Top 10 Best Practices for Windows 10 OSD

Table of Contents

Preface

In this report, you'll learn the Top 10 best practices on the use of the operating system deployment (OSD) capabilities of Microsoft System Center Configuration Manager (ConfigMgr) for Windows 10. The report goes into technical depth, and provides a brief explanation of relevant technical concepts.

Although the term "*Top 10*" is used, the numbering of an item does not mean it is any more important than any of the other items.

Also, as with pretty much anything to do with ConfigMgr, the way you implement and use the OSD feature is dependent on your specific environment. Therefore some of the best practices mentioned in this document may not be suitable for use in your particular case.

What is ConfigMgr OSD?

ConfigMgr allows administrative users to create and deploy an operating system image to computers both managed and unmanaged by ConfigMgr through the use of bootable media (such as a CD set, DVD, or USB flash drives), or by booting from the network.

The actual operating system image—created as a Windows Imaging Format (WIM) file—that is deployed using ConfigMgr can contain the relevant version of the Windows operating system including your customizations such as optional applications, files and software updates.

Using OSD you can:

- Capture a reference image of a computer that will be used to deploy the operating system to a new computer.
- Capture and restore user state by using the User State Migration Tool (USMT).
- Deploy the operating system image to a collection of computers.
- Upgrade an operating system to Windows 10.
- Create Task Sequences containing multiple steps that perform automatic actions without any user intervention.
- Create a Task Sequence that can perform tasks that require user intervention for example prompting for the computer name to use, presenting a list of applications to be installed, amongst others.

However even when using the OSD capabilities of ConfigMgr, deploying a new operating system, like Windows 10, can be a complicated and slow process. For an enterprise, these deployments often requires countless hours of preparation and planning as well as manual configuration changes and updates. The following best practices offer strategies to speed and automate operating system deployments to reduce the burden on IT professionals that manage these critical projects.

1. Define and Document Your Requirements and Process

This may sound obvious, and to some it may sound tedious. But if you are going to do Operating System Deployment (OSD) right, you first need to define your requirements such as which of the supported OSD scenarios you are going to support:

- **Bare metal –** Deploying a new version of Windows to a computer that currently has nothing installed on it.
- **Refresh –** Wiping an existing computer and deploying a new version of Windows to it migrating any user settings and data that need to be kept.
- **Replace –** Replacing an existing computer with a new one by installing a new operating system on the new computer and then optionally transferring over any user settings and data that need to be kept from the old computer.
- **Upgrade –** Upgrading the existing operating system from Window 7, 8.x to 10 whilst retaining all of the existing applications, settings and user data that is already on the computer.

Once you know which scenarios you are going to support, you then need to look at which computer models in your organization are in scope for OSD. Obviously the fewer models you have the less you need to worry about in terms of sourcing and maintaining drivers, plus your test matrix and the actual time spent testing is significantly reduced.

The type of image you are going to create (Thin or Thick which we look at in the "Decide on Your Image Strategy" section), is also a key consideration that needs to be decided before you actually start work.

You also need to consider any service level agreements (SLAs) you currently have in place. For example, if a user can only be without their machine for a set period of time or during certain hours you'll need to determine not just how you are you going to do OSD but also

when, factoring in other activities that may also be going on at the same time that you plan to do OSD.

Once you've got your requirements defined you should also do a review of your current ConfigMgr design as you may need extra Site System roles such as Distribution Points and State Migration Points in order to support OSD in locations that currently don't have them.

2. Choose Your OSD Method

One of the first things you need to decide is which OSD method to use:

- Microsoft Deployment Toolkit (MDT) Stand-Alone
- MDT and Configuration Manager
- Configuration Manager Stand-Alone
- MDT integrated with Configuration Manager

Each of these have their own advantages and disadvantages, which we'll examine below.

Microsoft Deployment Toolkit Stand-Alone

As its name suggests, the MDT is a free toolkit that helps you automate the deployment of a Windows Operating System (OS) to computers. It does this through the configuration of the unattended Setup files for Windows plus the packaging of any other required content and files. The end result is an image file you can deploy to reference computers and those you want to deploy Windows to.

