# Managing Surface Devices in the Enterprise – Operating System Deployment with System Center Configuration Manager 2012

This article will describe best practices for enterprise environments that wish to deploy Windows 8.1 Update, drivers, firmware, applications, and configurations to Surface Pro devices managed by using System Center R2 2012 Configuration Manager with Service Pack 1. Additional best practices and several techniques are identified and used in a Configuration Manager task sequence.

## Task Sequence Steps

In order to easily identify the task sequence group or step, it has been assigned a unique number sequence to identify its place within the task sequence. Since task sequence steps can have the same name as other steps, the number sequence will be referenced as part of the step's name. We'll look at a plain task sequence first and then a task sequence that is customized to take advantage of the Surface features more fully.

# Default 'Apply Operating System Image' task sequence

The template that Configuration Manager uses to create a task sequence when you select to install an operating system image will not work well enough on your Surface Pro deployments. Figure 1 shows the default task sequence based on this template. We will need to customize this task sequence in order to install additional software and make configuration changes to the device so that it can provide a stable and compelling experience for users in an enterprise environment.

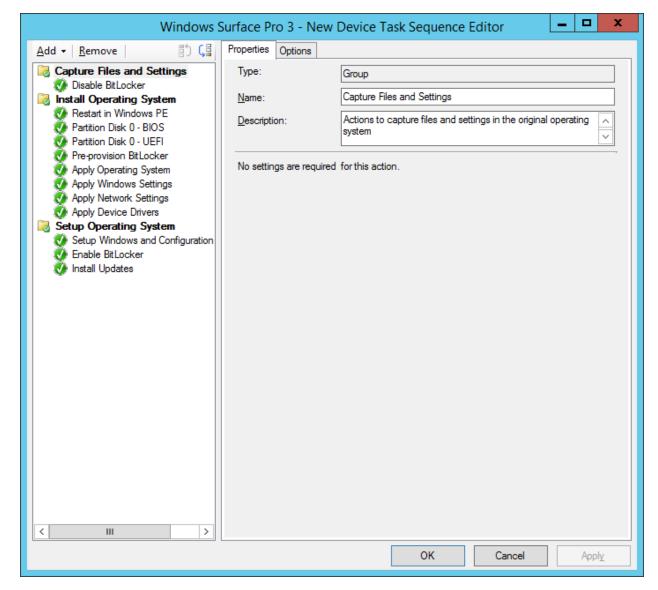


Figure 1: Default operating system image deployment task sequence

## High-Level Overview of Steps

- 1. Prepare software update processes for the operating system image and for future software updates.
- 2. Prepare drivers and firmware packages.
- 3. Customize the boot image.
- 4. Create a task sequence to install the operating system image.
- 5. Prepare the operating system image.
- 6. Determine which apps need to be deprovisioned from the image.
- 7. Determine which new apps need to be installed.
- 8. Deploy the task sequence.
- 9. Revise as needed for changing requirements.

## **Customized Boot Image**

A customized boot image is used here, as described at

http://blogs.technet.com/b/deploymentguys/archive/2015/03/27/reducing-windows-deployment-time-using-power-management.aspx. The customized boot image will contain **PowerCfg.exe** and can help decrease deployment time significantly. The only other customization that may be required for this boot image would be to include the Surface

Ethernet Adapter drivers. The Surface Ethernet Adapter is a USB 3.0 network adapter that can be used on the Surface directly to obtain network connectivity during all phases of the task sequence. The Surface docking station has a built-in Ethernet adapter, which can be used also, but only when the Surface is docked. In practice, no other customizations are necessary for the boot image here. For information on how to customize Windows PE boot images to use in Configuration Manager, see <a href="https://technet.microsoft.com/en-us/library/dn387582.aspx">https://technet.microsoft.com/en-us/library/dn387582.aspx</a>

# Windows Operating System Image

Windows 8.1 with Update in the 64-bit Enterprise Edition is used in this task sequence.

# Software Updates

The Windows 8.1 with Update operating system image used in this task sequence has been serviced by means of scheduling software updates to be injected into the OS Image by the Configuration Manager site server. See <a href="https://technet.microsoft.com/en-us/library/hh397283.aspx">https://technet.microsoft.com/en-us/library/hh397283.aspx</a> for a description of this process. In addition, for task sequence step 1-2-14 – Install Software Updates to succeed, you must deploy a software update group to the All Unknown Computers collection. Since the deployment of this task sequence will be for a device not previously known to Configuration Manager, it is considered "unknown" until the task sequence completes. You must therefore make software update deployments to the All Unknown Computers collection for software updates to install while running the task sequence.

# **Driver Packages**

Two Surface-specific sets of driver packages may be all you need for successful Surface Pro 3 operating system deployment. Refer to Surface Pro driver management documentation for best practices and details on creating and managing these packages.

#### 1. Surface Pro 3 Ethernet Adapter

This is the only adapter driver you need for Windows PE in order to use the Surface Ethernet dongle in the task sequence if you don't use a Surface docking station. The Surface Pro 3 Ethernet Adapter is the *only* driver you need to inject into the boot image(s) used for Surface operating system deployment. Use this boot image as a part of whichever boot media method you choose: PXE, bootable media, or stand-alone installation media. Putting this driver into a separate driver package is optional; however, it may be advisable for future updates to the driver in order to easily manage versioning.

#### 2. Surface Pro 3 Drivers and Firmware

To update the Surface as a part of the task sequence and operating system deployment, you'll want to use the latest Surface firmware and drivers, placed into a drivers package. Then, in the task sequence, call the **Apply Driver Package** task sequence step, referring to this package. This is the only driver package you need to successfully deploy Windows 8.1 Update x64 Enterprise Edition onto the Surface Pro 3. Take the following steps to prepare the Surface drivers and firmware into a package:

- a) Download the latest firmware and drivers package.
- b) Extract the drivers and firmware into a "driver download source" folder.
- c) Create a "drivers package source" folder in another location.
- d) Import all drivers from the "driver download source" folder and create a drivers package using the "drivers package source" folder as the storage location.
- e) Add the drivers package to distribution points closest to the devices where the task sequence will run.

Make sure to review Surface Pro driver management documentation and the download at <a href="http://www.microsoft.com/en-us/download/details.aspx?id=38826">http://www.microsoft.com/en-us/download/details.aspx?id=38826</a>, since the firmware and drivers are updated frequently.

# Customizing Windows Store App Provisioning

In many cases, the default apps that come with Windows 8.1 Update x64 Enterprise Edition may be removed from the surface. Whether that is company policy or whether you're replacing the built-in apps with third-party apps, you have some decisions to make about deprovisioning these apps. If you're creating your own reference image, the best practice is to remove the apps before you capture the image and bring it into Configuration Manager as a WIM. See <a href="https://technet.microsoft.com/en-us/library/hh852635.aspx#RemoveApps">https://technet.microsoft.com/en-us/library/hh852635.aspx#RemoveApps</a> for information on removing provisioned apps from the Windows image. If you did not create a reference image or if you are using Install.wim from the original Windows 8.1 Update x64 Enterprise Edition media, you will want to remove the provisioned apps from the operating system while it is online as a part of the task sequence. Methods for removing provisioned apps and provisioning an app package are discussed in 1-2-11 – Install Surface App and 1-2-12 – Remove Windows Store Apps.

