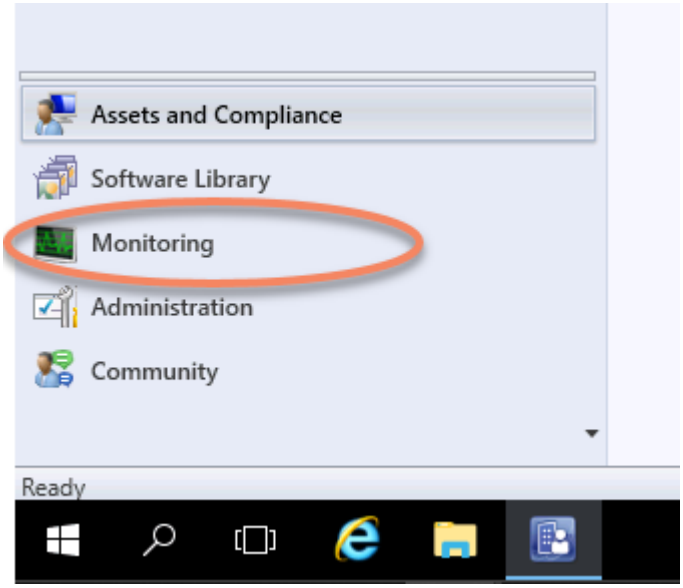
[SABranchOptions]
SAActive=1
CurrentBranch=1
[HierarchyExpansionOption]

Exercise 3 – Validate MEM-CM Core Functionality

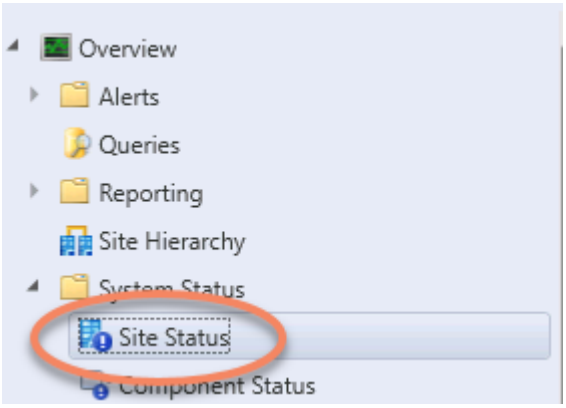
In the previous exercise, we ran the installation of Configuration Manager. All the roles should now be set up and functioning properly. We'll quickly make sure there's no problems, and get to automating the rest of the environment.

Instructions	Screenshot (if applicable)
1. Logon to CM01 as Administrator.	
2. Launch the MEM Console from the Start Menu.	

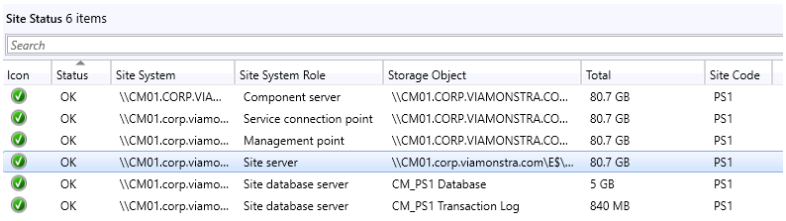
3. In the bottom left, click the Monitoring workspace.



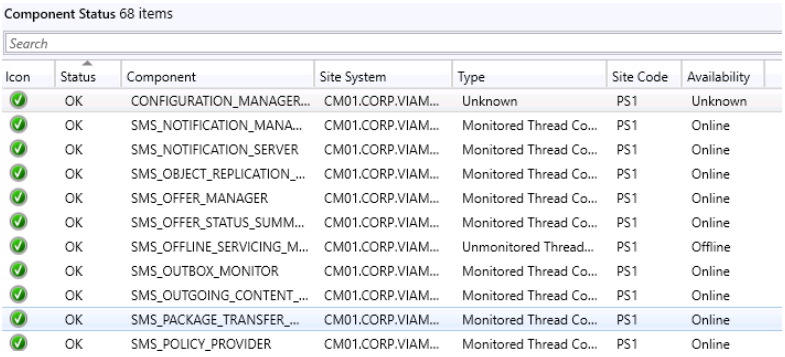
4. In the navigation pane, expand the **System Status** folder and click the **Site Status** node.



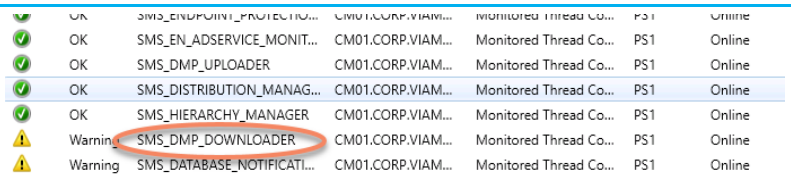
5. In the results pane you will see a list of Site System Roles with additional information including Status.



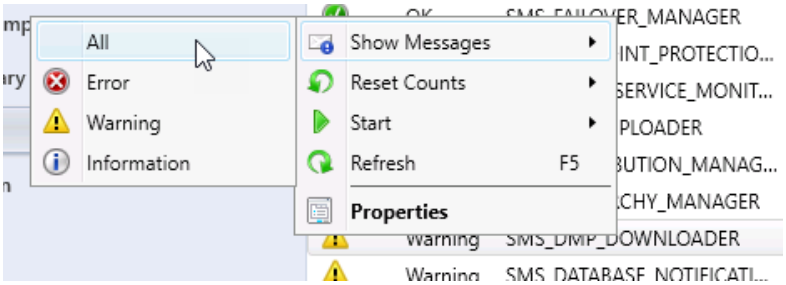
6. Next, click **Component Status** in the navigation pane. This list is much larger, and contains every component of the MEM implementation.



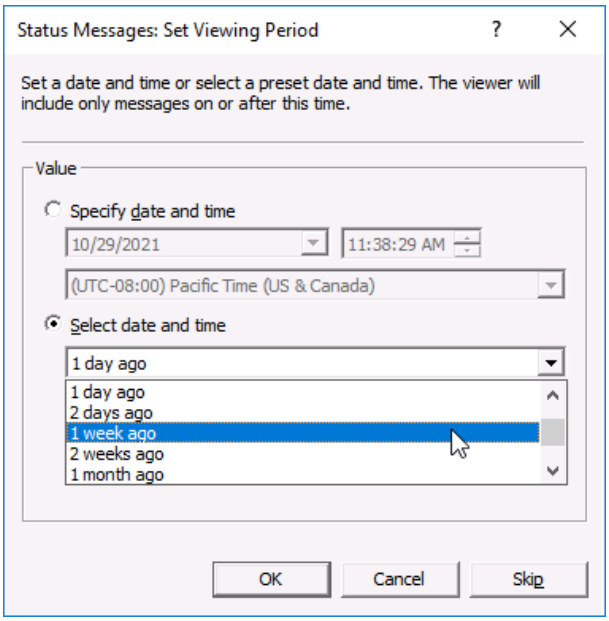
7. Scroll down the list and you should see a Warning for the SMS_DMP_DOWNLOADER.



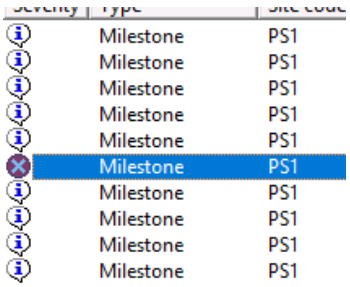
8. This is the service that downloads CM Updates from Microsoft. To see what's causing the error, right click on SMS_DMP_DOWNLOADER and chose Show Messages -> All



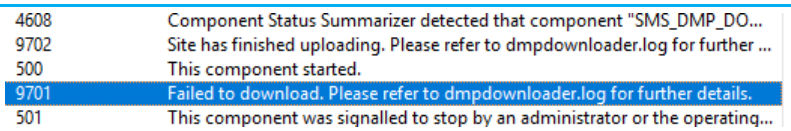
9. A wizard appears to select the viewing period. Change the period from "1 day ago" to **1 week ago** and click Ok.



10. The Status Message Viewer appears with a list of messageIDs. The severity of these items are either informational, warning, or error.



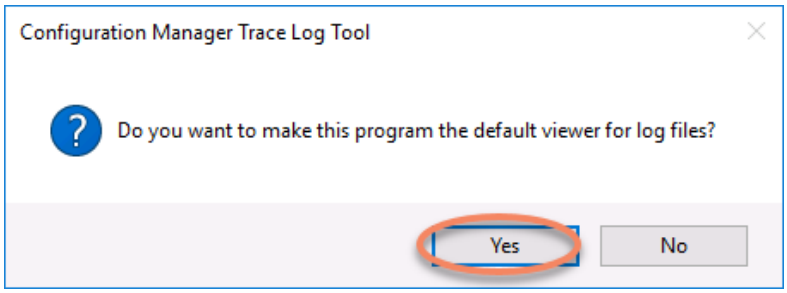
11. The error says there was a failure to download something and to check the dmpdownloader.log



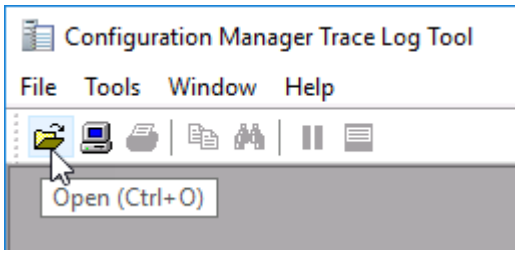
12. Launch File Explorer. Navigate to E:\Program Files\Microsoft

Configuration Manager\Tools.

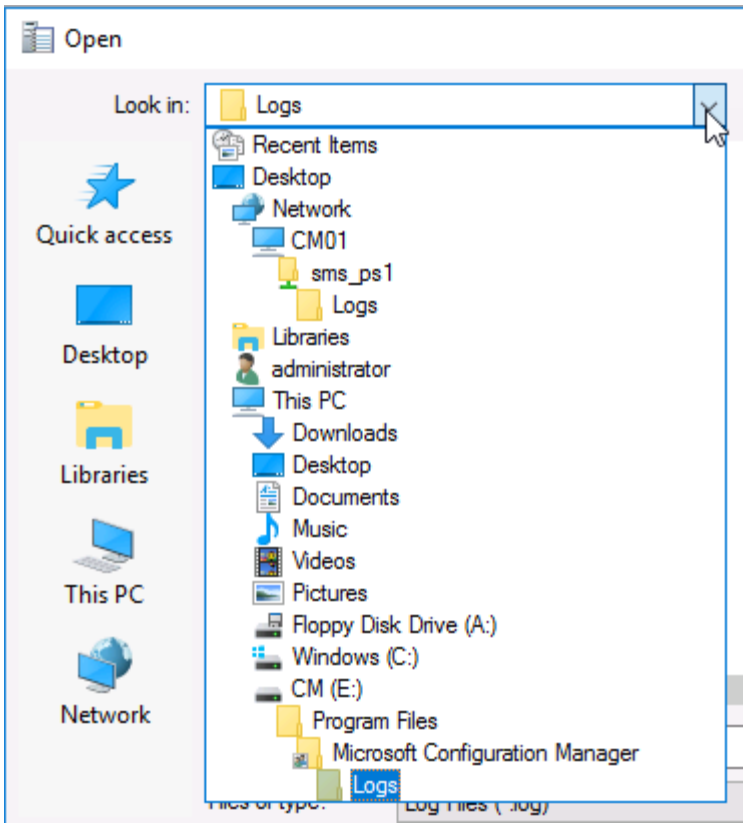
13. Right-click on CMtrace.exe and choose **Run as Administrator**. Set it as the default viewer for log files if prompted. If you don't receive this message…that's fine. Proceed.



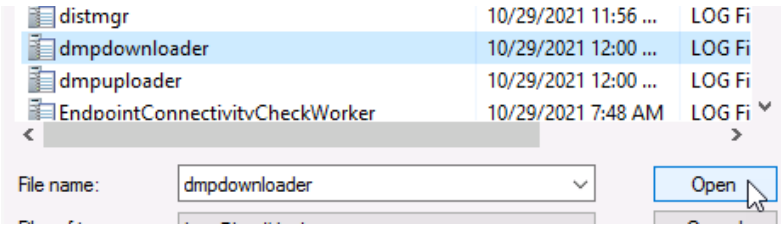
14. In CMTrace, click the open button at the top left



15. Browse to E:\Program Files\Microsoft Configuration Manager\Logs. This is the main location for *most* of the server log files, and is also accessible from the network share \\CM01\SMS_PS1\Logs.



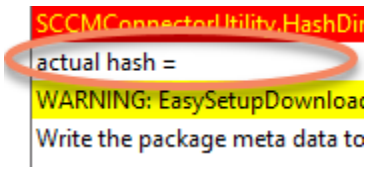
16. Scroll through the list until you find the dmpdownloader.log. Double-click it.



17. If we look in the log for the timestamp that was identified earlier, we find an error about being unable to validate a hash.

```
EasySetupDownloadSinglePackage downloading 42e1cf6e-95a1-4a8d-96ad-311e6247b3fb.
Generating state message: 8 for package 42e1cf6e-95a1-4a8d-96ad-311e6247b3fb
Write the state message in E:\Program Files\Microsoft Configuration Manager\inboxes\auth\statesys.box\incoming\high\__CMU0xkbomnj.SMX
Successfully Dropped the state message 8
Checking IsPayloadLocal with HashAlgorithm = SHA256 ...
expected hash = 8C6481748477759183FD4DEFA15521701BB37FE397355B61104328B668D630FE
Long path for hash calculation: \\?\E:\Program Files\Microsoft Configuration Manager\EasySetupPayload\42e1cf6e-95a1-4a8d-96ad-311e6247b3fb
Creating hash for algorithm 32780
AdminUI Content Download thread is exiting...
Waiting for child threads to exit.
waiting 5 seconds for child threads to exit
WARNING : Thread shutdown signalled
SCCMConnectorUtility.HashDirectoryFiles failed to calculate hash for E:\Program Files\Microsoft Configuration Manager\EasySetupPayload\42e1cf6e-95a1-4a8d-96ad
actual hash =
WARNING: EasySetupDownloadSinglePackage with exception: Object reference not set to an instance of an object.
Write the package meta data to connector's outbox
The CMU file name is E:\Program Files\Microsoft Configuration Manager\inboxes\hman.box\ForwardingMsg\__CMU42e1cf6e-95a1-4a8d-96ad-311e6247b3fb.MCM
```

18. If the actual hash equals nothing, there is either no content present or the server does not have permissions to access it.



19. A few lines down we notice that a package of the same GUID successfully downloads and the thread terminates.

```
The CMU file name is E:\Program Files\Microsoft Configuration Manager\inboxes\hman.box\ForwardingMsg\__CMU42e1cf6e-95a1-4a8d-96ad-311e6247b3fb.MCM
HasIntuneSubscription: Site has no Intune subscription.
outerxml is <ConfigurationManagerUpdateContent Guid="42e1cf6e-95a1-4a8d-96ad-311e6247b3fb" State="327679" ReportTime="2021-10-29T14:48:18" />
Successfully write the update meta into outbox for package 42e1cf6e-95a1-4a8d-96ad-311e6247b3fb
EasySetupDownloadSinglePackage finishes downloading 42e1cf6e-95a1-4a8d-96ad-311e6247b3fb.
STATMSG: ID=9701 SEV=E LEV=M SOURCE="SMS Server" COMP="SMS_DMP_DOWNLOADER" SYS=CM01.CORP.VIAMONSTRA.COM SITE=PS1 PID=1496 TID=3700 GMTDA
Updating LastSyncedTime registry value
Easy Setup Download interval is: every 1440 minutes
EasySetupDownload thread is exiting...
FastDownload thread exited.
EasySetupDownload thread exited.
AdminUI Content Download thread exited.
SMS_DMP_DOWNLOADER is exiting...
```

Its important to note, like that critical line above containing the empty hash value, errors in a process do not always equate to red or yellow lines in logs while viewed in CMTrace. When investigating an issue, you need to read every line that's logged around the time of the issue to accurately determine root cause, and if any manual intervention is necessary.

Each Component in ConfigMgr has different thresholds for triggering Warnings, and how many warnings constitute a failure.

This practice of checking the Site and Component Status within the console should be a daily task for every CM Admin.

Chapter 4 – Package Automation

Scenario

One of the primary functions of Configuration Manager is the ability to distribute software to multiple systems. Initially, however, the only software that's available to deploy is the CM client itself; the administrator is responsible for importing applications into CM before they can be deployed.

Generally, from a software packaging perspective, software falls into one of four categories:

- Internally created software
- Proprietary software not readily available
- Commercial software that requires modification
- Commercial software that can be deployed as-is

When software is built by internal teams, the creation process is limited by the expertise of the development team and does not follow the same scrutinous quality assurance process as most commercial software. Furthermore, packaging that application so that it can be distributed silently without requiring any user interaction can be a highly complicated process when applications are not created using standard development practices.

Oftentimes, successfully packaging these applications for deployment through Configuration Manager cannot be done with PowerShell and batch files alone, and requires purpose-built software. The most common software we see used for this purpose is either Advanced Installer, or AdminStudio by Flexera.

AdminStudio

AdminStudio Enterprise is the industry's leading solution designed to deliver unified Application Readiness for Windows, Mac, and Mobile applications in an environment of rapid and continual change.

AdminStudio improves service quality and streamlines service delivery of physical, virtual and mobile applications, allowing you to:

- Test, fix and package MSI and virtual packages, and cut MSI packaging time by up to 70%.
- Customize Mac .PKG installers to easily configure and prepare Mac applications.

- Automatically run Windows desktop and server compatibility tests and auto-generate standard MSI transforms to fix installation package related issues.
- Test Web site addresses (URLs) or Web applications to ensure they render correctly in Internet Explorer and Microsoft Edge.
- Assess Virtual suitability and convert MSIs and legacy installers to industry leading virtual formats including Microsoft App-V, VMware® Thin App™, and Symantec® Workspace Virtualization.
- Streamline the delivery of internal mobile apps and public links from Apple® App Store® and Google® Play®, to test for device and OS compatibility.

Orca

The vast majority of Windows applications today are installed from a Windows Installer .msi file. When this is the case, Microsoft provides a free database table editor for creating and editing merge modules (known as transforms) for the installer package. The tool is called Orca and is only available by downloading the SDK for Windows. You don't need to install the SDK, however, as the orca.msi file is contained within the ISO.

Orca is the go-to when you need to make changes to a commercial installer and don't have something like AdminStudio. There is one important point to mention about Orca and MSI files; using Orca, you can open an MSI file, modify it, and overwrite the original .msi file. This process can change the application GUID associated with the installation, and will adversely impact software monitoring and future patching of the application. As a result, it is recommended to never modify a commercial MSI, but rather create a TRANSFORMS file that contains the organization-specific settings you wish to alter or enable.

A packager's dream request is to prepare a commercially available product for deployment that does not require any modification. Think something like 7-Zip or Notepad++. Should be easy right? It is. How easy? Let's find out by looking at two of our favourite packaging tools: RuckZuck and Chocolatey.

Exercise 1 – RuckZuck

This is RuckZuck.tools, derived from a German word meaning “something that happens instantly,” a free Software Package Manager for Windows that’s designed to keep the Software on your System(s) up to date even if the Software was not installed with RuckZuck.

We use RuckZuck for ConfigMgr to integrate RuckZuck with Configuration Manager to automate the software import and packaging process for Commercial Off The Shelf (COTS) applications and updates. RuckZuck knows the links to the latest versions of COTS software and the parameters to perform a silent installation of the product. When instructed, the tool downloads the installation file(s) from the vendor's website, performs a hash check of the payload, then creates a silent installation Application/Package in CM.

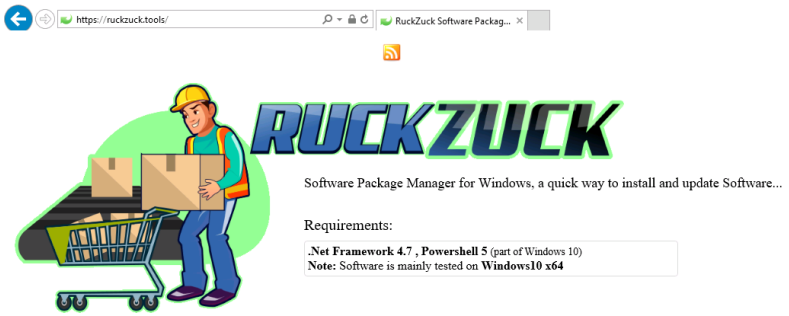
RuckZuck can also be used as a 3rd party patching tool, though does not provide the same level of automation in CM as the other (paid) products previously mentioned.

This product is free, open source, and therefore officially unsupported. There is no guarantee that the installation script used is functional as vendors sometimes change URL's and filenames with new versions. If an installation fails, however, there is a feedback form on the site to request an update, or you’re also welcomed to contribute to the community by creating and uploading a fixed software record for validation. Any data sent to RuckZuck as part of the tool’s operation is anonymized and does not contain any user or device information. This is good enough for most people on a tight budget, we always test our packages before deployment anyway, right? 😊

And now we’re gonna install it.

Instructions	Screenshot (if applicable)
1. Logon to CM01 as Administrator	

2. Launch Internet Explorer and navigate to https://ruckzuck.tools

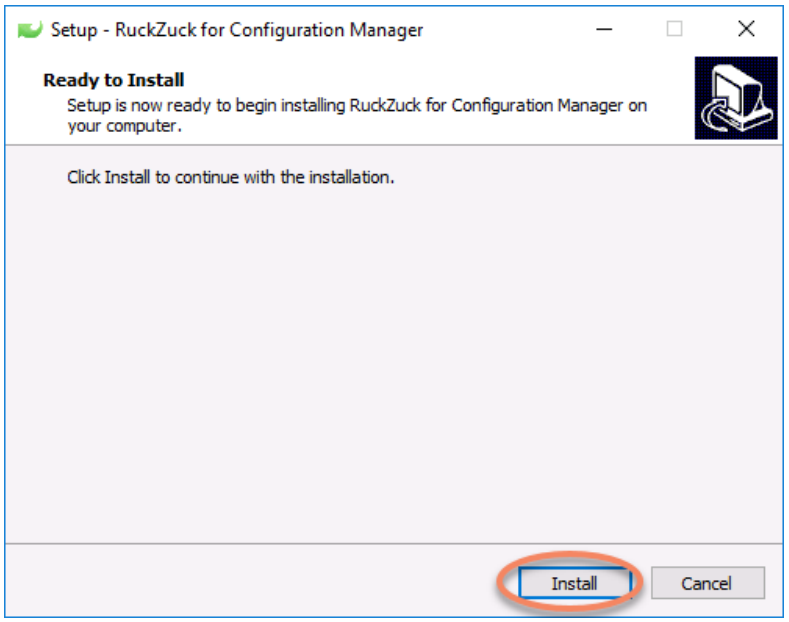


3.

4. In the downloads section click the link for RZ4ConfigMgr.exe

5. When the download pop-up appears, save the file to E:\installs, and click Run when it completes.

6. Click **Install** in the installation wizard.



7. When the installation completes, it will automatically open the RZ4CM.exe.config file in Notepad. Scroll to the ApplicationSettings section.

```
<applicationSettings>
    <RuckZuck_Tool.Properties.Settings>
        <setting name="CM_SQLServer" serializeAs="String">
            <value>localhost</value>
        </setting>
        <setting name="CM_SQLServerDBName" serializeAs="String">
            <value>CM_TST</value>
        </setting>
        <setting name="LimitingUserCollectionID" serializeAs="String">
            <value>SMS00003</value>
        </setting>
        <setting name="UpdateSettings" serializeAs="String">
            <value>True</value>
        </setting>
        <setting name="CreateDeployments" serializeAs="String">
            <value>True</value>
        </setting>
        <setting name="DPGroup" serializeAs="String">
            <value>All DP's</value>
        </setting>
        <setting name="DefaultADGroup" serializeAs="String">
            <value>%USERDOMAIN%\Domain Users</value>
        </setting>
        <setting name="OSRequirementsX64" serializeAs="Xml">
            <value>
```

8. Find "CM_SQLServerDBName" and replace the value of "CM_TST" with **CM_PS1**.

```
            <value>localhost</value>
        </setting>
        <setting name="CM_SQLServerDBName" serializeAs="String">
            value>CM_PS1</value
        </setting>
        <setting name="LimitingUserCollectionID" serializeAs="String">
```

9. Locate the DPGroup setting. Change the value from "All DP's" to **Main**. The Lab does not currently have a Main DP Group…we will need to create one.

```
        <setting name="CreateDeployments" serializeAs="String">
            <value>True</value>
        </setting>
        <setting name="DPGroup" serializeAs="String">
            <value>Main</value>
        </setting>
```

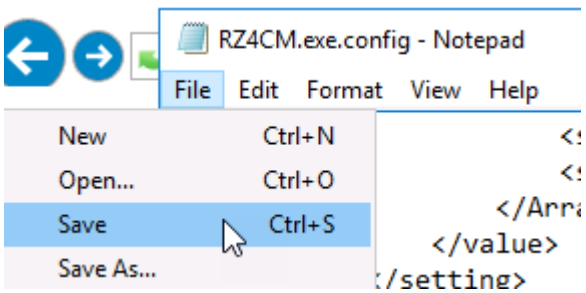
10. Further down, below the OS requirement stuff, is a setting for PrimaryUserRequired. Because this is a lab, we need to change that from "True" to **False**.

```
            xmlns:xsd="http://www.w3.org/2001/XMLSchema">
            <string>Windows/All_x86_Windows_7_Client</str
            <string>Windows/All_x86_Windows_8.1_Client</s
            <string>Windows/All_x86_Windows_10_and_higher
        </ArrayOfString>
    </value>
</setting>
<setting name="PrimaryUserRequired" serializeAs="String">
    <value>False</value>
</setting>
<setting name="CM_Server" serializeAs="String">
```

11. Next, update the package source location path to **\\localhost\Sources\RuckZuck**.

```
<setting name="PrimaryUserRequired" serializeAs="String">
    <value>False</value>
</setting>
<setting name="CM_Server" serializeAs="String">
    <value>localhost</value>
</setting>
<setting name="CMContentSourceUNC" serializeAs="String">
    <value>\\localhost\Sources\RuckZuck</value>
</setting>
```

12. Save the .config file and close it.

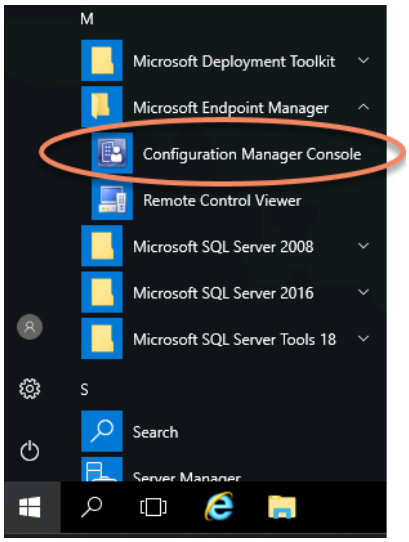


13. Open File Explorer
14. Browse to E:\
15. Right-Click on "sources" and choose **Properties**.
16. On the Sharing tab, click **Advanced Sharing**.
17. Check the box to share the folder, and click **Permissions**.

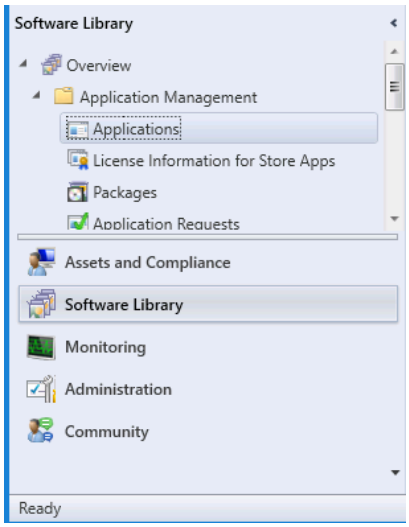
18. Check the box to grant Everyone Full Control of the share.

19. Click OK to close the Permissions, OK to close the Sharing, and OK to close the Properties.

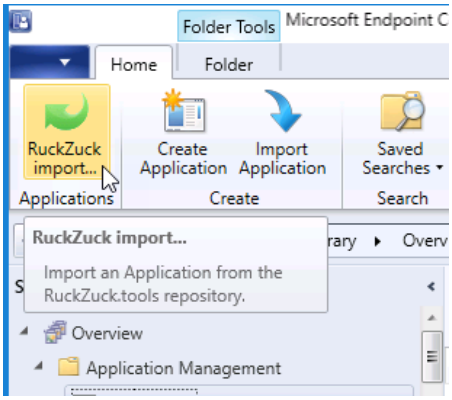
20. Launch the Configuration Manager Console



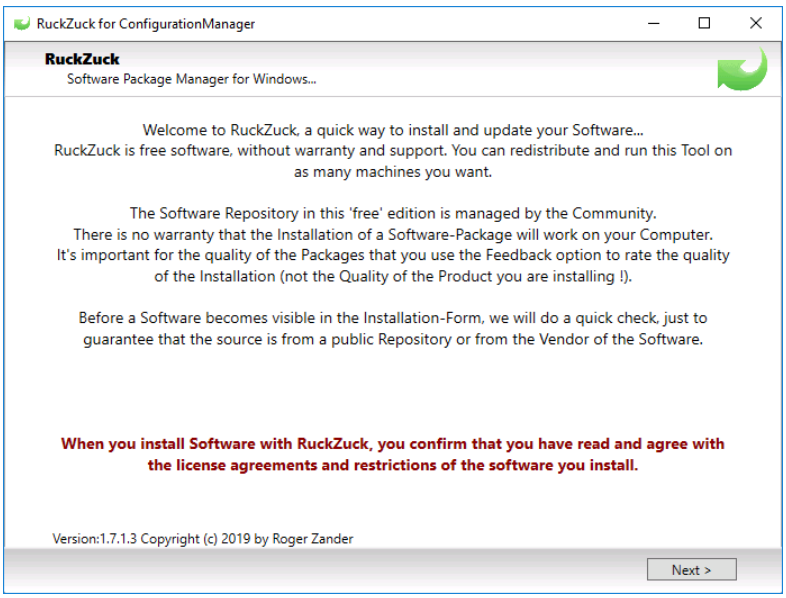
21. At the bottom left of the Configuration Manager Console, click the Software Library workspace, expand the Application Management folder, and click on Applications.



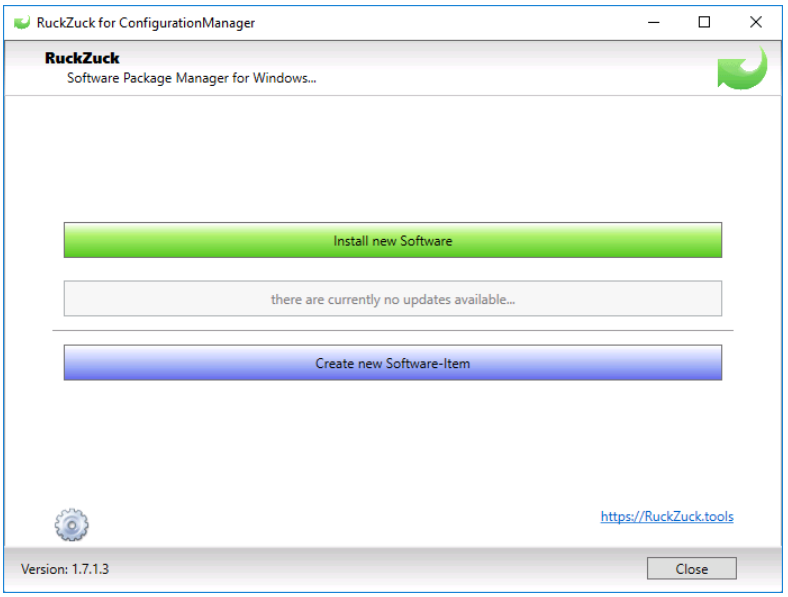
22. Up in the ribbon, you should see a new icon, “RuckZuck import…” Click it.



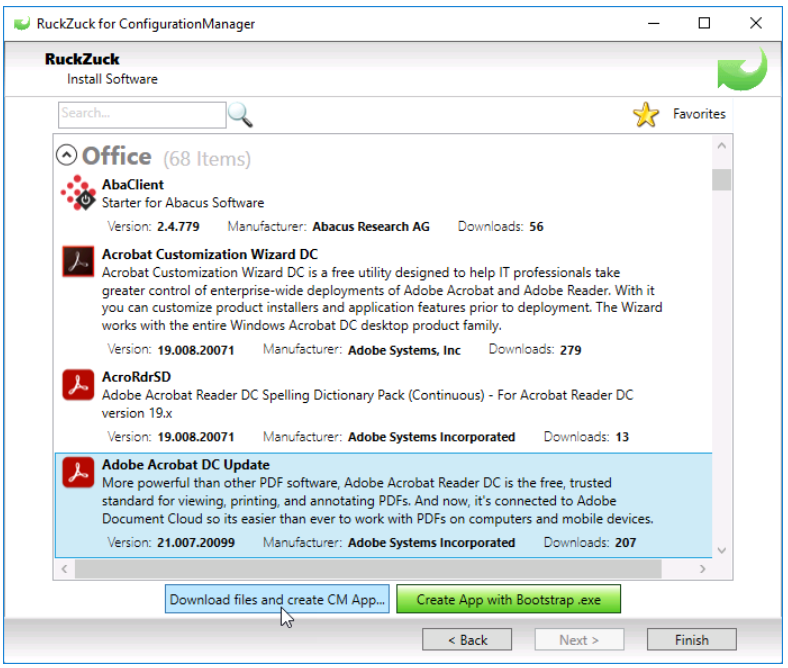
23. RuckZuck for ConfigurationManager will launch. Click Next.



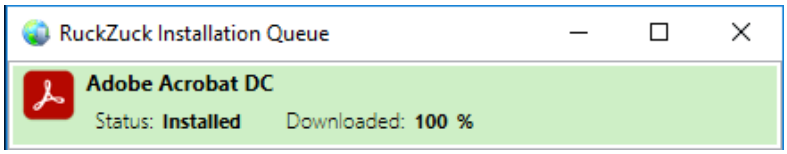
24. Click Install new Software.Click
25. On the next screen, click Install new Software.



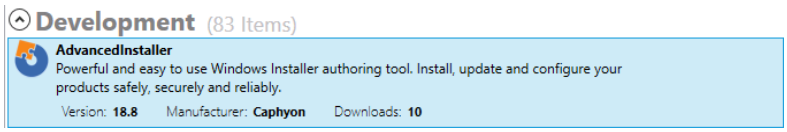
26. In the Install Software wizard, expand the Office group and select **Adobe Acrobat DC Update**, then click the **Download files and create CM App…** button.



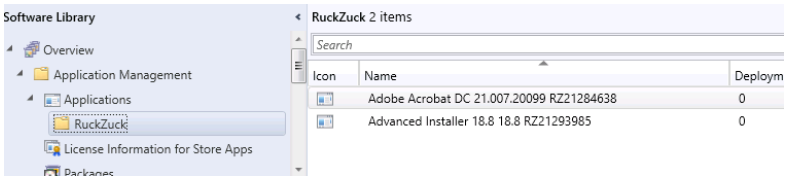
27. RuckZuck will download the application, package it up, distribute it, and add it to a collection. When finished, the Status will show Installed.



28. Now find AdvancedInstaller under the Development group. Grab that one too. Click **Finish** when it's installed.



29. Under Applications in the CM Console there is now a RuckZuck folder (you may have to press F5 to refresh the screen first).



Exercise 2 – Chocolatey

Another way we can do this is with Chocolatey. Chocolatey works different than RuckZuck, but has an Easter egg of sorts. Depending on your environment, this can be a blessing or a curse. Let's look into how it works:

Chocolatey works by deploying an agent to your workstation. When you need to install a piece of software, you just run a command, like "**choco install 7zip"** and the agent will download the installer directly from the internet and install it on the device.

From a packaging process, this is amazing. We only have to package the Chocolatey agent and deploy it out. If I want to deploy an application or upgrade using Chocolatey, I just need to create a one-line script to tell the clients to go grab and install the software. Not only does this exponentially reduce the packaging effort for commercial applications, but we don't have to needlessly store the software in content stores and on distribution points in the environment.

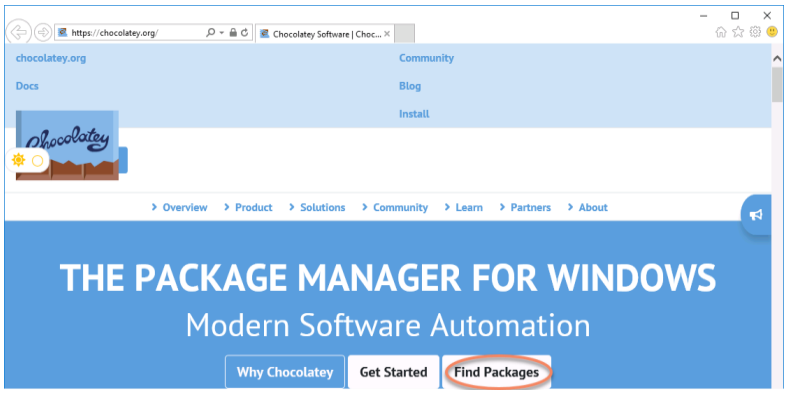
In a pandemic era when a lot of people are working from home, this can be advantageous as software distribution traffic is removed from the corporate environment. Unless your VPN is a

full tunnel, at which point the internet traffic to download the software to all your clients is all pushed unoptimized through your VPN gateway.

This is one example as to why implementing Configuration Manager is a project that involves multiple departments. The network team specifically needs to not only know what we're doing, but also how the tool will be utilized on the internal and external network.

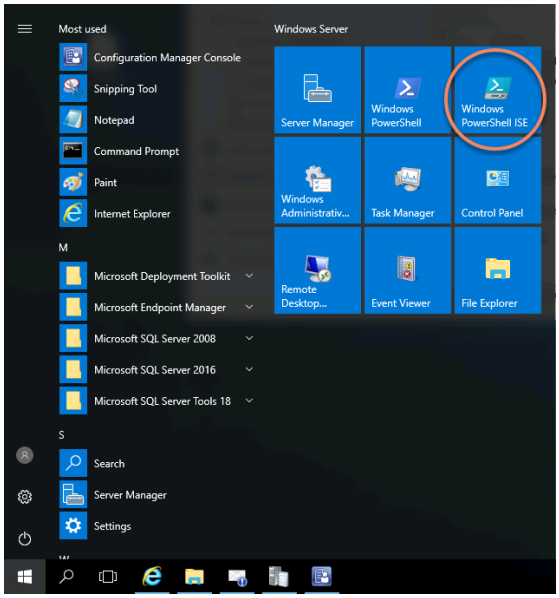
Chocolatey Client

Our first step is to install the Chocolatey client. On a workstation, we can simply call https://chocolatey.org/install.ps1 locally and be done with it, however a CM application requires just a little more effort.

Instructions	Screenshot (if applicable)
1. Logon to CM01 as Administrator	
2. Open Internet Explorer and go to https://www.chocolatey.org and click the Find Packages link	
3. The resulting page will give you a list of thousands of applications that can be installed. Notice the install string to	

the right. We don't need them now, but we will later on.

4. Open PowerShell ISE.

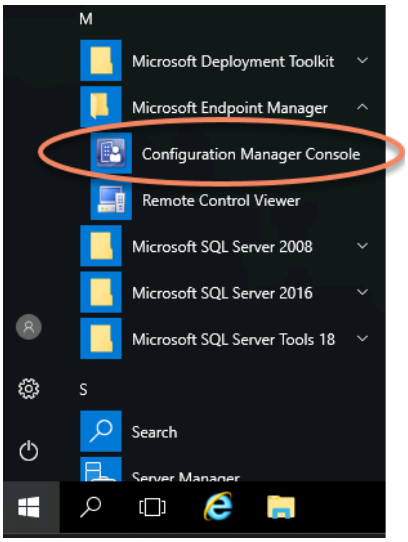


5. Copy and paste the following lines of code, and save the file as **Install.ps1** in **E:\Sources\Software\Applications\Chocolatey\Client**.

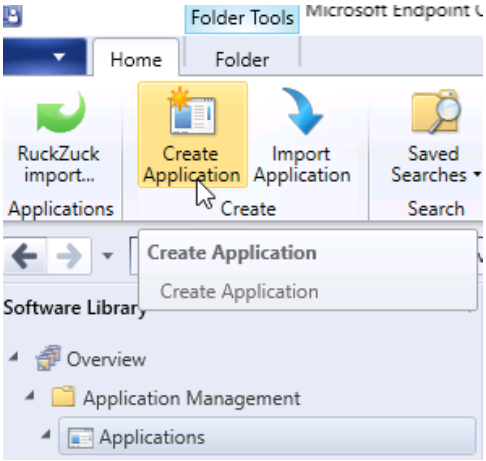
```
# The 3 variables below need to match with those specified in the Powershell
detection script
$ChocoPackageInstallCode = "choco upgrade `"`$packageName`" --yes";
$ChocoPackageUninstallCode = "choco uninstall `"`$packageName`" --yes";
$ChocoPackageDetectionCode = "try {if((choco list `"`$(`$packageName)`" --
localonly) -match `"`$(`$packageName)`") {`$vers = ([regex]::Matches((choco
upgrade `"`$(`$packageName)`" --yes --force --whatif),
`"(?<=(v|\s))(\d+|\.)+(?=\s)`")).Value;if(@(`$vers).Count -eq 2) {`"Installed
and up to date`";}}} catch {}";

# ------------------------------------------------------------------

if($env:Path -match 'chocolatey') {
choco upgrade chocolatey --yes;
}
```

```
else {
Set-ExecutionPolicy Bypass -Scope Process -Force;
iex ((New-Object
System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'));
};

choco feature disable -n ShowNonElevatedWarnings;

[System.Environment]::SetEnvironmentVariable("ChocoPackageInstallCode",
$ChocoPackageInstallCode, [System.EnvironmentVariableTarget]::Machine);
[System.Environment]::SetEnvironmentVariable("ChocoPackageUninstallCode",
$ChocoPackageUninstallCode, [System.EnvironmentVariableTarget]::Machine);
[System.Environment]::SetEnvironmentVariable("ChocoPackageDetectionCode",
$ChocoPackageDetectionCode, [System.EnvironmentVariableTarget]::Machine);
```

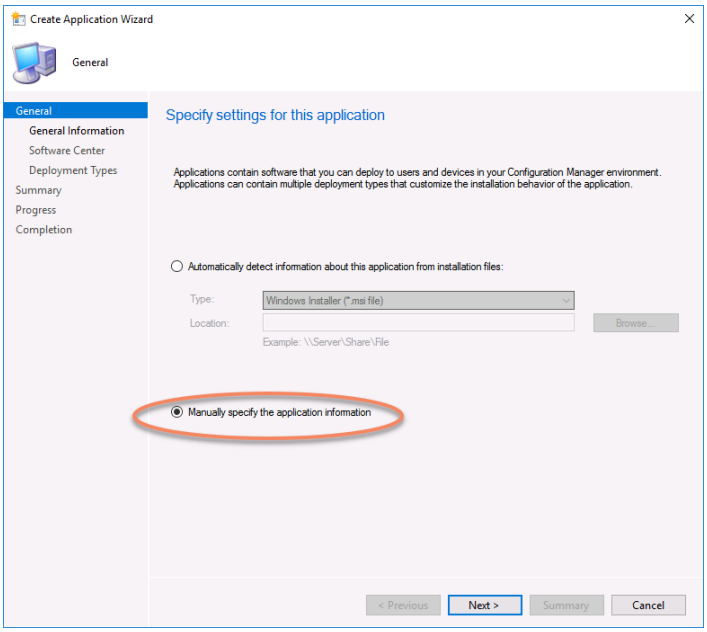
6. Launch the Configuration Manager Console.



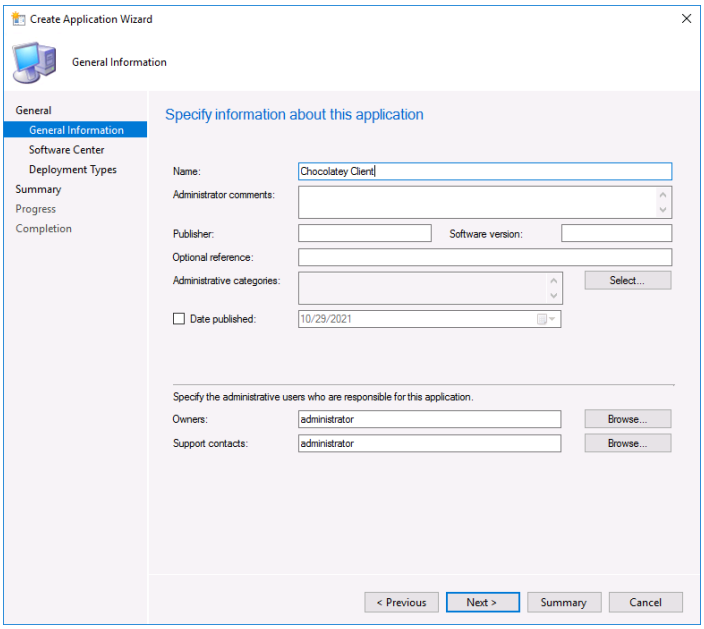
7. In the Software Library workspace, expand Application Management and select Applications. From the ribbon, choose **Create Application**.



