# ketterhead peelbackIntroduction

This document is intended to provide information on configuring an MDT Task Sequence for use with Workspace ONE (WS1) Factory Provisioning. The same Factory Provisioning (FP) Package can be utilised with SCCM OSD, as well as OEM and USB key deployments making it a flexible and time saving method.

An added benefit is that the applications included in the FP Package are also exactly the same as those in the WS1 console, requiring less administration.

Deploying a WS1 FP Package as part of an MDT (or SCCM OSD) build process can be utilised to support the following build scenarios that IT will need to support through the lifecycle of a Windows Desktop:

1. New HW builds
2. Rebuilds
3. Reprovisioning
4. Wipe & Load (Upgrades)

## Prerequisites

This solution requires the following:

1. Workspace ONE UEM enabled for Factory Provisioning (should be by default)
2. Windows staging account enabled (should be by default)
3. Applications that are to be deployed, ingested into the WS1 console
4. A WS1 Factory Provisioning Package created with Directory, OOBE experience, Domain Join, Enrolment configuration and Applications selected
5. Install WDS feature on a Windows Server 2016 or newer
6. Install the Assessment and Deployment Kit – Windows 10 on the above server
7. Install the Assessment and Deployment Kit Windows Preinstallation Environment Add-ons – Windows 10 on the above server
8. Install the Microsoft Deployment Toolkit on the above server
9. Configure DHCP options to support PXE Boot to the above server for the build network

# Setup MDT Server

1. On your Windows Server 2016 or newer server, install the Windows Deployment Services (WDS) feature via powershell or GUI and :  
   Install-WindowsFeature wds-deployment -includemanagementtools

A screenshot of a social media post

Description automatically generated

1. Install Assessment and Deployment Kit – Windows 10
2. Install Assessment and Deployment Kit Windows Preinstallation Environment Add-ons – Windows 10
3. Install Microsoft Deployment Toolkit

You can find links and instructions to download, install and configure the above ADK components from the link below:

<https://docs.microsoft.com/en-us/windows-hardware/get-started/adk-install>

**Note:** We only need to accept the defaults and do not import a boot image yet. If you want to run this in a VM with a VMXNET3 NIC, you will need the network drivers imported. To do this:

1. Follow this VMware KB <https://kb.vmware.com/s/article/2032184> to extract the drivers
2. From the extract folder, navigate to VMware\VMware Tools\VMware\Drivers\vmxnet3\Win8 folder
3. Copy contents to MDT Server
4. From MDT Server > Open Windows Deployment Services console
5. Expand Servers > MDT Server > Drivers > right click and Add Driver Packages to this Group…
6. Select the folder copied in step 3. You will select this

## Configure DHCP

We need to configure a referral or DHCP PXE boot configuration for the build network These settings can be set on the Subnet Scope:

Option 66 - Enter the IP Address of your WDS Server

Option 67 - Enter boot\x64\wdsmgf.efi (for 64-bit UEFI)

or

Option 67 - Enter boot\x86\Wdsnbp.com (for x86 and x64 BIOS)

**Note:** Referrals or DHCP PXE boot configuration only supports the referral of x86-based or x64 BIOS-based clients, and does not support the referral of Itanium-based and x64 EFI-based clients.

More information can be found here - <https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/cc732351(v=ws.10)>

I would also ensure the WDS PXE settings found under Windows Deployment Services console > Servers > right click MDT Server > Properties allows all clients to connect and continue automatically past the PXE stage. We will control the deployment within MDT.

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

# Configure MDT Components

On the MDT Server, we need to configure the following:

1. Import an OS to deploy
2. Add an application containing the WS1 FP Package
3. Create Task Sequence
4. Configure Deployment Share Rules & Bootstrap