MDT stand-alone is especially useful in very small environments or those that are disconnected.

If you are going to use MDT stand-alone Microsoft recommends creating "*Thin*" images i.e. those that contain the OS plus any necessary updates.

When you use MDT for your OS deployments it uses the Lite Touch Installation (LTI) method. In other words it's not 100% automated and there is some interaction required on each computer in order to complete the deployment.

However, because of its simplicity, MDT stand-alone only requires a small infrastructure unlike the other deployment methods. In the majority of cases it is the fastest for creating a reference image.

If you're working in a large company that also uses other Microsoft deployment technologies—such as Windows Deployment Services (WDS), ConfigMgr, System Center Virtual Machine Manager (SCVMM)—these can all re-use the images created by MDT.

A big advantage of using MDT stand-alone is that it runs in the context of the Local Administrator (compared to ConfigMgr which performs deployments in the context of the LocalSystem account). This means you can configure any settings you'd like to, and then use the CopyProfile functionality during the deployment to copy these changes to the default user. An alternative option is to understand any changes that need to be made and then import them to the default profile during the OSD. If you are using Microsoft User Experience Virtualization (UE-V) or AppSense, you could create a default profile view that will be imported once the user logs on.

MDT and Configuration Manager

This OSD method is recommended by Microsoft for use in large enterprises where there are thousands of computers and multiple applications that need to be managed.

This method consists of using MDT Stand-Alone to build and test your OS images and then deploying the finished image to a pilot group of production users which you can do by using WDS to perform LTI deployments via PXE.

Once you've completed a successful pilot deployment you can then perform large scale deployments of your MDT generated OS images. You can use features of ConfigMgr to minimize the impact of the deployment on your LAN/ WAN such as: bandwidth controlled distribution of your images out to your Distribution Points, Multicast installation to reduce LAN utilization, and reporting to help you keep track of deployment status.

Configuration Manager Stand-Alone

ConfigMgr on its own can be used to create OS images in WIM format. Typically the images created by this method are known as "*Thick*" images. Thick images contain the Windows OS plus any required applications. (Note that Thin images can also be created.)

Once the image has been created, ConfigMgr can be used to deploy it utilizing the features as discussed in the "MDT and Configuration Manager" section.

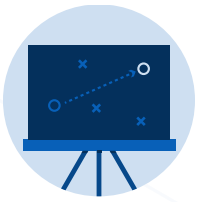
MDT Integrated with Configuration Manager

When MDT is integrated with ConfigMgr, Task Sequences can leverage MDT's rules to enhance them and provide greater flexibility with the deployment. For example in some cases you may need to create a User-Driven Installation (UDI) to gather some information from the user on the targeted computer such as the computer name to use, which

Organizational Unit (OU) the computer should reside in, which applications should be installed by the Task Sequence, etc. When MDT is integrated with ConfigMgr, you cannot only enable the UDI wizard to collect this information you can also customize the UDI wizard using the UDI Wizard Designer.

Using this OSD method you can also create fully automated deployments known as Zero Touch Installations (ZTI), where there is no user intervention on the targeted computer.

At the end of the day only you can decide which OSD method best fits your requirements.



3. Decide on Your Image Strategy

There are traditionally two schools of thought when it comes to creating your actual operating system image:

- **Thin:** Typically contains just the Windows operating system and any company approved updates.
- **Thick:** Contains both the Windows operating system (and approved updates), and other customizations such as applications, files, etc.

A key consideration in deciding which option to take is your business requirements. For example whether you have any Service Level Agreements (SLAs) in place that dictate how long it should take to provision a machine. Also do you have a core set of applications that need to be delivered to every machine or do different departments have different requirements?

Adopting a Thin image may mean the machine is deployed faster and ready for the user to log on once the OS has been deployed, but the user won't be able to do anything productive until the key applications they require have been installed and then patched.