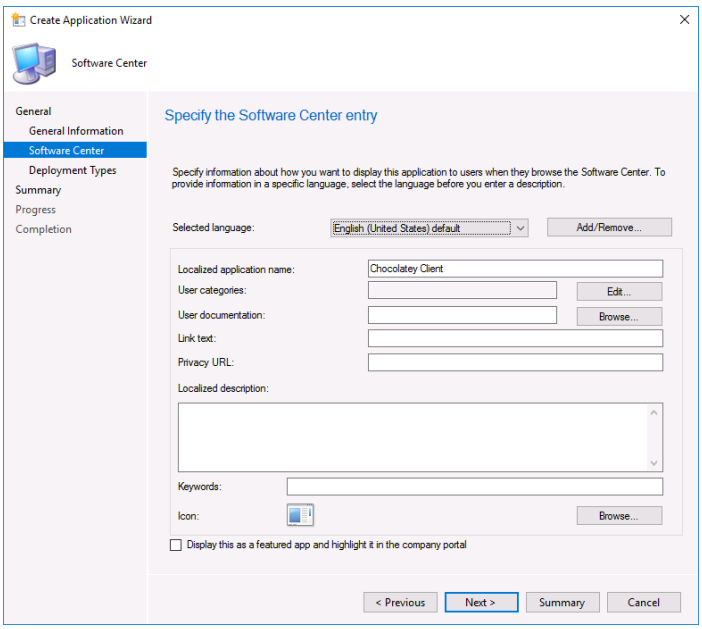
8. In the Create Application Wizard, choose the "Manually specify the application information" radio button and click Next to continue.



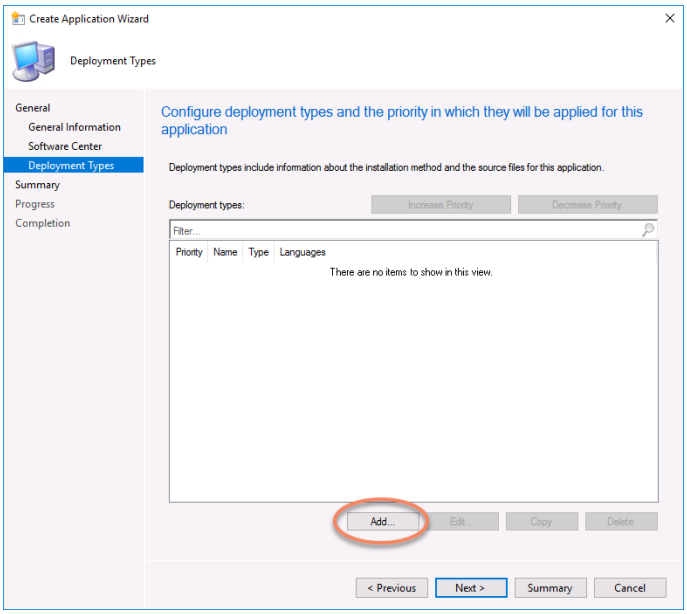
9. Call the application **Chocolatey Client** and leave all the other fields default.



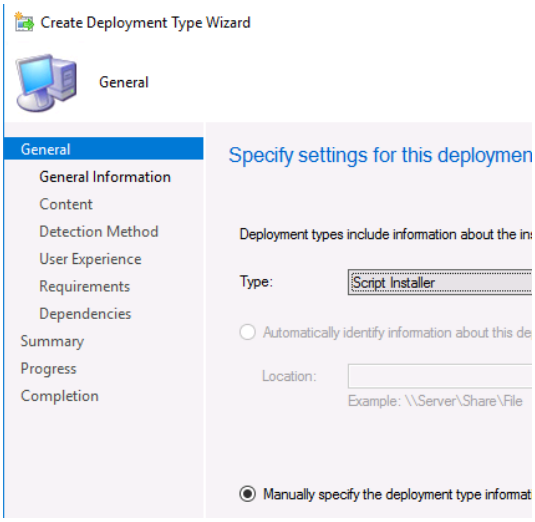
10. Leave the Software Center page as-is and click Next.



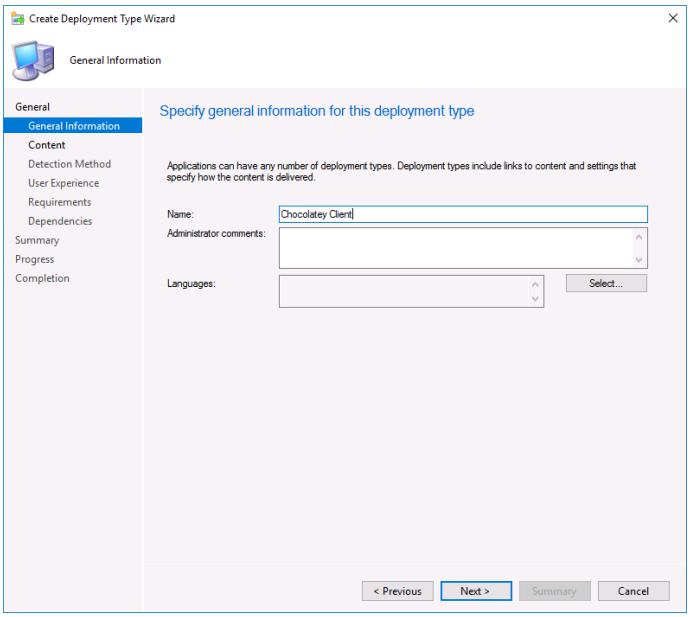
11. On the Deployment Types page, click Add.



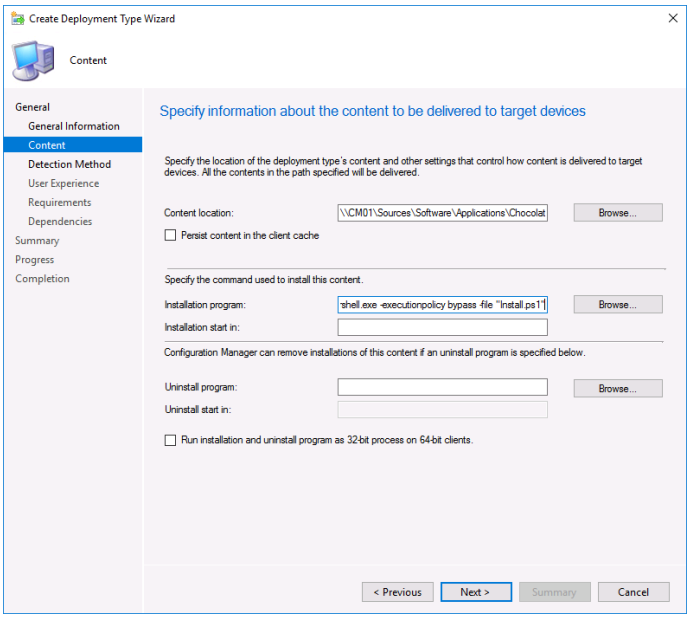
12. In the Create Deployment Type Wizard, choose “Script Installer” for the type and click Next.



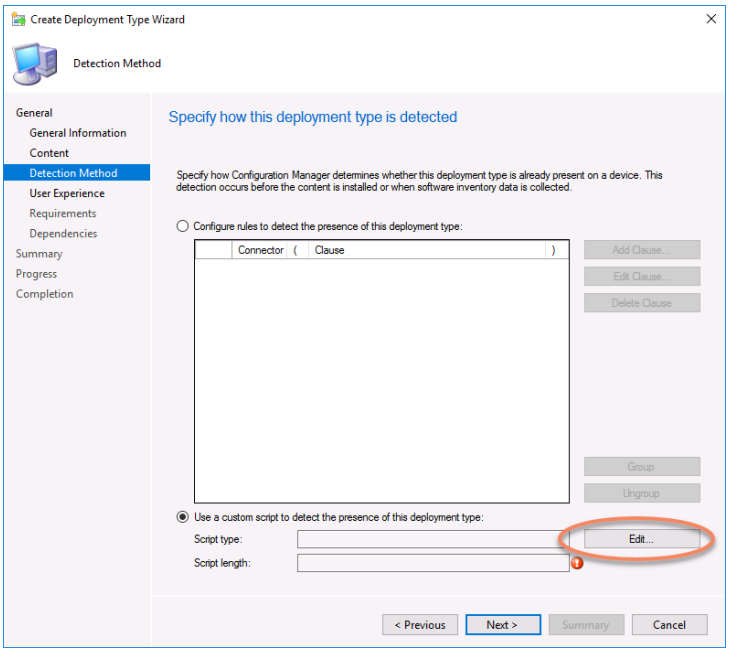
13. Under General Information, call it Chocolatey Client again.



14. On the Content Tab, populate the following 2 fields:
    Content Location: \\CM01\Sources\Software\Applications\Chocolatey
    Installation program: powershell.exe -executionpolicy bypass -file "Install.ps1"



15. For a Detection Method, we're going to select the bottom radio button to use a custom script. Click Edit.

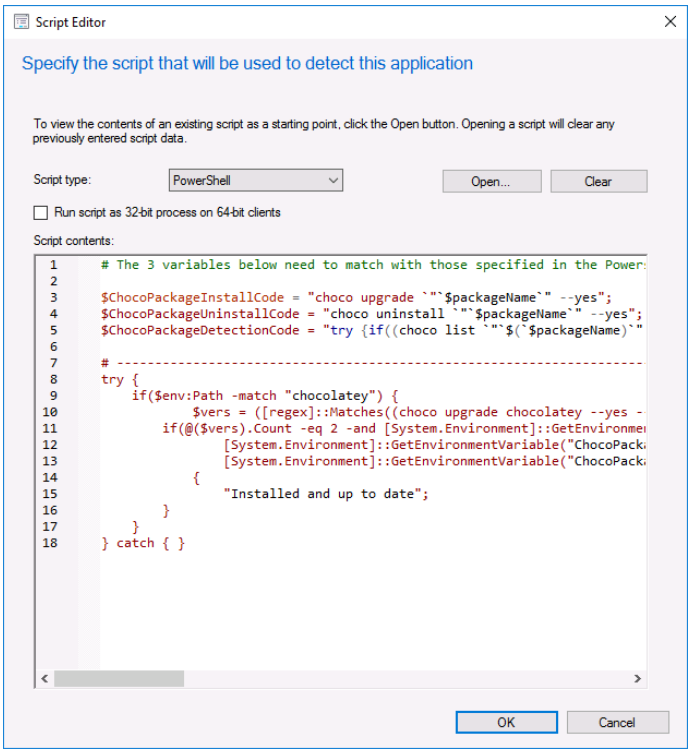


In the Script Editor window, choose Powershell (of course!) for the Script type, and paste the following lines of code in the script contents:

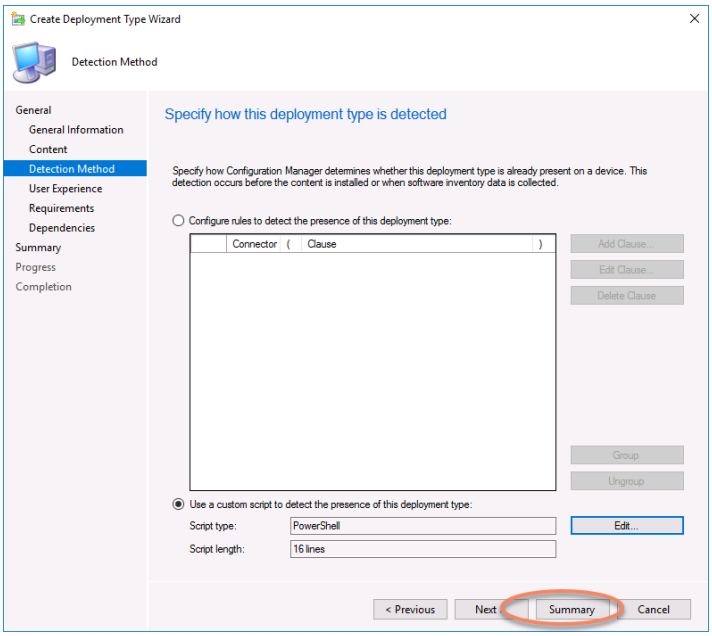
```
# The 3 variables below need to match with those specified in the Powershell
install script

$ChocoPackageInstallCode = "choco upgrade `"`$packageName`" --yes";
$ChocoPackageUninstallCode = "choco uninstall `"`$packageName`" --yes";
$ChocoPackageDetectionCode = "try {if((choco list `"`$(`$packageName)`" --
localonly) -match `"`$(`$packageName)`") {`$vers = ([regex]::Matches((choco
upgrade `"`$(`$packageName)`" --yes --force --whatif),
`"(?<=(v|\s))(\d+|\.)+(?=\s)`")).Value;if(@(`$vers).Count -eq 2) {`"Installed
and up to date`";}}} catch {}";
# --------------------------------------------------------------------------
try {
    if($env:Path -match "chocolatey") {
            $vers = ([regex]::Matches((choco upgrade chocolatey --yes --force --
whatif), "(?<=(v|\s))(\d+|\.)+(?=\s)")).Value;
        if(@($vers).Count -eq 2 -and
[System.Environment]::GetEnvironmentVariable("ChocoPackageInstallCode",
[System.EnvironmentVariableTarget]::Machine) -eq $ChocoPackageInstallCode -and
```

```
                [System.Environment]::GetEnvironmentVariable("ChocoPackageUninst
allCode", [System.EnvironmentVariableTarget]::Machine) -eq
$ChocoPackageUninstallCode -and
                [System.Environment]::GetEnvironmentVariable("ChocoPackageDetect
ionCode", [System.EnvironmentVariableTarget]::Machine) -eq
$ChocoPackageDetectionCode)
            {
                "Installed and up to date";
        }
    }
} catch { }
```

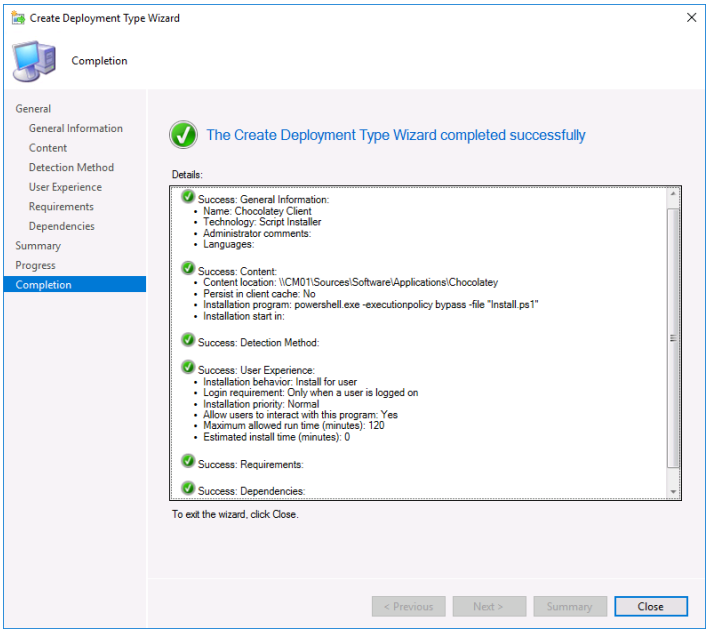
16. Once the script has been pasted, click OK to close the Script Editor.



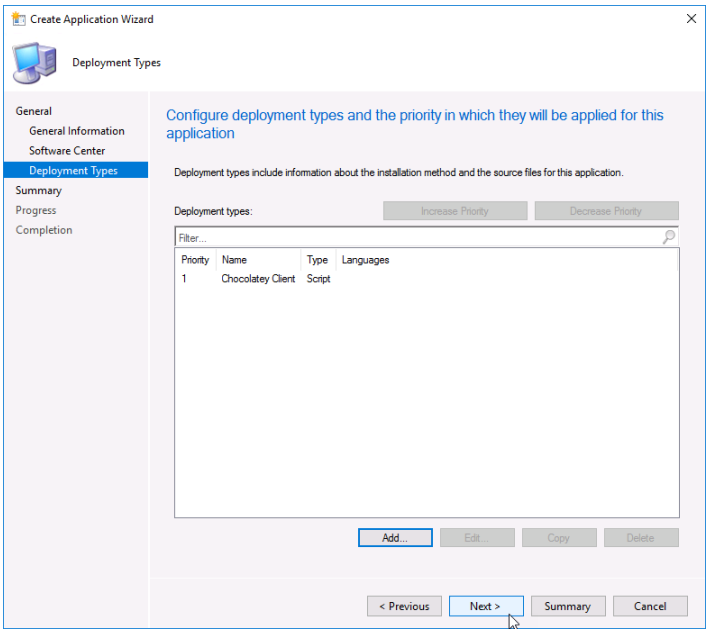
17. Click **Summary** at the bottom of the Detection Method page, then click Next.



18. Click Close on the completion page.



19. Back in the Create Application Wizard, Click Next, Next, and Close.

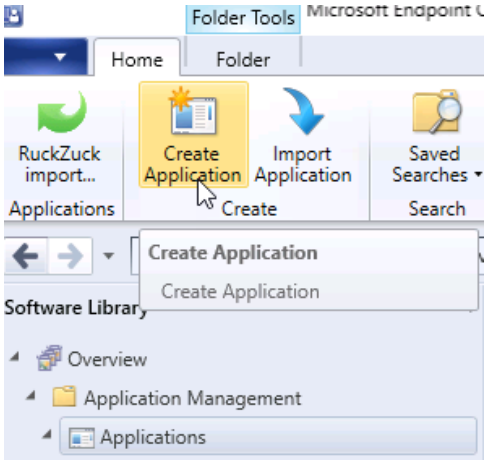


One of 8500+ titles

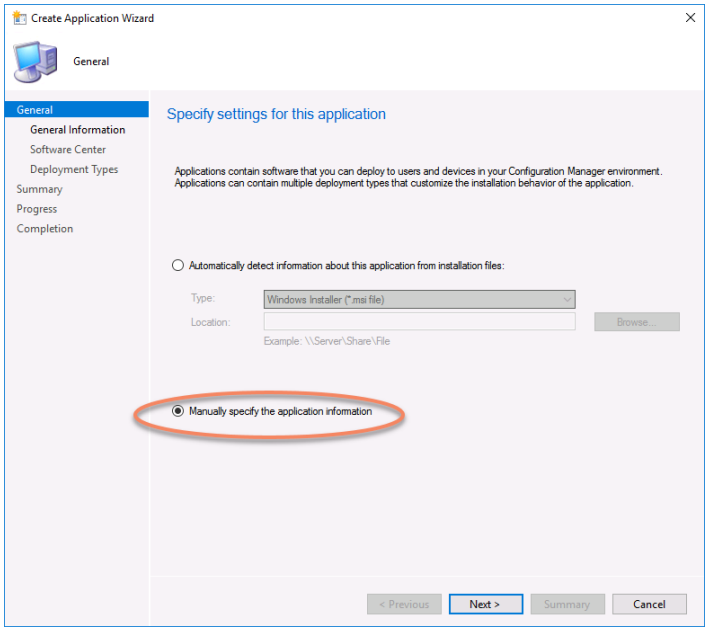
Application installations via Chocolatey are extremely easy to package, install, and update. The following process we use to install FileZilla is the same for any Chocolatey-based application install. The website we visited at the start of this exercise contains a searchable list of all the available applications, a description, and the required string to trigger the installation. That string simply needs to be replaced for the $packageName variable in the script below.

Instructions	Screenshot (if applicable)
1. Logon to CM01 as Administrator	

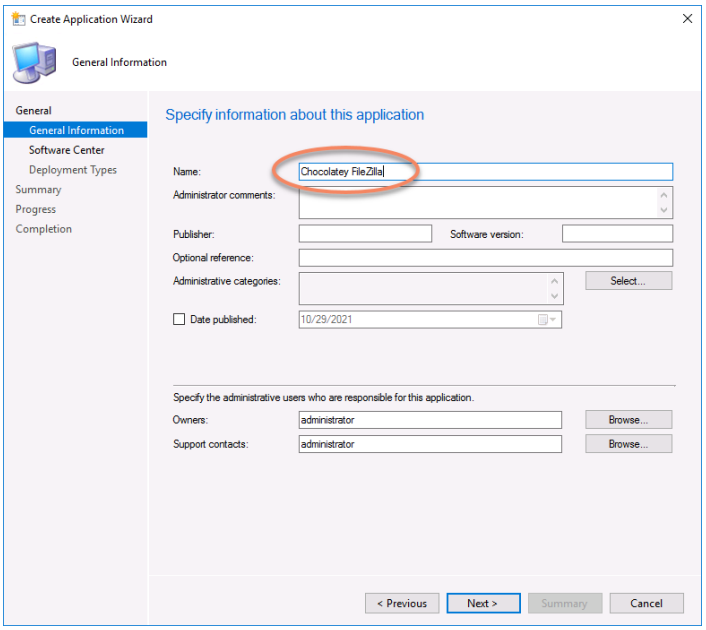
2. In the Software Library workspace, expand Application Management and select Applications. From the ribbon, choose **Create Application**.



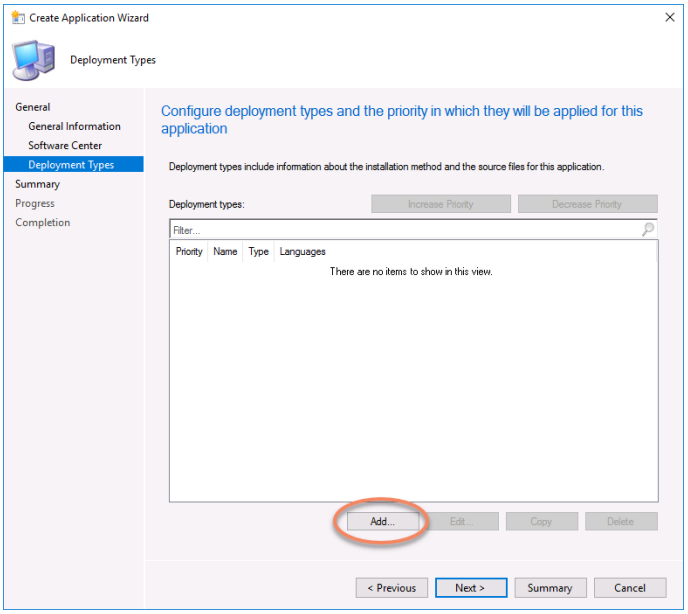
3. In the Create Application Wizard, choose the "Manually specify the application information" radio button and click Next to continue.



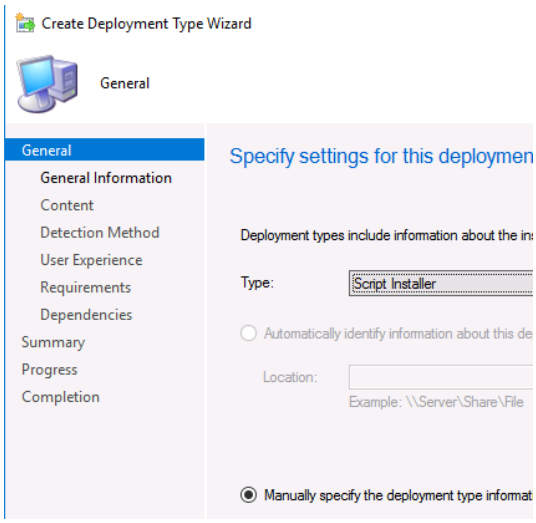
4. Call the application **Chocolatey FileZilla** and leave all the other fields default.



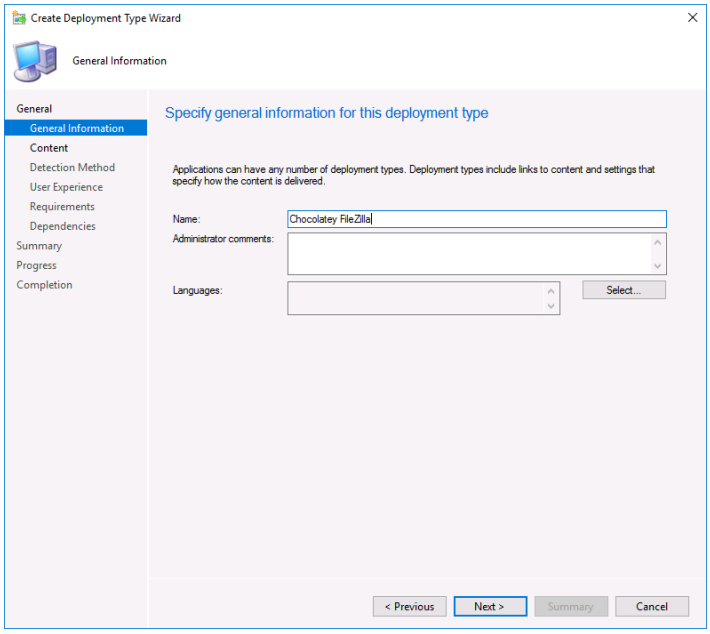
5. Skip the Software Center page again, and click Add on the Deployment Types page



6. In the Create Deployment Type Wizard, choose Script installer and select the bottom radio button to **Manually specify the deployment type information**



7. On the General Information page, call it **Chocolatey FileZilla** again.

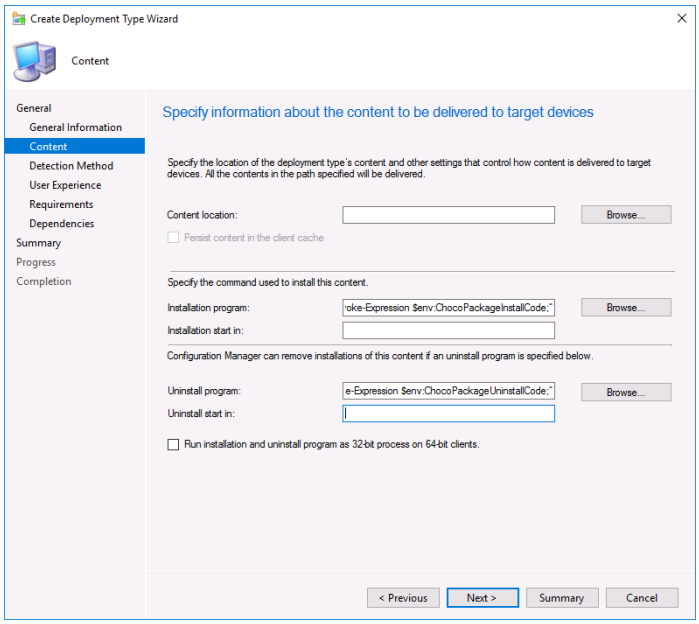


8. On the Content page, use the following code for the Installation Program.

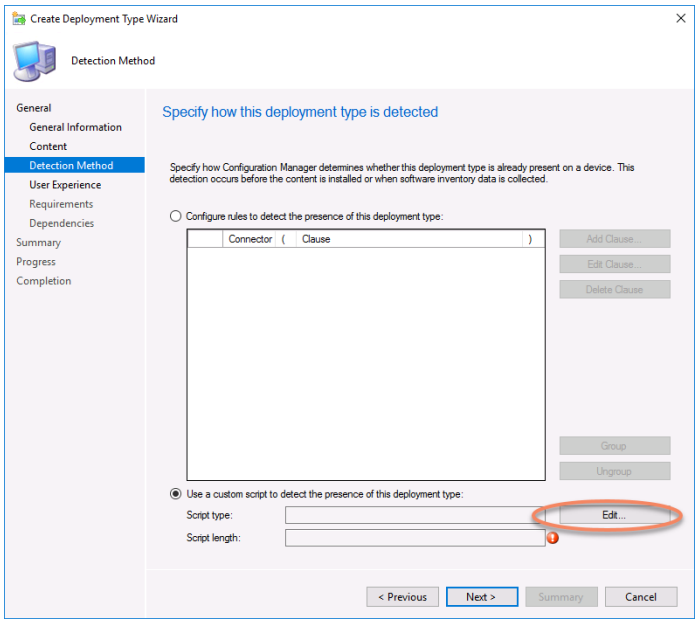
```
powershell.exe -executionpolicy bypass -command “$packageName = ‘filezilla’; Invoke-Expression $env:ChocoPackageInstallCode;”
```

9. For the Uninstall program on the Content page, use:
   ```
   powershell.exe -executionpolicy bypass -command “$packageName = ‘filezilla’; Invoke-Expression $env:ChocoPackageUninstallCode;”
   ```

10. The Content page should now look like this screenshot



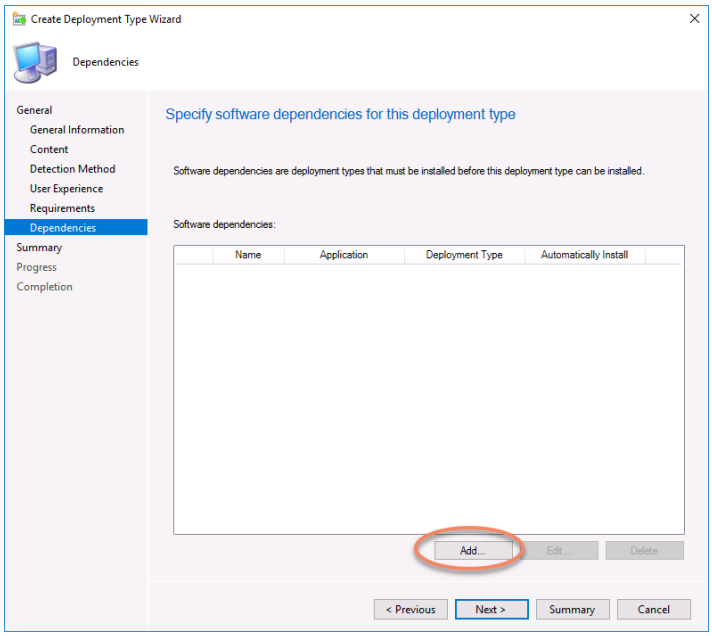
11. On the Detection Method screen, choose the bottom radio option to **Use a custom script to detect the presence of this deployment type** and click **Edit**.



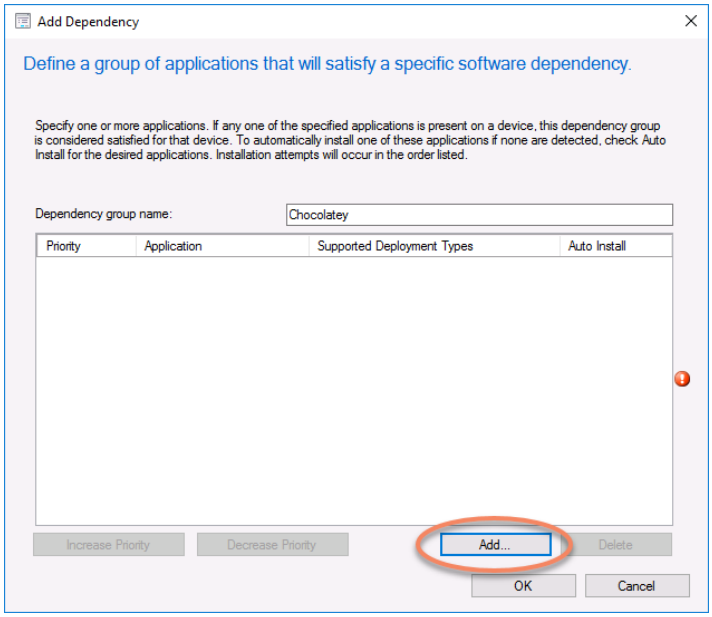
12. In the Script Editor, choose PowerShell and paste the following code:

```
$packageName = "filezilla";

if([System.Environment]::GetEnvironmentVariable
("ChocoPackageDetectionCode")) {
    Invoke-Expression
$env:ChocoPackageDetectionCode;
}
```

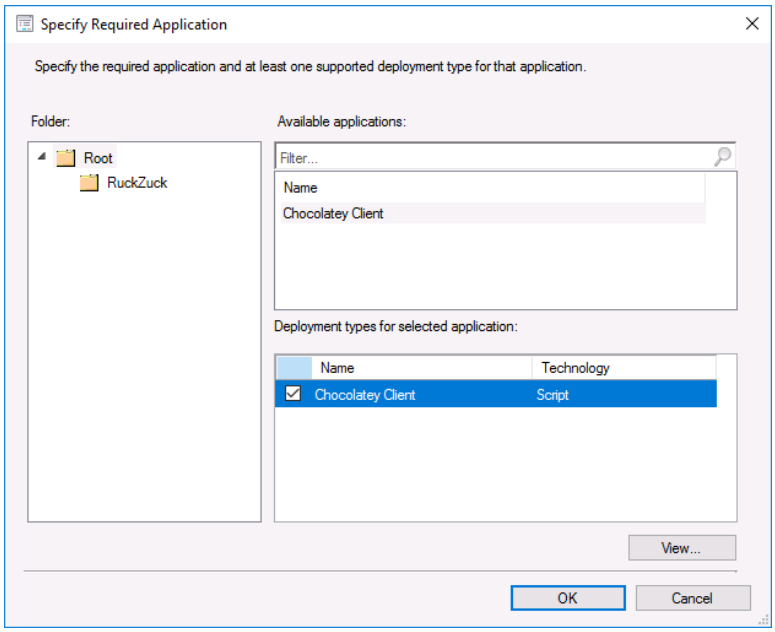
13. Continue clicking Next until you reach the Dependencies page. Click Add.



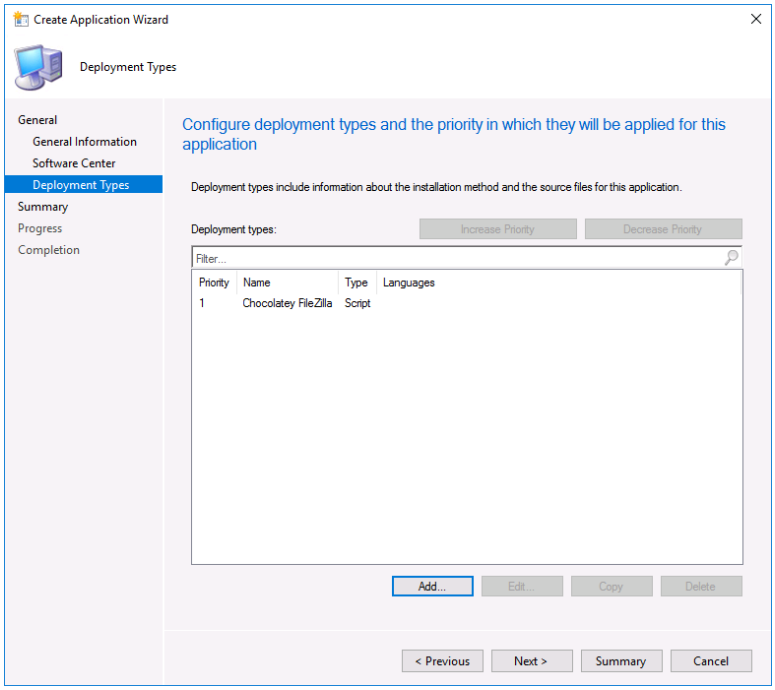
14. On the Add Dependency page, call the Dependency group name **Chocolatey** and click the Add button.



15. Choose the Chocolatey Client as a Required Application, then click OK.



16. Click OK a couple of times, and Next, Next, Close to complete the Deployment Type Wizard. Then Next, Next, Close again to complete the Create Application Wizard



17. Now we can go ahead and deploy these applications as Available to the All Systems Collection.

Chapter 5 – Deploying Windows 11

Scenario

Deploying operating systems to enterprise workstations can be a mind-blowingly complex procedure. It is not uncommon to see more than 300 steps defined to take a brand new device and turn it into a functioning workstation on the corporate network.

The Task Sequence engine we use in ConfigMgr has been virtually unchanged in more than 20 years. It has seen many improvements and advancements in that time, however its core operation has withstood the test of time.

Natively, the ConfigMgr Create Task Sequence Wizard is very basic, and requires significant modifications to allow any device versatility or automation, yes it still works out of box to perform very basic imaging. Though used in many places, it is not common to see a well-designed OSD solution in ConfigMgr with just the built-in task sequences.

Before Configuration Manager there was the Microsoft Deployment Toolkit, which itself evolved from Business Desktop Deployment. MDT is still supported and used today as a standalone product for imaging in small environments, creating reference images, and as a powerful integration component with Configuration Manager.

When MDT is integrated with ConfigMgr, hundreds of automation options instantly become available in your task sequences. Though considerably more complex than a deployment sequence created with the default CM wizard, an MDT-integrated task sequence can handle multiple platforms out of the box and ultimately reduces both the number of task sequences you need in production, as well as the amount of post-production edits you need to make to the sequence do what you need it to.

In the following exercises we will create all the workstation task sequences required for virtually any size of organization, three sequences in total. The first task sequence can be considered the main sequence, and is responsible for Bare Metal deployments, OS Refreshes, and Replacement workstations. Our second task sequence is designed to be ran on workstations being retired, where the third is built to perform the Windows In-Place Upgrade procedure.

Topic – Upgrading to ConfigMgr Current Branch

Configuration Manager, like Windows, follows a rapid release cadence tha regularly introduces new features and support options. In order to deploy and manage Windows 11 workstations, ConfigMgr must first be upgraded to a supported version. Typically, there is a new build of CM released the month before a new build of Windows, and for Windows 11 RTM support that's build 2107.

The in-console update process is highly automated, and will report on any errors or warnings identified during the installation. The process is, however, not a complete upgrade check and failure to validate the requirements can have drastic and immediate conseqeuences to the stability of your environment.

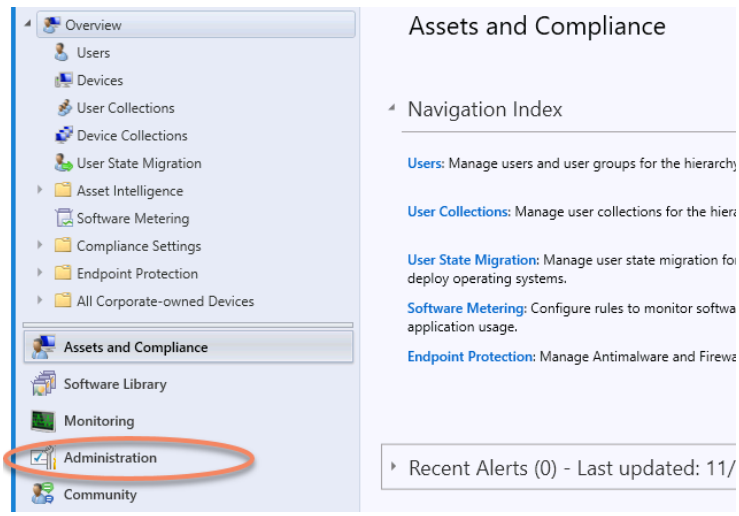
IMPORTANT NOTE: ***THESE COMMANDS HAVE ALREADY EXECUTED IN YOUR DOMAIN. YOU DO NOT NEED TO OPEN or DO ANYTHING FOR THIS EXERCISE. JUST REVIEW THE EXERCISE 2 IN THIS LABGUIDE. DO NOT PERFORM ANY OF THE STEPS BELOW. WE ARE SHOWING THEM JUST TO SHOW HOW EXERCISE 2 WAS SETUP.***

Instructions	Screenshot (if applicable)
1. Logon to the Site Server as Administrator	
2. On the taskbar, click **Internet Explorer.**	

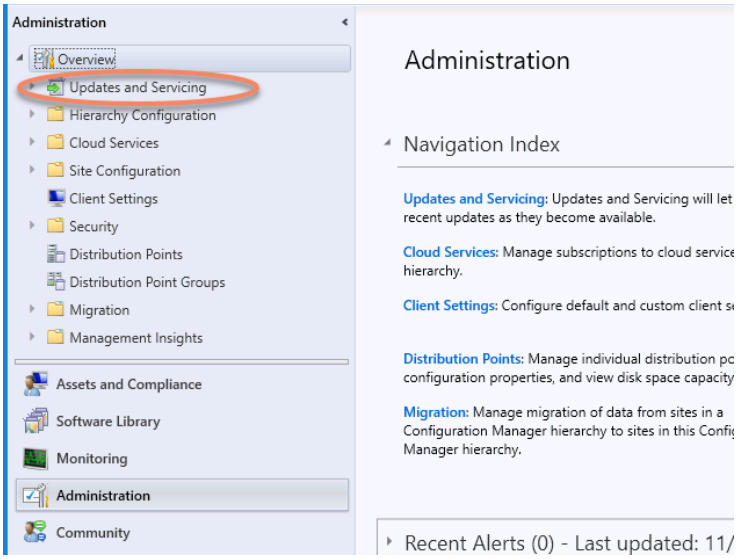
3. Navigate to https://docs.microsoft.com/en-us/mem/configmgr/core/servers/manage/checklist-for-installing-update-2107	**Checklist for installing update 2107 for Configuration Manager** • 14 minutes to read • M *Applies to: Configuration Manager (current branch)* When you use the current branch of Configuration Manager, you can install the in-console update for version 2107 to update your hierarchy from a previous version. To get the update for version 2107, you must use a service connection point at the top-level site of your hierarchy. This site system role can be in online or offline mode. To download the update when your service connection point is offline, use the service connection tool. After your hierarchy downloads the update package from Microsoft, find it in the console. In the **Administration** workspace, select the **Updates and Servicing** node.
4. Review Microsoft .NET	4.6.2 now required
5. Review Windows ADK	ADK and WinPE for Windows 11 is required to deploy Win11 a. https://aka.ms/adk b. Download adksetup.exe to E:\install c. Download adkwinpesetup.exe to E:\install d. From the Run menu, type **appwiz.cpl** and hit enter e. In Add/Remove Programs, uninstall the Windows Assessment and Deployment Toolkit – Windows 10 f. In File Explorer, browse to E:\install g. Install ADKSetup.exe, selecting the following components when prompted during the wizard: i. Deployment Tools ii. User State Migration Tool h. Install ADKWinPESetup.exe using the default installation options
6. Review SQL NCLI	Native CLI must support TLS 1.2 (Min version 11.*.7001.0)
7. Review Hierarchy Status	Shouldn't be any problems before we upgrade
8. Review Windows Updates	It's a server, it should be fully patched
9. Disable Site Maintenance	This should be done in production to avoid scheduled tasks from executing during the upgrade process
10. Disable Anti-Virus/etc.	Anti-Virus and threat servers can impact or interrupt the upgrade
11. Backup Site Database	This is the rollback strategy. Restoring a site from backup is fast
12. Back up OSDInjection.xml	Important for Nomad customers and custom OSD automation

13. Validate Client Upgrade	Ensure Automatic Client Upgrade settings are configured
14. Run Pre-req check	Monitor DMPDownloader.log and ConfigMgrPrereq.log
15. Install Update	Monitor CMUpdate.log
16. Enable Site Maintenance	After upgrade, turn on site maintenance tasks in production

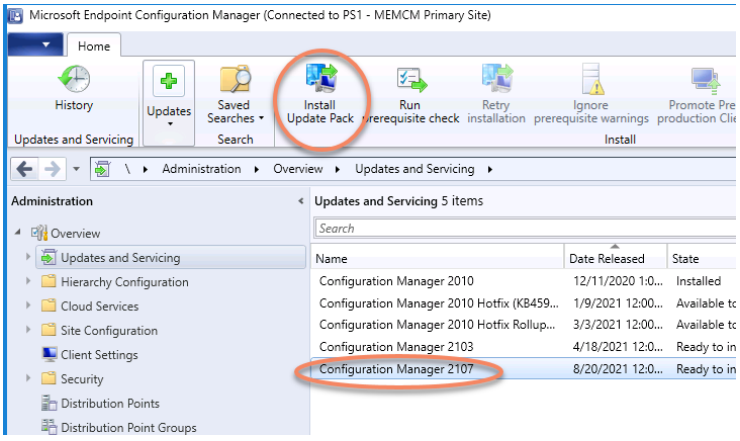
Topic – In-Console Upgrade Procedure

Instructions	Screenshot (if applicable)
1. Logon to the Site Server and launch the CM Console	
2. In the bottom left, click on the Administration workspace	

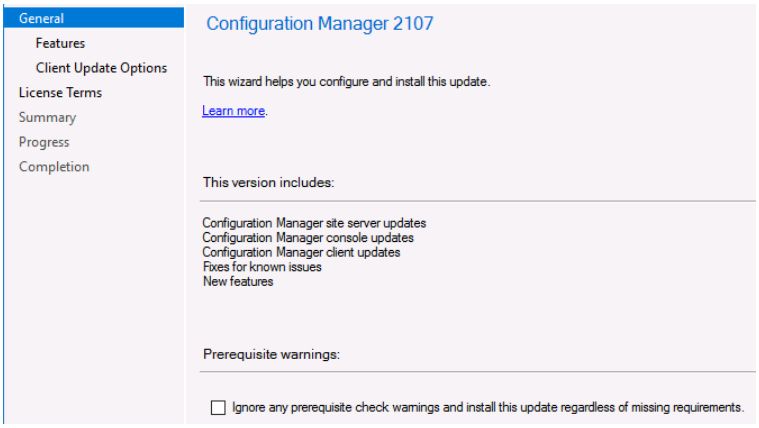
3. In the Navigation Pane, click Updates and Servicing



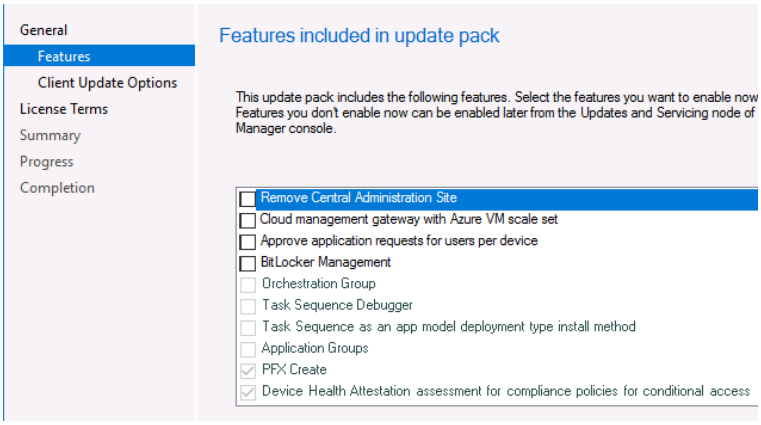
4. In the Results pane, select **Configuation Manager 2107** and choose **Install Update Pack** from the ribbon.