## Import OS

Mount the ISO for your Windows 10 Operating System

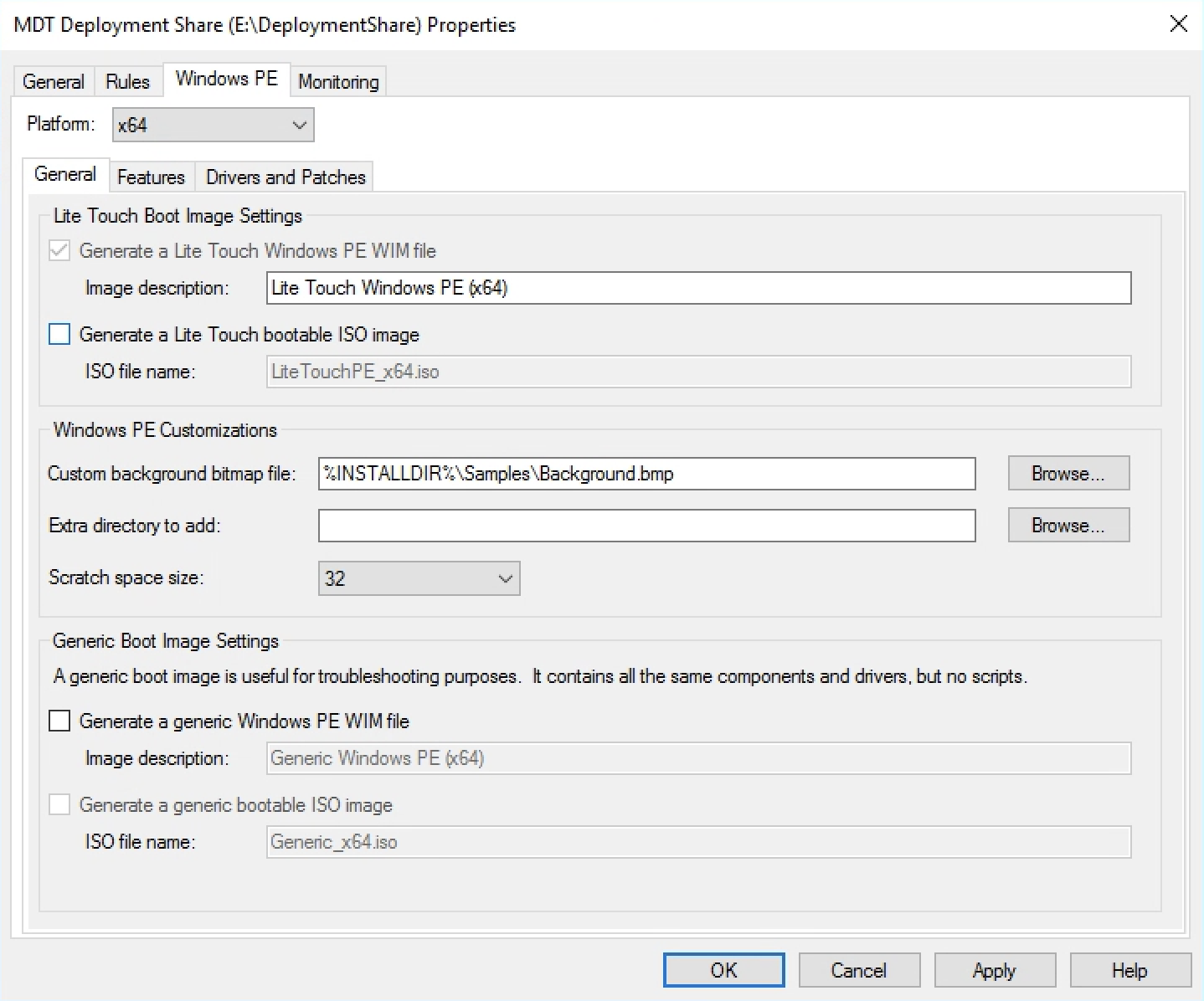
1. From the MDT Server > right click the ISO file > Mount
2. From Deployment Workbench, expand your MDT Deployment Share > right click Operating Systems > New Folder > Name it something like Windows 10 1909 x64  
   I recommend you create a folder for each version of Windows 10 so you don’t get confused later
3. From Deployment Workbench, expand your MDT Deployment Share > Operating Systems > right click Windows 10 1909 x64 > Import Operating System > select Full set of source files > browse to the drive the ISO is mounted > Change the name to Windows 10 1909 x64
4. You can delete the ‘flavours’ of Windows 10 you do not want, but not needed. We will select the ‘flavour’ or index number within the Task Sequence later.
5. Right click MDT Deployment Share > Update Deployment Share > Next…

A screenshot of a social media post

Description automatically generated

Now go back into the Windows Deployment Service console and add a Boot Image from the OS you just imported:

1. Right click Boot Images > Add Boot Image… > browse to MDT Deployment Share\Boot and select the LiteTouchPE\_x64.wim. This is created when updating the Deployment Share



A screenshot of a social media post

Description automatically generated

## Add Application

The application we add to MDT contains the WS1 FP Package that not only installs applications, but will also SYSPREP the device using the UNATTEND.XML provided with the WS1 FP Package. Apart from driver bundles or driver applications, no other Applications or Packages should need to be added to MDT.

