Building A Windows 10 1809 (October 2018 Update) Reference Image with MDT

This post will walk through installing and configuring Microsoft Deployment Toolkit to build a reference image of Windows 10 1809 (October 2018 Update) using a Hyper-V Virtual Machine. It is assumed that you have a Server or PC ready to install MDT onto and create an file share for MDT to build the image with.

Here are the links to the software we'll be using:

- **Windows 10 1809 Assessment and Deployment Kit (ADK)**
- **Windows PE add-on for the ADK**
- **Microsoft Deployment Toolkit (8456)**
- Windows 10 1809 x64 Volume Licensing Service Center | **MSDN Subscriptions site**

Additional software which may be useful to you:

- **Windows USB/DVD Download Tool** – For creating a bootable USB from the Windows 10 ISO to build a PC for admin purposes.
- Remote Server Administration Tools for Windows 10 – Be sure to download the version of RSAT for the version of Server you want to administer.

Installing Microsoft Deployment Toolkit and Dependencies

1. First we'll install the Windows 10 1809 ADK. During setup additional files will need to be downloaded, so it may take some time depending on your internet connection.
2. On the **Select the features you want to install** screen select:

- Deployment Tools
- Imaging And Configuration Designer (ICD)
- Configuration Designer
- User State Migration Tool (USMT)

1. Starting with the Windows 10 1809 ADK, WinPE is a separate install. Install the WinPE add-on by running the adkwinpesetup.exe, there is no specific configuration during the install wizard.
2. Now install MDT by running the setup file downloaded earlier. There is no specific configuration during the install wizard.

Creating the Deployment Share

1. Open the **Deployment Workbench** from the Start Menu.
2. Right click on **Deployment Shares.**
3. Select **New Deployment Share**.
4. Enter the **path** for the Deployment Share: **E:\Build**.
5. Enter the **Share name**: **Build$**.
6. Give the share a **description**.
7. On the **Options** screen, accept the defaults as you can change them later.
8. Complete the wizard to create the share.
9. By default, the share permissions are set the local administrators group. We'll revisit this later.

MDT

Adding an Operating System

1. Mount the Windows 10 1809 ISO in File Explorer.
2. Go to **Deployment Workbench** > **Operating Systems.**
3. Right click and select **New Folder**.
4. Enter the name **Windows 10 1809 x64** and click through the wizard to create the folder.
5. Right click again and select **Import Operating System**.
6. In the wizard, select **Full set of source files** and then enter the root of the mounted ISO as the **Source directory**.
7. For the destination directory name enter **Windows 10 1809 x64** and complete the wizard.
8. Go to the **Operating Systems/Windows 10 1809 x64** node and rename the new entries added to **Windows 10 1809<Edition>x64** for ease of use.

Creating Package Folder For Future Updates

1. Go to **Deployment Workbench** > **Packages.**
2. Create a folder named **Windows 10 1809 x64**.

Now we'll create a selection profile so that the Task Sequence only attempts to install the updates for Windows 10 1809 x64.

Creating A Selection Profile

1. Expand the **Advanced Configuration** node.
2. Right click on **Selection Profiles** and select **New Selection Profile**.
3. Name it **Windows 10 1809 x64.**
4. On the **Folders** page, tick the **Windows 10 1809 x64** folder under **Packages** and complete the wizard.

Importing Applications

If you want to add some applications to be a part of your reference image, here I'll cover how to add Microsoft Office. MDT recognises Microsoft Office and provides automated/silent install options.

1. Go to **Deployment Workbench** > **Deployment Share** > **Applications.**
2. Right click on **Applications** and select **New Application.**
3. In the New Application Wizard, choose **Application with source files.**
4. Give the application the name: **Microsoft Office**.
5. Enter the **Source** directory of the installation files.
6. Enter the **Destination** directory: **Microsoft Office**.
7. For the **Command line** enter anything, we'll revisit this later.
8. On the summary page, click **Next** and after the files are copied click **Finish** to complete the wizard.

Configuring Applications

1. Right click on **Microsoft Office**, go to the **Office Products Tab.**
2. Choose the desired **Office Product to Install** from the drop down menu.
3. Check the desired **Office language**.
4. Enter a product key, unless you will be activating Office via KMS in which case leave the **Product Key** option unchecked.

5. Check the **Customer name** option and enter the desired information.
6. Check the **Display level** option and select **None** in the drop down menu.
7. Check the **Accept EULA** option.
8. Check the **Always suppress reboot** option.
9. Click **Apply**.
10. Go to the **Details** tab and the **Quiet install command** should now read:

```
1
                                        setup.exe /config proplus.ww\config.xml
```

Microsoft Office is now set up to be installed silently by a Task Sequence. If you wish to customise the installation to a greater degree, the **Office Customization Tool** can be launched from the **Office Products tab**. This process can also be done for Microsoft Visio and Project.