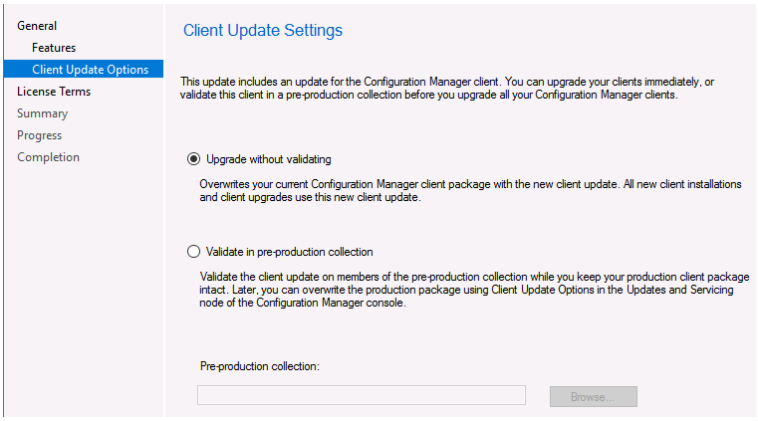
5. On the General page, ensure the “Ignore any prerequisite check warnings and install this update regardless of missing requirements.” box is **CHECKED**



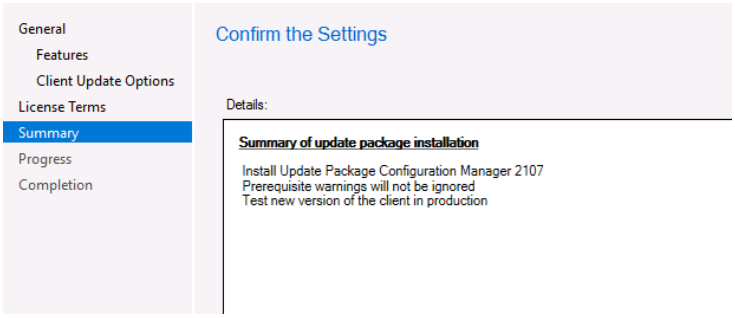
6. On the Features page, ensure all of the optional features are **unchecked**



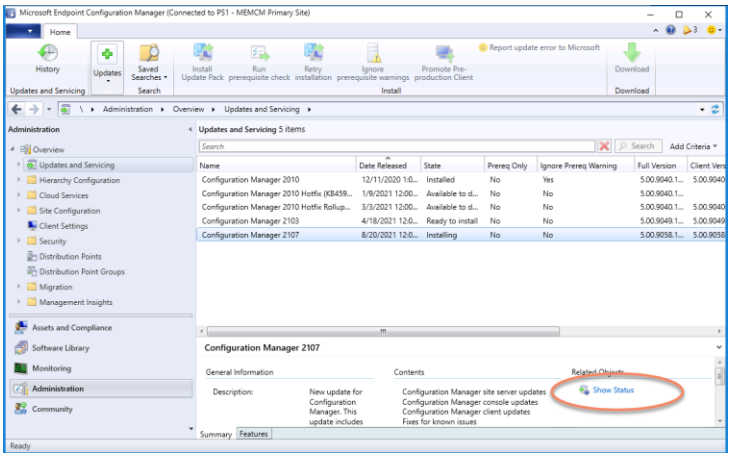
7. For Client Update opens, leave the default selection of “Upgrade without validating.” This puts the environment in a pre-production state and allows the new client to be tested early (before the default “Install client” package is updated)



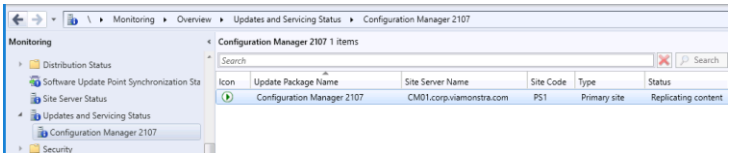
8. Accept the license terms if you agree with them, then Next, Next, Close



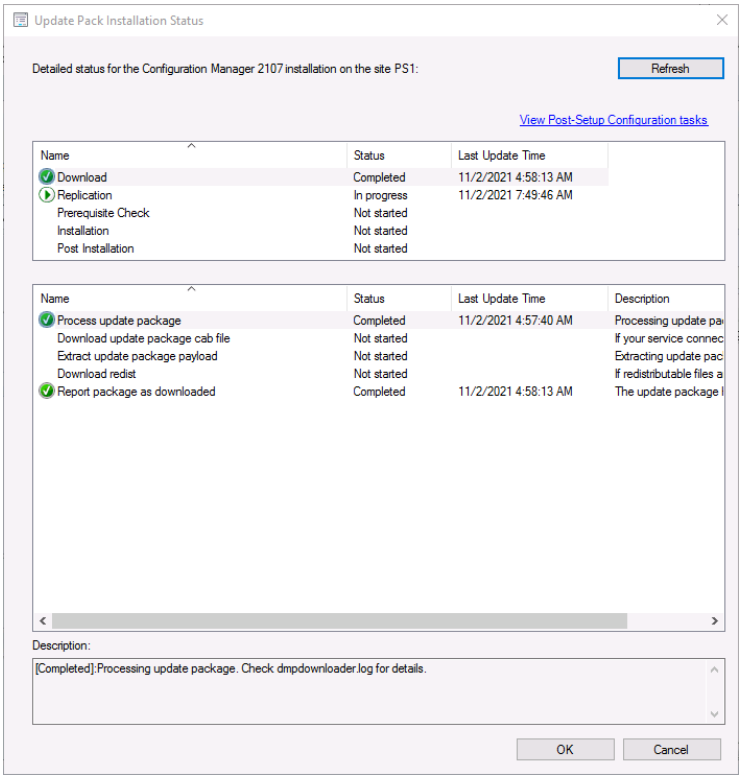
9. Next, in the Details pane, click the Show Status link



10. Though not immediately visible, after some time the installation status is viewable in the Monitoring workspace.

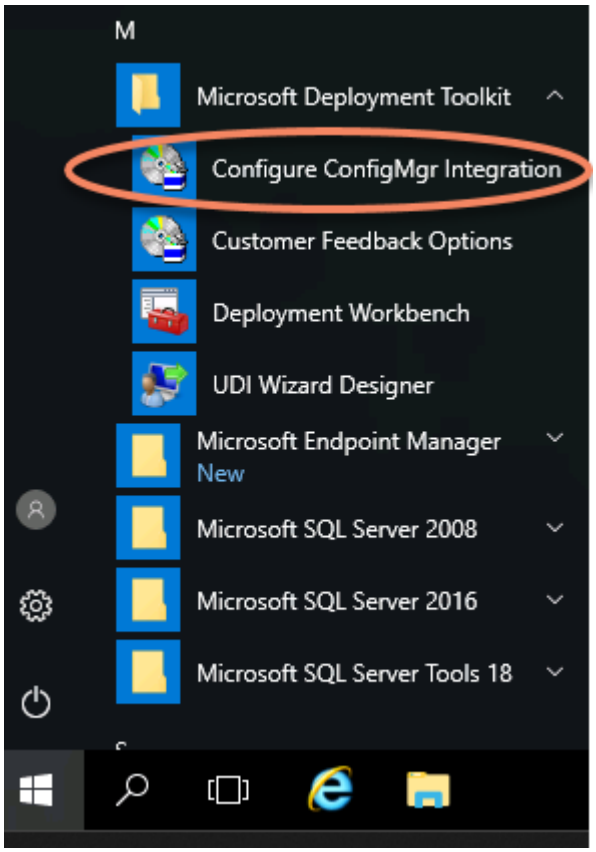


11. Selecting Show Status from the ribbon will provide a nice status window of the install.

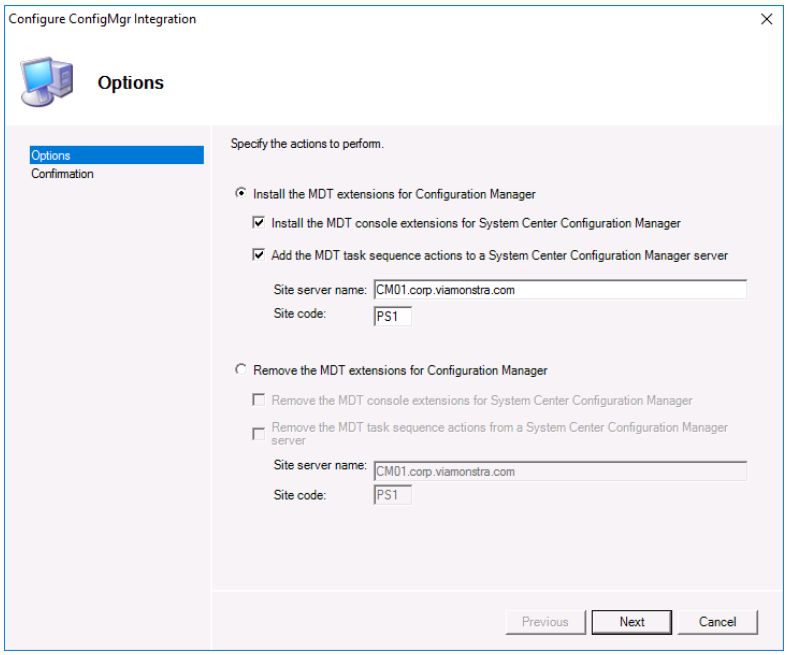


12. DMPDownloader, CMUpdate, and ConfigMgrPrereq log files are also helpful in monitoring the process.

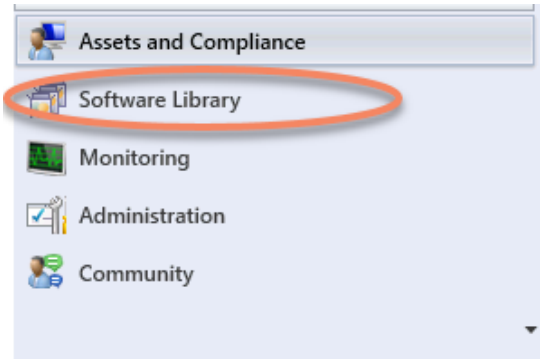
Exercise 1 – Build the MDT-Integrated Task Sequence

Instructions	Screenshot (if applicable)
13. Logon to the Site Server as Administrator and click the Start button.	
14. Click the “Microsoft Deployment Toolkit” folder and choose **Configure ConfigMgr Integration**	

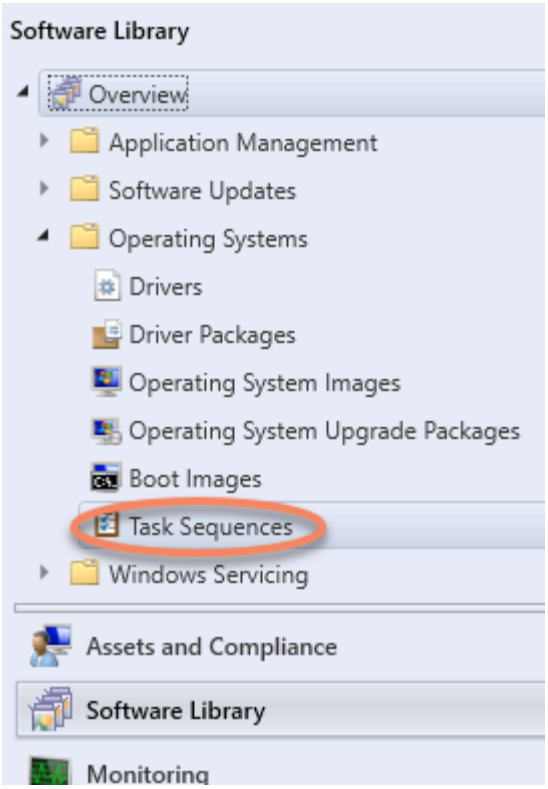
15. After accepting the UAC prompt, a wizard appears. Click Next, Next, Close.



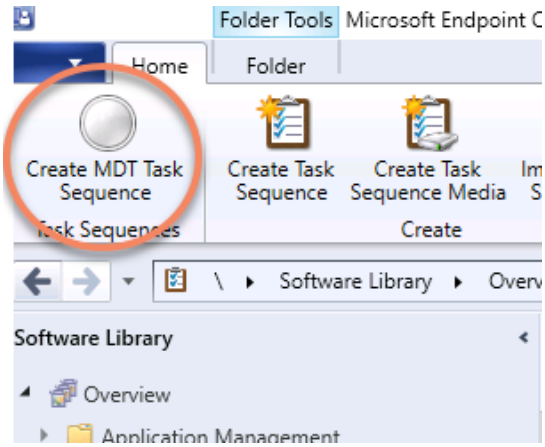
16. Launch the CM Console. In the bottom right, select the Software Library workspace.



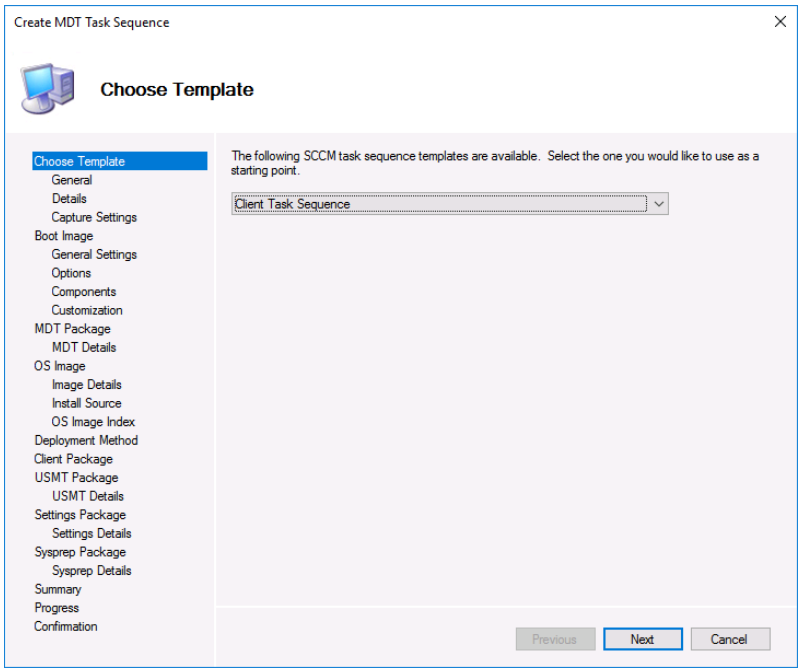
17. In the Software Library, expand the “Operating Systems” node and select **Task Sequences.**



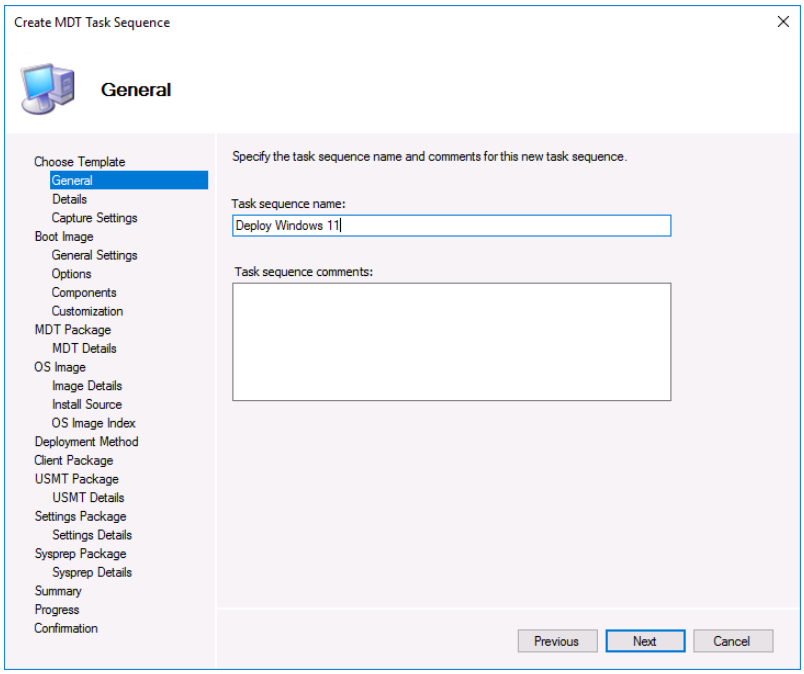
18. At the top left of the ribbon, click the new **Create MDT Task Sequence** button.



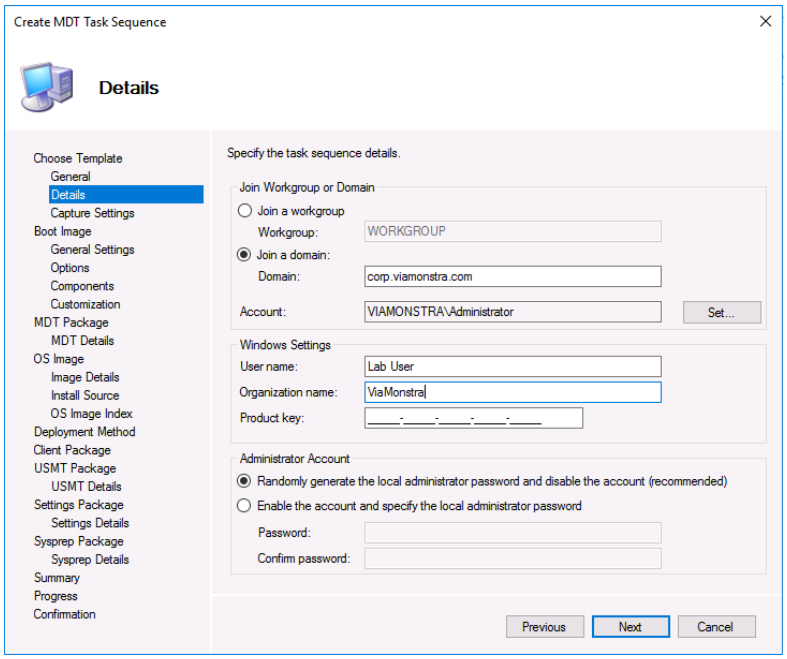
19. In the Create MDT Task Sequence wizard, keep the default “Client Task Sequence” option selected on the Choose Template page and click **Next**



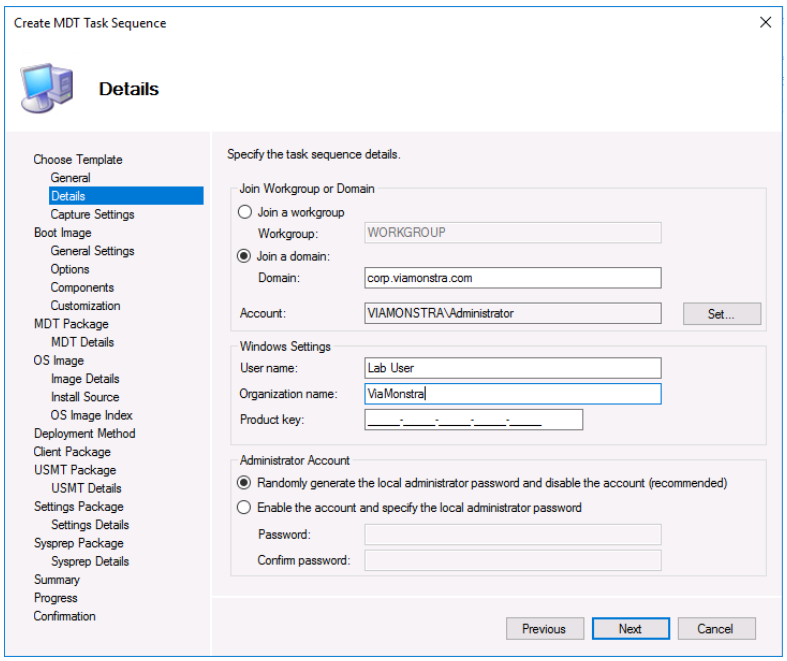
20. On the General page, specify a name for the sequence and click Next



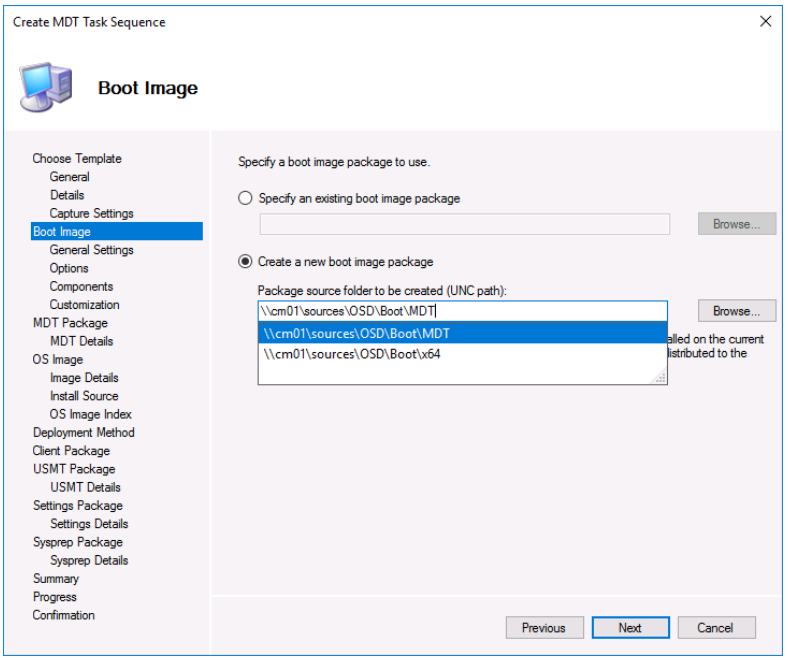
21. The Details page is where Domain Join information is specified. Join corp.viamonstra.com using the domain administrator account, and populate a user name and organization in the respective fields in order to proceed



22. This is not a sysprep and capture sequence, and would never be used for this purpose

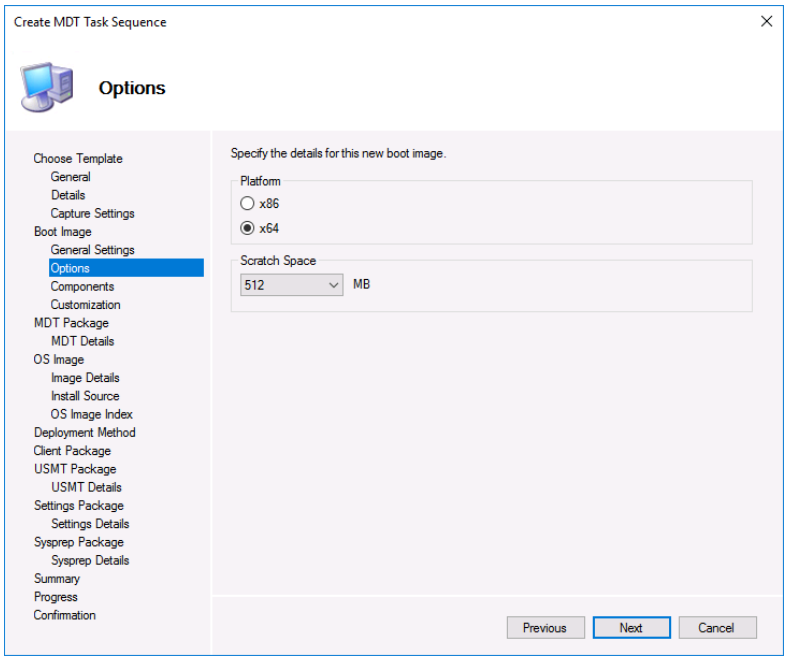


23. An MDT boot image contains more files than the default one, so we need to create a new package. Place it in \\CM01\sources\osd\boot\mdt\

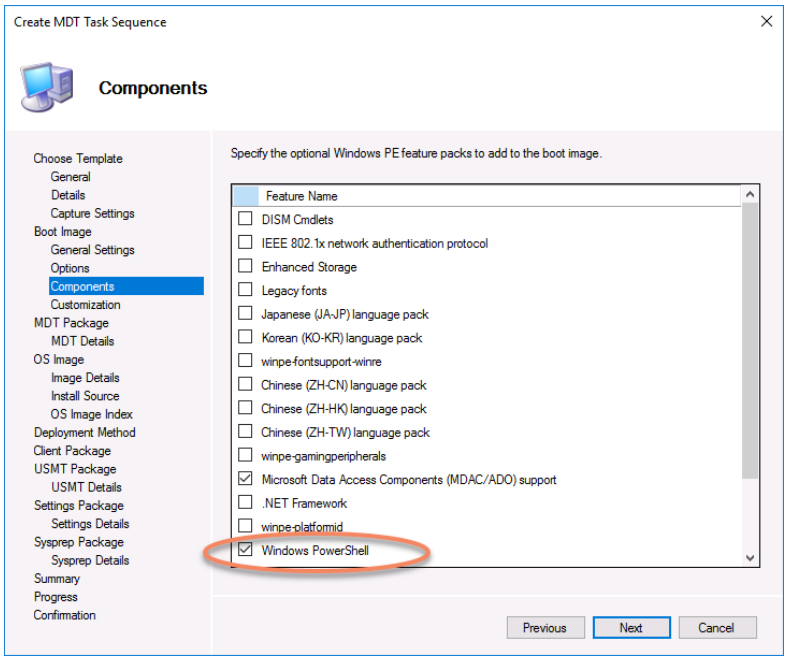


24. Call it **MDT Boot Image** and click Next.

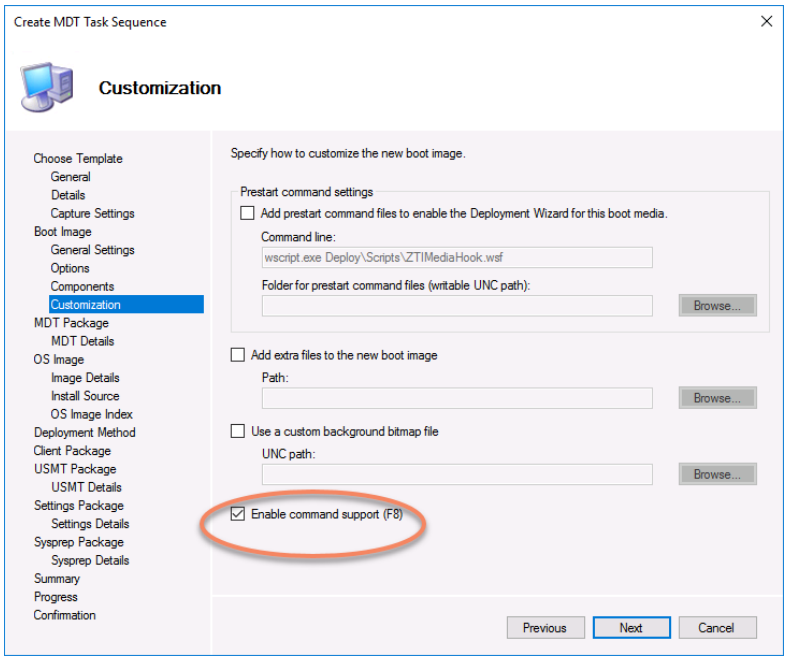
25. On the Options page, choose **x64** for the Platform, and **512** for the Scratch Space.



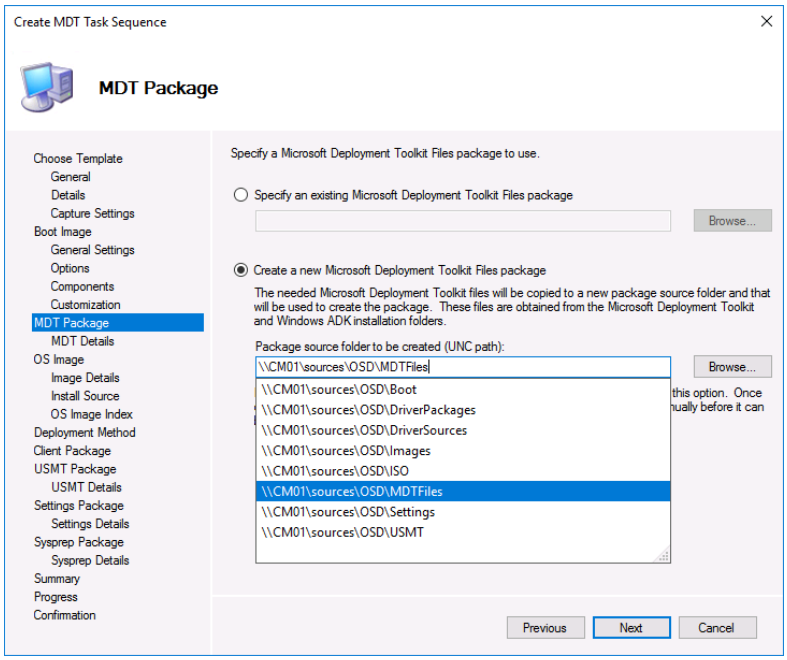
26. We need to add **PowerShell** to the boot image, as this will assist with UEFI and Driver detection later on. This is done at the Components page



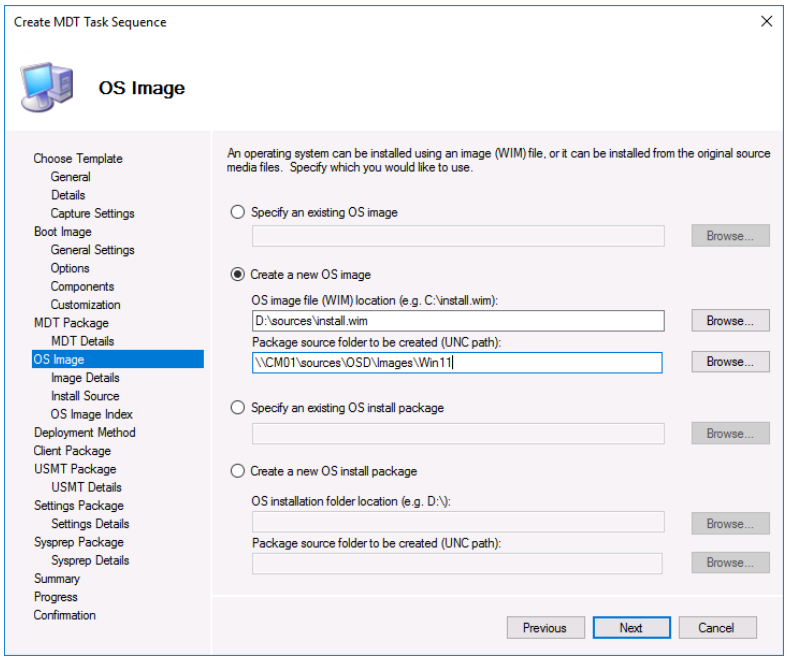
27. Ensure “Enable command support (F8)” is **checked** on the Customization page



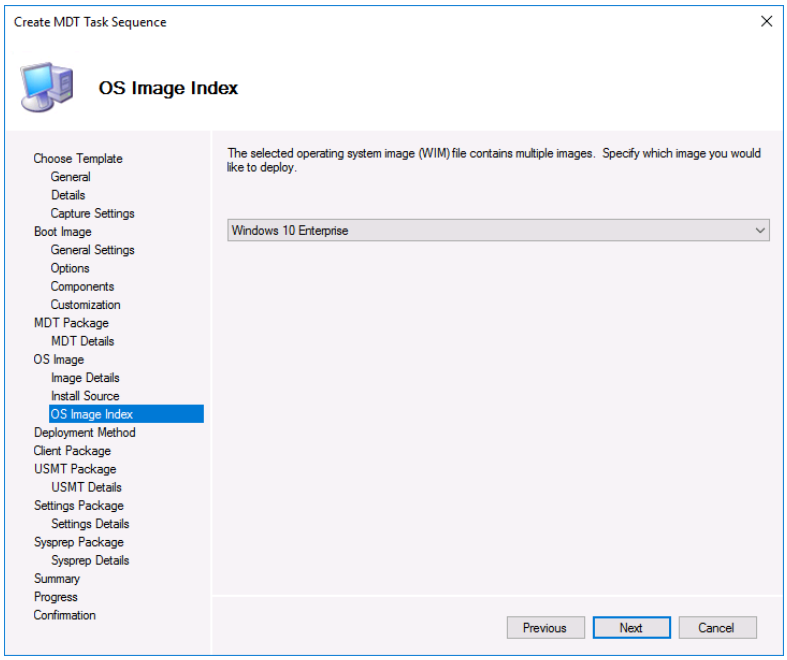
28. Create a new MDT Package, storing the files in \\CM01\sources\OSD\MDTFiles. On the next screen, we’ll call it **MDT Files**



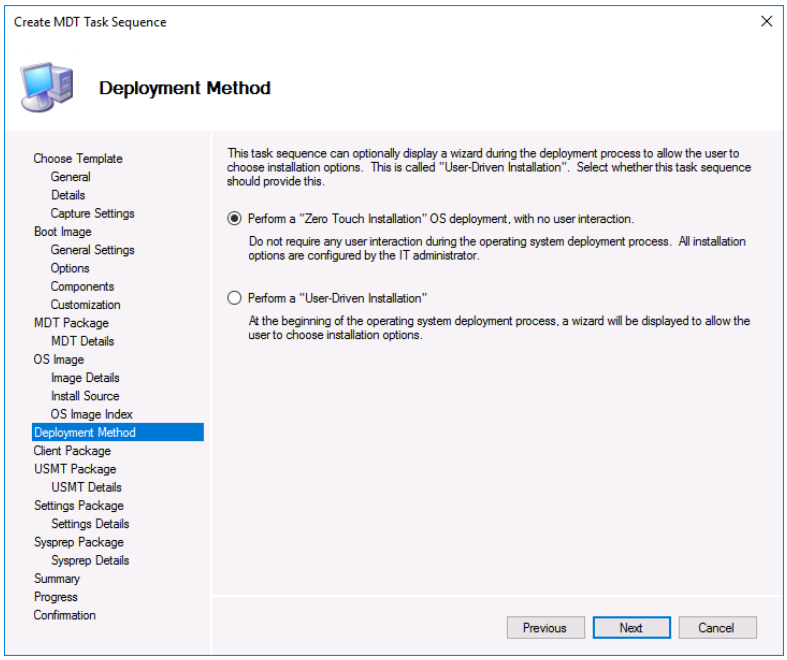
29. On the OS Image page, we're going to choose to **Create a new OS Image**, specifying \Sources\install.wim from the ISO file we mounted earlier, and creating the package in \\CM01\sources\OSD\Images\WIn11. Call the OS Image **Windows 11 Enterprise RTM** on the Image Details screen.



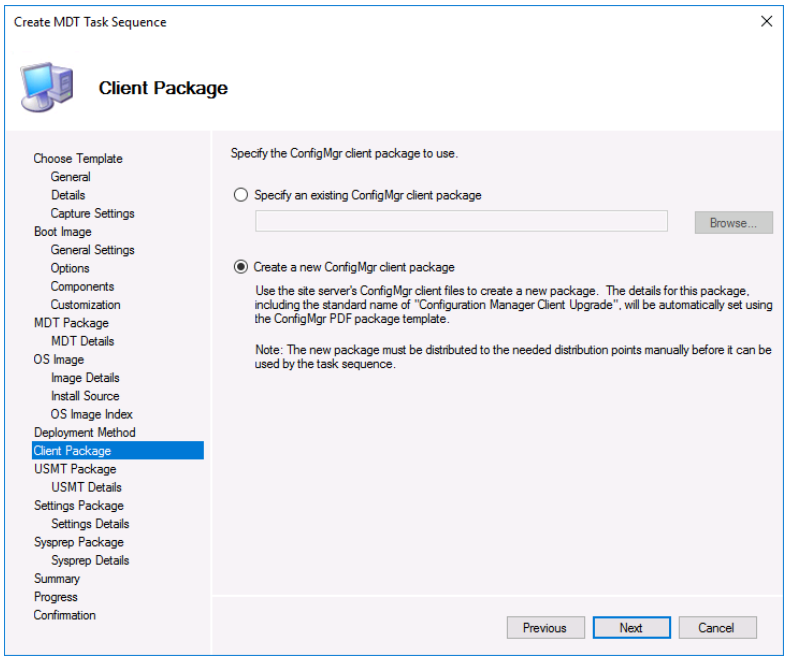
30. On the OS Image Index page, choose Enteprise (or whatever your license allows)



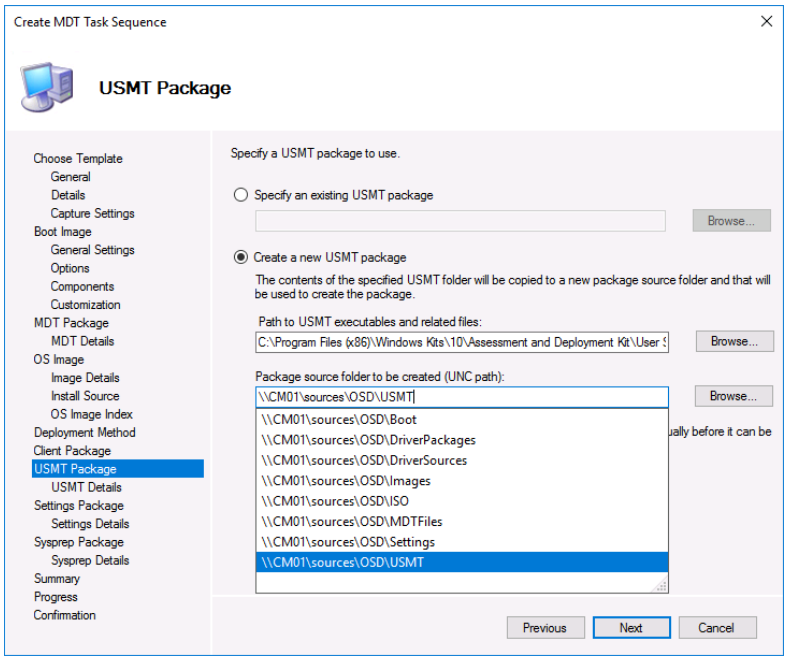
31. On the Deployment Method page, choose the default option for a Zero-touch installation



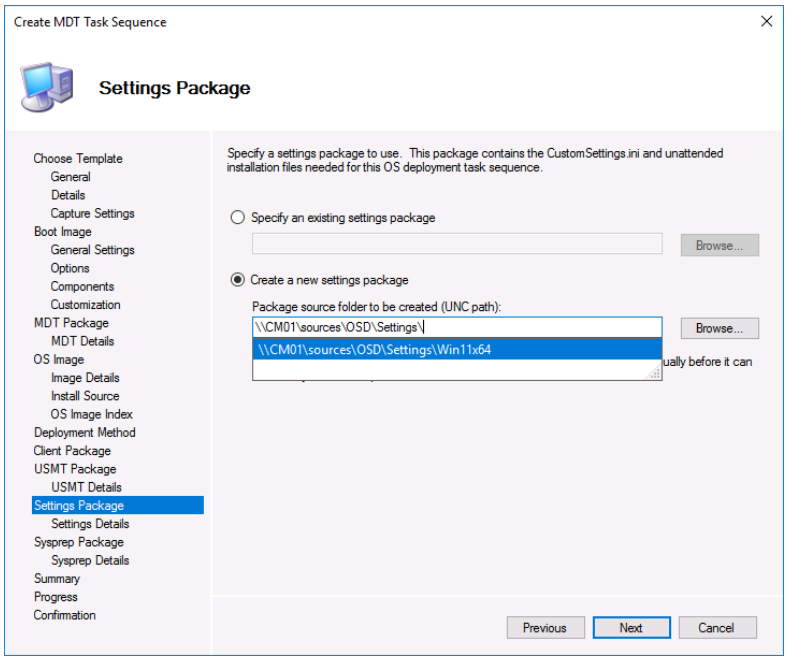
32. The built-in Client Installation Package is locked in a Read-Only state, preventing us from making many changes. To alleviate issues down the road, we typically create a separate one for OS Deployment scenarios



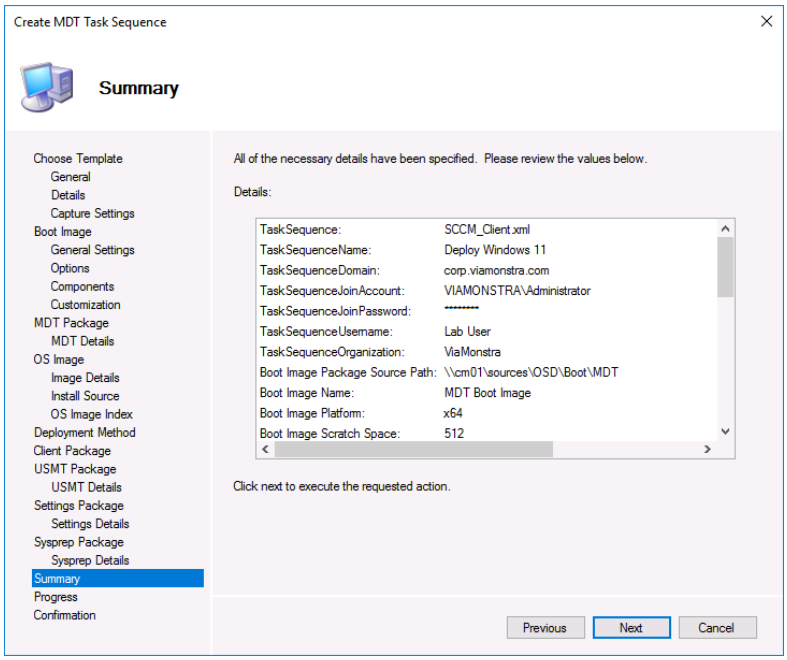
33. The USMT package is created from the ADK installation folder, and should automatically populate. We just need to specify the target location of the new package, \\CM01\sources\OSD\USMT . Call it **USMT** on the USMT Details page



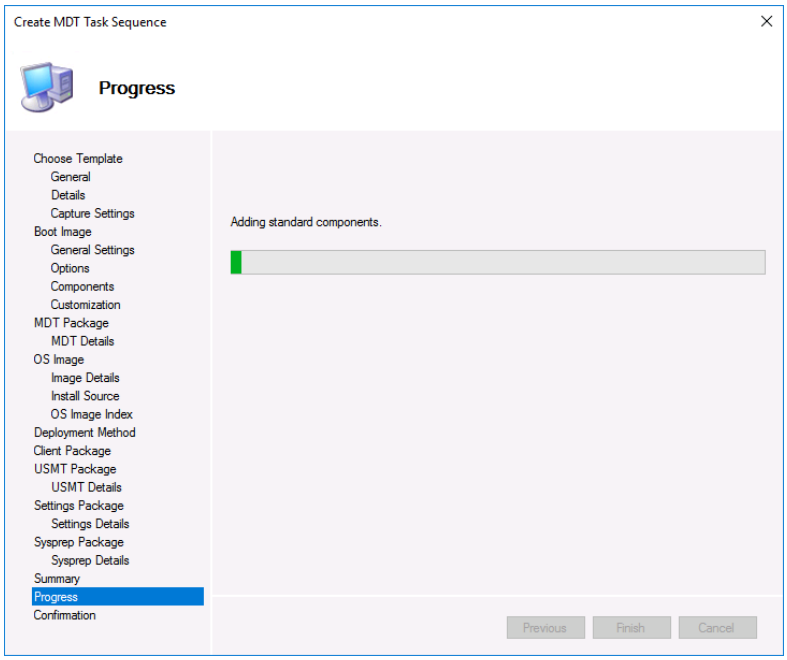
34. The Settings package is where we place custom scripts and ini files to modify the mini-setup process. We need to create a new one, and point it to \\CM01\sources\OSD\Settings\Win11x64 . On the Settings Details page, call the package **Win11 x64 Settings**



35. Click Next down to the Summary screen, and review the information specified during the wizard.



36. Click Next to start the task sequence creation process. During this stage, MDT has to mount the boot image to the Temp directory and extract files. You may receive a UAC prompt after starting the process to permit this action.