On the other hand, if you decide to use a Thick image, it may take longer for the initial deployment to complete (downloading a larger image, installing the OS, installing apps, patches, etc.), but at least once the machine has been deployed the user can log in and start working.

You may decide on a hybrid image i.e. something in between a Thick and Thin image which includes the OS and a core set of applications that don't change frequently or need regular updates. Any optional or non-key business applications are installed post deployment.

In previous versions of ConfigMgr it was possible to do an in-place upgrade of the OS on a machine rather than wiping everything and starting again with a new OS, but this required you to develop and test the process. Now in Configuration Manager Current Branch, Microsoft provides this capability out of the box. The idea here is that once you have all of your applications and user settings and data in place, it should be easy to upgrade the operating system in the same way as you'd change the tires on a car.

If you don't have SLAs or business drivers that dictate you adopt one strategy over another, the other thing to consider is how often are you going to update your reference image and the effort involved?

For example Microsoft releases security updates every month that could potentially need to be applied to your reference image. However, the more frequently you update your image the more work and testing it involves. On the other hand if you wait say six months to patch your reference image every machine built from the old image will need to have all of the missing updates applied via ConfigMgr once it's been built. This process is far slower and potentially impacts the user with reboots versus keeping the reference image up-to-date so any updates/patches are installed during the initial deployment of the reference image.

Also the more you have in an image then potentially the more maintenance it is going to need versus say a Thin image where you only need to worry about OS updates, as application updates can be handled outside of the imaging process as part of the normal application lifecycle.

A lot of people decide that updating their reference image on a quarterly basis is a good trade-off between the amount of work required in updating and testing the new image versus the impact of machines having to download any updates that are not included in the reference image.

Finally, consider your network infrastructure. It's obviously a lot less stressful to deploy a smaller Thin image across a WAN than it is to deploy a monster Thick image that contains everything.

Whatever you decide, you should create your base image on a virtual machine rather than a physical one to keep the image as driver independent as possible. Make sure you keep two versions of your build (so you have something to go back to if future modifications don't work), and that you version each version of your build so you can keep a version history.

4. Establish a Driver Management Strategy

Every machine that you deploy with Operating System Deployment (OSD) will potentially require a number of drivers for things like:

- Network card
- Hard disk
- Display
- Audio
- Machine specific functionality such as hotkeys

This is potentially a lot of drivers and there are several things you need to think about to make this as easy as possible.

To start with, use certified hardware drivers. The drivers you use should be certified by the manufacturer to run on the version of Windows you are deploying as the last thing you want to be doing is deploying faulty drivers.

When it comes to driver management you could create a central driver store and then let Plug and Play on the machine choose what it thinks is the correct the driver for each component. The drawbacks with this approach are:

- Plug and Play doesn't always correctly identify the correct driver for the hardware component.
- Plug and Play will always try to use the latest version of the driver even those that have not been tested and certified.
- Whatever driver Plug and Play chooses to use it will download it from the network which if you're building lots of machines could be a significant amount of traffic.
- Trying to work out exactly what drivers are being used on which machines becomes tricky.
- You could end up with conflicts because different combinations of drivers are used on different types of machines.

A better alternative is to create driver packages for each particular hardware model. For example, create separate driver packages for each kind of driver like hard disk, audio,

display, etc. Of course there is the overhead of initially creating these packages but there are also several benefits:

- You can decide which drivers you want to use rather than importing all drivers which potentially bloats your database and consumes disk space unnecessarily on your Site Server/Distribution Points, plus you're consuming network bandwidth unnecessarily by deploying content that isn't required.
- All of the required drivers can be copied to your media such as a USB key saving network bandwidth and decreasing build time as the driver doesn't need to be downloaded before it can be installed.
- Being modular you can easily test them in isolation or in groups.
- You can configure them to meet your organizational needs.
- Some drivers may apply to more than one model so it's easy to re-use them.

TIP: Check with your vendor whether they have driver packages available that are compatible with ConfigMgr (some do, some don't). If they do, don't just import the **.cab** file; instead, extract and import just the drivers you need.