# Task Sequence Deployment

This deployment for the Surface Pro 3 demonstrates preferred settings, which include:

- Deploy to the **All Unknown Computers** collection
- Action Install as Purpose Available
- Make available to the following: Only media and PXE

Making the deployment **Available** makes it less prone to unintended deployments, and targeting **All Unknown Computers** ensures that the operating system deployment task sequence can be installed only on devices that are not already managed by Configuration Manager. Setting the availability to **Only Media and PXE** ensures that only devices that boot from PXE or local media can start this task sequence, so that it can't be initiated from the full-function Configuration Manager client. The other deployment options are at their defaults.

# **Customized Task Sequence**

For a richer deployment experience, the task sequence from Figure 1 has been customized to incorporate some additional techniques and best practices. Figure 2 shows the task sequence. The groups and the individual steps are numbered for reference. This task sequence has been specialized for new Surface Pro 3s. Only a single driver package is used, as are installations for the UEFI tools, asset tag tools, and the Surface App. A discussion of each task sequence group and step follows.

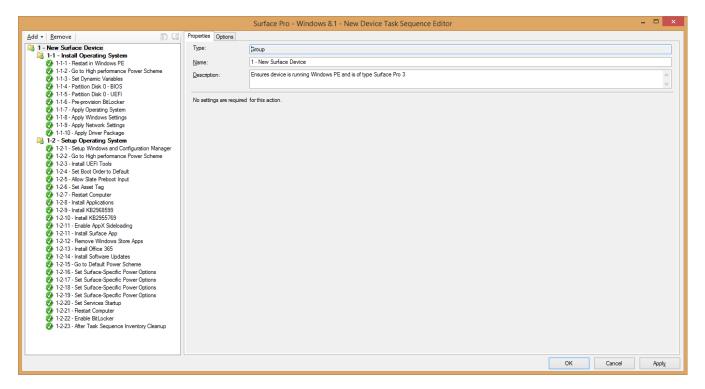


Figure 2: Task sequence specialized for Surface Pro 3 deployment

#### 1 – New Surface Device

This is a root group inserted into the task sequence. All other groups are children of this group, and all steps are children of the subgroups. As shown in Figure 2, **1** – **New Surface Device** is the farthest left object in the task sequence, indicating it is at the root level. For the task sequence to begin running, the conditions specified on the **Options** tab of this group must all evaluate to **true** in order to process subgroups and steps. Figure 3 shows the conditions that must all evaluate to true before the steps inside this group can run.

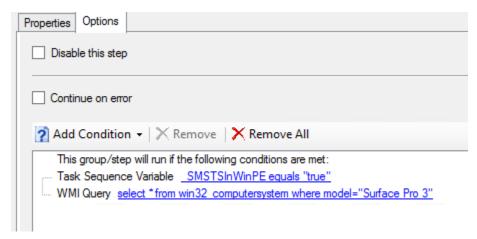


Figure 3: Options for 1 - New Surface Device

First, a task sequence variable named \_SMSTSInWinPE must contain a value of true. This is a read-only variable for task sequences, as documented at <a href="https://technet.microsoft.com/en-us/library/hh273375.aspx">https://technet.microsoft.com/en-us/library/hh273375.aspx</a>. If this condition is true — in other words, if the task sequence has started in our boot image, Windows PE — then we can evaluate the second condition. For the second condition, issue a query into Windows Management Instrumentation (WMI) to detect the computer system model. In this case, you are looking to verify that the computer system model is of type "Surface Pro

3." This data is returned from the WMI query against the device hardware. Here, if you are in Windows PE and are running on a Surface Pro 3, the rest of the task sequence has permission to continue to run.

#### 1-1 – Install Operating System

This is referred to as the "Windows PE phase" because all steps in this group are carried out from inside Windows PE. There are no special conditions attached to this group. All steps inside this group have the "1-1" prefix for ease of reference.

#### 1-1-1 – Restart in Windows PE

Figure 4 shows the settings for **1-1-1** – **Restart in Windows PE**. This step runs only if the task sequence is started in the full Windows operating system. A criterion on the **Options** section of this step ensures that it does not run if the task sequence is running inside Windows PE already. If there is a different Windows PE boot image assigned to the task sequence than the one the client is currently booted into, the boot image assigned to the task sequence will be staged on the local hard disk and the Surface will restart into the staged boot image. This task sequence uses a customized boot image to take advantage of the high-performance power scheme capability.

Properties	Options			
Type:		Restart Computer		
Name:		1-1-1 - Restart in Windows PE		
Description:				
Specify v	Specify what to run after restart:			
● The b	The boot image assigned to this task sequence			
○ The o	The currently installed default operating system			
✓ Notify the user before restarting				
Notification	on message:			
A new M	icrosoft Windov	ws operating system is being installed. The computer must reboot to continue.		

Figure 4: 1-1-1 – Restart in Windows PE

## 1-1-2 – Go to High Performance Power Scheme

To place the currently booted Windows PE into the **High Performance** power scheme, you must run a command to configure the power management capabilities of Windows PE to use that power scheme. Figure 5 shows the **Run Command Line** step with command line:

POWERCFG.EXE /S 8C5E7FDA-E8BF-4A96-9A85-A6E23A8C635C

This will place Window PE into the high-performance power scheme for the duration of this running operating system instance. If Windows PE is rebooted, it will revert to the default **Balanced** power scheme settings. The amount of deployment time you can save by using the high-performance setting varies depending on hardware and network conditions, but you will see a tangible time savings for each deployment. For example, saving 300 seconds in deployment time for a single device may seem trivial, but when multiplied by the thousands of devices you may need to deploy, using the high-performance settings can result in significant time and cost savings. It will be well worth your time to review the blog and adjust your boot images accordingly. See

http://blogs.technet.com/b/deploymentguys/archive/2015/03/27/reducing-windows-deployment-time-using-power-management.aspx for step-by-step instructions.

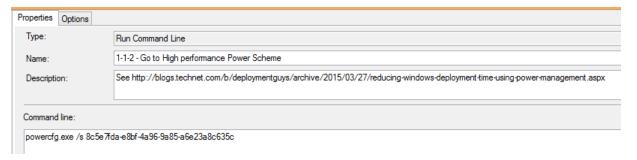


Figure 5: 1-1-2 – Go to high-performance power scheme

## 1-1-3 – Set Dynamic Variables

This step can represent the consolidation of several steps you might have used in the past to set variable value pairs for use later in the task sequence. A single step can conditionally check for specific named conditions and then set existing or new variables to whatever value you specify. Figure 6 shows an example of how you could use this single step to discover the Surface's current physical location (when the task sequence is running) if you have a gateway established for different locations in different time zones. In this example, if the device receives an IP address where the gateway was "10.1.1.1," the step will set the variable OSDTimeZone to "Eastern Standard Time." One of the other gateway IP addresses will set the variable OSDTimeZone to "Pacific Standard Time." This is one example of the usefulness of the Set Dynamic Variables task sequence step. See <a href="https://technet.microsoft.com/en-us/library/hh846237.aspx#BKMK">https://technet.microsoft.com/en-us/library/hh846237.aspx#BKMK</a> SetDynamicVariables for more information.

