adaptiva

Adaptiva OneSite OSD User Guide

Version 4.0

Updated 810

adaptiva

Table of Contents

adaptiva

Overview

Adaptiva OneSite provides administrators the ability to enhance their ConfigMgr OSD process with multiple tools and options. This document describes how to enable and configure specific features which can be used during the OSD process. The features and tools described in this document are:

- OneSiteDownloader
- Peer-to-Peer PXE (P2P PXE)
- Virtual State Migration Points (vSMP)

Basic OSD Functionality

To use Adaptiva OneSite as an Alternate Content Provider (ACP) during the ConfigMgr OSD process, the following steps must be completed.

- Boot Image Customization: Describes how to prepare boot images with OneSiteDownloader, which is a utility used to invoke the OneSite Alternate Content Provider during WinPE.
- Task Sequence Modification: Describes how to customize a ConfigMgr task sequence to invoke OneSiteDownloader during WinPE and to install the Adaptiva Client.
- Task Sequence Content Push: Describes the process of pre-staging content, referenced in an OSD task sequence, to remote offices so they can be used during a task sequence. This includes OS and boot image files, drivers, software updates and packages.

Advanced OSD Functionality

- Adaptiva OneSite includes additional features to allow for bare metal deployments, and user state migration between machines during a task sequence.
- P2P PXE Configuration: Describes how to enable P2P PXE in OneSite which allows for Adaptiva clients to function as a PXE responder without configuring a ConfigMgr Distribution Point for PXE, or setting up IP helpers or DHCP scope options.
- vSMP Configuration: Describes how to enable the vSMP feature in OneSite which allows for Adaptiva clients to function as State Migration Points during a user state capture and restore during a task sequence.

adaptiva

Customizing Boot Images for OneSiteDownloader and P2P PXE

In order to use Adaptiva OneSiteDownloader, during the OSD process, a boot image must be customized for OneSite and added to ConfigMgr.

> *NOTE: Adaptiva provides a PowerShell script that automates this process for you. The readme and download link can be found here.*

This section explains the steps to create, customize and deploy boot images which will be used during OSD.

> *NOTE: By default, the ConfigMgr installation creates a boot image for x86 and x64 architectures. If both boot images are required, repeat the process below for both architectures. The process described below references a 64-bit boot image.*

Creating a New Custom Boot Image

Prior to creating a custom boot image, a copy of the default boot image of the desired architecture should be made and distributed.

> *NOTE: If you want to update your boot image to the latest ADK, please do so before continuing as any update required later would require these steps to be repeated.*

1. On the ConfigMgr server, create a new folder to store the custom boot image. The following path will be used as an example in this document:

```
D:\PkgSource\OSD\OneSiteBootx64
```

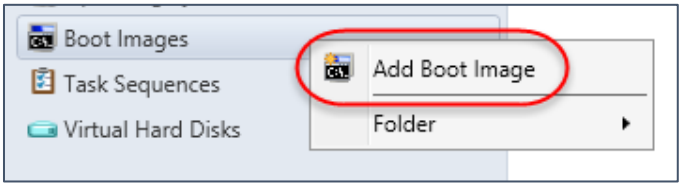
> *NOTE: This folder will be the package source for the boot image, so it should be created in a static location. It must be shared so that you can enter the URL later. Perhaps use the location where your other package source is stored.*

2. To find the default ConfigMgr boot images, use Windows Explorer to navigate to the location that ConfigMgr is installed and open the sub folder: **\OSD\Boot\<architecture>**

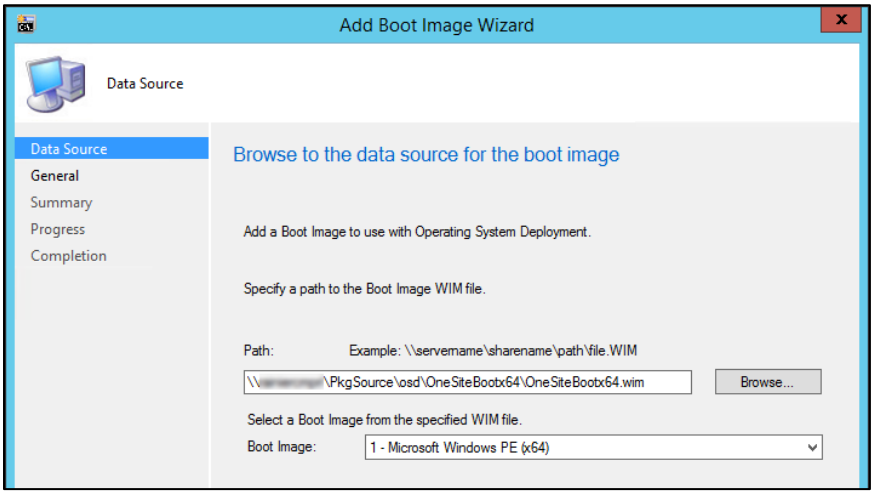
   For x86: *C:\Program Files\Microsoft Configuration Manager\OSD\Boot\i386*

   For x64: *C:\Program Files\Microsoft Configuration Manager\OSD\Boot\x64*

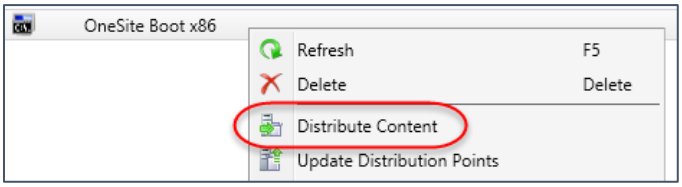
3. Copy only the **boot.wim** file to the folder created in Step 1.
4. Rename the boot.wim file to something more descriptive, such as: **OneSiteBootx64.wim**.
5. In the ConfigMgr console navigate to the **Software Library workspace / Operating Systems** then right-click **Boot Images**. In the context menu, select **Add Boot Image**.



6. In the "Add Boot Image Wizard", in the "Data Source" screen, browse or enter the UNC path to the new boot image file. Click **Next** to continue.



7. At the "General" screen, enter a readable, friendly **Name** that will appear in the console for this custom boot image (i.e. OneSite Boot (x64)), and then click **Next**.
8. Continue the wizard until complete. The new boot image should be listed in the console.
9. Right-click the new boot image, and in the context menu, select **Distribute Content**.

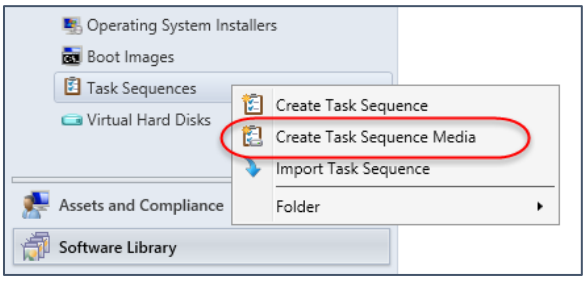


10. Complete the "Distribute Content Wizard" to send the boot image package to at least one DP.

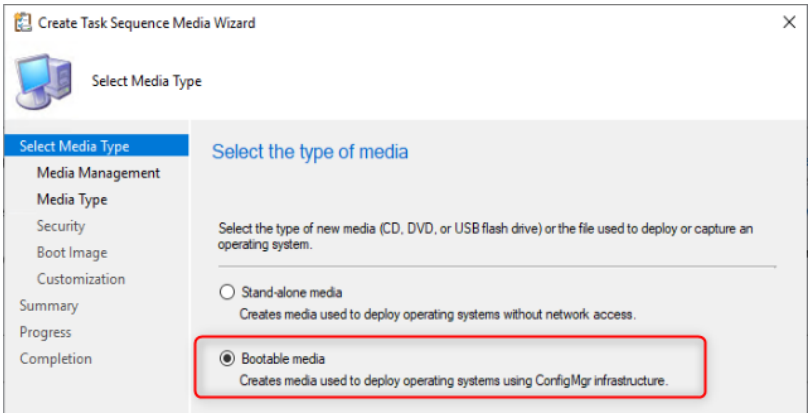
Creating a Scratch ISO File

Several files will need to be added to the boot image. These will come from the media for the task sequence. To do this, a temporary ISO file should be compiled and mounted.

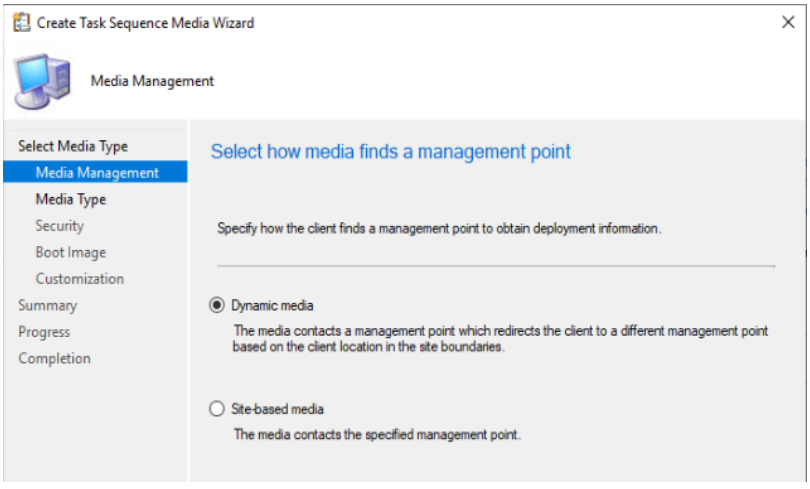
1. In the ConfigMgr console, navigate to the **Software Library workspace / Operating Systems** and then right-click Task Sequences. In the context menu, select **Create Task Sequence Media**.



