

SCCM In-place migration of encrypted Disks

Note!: The following content was provided from my Colleagues Armin Denzler and Markus Stucki. Thanks these guys to share the knowledge of their work.

Warning, use at your own Risk!

The final WinPE boot image may contain a mechanism to format the partition(s) on the local drive before SCCM Task Sequence logic comes up. Thus, simply booting from this image could cause loss of data!

The following Solution may not be supported by Microsoft.

Contents

Problem.....	2
Solution	2
Technical Details	4
Step by Step	6

Problem

In-place migration of Systems with encrypted Disks is out-of-the box not possible with SCCM2007 Task-Sequences. This is because the Task-Sequence has to boot into WinPE to apply the new Operating System but WinPE will not be able to read the Task-Sequence Environment from the encrypted Disk.

Booting WinPE from DVD or USB is not a Problem, because WinPE will initially load the TS-Environment from the Boot-Media and does not touch the encrypted disk.

Now, the Question with an in-place migration scenario is: how to boot into WinPE from the encrypted Disk, but let WinPE assume that it was started from a standalone media...

Solution

The following chapters describe a solution to adopt this workflow by creating a highly customized WinPE boot image (WinPE 3.0, version 6.1.7600). This solution, which was found in teamwork by Armin Denzler and Markus Stucki, might not be supported by Microsoft :-).

The Solution requires two Task-Sequences with two different Boot-Images assigned to the target Computer.

Task-Sequence #1 is defined to run only on the installed OS (specify the platform requirements on the TS to run only on the specified platform like "All x86 Windows XP"). This TS can be used to backup local data to a Network-Share.

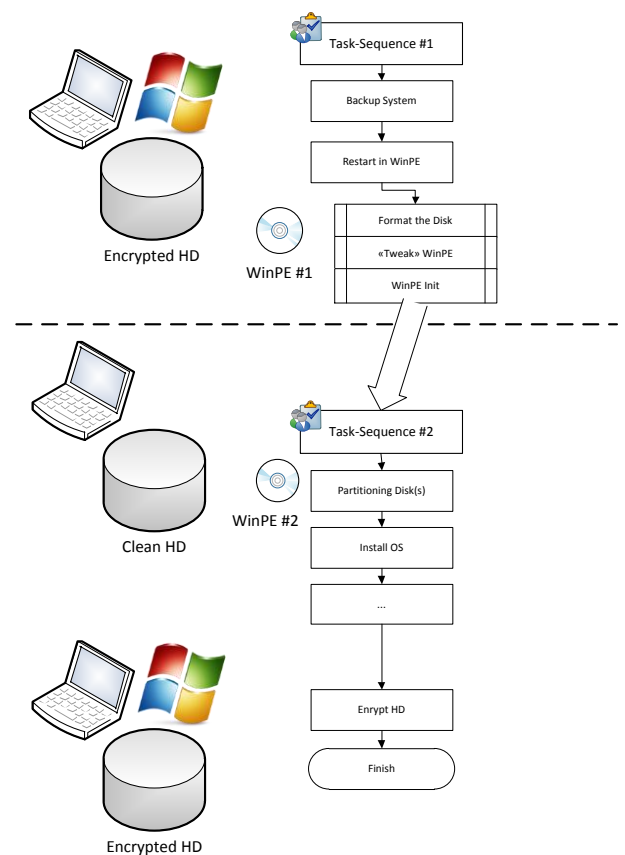
Task-Sequence #2 is defined to be "available to boot media and PXE" (Advertisement Option) and it must only run from WinPE (You can limit the platform requirements to an unused OS like "x64 Windows XP Professional SP1").

You may also have to define some conditions in the TS to prevent running the TS on the wrong OS.

TS#2 needs the standard Boot-Image that also is used with DVD/PXE boot.

TS#1 needs a modified Boot-Image that format the encrypted Disk (Dangerous!) and copy the TS Environment from WinPE#2 to the local empty disk.

As a Result, WinPE#1 does look for SCCM like WinPE#2 from a Boot-Media or PXE and the assigned TS#2 will start.



Note:

In general, it would also be possible to use this solution for just prestaging the modified boot image on target computers but preferably, SCCM R3 feature "Prestaged Media Provisioning" should be used.