Properties Options			
Type:	Set Dynamic Variables		
Name:	1-1-3 - Set Dynamic Variables		
Description:			
Dynamic rules and variab	oles:		
The following rules	The following rules and variables will be evaluated in order:		
_	□ IF Default gateway equals "10.1.1.1" THEN		
SET OSDTime	SET_OSDTimeZone = "Eastern Standard Time"		
☐ IF Default gateway	☐ IF Default gateway equals "10.1.2.1" THEN		
SET OSDTime	Zone = "Central Standard Time"		
	equals "10.1.3.1" THEN		
SET OSDTimeZone = "Mountain Standard Time"			
SET OSDTime	Zone = "Pacific Standard Time"		

Figure 6: 1-1-3 – Set Dynamic Variables

## 1-1-4 - Partition Disk 0 - BIOS

Surface Pro 3s do not use Basic Input/Output System (BIOS) firmware. This step is skipped on Surface Pro 3s, so you can remove it from this task sequence with no harmful effects.

#### 1-1-5 – Partition Disk 0 – UEFI

Surface Pro 3s use the Unified Extensible Firmware Interface (UEFI) as their software interface between the operating system and platform. No changes are required of this step to optimize Surface Pro 3

deployments, however the **Recovery** partition is optional and may be removed if desired. For more information about this step, see <a href="https://technet.microsoft.com/en-us/library/hh846237.aspx#BKMK">https://technet.microsoft.com/library/hh824839.aspx</a> and <a href="https://technet.microsoft.com/library/hh824839.aspx">https://technet.microsoft.com/library/hh824839.aspx</a>

## 1-1-6 – Pre-provision BitLocker

If your organization uses BitLocker to encrypt the hard drive of devices, you must carry out this step in the Windows PE phase of the task sequence. With the default settings for the **Pre-provision BitLocker** task sequence step, only the used drive space is encrypted. For this reason, it is a best practice to ensure that files are encrypted as they are written to the hard drive and to make sure this step comes before the **Apply Operating System** step. Unless you plan to significantly change the default disk volume settings, the default options for this step will work just fine. See <a href="https://technet.microsoft.com/en-us/library/hh846237.aspx#BKMK">https://technet.microsoft.com/en-us/library/hh846237.aspx#BKMK</a> PreProvisionBitLocker for more details.

# 1-1-7 – Apply Operating System Image

The **Apply Operating System Image** step is where things start to get interesting in your Surface deployment task sequence.

In most cases, you already will have prepared an image for use in Configuration Manager operating system deployment. There are a number of good resources on this topic:

- How to Manage Operating System Images and Installers in Configuration Manager: https://technet.microsoft.com/en-us/library/hh397283.aspx
- Create a Windows 8.1 Reference Image: <a href="https://technet.microsoft.com/library/dn744290.aspx">https://technet.microsoft.com/library/dn744290.aspx</a>
- Desktop Image Management: Build a Better Desktop Image: <a href="https://technet.microsoft.com/en-us/magazine/ff721826.aspx">https://technet.microsoft.com/en-us/magazine/ff721826.aspx</a>

In most cases, for enterprise deployments of the operating system on Surface devices, an operating system image will be created that will already have certain line-of-business applications installed. Figure 7 shows the selection of an operating system to deploy based on a captured image.

Properties Options			
Туре:	Apply Operating System Image		
Name:	1-1-7 - Apply Operating System		
Description:	Actions to apply operating system		
	em from a captured image		
lmage package:	Windows 8.1 Upd x64 Surface en-US		
Image:	2-2		
<ul> <li>Apply operating syst</li> </ul>	em from an original installation source		
Package:			
Edition:			
✓ Use an unattended or Sysprep answer file for a custom installation  Package: Surface OSD Answer Files			
File name:	SP3.xml		
	re you want to apply this operating system.		
Destination:	Next available formatted partition		

Figure 7: 1-1-7 – Apply Operating System

Once you have your operating system image created, imported into Configuration Manager, and residing on distribution points, you will want to consider the use of your own unattended answer file. As discussed in Step 9 at <a href="http://blogs.technet.com/b/askcore/archive/2014/07/15/deploy-windows-to-surface-pro-3-using-microsoft-deployment-toolkit.aspx">http://blogs.technet.com/b/askcore/archive/2014/07/15/deploy-windows-to-surface-pro-3-using-microsoft-deployment-toolkit.aspx</a> you can create an unattended XML file, add it to a package, place it on distribution points, and reference it in this task sequence.

Figure 8: SP3.xml file used in task sequence

Notice in Figure 8 that **HideWirelessSetupInOOBE** is set to a value of **true** so that the users' first experience with the Surface Pro does not require them to select a wireless access point. In most cases, the wireless access point policies will be set via Active Directory Group Policy or some other means. The setting to ensure that the power button is displayed on the start screen,

**ShowPowerButtonOnStartScreen**, is also enabled. Note that this unattended XML file is merged with an unattended XML file created in the background by the task sequence at run time during the system deployment process. Any settings that are specified in your customer XML file become the effective values used for the operating system when deployed.

#### 1-1-8 – Apply Windows Settings

In step 1-1-8 – Apply Windows Settings, things start to get interesting. In step 1-1-7 – Apply Operating System, you installed the Enterprise Edition of Windows 8.1 Update x64. This means that you typically do not need to specify a value for Product Key here, as there will be a KMS Server configured in DNS and Active Directory to give this device a product identifier (PID) and activate. See <a href="http://www.microsoft.com/en-us/licensing/existing-customer/FAQ-product-activation.aspx">http://www.microsoft.com/en-us/licensing/existing-customer/FAQ-product-activation.aspx</a> for an FAQ on volume license keys. In Figure 9, values for User name and Organization name come from the values of the task sequence variables "OSDRegisteredUserName" and "OSDRegisteredOrgName", respectively. The value of these variables could have been set in step 1-1-3 – Set Dynamic Variables or in any other task sequence step that sets the value of these variables. Using these variables in this task sequence step is optional, and in this case the value of the variables was set as collection variables on the All Unknown Computers collection. Setting the value of these variables overrides anything manually specified in this step, so take care to set these appropriately for your organization. Figure 10 shows the Collection Variables tab of the properties for the All Unknown Computers collection.

Properties Options			
Type:	Apply Windows Settings		
Name:	1-1-8 - Apply Windows Settings		
Description:	Actions to apply Windows settings		
Enter licensing and registration information for installing Windows.			
User name:	%OSDRegisteredUserName%		
Organization name:	%OSDRegisteredOrgName%		
Product key:			
Server licensing:	Do not specify		
Maximum connections:	5 🚖		
Randomly generate the local administrator password and disable the account on all supported platforms (recommended)     Enable the account and specify the local administrator password			
Password:			
Confirm password:			
Select the default time zone for this installation of Windows.			
Time zone:	(UTC-05:00) Eastern Time (US & Canada)		

Figure 9: 1-1-8 – Apply Windows settings

Local administrator password is left at the default value, to randomly generate a password. This setting is overridden, however, because of the use of the collection variables **OSDRandomAdminPassword** being set to a value of **false** and **OSDLocalAdminPassword** being set to a secret value. The value of this collection variable can be set by key security staff or administrators other than the Configuration Manager administrator in order to preserve security integrity for deployed systems.