2. In the Create Task Sequence Media Wizard, at the "Select Media Type" screen, select **Bootable media** and then click **Next**.



3. At the "Media Management" screen, choose the best option for your environment.

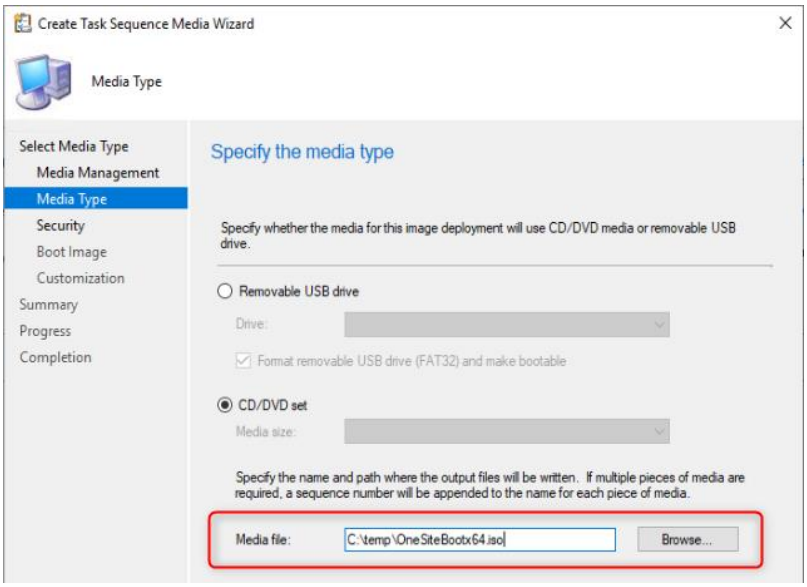


- Select **Dynamic media** if you want to allow a management point to redirect the media to another management point, based on the client location in the site boundaries. This is the recommended setting for a multi-site hierarchy.
- Select **Site-based** media if you want the media to contact only the specified management point. This is recommended for a single Primary Site hierarchy.

By default, Dynamic media is selected. Click **Next** to continue.

4. At the "Media Type" screen, **CD/DVD set** should be selected. In the Media file text box, **enter the path and file name** of the temporary ISO file that you will create. Click **Next** to continue.

> *NOTE: This is a temporary file and does not need to be shared. You can put it anywhere with sufficient space. It should be about 400MB.*



5. At the "Select security settings for the media" screen, choose the best options for your environment.
   - **Enable Unknown Computer Support –** Select this option if you want to be able to PXE boot and build unknown computers that are not yet registered in SCCM.
   - **Specify a password to protect task sequence media…** – If you require a Password to be entered after the boot image is loaded then enter this here. As Adaptiva does not use a DP, this replaces the **Require a password when computers use PXE** setting you would usually set on the PXE tab options under Distribution Point properties.

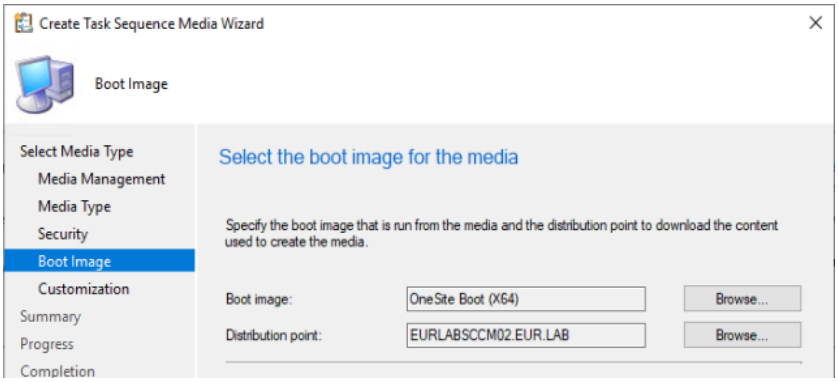
> *NOTE: Protect media with a password is checked by default, either uncheck or supply a password to continue.*

- **Create a self-signed media certificate… -** If ConfigMgr is configured to use HTTPS for MPs and DPs, you should import your PKI certificate at this screen. If not, select the default option of **Create self-signed media certificate.**

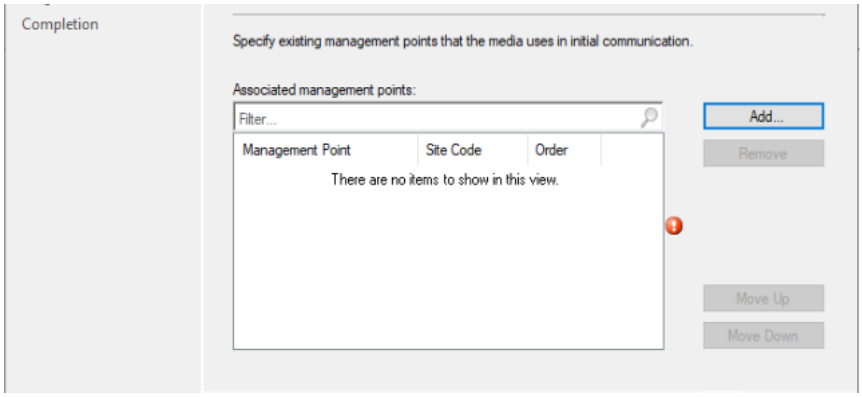
> *NOTE: We recommend you set the expiration date to some distance in the future, for example over 2 years, otherwise the boot image will stop working and this process will need to be repeated once the expiration date has passed.*

Click **Next** to continue.

6. At the "Select the boot image for the media" screen, at the "Boot image" section, click **Browse** and select the new OneSite boot image. In the "Distribution point" section, click **Browse** and select the DP to download the boot image from for the ISO image creation.

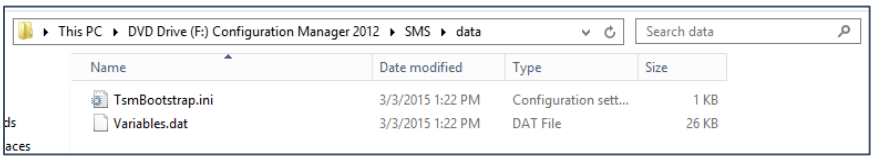


- If you selected **Dynamic Media** in step 3, the "Associated management points" section will be shown. Select **Add** to choose one or more Management Points that the media will use in its initial communication.



- if using **Site-based media** in step 3, select **Browse** to choose the Site the media will use. Click **Next** and continue the wizard until it is complete.

7. Click **Next** and continue the wizard until it completes.
8. When done, navigate to the folder that you specified for the resulting ISO file.
9. To capture the configuration information of the boot image just created, two files must be extracted from the ISO. For newer Windows operating systems starting with Windows 8 and Server 2012, you can right-click the ISO and **Mount** the ISO as a drive letter. For older versions of Windows, use a free utility such as 7-Zip to extract the ISO to a folder.
10. Once mounted or extracted, navigate to the **SMS\Data** folder and copy out the following files:
    - TsmBootstrap.ini
    - Variables.dat

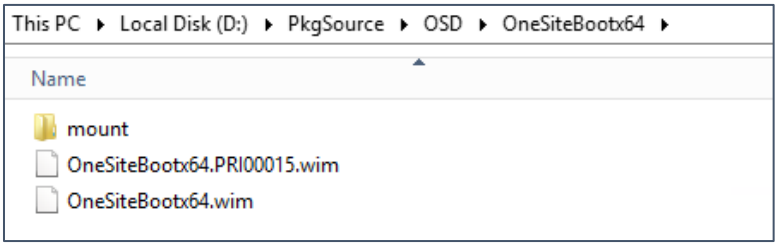


11. The above files will be referenced later in this document. Once complete, the ISO file is not needed anymore and can be unmounted and deleted.

Mounting and Finalizing the Boot Image

Now that we have these files, we can add them and the OneSiteDownloader.exe to the boot image. To do this, perform the following steps.

1. In the folder that you placed the custom boot image, create a sub-folder named **Mount**.



2. Open an administrative command prompt and run the following DISM command to mount the boot image file to the Mount folder created earlier.

   For Server 2012-2016 / Windows 8-10:

```
DISM.exe /Mount-Image
/ImageFile:D:\PkgSource\OSD\OneSiteBootx64\OneSiteBootx64.wim /Index:1
/MountDir:D:\PkgSource\OSD\OneSiteBootx64\Mount
```

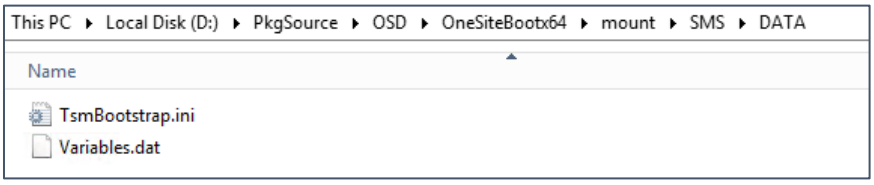
For Server 2008 / Windows 7:

```
DISM.exe /Mount-Wim
/WimFile:D:\PkgSource\OSD\OneSiteBootx64\OneSiteBootx64.wim /Index:1
/MountDir:D:\PkgSource\OSD\OneSiteBootx64\Mount
```

3. Once complete, navigate to the Mount folder and create the following folder structure from the root of the Mount directory.