To add other popular third party software, you'll need to repeat the steps above, with the relevant **Command line** to quietly or silently install them.

Google Chrome – Enterprise Installer

```
1
                                        msiexec
                                        /I googlechromestandaloneenterprise64.msi /qn
```

Adobe Reader – Enterprise Installer

```
1
                                        AdobeReaderDC.exe /sAll
```

We now need to create a new Task Sequence to create a reference image.

Creating a Task Sequence

1. In **Deployment Workbench**, go to **Task Sequences.**
2. Right click and select **New Task Sequence**.
3. For the ID enter: W10-1809**.**
4. Name it **Build Windows 10 1809.**
5. Select **Standard Client Task Sequence.**
6. Select the Operating System **Windows 10 1809 x64.**
7. Select **Do not specify a product key at this time.**
8. Enter an **Organization** name.
9. Select **Do not specify an Administrator password at this time.**
10. Complete the wizard.

Now we'll configure the Task Sequence.

Configuring the Task Sequence

1. Right click on the **Task Sequence** just created and select **Properties**.
2. Go to the **Task Sequence** tab on the **Properties** window of the Task Sequence.
3. Expand the **Preinstall** folder, and select the **Apply Patches** item.
4. Change the **Selection Profile** to **Windows 10 1809 x64**.
5. Go to the **State Restore** folder and select **Windows Update (Pre-Application Installation)**.
6. On the right side of the **Properties** window, go to the **Options** tab.
7. Uncheck the **Disable this step** tick box and do the same with **Windows Update (Post-Application Installation)**.
8. If you skipped the **Importing Applications** section, please disable the Install Applications item and go to step 16, if not please continue.
9. Go to the **Install Applications** item.
10. In the right side of the **Properties** box, select the **Install a single application** option and click the **Browse…** button.
11. Select **Microsoft Office** and change the name Install Applications to **Microsoft Office**.
12. Install other Applications, copy and paste the **Install Applications** item and repeat steps 13 – 15 for the applications of your choice.
13. Click **Apply** and close the Task Sequence.

Blocking Internet Access to prevent Microsoft Store App Updates

To block internet access to the VM whilst the image is building, we'll use the script from Peter Löfgren's System Center Ramblings post.

1. First create a **PowerShell** script file called **Internet-Access.ps1** with the following code:

```
## Creates the disable option used by the script
param (
  [Parameter(Mandatory=$False,Position=0)]
  [Switch]$Disable
)

## If the Disable command line option is not added, the script adds a Firewall Rule to block traffic on ports 80 (http) and 443 (https).
If (!$Disable)
{
  Write-Output "Adding internet block"
  New-NetFirewallRule -DisplayName "Block Outgoing 80, 443" -Enabled True -Direction Outbound -Profile Any -Action Block -Protocol TCP -RemotePort 80,443
```

```
15                                  }
16
17                                  ## If the Disable command line option is added,
                                    the script removes the Firewall Rule created
18                                  above.
19                                  If ($Disable)
                                    {
                                      Write-Output "Removing internet block"
                                      Get-NetFirewallRule -DisplayName "Block
                                    Outgoing 80, 443" | Remove-NetFirewallRule
                                    }
```

1. Save the script in your MDT share, where the Task Sequence will be able to access it. I save my custom scripts in a folder called **_scripts** the **Applications** folder.
2. In the **Task Sequence** created above, we'll add the items required to run the PowerShell script to enable and disable the internet blocking firewall rules.

- Go to the **Task Sequence** tab on the **Properties** window of the Task Sequence.
- Go to **State Restore** and click on the **Add** button.
- Go to **General > Run PowerShell Script**.
- Name the new item **PS Script – Disable Internet Access**.
- Enter **Z:\Applications_scripts\Internet-Access.ps1** or your own path to the PowerShell script we just created.
- Scroll down the Task Sequence to just above the **Imaging** folder.
- Once again, add a new **Run PowerShell Script** item.
- Name it **PS Script – Enable Internet Access**.
- Again, enter **Z:\Applications_scripts\Internet-Access.ps1** or or your own path to the PowerShell script.
- **Important:** Add **-Disable** to the Parameters section.
- Click **Apply** and OK to close the Task Sequence.

What will happen now is that after Windows boots up, a firewall rule will be added to block internet traffic on ports 80 and 443, and just before starting the SysPrep and capture process the firewall rule will be removed.

Next, we'll create a domain user account for MDT.