Use the **Time zone** setting displayed in Figure 9 unless the variable **OSDTimeZone** contains a different value. Step **1-1-3 – Set Dynamic Variables** provides an example of when it might make sense to set a different time zone. To get a complete list of all time zone IDs from which to choose for your task sequence, open a command prompt and type **TZUTIL/L** 

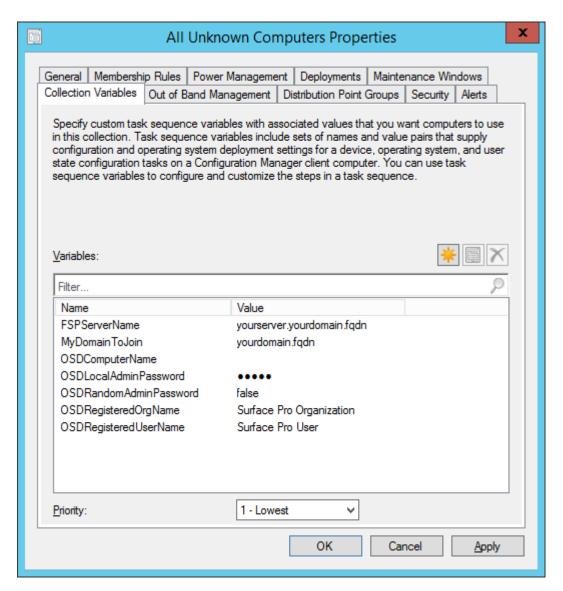


Figure 10: Collection variables set on the All Unknown Computers collection

Note also in Figure 10 that **OSDComputerName** is set to a blank value. Because this variable is used to join this system to the domain, it must contain a value. In this case, the task sequence will prompt for the value of **OSDComputerName** before the task sequence starts. If you require complete automation in your task sequence deployments, use a pre-start command to populate the value of this variable as described at <a href="https://technet.microsoft.com/en-us/library/ji651034.aspx">https://technet.microsoft.com/en-us/library/ji651034.aspx</a>

#### 1-1-9 – Apply Network Settings

In Figure 11, use the collection variable %MyDomainToJoin% set on the All Unknown Computers collection to specify the value for the name of the domain to join. OSDDomainName could also be set as a collection variable. As before, if you set the value of the built-in variable in a previous step, setting the variable in this step is ignored. You can set either the NetBIOS domain name or the Fully Qualified Domain Name for the value. Figure 10 shows the value of this variable set to yourdomain.fqdn. Of course, the username and password pair specified under Account must be valid for the Active Directory in which you wish to join this computer account. You can find the technical reference for accounts used in Configuration Manager at <a href="https://technet.microsoft.com/en-us/library/hh427337.aspx">https://technet.microsoft.com/en-us/library/hh427337.aspx</a>. The account name and password could also be specified as collection variables on All Unknown Computers for the task sequence variables OSDJoinAccount and OSDJoinPassword. Review <a href="https://technet.microsoft.com/en-us/library/0b3df5ab-dce7-4dcf-a49e-3bf046798076#BKMK">https://technet.microsoft.com/en-us/library/0b3df5ab-dce7-4dcf-a49e-3bf046798076#BKMK</a> ApplyNetworkSettings for a full list of other variables that could be set prior to this task sequence step for this step to consume.

Ī	Properties Options			
	Type:	Apply Network Settings		
	Name:	1-1-9 - Apply Network Settings		
	Description:	Actions to configure network settings		
	If you did not specify a do	omain or workgroup setting during the capture step		
	O Join a workgroup			
	Workgroup:			
	<ul><li>Join a domain</li></ul>			
	Domain:	%MyDomainToJoin%		
	Domain OU:			
	Enter the account that h	as permission to join the computer to any domain		
	Account:	MYDOMAIN\DomainJoinAcct		

Figure 11: 1-1-9 – Apply network settings

## 1-1-10 – Apply Driver Package

Since this task sequence is specialized for Surface Pro, it doesn't need to contain complex logic to determine the proper drivers to install. Figure 12 shows the **1-1-10 – Apply Driver Package** step, which applies drivers only from the **Surface Pro 3** driver package. You must create this package and send it to distribution points close to where the Surface will be running the task sequence since it is over 400 MB in size and contains drivers and firmware to be installed on the Surface. Since the drivers are specific to Surface Pro 3, it is a good idea to use a condition on the **Options** tab so the drivers apply only when the computer system model is "Surface Pro 3," as depicted in Figure 3. Refer to Surface Pro driver management documentation for best practices and details on creating and managing this package.

Properties	Options	
Type:		Apply Driver Package
Name:		1-1-10 - Apply Driver Package
Description	on:	
Select th	e driver packa	ge containing drivers to be made available during Windows setup.
Driver Pa	ockage:	Surface Pro 3

Figure 12: 1-1-10 – Apply driver package

#### 1-2 – Set Up Operating System

The task sequence now moves into the **1-2 – Setup Operating System** group, as Figure 13 shows. This phase of the task sequence may be referred to as the "local operating system" phase, because the task sequence is finished with Windows PE and will shortly reboot into the deployed operating system on the local hard drive.

Properties Options		
Туре:	Group	
Name:	1-2 - Setup Operating System	
Description:	Setup Operating System	
No settings are required for this action.		

Figure 13: 1-2 – Set up operating system

# 1-2-1 – Set up Windows and Configuration Manager

There's a lot going on behind the scenes with this step. See <a href="https://technet.microsoft.com/en-us/library/6b116f87-a1df-4194-ad57-f01d797b7d13#BKMK">https://technet.microsoft.com/en-us/library/6b116f87-a1df-4194-ad57-f01d797b7d13#BKMK</a> SetupWindowsandConfigMgr for the full set of steps. Final phases of Windows setup consume the settings from the unattended XML file, as well as download and install the Configuration Manager Client. While the Configuration Manager Client installation properties are all optional, it's best to specify the management point fully qualified domain name and the fully qualified domain name of the fallback status point server at the site to which this client will be assigned. As Figure 14 shows, the FSP and SMSMP client installation properties are listed. For the full set of Configuration Manger Client installation properties, see <a href="https://technet.microsoft.com/en-us/library/gg699356.aspx">https://technet.microsoft.com/en-us/library/gg699356.aspx</a>. Note: SMSSITECODE is not supported here in the task sequence. Once this step has completed, more steps later in the task sequence can use packages, applications, and software updates deployments.

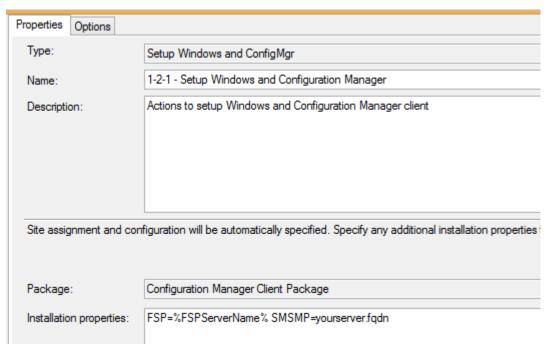


Figure 14: 1-2-1 – Set Up Windows and Configuration Manager

#### 1-2-2 – Go to High Performance Power Scheme

Regardless of the operating system image source and the default power policy chosen, the deployment should complete as quickly and efficiently as possible. The command shown in Figure 15, 1-2-2 — Go to High Performance Power Scheme, will put the deployed operating system into high-performance power settings. Note that the PowerCfg.exe run by this step is a part of the local operating system. The Globally Unique Identifier (GUID) is that of the high-performance power scheme. From a PowerShell window, use

POWERCFG /L to see all power schemes on the local system. The listed power scheme with the asterisk (\*) after the friendly name is the active scheme. You don't need administrative rights to list the schemes, but you do need them to set the active scheme.