If necessary, 802.1x Network Authentication may also be included, see

<http://myitforum.com/cs2/blogs/lakey81/archive/2011/07/06/configuring-802-1x-network-authentication-for-winpe-3-0-and-configmgr-deployments.aspx>.

Technical Details

The following difficulties are to be solved:

- A) Per design, SCCM Task Sequence Engine checks where WinPE is booted from (PXE / Removable Media / Local Disk) and sets variable %CONFIGPATH% accordingly.

If WinPE was booted from a local disk, SCCM expects that the WinPE boot was initiated by a previously started TS and tries to read current status from local disk.

Details:

- a. file ".\sms\bin\x64\TsBootShell.ini" refers to variable %CONFIGPATH%
- b. %CONFIGPATH% specifies where the TS environment is located (→ VARIABLES.DAT)
- c. if WinPE is booted from
 - CD/DVD:
SCCM sets %CONFIGPATH% to "<CD-drive-letter>:\"
SCCM automatically contacts the SCCM MP to check for assigned Task Sequences.
 - PXE:
SCCM sets %CONFIGPATH% to "X:\sms\data\"
SCCM automatically contacts the SCCM MP to check for assigned Task Sequences.
 - Local Disk,
SCCM sets %CONFIGPATH% to "<local-HD-drive-letter>:_SMSTaskSequence"
This implies that a TS is already running and SCCM TS Engine tries to read status from local disk

This means that if SCCM TS Engine shall "forget" that there is a TS in progress (even if WinPE was booted from local disk), the path to configuration data must be manipulated in TsBootShell.ini not to point to a location on local disk:

- If the disk is NOT encrypted, SCCM TS Engine can access the files and finds that a Task Sequence is in progress (the TS which was started in full OS) and continues the TS (even if reboot to WinPE is the last step, TS resumes just to finish)
- If the disk IS encrypted, SCCM TS Engine cannot access files on local disk and fails

The solution is to

- I) inject the TS environment information from an SCCM Bootable Media (complete content of folder .\SMS\DATA) into the boot image
- II) modify ".\sms\bin\x64\TsBootShell.ini" in the boot image to point to a fixed path in RAMDRIVE X: rather than using the variable (/configpath:X:\sms\DATA instead of /configpath:%CONFIGPATH%)

- B) When a boot image is added to SCCM (via console), it is modified and amongst other changes, the modified file ".\sms\bin\x64\TsBootShell.ini" is overwritten (SCCM defaults).

The solution is to

- I) Mount the boot image and copy the required information (modified TsBootShell.ini and content of folder .\SMS\DATA from SCCM Boot Media) to a custom folder
- II) Modify the WinPE startup process to copy the customized data from the custom folder to the according SCCM folders before the SCCM TS Engine is started:
 - Per WinPE / SCCM default, TsBootShell.exe is called by winpeshl.exe as defined in winpeshl.ini.
 - This process can be changed via modification of registry key
HKLM\SYSTEM\Setup\CmdLine

Instead of directly calling winpeshl.exe, a batch file is called which performs the necessary copy actions and calls winpeshl.exe afterwards.

For more information, see [http://technet.microsoft.com/en-us/library/dd744556\(WS.10\).aspx](http://technet.microsoft.com/en-us/library/dd744556(WS.10).aspx)

- C) When the SCCM TS Engine is started in WinPE (PE booted from local disk), it checks for a specific file in the root of the drive where boot.wim is located. File name is identical in each environment: "_SMSTSVolumeID.7159644d-f741-45d5-ab29-0ad8aa4771ca"
- If the disk is encrypted, SCCM Task Sequence is unable to read this file and fails with error "*Unable to read task sequence configuration disk*".

The solution is to

- I) Capture this file by starting a Task Sequence which just contains the instruction "Restart Computer" to "The boot image assigned to this task sequence" (with a long time out to reboot, this allows copying the file before the shutdown is initiated).
- II) Place this file in the custom folder in the boot image
- III) Format the local disk and copy the file to the root of the partition on local disk before the SCCM TS Engine is started

- D) When a TS starts in WinPE, it checks the ID of the boot image which is associated to the task sequence.

As the customized boot image contains the environment (VARIABLES.DAT) which was obtained from a default SCCM Boot Media, it "inherits" the "BootpackageID" (which is the ID of the boot image that was selected in the "Task Sequence Media Wizard").