You may also find that some hardware needs more than just a driver to work such as a display driver which might also need associated tools to be installed as well in order to work.

In such cases it's better to install these drivers as an application (so you can install any required software as well), instead of installing the driver as just a driver.

Try to keep the number of hardware models in your organization to a minimum and also try to ensure that each model has an agreed standard. This way you keep the costs of having to develop, test and maintain driver packages to a minimum.

5. Boot Images

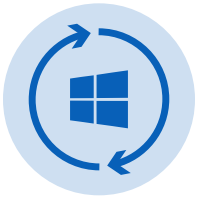
A Boot Image contains a cut down version of Windows known as the Windows Preinstallation Environment (Windows PE) that contains a limited number of components and sufficient device drivers that prepare a destination computer for Windows to be installed.

By default ConfigMgr ships with two default boot images (one for x86 machines and the other for x64 machines). You can modify and customize Boot Images to meet your requirements, but before doing so consider the following:

- Make a copy of the default Boot Images and customize the copies. In this way when you upgrade ConfigMgr and the default Boot Images get overwritten, you don't lose all of
  your customizations.
- Test to see if the default Boot Images work with your hardware as there is no point needlessly adding extra network and disk drivers to your Boot Images. The only drivers you should need to potentially add to your Boot Images are network and hard disk.
- Consider having two of each type of Boot Image (for example two x64 Boot Images and two x86 Boot Images (assuming you need x86 Boot Images)). Have one set of Boot Images for testing on which you enable Command Support. This way, if when you try to build a machine it fails you can press **F8** to start a Command Prompt to perform troubleshooting such as looking at the **smsts.log**. Keep the other set of Boot Images for production on which you don't enable Command Support. See below for details of how to enable Command Support for a Boot Image if you don't know how to do this.
- Try to use Unified Extensible Firmware Interface (UEFI) mode instead of BIOS mode. UEFI is more secure as when the computer starts the firmware checks the signature of the boot software and OS before booting it to prevent the boot partition from getting infected with viruses.

If you need to enable Command Support for a Boot Image:

1. Load the ConfigMgr Console.
2. Click the Software Library workspace.
3. Navigate to Overview\ Operating Systems\ Boot Images
4. Double click the relevant Boot Image you want to enable Command Support for.
5. Click the Customization tab.
6. Check the Enable command support (testing only) checkbox.
7. Click **OK**.
8. Update the Boot Image on any Distribution Points it's already been distributed to.



6. User State Migration

One of the biggest problems facing companies is how to migrate users and all of their data and settings from one machine to another, or upgrade the OS on a user's machine to a new version without any data or settings being lost. Not all companies are using folder redirection or solutions such as UE-V. In an ideal world no company data would be stored locally (and some companies do have this policy), but for some (especially laptops), users having local data cannot always be avoided.

This is where the USMT comes in. USMT is a Microsoft application designed to help you migrate the user profile settings for users (such as data files and both OS and application settings), from one Windows-based machine to another. It is included as part of the Windows Assessment and Deployment Kit (ADK), which is a prerequisite that has be installed before you can install ConfigMgr. The Windows ADK can be used in the following OSD scenarios:

- **Refresh –** Where the OS on a machine is upgraded to a later version.
- **Replace –** Where the user is given a new machine.

When using USMT you have several options for what you want to migrate and how. Once you've decided what you want to migrate, you then need to decide which type of Migration Store you want to create and where you want to store it (locally or remote).

In the case of the Refresh scenario the best approach is to use a Hard-Link migration which basically allows you to keep all of the user state on the computer whilst the old OS and any other files which are no longer required are removed and the new one OS is installed. It works by creating a map of all of the files on the machine so it knows which files belong where on the disk. In this way the actual files themselves don't need to be copied off the machine, plus only the files that are mapped are kept.

If you are not using one of the built-in ConfigMgr Task Sequences then providing there is sufficient disk space you should look to create a local store. In this way the user state is stored locally on the device rather than being copied off to the network. This approach avoids impacting the network and consuming hard disk space on the State Migration Point (SMP), and it is faster as no data is actually copied.