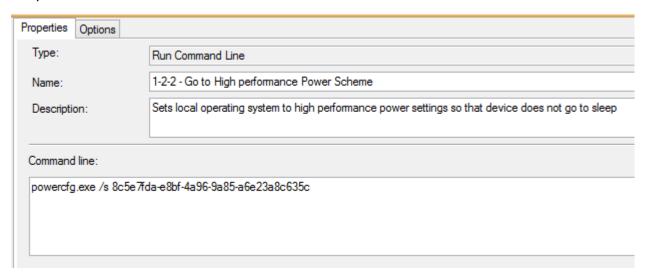


Figure 15: 1-2-2 – Go to high-performance power scheme

#### 1-2-3 - Install UEFI Tools

If the Surface Pro 3 drivers and firmware package you use is from March 26, 2015, or later, you will be able to install the UEFI Manager tool as a part of the task sequence. The Surface Pro 3 Firmware tools are available from <a href="http://download.microsoft.com/download/9/D/5/9D5FE91C-A294-45AF-98D3-016C36213962/Surface%20Firmware%20Tool.msi">http://download.microsoft.com/download/9/D/5/9D5FE91C-A294-45AF-98D3-016C36213962/Surface%20Firmware%20Tool.msi</a> and are described at <a href="https://technet.microsoft.com/en-us/windows/dn965440">https://technet.microsoft.com/en-us/windows/dn965440</a>. Once you have the MSI downloaded, it is easy to set it up as an application or as a package in Configuration Manager to use in your task sequence, as shown in Figure 16.

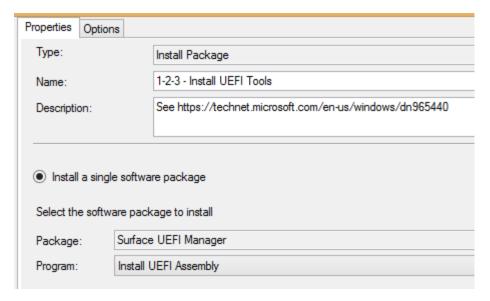


Figure 16: 1-2-3 - Install UEFI Tools

#### 1-2-4 – Set Boot Order to Default

Once the UEFI tools are installed, you can configure various options in the UEFI firmware from a script. In **1-2-4 – Set Boot Order to Default**, use the task sequence step to call a PowerShell script. Figure 17 refers to the PowerShell script that sets the device boot order. In this case, Figure 18 shows the PowerShell script that sets the boot device to only SSD. This may save time in the task sequence and subsequent reboots of the Surface if it had been installed via a network installation server, such as the PXE boot

capabilities of Configuration Manager. Another option may be to set the UEFI password and disable the USB port. Sample scripts for these configuration settings can be found at <a href="https://technet.microsoft.com/en-us/windows/dn965440">https://technet.microsoft.com/en-us/windows/dn965440</a>

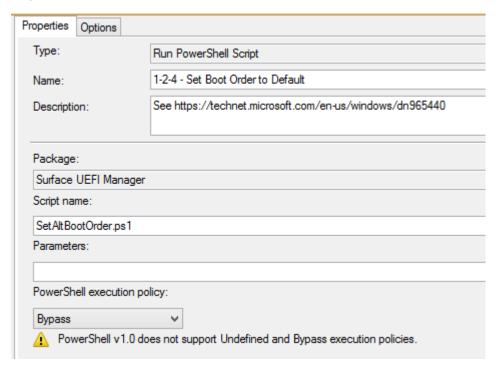


Figure 17: 1-2-4 – Set boot order to default

Figure 18: PowerShell script fragment showing use of the UEFI manager to set device boot order

#### 1-2-5 - Allow Slate Preboot Input

Before you can enable BitLocker on the Surface as a part of the task sequence, you need to add a registry entry in the task sequence step **1-2-5 – Allow Slate Preboot Input** as shown in Figure 19. This will allow the Surface auto-unlock as a part of the boot process. This, and other registry entries, are documented at <a href="https://technet.microsoft.com/en-us/library/jj721578.aspx">https://technet.microsoft.com/en-us/library/jj721578.aspx</a> and <a href="https://technet.com/b/askpfeplat/archive/2014/07/14/bitlocker-pin-on-surface-pro-3-and-other-tablets.aspx">https://technet.com/b/askpfeplat/archive/2014/07/14/bitlocker-pin-on-surface-pro-3-and-other-tablets.aspx</a>. In whatever pre-boot authentication BitLocker mode you require for the Surface, this registry setting will help the device configuration process. This setting also is available as a Group Policy configuration as Figure 20 shows, but is set here for the deployment of this task sequence.

Properties	Options				
Туре:		Run Command Line			
Name:		1-2-5 - Allow Slate Preboot Input			
Description: See Windows to Go deployment script, https://technet.microsoft.com/en-us/library/jj721578.aspx		See Windows to Go deployment script, https://technet.microsoft.com/en-us/library/ij721578.aspx			
Command	l line:				
reg add '	reg add "HKLM\Software\Policies\Microsoft\FVE" /v OSEnablePrebootInputProtectorsonSlates /t REG_DWORD /d 1 /f				
Disabl	Disable 64-bit file system redirection				
Start in:					
Package:					
✓ Time-out (minutes):					

Figure 19: 1-2-5 - Allow Slate Preboot Input

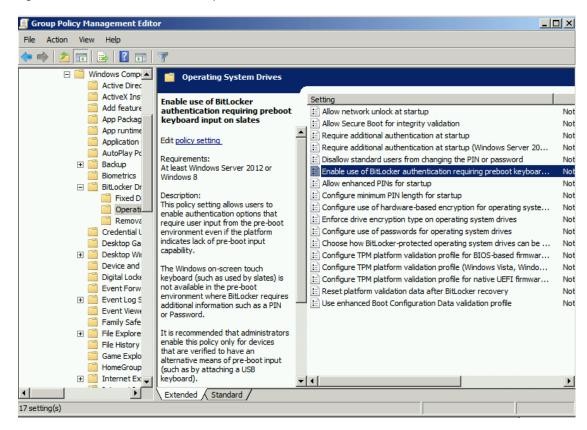


Figure 20: Group Policy showing the preboot authentication keyboard input setting

#### 1-2-6 – Set Asset Tag

Many organizations require asset tags for inventory management of their devices. Surface Pro 3 now allows automation when setting the asset tag, via the Surface Pro 3 Asset Tag CLI Utility, available from <a href="https://www.microsoft.com/en-us/download/details.aspx?id=44076">https://www.microsoft.com/en-us/download/details.aspx?id=44076</a>. The download comprises a single ZIP file, which contains **AssetTag.exe**. As Figure 21 shows, **AssetTage.exe** was added to a Configuration Manager package, but is used from a **Run Command Line** step. Using this method, a task sequence variable can be passed to the command line directly in order to set the value of the asset tag on the device. The asset tag value must contain between one and 36 characters, limited to A-Z, a-z, 0-9, period (.) and hyphen (-), so besides **%OSDComputerName%**, other options for setting the asset tag are the MAC address (**%OSDAdapterOMacAddress %**) or even the Configuration Manager client GUID (**%\_SMSTSClientGUID%**).