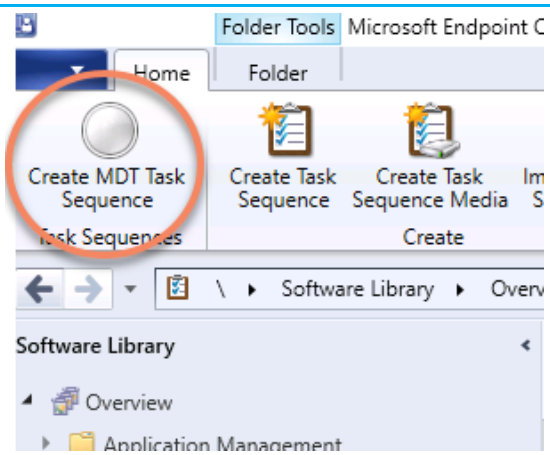


Exercise 2 – Build the Backup Task Sequence

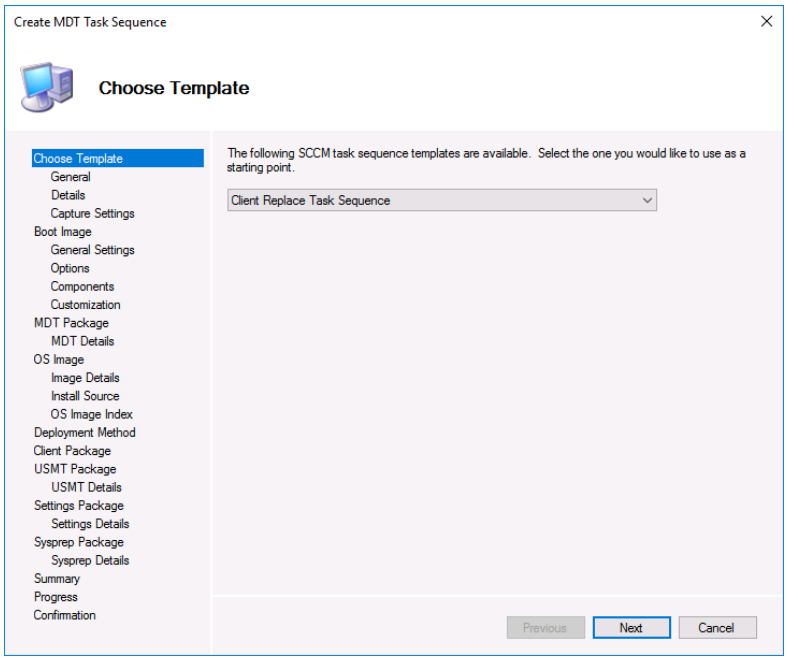
In this exercise, we will create a task sequence that's the first technical step in a user's workstation replacement. It's often made available to the end user in Software Center for them to trigger, starting the process.

When executed, this sequence will connect to the State Migration Point and perform a backup of the device to the SMP share. The contents of the backup are fully configurable and utilize .XML files for customization. By default, it'll back up pretty much everything, and for organizations that are conscious of intellectual property loss, the drives can be automatically scrubbed after a successful backup.

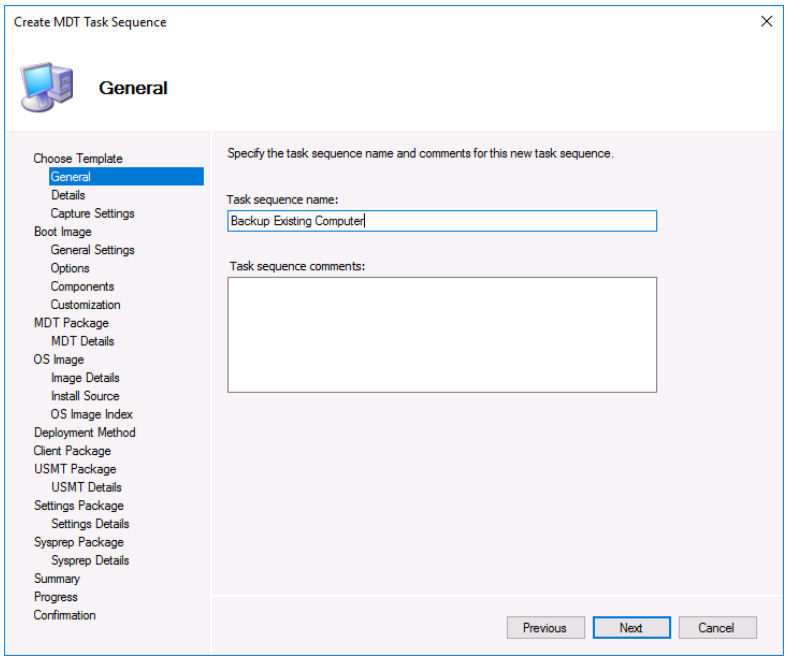
In ConfigMgr, when we import the user's new workstation, we specify the old machine as the Source Computer. This action will set a flag on the new computer's resource resource record, and when it is imaged and reaches the State Restore section, it will search the SMP for the user's original computername and restore the data that was backed up.

Instructions	Screenshot (if applicable)
1. With the Task Sequences node selected, at the top left of the ribbon, click the new **Create MDT Task Sequence** button.	

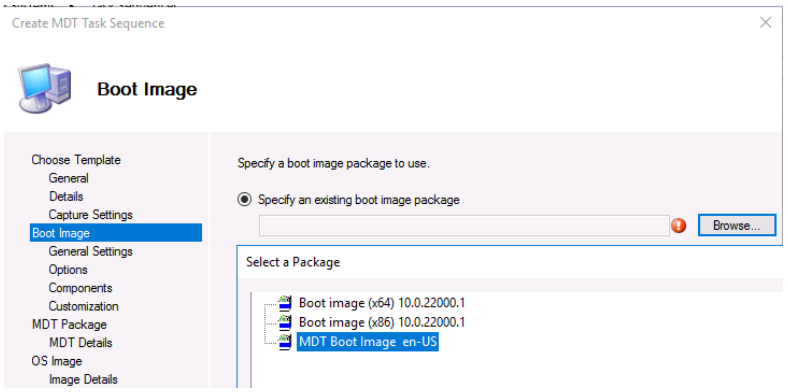
2. On the Choose Template screen, click the drop-down and select **Client Replace Task Sequence**



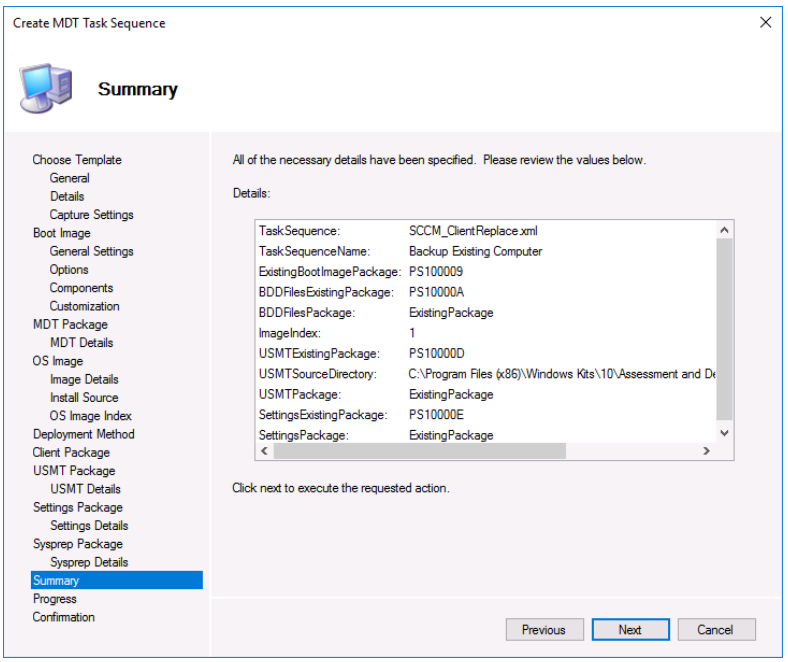
3. On the General page, call this sequence **Backup Existing Computer**



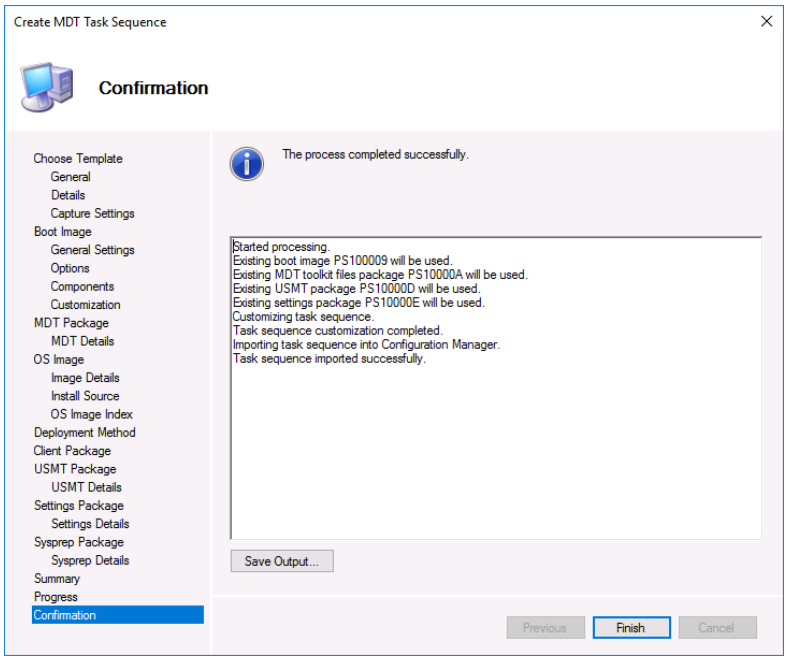
4. For each remaining step, choose the appropriate packages that we created during the last exercise.



5. After selecting all the packages, the Summary screen will show the defined options



6. Click Next to create the sequence, and Finish when done



Exercise 3 – Build the IPU Task Sequence

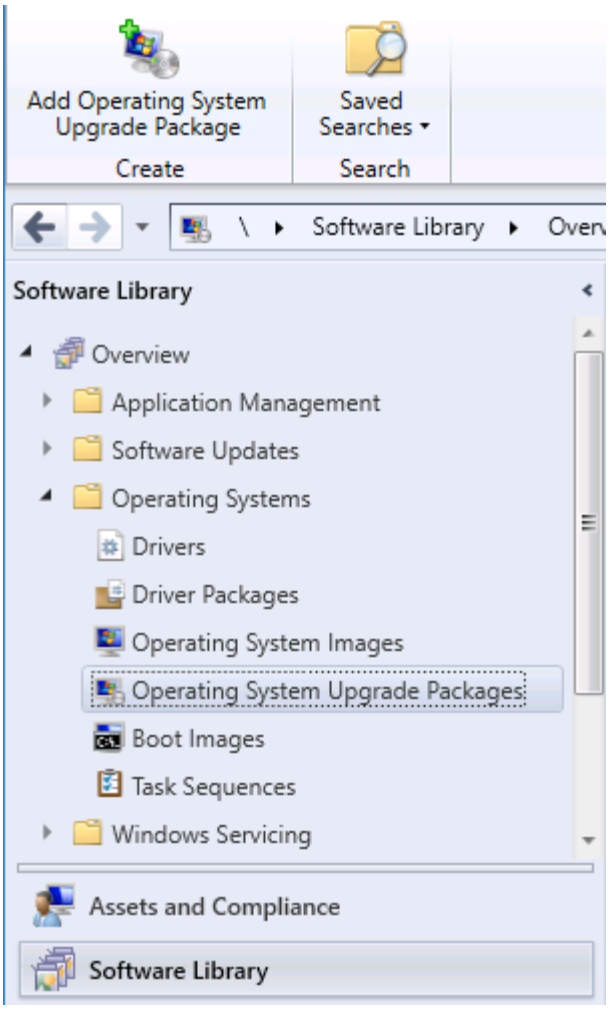
One of the most welcomed advancements of Windows 10 and Server 2016 was the supported, and functional, in-place operating system upgrade. Contrary to the previous 30+ years of Microsoft operating systems, a full wipe and reimage is no longer the best way to upgrade an OS. The IPU sequence is the Task Sequence method of installing a Feature Update.

The Windows 11 In-Place Upgrade task sequence will typically perform the following high-level tasks:

- Upgrade the BIOS/UEFI firmware to the latest version
- Query for, download, and prepare installation of any required device drivers
- Perform pre-flight checks to proactively identify any issues that would affect the upgrade
- Use setup.exe from the Windows 11 installation media to upgrade the operating system

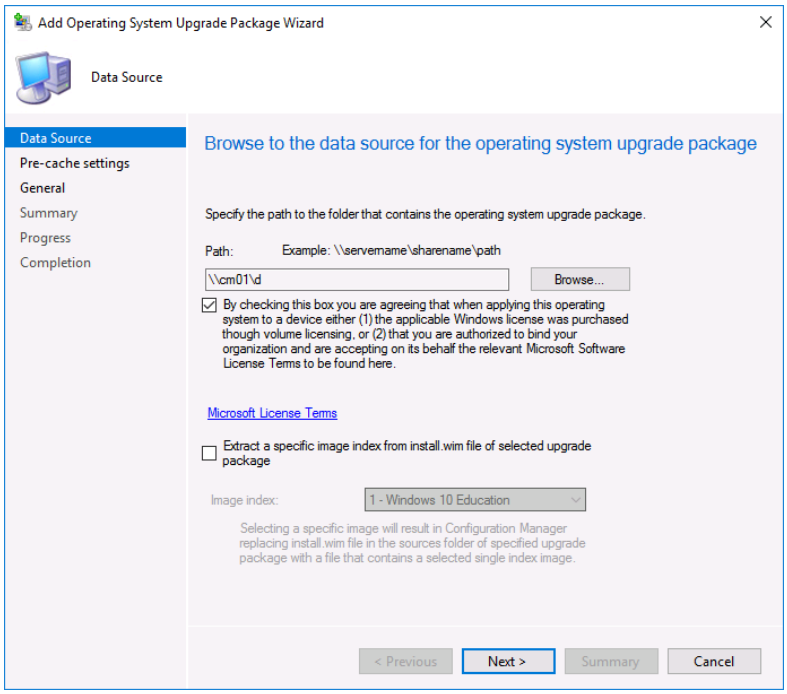
Instructions	Screenshot (if applicable)

1. In the Operating Systems node of the Software Library workspace, select **Operating System Upgrade Packages**. In the ribbon, click **Add Operating System Upgrade Package**

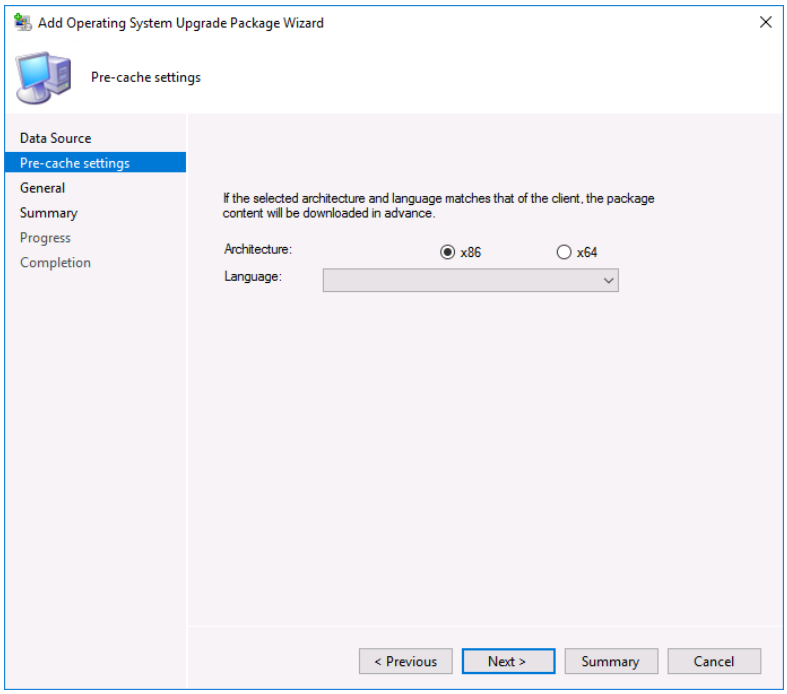


2. In the wizard, click Browse and navigate to \\CM01\D\, where the Windows 11 ISO was previously mounted.

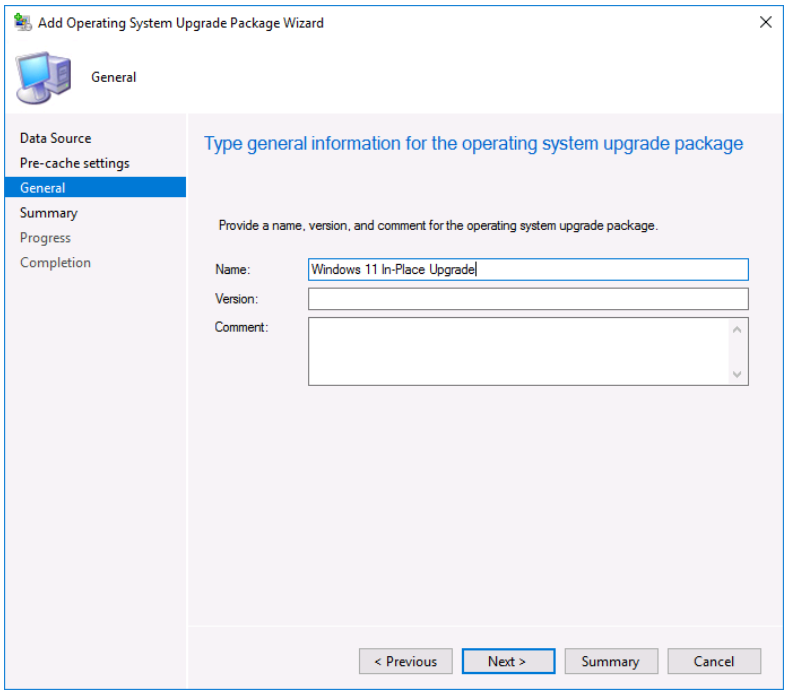
   NOTE: If D: does not exist or is not shared (or both), open File Explorer without closing the wizard, and mount the ISO from E:\sources\osd\iso folder, then share as "D".



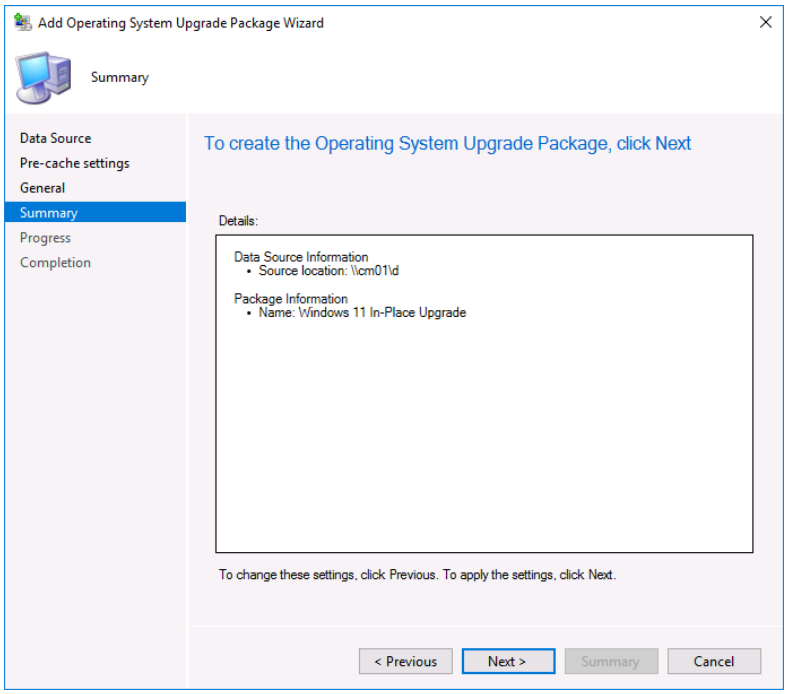
3. Skip the Pre-cache settings page, as not everyone will receive this upgrade.



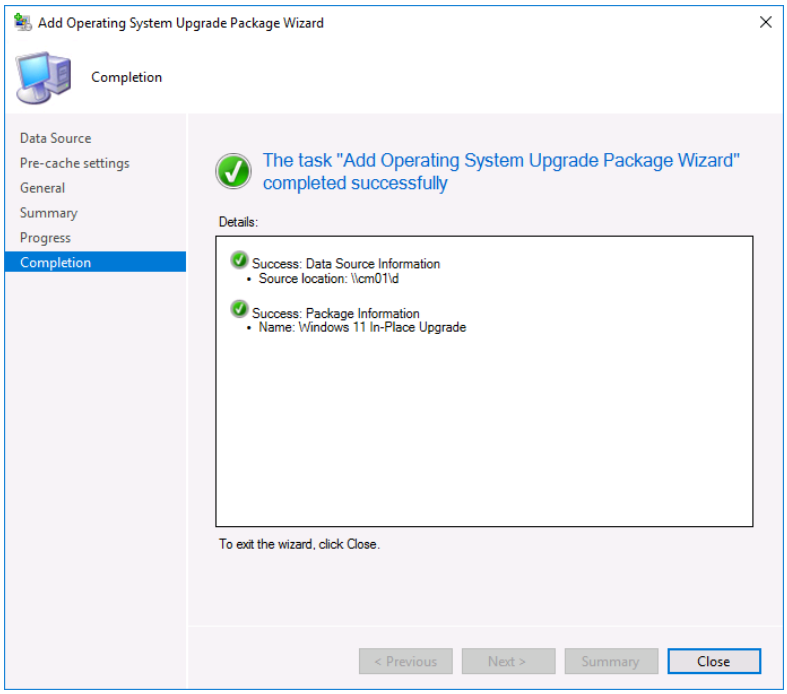
4. On the General page, call this **Windows 11 In-Place Upgrade**



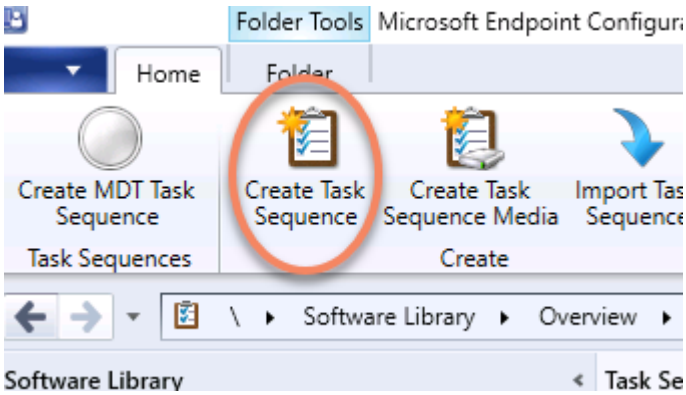
5. On the Summary page, verify the proper information has been populated and click next to create the upgrade package



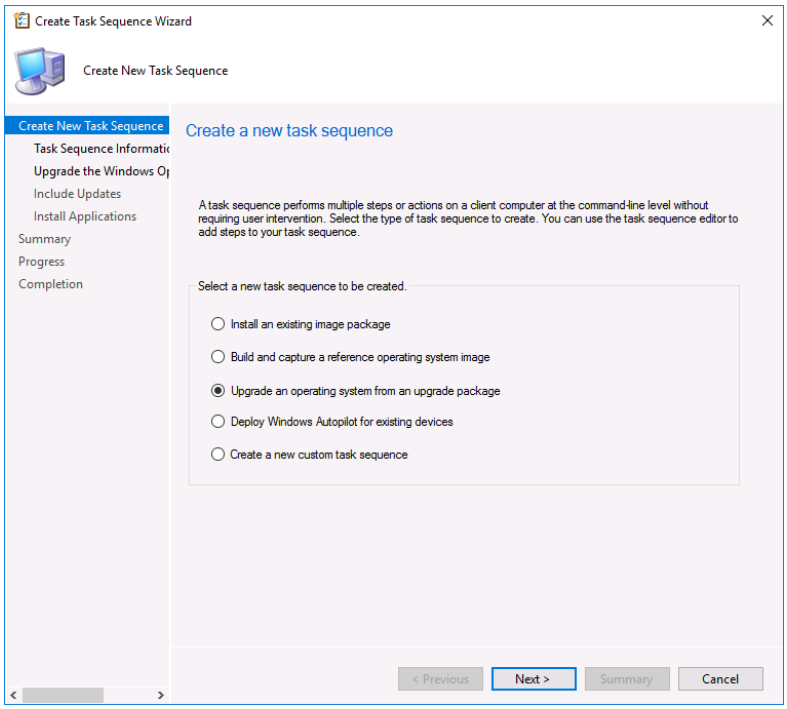
6. Click **Close** once the process finishes.



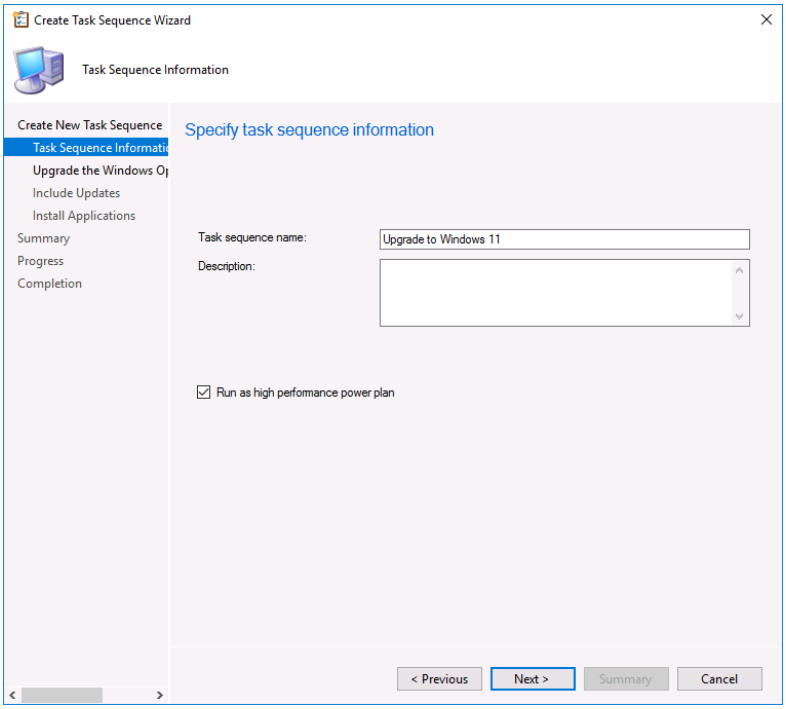
7. With the Task Sequences node selected, at the top left of the ribbon, click the **Create Task Sequence** button.



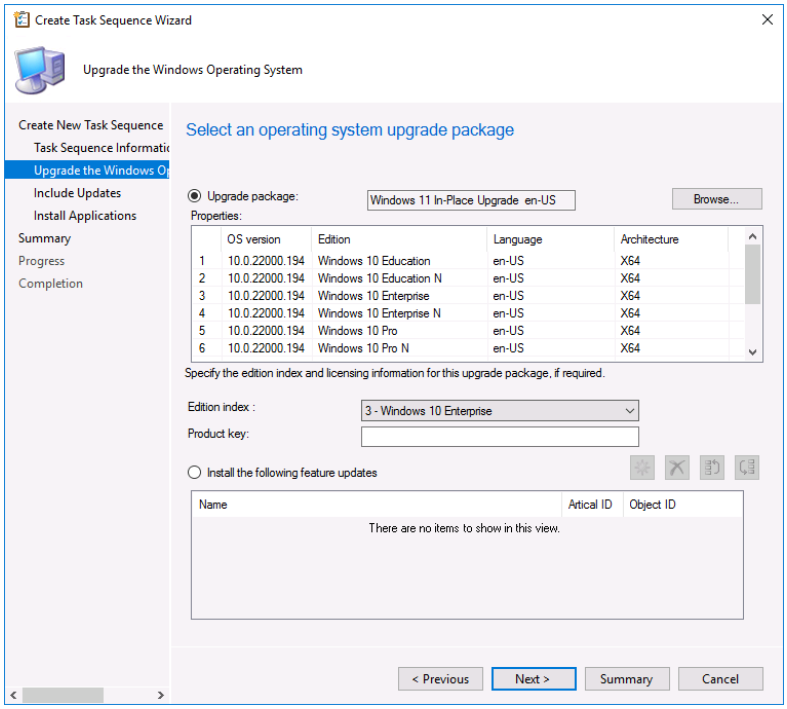
8. In the Create Task Sequence Wizard, choose **Upgrade an operating system from an upgrade package**



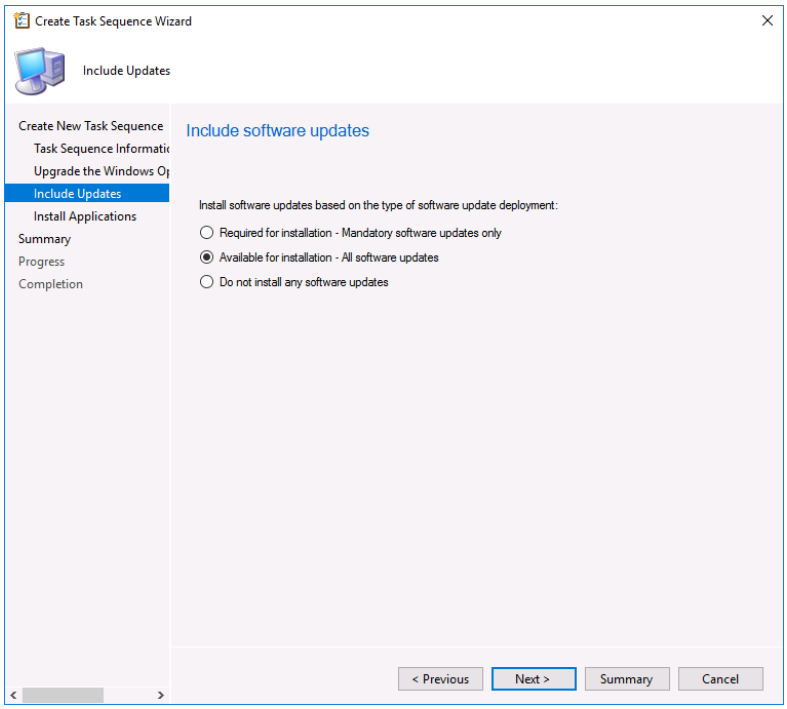
9. On the Task Sequence Information page, call the sequence **Upgrade to Windows 11** and **check** the box below to “Run as high performance power plan”



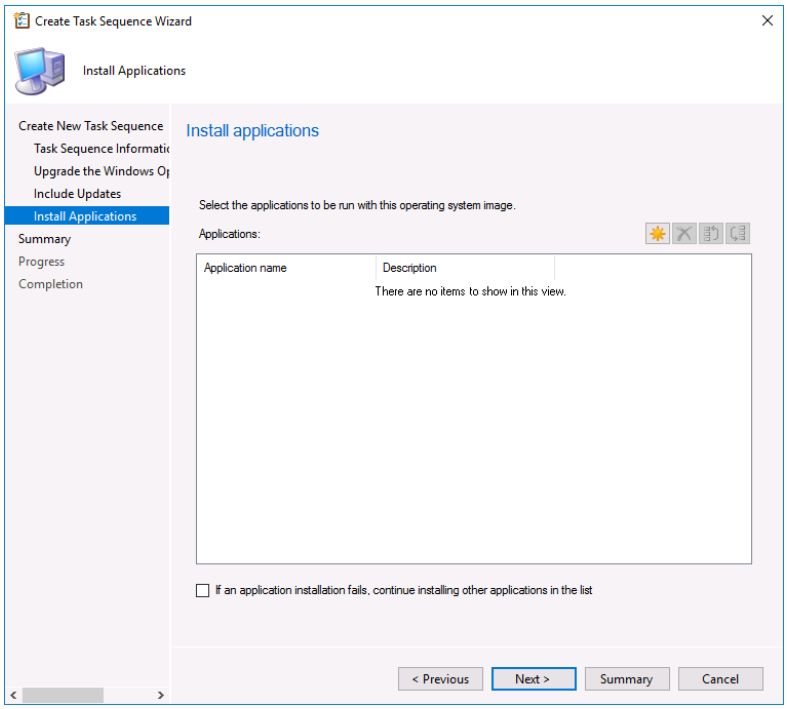
10. Click Browse to select the Ugprade Package, and below ensure that the Enterprise index is selected. Someone at Microsoft must have forgotten to update the Windows version in the Index ☹



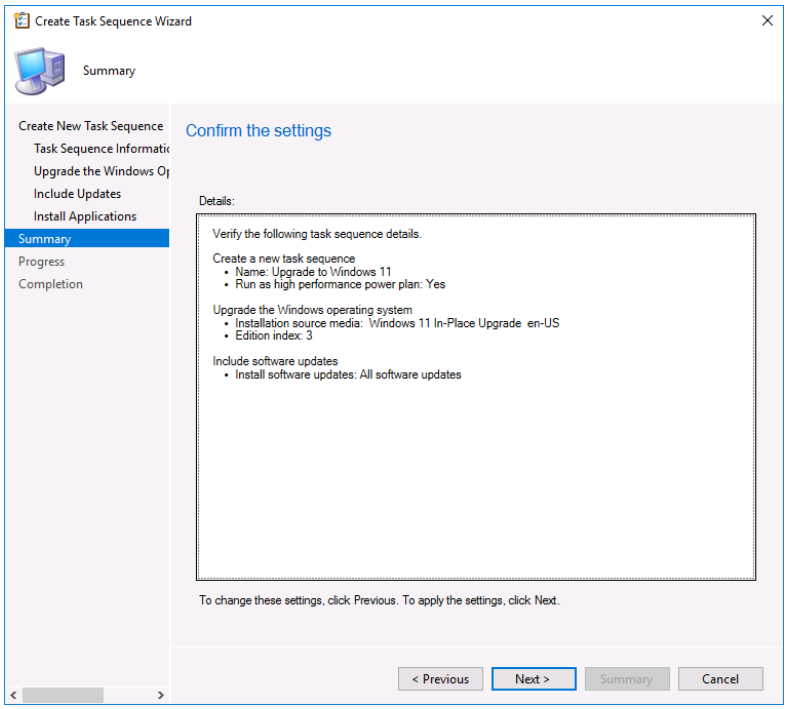
11. A completely patched workstation is a completely happy workstation, so we'll allow all available updates to be installed during the upgrade.



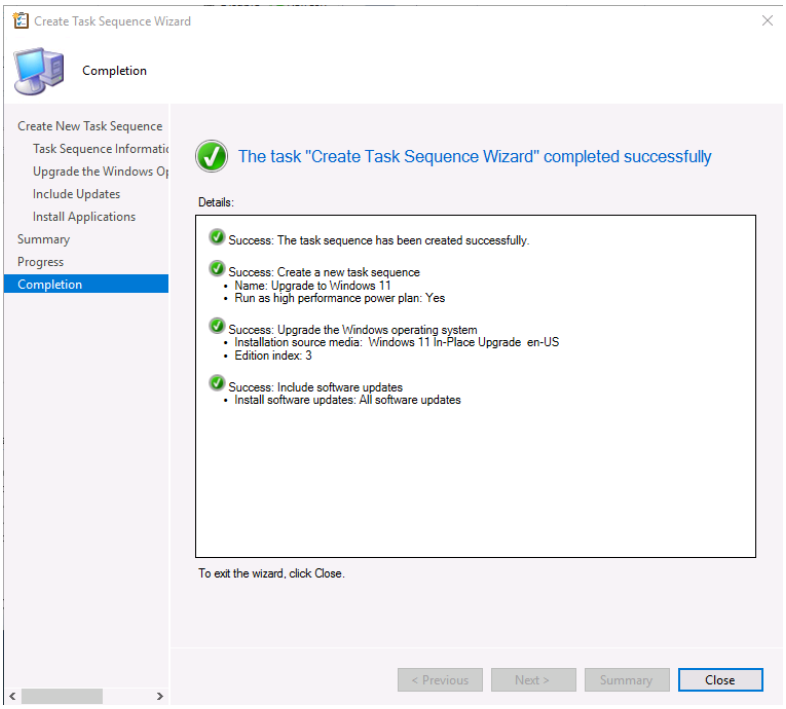
12. We're not going to change any application installs with the upgrade, so leave the Install Applications page empty



13. Confirm the settings on the Summary page, and click Next to create the sequence



14. As there are no new packages to create for this sequence, it should be instantaneous.



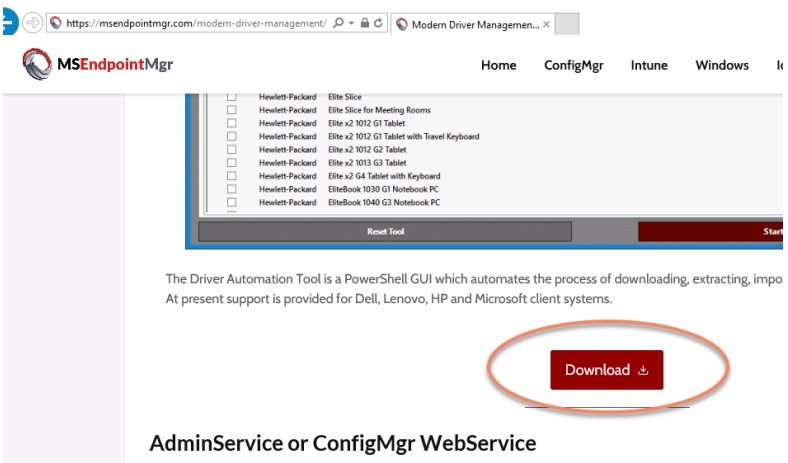
Exercise 4 – Drivers and Firmware

At this point, we have three task sequences that will work exceptionally well on virtual machines and any physical computers that are built from components that are on the Windows Hardware Compatibility List. For everything else, we need Drivers and BIOS/UEFI updates.

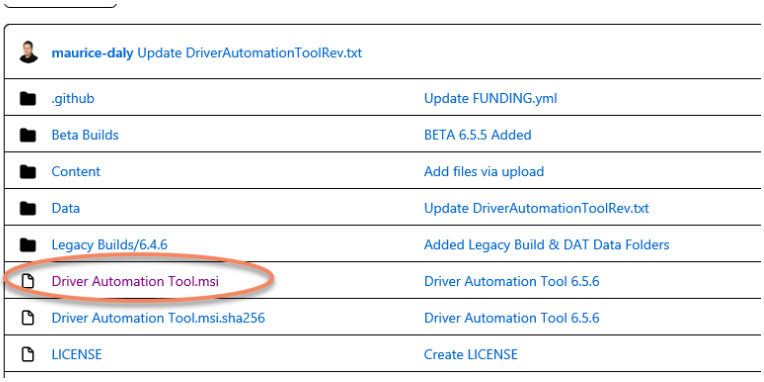
Driver and Firmware management has traditionally been a very manual process that required significant time out of an admin's duties to complete. Despite automation products by most vendors, the integration into Configuration Manager is sparse and forces an admin to hand-hold much of the process.

In the 3rd party chapter earlier, we highlighted MSEndpointMgr's Configuration Manager Prerequisites Tool. What's more popular than their prereq tool is their Modern Driver Management and Modern BIOS Management utilities. These tools work for myriad environment designs and take a lot of the guesswork away from this type of management.

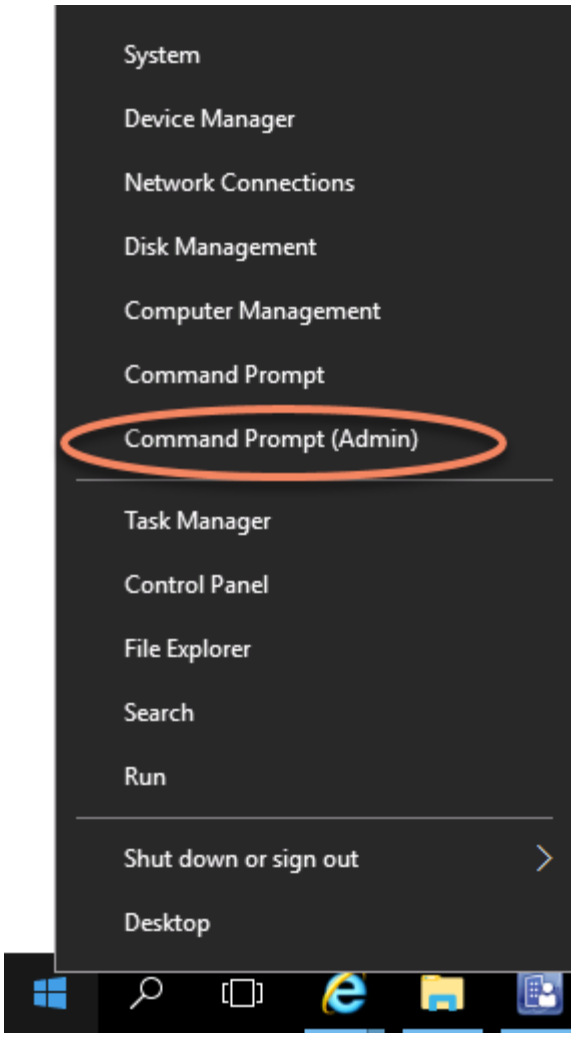
Both tools are designed and operated in the same way, and the instructions for each on the MSEndpointMgr.com website are intuitive and easy to follow. For the purposes of this exercise, we will focus on driver management, knowing a similar process can (and should) be adopted to maintain device firmware revisions within the enterprise.

Instructions	Screenshot (if applicable)
1. Logon to the Site Server as Administrator	
2. On the taskbar, click **Internet Explorer.**	
3. Navigate to https://msendpointmgr.com/modern-driver-management/ , scroll down past the first screenshot and click the Download button	

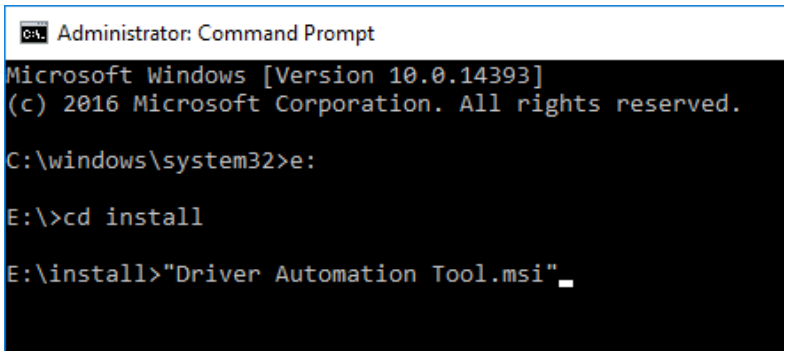
4. This will redirect you to MVP Maurice Daly's GitHub repo where you can download **Driver Automation Tool.msi** to the E:\install directory



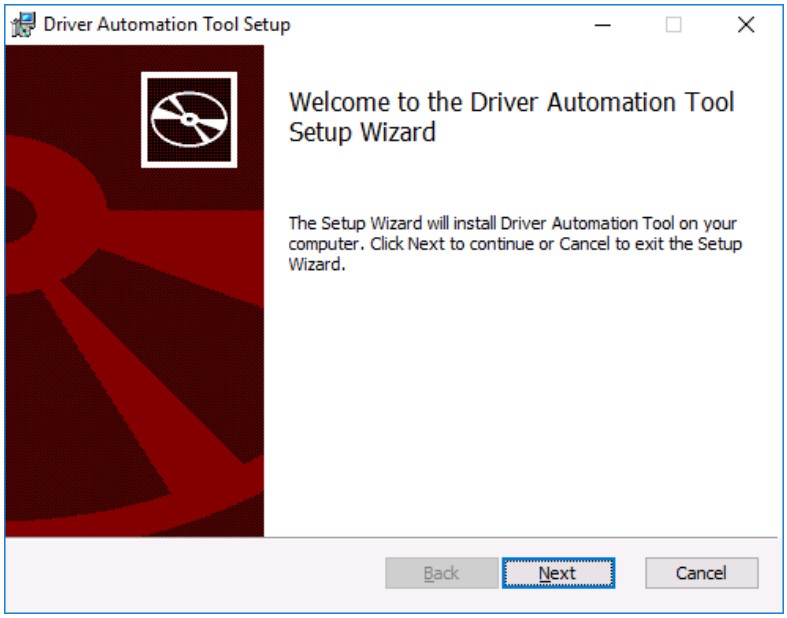
5. Right-click on the Start button and choose Command Prompt (Admin)



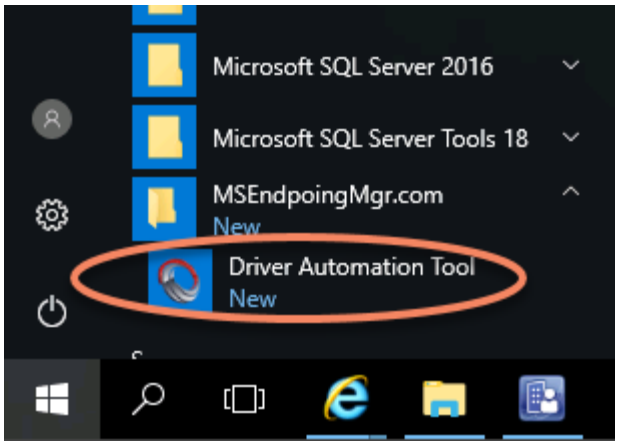
6. In the Command Prompt, change directory to the **E:** drive, then go to the **install** directory and launch **Driver Management Tool.msi**



7. Click Next, Next, Close through the wizard, accepting the default installation location in the C: drive.

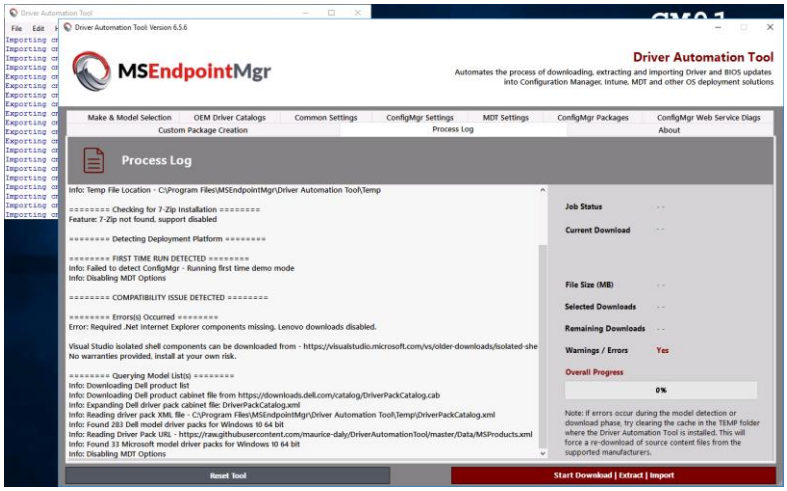


8. Click Start. Under the new MSEndpoingMgr.com (sic) folder, launch the Driver Automation Tool

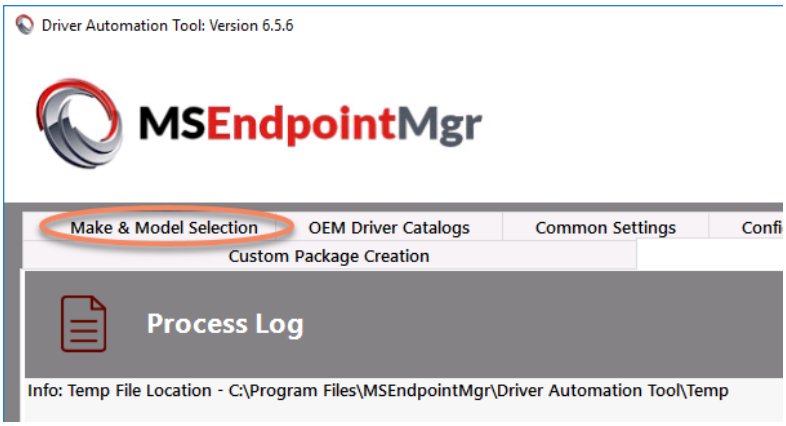


9. This will start the execution of a series of PowerShell scripts, and after a short while the application window will appear.