   .\SMS\DATA

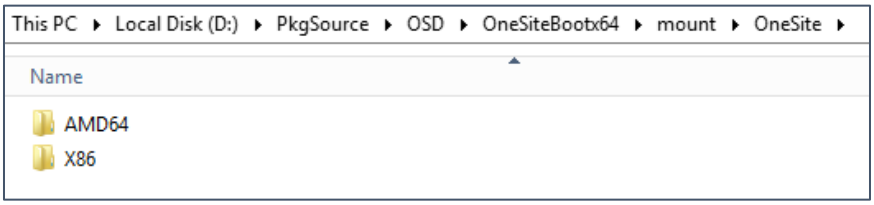
   Copy the **TsmBootstrap.ini** and **Variables.dat** files, that you extracted from the ISO, to the **DATA** folder.



4. The path to OneSiteDownloader is referenced in a task sequence variable. Since there are two versions, 32-bit and 64-bit, In the case where the task sequence variable is defined at the collection level and not in the task sequence itself, the variable %processor_architecture% can be used to define the location of OneSite Downloader. At the root of the mount directory, create the following folder structure:

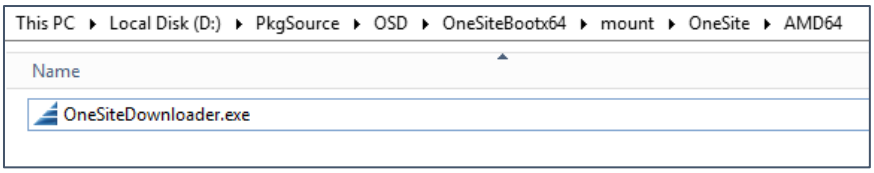
   .\OneSite\AMD64

   .\OneSite\X86



5. If customizing the 32-bit version, copy **OneSiteDownloader.exe** to the X86 folder created above. If customizing the 64-bit version, copy **OneSiteDownloader64.exe** to the AMD64 folder, and then rename it to **OneSiteDownloader.exe**.

> *NOTE: OneSiteDownloader64.exe must be renamed to OneSiteDownloader.exe so it can be referenced in the task sequence variable.*



6. Copy any other tools and utilities that you want available in your boot image.
7. Once all files are copied to the mount directory close Windows Explorer and make sure there are no open files or folders in the Mount folder.
8. Use the DISM command to unmount and commit the contents of the mount folder into the original WIM.

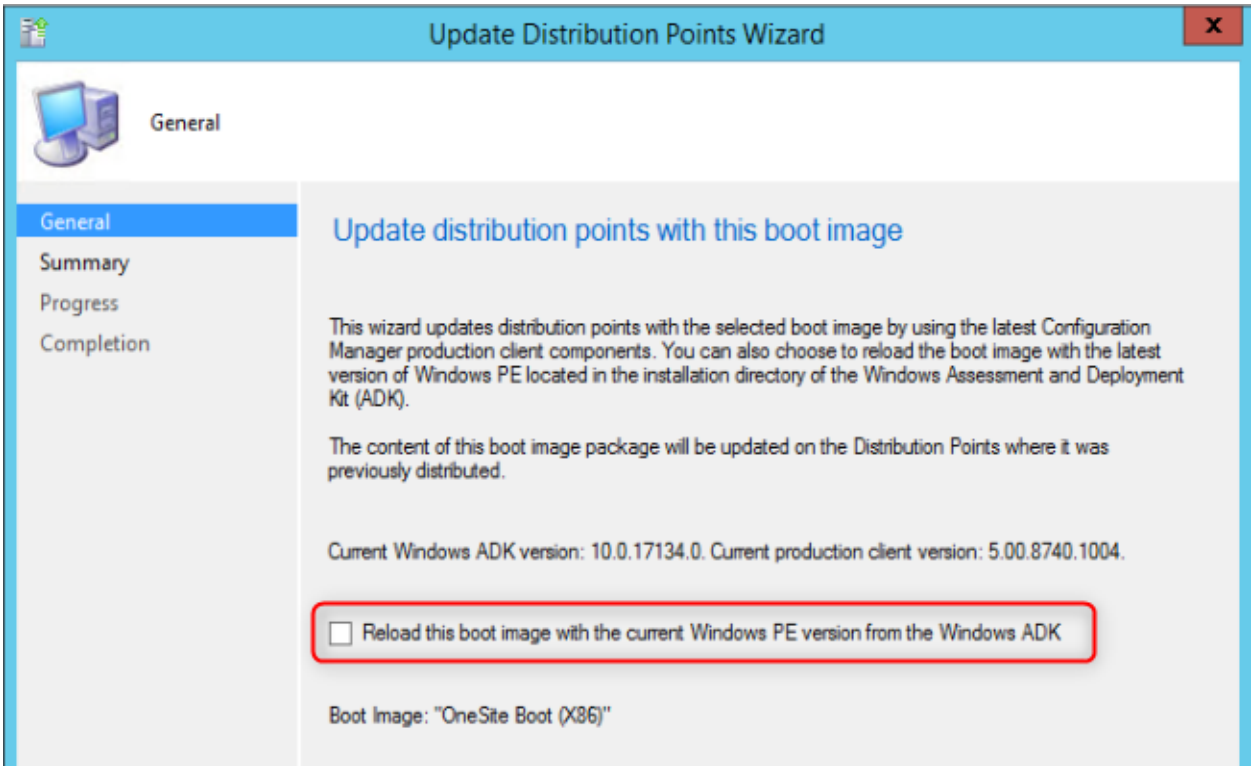
   For Server 2012-2016 / Windows 8-10:

```
DISM.exe /Unmount-Image /MountDir:D:\PkgSource\OSD\OneSiteBootx64\Mount
/commit
```

   For Server 2008 / Windows 7:

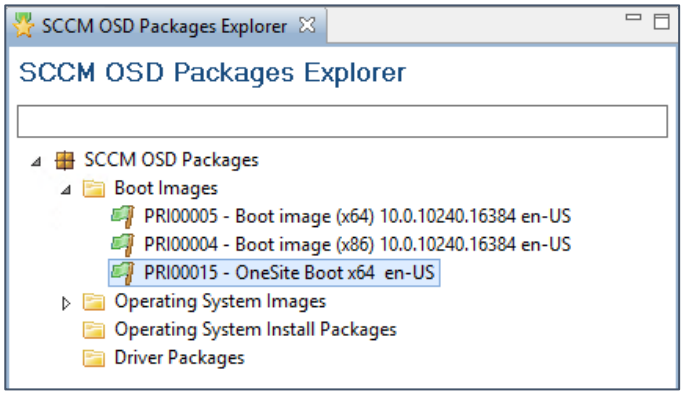
```
DISM.exe /Unmount-Wim /MountDir:D:\PkgSource\OSD\OneSiteBootx64\Mount
/commit
```

9. Now that the OneSiteBootx64.wim has been updated, the mount folder can be deleted.
10. In the ConfigMgr console, right-click the boot image and select **Update Distribution Points**
11. On the initial **General** page of **the Update Distribution Points Wizard**, make sure you **DO NOT** tick the option to "Reload this boot image…";



If you tick this option, the custom files for the TS Boot Media and the OneSiteDownloader.exe you added to the boot image will be lost.

12. Complete the wizard to update the DP(s).
13. If automatic publication is not enabled in OneSite, the new boot image needs to be published in Adaptiva. In the Adaptiva Workbench, open the **OneSite – Package Perspective**. In the **SCCM OSD Packages Explorer** view, verify that the new OneSite Boot x64 image is published (green flag). If not, right-click the image, and select **Publish as adaptiva content**.



Conclusion

The boot image is now ready for use. When creating/modifying your OSD task sequence, select the boot image you just created.

Task Sequence Modifications for OneSiteDownloader

> *NOTE: Integrating OneSiteDownloader.exe into a task sequence varies between ConfigMgr 2007 and ConfigMgr 2012/Current Branch so this guidance will cover both independently. When using OneSiteDownloader in ConfigMgr 2012/Current Branch, OneSiteDownloader is executed based on a task sequence variable. When using OneSiteDownloader in ConfigMgr 2007, OneSiteDownloader is executed via a command line.*

ConfigMgr 2012/Current Branch Integration

In order to use OneSite as an Alternate Content Provider (ACP) in the WinPE phase of a ConfigMgr 2012/Current Branch task sequence, the task sequence uses a task sequence variable to identify the location of OneSiteDownloader. The name of the variable is **SMSTSDownloadProgram**. There are two ways to implement the variable. One option is to set the variable within the task sequence itself using the "Set Task Sequence Variable" task. Another option is to set the variable as a collection variable against the collection which contains the machines the task sequence will be targeted to.

There are also other TS variables that can be added to configure certain settings for OneSite to use (see Appendix A)

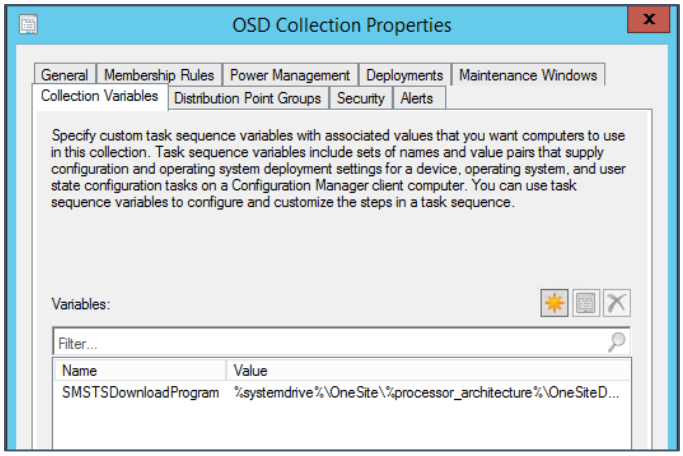
Setting the Task Sequence Variables as a Collection Variable

To set a collection variable, right-click the collection and select **Properties**. Select the **Collection Variables** tab and add a new variable:

The **SMSTSDownloadProgram** must be set;

Name: SMSTSDownloadProgram

Value: %systemdrive%\OneSite\%processor_architecture%\OneSiteDownloader.exe



It is also recommended to use the **OneSiteServerNameOrIP** variable so the OneSiteDownloader can communicate to the Adaptiva server to assist for P2P discovery.