Properties Options			
Type:	Run Command Line		
Name:	1-2-6 - Set Asset Tag		
Description:	See https://www.microsoft.com/en-us/download/details.aspx?id=44076		
Command line:			
AssetTag -s %OSDComputerName%			
☐ Disable 64-bit file system redirection			
Start in:			
✓ Package:			
Surface Asset Tag Tool			

Figure 21: 1-2-6 - Set Asset Tag

## 1-2-7 — Restart Computer

In order to commit the changes to the asset tag, reboot the Surface. Figure 22 shows the settings for the **Restart Computer** task sequence action used to do this. This step is optional, unless no other steps later in the task sequence might reboot the Surface. It's a good idea to commit pending changes before moving on to make other changes to the Surface, whether to the firmware or the operating system.

Properties	Options	
Type:		Restart Computer
Name:		1-2-7 - Restart Computer
Description	on:	Restart to committ asset tag settings
Specify v	what to run after	restart:
The boot image assigned to this task sequence		
<ul> <li>The currently installed default operating system</li> </ul>		

Figure 22 1-2-7 - Restart computer

# 1-2-8 — Install Applications

You can install as many as nine applications with a single **Install Application** step; however, the example in Figure 23 shows only two applications. If you need to install more than nine applications with the task sequence, you can use another **Install Application** task sequence step immediately after this step. You also can use a dynamic variable list to process up to 99 in a single step. See <a href="https://technet.microsoft.com/en-us/library/hh846237.aspx#BKMK">https://technet.microsoft.com/en-us/library/hh846237.aspx#BKMK</a> InstallApplication for details.

Properties	Options				
Type:		Install A	Install Application		
Name:	Name:		1-2-8 - Install Applications		
Description	on:				
Install the	ne followir	ng applications			
Name			Status	Administrator Comments	
.NET Fra	mework 4	4.5.2	Valid		
ConfigM	gr 2012 T	oolkit R2	Valid		
O Install a	Install applications according to dynamic variable list				
The list of	The list of applications consists of a series of task sequence variables with a common base				
Base variable name:					
✓ If an application installation fails, continue installing other applications in the list					

Figure 23: 1-2-8 – Install Applications

#### 1-2-9 - Install KB2968599

Microsoft support knowledge base article KB2968599, <a href="https://support.microsoft.com/en-us/kb/2968599/en-us">https://support.microsoft.com/en-us/kb/2968599/en-us</a>, details the Quick Note-Taking Experience for Windows 8.1 features. The download for this update comes as an MSU file. As Figure 24 shows, it is added to a Configuration Manager package and the command line run so that no user interface is displayed and any reboot is suppressed. This step, and the next step, 1-2-10 – Install KB2955769, could both be done in a single step with a single package, by adding both command lines to a script and calling the script from a Run Command Line task sequence step. They are shown separately here because operating system deployment is often an iterative process, where refinements and consolidation of several related steps is often possible, and these are good examples of prime consolidation candidates.

Properties Options			
Type:	Run Command Line		
Name:	1-2-9 - Install KB2968599		
Description:	Quick Note-Taking Experience Feature		
Command line:			
WUSA Windows8.1-KB2968599-x64.msu /quiet /norestart			
Disable 64-bit file	e system redirection		
Start in:			
Package:			
KB2968599			

Figure 24: 1-2-9 – Install KB2968599

# 1-2-10 — Install KB2955769

This step addresses an issue to allow power policy options to enable preferred connected standby settings on Surface Pro 3s with Windows 8.1 Update.

Properties Options				
Туре:	Run Command Line			
Name:	1-2-10 - Install KB2955769			
Description:	Power poilicy options to enable preferred connected standby settings			
Command line:				
WUSA Windows8.1-KB2955769-v3-x64.msu /quiet /norestart				
Disable 64-bit file system redirection				
Start in:				
✓ Package:				
KB2955769				

Figure 25: 1-2-10 – Install KB2955769

# 1-2-11 — Enable AppX Sideloading

Until your organization's group policy can be applied to the Surface Pro being imaged, you will need to add some registry entries to sideload applications onto Windows 8.1 Enterprise with Update. The steps

**1-2-11** – **Enable AppX Sideloading** and **1-2-11** – **Install Surface App** are number-matching companions, in that the registry entries must be in place before you sideload the Surface App as a part of the task sequence. These steps are also good candidates for consolidation into a single step.

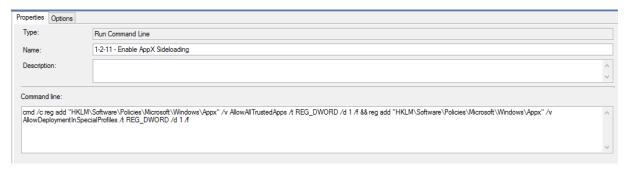


Figure 26: 1-2-11 - Enable AppX Sideloading

# 1-2-11 – Install Surface App

Surface App is a unique and immersive app experience available for use on Surface devices, available as a Windows Store app, but also as an app package download for sideloading. Learn more about sideloading at <a href="https://technet.microsoft.com/en-us/library/hh852635.aspx">https://technet.microsoft.com/en-us/library/hh852635.aspx</a>. You may need to enable the AllowTrustedApps and AllowDeploymentInSpecialProfiles registry settings in the task sequence before this step to allow the Surface App to install. The Surface App provides you with customization options and the opportunity to learn more about the Surface. See <a href="http://www.microsoft.com/en-us/download/details.aspx?id=46703">http://www.microsoft.com/en-us/download/details.aspx?id=46703</a> for the download.

Properties Options				
Type:	Run Command Line			
Name:	1-2-11 - Install Surface App			
Description:				
Command line:				
PowerShell -Executionpolicy Bypass -File Install.ps1				
Disable 64-bit file system redirection				
Start in:				
✓ Package:				
Surface App				

Figure 27: 1-2-11 – Install Surface App

Figure 28: PowerShell script showing DISM cmdlets to sideload the Surface App

#### 1-2-12 – Remove Windows Apps

By default, Windows 8.1 Update Enterprise Edition contains Windows apps such as

Microsoft.XboxLIVEGames and Microsoft.BingFoodAndDrink. For a complete list of the Windows apps in the operating system by default, open an elevated PowerShell window and enter Get-AppxPackage –

ALLUSERS | SELECT Name. Some enterprise environments require that apps that are not approved be removed from devices, or not installed in the first place. Figure 29 shows a command to use to deprovision a Windows app from the Surface during deployment. You can write a script to remove more than one app at a time, but only a single app is listed here. Note that the PackageName parameter contains the fully decorated app package name. The full app package name can be obtain in an elevated PowerShell window with the following command: Get-AppxProvisionedPackage -Online | Select PackageName. In task sequence step 1-2-12 - Remove Windows Store Apps, the DISM command is used instead of the PowerShell Remove-AppxProvisionedPackage cmdlet. See

https://technet.microsoft.com/library/dn376474.aspx and https://technet.microsoft.com/enus/library/hh852635.aspx for more information.