   NOTE: Successful installation of this tool requires a service account to be created and delegated access in Configuration Manager. The process is thoroughly documented on the MSEndpointMgr site, however for this lab we will use the Administrator account.

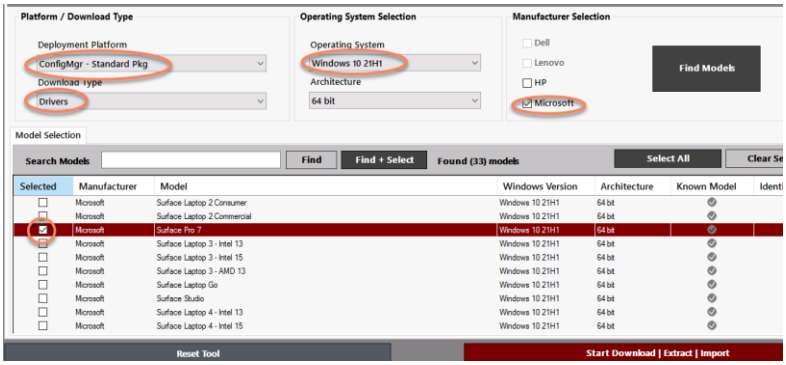


10. In the upper-left of the window, select the **Make & Model Selection** tab



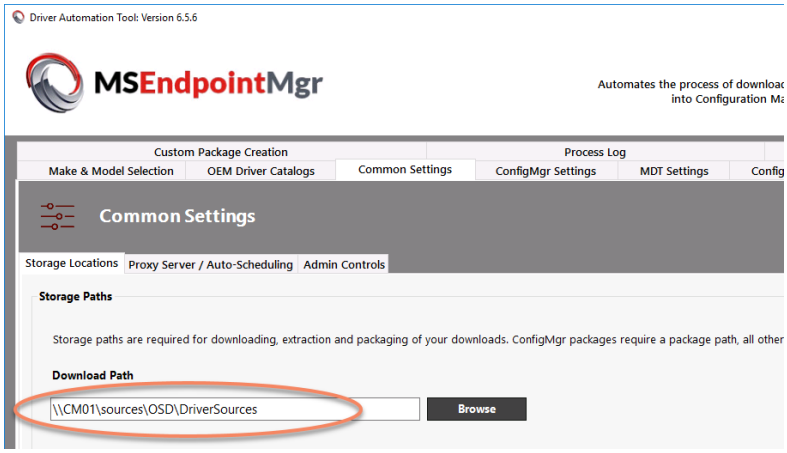
11. This page will tell the tool what we're using, how we want the packages to be created, the types of packages we're downloading, and the applicable models. Choose the appropriate options to match the screenshot → and choose Surface Pro 7 in the Model Selection list below.

    NOTE: You cannot select Microsoft until Windows 10 21H1 is selected.
    ALSO NOTE: Windows 11 enteprise drivers

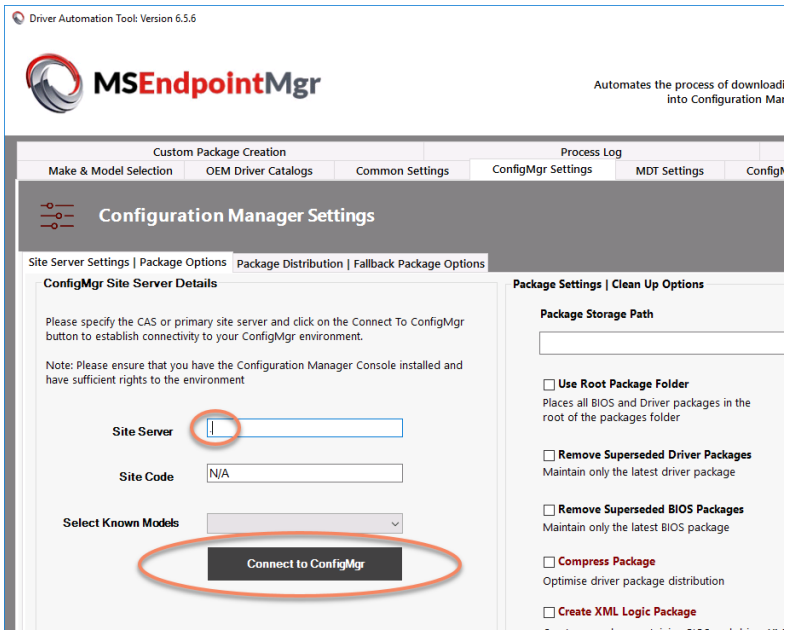


were not yet available when this screenshot was taken

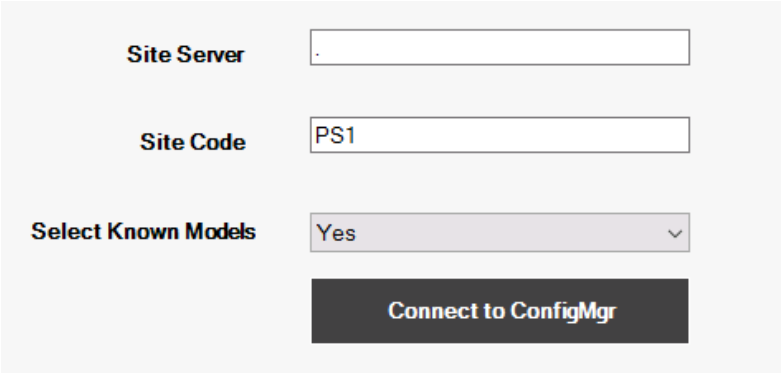
12. Click on the Common Settings tab, and change the download path to \\CM01\sources\OSD\DriverSources



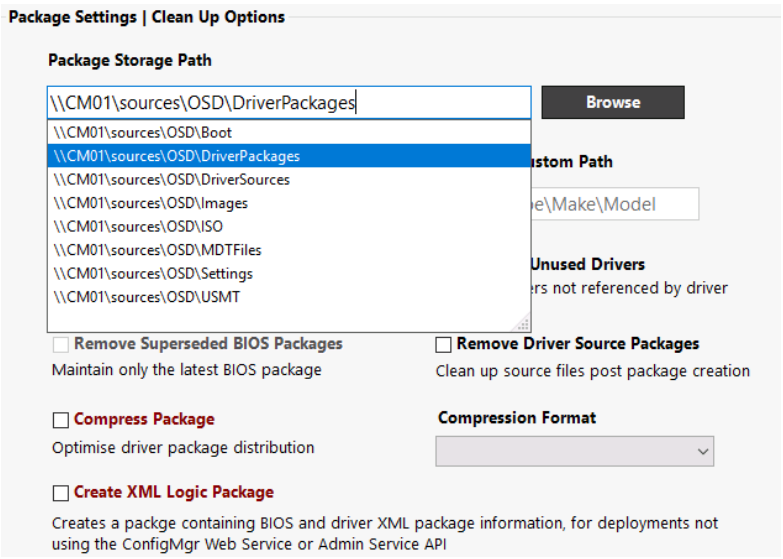
13. Click on the ConfigMgr Settings tab, use a period (.) for the Site Server and click the **Connect to ConfigMgr** button



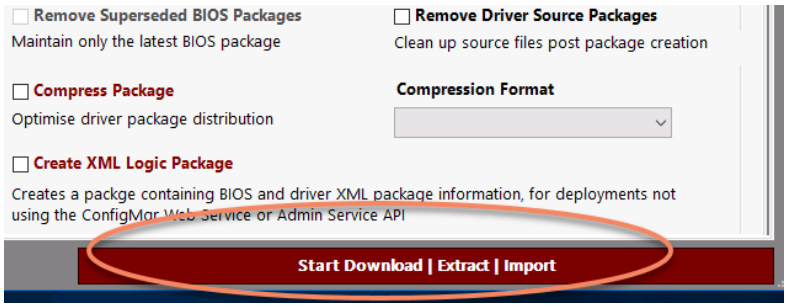
14. On a successful connection, ths will populate the Site Code. In the “Select Known Models” dropdown, choose **Yes**.
Extremely powerful in a production environment, the “Find Models” button will scan the ConfigMgr database and automatically select the models it found in your enterprise.



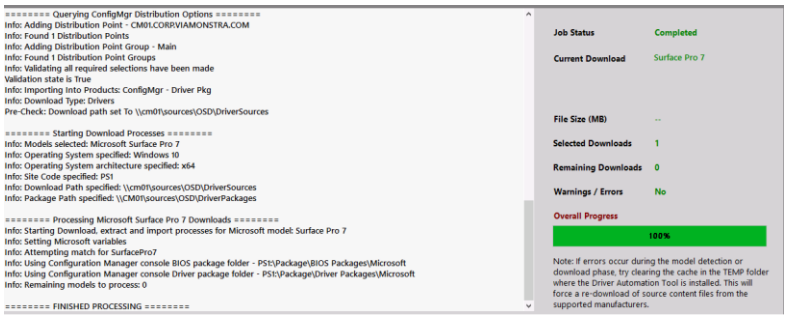
15. On the right side, specify \\CM01\sources\OSD\DriverPackages for the storage path, and leave the other options unchecked.



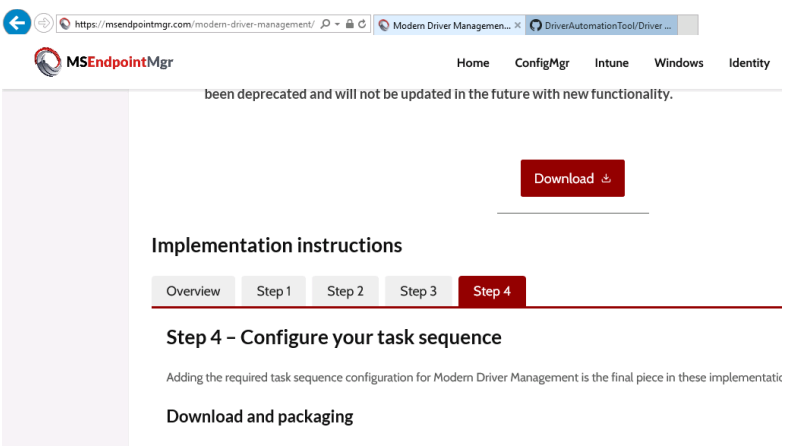
16. In Production, at this point you'd go back to the Make & Model Selection tab and click that magic "Find Models" button, however since we don't have any physical machines, it won't find any. At the bottom of the screen, click the **Start Download | Extract | Import** button that's at the bottom of every screen.



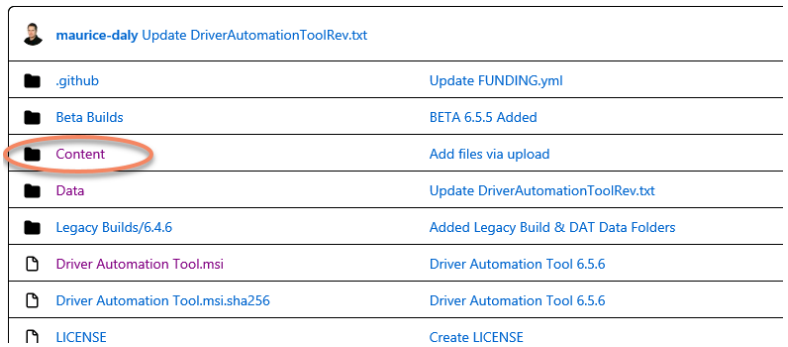
17. The process will complete and create a Surface Pro 7 Driver Pack. In production, this would create a driver pack for every model you have in ConfigMgr.



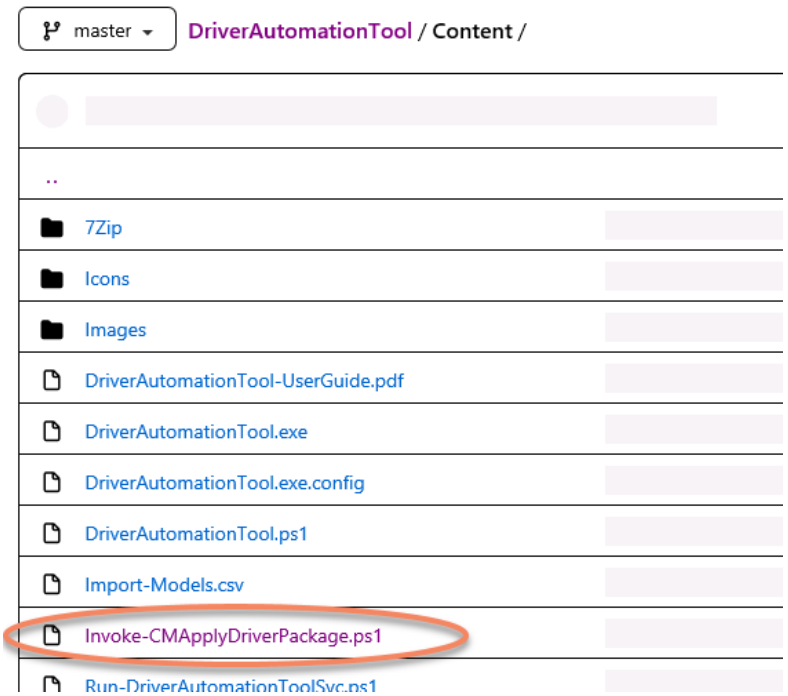
18. Back in Internet Explorer, on the MSEndpointMgr.com tab, scroll to the bottom of the page to the “Implementation instructions” section and click on Step 4. Here, the team describes the steps we need to take to integrate this this tool into OS Deployments



19. Switch to the other tab in Internet Explorer, Maurice’s GitHub Repository, and click on the **Content** folder

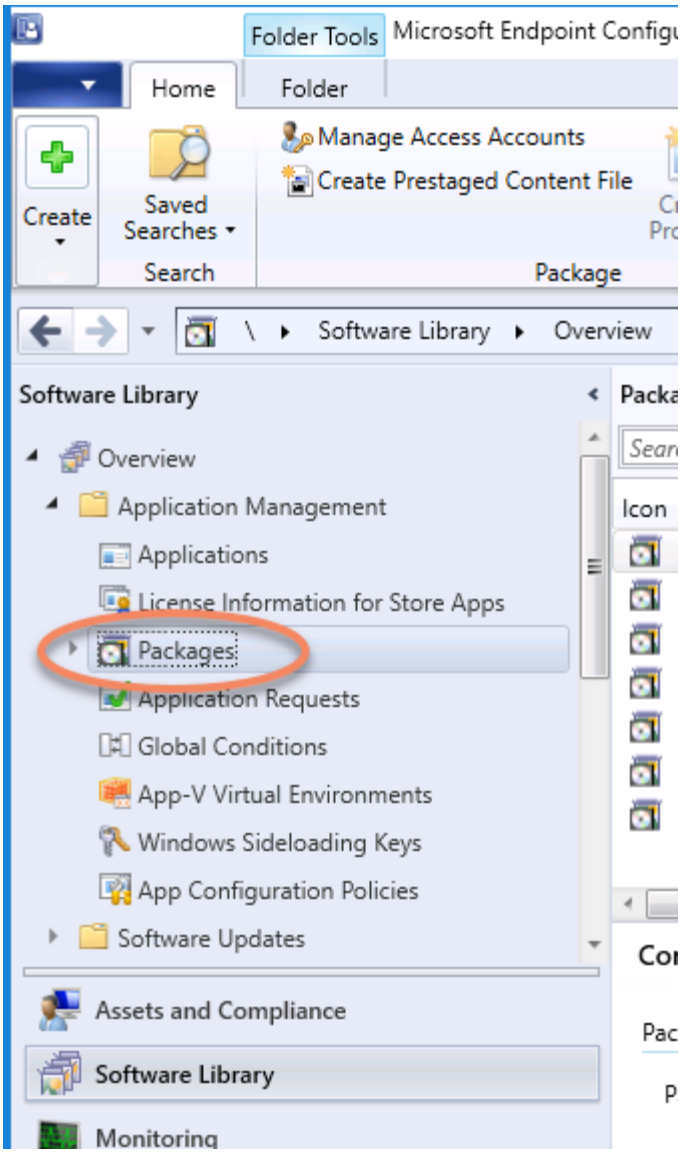


20. Next, click the **Invoke-CMApplyDriverPackage.ps1** link

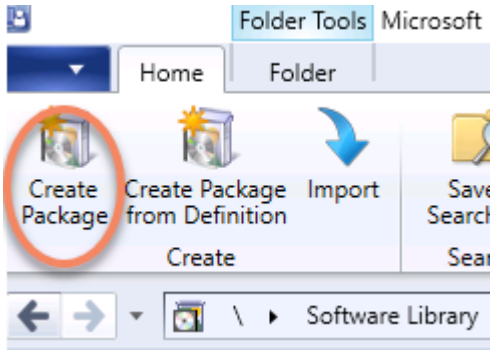


21. On the next page, click the **Raw** button, which will give you a clean window where you can use CTRL+A to copy all the text and save it in "E:\sources\Software\Packages\ModernDriverMgmt" as **Invoke-CMApplyDriverPackage.ps1** using Notepad.

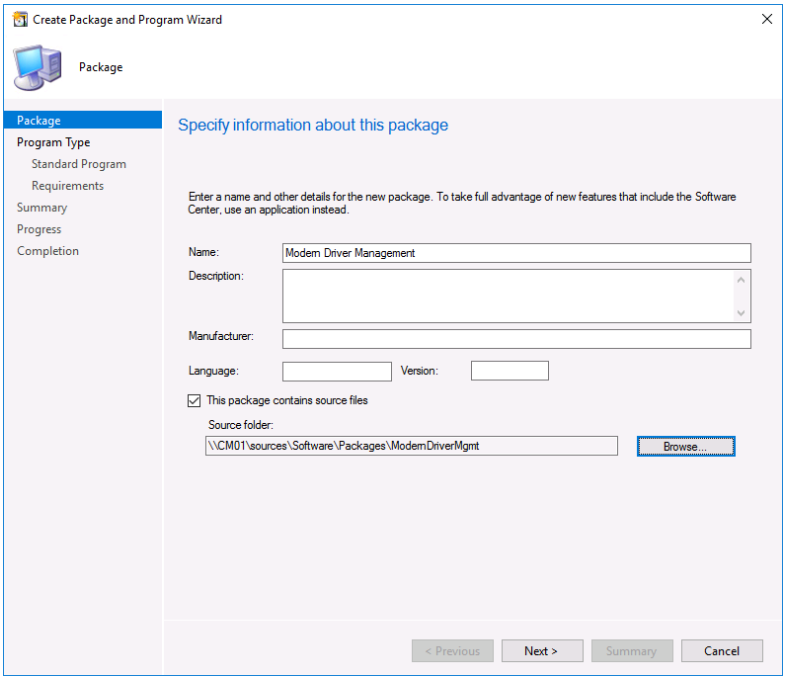
22. In the Configuration Manager Console, expand Application Management in the Software Library workspace and select **Packages**



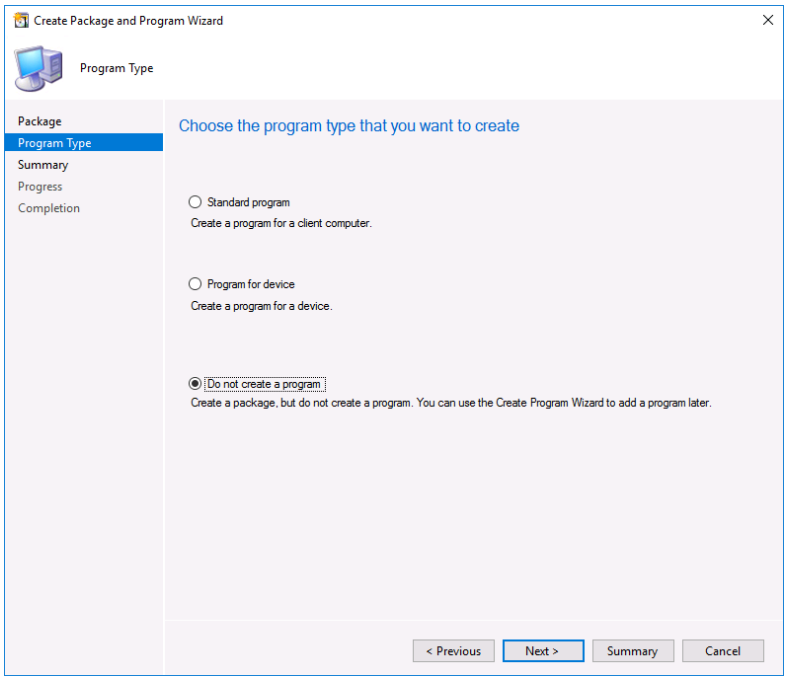
23. At the top-left of the ribbon, choose **Create Package**



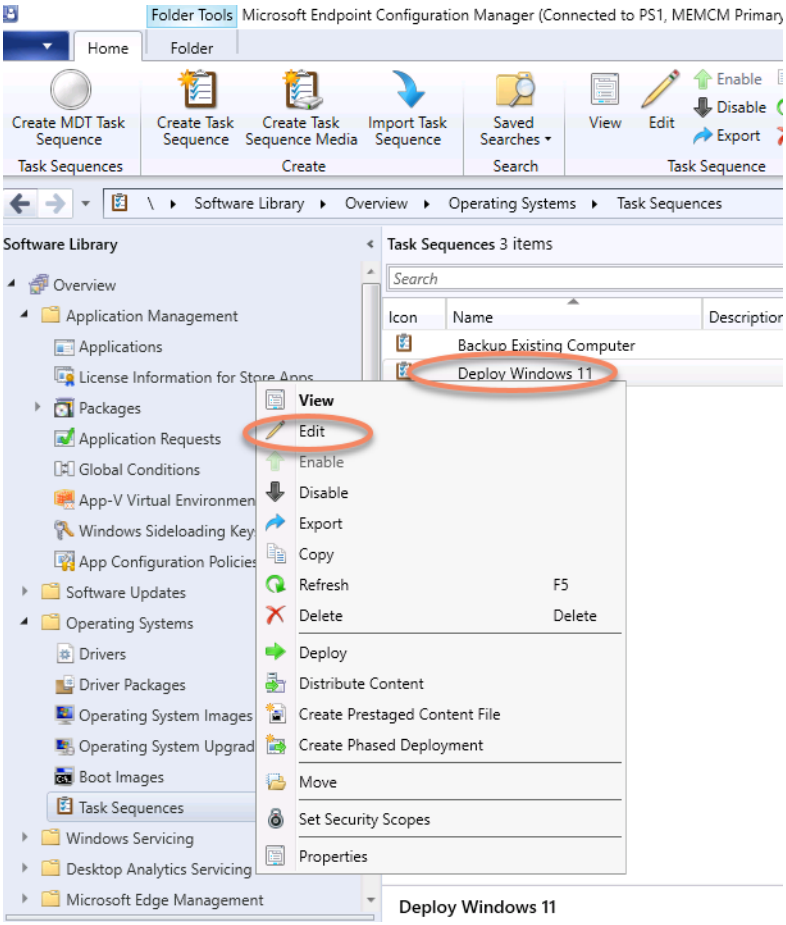
24. Call the package **Modern Driver Management** and specify \\CM01\sources\Software\Packages\ModernDriverMgmt as the source location



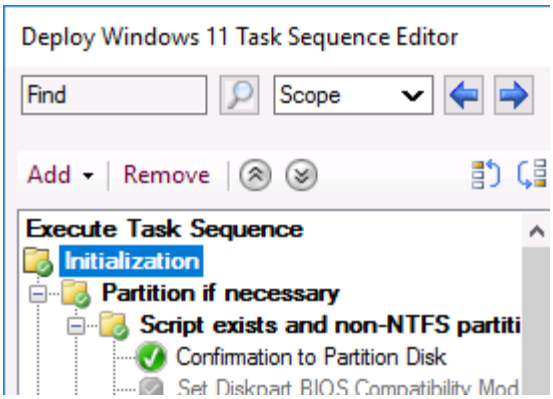
25. On the Program Type page, select the bottom radio button to “Do not create a program” and Next, Next, Close the wizard



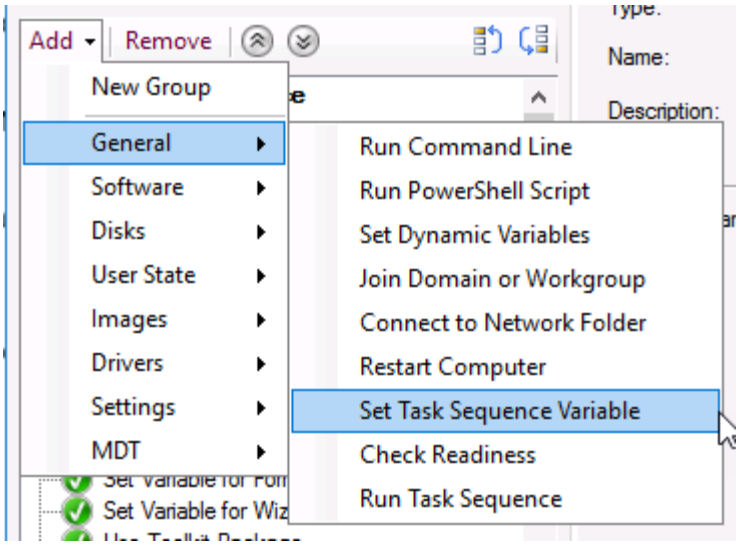
26. Now, let's go back to our Task Sequences node and edit the Deploy Windows 11 sequence



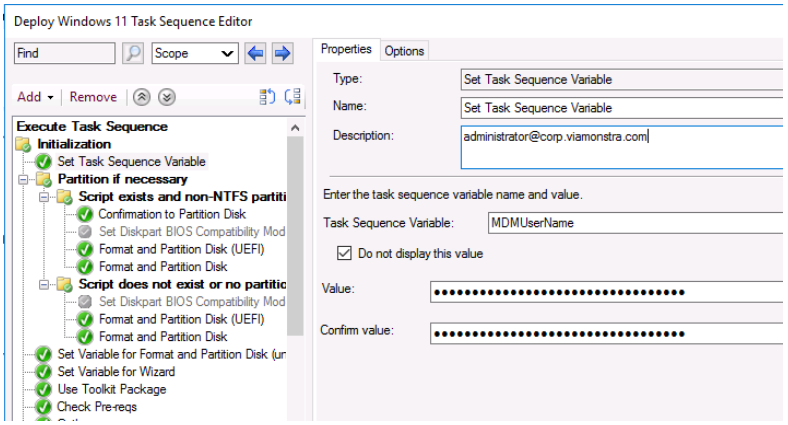
27. In the Task Editor window, at the top of the sequence, click **Initialization**



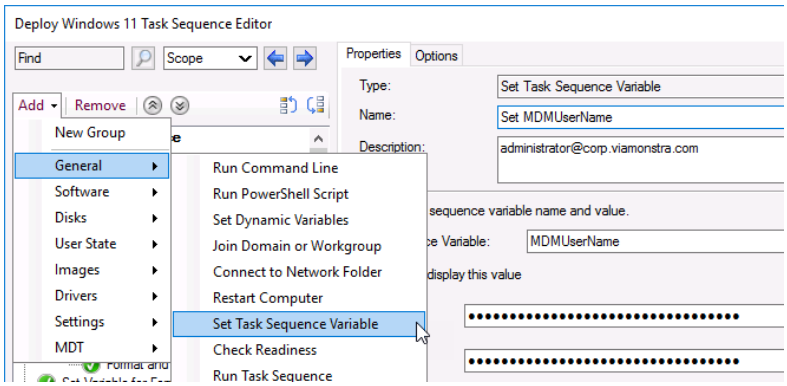
28. Above that, click Add, then General and **Set Task Sequence Variable**



29. The Task Sequence Variable Name should be **MDMUserName** and the value should be **administrator@corp.viamonstra.com**. Ensure that you check the box that says "Do not display this value"

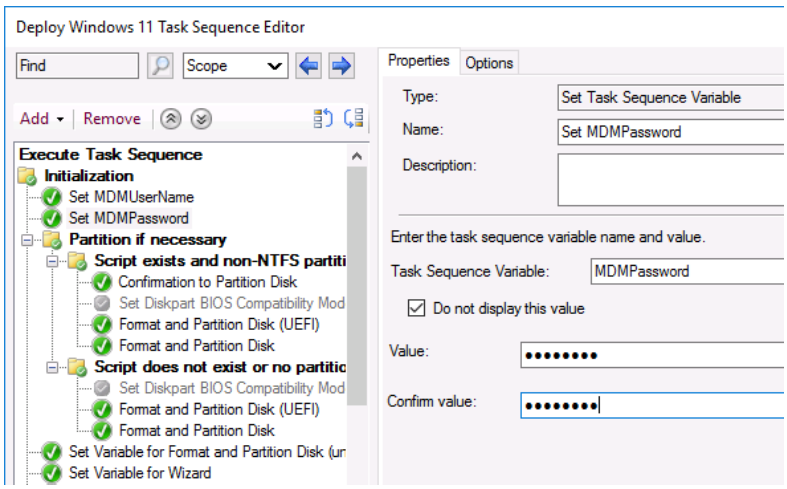


30. Change the name at the top to **Set MDMUserName** and then click Add -> General -> Set Task Sequence Variable again

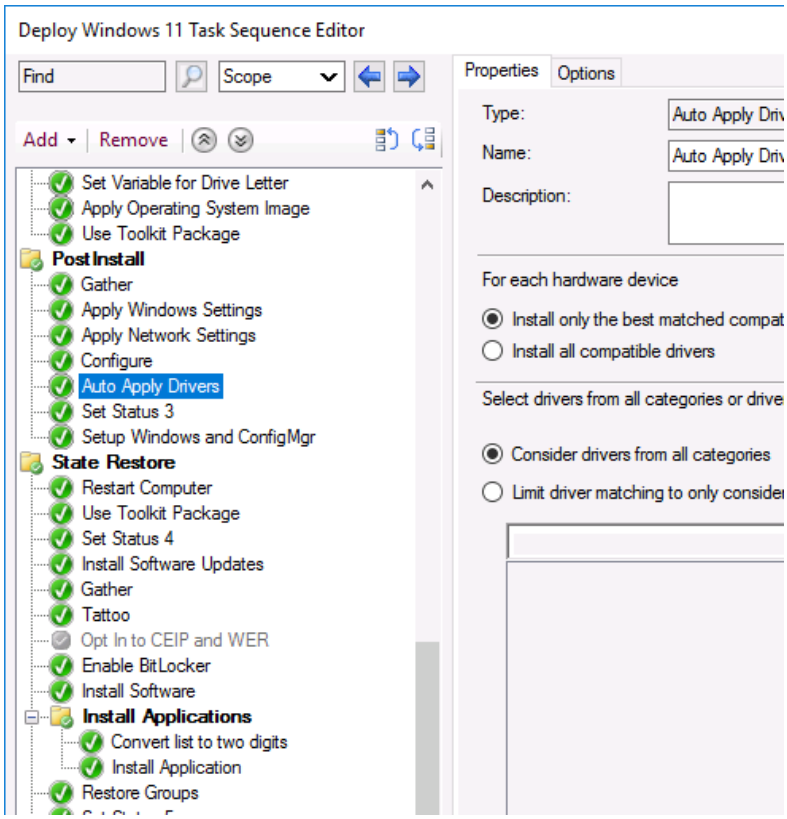


31. The second variable is called **MDMPassword** and needs to specify the Domain Administrator password you've been provided. Again, this value should be kept hidden.

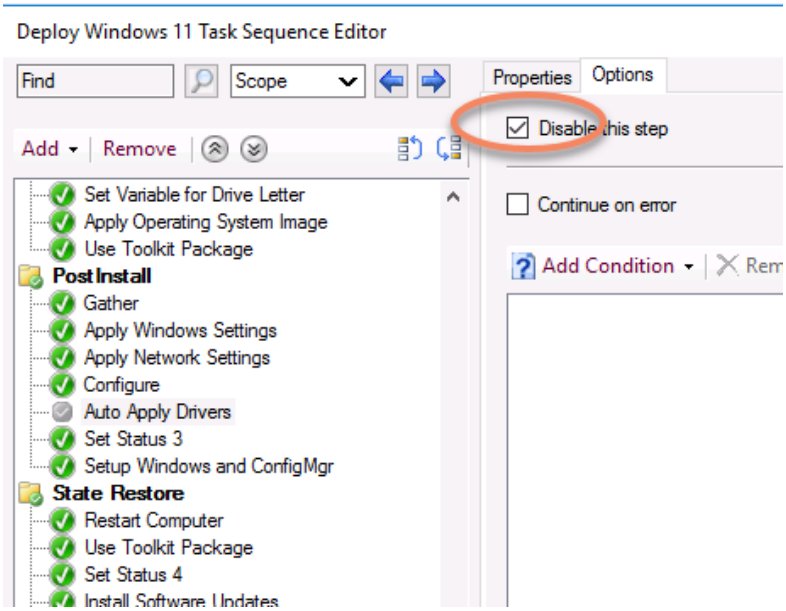
    NOTE: As previously mentioned, in production environments you would create a service account for this purpose as outlined on the MSEndpointMgr.com site



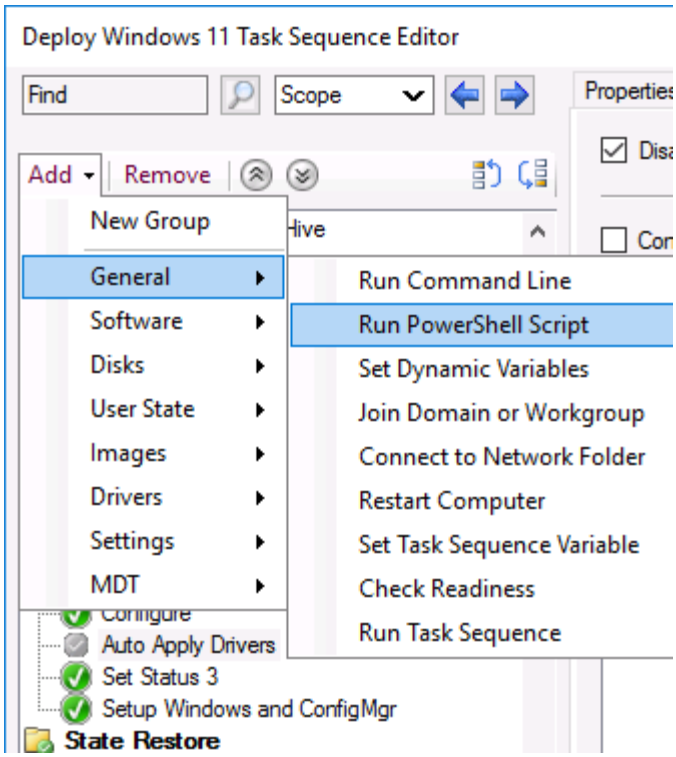
32. Scroll down in the sequence and locate the Post Install section. Click the **Auto Apply Drivers** step



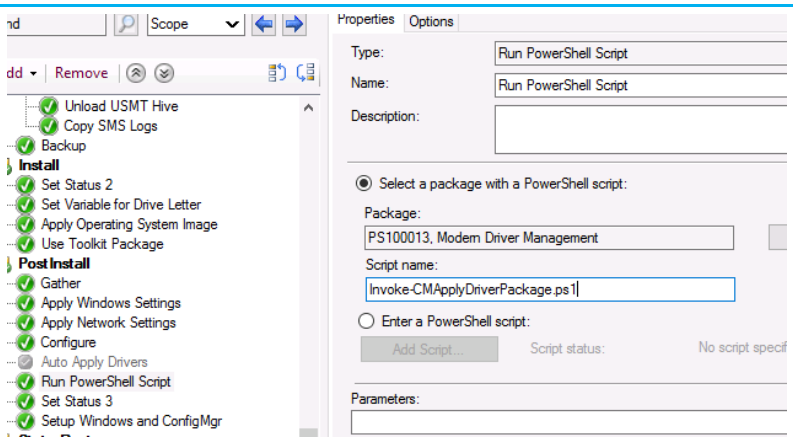
33. On the right-hand side, click the Options tab and check the **Disable this step** box



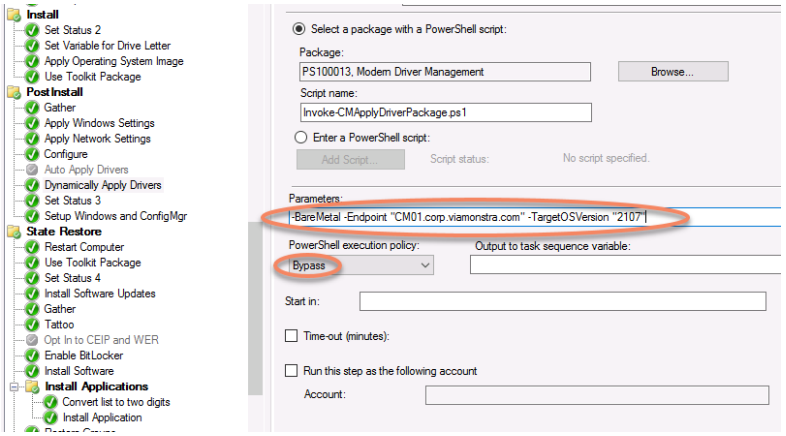
34. Click Add -> General -> Run PowerShell Script



35. Select the Modern Driver Management Package, and type **Invoke-CMApplyDriverPackage.ps1** for the Script name. Call the step **Dynamically Apply Drivers**

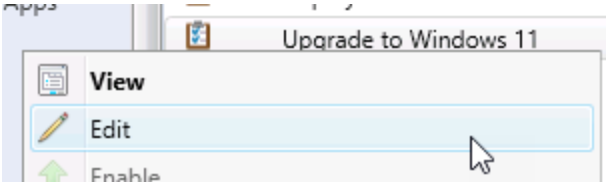


36. In the Parameters section below, type the following: **-BareMetal -Endpoint "CM01.corp.viamonstra.com" -TargetOSVersion "2107"** and change the PowerShell Execution Policy to **Bypass**

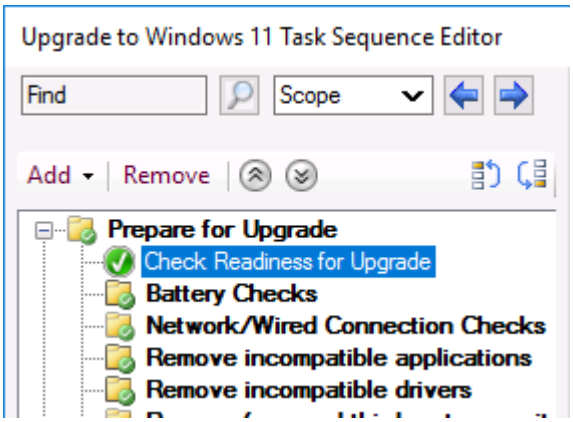


37. Click OK to close the Task Sequence

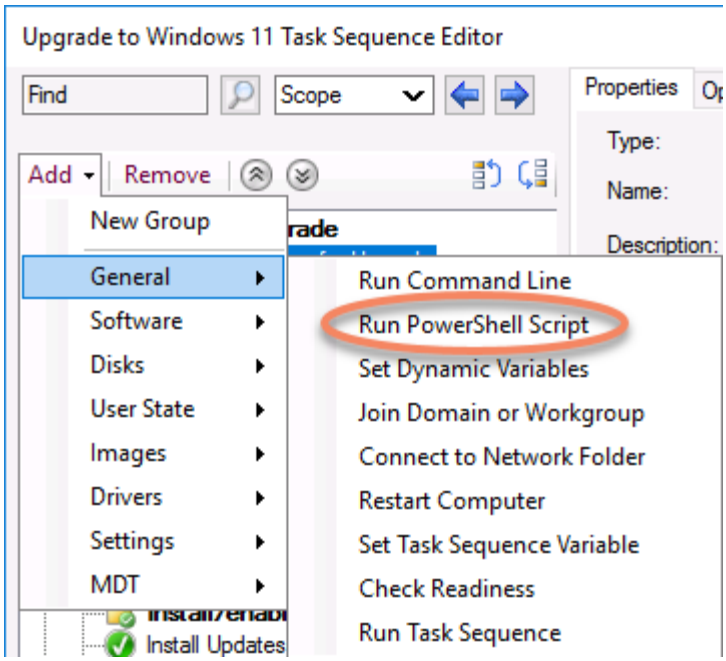
38. Next, Edit the Upgrade to Windows 11 Task Sequence



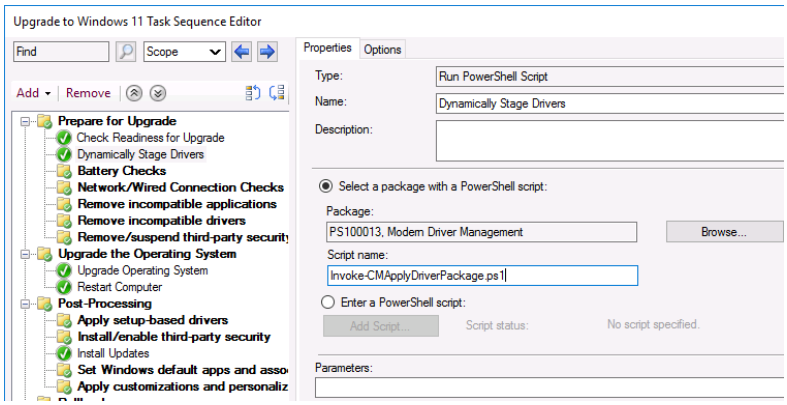
39. In the Task Sequence Editor, select **Check Readiness for Upgrade**



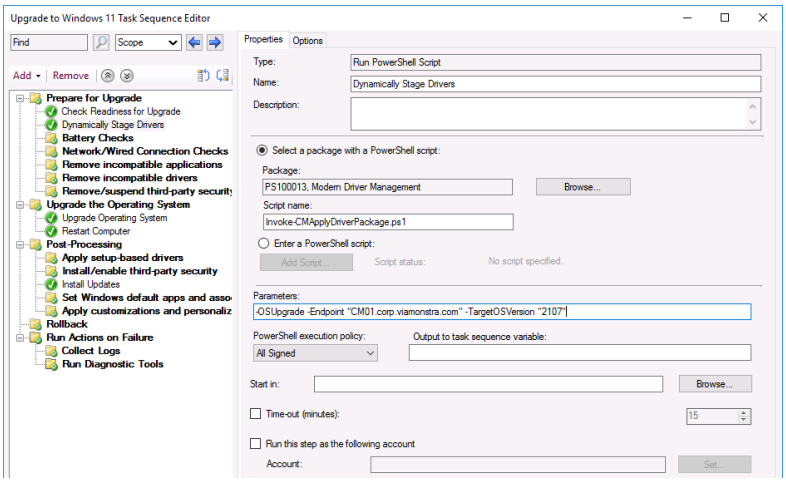
40. Click Add -> General -> Run PowerShell Script



41. Call the step **Dynamically Stage Drivers** and select the same Modern Driver Management package as before. The script name is also stil **Invoke-CMApplyDriverPackage.ps1**



42. Parameters for the script this time will be -OSUpgrade -Endpoint “CM01.corp.viamonstra.com” -TargetOSVersion “2107”



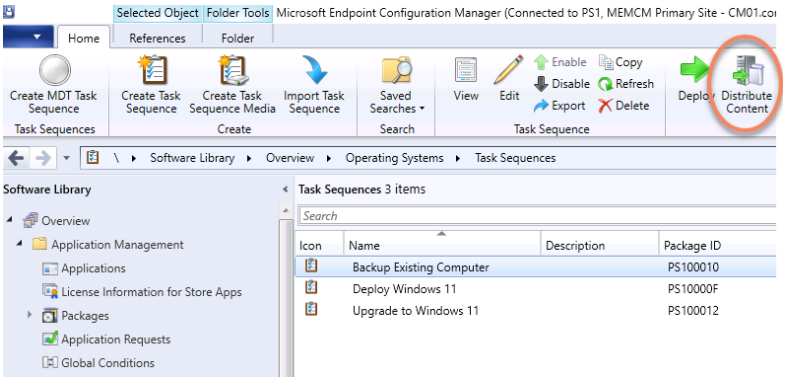
43. Click OK to close the Task Sequence Editor

Exercise 5 – Distribute and Deploy

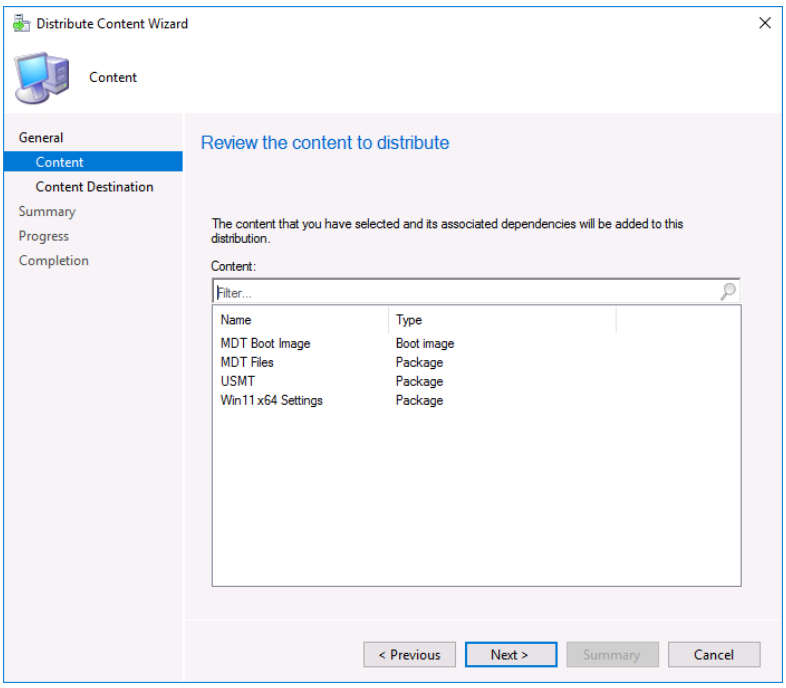
With the sequences all created, we need to get the content sent to the distribution point, and the sequences made available through a collection advertisement.