We need to get a few components together in a folder to then import into MDT:

1. From the WS1 UEM Console > download the WS1 FP Package (PPKG) and UNATTEND.XML to a folder on the MDT Server. If you haven’t created the WS1 FP Package, follow the instructions in the documentation:  
   <https://docs.vmware.com/en/VMware-Workspace-ONE-UEM/2011/Provisioning_WorkspaceONE/GUID-47134981-1004-4046-8D1E-09E74D85C2B6.html>
2. Download the Workspace ONE Provisioning Tool for Windows and place it in the same location as your WS1 FP Package (PPKG) and UNATTEND.XML above. You can download it from here:  
   <https://resources.workspaceone.com/view/cc56rq4xhtw449g8cyj7/en>
3. Unzip the Provisioning Tool package and copy the contents to the same location as your WS1 FP Package (PPKG) and UNATTEND.XML above
4. Save the RunWS1FP.ps1 script to the same location as your WS1 FP Package (PPKG) and UNATTEND.XML above

You should have a folder with similar contents to this:

A screenshot of a cell phone

Description automatically generated

The RunWS1FP.ps1 script is providing in a section below.

To add the WS1FPPackage Application to MDT:

1. Within the Deployment Workbench, expand your MDT Deployment Share > right click Applications > New Application > Application with source files > Enter relevant Publisher and Application Name details > Browse to the folder above
2. Enter the following command line \*  
   €
3. Leave the Working Directory blank, update this by copying the Source directory after viewing the properties of the Application after saving

\* Note: The “-windowstyle hidden” parameters on the command line are not required, however reduces the number of windows open whilst running the script and subsequent tools.

A screenshot of a social media post

Description automatically generatedA screenshot of a cell phone

Description automatically generated

## Create Task Sequence

Bringing all the components together is the Task Sequence.

1. From the Deployment Workbench > expand your Deployment Share > right click Task Sequence > New Task Sequence
2. Specify a Task Sequence ID and name. I suggest similar to below

A screenshot of a cell phone

Description automatically generated

1. Standard Client Task
2. Expand the Operating System folders > select the version and index required
3. Do not specify a product key at this time
4. Provide Name, Organisation and leave Internet Explorer Homepage as about:blank
5. Provide Administrator Password
6. Next … Save

## Modify the Task Sequence

Now we need to modify the Task Sequence to add our Application and disable some tasks we don’t want.

1. Under State Restore > Custom Tasks > Add > General > Install Application > select Install a single application > Browse and select the WS1FPPackage application

A screenshot of a cell phone

Description automatically generated

1. If you want to change your Operating System or which ‘flavour’/index number to deploy, select Install > Install Operating System > Browse and select the appropriate WIM

A screenshot of a social media post

Description automatically generated

1. Disable SYSPREP running from the Task Sequence. We do this because we want the WS1FPPackage Application with our UNATTEND.XML configuring the device, rather than the settings within MDT. It removes the requirement to maintain an UNATTEND.XML for the MDT build as well as an OEM that may also be utilising WS1 Factory Provisioning.  
     
   Select State Restore > Imaging > Prepare Only > Options tab > Disable this step. Repeat for the Imaging > Sysprep Only and Imaging > Capture Image Folders.

A screenshot of a cell phone

Description automatically generated

## Deployment Share Rules

You can control the behaviour of the WinPE phase of the build by automating the various wizard screens. Some require you to specify the values, others have defaults, but by skipping sections or screens, you can get an almost 100% automated build.

I would recommend you do not skip the Task Sequence selection screen and ensure SkipTaskSequence is configured with NO, so you don’t accidentally rebuild a machine.

A screenshot of a cell phone

Description automatically generated

Here are my settings.

You will note that I do not join the domain, set the computername or regional settings etc. This is because these settings control the initial image, that we customise after with our WS1 Factory Provisioning Package. No point configuring twice!

To configure these settings, right click your MDT Deployment Share > Properties > Rules tab.

### Bootstrap.ini

[Settings]

Priority=Default

[Default]

DeployRoot=\\FULCRUM-MDT\DeploymentShare$

UserDomain=FULCRUM.LOCAL

UserID=Administrator

UserPassword=VMware1!