Properties Options	
Type:	Run Command Line
Name: 1-2-12 - Remove Windows Store Apps	
Description:	
Command line:	
dism.exe /online /f	Remove-ProvisionedAppxPackage /PackageName:Microsoft.BingFoodAndDrink_2015.423.9.2481_neutral_~_8wekyb3d8bbwe

Figure 29 1-2-12 – Remove Windows Store Apps

#### 1-2-13 - Install Office 365

There are valid points on both sides of the argument about whether to include Office installations in the base image. Prior to **Office 2013 with Service Pack 1** and **Office 365 ProPlus**, it might have been faster and easier to include Office in the base image. It may no longer be so, and the convenience of not having Office in the base image might outweigh the drawbacks. This frees Configuration Manager administrators from combining a specific version of Office with a specific version of Windows, and the burdens of updating each. Figure 29 shows **Office 365 ProPlus** being installed as a part of the task sequence deployment. For details about **Office 365 ProPlus** deployment, see <a href="https://technet.microsoft.com/en-us/library/gg715562%28v=office.15%29.aspx">https://technet.microsoft.com/en-us/library/gg715562%28v=office.15%29.aspx</a>, and for details about creating an application in Configuration Manager for **Office 365 ProPlus**, see <a href="https://technet.microsoft.com/en-us/library/dn708063.aspx">https://technet.microsoft.com/en-us/library/dn708063.aspx</a>

Properties Options					
Type:	Install Package				
Name:	1-2-13 - Install Office 365				
Description:	Install Office 365, see https://technet.microsoft.com/en-us/library/gg715562%28v=office.15%29.aspx				
Install a single software package					
Select the software package to install					
Package:	Office 365				
Program:	Install				

Figure 30: 1-2-13 - Install Office 365

#### 1-2-14 – Install Software Updates

After all applications and configuration settings are in place for the Surface deployment, it's a good idea to install any approved software updates that might be detected. For an introduction to software updates in Configuration Manager, see <a href="https://technet.microsoft.com/en-us/library/gg682168.aspx">https://technet.microsoft.com/en-us/library/gg682168.aspx</a>, and for best practices in software updates see <a href="https://technet.microsoft.com/en-us/library/hh692394.aspx">https://technet.microsoft.com/en-us/library/hh692394.aspx</a>. You must deploy a software updates group to the All Unknown Computers collection for the task sequence to detect applicability and install any of the software updates. This is because the Surface is still not fully known to Configuration Manager, so is still getting deployment policy and collection variables from the All Unknown Computers collection. Whether deploying a server operating system on server hardware or Windows onto a Surface, there must be a software updates deployment for this setting to work. In Figure 31, the All Software Updates setting is selected so the device is as secure and as fully patched as possible before being turned over to the end user.

Properties Options				
Type:	Install Software Updates			
<u>N</u> ame:	1-2-14 - Install Software Updates			
<u>D</u> escription:	Detect and install deployed software updates			
Install software updates assigned to the destination computer.				
○ <u>M</u> andatory Software Updates				
All <u>S</u> oftware Updates				

Figure 31: 1-2-14 – Install Software Updates

#### 1-2-15 – Go to Default Power Scheme

The Surface Pro 3 uses an optimized power plan to take advantage of the connected standby features. Since this task sequence employed the high-performance power scheme during the Windows PE phase and the early part of the local OS phase, it is now time to begin applying settings on the Surface so the end users can enjoy the optimized power plan. Figure 31 shows the command to place the local operating system back into the **Balanced** power plan. Just as in step 1-2-2 – **Go to High Performance Power Scheme**, the GUID referenced here points to a built-in power plan otherwise known as **Balanced**. Note that the **PowerCfg.exe** run by this step is a part of the local operating system. From a PowerShell

window, use PowerCFG /L to see all power schemes on the local system. The listed power scheme with the asterisk (\*) after the friendly name is the active scheme. You don't need administrative rights to list the schemes, but you do need them to set the active scheme.

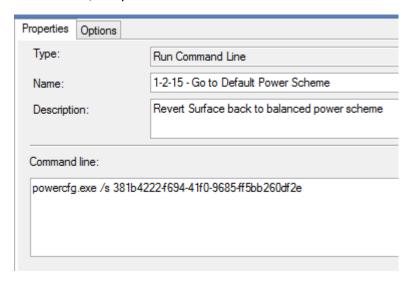


Figure 32: 1-2-15 – Go to the default power scheme

#### 1-2-16 – Set Surface-Specific Power Options

Once the active power plan has been changed to **Balanced** in step **1-2-15** – **Go to Default Power Scheme**, there are still some optimizations required to take full advantage of the connected standby features. The commands used in **1-2-16** – **Set Surface-Specific Power Options**, **1-2-17** – **Set Surface-Specific Power Options**, and **1-2-19** – **Set Surface-Specific Power Options** can all be combined into a single step that runs a script successfully. They are unwound here for demonstration purposes.

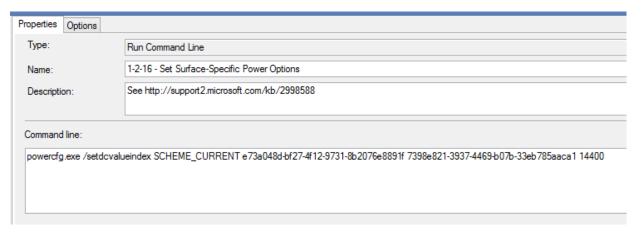


Figure 33: 1-2-16 – Set Surface-specific power options

# 1-2-17 – Set Surface-Specific Power Options

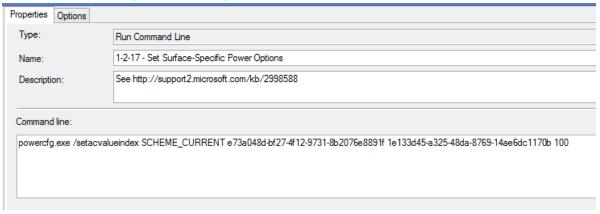


Figure 34: 1-2-17 – Set Surface-specific power options

# 1-2-18 – Set Surface-Specific Power Options

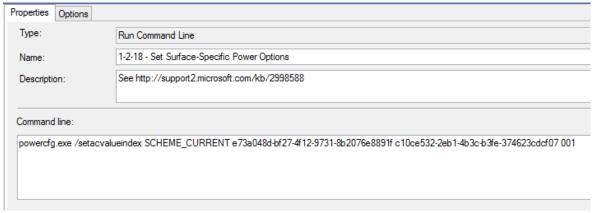


Figure 35: 1-2-18 – Set Surface-specific power options

#### 1-2-19 — Set Surface-Specific Power Options

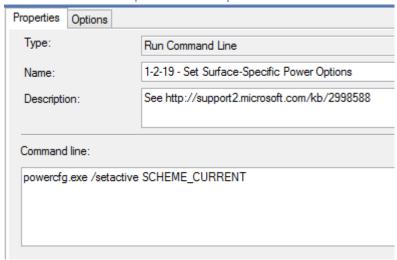


Figure 36: 1-2-19 – Set Surface-specific power options

#### 1-2-20 — Set Services Startup

To avoid certain remote procedure call (RPC) issues described in <a href="https://support.microsoft.com/en-us/kb/2976660">https://support.microsoft.com/en-us/kb/2976660</a>, add the following registry setting to the deployed operating system. This and other registry settings added or changed are good candidates for consolidation into a single script and single task sequence step.