Instructions	Screenshot (if applicable)

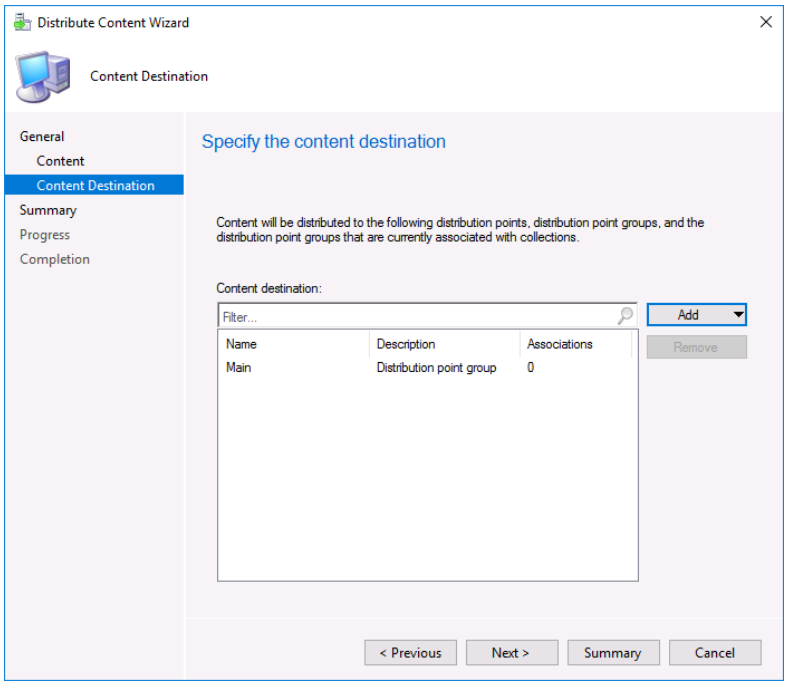
1. Select the “Backup Existing Computer” task sequence and choose **Distribute Content** from the ribbon



2. This will open the Distribute Content Wizard, and on the Content tab will show the packages that are referenced in the sequence



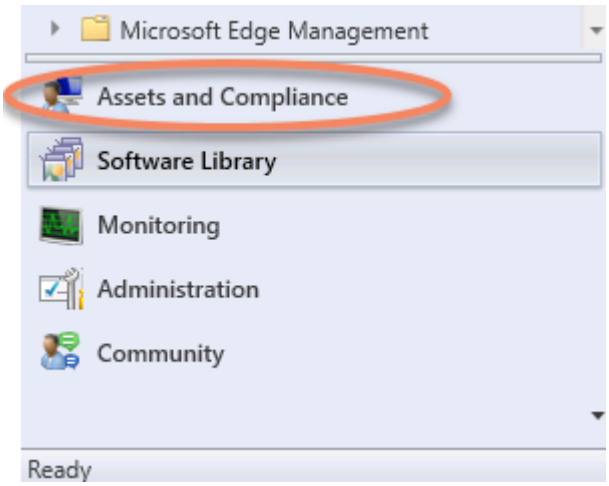
3. On the Content Destination, choose Add -> Distribution Point Group and select the **Main** DP Group. Even though we have only one DP, it's standard practice to deploy content to groups rather than individual DPs.



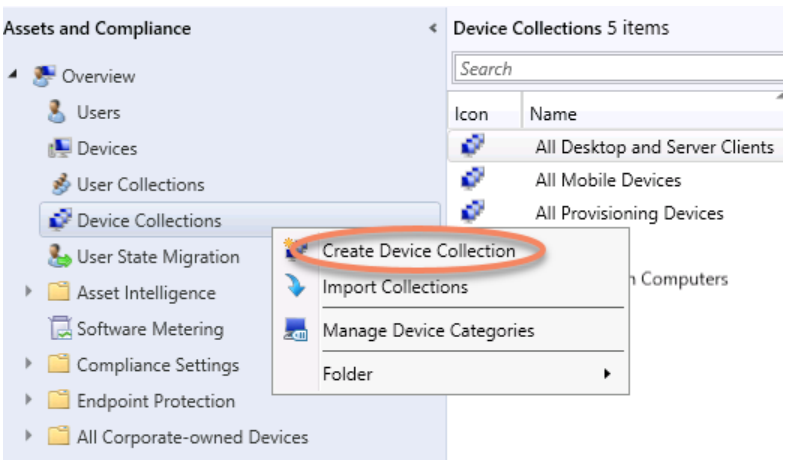
4. Click Next, Next, Close to finish the process

5. Repeat steps 1-4 for the other two task sequences, ensuring all referenced content for OS Deployments has been copied to the DP

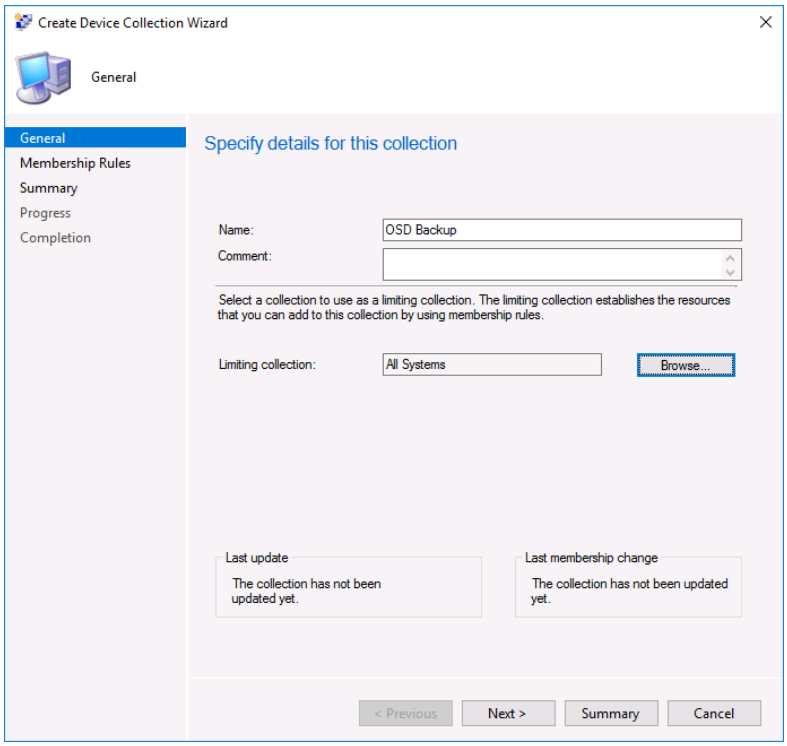
6. At the bottom-left of the Configuration Manager Console, click the **Assets and Compliance** workspace and select the Device Collections Node



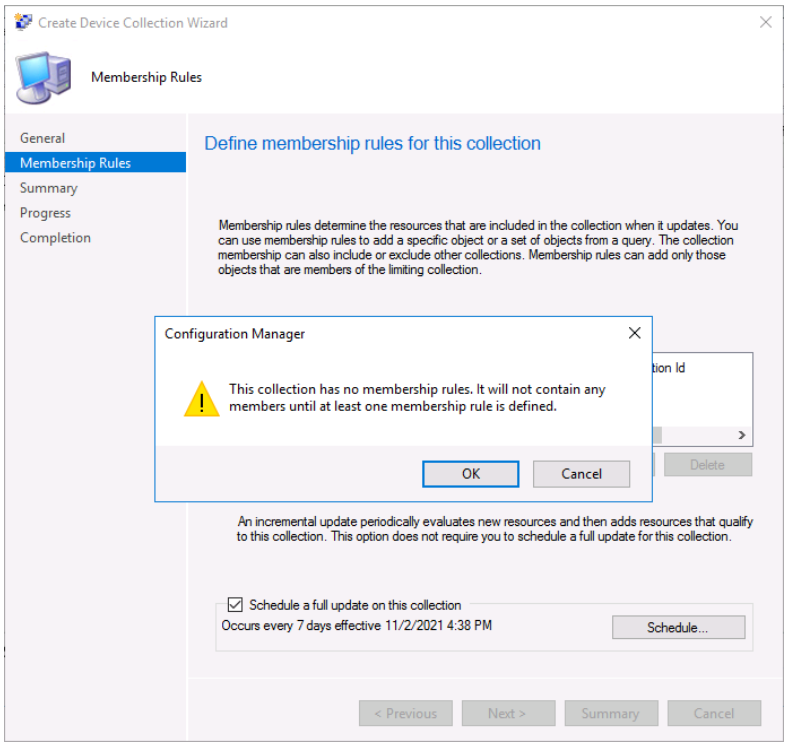
7. Right-click Device Collections and choose **Create Device Collection**



8. Give the collection a name of **OSD Backup** and choose **All Systems** as the Limiting Collection

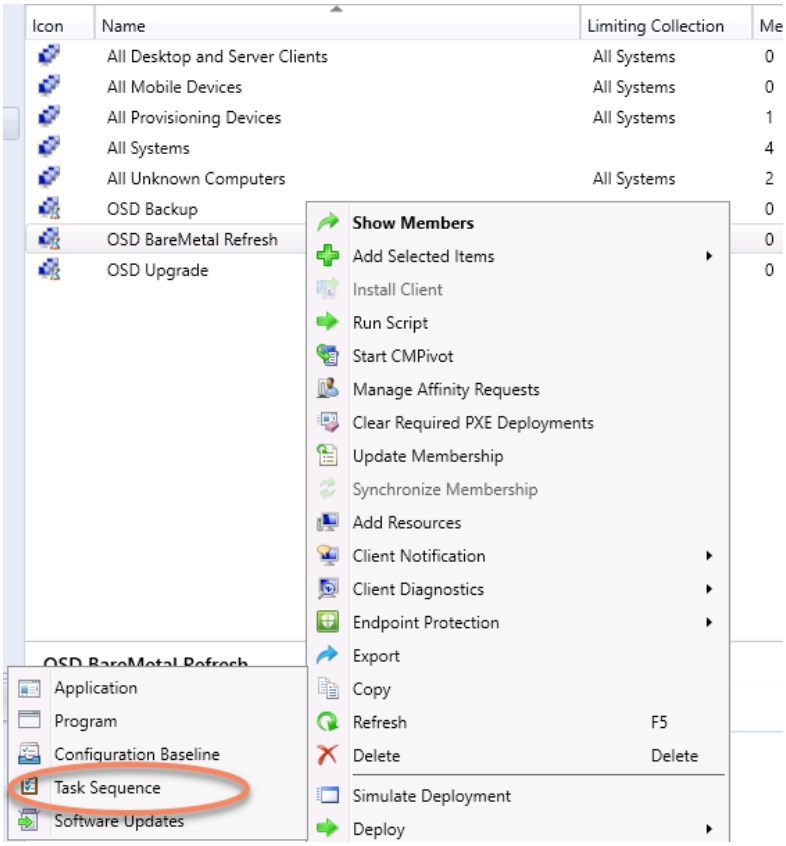


9. Click Next until the wizard is complete. As the collection contains no membership rules (devices will be manually added later), you must click OK on the warning message when it pops up

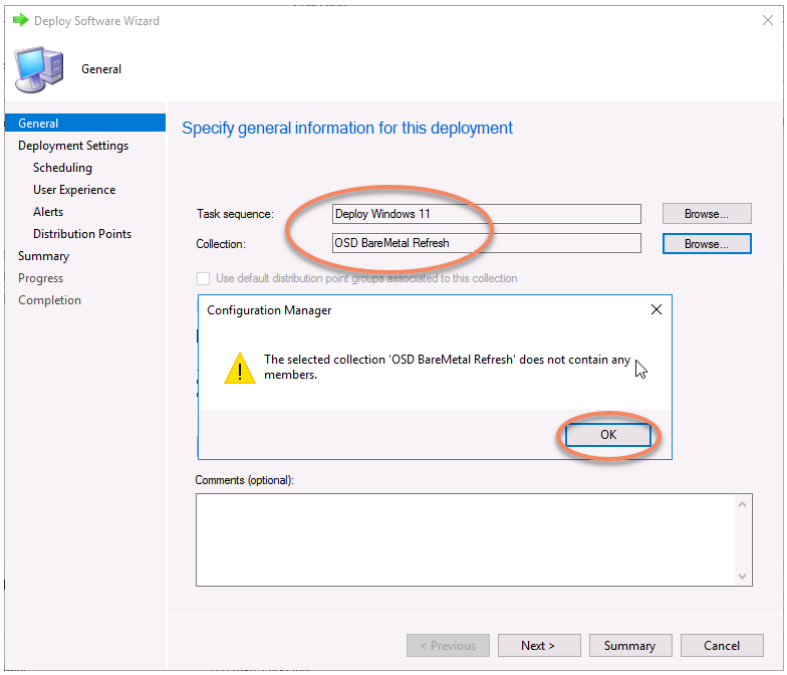


10. Repeat this process to create two more collections with zero members. The first collection is called **OSD Upgrade** and the second is **OSD BareMetal Refresh**

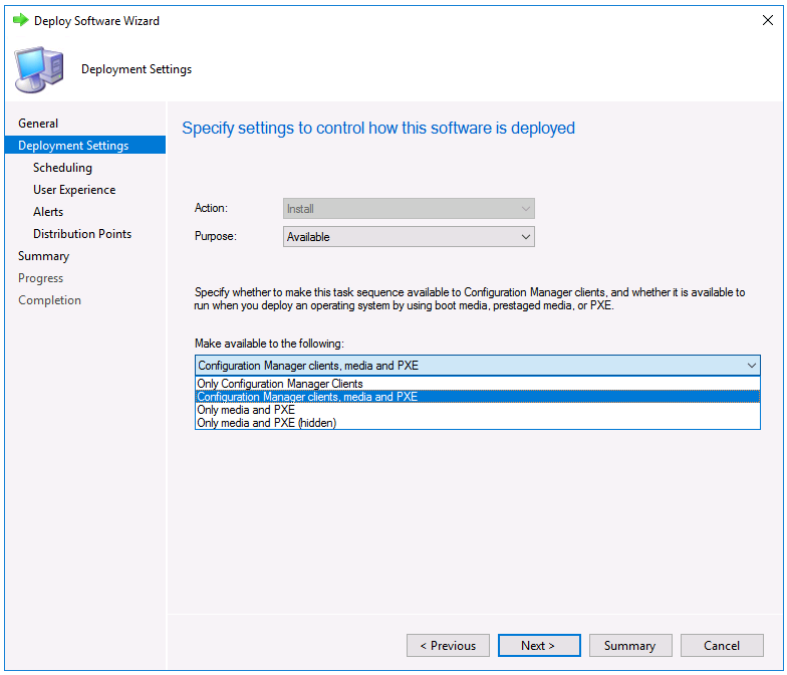
11. Next, we're going to deploy each task sequence to their respective collection. **The Deploy Windows 11** sequence is used for both Bare Metal and Refresh scenarios, as well as the target machine for Replacements



12. After selecting the sequence and collection, you may have to re-select the collection again due to the lack of membership

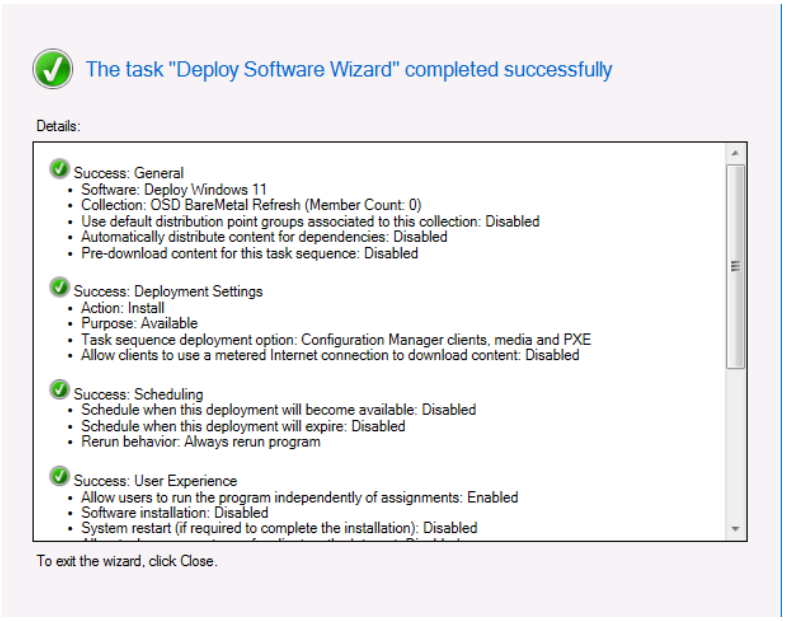


13. On the Deployment Settings page, the three task sequences will be deployed as **Available**. What they're made available to, however, is different. Upgrade and Backup sequences are made available to **Only Configuration Manager Clients**, where the BareMetal/Refresh sequence is



advertised to **Configuration Manager Clients, media and PXE**

14. Click Next, Next, Next, Next, Next, and Close to finish the wizard using the remaining default settings



Chapter 6 – Integration of MEM-Intune

Scenario

As organizations explore the features and enhancements available to them through Microsoft Azure and make them available to their staff, it becomes quickly evident that traditional device management and security controls will not always apply when users attempt a connection.

During Ignite 2019, Microsoft announced Co-Management capabilities between Configuration Manager and Intune. When enabled, Co-Management allows administrators to shift management of particular workloads for specific devices to Intune from Configuration Manager. This allows organizations to leverage the versatility of cloud-powered solutions like Intune while retaining the on-premises management they are accustomed to with ConfigMgr.

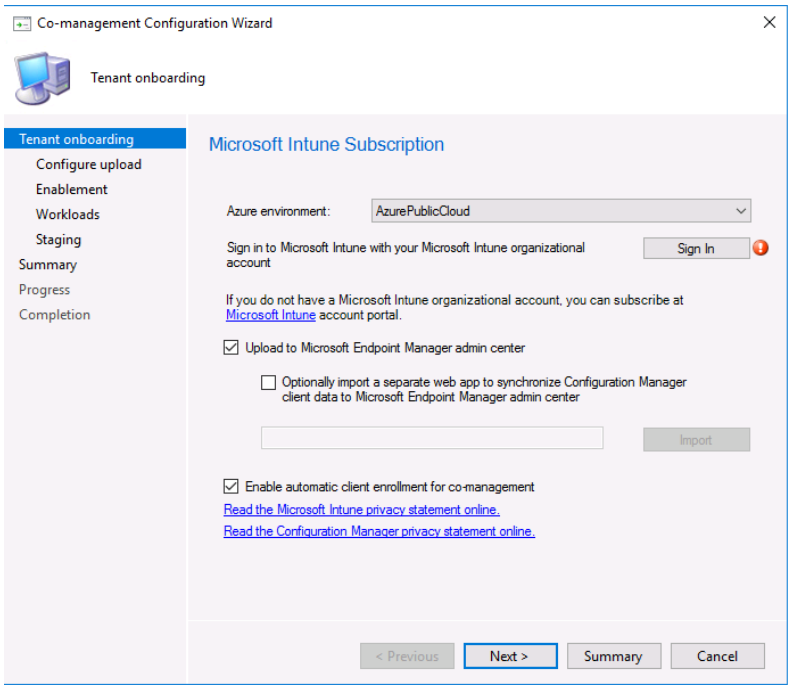
Exercise 1 – Configure Tenant Attach

The following steps will create the collections required to transition pilot workloads and enable tenant attach in the Configuration Manager console. Tenant attach is not the same as co-management, Tenants attach allows you to see and manage your Configuration Manager managed devices in the Microsoft Endpoint Manager admin center.

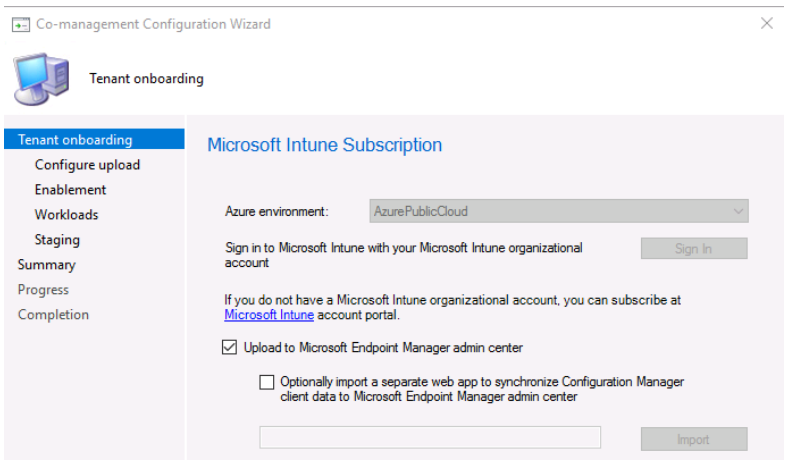
Instructions	Screenshot (if applicable)
1. Logon to the CM01 as VIAMONSTRA\Administrator	
2. Start the Configuration Manager console 3. Navigate to **Administration >**	

Cloud services >
Cloud Attach

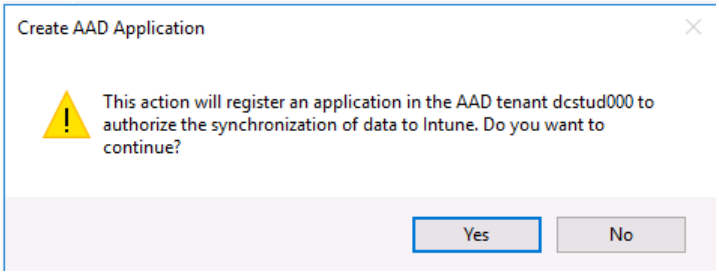
4. Click **Configure co-management**



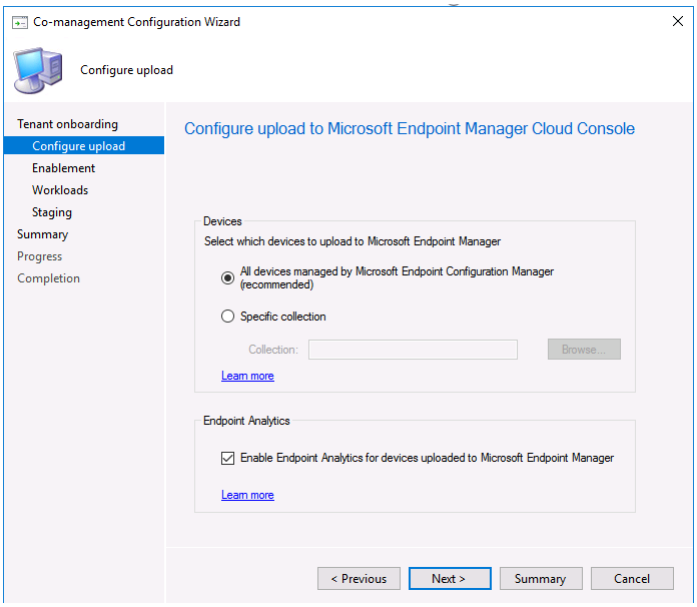
5. Click Sign In and login with your Azure Admin account
6. Make sure that **Upload to Microsoft Endpoint Manager admin center is checked**
7. Click **Next**.



8. Click **Yes**



9. Leave all by default and click **Next** so that all devices managed by Configuration Manager will be synced and that Endpoint Analytics for the devices is also uploaded to Microsoft Endpoint Manager.
10. Click **Next.**



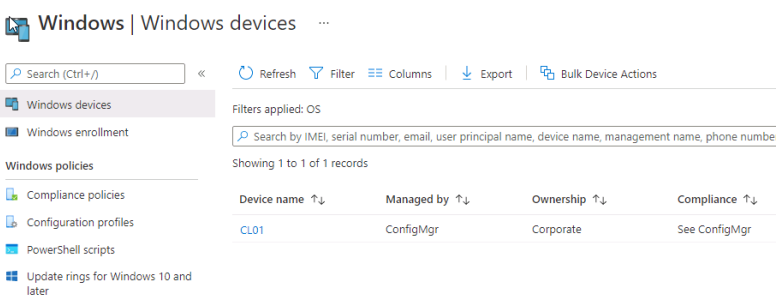
11. Leave co-management settings unconfigured and click **Next**
12. Leave workloads settings unconfigured and click **Next**
13. Leave staging settings unconfigured and click **Next**

14. Click **Next** and **Close.**

15. Review the dmpuploader.log in E:\Program Files\Microsoft Configuration Manager\Logs

16. Go to CL01, open Microsoft Edge en browse to **https://endpoint.microsoft.com**

17. Browse to **Devices > Windows** and review the devices that have synced to Microsoft Endpoint Manager

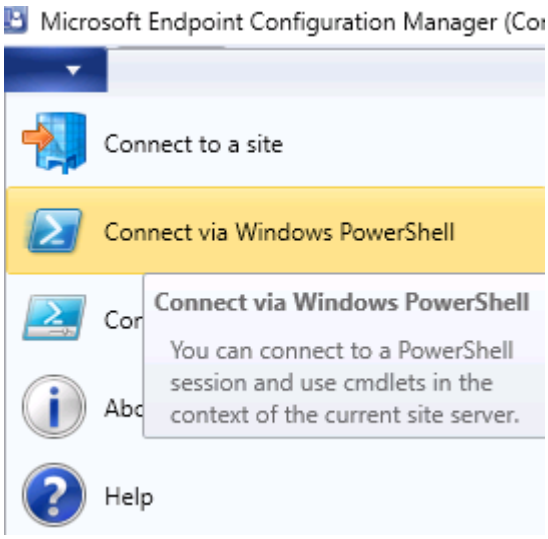
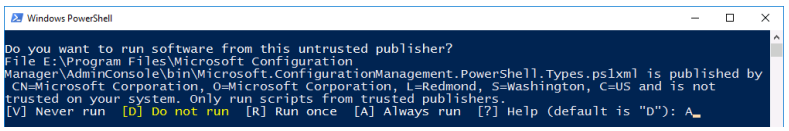


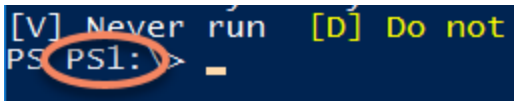
18. Click one of the synced devices and review the options that are shared

Exercise 2 – Configure Co-Management

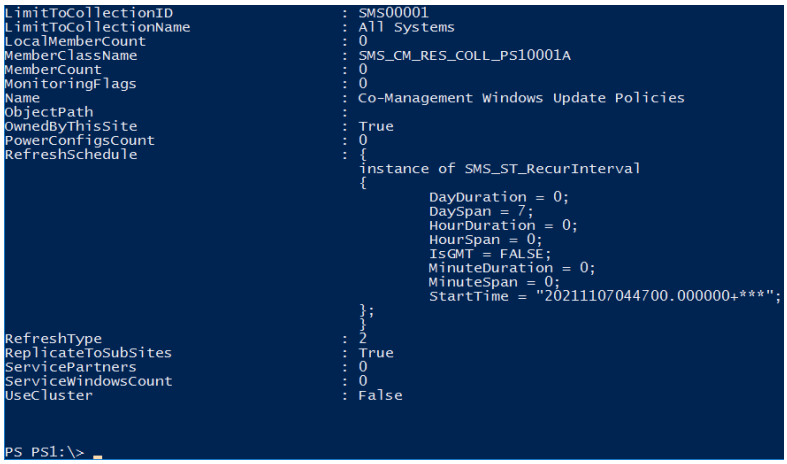
The following steps will create the collections required to transition pilot workloads and enable Co-Management in the Configuration Manager console. There are six potential workloads that can be transitioned to Intune, and this can be enabled for all devices, or a subset through the Pilot option.

In this exercise we will configure the Collections that we will use for the co-management pilot and configure the pilot collections.

Instructions	Screenshot (if applicable)
1. Logon to the CM01 as **VIAMONSTRA\Administrator**	
2. Start the **Configuration Manager console** 3. At the top-left of the Console, click the blue menu button, followed by **Connect via Windows PowerShell**	
4. This will launch a PowerShell session and attempt to connect to the Site Server. When prompted, type **A** to Always Run the script and press **Enter**	

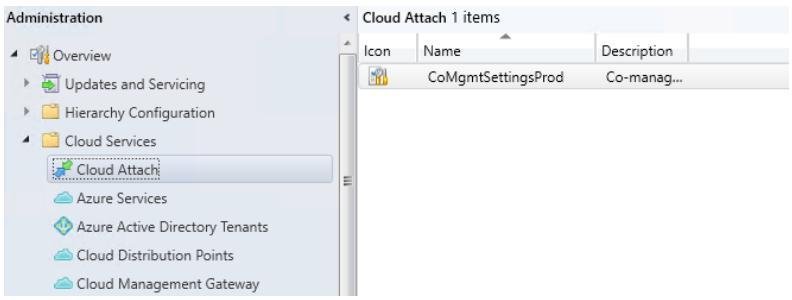
5.	When successfully connected to the Site Server, PowerShell will show the SIteCode as the current directory	
6.	We're going to use PowerShell to create the collections we need to define the co-management workloads. Copy them to your clipboard (**CTRL+C**) and right-click in the PowerShell window to Paste.	New-CMDeviceCollection -Name "Co-Management Compliance Policies" -LimitingCollectionName "All Systems" New-CMDeviceCollection -Name "Co-Management Device Configuration" -LimitingCollectionName "All Systems" New-CMDeviceCollection -Name "Co-Management Endpoint Protection" -LimitingCollectionName "All Systems" New-CMDeviceCollection -Name "Co-Management Resource Access Policies" -LimitingCollectionName "All Systems" New-CMDeviceCollection -Name "Co-Management Client Apps" -LimitingCollectionName "All Systems" New-CMDeviceCollection -Name "Co-Management Office CTR" -LimitingCollectionName "All Systems" New-CMDeviceCollection -Name "Co-Management Windows Update Policies" -LimitingCollectionName "All Systems"

7. As soon as you right-click, PowerShell will come to life and start creating collections. This is because there are Hard Returns in the lines that were pasted. Since there's no Hard Return after the last line, you need to press **Enter** to create the last collection

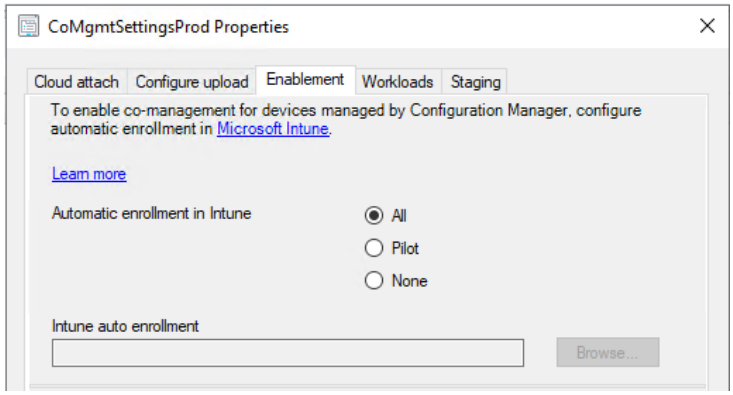


8. Close the PowerShell window

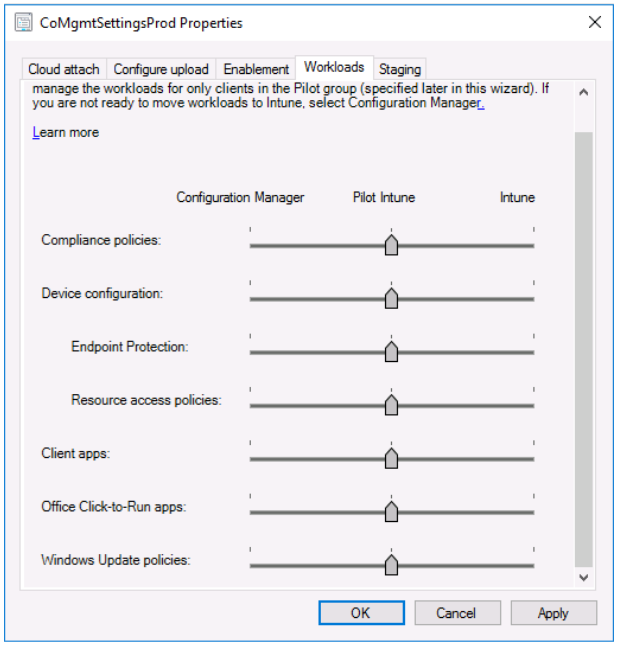
9. Back in the Configuration Manager Console, navigate to **Administration > Cloud services > Cloud Attach**



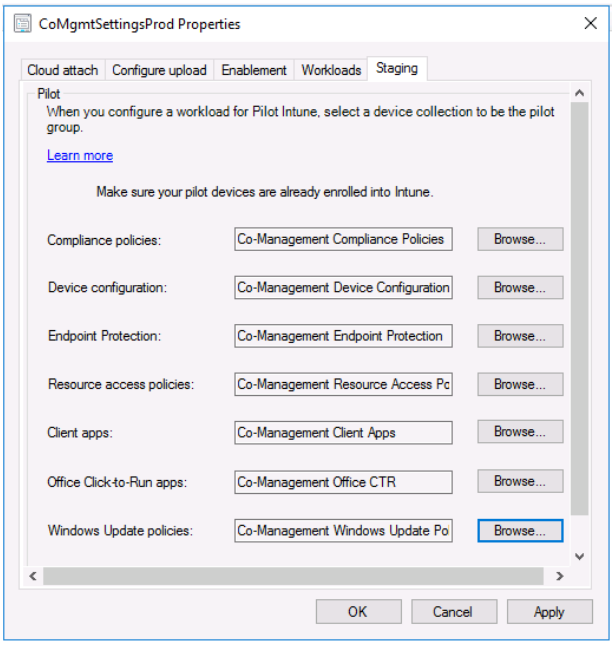
10. Right click **CoMgmtSettingsProd** and click Properties
11. Browse to **Enablement** and select **All** for **Automatic enrollment in Intune.**
12. Click **OK**



13. Click Workloads and switch all of the workloads to **Pilot Intune**



14. Click the **Staging tab** and configure the colletions as shown in the figure.
15. Click **OK**.



Make sure that before all of the workloads are shifted to Microsoft Intune that all the configuration in Microsoft Intune needs to be completed to take over the management.

If you want to for instance migrate your group policies to Microsoft Intune, use the Group Policy Analyzer in Microsoft Endpoint Manager to evaluate the group policies and to see if there are equivalent policies available in Microsoft Endpoint Manager.

Also domain-level Group Policy takes precedence over Intune policies when there is a conlict. There is a CSP for Windows to configure the ControlPolicyConflict setting. Find more on this here: https://docs.microsoft.com/en-us/windows/client-management/mdm/policy-csp-controlpolicyconflict

Exercise 3 – Configure Conditional Access

Even though Conditional Access is accessible from the Intune console, it is not an Intune feature, but that of Azure Active Directory. The policies created within are evaluated whenever a user or device attempts to access the environment or organizational data and will ensure that they are compliant before access is granted. Users and devices are not required to be enrolled in Intune for Conditional Access policies to take effect.

When an organization is first exploring Office365 they will identify access issues that are created by having a collaboration system that's accessible off-network, and Conditional Access enables the creation of policies that ensure access restrictions are maintained despite where data is stored or where it's potentially accessible from.

CA policies are all-encompassing, and it is easily possible to lock all users and administrators out of an Azure tenant. This is where Break Glass accounts come in. Break Glass accounts are accounts with specific access and permissions that allow them to be used to gain access to Azure Active Directory in the event a policy has been created in error.

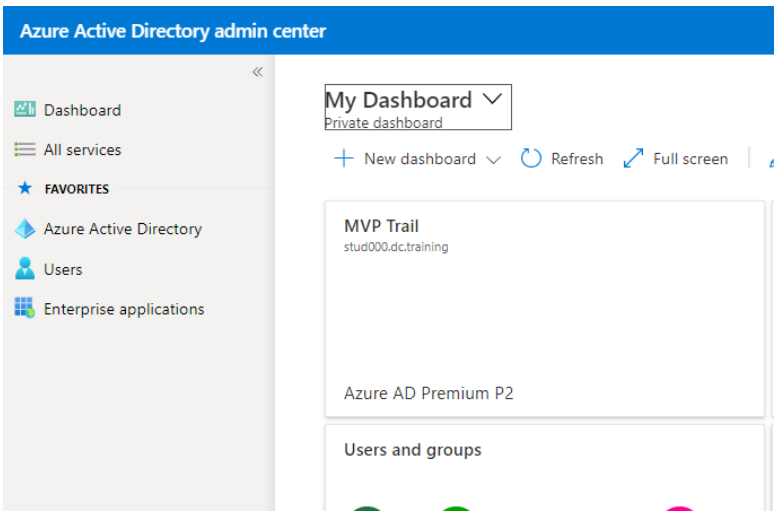
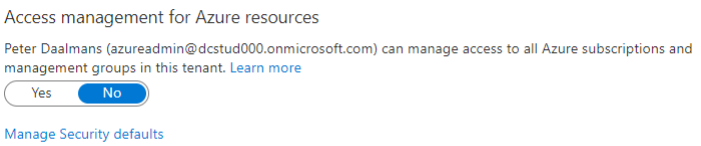
To further reduce the possibility of locking ourselves out of the tenant, we'll stick to the recommended policies…we know they work and they're what most people use to sufficiently restrict access to their Azure tenant.

By default Security Defaults are enabled by Microsoft to help customers to make sure that a minimum set of security measurements is enabled on the tenant. Security defaults will take care of the following:

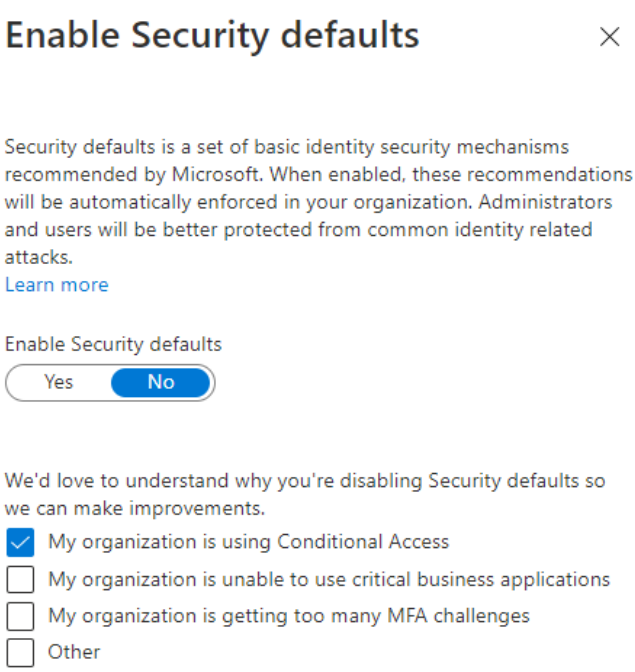
- Protecting administrators - add MFA for users member of the following roles:
    - Global administrator
    - Application administrator
    - Authentication administrator
    - Billing administrator
    - Cloud application administrator
    - Conditional Access administrator
    - Exchange administrator
    - Helpdesk administrator
    - Password administrator
    - Privileged authentication administrator
    - Security administrator
    - SharePoint administrator

    - User administrator
- Protecting all users
- Blocking legacy authentication
- Protecting privileged actions (access via Powershell, Azure CLI and Azure Portal)

Before you are able to configure Conditional Access, Security defaults need to be disabled.

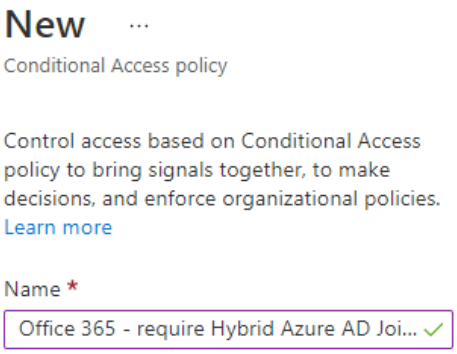
Instructions	Screenshot (if applicable)
1. Logon to the CL01 as VIAMONSTRA\Administrator	
2. Open Microsoft Edge en browse to **https://aad.portal.azure.com** and login with your **AzureAdmin**	
3. Browse to **Azure Active Directory > Properties** 4. Click **Properties** 5. Click **Manage Security defaults**	

6. Select **No** to disable the Security defaults and select **My organization is using Conditional Access**

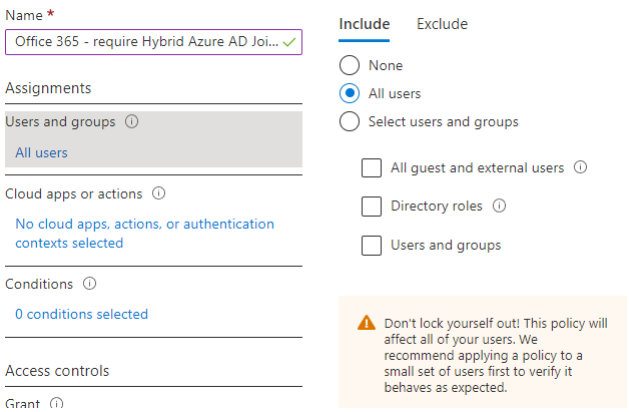


7. Click **Save**

8. To create a Conditional Access rule, go to **Azure Active Directory > Security > Conditional Access**
9. Click **New Policy.**
10. Supply a **Name** for the Policy, (e.g. Office 365 – require Hybrid Azure AD Joined or Compliant Devices)

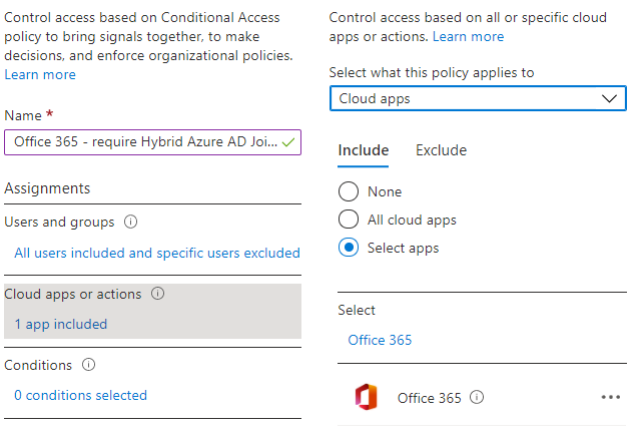


11. Click **Users and Groups**, select **All Users.**
12.

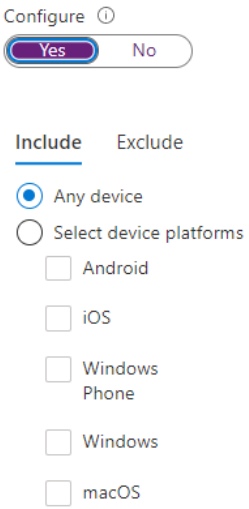


13. Click **Exclude,** and select **Select users and groups** and click **Select excluded users.**
14. For demo purpose search and select your **Azure Admin. (today this can be your break the glass account)**
15. Click **Select**

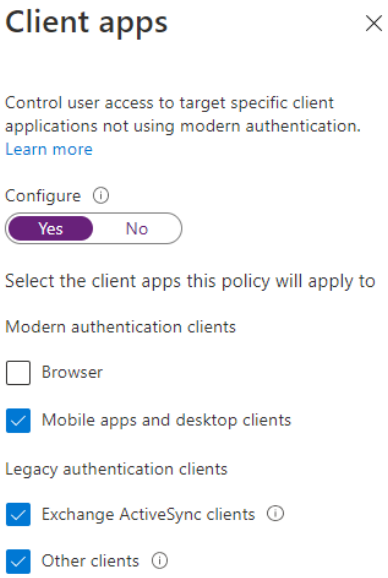
16. Click **Cloud Apps or actions** and select **Select apps.**
17. Click **Select** and search for **Office 365** and select the group by **enabling the checkbox** of **Office 365**.
18. Click **Select**.