Name: OneSiteServerNameOrIP

Value: <*Adaptiva Server FQDN*> OR <*Adaptiva Server IP*>

Task Sequence Modifications

> *NOTE: Step 1 below can be skipped if a collection variable was used as described above.*

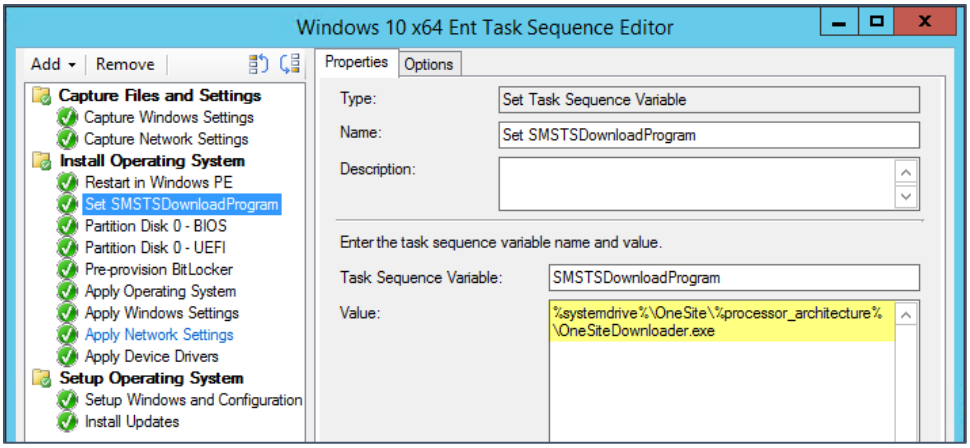
1. Add OneSite TS Variables

   Edit the task sequence you will be using with OneSite. After the first "Restart in Windows PE" task, or for any restart, add the task:

   Set Task Sequence Variable:

   Variable: SMSTSDownloadProgram

   Value: The location of OneSiteDownloader relative to the root (X:) of your boot image. For example, in the boot image created earlier, the location would be: **%systemdrive%\OneSite\%processor_architecture%\OneSiteDownloader.exe**



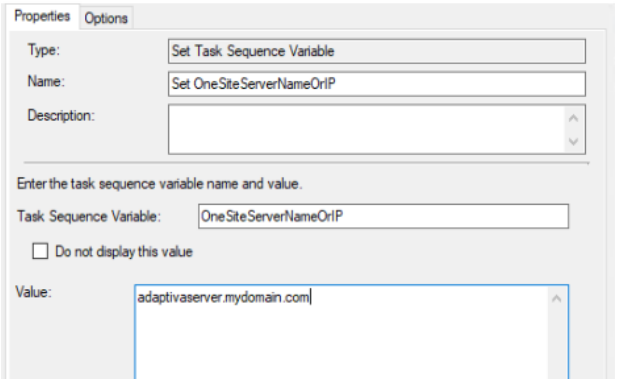
Warning: This variable must be set at the beginning of the task sequence and after every reboot until the Adaptiva Client is installed.

It is also recommended to add the **OneSiteServerNameOrIP** variable so the OneSiteDownloader can communicate to the Adaptiva server to assist for P2P discovery.

Set Task Sequence Variable:

Variable: OneSiteServerNameOrIP

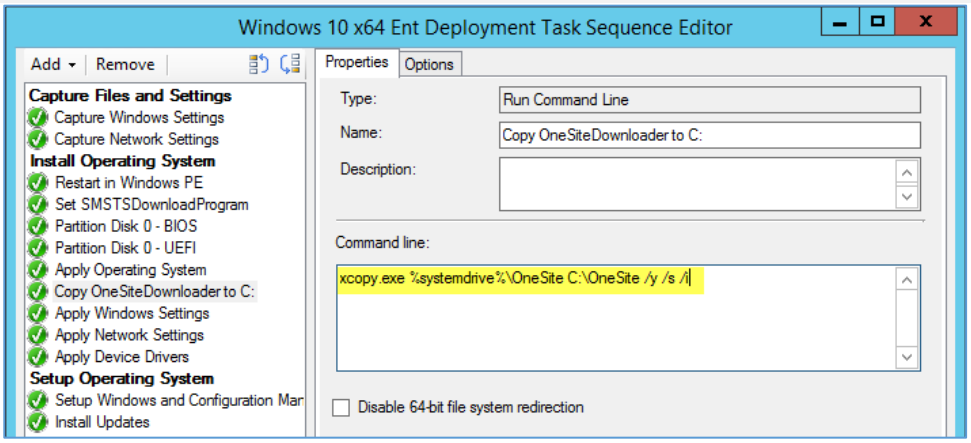
Value: The FQDN or IP address of the Adaptiva server:



2. OneSiteDownloader can continue to be used in task sequences after the WinPE section is completed and the build machine has been booted into Windows. Because the WinPE X: drive will no longer be available after booting into Windows, a task sequence step must be included to copy the OneSiteDownloader utility to the local C: drive. This task should be added after the "Apply Operating System" task and before the "Setup Windows and ConfigMgr" task.

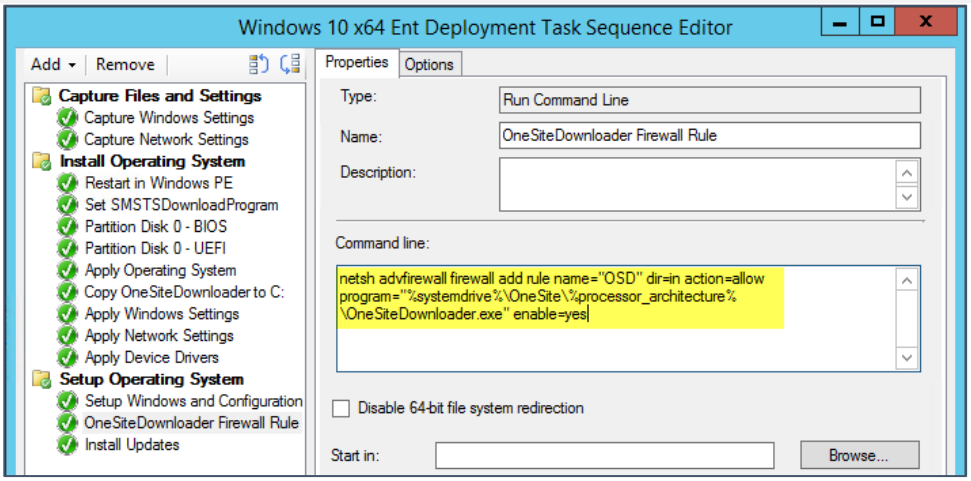
   Add the **Run Command Line** task after the Apply Operating System task with the command line:

```
xcopy.exe %systemdrive%\OneSite C:\OneSite /y /s /i
```



3. Once in the Windows environment, the Windows firewall may block OneSiteDownloader communication. After the "Setup Windows and ConfigMgr" task, a **Run Command Line** task should be added to open the Windows Firewall to OneSiteDownloader.

```
netsh advfirewall firewall add rule name="OSD" dir=in action=allow
program="%systemdrive%\OneSite\%processor_architecture%\OneSiteDownload
er.exe" enable=yes
```



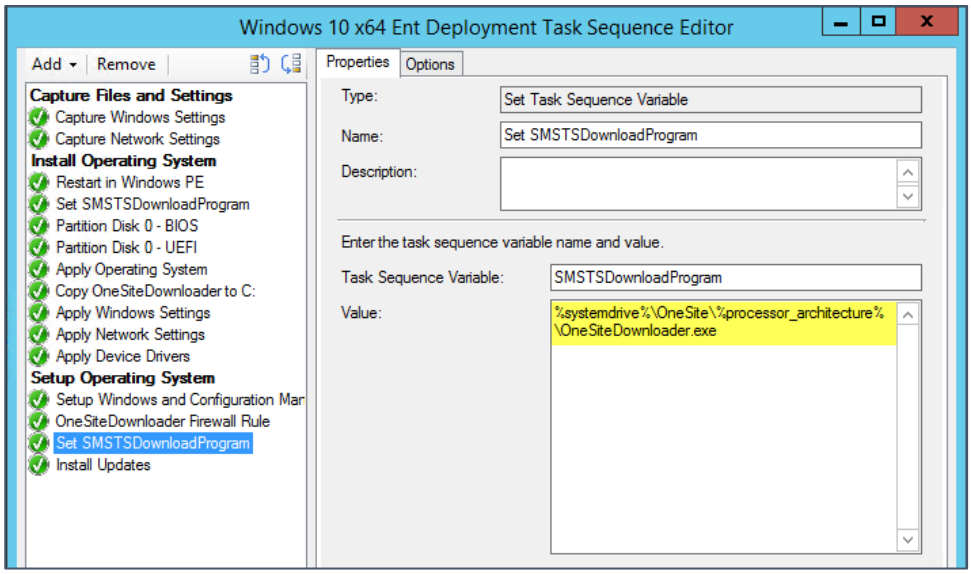
> ***Warning: In some versions of WinPE, this command may not be supported, so just in case, in the task Options tab, check the box "Continue on error".***

4. If a collection variable was used, this step can be skipped. Since OneSiteDownloader was copied to the C: drive, the SMSTSDownloadProgram must be set again to point to the new location.