If you are using one of the built-in ConfigMgr Task Sequences you have a choice of performing a Hard-Link migration (so everything is kept local to the machine), or a compressed/uncompressed store that is stored securely on the SMP which can be retained for a period of time in case the restore didn't work or something got missed.

If it's not possible to use an SMP, another option would be to manually configure USMT to copy the data to a peer/local file share and then restore it afterwards to avoid copying the data over the WAN.

If you are replacing a user's computer (i.e., moving them to new hardware), or repartitioning the hard drive, or the computer you are refreshing doesn't have enough local disk space, then regardless of the type of Migration Store, you'll need to store the state data remotely in the case of ConfigMgr on the SMP.

Finally if you are going to use USMT and need to use a SMP to store the data, make sure the SMP is installed locally to the machines that are going to need it and that you perform regular housekeeping on your SMPs to ensure they have sufficient free disk space to store the files and settings from computers as you deploy to them.

7. Using the "All Unknown Computers" Collection

In order to be managed by ConfigMgr a computer needs to have the ConfigMgr Client installed on it and a record needs to exist for it in the ConfigMgr Database. Any computers that ConfigMgr doesn't know about are referred to as *Unknown Computers.* A computer could be unknown for the following reasons:

- ConfigMgr has not discovered the computer.
- The ConfigMgr Client has not been installed on the computer.
- The computer has not been imported to ConfigMgr.

If a computer is unknown, can you deploy an OS to it? The answer is yes.

If you look in the Configuration Manager console in the **Assets and Compliance** workspace, under **Device Collections** you will notice a Collection called **All Unknown Computers**. This Collection contains two generic computer records (**x64 Unknown Computer**, and **x86 Unknown Computer**).

Using either the relevant individual computer object in the **All Unknown Computers** Collection, or by adding the relevant object(s) in this Collection to another Collection, you can deploy an OS to unknown computers. This can be done with either bootable or prestaged media, or a PXE deployment.

TIP: If you only plan to deploy to x86 computers use the **x86 Unknown Computer** object. If you have either just x64 or x86 and x64 computers in your environment you can use just the **x64 Unknown Computer** object. These objects have nothing to do with the version of the OS being installed but relate to the architecture of the destination computer.

This may sound like a great way of getting any new unknown computers deployed and managed by ConfigMgr but there is a potential security risk. When you enable support for unknown computers in an environment an unauthorized computer could be connected to your network, be PXE booted, receive an IP address, and then be built using a corporate OS image. Once built, if the attacker can access the computer, they can work from an unauthorized computer in your domain and on your network.

To prevent this from occurring:

- Make sure access to your physical network is restricted.
- Password protect your Build Media.
- When configuring your Distribution Points (DPs) for PXE select the **Require a password when computers use PXE** check box, and then specify a strong password which will need to be entered whenever a PXE boot initiated.
- Consider establishing a build room which is connected to a network segment which is the only one that allows PXE boots.
- Monitor for any computers that are unauthorized on your network. You could configure your Task Sequence to check for something like the model number, Media Access Control (MAC) address or something that's unique to the specification of your standard machines and if this isn't found the Task Sequence fails with an error. Or you could also implement a MAC whitelist or create a start-up script to perform validation that the computer is authorized to be on your network.

Another potential problem in allowing unknown computers to respond to PXE-initiated OS deployments is that of *accidental* re-imaging because any system local to the PXE server will get a new image if it PXE boots. The end result being the computer receives whatever is deployed to it, potentially resulting in data loss and system outage.

If you are going to enable support for Unknown Computers make sure your machines are only in Collections that are targeted by OSD long enough for them to be built/rebuilt. Once the build is complete the machine should be removed from any OSD-related Collections.

Once again, the decision as to whether you enable deployments to unknown computers depends on your business and security requirements. The most secure option is to not use unknown computer support.

8. Options for Managing Content Deployment

OSD by its very nature involves moving potentially large amounts of content around which in most organizations can be problematic from a LAN/WAN perspective.