SkipBDDWelcome=YES

### CustomSettings.ini

[Settings]

Priority=Default

[Default]

\_SMSTSORGNAME=VMWARE

UserDataLocation=NONE

DoCapture=NO

OSInstall=YES

AdminPassword=VMware1!

TimeZoneName=Pacific Standard Time

JoinWorkgroup=WORKGROUP

HideShell=NO

FinishAction=SHUTDOWN

DoNotCreateExtraPartition=NO

ApplyGPOPack=NO

SLShare=\\FULCRUM-MDT\deploymentshare$\Logs

SkipAdminPassword=YES

SkipProductKey=YES

SkipComputerName=YES

SkipDomainMembership=YES

SkipUserData=YES

SkipLocaleSelection=YES

SkipTaskSequence=NO

SkipTimeZone=YES

SkipApplications=YES

SkipBitLocker=YES

SkipSummary=YES

SkipRoles=YES

SkipCapture=YES

SkipFinalSummary=YES

You shouldn’t need to change any other settings except for selecting a driver bundle, such as is required for a VM:

A screenshot of a cell phone

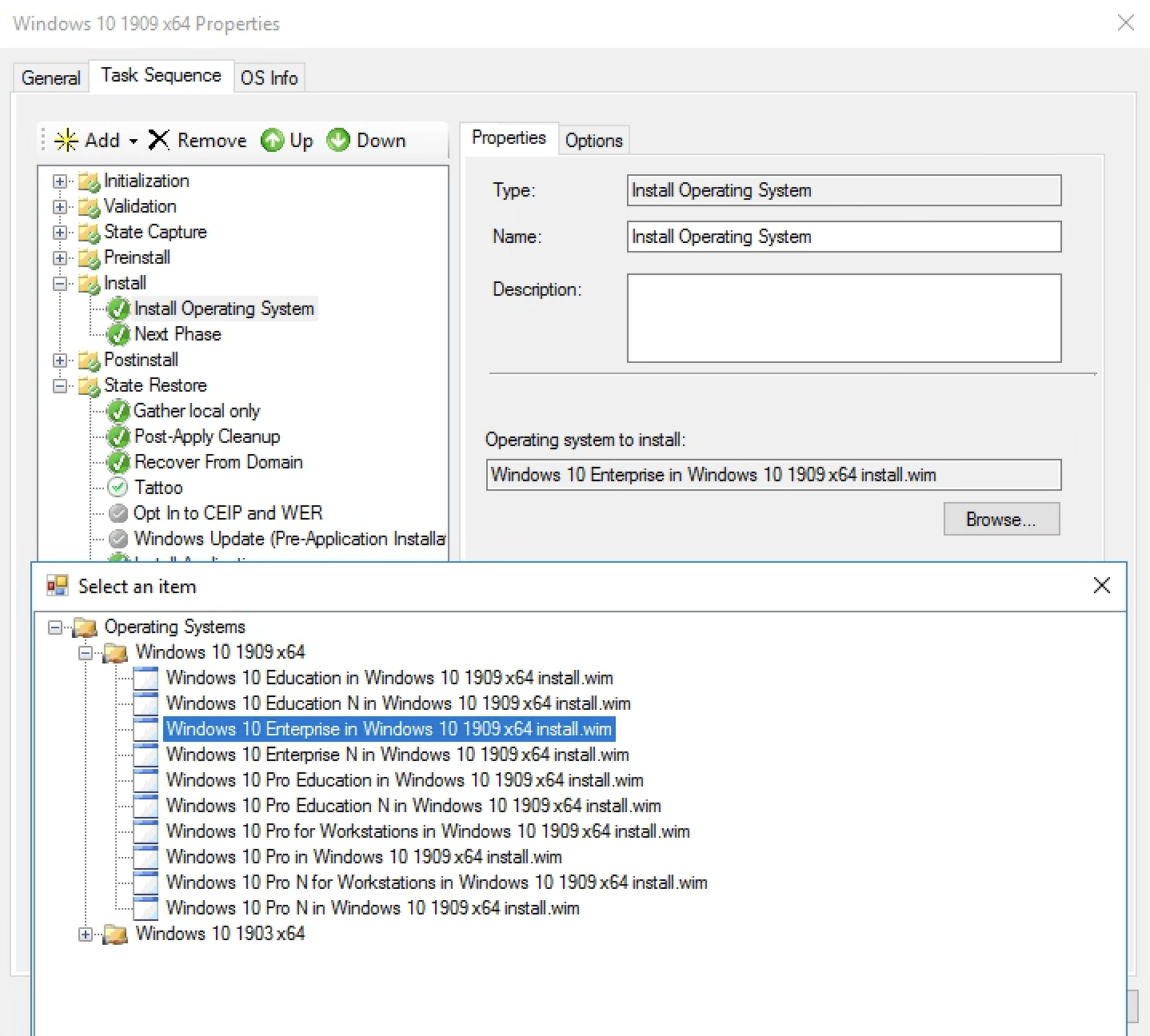
Description automatically generated

# Lifecycle Management

## Windows 10 OS Update

Simply create a new folder under MDT Deployment Share > Operating Systems within Deployment Workbench. Name it something like Windows 10 VERSION x64.

Either create a new Task Sequence for that specific OS version, or edit the Install Operating System task and change to the new OS Version and index number.



## Update PPKG and UNATTEND.XML

A great thing about this application is that the PPKG and UNATTEND.XML can be replaced with new versions with minimal effort. Simply export the new PPKG and UNATTEND.XML from WS1 UEM console and copy to the MDT Deployment Share\Applications\Application folder. In my environment, this is:

E:\MDT\Applications\VMware WS1FPTool

Next, delete the existing PPKG and UNATTEND.XML as only one of each file can reside in the folder.

New deployments will utilise the new PPKG and UNATTEND.XML files. I would suggest you do this when updating major versions of your Tier 1 applications within WS1 UEM.

# RunWS1FP.ps1

The following script uses the Workspace ONE Provisioning Tool for Windows to install the Applications included in the PPKG, deploy the WS1 UEM Agent ready for staging enrolment, creates a Recovery folder structure and necessary files to provide Enterprise Reset capabilities, and runs SYSPREP to configure Windows ready for a zero touch OOBE experience.