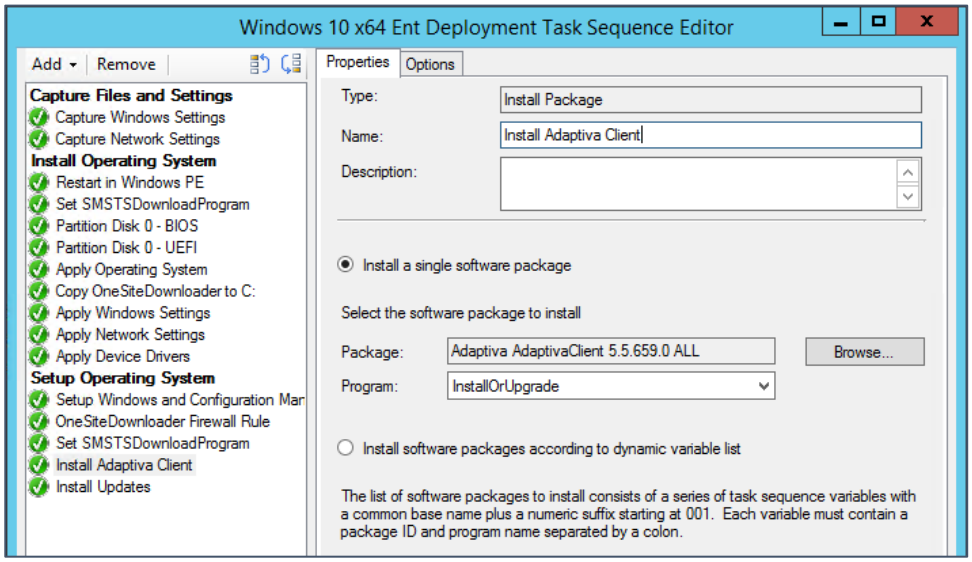
   Add the task: **Set Task Sequence Variable**

Variable: **SMSTSDownloadProgram**

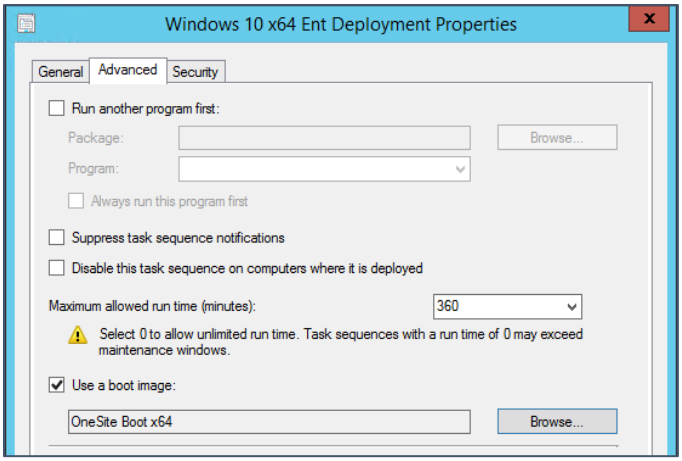
Value: **%systemdrive%\OneSite\%processor_architecture%\OneSiteDownloader.exe**



5. OneSiteDownloader can't be used to download CI based content such as Software Updates or Applications, so before the Install Updates or any Install Application tasks, add an **Install Package** task to install the Adaptiva client.



6. Customize the task sequence further as needed.
7. Prior to deploying the task sequence, open the task sequence **Properties**, select the **Advanced** tab, then click the **Browse** button to select the **OneSite Boot image** created in the previous section.



At this point the Task Sequence is ready to be added to a content push policy. For additional options for OneSiteDownloader and other variables which can be used, see

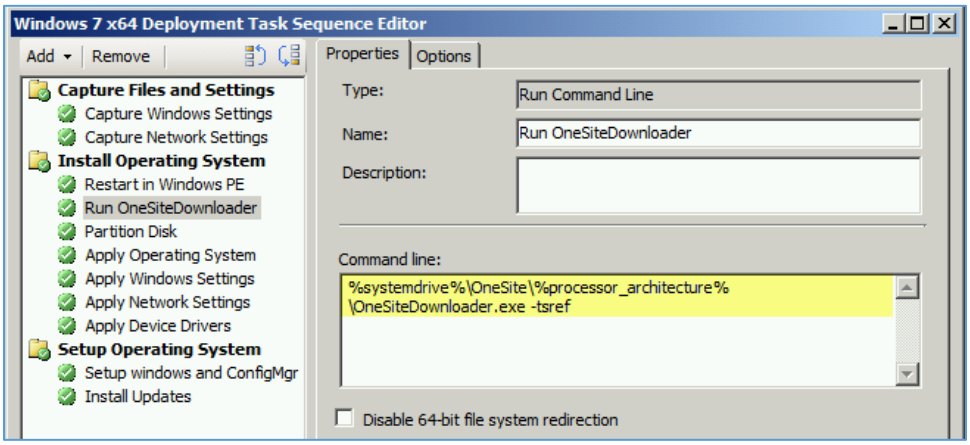
Appendix A: ConfigMgr 2012/Current Branch OneSiteDownloader Task Sequence Variables.

ConfigMgr 2007 Integration

OneSiteDownloader is also compatible with ConfigMgr 2007, but instead of using task sequence variables, OneSiteDownloader is called via command line. The following steps describes the process of using OneSiteDownloader in a ConfigMgr 2007 task sequence:

1. Edit the task sequence you will be using with OneSite. After the first "Restart in WindowsPE" task, add the **Run from Command Line** task with a command line to execute OneSiteDownlaoder with the "-tsref" parameter (and additional parameters).

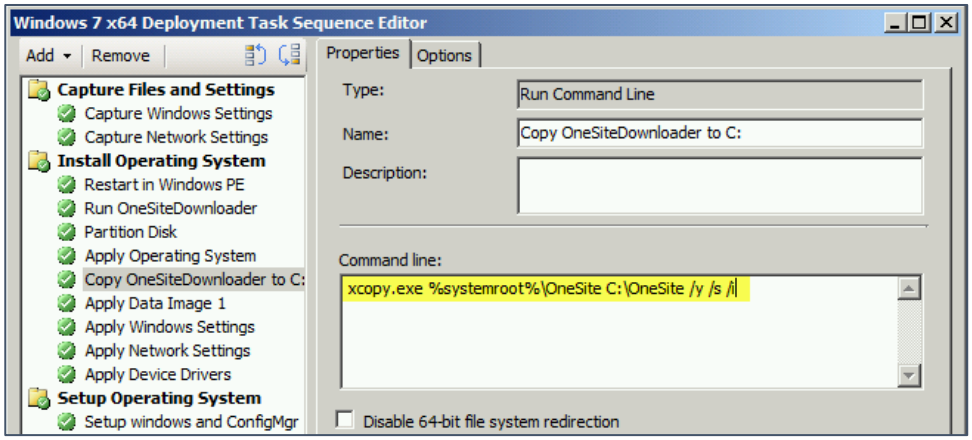
```
%systemdrive%\OneSite\%processor_architecture%\OneSiteDownloader.exe -tsref
```



2. OneSiteDownloader can continue to be used in task sequences after the WinPE section is completed and the build machine has been booted into Windows. Because the WinPE X: drive will no longer be available after booting into Windows, a task sequence step must be included to copy the OneSiteDownloader utility to the local C driver. This task should be added after the "Apply Operating System" task and before the "Setup Windows and ConfigMgr" task.

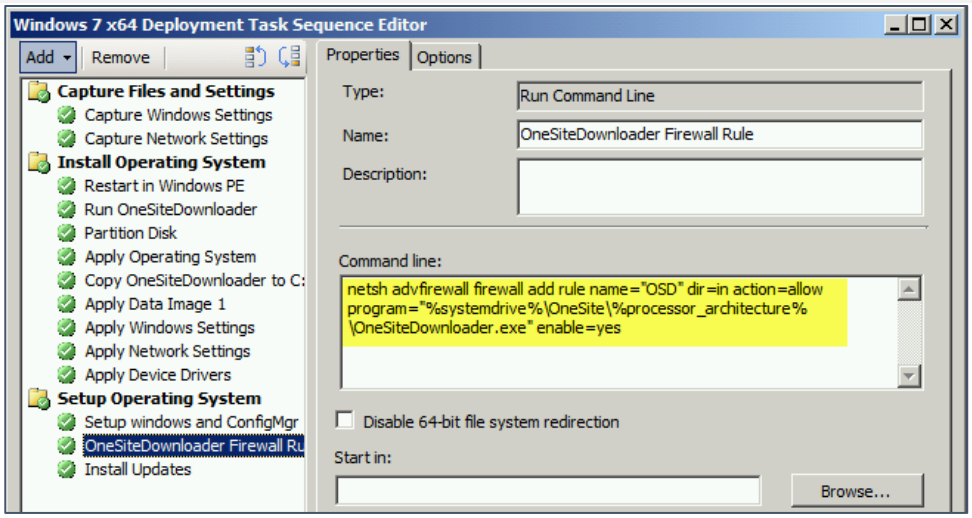
   Add the **Run Command Line** task after the Apply Operating System task with the command line:

```
xcopy.exe %systemroot%\OneSite C:\OneSite /y /s /i
```