Thankfully there are ways you can minimize this impact with ConfigMgr:

- Prestaged Media
- Stand-Aalone Media
- Content Referenced by Task Sequences

Prestaged Media

Prestaged media, as its name suggests, is a way of putting a copy of your boot media and the associated WIM file (containing the OS image, driver packages, applications, etc.) on a Bare Metal machine. A Bare Metal machine is one that is currently not in use.

NOTE: If you're using Auto Apply Drivers rather than Driver Packages, no drivers will be copied to your media.

Some vendors allow you to send them your Prestaged Media. They can place it on the machine so when it arrives at your company it's simply a case of unboxing the machine, connecting it to your network and switching it on. At this point the computer uses the Boot Image to boot into Windows PE, connects to the relevant Management Point to download the Task Sequence and the deployment starts. As pretty much everything that is needed is on the local Prestaged Media there is very little impact to your network so you can deploy more computers simultaneously compared to not prestaging the media and everything then having to come across your network.

If your vendor doesn't provide this service another option is for you to establish your own staging area where new machines are delivered to, the prestaged media is loaded and then it's shipped to the user.

From a security standpoint just be aware that any packages you include on Prestaged Media are not encrypted so you'll need to password protect the media in order to protect these if this is a concern.

Stand-Alone Media

If you have locations that aren't on the same network or in locations with extremely poor or intermittent WAN connectivity, you can create Stand-Alone Media. This will contain everything you need to deploy an OS to a machine so nothing needs to go across your network (assuming you are using Driver Packages rather than Auto Apply Drivers).

In this scenario the machine doesn't need to contact your ConfigMgr infrastructure to download anything. It also does not connect to a Management Point to validate the media.

Just bear in mind that if you need to make any changes to the media, you'll need to recreate it. You can't just update the existing media. This can potentially cause issues if strict version control isn't implemented with build engineers potentially using the wrong version of the media to build machines.

Content Referenced by Task Sequences

At the time you deploy a Task Sequence, on the **Distribution Points** page of the **Deploy Software Wizard** the following two **Deployment options** can have a big impact:

- Download content locally when needed by the running task sequence
- Download all content locally before starting task sequence

As with everything there are compromises. If you choose the **Download content locally when needed by the running task sequence** option, although you don't have the big initial *"hit"* on your network of everything being loaded compared to the **Download all content locally before starting task sequence** option, the overall deployment can potentially take longer if it keeps having to pause to download content. You may also get issues with downloads timing out. Although the sustained impact to the network is less but over a longer period of time compared to a short, potentially intensive burst when using the **Download all content locally before starting task sequence** option, this sustained activity can be problematic in some environments.

The downside to using the **Download all content locally before starting task sequence** option is that it will download literally everything used by the Task Sequence. If the Task Sequence contains dynamic conditions where certain items should only be installed depending on certain conditions, you're potentially downloading content a specific machine doesn't necessarily need. Conversely using this option allows you to run the Task

Sequence offline (the machine doesn't need to be connected to the network), because all of the required content is cached locally on the Client. Make sure your Client cache is big enough to accommodate everything if you decide to use this option.

9. Security

No discussion of OSD would be complete without mentioning Security. Due to its complexity, there is plenty you need to consider for OSD especially in environments where security is of extra importance. Here are some things to consider:

- **Restrict access to the ConfigMgr Console and Media -** Make sure that only those users that should have access to the ConfigMgr Console and Media have access. If someone were to gain unauthorized access to your environment they could potentially deploy Task Sequences to your computers resulting in unintended data loss, system outages as well as discovering sensitive information such as account credentials and volume licensing keys used by ConfigMgr to perform OSD-related tasks.
- **Use built-in ConfigMgr Security Roles -** Make use of built-in ConfigMgr Security Roles such as the **Operating System Deployment Manager** and if necessary Custom Security Roles to ensure only those users involved with OSD have access to this functionality and they can only deploy to certain Collections. The last thing you want is someone who shouldn't have access to OSD causing potential mayhem either maliciously or because they have access to features they don't understand and they end up deploying to a Collection like **All Systems** or others that may have a big impact if targeted incorrectly.
- **Collection Variables -** Be careful with your use of Collection Variables as local Administrators can potentially read sensitive information they may contain.
- **State Migration Points -** Be aware there is no way to limit the amount of data a machine stores on a State Migration Point (SMP). It is therefore possible for an attacker's machine to consume all of the available disk space on the SMP, which would cause a denial of service.
- **Block the Client Certificate if compromised -** If you discover the Client Certificate has been compromised (which is required in order to deploy an OS), then if the certificate is a PKI certificate revoke it. Also, you should block it in the ConfigMgr Console (within the **Administration** workspace navigate to **Security\ Certificates**

right-click on the certificate and select **Block**). Failure to do so could result in an attacker being able to impersonate a valid ConfigMgr Client and therefore have the capability to download Policies which can contain sensitive information.

- **Don't enable Command Support on your production Boot Images –** Enabling Command Support allows you to press **F8** to start a Command Prompt if a machine build fails so that you can perform troubleshooting such as looking at the **smsts.log**. This is obviously a security risk as an attacker potentially has access to your network plus access to variables in the Task Sequence environment which could contain sensitive data.
- **Protect the Client Authentication Certificate during its capture –** If an attacker were to obtain the Client Authentication Certificate it would allow them to impersonate a valid Client on your network as they would have access to the Private Key contained in the certificate.
- **Do not grant the Network Access Account (NAA) excessive rights –** The NAA requires just the **Access this computer from the network** right on any Distribution Points (or other servers), that hold the package content the machine needs to access.
- **Do not re-use the account configured as the NAA –** The NAA should only be used by Client computers when they cannot use their local computer account to access content on DPs.

  Do not configure the same account used for the NAA for the following:

  - Capture Operating System Image Account.
  - Task Sequence Editor Domain Joining Account.
  - Task Sequence Run As Account

Instead, configure a unique account for each of the above.

10. Troubleshooting

Things will invariably go wrong when you're doing OSD. The first thing to understand is that the majority of OSD-related activity gets logged to the **smsts.log**. What can become confusing is where to find this log as its location changes depending on the phase of the Task Sequence, as shown in the table below:

Phase	Log Location
Windows PE - Before the hard disk is formatted	**x:\windows\temp\smstslog\smsts.log**
Windows PE - After the hard disk has been formatted	**x:\smstslog\smsts.log** and copied to **c:_SMSTaskSequence\Logs\Smstslog\smsts.log**
Windows Operating System - Before the ConfigMgr Client is installed	**c:_SMSTaskSequence\Logs\Smstslog\smsts.log**
Windows Operating System - After the ConfigMgr Client has been installed	**c:\windows\ccm\logs\Smstslog\smsts.log**
Windows Operating System - Once the Task Sequence has completed	**c:\Windows\CCM\Logs\smsts.log**

Other logs you might find useful are those located in the **C:\Windows\Panther** directory which is used during the installation of Windows. These and many others are detailed in Microsoft Knowledge Base Article 927521 (see the Appendix for a link). In addition, ***%SystemRoot%*\debug\Netsetup.log** details the domain join process including any errors or problems.

Of course you should also be familiar with the ConfigMgr logs both on the Site Server/Site Systems and on the Client which can be of great use in troubleshooting when things go wrong.

Whilst on the subject of logging, it's a good idea to make a copy of the logs in case of a failure which you can achieve by creating a simple **.bat** file. All it needs to do is map a drive to a network share, create a directory for the computer, and then use xcopy to copy the logs.

Exploring Peer-to-Peer Alternatives

Even with following these best practices that utilize ConfigMgr's powerful OSD capabilities, migrating thousands or hundreds of thousands of systems can present serious challenges:

- **PXE Enablement:** For many Windows 10 deployment scenarios, you need one or more PXE servers at every facility, and numerous time consuming network configuration changes (setting up IP helpers, DHCP Scope options, etc.).