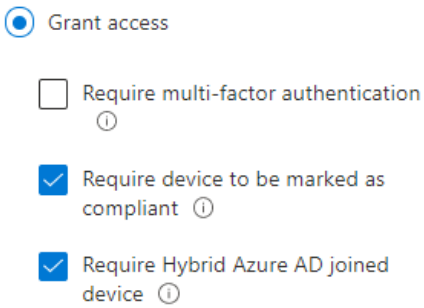
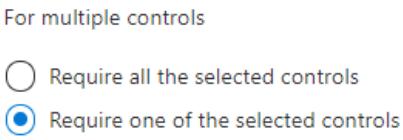
19. Click **Conditions**, **User risk** and **Sign-in risk**, review the settings and leave default and click Select. (Azure Identity Protection integrates with Azure AD Conditional Access)
20. Click **Device Platforms**, enable condition by set Configure to **Yes** and let **Any device** to be selected.
21. Click **Done**



22. Click **Locations**, leave default.
23. Click **Client apps**, **enable** condition by set Configure to **Yes**
24. **Uncheck Browser** and leave Mobile apps and desktop clients selected as default and click **Done**.



25. Click **Device state (preview)**, leave default and click **Done**

26. Click **Device filters**, leave default and click **Done**	
27. Click **Grant** under Access Controls and select **Require device to be marked as compliant** and **Require Hybrid Azure Adjoined device**.	
28. Enable **Require one of the selected controls.** 29. Click **Select**. 30. Click **On** to Enable the policy. 31. Click **Create**.	

Read more on common conditional access policies here: https://docs.microsoft.com/en-us/azure/active-directory/conditional-access/concept-conditional-access-policy-common

Exercise 4 – Enforcing MEM Device Compliance

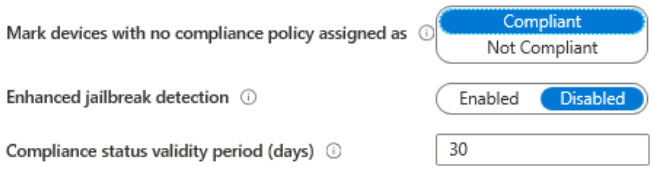
Device compliance is handled very differently between Configuration Manager and Intune, and where the workload should be managed becomes an easy decision once we look into how each service operates.

Instructions	Screenshot (if applicable)

1. On **CL01** open Microsoft Endpoint Manager admin center https://endpoint.microsoft.com/ and login with the Global Administrator of your tenant.

2. Click **Devices** > **Compliance Policies** and click **Compliance Policy Settings.**

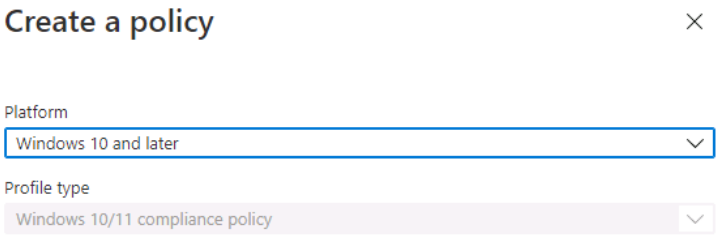
3. Configure that devices with no compliance policy assigned are marked as **Not Compliant**.



4. Click **Save**

5. Still in the Microsoft Endpoint Manager admin center, click **Devices** > **Compliance Policies**.

6. To create a Compliance policy for the Windows 10 devices, click **Create Policy**
7. Select **Windows 10 and later** as the **platform** and click **Create**.
8.



9. Configure the Policy as shown in the figure on the right side.

Section	*Setting*	*value*
Basics	Name	Windows 10 Compliance Policy
Device Health	Require BitLocker	**Not configured**
Device Properties	Minimum OS version	10.0
Configuration Manager compliance	Require device compliance from Configuration Manager	Require
System Security	Require password to unlock mobile device	Require
	Simple passwords	Block
	Minimum password length	4
	Firewall	Require
	Antivirus	Require
	AntiSpyWare	Require
	Windows Defender Antimalware	Require
	Windows Defender Antimalware signature up-to-date	Require
	Real-time protection	Require
Assignments	Included groups	All Users

10. Click **Create** to create the policy.

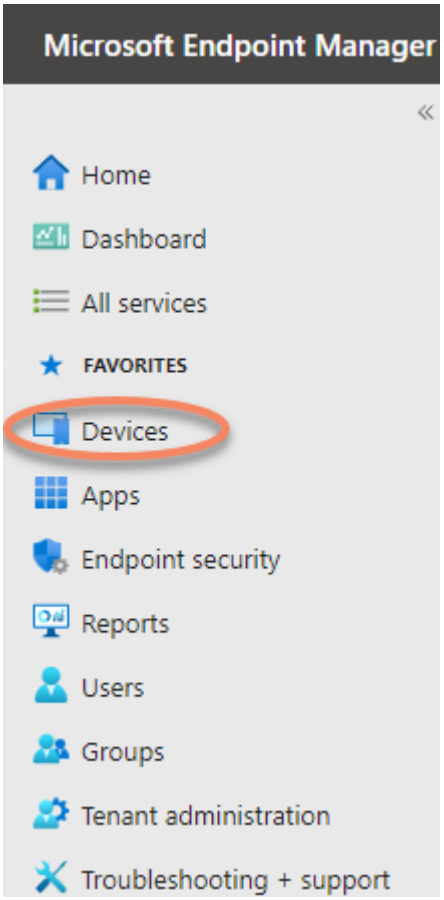
Exercise 5 – Enabling Windows Update for Business

Windows Update for Business is a great option for organizations that are only concerned about scheduling the installation of updates and not controlling the update list for each maintenance cycle. It uses the same Windows Update engine as consumer devices, however adheres to the settings defined in WUfB (pronounced WOOF-b) to ensure updates/upgrades are not installed before they are properly tested in the enterprise.

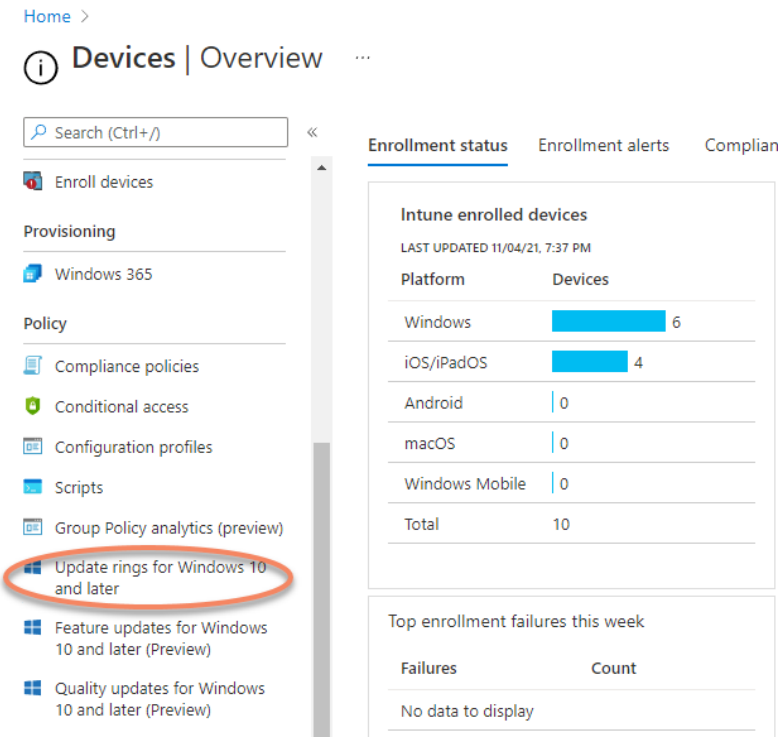
Instructions	Screenshot (if applicable)
1. From a web browser, log into the Intune	

portal at https://endpoint.microsoft.com

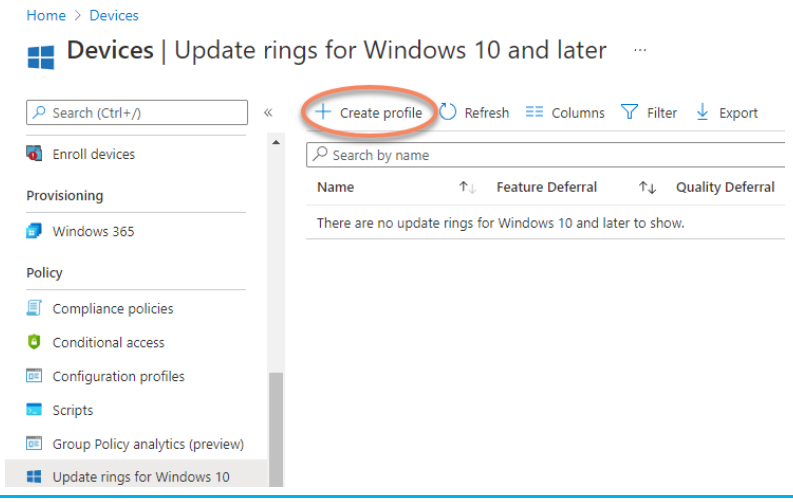
2. In the left-hand navigation pane, click **Devices**



3. In the navigation pane of the Devices blade, click **Update rings for Windows 10 and later**

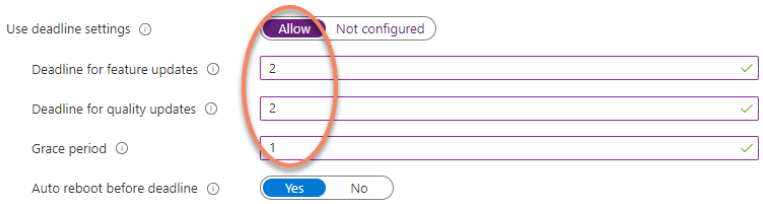


4. This will bring you to the Update rings blade, where we will specify the rings we want to use in the enviornment. Click **Create Profile** at the top

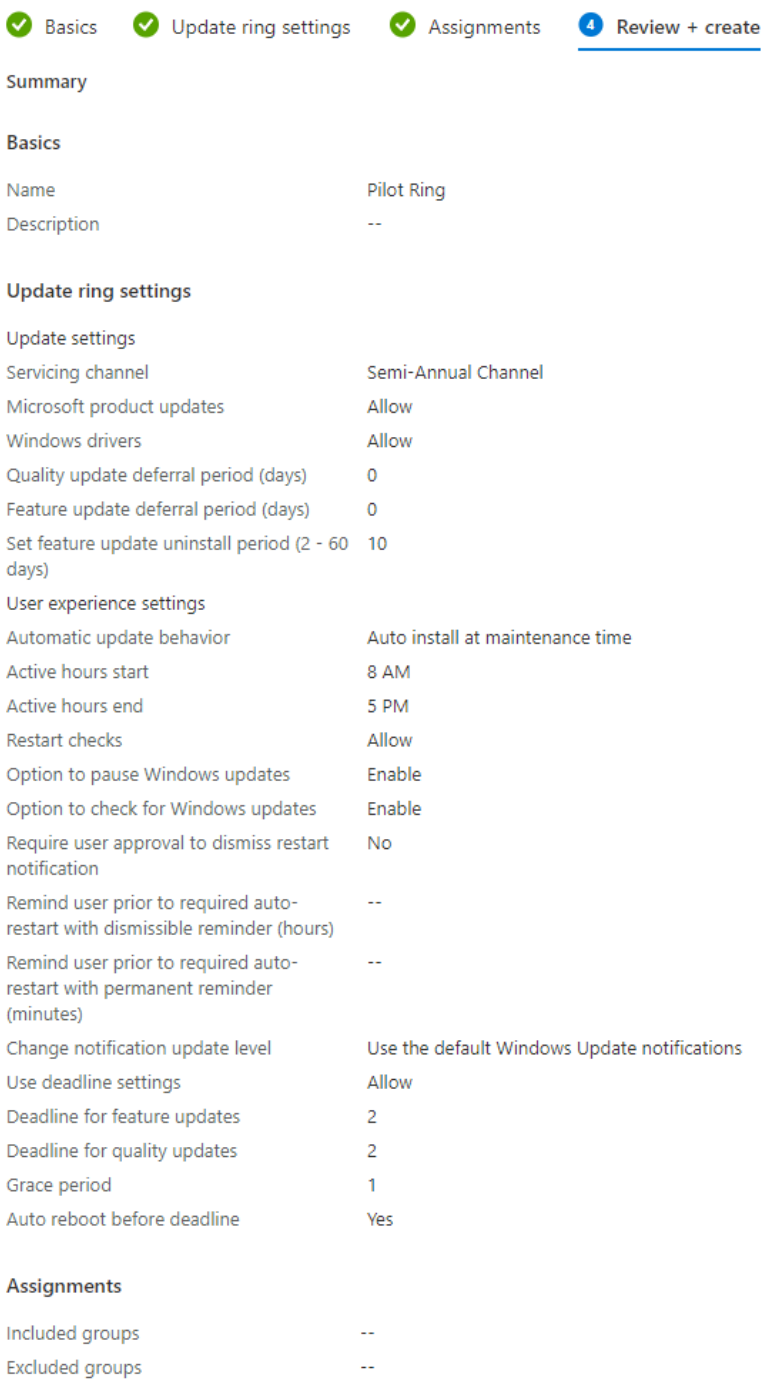


5. In the wizard, call it **Pilot Ring** and click next.

6. On the Update ring settings page, scroll to the bottom. Toggle the “Use deadline settings” switch to **Allow** and set a **2 day deadline** for feature and quality updates, and a **1 day grace period**



7. Click Next, Next, and Create. Once we have defined a list of pilot devices, we will add them to the Pilot Patching group and assign that group to this profile.



Basics | Update ring settings | Assignments | 4 Review + create

Summary

Basics

Name	Pilot Ring
Description	--

Update ring settings

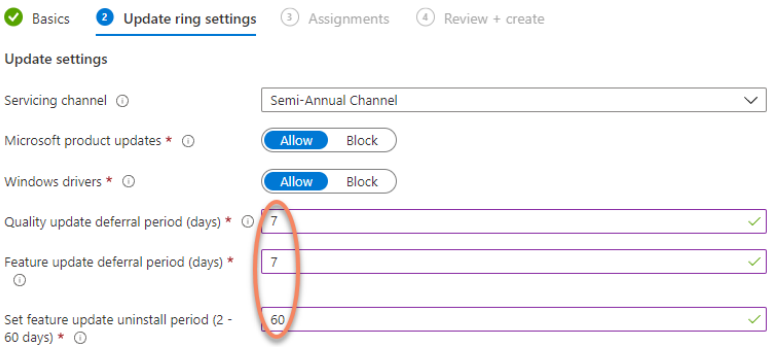
Update settings	
Servicing channel	Semi-Annual Channel
Microsoft product updates	Allow
Windows drivers	Allow
Quality update deferral period (days)	0
Feature update deferral period (days)	0
Set feature update uninstall period (2 - 60 days)	10
User experience settings	
Automatic update behavior	Auto install at maintenance time
Active hours start	8 AM
Active hours end	5 PM
Restart checks	Allow
Option to pause Windows updates	Enable
Option to check for Windows updates	Enable
Require user approval to dismiss restart notification	No
Remind user prior to required auto-restart with dismissible reminder (hours)	--
Remind user prior to required auto-restart with permanent reminder (minutes)	--
Change notification update level	Use the default Windows Update notifications
Use deadline settings	Allow
Deadline for feature updates	2
Deadline for quality updates	2
Grace period	1
Auto reboot before deadline	Yes

Assignments

Included groups	--
Excluded groups	--

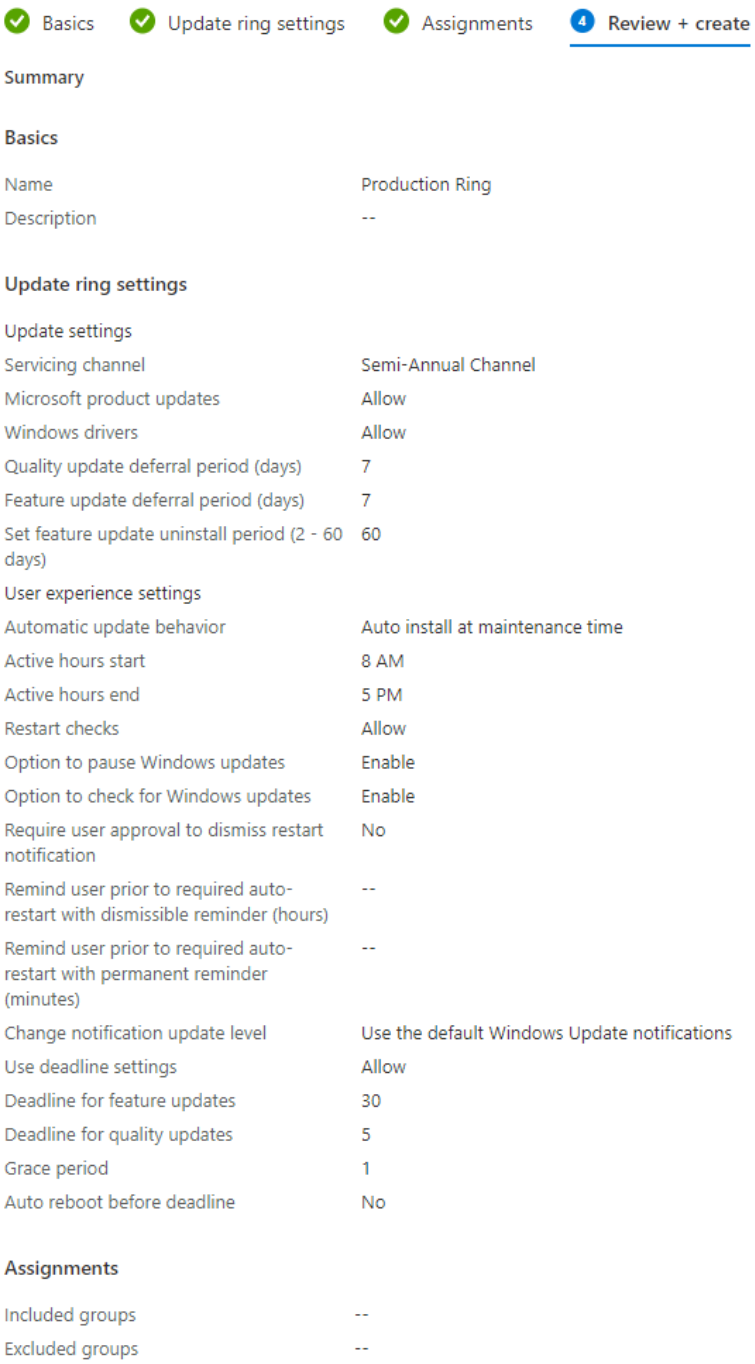
8. Create another Update ring profile, this time for **Production Ring**.

9. This time, we're going to define a **7 day deferral** for both quality and feature updates, and allow the uninstall of feature updates for **60 days**

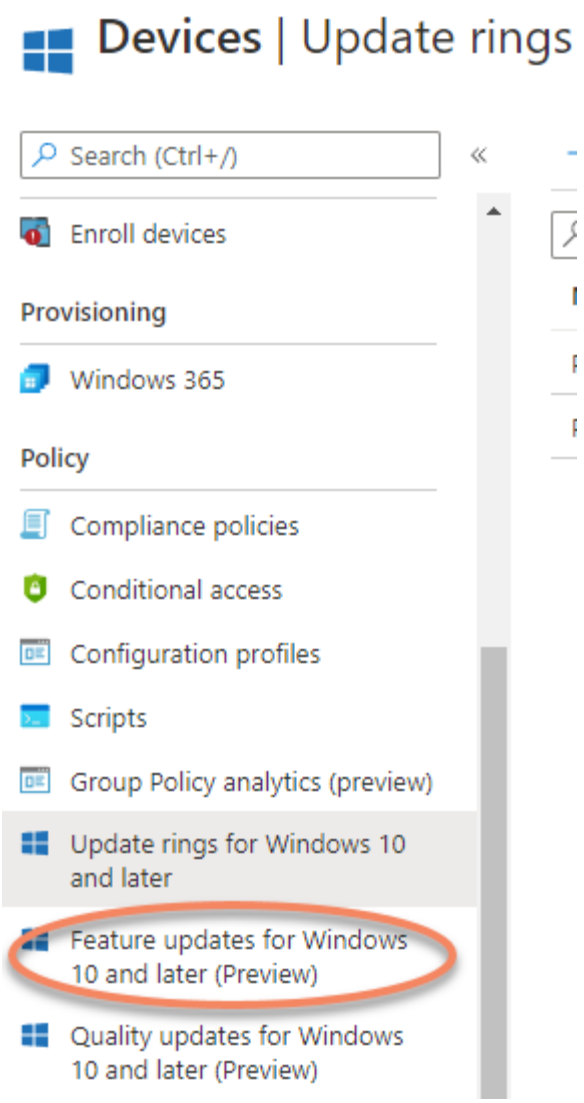


10. Scroll to the bottom and configure the deadline settings. For the production ring, we will configure a **30-day deadline** for feature updates, a **5-day deadline** for quality updates, and a **1-day grace** period. We'll also change the "Auto reboot before deadline" setting to **No**

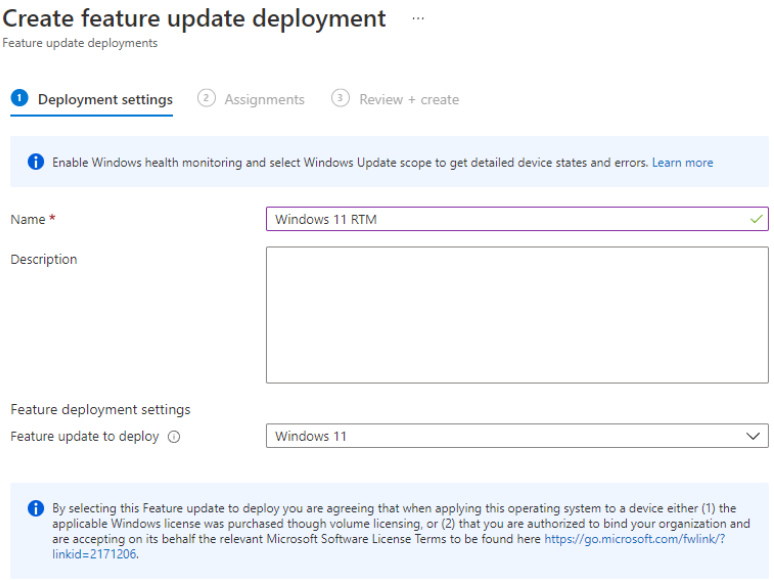
11. We won't assign a group to this ring yet. Once we have defined a list of pilot devices, we will add the remaining devices to the Production Patching group and assign that group to this profile. Compare the settings with the screenshot and finish creating the policy.



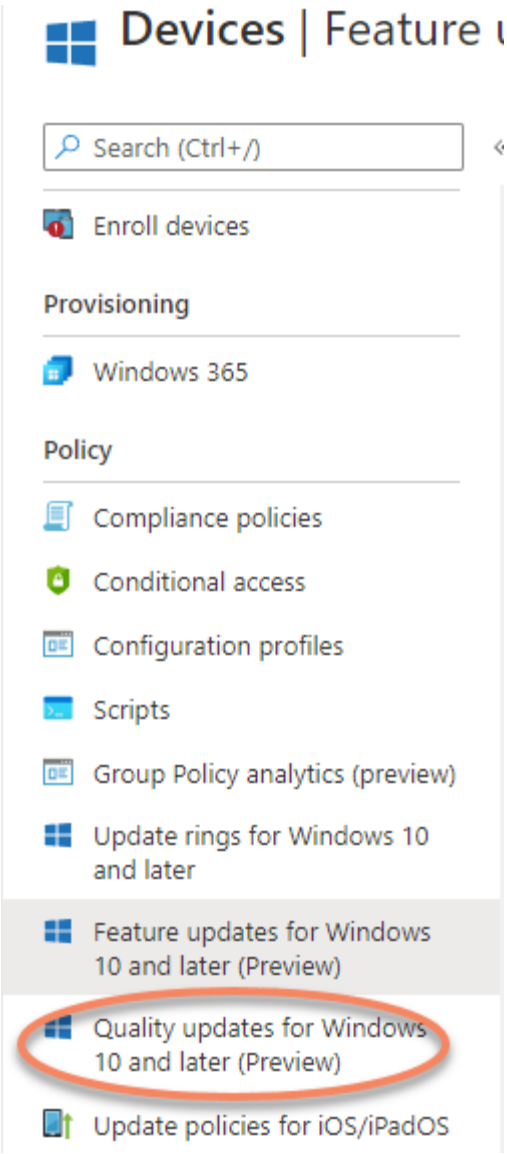
12. In the navigation pane of the Devices blade, just underneath the Update rings, click **Feature updates for Windows 10 and later**



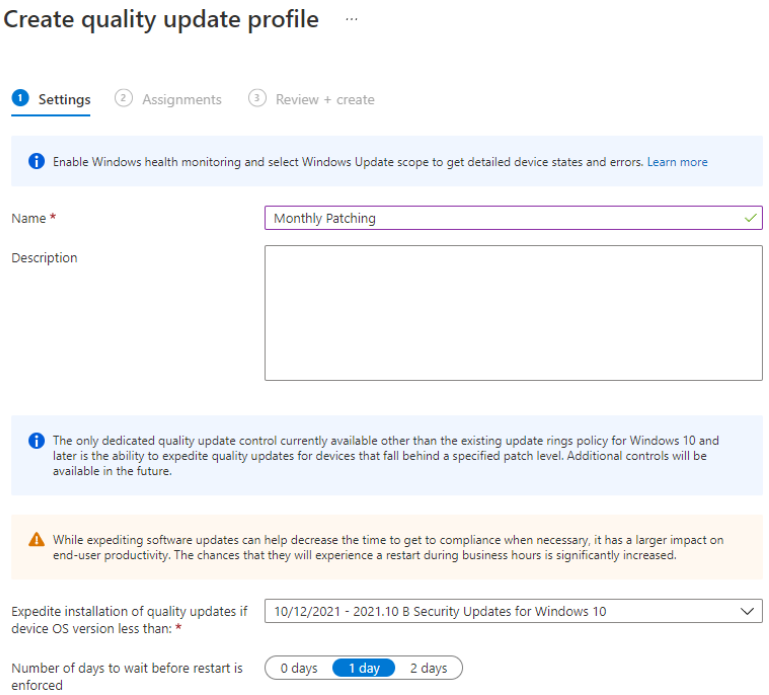
13. Create a policy, and call it Windows 11 RTM. Confirm that **Windows 11** is selected for the "Feature update to deploy" and click Next, Next, Create



14. In the navigation pane of the Devices blade, just underneath the “Feature updates for Windows 10 and later” link, click **Quality updates for Windows 10 and later**



15. Create a profile, call it **Monthly Patching**. Take notice of the options at the bottom of the window, but do not modify them

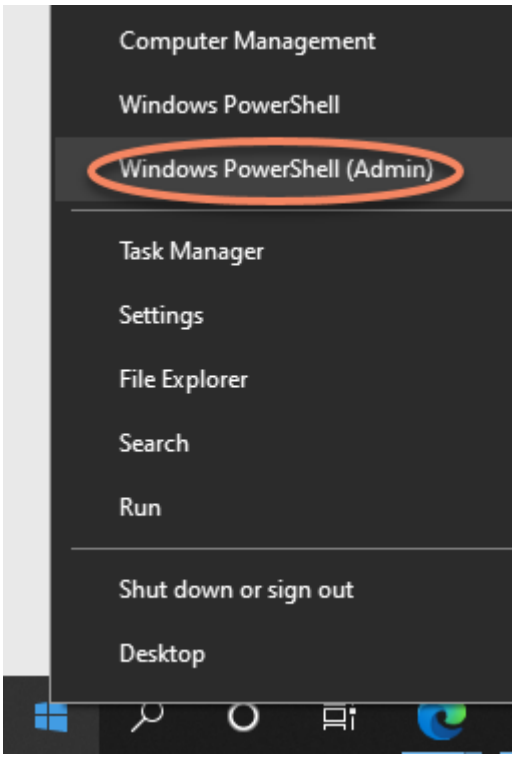


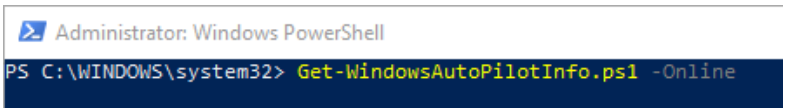
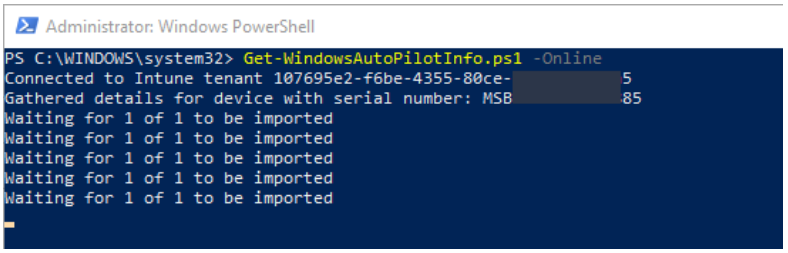
16. Click Next, Next, and Create to complete creating the Patching profile

Exercise 6 – Configuring Windows AutoPilot

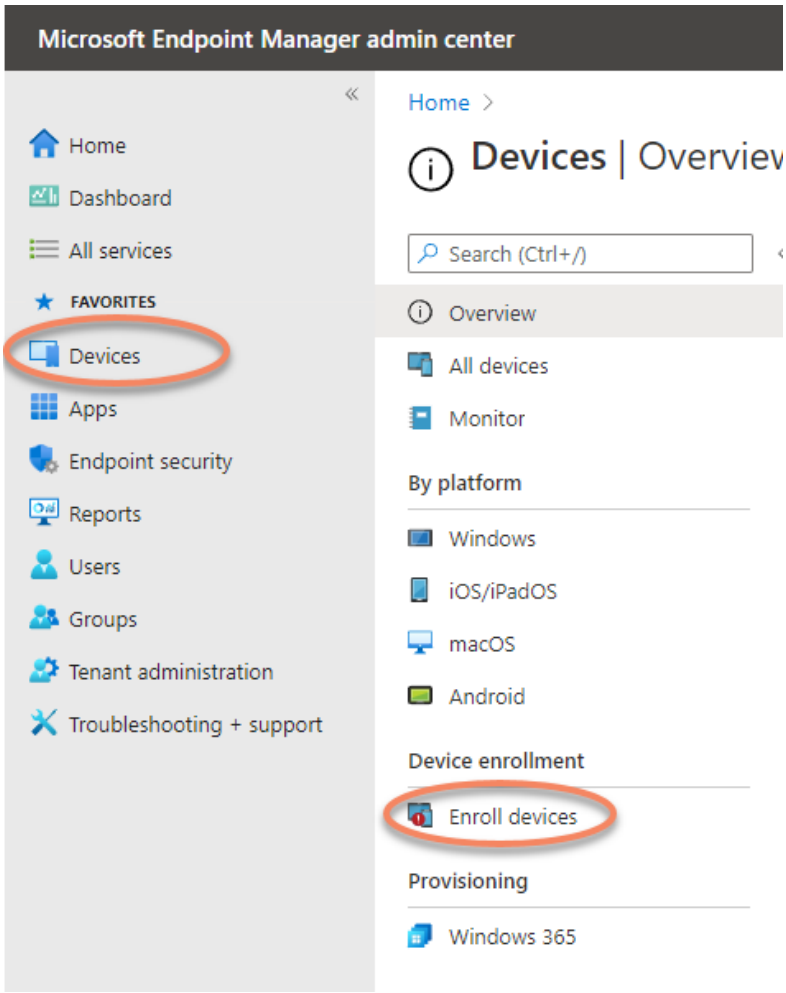
Windows Autopilot is a feature of Intune that facilitates the initial provisioning of a new or refreshed device. It leverages the current installed operating system as a base install and performs a series of actions to configure the device for corporate use. It has become the practical method for initial device configuration of Windows devices performing the initial set up at home or off the corporate network.

Instructions	Screenshot (if applicable)

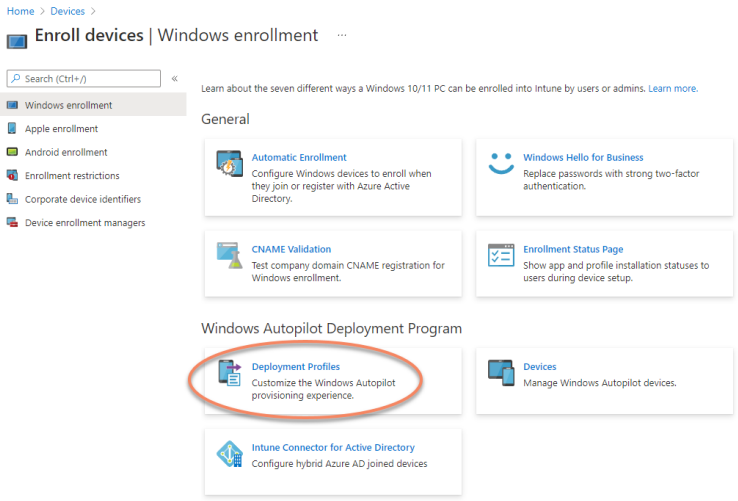
1. Log into the CL01 workstation	
2. Right-click on the Start button and select **Windows PowerShell (Admin)**	
3. In the PowerShell window, type the following commands:	**Set-ExecutionPolicy Bypass** **Install-script Get-WindowsAutopilotInfo**
4. This will connect to the PowerShell Gallery and attempt to install the NuGet provider and the Get-	

	WindowsAutopilotInfo PowerShell Module. If prompted, choose **Yes, Run, Always Run, or Always Allow** as appropriate	
5.	When complete, type the command listed to the right. You may use PowerShell's Tab-Complete feature to ensure that the script is properly loaded	**Get-WindowsAutopilotinfo.ps1 -Online** 
6.	When you press Enter, this will initiate an authentication prompt. Log in with your Azure Lab credentials. The script will then connect to Intune using the GraphAPI and import the hardware information for the workstation	
7.	From a web browser, log into the Intune portal at https://endpoint.microsoft.com	

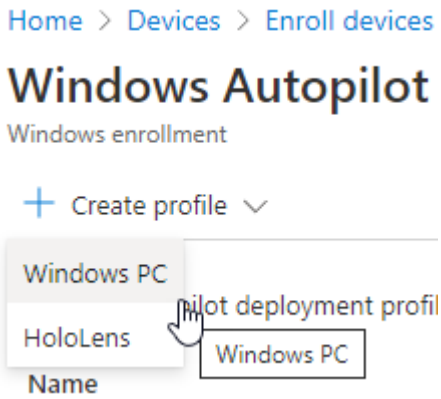
8. In the left-hand navigation pane, click **Devices**. In the Devices blade, click **Enroll Devices**



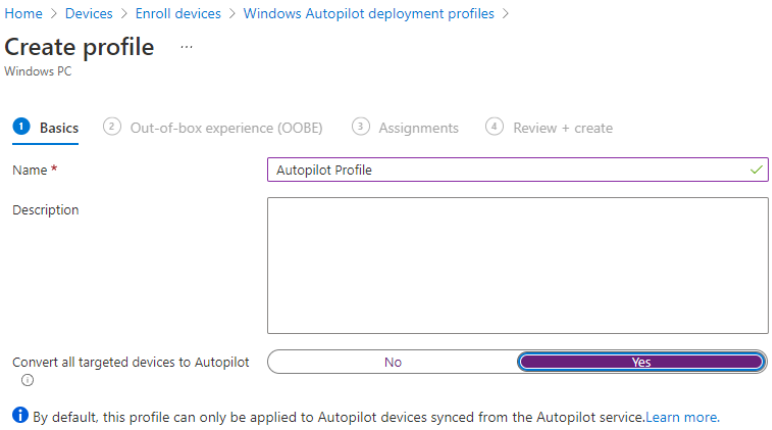
9. In the Enroll Devices blade, click the **Deployment Profiles** tile under “Windows Autopilot Deployment Program”



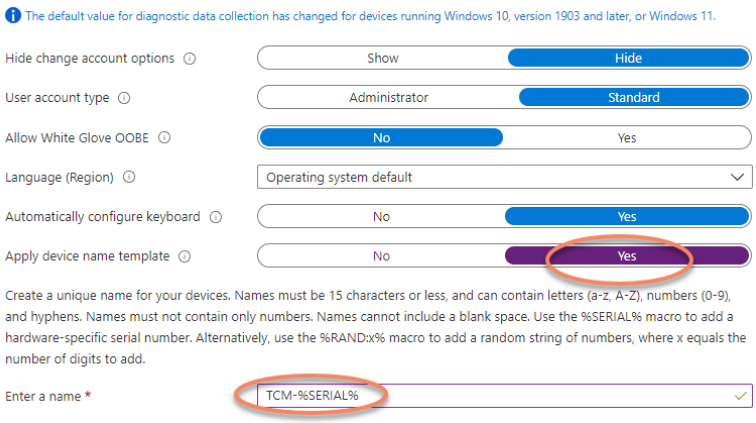
10. Next, click **Create Profile** and choose **Windows PC**



11. On the Create Profile, give it the creative name of **Autopilot Profile**, and change the “Convert all targeted devices to Autopilot” slider to **Yes**



12. We will only make one customization to the Out-of-Box Environment, and that is to use a workstation naming convention. Change the “Apply device name template” slider to Yes, and specify **TCM-%SERIAL%** for the name



13. We want Autopilot profiles to be automatically assigned, which we still need to figure out. For now, skip assigning the profile and just create it. The summary screen should now look like this:

Basics

Name	Autopilot Profile
Description	--
Convert all targeted devices to Autopilot	Yes
Device type	Windows PC

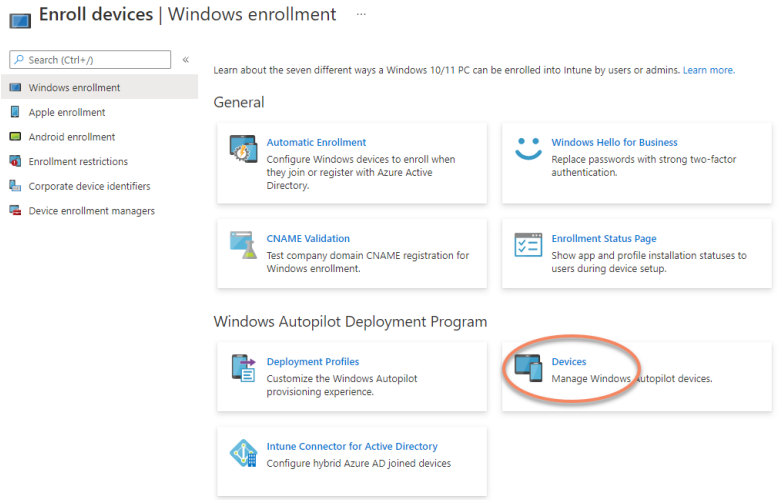
Out-of-box experience (OOBE)

Deployment mode	User-Driven
Join to Azure AD as	Azure AD joined
Skip AD connectivity check (preview)	No
Language (Region)	Operating system default
Automatically configure keyboard	Yes
Microsoft Software License Terms	Hide
Privacy settings	Hide
Hide change account options	Hide
User account type	Standard
Allow White Glove OOBE	No
Apply device name template	Yes
Enter a name	TCM-%SERIAL%

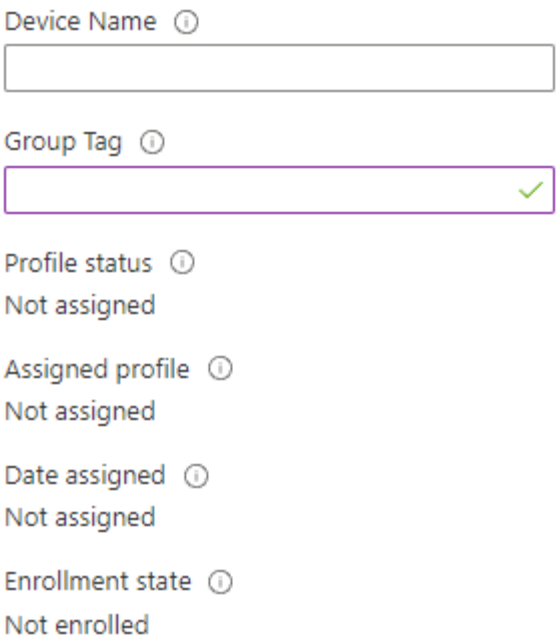
Assignments

Included groups	--
Excluded groups	--

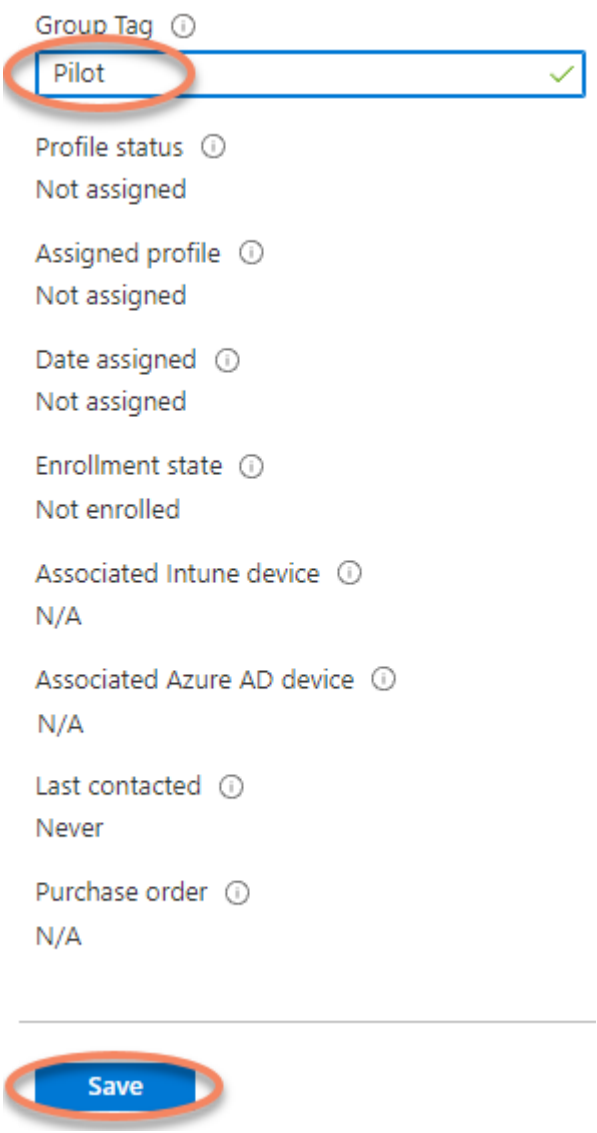
14. Back in the Enroll Devices blade, click on the **Devices** tile



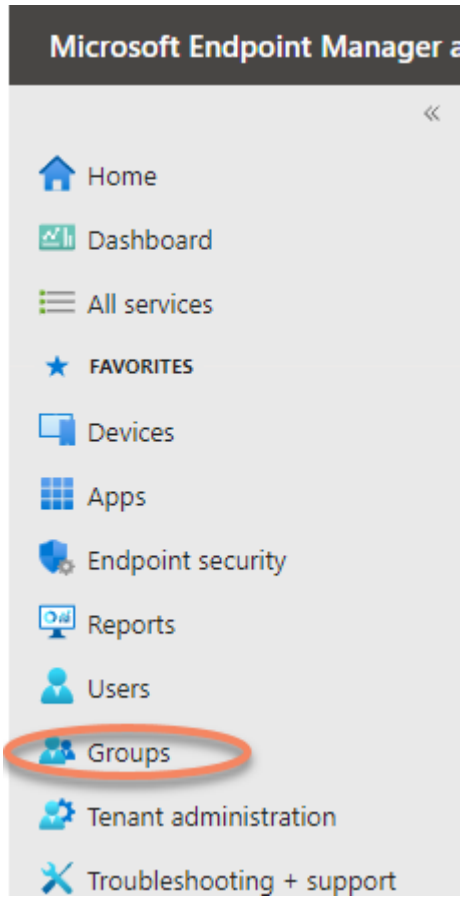
15. Select the device that was just imported, and examine the details on the right



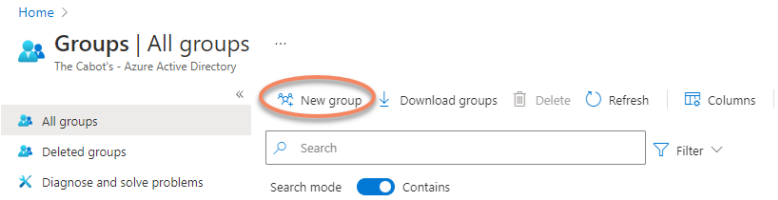
16. When we import a device into Autopilot, we have the option to define a group tag. We're going to create a Dynamic Group in AAD that adds as a member any device that has a group tag of "Pilot." To ensure the machine gets added to that group, as we didn't include it with the upload, type **Pilot** in the Group Tag field for the device, and click **Save**



17. In the far-left navigation pane, click **Groups**



18. Create a new group called **Autopilot Devices**. The Group type is **Security** and the Membership type is **Dynamic Device**

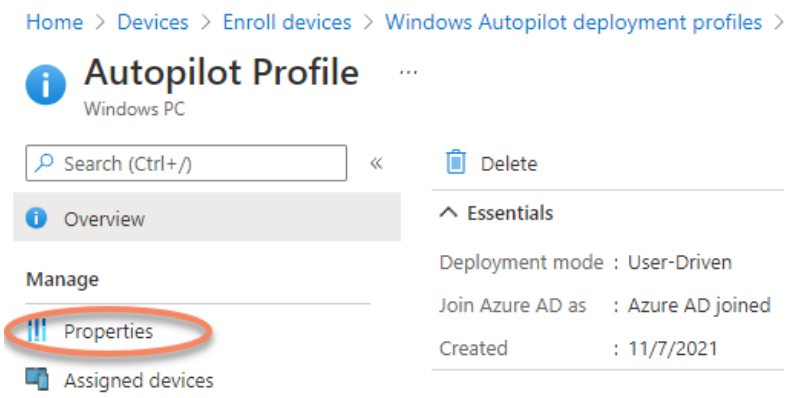


19. Under Dynamic device members, select **Add Dynamic Query**, **Add Expression**. The

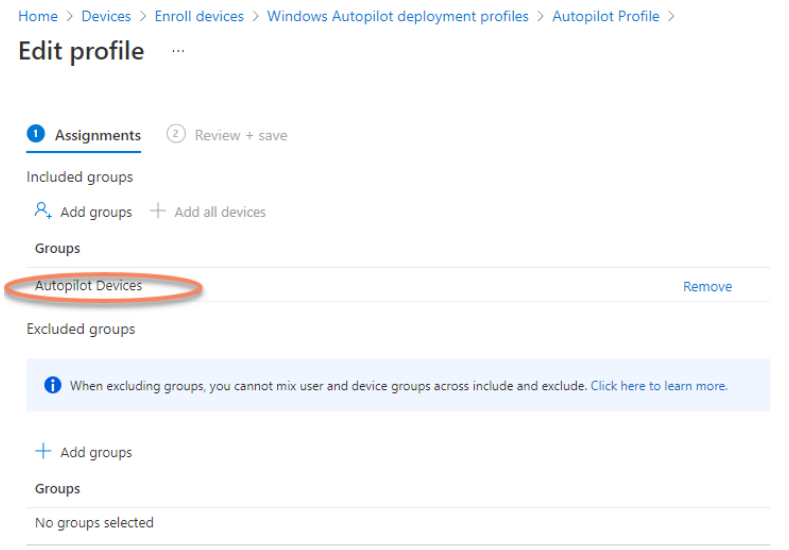
(device.devicePhysicalIds -any _ –startsWith "[OrderID]:Pilot")

Intune Group Tag attribute pairs with "OrderId" in AAD, and this method will allow you to use multiple Autopilot profiles. Use the following query:

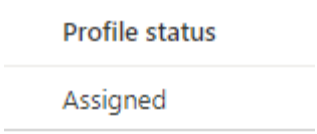
20. Back in the Enroll Devices blade, click the **Deployment Profiles** tile and go into the properties of the **Autopilot profile** we created earlier



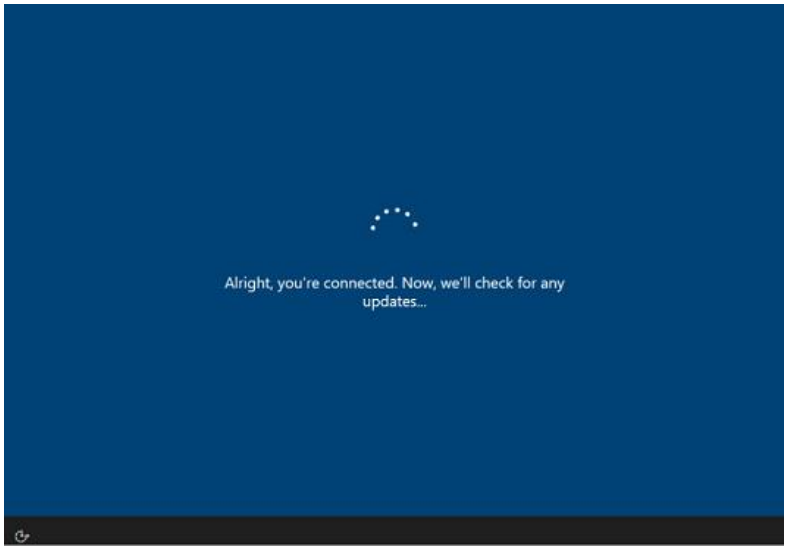
21. Edit the Assignments and add **Autopilot Devices** to the Included groups



22. Now, if we go back to Devices, Enroll Devices, and click the Devices tile once more, we should now see the machine with a Profile assigned. *Note this may take up to 15 minutes



23. That's it! For Autopilot to work, the device needs to be at the OOBE setup screen, similar to a brand-new PC. When this profile is applied to existing machines, Autopilot will be initiated if/when the device is reset to factory defaults.