The VMwareWS1ProvisioningTool.exe parameters and usage is documented here:

<https://docs.vmware.com/en/VMware-Workspace-ONE-UEM/services/Provisioning_WorkspaceONE/GUID-8B0AC980-D41F-46C8-915D-D6AA4B5262EE.html>

<#

.Synopsis

This powershell script runs the WS1 Factory Provisioning Tool with a PPKG and UNATTEND.XML

and runs SYSPREP then quits.

For use with Microsoft Deployment Toolkit (MDT) as a Command Line Application.

.NOTES

Created: June, 2020

Created by: Phil Helmling, @philhelmling

Organization: VMware, Inc.

Filename: RunWS1FP.ps1

.DESCRIPTION

Runs Workspace ONE Factory Provisioning Tool when used in a MDT Task Sequence

.EXAMPLE

powershell.exe -ep bypass -file .\RunWS1FP.ps1

#>

$current\_path = $PSScriptRoot;

if($PSScriptRoot -eq ""){

#PSScriptRoot only popuates if the script is being run. Default to default location if empty

$current\_path = "C:\Temp";

}

$ppkg = Get-ChildItem -Path $current\_path -Include \*.ppkg\* -Recurse -ErrorAction SilentlyContinue

$unattend = Get-ChildItem -Path $current\_path -Include \*.xml\* -Recurse -ErrorAction SilentlyContinue

$ws1fptool = "VMwareWS1ProvisioningTool.exe"

Start-Process -filepath $current\_path\$ws1fptool -ArgumentList "-a full -p `"$ppkg`" -u `"$unattend`" -q -g" -wait

#Move C:\Temp\PpkgInstaller\PpkgInstallerLog.txt & delete remnant folder

Copy-Item -Path "C:\Temp\PpkgInstaller\PpkgInstallerLog.txt" -Destination "$env:ProgramData\Airwatch\UnifiedAgent\Logs\PpkgInstallerLog.txt"

Remove-Item -Path "C:\Temp" -Recurse -Force

# Build Errors

If you get a build error like the below, it might be the Task Sequence scripts within MDT not detecting the BIOS correctly.

A screenshot of a social media post

Description automatically generated

1. Download the FixUEFIDetection.wsf script from <https://github.com/DeploymentResearch/DRFiles/blob/master/Scripts/FixUEFIDetection.wsf> and save into the MDT Deployment Share\Scripts folder.
2. Edit the Task Sequence > Preinstall > New Computer Only > Add > General > Run Command Line as per below  
   cscript.exe “%SCRIPTROOT%\FixUEFIDetection.wsf”

A screenshot of a cell phone

Description automatically generated