- **Content Delivery:** With Windows 10 images and corollary files easily exceeding 20GB, you could you could flood petabytes of data onto your WAN by delivering and updating OSD files without intelligent automation.
- **State Migration:** Saving and restoring user settings and data before and after migration can require massive bandwidth, scores of distributed servers, or both.
- **Serverless Pre-staging:** Caching OSD content at offices around the globe using remote Distribution Point servers can be an expensive and labor-intensive effort that requires frequent attention, troubleshooting, and updating.

To alleviate these issues it is also recommended that enterprises explore peer-to-peer solutions which can speed and automate OSD. These solutions enable any Windows client to become a PXE server, eliminating the need for additional server purchases, IP helpers, DCHP scope options, router permissions, and configuration changes.

For more information on peer-to-peer PXE, visit: http://www.adaptiva.com/windows-10-deployment-rapid-osd-migration-sccm/

About Adaptiva

Adaptiva is a leading, global provider of IT systems management solutions that advance the power of Microsoft System Center Configuration Manager (ConfigMgr). Founded in 2004 by the lead architect of Microsoft SMS 2003, Adaptiva enables IT professionals to securely speed enterprise-wide software deployments without adding costly servers or throttling network bandwidth. The company's breakthrough peer-to-peer technology securely distributes software across enterprises faster than any other systems management solution available today. Adaptiva's suite of smart scaling systems management products includes OneSite™ for rapid content distribution and management, Client Health™ for endpoint security, troubleshooting, and remediation, and Green Planet™ for energy-efficient power management and patching. The company's software is used by Fortune 500 companies and deployed on millions of devices in over 100 countries. Learn more at www.adaptiva.com.

Acknowledgements

Cliff Hobbs and FAQSHOP.COM

This report was written by Cliff Hobbs. Cliff is a 13 times Microsoft Most Valuable Professional (MVP), the first to be awarded in the UK for Microsoft System Center Configuration Manager (ConfigMgr / SCCM) and Systems Management Server (SMS).

He has worked as a Consultant with the product since 1998, during which time he has gained extensive experience of designing, deploying, supporting, and documenting enterprise-wide Configuration Manager implementations on behalf of many companies such as Microsoft, HP, EDS, Getronics, Abbey (now Santander), across multiple industry sectors.

Since 1998 Cliff has been writing Frequently Asked Questions (FAQs) related to ConfigMgr and its related products. In 2003 he founded http://faqshop.com which is one of the most popular websites for ConfigMgr-related information.

Raphael Perez

Special thanks from Cliff to fellow Microsoft MVP in Enterprise Mobility Raphael Perez for his help in reviewing this report. You can find out more about Raphael, the software he has developed, and the services he offers at http://www.thedesktopteam.com/raphael.

Appendix

This Appendix contains useful links to help you learn more about Operating System Deployment and Configuration Manager.

Microsoft Deployment Toolkit

Microsoft Deployment Toolkit (MDT) 2013 Update 1:

https://www.microsoft.com/en-us/download/details.aspx?id=48595.

System Center Configuration Manager (Current Branch)

Documentation Library:

https://technet.microsoft.com/en-us/library/mt346023.aspx

Manage enterprise operating systems with System Center Configuration Manager:

https://technet.microsoft.com/en-us/library/mt627933.aspx

System Center 2012 Configuration Manager

Documentation Library for System Center 2012 Configuration Manager:

https://technet.microsoft.com/en-us/library/gg682041.aspx

Operating System Deployment in Configuration Manager:

https://technet.microsoft.com/en-us/library/gg682018.aspx

Technical Reference for Log Files in Configuration Manager:

https://technet.microsoft.com/en-us/library/hh427342.aspx

Windows

Join and Authentication Issues:

https://technet.microsoft.com/en-us/library/cc961817.aspx

Microsoft Knowledge Base Articles:

Windows 7, Windows Server 2008 R2, and Windows Vista setup log file locations: https://support.microsoft.com/en-us/kb/927521