This leads to a discrepancy between the Image ID which is registered in SCCM and the Image ID stored in the image itself (it can be considered it as a mismatch between "internal" and "external" ID).

This means that the customized boot image *cannot* be assigned to an OS task sequence because it would result in a reboot loop:

SCCM checks if the currently running WinPE matches the assigned boot image ID. As the *internal* ID does not match, TS Engine loads the boot image with the according *external* ID again.

The solution is to use two different task sequences:

- 1) First Task Sequence is started in full OS to perform some actions like backing up data to the network followed by the step "Restart Computer" (to "The boot image assigned to this task sequence").

- 2) Second Task Sequence is a mandatory OS Task Sequence (associated with a "normal" boot image).

When WinPE is started from the modified boot image on local disk, SCCM will behave like being booted from a DVD/USB boot media and contact the MP and start execution of the assigned OS Task Sequence.

Step by Step

The following steps show how to create a customized x64 WinPE boot image (WinPE 3.0, version 6.1.7600) which allows booting WinPE from local disk but nevertheless start a task sequence from network:

- 1) Create a Task Sequence Media (Bootable Media → CD/DVD Set)
- 2) Extract ISO generated in step 1, make a copy of folder <extracted-iso-folder>\SMS\DATA
- 3) mount the boot image which will be associated with the task sequence
E.g.:
`dism /mount-wim /wimfile:"C:\SCCM\BootMedia\boot-WinPE-from-HD.wim" /index:1 /mountdir:"C:\Temp\WimMountBootWim"`
- 4) In the "root" of the mounted WIM (parameter "MountDir" in dism command),
 - a. create a folder named "CustomBoot"
 - b. copy complete DATA folder from step 2) to "CustomBoot"
 - c. if necessary, edit file TSMBOOTSTRAP.INI to set "Unattended=TRUE"
 - d. in folder "CustomBoot" create file "TsBootShell.ini" with the following content:

```
[Shell]
OrgName=
EnableDebugShell=true
Run=X:\sms\bin\x64\TsmBootstrap.exe /env:WinPE /configpath:X:\sms\DATA
```
 - e. if disk encryption is in place:
 - i. Create a task sequence which contains the step "Restart Computer" to "The boot image assigned to this task sequence". Set a long timer for reboot, e.g. 600 seconds.
 - ii. Advertise the TS to a test machine
 - iii. When the boot image is staged on the local disk and the reboot countdown begins, file "_SMSTSVolumeID.7159644d-f741-45d5-ab29-0ad8aa4771ca" should exist in the root of %SystemDrive%.
 - iv. copy the file to a temporary network location and add it to folder "CustomBoot" in the mounted boot image
- 5) Modify registry
 - a. load hive SYSTEM from "<mountdir>\Windows\System32\Config\SYSTEM"):

```
[HKEY_LOCAL_MACHINE\_WinPE-Reg-SYSTEM\Setup]
"CmdLine"="CustomStart.cmd"
```
 - b. unload hive
- 6) Create the file "<mountdir>\windows\system32\CustomStart.cmd" with the following content:

```
copy /y x:\CustomBoot\TsBootShell.ini x:\sms\bin\x64\*. *
xcopy /e /i /y x:\CustomBoot\DATA x:\sms\DATA
winpeshl.exe
```

If the disk is encrypted, a few more commands need to be added to CustomStart.cmd.

Additionally required actions are

- format the partition(s)
- copy file _SMSTSVolumeID... to the root of the drive where WinPE was bootet from (SCCM TS Engine expects this file to be located there), for example drive "C:"

```
copy /y x:\CustomBoot\TsBootShell.ini x:\sms\bin\x64\*. *
xcopy /e /i /y x:\CustomBoot\DATA x:\sms\DATA
format c: /q /v:tmpname /y
copy x:\CustomBoot\_SMSTSVolumeID.7159644d-f741-45d5-ab29-0ad8aa4771ca c:\
winpeshl.exe
```

- 7) unmount the WIM and use dism parameter "/commit", add boot image to SCCM and/or update Distribution Points
- 8) Associate the customized WIM image with a task sequence which is started in full OS, then create an advertisement for this TS.
- 9) Create a 2nd task sequence for OS deployment and create a mandatory advertisement.