Creating a service account for MDT in Active Directory

1. Go to **Active Directory Users and Computers.**
2. Create a user called **mdt_admin** and give it a complex password.
3. Go to the Server or PC where the **Deployment Share** is hosted.
4. Give the user mdt_admin **Full Control** share permissions and **Full Control** permissions to all the files and folders in the **Deployment Share**.

Next we need to configure the **Bootstrap.ini** and the **CustomSettings.ini** files to control certain aspects of the deployment environment. The settings below enable auto log in and skip the welcome screen, so these should only be used for lab or closed development environments.

Configuring Bootstrap.ini

1. In **Deployment Workbench**, right click the **Deployment Share** and select **Properties**.
2. Select the **Rules** tab and click the **Edit Bootstrap.ini** button.
3. Add the settings below to the **Bootstrap.ini**.
4. **Close** and **Save** the Bootstrap.ini

```
[Settings]
Priority=Default


[Default]
DeployRoot=\\SERVER-NAME\Build$
UserDomain=contoso.com
UserID=mdt_admin
UserPassword=p@ssw0rd
SkipBDDWelcome=YES
```

Configuring CustomSettings.ini

On the **Rules** tab of the **Deployment Share** properties window, add the settings below.

```
[Settings]
Priority=Default
Properties=MyCustomProperty

[Default]
OSInstall=Y
SkipCapture=YES
SkipAdminPassword=YES
SkipProductKey=YES
SkipComputerBackup=YES
SkipBitLocker=YES
SkipLocaleSelection=YES
SkipTimeZone=YES
SkipDomainMembership=YES
SkipSummary=YES
SkipFinalSummary=YES
SkipComputerName=YES
SkipUserData=YES

_SMSTSORGNAME=Build Share
_SMSTSPackageName=%TaskSequenceName%
DoCapture=YES
ComputerBackupLocation=\\SERVER-NAME\Build$\Captures

BackupFile=%TaskSequenceID%_#year(date) & "-" & month(date) & "-" & day(date) & "-" & hour(time) & "-" & minute(time)#.wim

WSUSServer=http://WSUS-SERVER-NAME:8530

FinishAction=SHUTDOWN
SLShare=\\SERVER-NAME\Build$\Logs
EventService=http://SERVER-NAME:9800
```

We now need to create the boot media to boot the VM into the deployment environment.

Creating The Boot Media

1. In **Deployment Workbench**, right click on the **Deployment Share.**
2. Select **Update Deployment Share**.
3. Select **Completely regenerate the boot images.**
4. Complete the wizard. It will take some time to create the boot images.

Testing and Capturing a Reference Image

To test everything we need to copy the ISO file that we just generated. It is located in the **Boot** folder in the **Deployment Share**. Go to the Server or PC that is hosting the deployment share and navigate to the boot folder. Inside there should be a file named **LiteTouchPE_x64.iso**. Copy this file to a location where a Hyper-V Virtual Machine will be able to access it.

Create a new VM in Hyper-V with the following configuration:

- 2x vCPUs
- 4GB of RAM
- Network Adapter with access the local network.
- Virtual Hard Drive of at least 40GB, preferably on an SSD.
- Boot from CD using the **LiteTouchPE_x64.iso** from MDT.
- If using Hyper-V on Windows 10 1709 and above, make sure **Use Automatic Checkpoints** is disabled.

Start the VM and it will boot from the **LiteTouchPE_x64.iso** into the deployment environment. You will be presented with a screen with the name of the **Task Sequence** you created earlier. **Select** your **Task Sequence** and click **Next** and the task sequence will begin.

The Task Sequence will install Windows 10 1809, update from the WSUS server, install the optional applications if you added them, and then run Windows Update from the WSUS server again. It will then run SysPrep and the reboot back into the deployment environment and MDT will capture the image.

When this process completes the VM will be shutdown and a file named **W10-1809_YEAR_MONTH_DAY_HOUR_MINUTE.wim** will be in the **Captures** folder in the **Deployment Share**.

You now have a reference image for Windows 10 1809 and a Microsoft Deployment Toolkit installation, with a deployment share specifically configured for building reference images.

We'll cover setting up a deployment share and focus on tasks to support deploying Windows to real hardware **in this article**.

I take great care to test my ideas and make sure my articles are accurate before posting, however mistakes do slip through sometimes. If you'd like to get in touch with me please use the comments, **Twitter** (you can tweet me and my DMs are open) or my **contact form**.

I hope this article helps you out, please consider **supporting my work here**. Thank you.

MDT

Unieke FAQ ID: #1055
Auteur: diode
Laatst bijgewerkt:22-04-2020 14:45