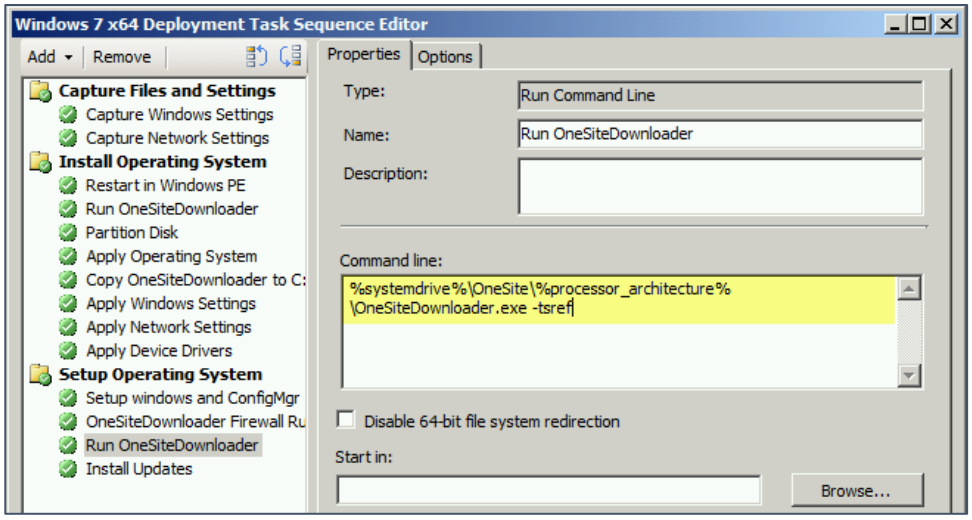
3. Once in the Windows environment, the Windows firewall may block OneSiteDownloader communication. After the Setup Windows and ConfigMgr task, a **Run Command Line** task should be added to open the Windows Firewall to OneSiteDownloader:

```
netsh advfirewall firewall add rule name="OSD" dir=in action=allow
program="%systemdrive%\OneSite\%processor_architecture%\OneSiteDownload
er.exe" enable=yes
```

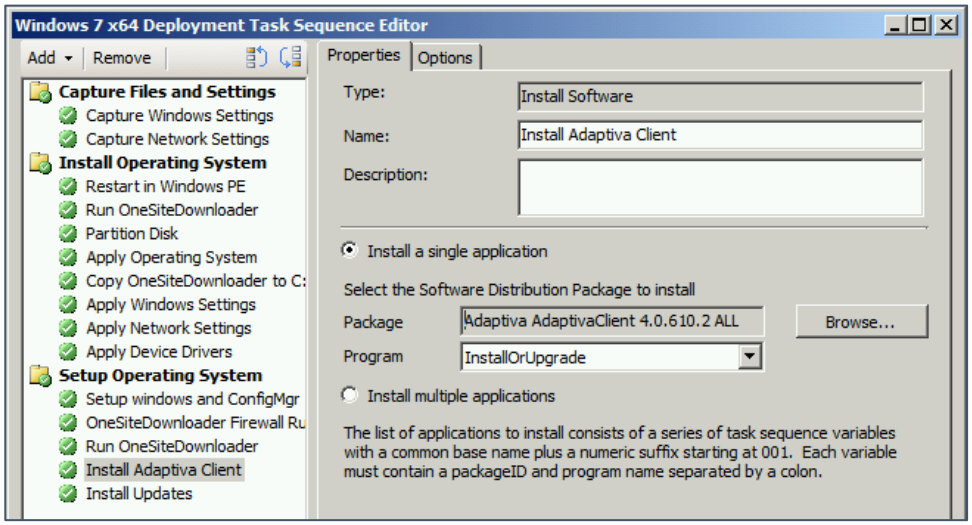


4. In order to use OneSiteDownloader after the system reboots, the OneSiteDownloader command must be run again. Add a Run from Command Line with the command line to execute OneSiteDownloader from C:\ plus the "-tsref" parameter (and additional parameters).

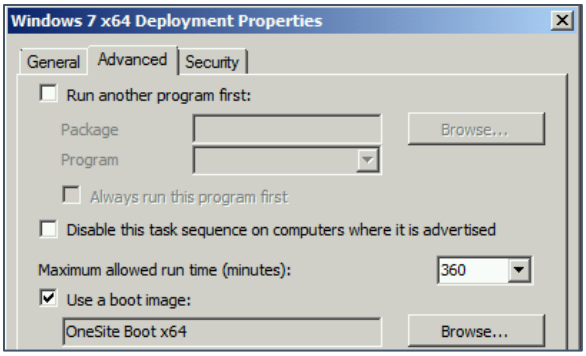
```
%systemdrive%\OneSite\%processor_architecture%\OneSiteDownloader.exe -
tsref
```



5. An additional task should be added to install the Adaptiva client. Add the Install Software task to run the Adaptiva Client package. Once this is added, OneSiteDownloader won't need to be called again. The Adaptiva Client should be installed before the "Install Software Updates" task.



6. Prior to deploying the task sequence, open the task sequence **Properties**, select the **Advanced** tab, then click the **Browse** button to select the **OneSite Boot image** created in the previous section.



At this point the Task Sequence is ready to be added to a content push policy. For additional command line options for OneSiteDownloader, see

Appendix B: ConfigMgr 2007 OneSiteDownloader Command Line Options at the bottom of this document.

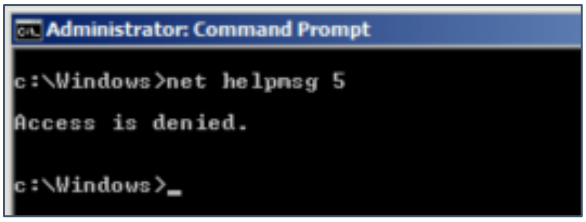
Logging and Exit Codes

Whenever the OneSiteDownloader utility is executed, it creates a log file named **OneSiteDownloader.LOG** in the same folder where it was run. This also includes on the X: drive.

Whenever the OneSiteDownloader tool is executed, it returns an exit code of 0 in case of success, and a non-zero WIN32 error code in case of failure.

The cause of failure may be determined by:

- Examining the OnesiteDownloader.LOG file
- Decoding the exit code using the Windows "net helpmsg" command. For example, if a code 5 is returned, you can run the following command in a command prompt for more information: "net helpmsg 5".

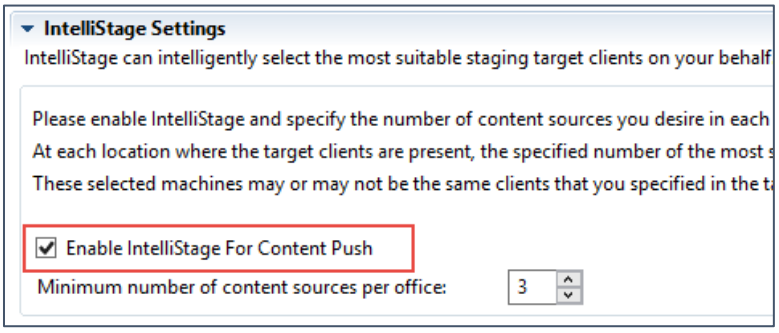


Task Sequence Content Push

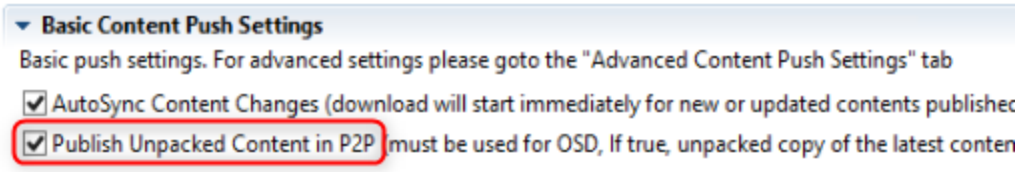
During the execution of a task sequence, OneSiteDownloader and the Adaptiva Client attempt to retrieve content from local peers on the same subnet. If the content isn't available at the site, a WAN download will be attempted. This is not an ideal scenario as files used during the imaging process tend to be large which will delay the completion of the task sequence. To avoid this issue, it is highly recommended that a Content Push policy be created and deployed to one or more machines at the local office in advance so that content is available locally prior to the task sequence execution.

Creating a Content Push Policy for a Task Sequence

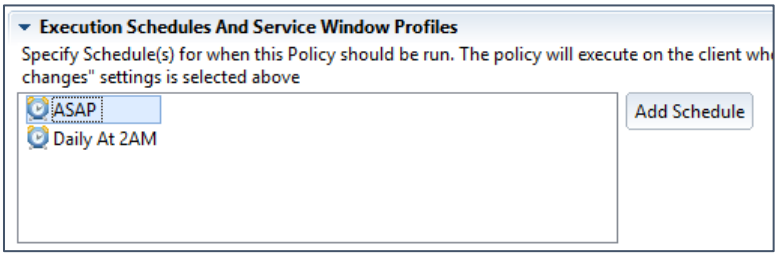
1. Prior to creating the Content Push policy, create a collection in SCCM, which includes one or more clients that are located at the office(s) where you plan to deploy the task sequence. These clients should be up and running so they are able to run the policy, but they are not required to be up during the task sequence execution.
2. In the Adaptive Workbench, expand the **OneSite** folder, then execute the **OneSite – Content Push Perspective**.
3. In the "Content Push Policy Explorer" pane, click the green + button to create a new content push policy.
4. In the "Content Push Policy Settings" editor, enter a Name and Description (optional) for the policy. Ex: **Windows 7 x64 Task Sequence Content**
5. In the "IntelliStage Settings" section, check the box**: Enable IntelliStage for Content Push**. This setting instructs the policy to make multiple copies of the content at the local office. It is recommended to choose 3 or greater.