Properties Options			
Туре:	Run Command Line		
Name:	1-2-20 - Set Services Startup		
Description:			
Command line:			
reg add "HKLM\SYSTEM\CurrentControlSet\Services\gpsvc" /v Type /t REG_DWORD /d 0x10 /f			

Figure 37: 1-2-20 – Set services startup

#### 1-2-21 – Restart Computer

Step **1-2-21** – **Restart Computer** will reboot the computer and commit any pending changes. This will be the last reboot of the system before it is ready for end users to log in.

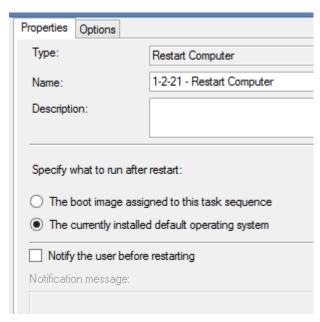


Figure 38: 1-2-21 - Restart computer

#### 1-2-22 – Enable BitLocker

For organizations that require the use of whole-disk encryption, BitLocker provides an integrated solution. Detailed at <a href="http://blogs.technet.com/b/askpfeplat/archive/2014/07/14/bitlocker-pin-on-surface-pro-3-and-other-tablets.aspx">https://blogs.technet.com/b/askpfeplat/archive/2014/07/14/bitlocker-pin-on-surface-pro-3-and-other-tablets.aspx</a>, the **TPM only** option is selected here to represent the best balance between security and usability. In addition, the option to store the recovery key **In Active Directory Domain Services** is used to provide recovery options. To make sure your organization is prepared, review <a href="https://technet.microsoft.com/library/jj592683.aspx?ocid=wc-tn-security">https://technet.microsoft.com/library/jj592683.aspx?ocid=wc-tn-security</a> for policies and configuration requirements. Lastly, everything you ever wanted to know about BitLocker is probably covered at <a href="https://technet.microsoft.com/en-us/library/hh831507.aspx">https://technet.microsoft.com/en-us/library/hh831507.aspx</a>.

Properties	Options				
Type:		Enable BitLocker			
Name: Description:		1-2-22 - Enable BitLocker			
		Enable BitLocker			
	Choose the drive to encrypt  © Current operating system drive				
Cho	oose the key ma	nagement to use for the oper	ating system drive		
•	TPM only				
Startup Key on USB only		USB only	First available USB flash drive		
○ TPM and Startup Key on USB		up Key on USB	First available USB flash drive		
0	TPM and PIN				
O Specific Drive			C:		
Choose	e where to creat	te the recovery key			
In Active Directory Domain Services					
O Do not create a recovery key (not recommended)					
Wait for BitLocker to complete the drive encryption process on all drives before Configuration Manager continues to run the task sequence					
To use this task sequence step, the Windows operating system must support BitLocker.					

Figure 39: 1-2-22 - Enable BitLocker

## 1-2-23 - After Task Sequence Inventory Cleanup

Figure 40 shows setting the value of the task sequence variable **SMSTSPostAction**. The value of this variable is set as a command that will run after the task sequence executes. In this case, a five-minute delay is added to the command to allow enough time to remove the Surface from a docking station if used, and then **WMIC.exe** is used to make method calls in the Configuration Manager client namespace to perform a heartbeat discovery followed by a hardware inventory. This command triggers a client-side schedule, specifically to carry out the heartbeat discovery and the hardware inventory. The sooner the Configuration Manager site server receives the client hardware discovery and inventory data, the sooner the client is added to collections whose query-based rules include classes from hardware inventory. Other ideas for this command would be to set as the value of **SMSTSPostAction** to shut the Surface down after the task sequence completes, or call a script that enables write filters on embedded devices after the task sequence deploys an operating system to the device. See <a href="https://technet.microsoft.com/en-us/library/hh273375.aspx">https://technet.microsoft.com/en-us/library/hh273375.aspx</a> for more information about the **SMSTSPostAction** variable.

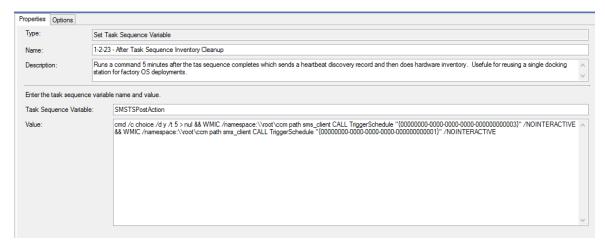


Figure 40: 1-2-23 – After task sequence inventory cleanup

# List of resources and references used to customize the task sequence

- Surface Software, Firmware, and Drivers
   http://www.microsoft.com/en-us/download/details.aspx?id=38826
- Surface Pro 3 Asset Tag CLI Utility https://www.microsoft.com/en-us/download/details.aspx?id=44076
- Installing Office 365
   <a href="https://technet.microsoft.com/en-us/library/gg715562%28v=office.15%29.aspx">https://technet.microsoft.com/en-us/library/gg715562%28v=office.15%29.aspx</a>
- Setting Surface Pro 3 Power Settings on Enterprise Install
   http://blogs.technet.com/b/askcore/archive/2014/11/05/surface-pro-3-hibernation-doesn-t-occur-on-enterprise-install.aspx
- Reducing Windows Deployment Time Using Power Management
   http://blogs.technet.com/b/deploymentguys/archive/2015/03/27/reducing-windows-deployment-time-using-power-management.aspx
- Deploying Windows to Surface Pro 3 using Microsoft Deployment Toolkit
   <a href="http://blogs.technet.com/b/askcore/archive/2014/07/15/deploy-windows-to-surface-pro-3-using-microsoft-deployment-toolkit.aspx">http://blogs.technet.com/b/askcore/archive/2014/07/15/deploy-windows-to-surface-pro-3-using-microsoft-deployment-toolkit.aspx</a>
- Advanced UEFI Security Features for Surface <a href="https://technet.microsoft.com/en-us/windows/dn965440">https://technet.microsoft.com/en-us/windows/dn965440</a>
- Deploying Windows to Go in your organization https://technet.microsoft.com/en-us/library/jj721578.aspx
- Installing the Surface App https://technet.microsoft.com/en-us/windows/dn705824
- Deploying Windows 8.1 with System Center 2012 R2 Configuration Manager <a href="https://technet.microsoft.com/library/dn744284.aspx">https://technet.microsoft.com/library/dn744284.aspx</a>

- How to Update the Surface Pro 3 Firmware Offline using a USB Drive
   <a href="http://blogs.technet.com/b/askpfeplat/archive/2014/10/20/how-to-update-the-surface-pro-3-firmware-offline-using-a-usb-drive.aspx">http://blogs.technet.com/b/askpfeplat/archive/2014/10/20/how-to-update-the-surface-pro-3-firmware-offline-using-a-usb-drive.aspx</a>
- How to Use The Same External Ethernet Adapter For Multiple SCCM OSD
   <a href="http://blogs.technet.com/b/askpfeplat/archive/2014/07/28/how-to-use-the-same-external-ethernet-adapter-for-multiple-sccm-osd.aspx">http://blogs.technet.com/b/askpfeplat/archive/2014/07/28/how-to-use-the-same-external-ethernet-adapter-for-multiple-sccm-osd.aspx</a>