Chapter 7 – CIS and Microsoft Premiere

CIS®

The Center for Internet Security, Inc. (CIS®) makes the connected world a safer place for people, businesses, and governments through our core competencies of collaboration and innovation.

They are a community-driven nonprofit, responsible for the CIS Controls® and CIS Benchmarks™, globally recognized best practices for securing IT systems and data. They lead a global community of IT professionals to continuously evolve these standards and provide products and services to proactively safeguard against emerging threats. CIS Hardened Images® provide secure, on-demand, scalable computing environments in the cloud.

CIS is home to the Multi-State Information Sharing and Analysis Center® (MS-ISAC®), the trusted resource for cyber threat prevention, protection, response, and recovery for U.S. State, Local, Tribal, and Territorial government entities, and the Elections Infrastructure Information Sharing and Analysis Center® (EI-ISAC®), which supports the rapidly changing cybersecurity needs of U.S. elections offices.

Specifically for device management, membership to CIS provides administrators with spreadsheets, templates, documentation, etc. to assist in implementing a level of controls in the organization. The ability to directly import a policy that is both functional and contains the appropriate level of configuration to enforce a benchmark significantly reduces design and testing time when creating these policies.

Microsoft Premier Support

Premier Support for Enterprise offers end-to-end managed support for customers across the spectrum of Microsoft products and services. Tailored to your business priorities, Premier Support for Enterprise helps you realize value on your software investments by minimizing risk and reducing downtime.

As part of the premier support agreement, customers have the ability to request a Risk Assessment Plan be completed for a particular Microsoft Technology. Historically performed by an on-site Microsoft engineer, most (if not all) products can now complete a Risk Assessment Plan as a service, and generate the same level of report through a self-service engine. RAPaaS, specifically for Configuration Manager and Active Directory, are among the best analysis tools in

the world for these products and have the ability to quickly identify areas of security vulnerability or mis-configuration.

Anyone that is currently a Microsoft Premier Support customer and has not performed a RAPaaS for CM, AD, SQL and others, should take the time to investigate this opportunity.

Appendix A

DiskSpd

The information in this appendix is a complete copy from https://docs.microsoft.com/en-us/azure-stack/hci/manage/diskspd-overview (October 2021) and the authors of this book have not contributed to the content contained herein. It has been provided for reference purposes as standard practice when configuring new server hardware.

What is DISKSPD?

DISKSPD is an I/O generating, command-line tool for micro-benchmarking. Great, so what do all these terms mean? Anyone who sets up an Azure Stack HCI cluster or physical server has a reason. It could be to set up a web hosting environment, or run virtual desktops for employees. Whatever the real-world use case may be, you likely want to simulate a test before deploying your actual application. However, testing your application in a real scenario is often difficult – this is where DISKSPD comes in.

DISKSPD is a tool that you can customize to create your own synthetic workloads, and test your application before deployment. The cool thing about the tool is that it gives you the freedom to configure and tweak the parameters to create a specific scenario that resembles your real workload. DISKSPD can give you a glimpse into what your system is capable of before deployment. At its core, DISKSPD simply issues a bunch of read and write operations.

Now you know what DISKSPD is, but when should you use it? DISKSPD has a difficult time emulating complex workloads. But DISKSPD is great when your workload is not closely approximated by a single-threaded file copy, and you need a simple tool that produces acceptable baseline results.

Quick start: install and run DISKSPD

Without further ado, let's get started:

1. From your management PC, open PowerShell as an administrator to connect to the target computer that you want to test using DISKSPD, and then type the following command and press Enter.

PowerShellCopy

Enter-PSSession -ComputerName <TARGET_COMPUTER_NAME>

In this example, we're running a virtual machine (VM) called "node1."

2. To download the DISKSPD tool, type the following commands and press Enter:

PowerShellCopy

$client = new-object System.Net.WebClient

PowerShellCopy

$client.DownloadFile("https://github.com/microsoft/diskspd/releases/download/v2.0.21a/DiskSpd.zip","<ENTER_PATH>\DiskSpd-2.0.21a.zip")

3. Use the following command to unzip the downloaded file:

PowerShellCopy
Expand-Archive -LiteralPath <ENTERPATH>\DiskSpd-2.0.21a.zip -DestinationPath C:\DISKSPD

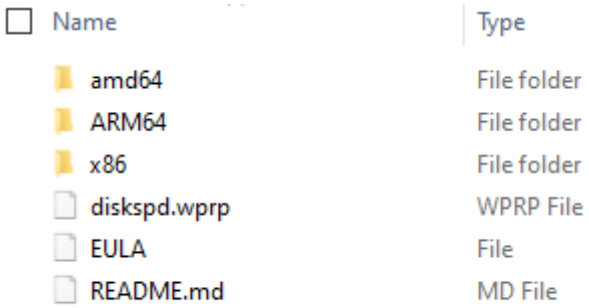
4. Change directory to the DISKSPD directory and locate the appropriate executable file for the Windows operating system that the target computer is running.

In this example, we're using the amd64 version.

Note

You can also download the DISKSPD tool directly from the **GitHub repository** that contains the open-source code, and a wiki page that details all the parameters and specifications. In the repository, under **Releases**, select the link to automatically download the ZIP file.

In the ZIP file, you'll see three subfolders: amd64 (64-bit systems), x86 (32-bit systems), and ARM64 (ARM systems). These options enable you to run the tool in every Windows client or server version.



5. Run DISKSPD with the following PowerShell command. Replace everything inside the square brackets, including the brackets themselves with your appropriate settings.

PowerShellCopy
.\[INSERT_DISKSPD_PATH] [INSERT_SET_OF_PARAMETERS] [INSERT_CSV_PATH_FOR_TEST_FILE] > [INSERT_OUTPUT_FILE.txt]

Here is an example command that you can run:

PowerShellCopy
.\diskspd -t2 -o32 -b4k -r4k -w0 -d120 -Sh -D -L -c5G C:\ClusterStorage\test01\targetfile\IO.dat > test01.txt

Note

If you do not have a test file, use the **-c** parameter to create one. If you use this parameter, be sure to include the test file name when you define your path. For example: [INSERT_CSV_PATH_FOR_TEST_FILE] = C:\ClusterStorage\CSV01\IO.dat. In the example command, IO.dat is the test file name, and test01.txt is the DISKSPD output file name.

Specify key parameters

Well, that was simple right? Unfortunately, there is more to it than that. Let's unpack what we did. First, there are various parameters that you can tinker with and it can get specific. However, we used the following set of baseline parameters:

Note

DISKSPD parameters are case sensitive.

-t2: This indicates the number of threads per target/test file. This number is often based on the number of CPU cores. In this case, two threads were used to stress all of the CPU cores.

-o32: This indicates the number of outstanding I/O requests per target per thread. This is also known as the queue depth, and in this case, 32 were used to stress the CPU.

-b4K: This indicates the block size in bytes, KiB, MiB, or GiB. In this case, 4K block size was used to simulate a random I/O test.

-r4K: This indicates the random I/O aligned to the specified size in bytes, KiB, MiB, Gib, or blocks (Overrides the **-s** parameter). The common 4K byte size was used to properly align with the block size.

-w0: This specifies the percentage of operations that are write requests (-w0 is equivalent to 100% read). In this case, 0% writes were used for the purpose of a simple test.

-d120: This specifies the duration of the test, not including cool-down or warm-up. The default value is 10 seconds, but we recommend using at least 60 seconds for any serious workload. In this case, 120 seconds were used to minimize any outliers.

-Suw: Disables software and hardware write caching (equivalent to **-Sh**).

-D: Captures IOPS statistics, such as standard deviation, in intervals of milliseconds (per-thread, per-target).

-L: Measures latency statistics.

-c5g: Sets the sample file size used in the test. It can be set in bytes, KiB, MiB, GiB, or blocks. In this case, a 5 GB target file was used.

For a complete list of parameters, refer to the GitHub repository.

Understand the environment

Performance heavily depends on your environment. So, what is our environment? Our specification involves an Azure Stack HCI cluster with storage pool and Storage Spaces Direct (S2D). More specifically, there are five VMs: DC, node1, node2, node3, and the management node. The cluster itself is a three-node cluster with a three-way mirrored resiliency structure. Therefore, three data copies are maintained. Each "node" in the cluster is a Standard_B2ms VM with a maximum IOPS limit of 1920. Within each node, there are four premium P30 SSD drives with a maximum IOPS limit of 5000. Finally, each SSD drive has 1 TB of memory.

You generate the test file under the unified namespace that the Cluster Shared Volume (CSV) provides (C:\ClusteredStorage) to use the entire pool of drives.

Note

The example environment does *not* have Hyper-V or a nested virtualization structure.

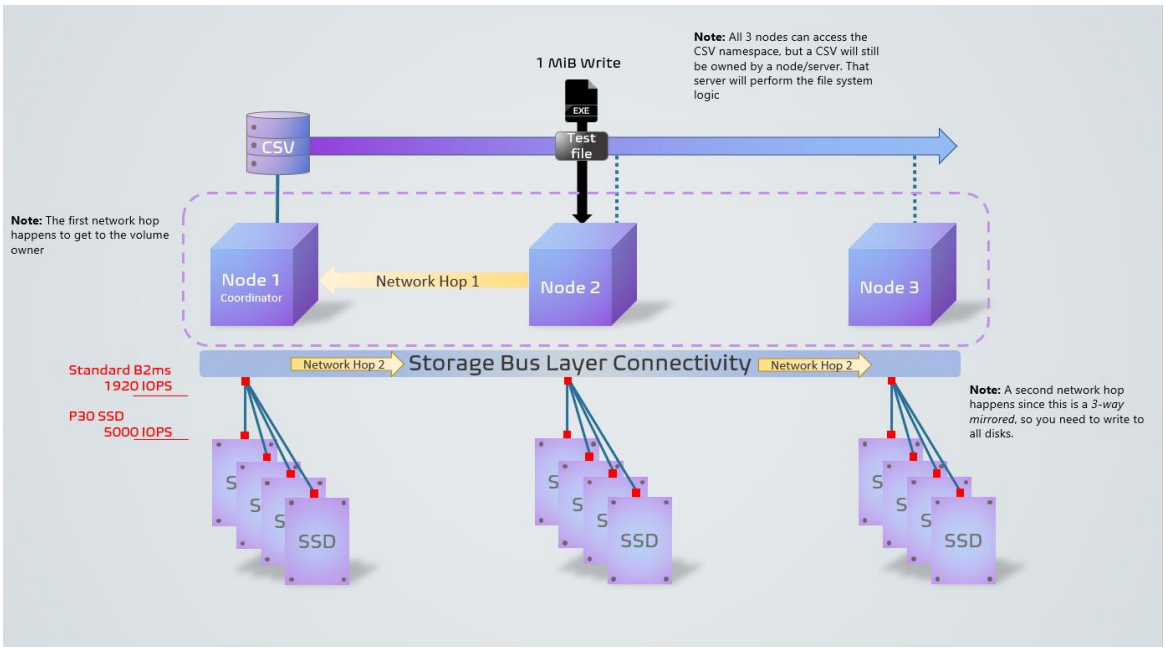
As you'll see, it's entirely possible to independently hit either the IOPS or bandwidth ceiling at the VM or drive limit. And so, it is important to understand your VM size and drive type, because both have a maximum IOPS limit and a bandwidth ceiling. This knowledge helps to locate bottlenecks and understand your performance results. To learn more about what size may be appropriate for your workload, see the following resources:

- VM sizes
- Disk types

Understand the output

Armed with your understanding of the parameters and environment, you're ready to interpret the output. First, the goal of the earlier test was to max out the IOPS with no regard to latency. This way, you can visually see whether you reach the artificial IOPS limit within Azure. If you want to graphically visualize the total IOPS, use either Windows Admin Center or Task Manager.

The following diagram shows what the DISKSPD process looks like in our example environment. It shows an example of a 1 MiB write operation from a non-coordinator node. The three-way resiliency structure, along with the operation from a non-coordinator node, leads to two network hops, decreasing performance. If you're wondering what a coordinator node is, don't worry! You'll learn about it in the Things to consider section. The red squares represent the VM and drive bottlenecks.



Now that you've got a visual understanding, let's examine the four main sections of the .txt file output:

1. Input settings

This section describes the command you ran, the input parameters, and additional details about the test run.

```
Command Line: C:\DiskSpd-2.0.21a\amd64\diskspd.exe -t2 -o32 -b4k -r4k -w0 -d120 -Sh -D -L C:\ClusterStorage\test01\targetfile\IO_5g.dat

Input parameters:

        timespan:   1
        ------------
        duration: 120s
        warm up time: 5s
        cool down time: 0s
        measuring latency
        gathering IOPS at intervals of 1000ms
        random seed: 0
        path: 'C:\ClusterStorage\test01\targetfile\IO_5g.dat'
                think time: 0ms
                burst size: 0
                software cache disabled
                hardware write cache disabled, writethrough on
                performing read test
                block size: 4096
                using random I/O (alignment: 4096)
                number of outstanding I/O operations: 32
                thread stride size: 0
                threads per file: 2
                using I/O Completion Ports
                IO priority: normal
```

2. CPU utilization details

This section highlights information such as the test time, number of threads, number of available processors, and the average utilization of every CPU core during the test. In this case, there are two CPU cores that averaged around 4.67% usage.

```
actual test time:       120.01s
thread count:           2
proc count:             2

CPU |  Usage |  User  |  Kernel |  Idle
-------------------------------------------
   0|   5.89%|   0.77%|    5.12%|  94.11%
   1|   3.45%|   0.69%|    2.76%|  96.55%
-------------------------------------------
avg.|   4.67%|   0.73%|    3.94%|  95.33%
```

3. Total I/O

This section has three subsections. The first section highlights the overall performance data including both read and write operations. The second and third sections split the read and write operations into separate categories.

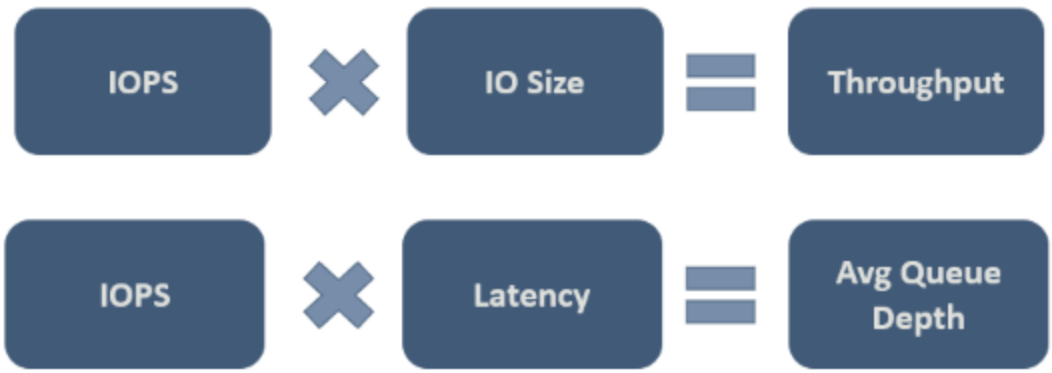
In this example, you can see that the total I/O count was 234408 during the 120-second duration. Thus, IOPS = 234408 /120 = 1953.30. The average latency was 32.763 milliseconds, and the throughput was 7.63 MiB/s. From earlier information, we know that the 1953.30 IOPS are near the 1920 IOPS limitation for our Standard_B2ms VM. Don’t believe it? If you rerun this test using different parameters, such as increasing the queue depth, you'll find that the results are still capped at this number.

The last three columns show the standard deviation of IOPS at 17.72 (from -D parameter), the standard deviation of the latency at 20.994 milliseconds (from -L parameter), and the file path.

```
Total IO
thread |       bytes     |     I/Os     |    MiB/s   |  I/O per s |  AvgLat  | IopsStdDev | LatStdDev |  file
------------------------------------------------------------------------------------------------------------------
     0 |       479649792 |       117102 |       3.81 |     975.80 |   32.794 |      12.43 |    21.000 | C:\ClusterStorage\test01\targetfile\IO_5g.dat (5120MiB)
     1 |       480485376 |       117306 |       3.82 |     977.50 |   32.732 |      14.98 |    20.987 | C:\ClusterStorage\test01\targetfile\IO_5g.dat (5120MiB)
------------------------------------------------------------------------------------------------------------------
total:         960135168 |       234408 |       7.63 |    1953.30 |   32.763 |      17.72 |    20.994

Read IO
thread |       bytes     |     I/Os     |    MiB/s   |  I/O per s |  AvgLat  | IopsStdDev | LatStdDev |  file
------------------------------------------------------------------------------------------------------------------
     0 |       479649792 |       117102 |       3.81 |     975.80 |   32.794 |      12.43 |    21.000 | C:\ClusterStorage\test01\targetfile\IO_5g.dat (5120MiB)
     1 |       480485376 |       117306 |       3.82 |     977.50 |   32.732 |      14.98 |    20.987 | C:\ClusterStorage\test01\targetfile\IO_5g.dat (5120MiB)
------------------------------------------------------------------------------------------------------------------
total:         960135168 |       234408 |       7.63 |    1953.30 |   32.763 |      17.72 |    20.994

Write IO
thread |       bytes     |     I/Os     |    MiB/s   |  I/O per s |  AvgLat  | IopsStdDev | LatStdDev |  file
------------------------------------------------------------------------------------------------------------------
     0 |               0 |            0 |       0.00 |       0.00 |    0.000 |       0.00 |       N/A | C:\ClusterStorage\test01\targetfile\IO_5g.dat (5120MiB)
     1 |               0 |            0 |       0.00 |       0.00 |    0.000 |       0.00 |       N/A | C:\ClusterStorage\test01\targetfile\IO_5g.dat (5120MiB)
------------------------------------------------------------------------------------------------------------------
total:                 0 |            0 |       0.00 |       0.00 |    0.000 |       0.00 |       N/A
```

From the results, you can quickly determine that the cluster configuration is terrible. You can see that it hit the VM limitation of 1920 before the SSD limitation of 5000. If you were limited by the SSD rather than the VM, you could have taken advantage of up to 20000 IOPS (4 drives * 5000) by spanning the test file across multiple drives.

In the end, you need to decide what values are acceptable for your specific workload. The following figure shows some important relationships to help you consider the tradeoffs:



The second relationship in the figure is important, and it's sometimes referred to as Little’s Law. The law introduces the idea that there are three characteristics that govern process behavior and that you only need to change one to influence the other two, and thus the entire process. And so, if you're unhappy with your system’s performance, you have three dimensions of freedom to influence it. Little's Law dictates that in our example, IOPS is the "throughput" (input output operations per second), latency is the "queue time", and queue depth is the "inventory".

4. Latency percentile analysis

This last section details the percentile latencies per operation type of storage performance from the minimum value to the maximum value.
This section is important because it determines the “quality” of your IOPS. It reveals how many of the I/O operations were able to achieve a certain latency value. It's up to you to decide the acceptable latency for that percentile.
Moreover, the “nines” refer to the number of nines. For example, “3-nines” is equivalent to the 99th percentile. The number of nines exposes how many I/O operations ran at that percentile. Eventually, you'll reach a point where it no longer makes sense to take the latency values seriously. In this case, you can see that the latency values remain constant after “4-nines.” At this point, the latency value is based on only one I/O operation out of the 234408 operations.

```
total:
  %-ile |  Read (ms) | Write (ms) | Total (ms)
----------------------------------------------
    min |      2.711 |        N/A |      2.711
   25th |      3.475 |        N/A |      3.475
   50th |      3.983 |        N/A |      3.983
   75th |     19.848 |        N/A |     19.848
   90th |     24.306 |        N/A |     24.306
   95th |     24.730 |        N/A |     24.730
   99th |     25.537 |        N/A |     25.537
3-nines |     33.300 |        N/A |     33.300
4-nines |     55.296 |        N/A |     55.296
5-nines |     81.921 |        N/A |     81.921
6-nines |     81.921 |        N/A |     81.921
7-nines |     81.921 |        N/A |     81.921
8-nines |     81.921 |        N/A |     81.921
9-nines |     81.921 |        N/A |     81.921
    max |     81.921 |        N/A |     81.921
```

Things to consider
Now that you've started using DISKSPD, there are several things to consider to get real-world test results. These include paying close attention to the parameters you set, storage space health and variables, CSV ownership, and the difference between DISKSPD and file copy.
DISKSPD vs. real-world
DISKSPD’s artificial test gives you relatively comparable results for your real workload. However, you need to pay close attention to the parameters you set and whether they match your real scenario. It's important to understand that synthetic workloads will never perfectly represent your application’s real workload during deployment.
Preparation
Before running a DISKSPD test, there are a few recommended actions. These include verifying the health of the storage space, checking your resource usage so that another program doesn't interfere with the test, and preparing performance manager if you want to collect additional data. However, because the goal of this topic is to quickly get DISKSPD running, it doesn't dive

into the specifics of these actions. To learn more, see Test Storage Spaces Performance Using Synthetic Workloads in Windows Server.

Variables that affect performance

Storage performance is a delicate thing. Meaning, there are many variables that can affect performance. And so, it's likely you may encounter a number that is inconsistent with your expectations. The following highlights some of the variables that affect performance, although it's not a comprehensive list:

- Network bandwidth
- Resiliency choice
- Storage disk configuration: NVME, SSD, HDD
- I/O buffer
- Cache
- RAID configuration
- Network hops
- Hard drive spindle speeds

CSV ownership

A node is known as a volume owner or the **coordinator** node (a non-coordinator node would be the node that does not own a specific volume). Every standard volume is assigned a node and the other nodes can access this standard volume through network hops, which results in slower performance (higher latency).

Similarly, a Cluster Shared Volume (CSV) also has an “owner.” However, a CSV is “dynamic” in the sense that it will hop around and change ownership every time you restart the system (RDP). As a result, it’s important to confirm that DISKSPD is run from the coordinator node that owns the CSV. If not, you may need to manually change the CSV ownership.

To confirm CSV ownership:

1. Check ownership by running the following PowerShell command:

PowerShellCopy

```
Get-ClusterSharedVolume
```

2. If the CSV ownership is incorrect (For example, you are on Node1 but Node2 owns the CSV), then move the CSV to the correct node by running the following PowerShell command:

PowerShellCopy

```
Get-ClusterSharedVolume <INSERT_CSV_NAME> | Move-ClusterSharedVolume <INSERT
_NODE_NAME>
```

File copy vs. DISKSPD

Some people believe that they can “test storage performance” by copying and pasting a gigantic file and measuring how long that process takes. The main reason behind this approach is most likely because it's simple and fast. The idea is not wrong in the sense that it tests a specific workload, but it's difficult to categorize this method as “testing storage performance.”

If your real-world goal is to test file copy performance, then this may be a perfectly valid reason to use this method. However, if your goal is to measure storage performance, we recommend to not use this method. You can think of the file copy process as using a different set of "parameters" (such as queue, parallelization, and so on) that is specific to file services.
The following short summary explains why using file copy to measure storage performance may not provide the results that you're looking for:

- **File copies might not be optimized,** There are two levels of parallelism that occur, one internal and the other external. Internally, if the file copy is headed for a remote target, the CopyFileEx engine does apply some parallelization. Externally, there are different ways of invoking the CopyFileEx engine. For example, copies from File Explorer are single threaded, but Robocopy is multi-threaded. For these reasons, it's important to understand whether the implications of the test are what you are looking for.
- **Every copy has two sides.** When you simply copy and paste a file, you may be using two disks: the source disk and the destination disk. If one is slower than the other, you essentially measure the performance of the slower disk. There are other cases where the communication between the source, destination, and the copy engine may affect the performance in unique ways.

To learn more, see Using file copy to measure storage performance.

Experiments and common workloads

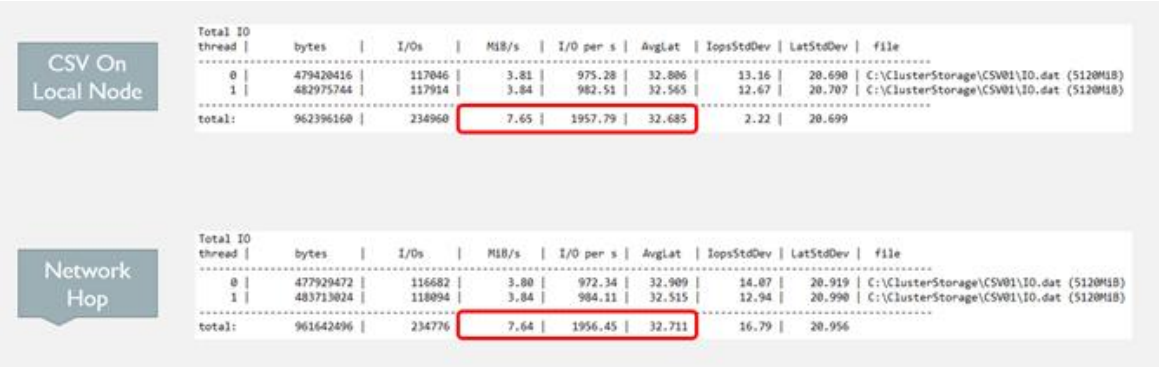
This section includes a few other examples, experiments, and workload types.

Confirming the coordinator node

As mentioned previously, if the VM you are currently testing does not own the CSV, you'll see a performance drop (IOPS, throughput, and latency) as opposed to testing it when the node owns the CSV. This is because every time you issue an I/O operation, the system does a network hop to the coordinator node to perform that operation.
For a three-node, three-way mirrored situation, write operations always make a network hop because it needs to store data on all the drives across the three nodes. Therefore, write operations make a network hop regardless. However, if you use a different resiliency structure, this could change.
Here is an example:

- **Running on local node:** .\DiskSpd-2.0.21a\amd64\diskspd.exe -t4 -o32 -b4k -r4k -w0 -Sh -D -L C:\ClusterStorage\test01\targetfile\IO.dat
- **Running on nonlocal node:** .\DiskSpd-2.0.21a\amd64\diskspd.exe -t4 -o32 -b4k -r4k -w0 -Sh -D -L C:\ClusterStorage\test01\targetfile\IO.dat

From this example, you can clearly see in the results of the following figure that latency decreased, IOPS increased, and throughput increased when the coordinator node owns the CSV.



Online Transaction Processing (OLTP) workload

Online Transactional Processing (OLTP) workload queries (Update, Insert, Delete) focus on transaction-oriented tasks. Compared to Online Analytical Processing (OLAP), OLTP is storage latency dependent. Because each operation issues little I/O, what you care about is how many operations per second you can sustain.

You can design an OLTP workload test to focus on random, small I/O performance. For these tests, focus on how far you can push the throughput while maintaining acceptable latencies.

The basic design choice for this workload test should at a minimum include:

- 8 KB block size => resembles the page size that SQL Server uses for its data files
- 70% Read, 30% Write => resembles typical OLTP behavior

Online Analytical Processing (OLAP) workload

OLAP workloads focus on data retrieval and analysis, allowing users to perform complex queries to extract multidimensional data. Contrary to OLTP, these workloads are not storage latency sensitive. They emphasize queueing many operations without caring much about bandwidth. As a result, OLAP workloads often result in longer processing times.

You can design an OLAP workload test to focus on sequential, large I/O performance. For these tests, focus on the volume of data processed per second rather than the number of IOPS. Latency requirements are also less important, but this is subjective.

The basic design choice for this workload test should at a minimum include:

- 512 KB block size => resembles the I/O size when the SQL Server loads a batch of 64 data pages for a table scan by using the read-ahead technique.
- 1 thread per file => currently, you need to limit your testing to one thread per file as problems may arise in DISKSPD when testing multiple sequential threads. If you use more than one thread, say two, and the **-s** parameter, the threads will begin non-deterministically to issue I/O operations on top of each other within the same location. This is because they each track their own sequential offset.

There are two “solutions” to resolve this issue:

  - The first solution involves using the **-si** parameter. With this parameter, both threads share a single interlocked offset so that the threads cooperatively issue a

single sequential pattern of access to the target file. This allows no one point in the file to be operated on more than once. However, because they still do race each other to issue their I/O operation to the queue, the operations may arrive out of order.

This solution works well if one thread becomes CPU limited. You may want to engage a second thread on a second CPU core to deliver more storage I/O to the CPU system to further saturate it.

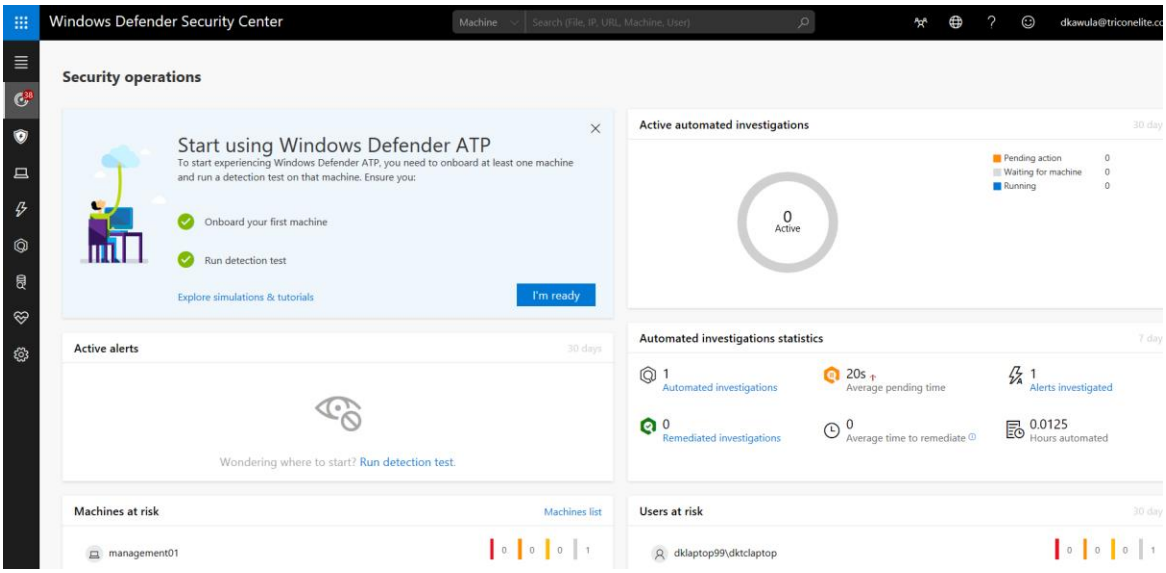
- The second solution involves using the -T<offset>. This allows you to specify the offset size (inter-I/O gap) between I/O operations performed on the same target file by different threads. For example, threads normally start at offset 0, but this specification allows you to distance the two threads so that they will not overlap each other. In any multithreaded environment, the threads will likely be on different portions of the working target, and this is a way of simulating that situation.

Appendix B

Defender Advanced Threat Protection ATP

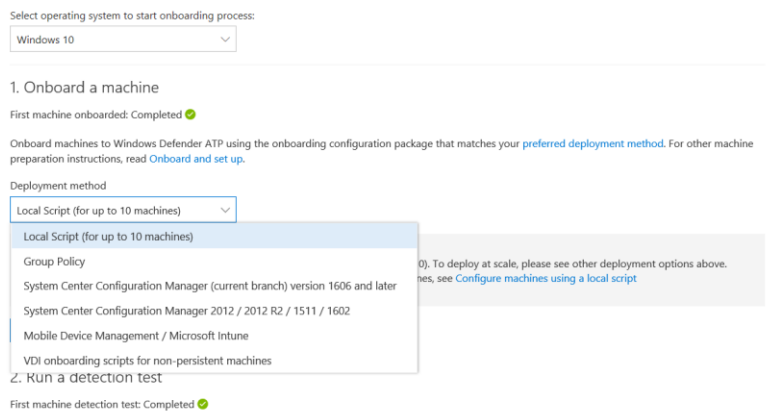
Windows Defender Advanced Threat Protection (ATP) is an extremely useful add-on to help protect your Windows Servers and Workstations. This tool gives the capabilities of Windows Defender that are included with Windows Server 2022 and Windows 11.

Given the number of external connections that Configuration Manager utilizes in its regular operation, having some advanced monitoring on these servers is an exceptionally good idea.

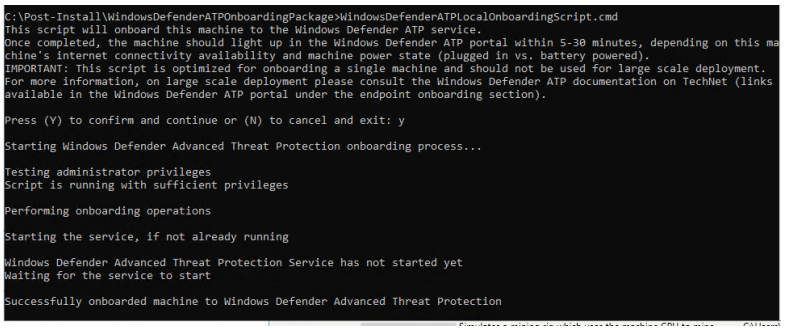


In this chapter, we will give a brief overview of some of the features. To start things off, you will need to sign up for a trial here: https://www.microsoft.com/en-us/WindowsForBusiness/windows-atp?ocid=docs-wdatp-portaloverview-abovefoldlink

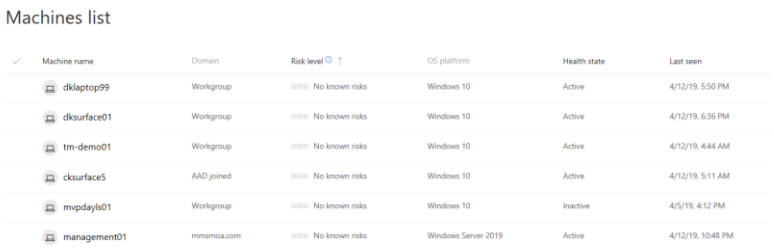
Onboarding a Server with Windows Defender ATP

Instructions	Screenshot (if applicable)
1. Browse to https://securitycenter.windows.com/dashboard 2. Log in with your Admin Credentials	
3. Click on the Settings Wheel, and scroll down to Machine Mangement 4. You will notice that there are many different deployment options from local installation, Group Policy, Configuration Manager, etc. 5. Choose Local Script	

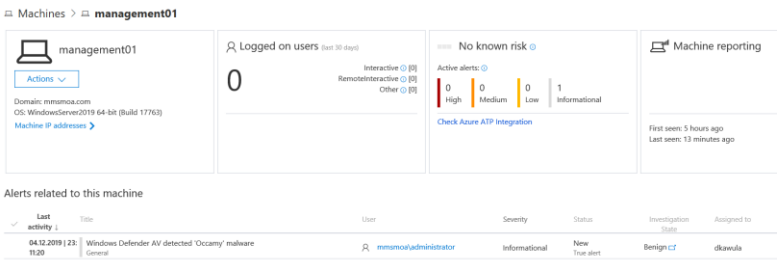
6. Download the Deployment Package to the Target Server Management 01
7. Open an Administrative Command Prompt and run WindowsDefenderATPLocalOnboardingSCript.cmd



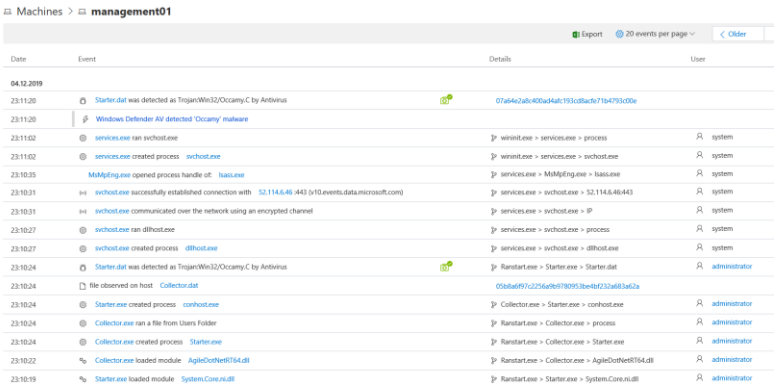
8. Wait approximately 5 minutes and check the machines List in the Portal



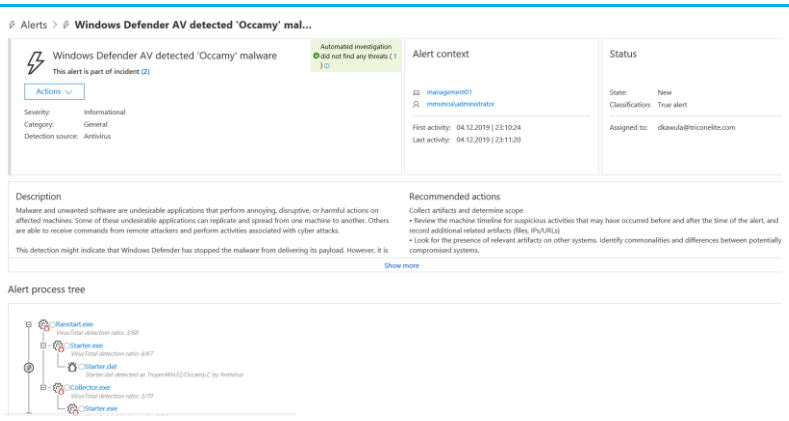
Reviewing an Incident with Windows Defender Advanced Threat Protection

Instructions	Screenshot (if applicable)
1. Browse to https://securitycenter.windows.com/dashboard 2. Log in with your Admin Credentials	
3. Here we can see that our machine Management01 has had Occamy Malware detected. We will look at this attack later in the book.	

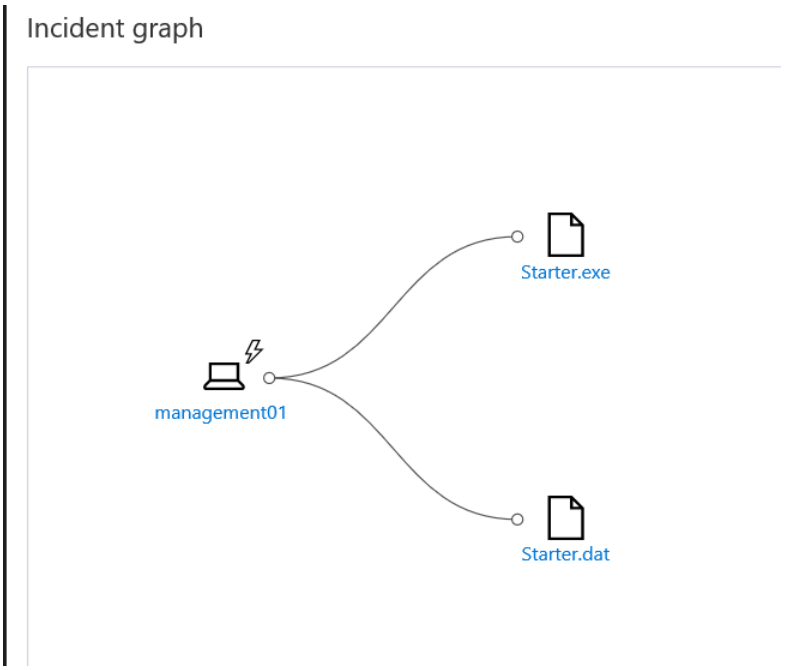
4. If we scroll down on the machine, we can see a timeline of the infection



5. We can also drill into the alert giving more information about the incident



6. We can also see an incident Graph



7. We can also drill into the live investigation that took place for this incident

Appendix C

Join us at MVPDays and meet great MVP's like this in person

If you liked their books, you would love to hear them in person.

Live Presentations

Emile frequently speaks at Microsoft conferences around North America, such as TechEd, VeeamOn, TechDays, and MVPDays Community Roadshow.

Peter frequently speaks at Microsoft conferences in the US and Europe, such as TechEd, Ignite, Midwest Management Summit, Workplace Ninja Summit, Nordic Infrastructure Conference and Techmentor.

You can find additional information on the following blogs:

www.checkyourlogs.net

www.configmgrblog.com

Video Training

For video-based training, see the following site:

www.mvpdays.com

Live Instructor-led Classes

Peter is a Microsoft Certified Trainer (MCT) and presents scheduled custom instructor-led classes in Europe. For current dates and locations, see the following site:

- www.daalmansconsulting.com

Emile regularly performs training sessions for enterprise customers in North America, focusing on Modern Workplace and technology adoption. For current dates and locations, see the following site:

- www.mvpdays.com

Consulting Services

Peter and Emile have worked with some of the largest companies in the world and had a wealth of experience and expertise. Customer engagements are typically between two weeks and six months.