6. In the "Basic Content Push Settings" section, check the box: **Publish Unpacked Content in P2P (must be used for OSD)**. This will unpack the content on the clients that host it, so it is accessible to the OneSiteDownloader which is only a partial client.

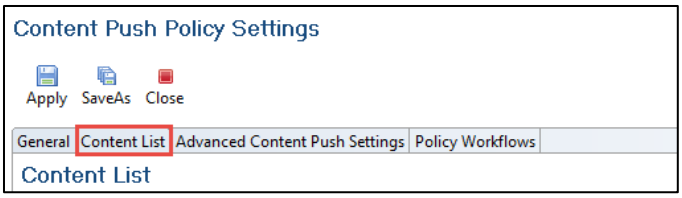


7. In the "Execution Schedules and Service Windows Profiles" section, click **Add Schedule**, and choose a schedule to deploy the policy.

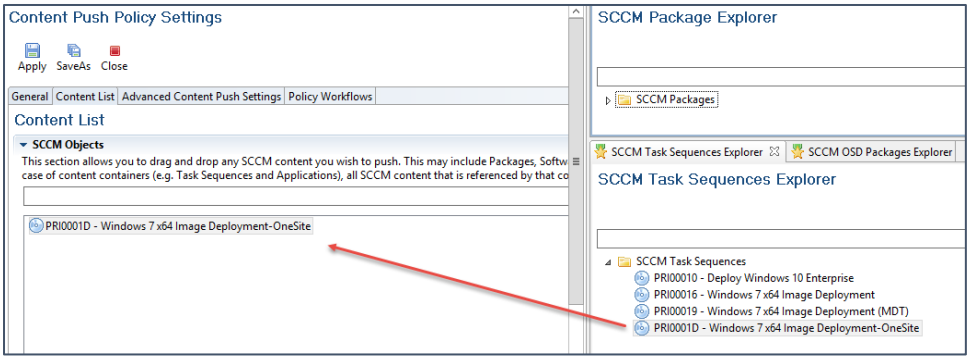


> ***Warning: If the "ASAP" schedule is the only schedule selected, then clients which are to be targeted by this policy will run the policy only once. In the case where you want to guarantee that the content will be available and enforced in the targeted offices, a recuring schedule should be specified.***

8. In the "Target Collections" section, click **Add Collection** and select the SCCM collection created in Step 1.
9. Scroll to the top of the editor and, select the "Content List" tab.



10. In the "SCCM Objects" section, **drag and drop** the specific task sequence to be deployed from the "SCCM Task Sequences Explorer" view into the SCCM Objects box. In the "Referenced Content" section below, you will see all the content which will be included in the content push policy.

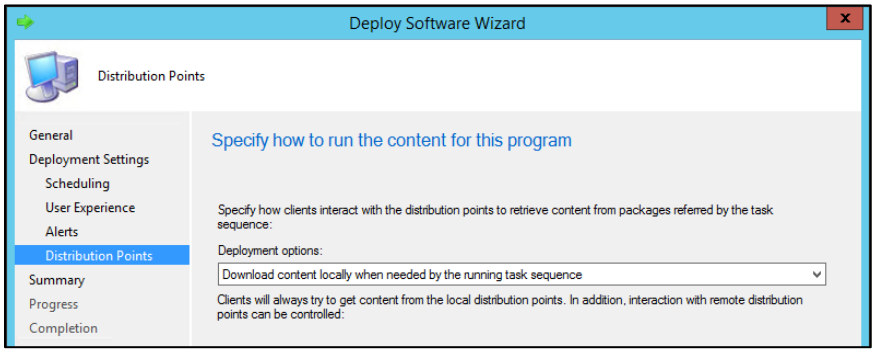


> *NOTE: It is important to add additional content items which may be called, but not directly referenced in the task sequence such as driver packages, software updates, or packages which may be dynamically requested using an MDT integrated task sequence.*

11. To deploy the Content Push Policy, click the **Apply** button. Once the defined schedule is applicable, clients in the collection will download the content referenced in the task sequence and replicate it to their peers.

Task Sequence Deployment

Once the task sequence content has been distributed and unpacked by the content push policy you can deploy the task sequence in SCCM. It is important to remember to set the deployment to **Download content locally when needed by the running task sequence**.



> ***Warning: If the task sequence is set to run from the distribution point, OneSiteDownloader and the Adaptiva Agent will not be called.***

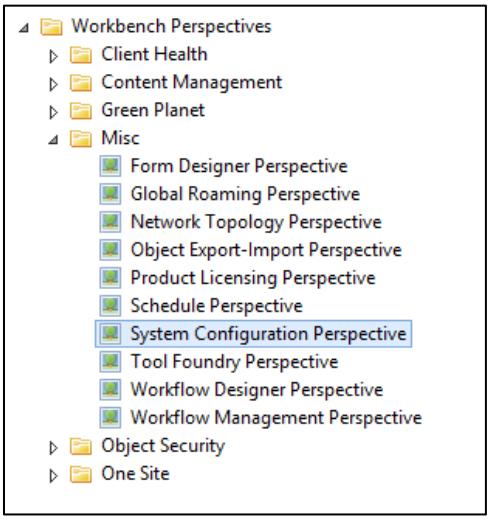
OneSiteDownloader and OSD Content Push Shares

When a content push policy is configured for OSD and the setting **Publish Unpacked Content in P2P** is selected, each piece of content referenced in the policy is unpacked into a folder in the target client's Adaptiva Cache folder. By default, each folder is shared so that OneSiteDownloader can access the unpacked content within the share. In the case where it would be preferred that shares are not created; a configuration option is available to restrict share creation. As well as the default share creation, when installing the Adaptiva client, a local user is created on each client in order to provide access to the OSD content push shares, there is also a configuration option available to remove the local AdaptivaClient account.

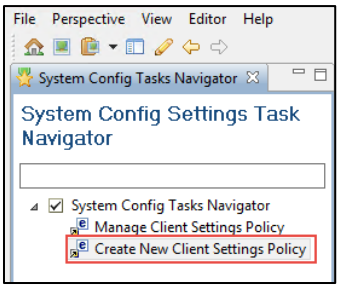
Disabling OSD Content Push Shares and Adaptiva Local User Accounts

A System Configuration policy can be set to disable OSD Content Push shares as well as remove the local AdaptivaClient user account by following the below procedure:

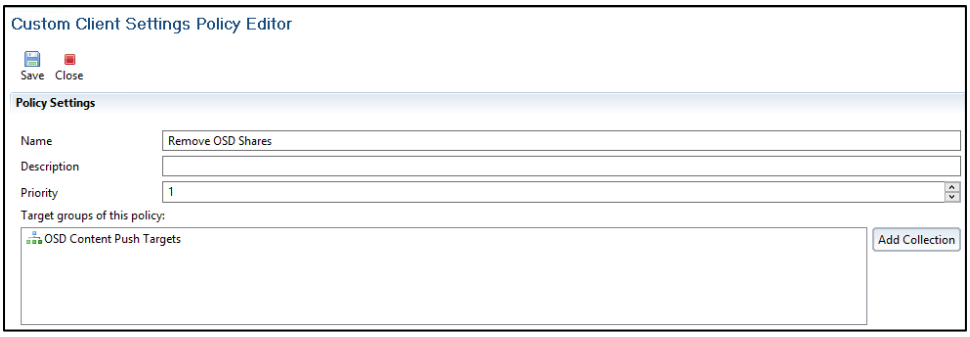
1. In the Adaptiva Workbench, in the "Workbench Perspectives" home, expand the **Home** folder and launch the **System Configuration Perspective**.



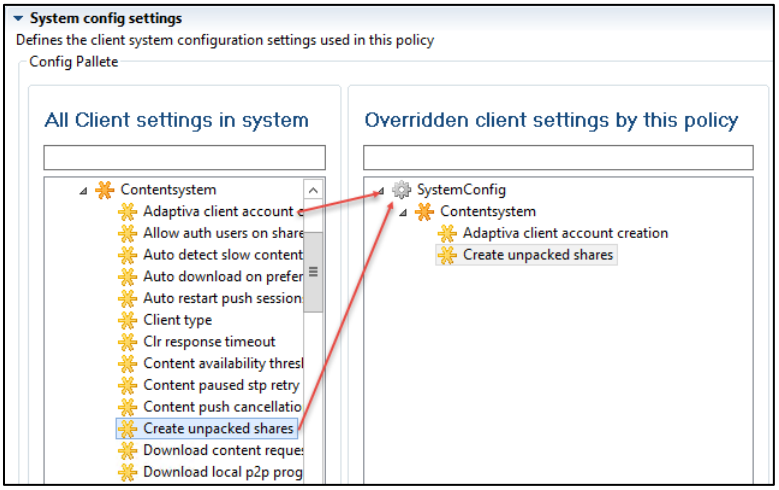
2. In the "System Config Settings Task Navigator", select **Create New Client Settings Policy** which should open a new policy in the editor.



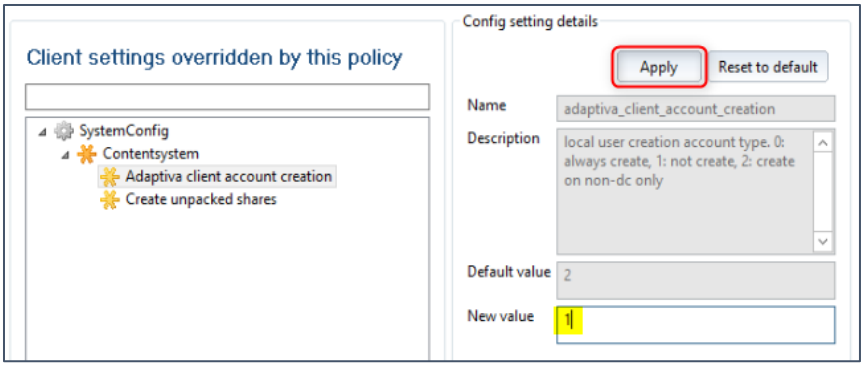
3. In the "Custom Client Settings Policy Editor", in the **Name** field, enter a name for the policy such as "Remove OSD Shares". Optionally, enter a **Description** or change the **Priority** field as appropriate.
4. In the "Target groups of this policy:" field, click the **Add Collection** button to add a collection which is targeted for OSD Content Push policies.



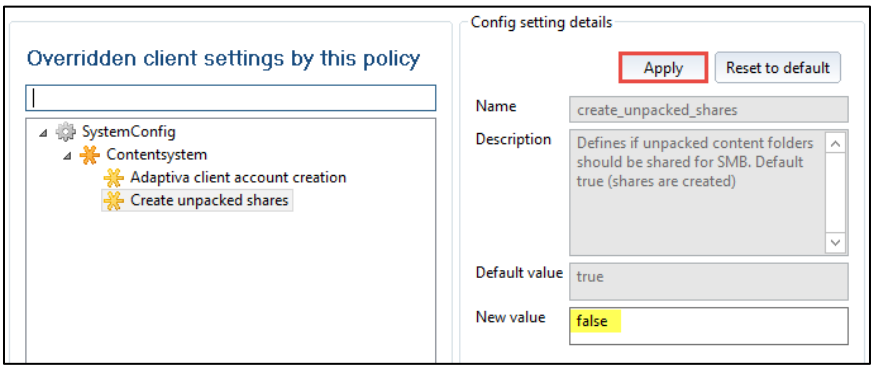
5. At the "System config settings" section, within the "All Client settings in system" field, expand **SystemConfig – Contentsystem**, then select **Adaptiva client account creation** and drag it into the **SystemConfig** in the "Overridden client settings by this policy" field. Repeat the process with the **Create unpacked shares** setting.



6. In the "Overridden client settings by this policy" section, select **Adaptiva client account creation** and in the "Config settings details" section, in the "New value" field, enter the value **1** then click **Apply**.



7. In the "Overridden client settings by this policy" section, select **Create unpacked shares** and in the "Config settings details" section, in the "New value" field, enter the value **false** then click **Apply**.



8. Once complete, scroll to the top of the editor and click **Save** to apply the policy to the target clients. The clients should receive the policy immediately.

Manually Disabling OSD Content Push Shares and Adaptiva Local User Accounts

Alternatively, these can be disabled by modifying the following registry values.

To disable OSD Content Push shares manually, the following registry value can be set on clients:

```
Key: HKLM\Software\<WoW6432Node>\Adaptiva\Client
Value Name: contentsystem.create_unpacked_shares
Value: false
```

To remove local AdaptivaClient user accounts, the following registry value can be set on clients:

```
Key: HKLM\Software\<WoW6432Node>\Adaptiva\Client
Value Name: contentsystem.adaptiva_client_account_creation
Value: 1
```

adaptiva

Implementing OneSite Peer-to-Peer PXE

Overview

PXE (pronounced as pixie) is a set of protocols designed to boot computers using a network card, without requiring any pre-existing operating system. It was introduced by Intel in 1999 and builds upon widely used protocols such as IP, UDP, DHCP, and TFTP.

In fact, PXE does not use its own protocol, but rather is an extension of DHCP. It adds headers to the DHCP broadcast packages to declare its request for a PXE response. Originally, it was intended to have a PXE responder on the same subnet as the PXE client to respond to the broadcast messages.

Microsoft ConfigMgr provides support for PXE protocol to enable bare metal image deployment scenarios. This would normally require the installation of a "PXE Service Point" site system role on a server (SCCM 2007) or a PXE enabled Distribution Point (SCCM 2012/Current Branch), along with prerequisites such as Windows Deployment Services (WDS), and the associated changes to network infrastructure. Most environments utilize IP Helpers on their network equipment to forward DHCP broadcasts to the DHCP server. Likewise, one of the following is required to forward the PXE broadcast traffic across network segments:

- IP helper on network routers and level 3 switches to forward packets to PXE.
- Option configuration on DHCP server to point clients to a specific PXE responder.

These requirements for deploying servers, WDS, and network infrastructure changes present serious challenges in large distributed networks, where bare metal provisioning capabilities need to be provided to hundreds or thousands of far flung locations.

Benefits of Peer-to-Peer PXE

Adaptiva's Peer-to-Peer PXE technology is a revolutionary advance in bare metal provisioning for large global networks because:

- No servers or server roles are required.
- No router changes are required.
- No DHCP server configuration changes are required.

Enabling Peer-to-Peer PXE is straightforward:

- Decide which parts of your network require PXE capabilities.
- Create one or more SCCM collections that correspond to these parts of your network or check the "Use All Adaptiva Clients" to enable PXE globally.

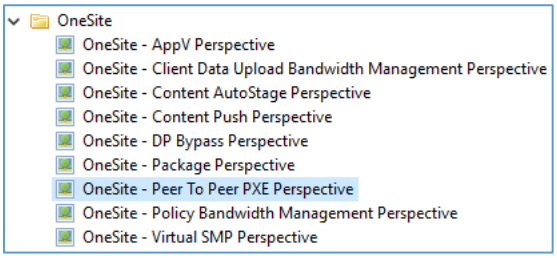
- Enable P2P PXE in the Adaptiva Workbench.

Peer-to-Peer PXE capabilities will be enabled on each subnet within the selected parts of your network. All the necessary software components will be automatically deployed to clients, and PXE capabilities will become available for use within the next 1-2 minutes.

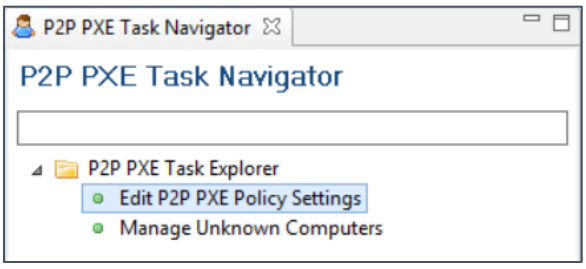
Peer-to-Peer PXE Perspective

To enable Peer-to-Peer PXE, you must first open the "Peer to Peer PXE Perspective", which contains UI for enabling and using P2P PXE.

1. In the Adaptiva Workbench, open the "Home" perspective by clicking on the **Home** icon in the toolbar.
2. In the "Workbench Perspectives" pane, expand the **OneSite** folder and open the **OneSite – Peer To Peer PXE Perspective**.



3. In the "P2P PXE Task Navigator" pane on the left, there are two entries. To enable or disable Peer-to-Peer PXE open the **Edit P2P PXE Policy Settings** item.



4. The right part of your screen contains a Collection explorer. It displays all your SCCM collections and allows you to drag and drop them into the "P2P PXE Settings" editor. As always, the Collection explorer automatically detects the changes whenever you create, delete, or modify any collections, and refreshes itself. It also lets you find the collection you want by typing a few characters of its name in the search box.

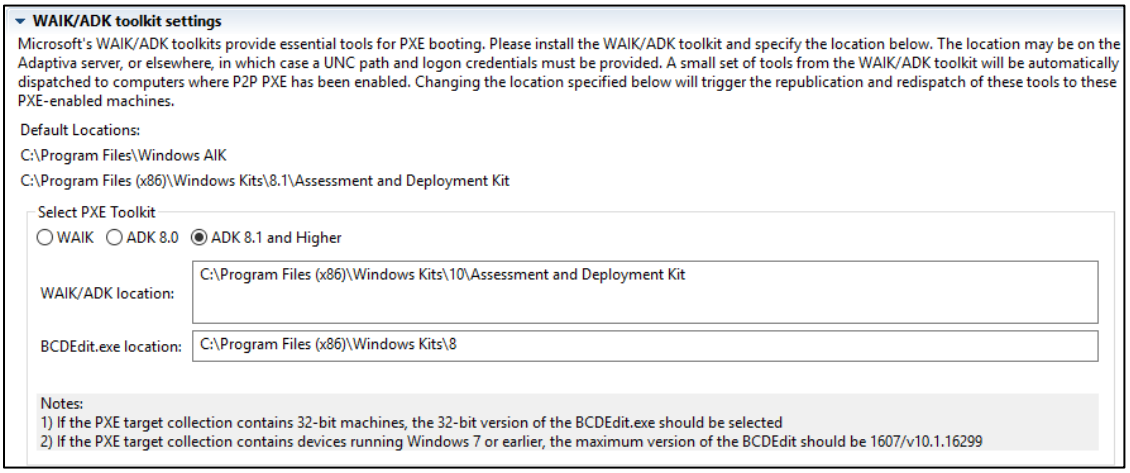
Enabling Peer-to-Peer PXE

P2P PXE uses specific Microsoft utilities for PXE boot which would normally be present on a ConfigMgr server.

- For Vista/Windows 7: Windows Automated Installation Kit (WAIK).
- For Windows 8-10: Assessment and Deployment Toolkit (ADK).
- From Windows 10 1809 version: Windows PE Add-on for ADK.

> *NOTE: These only need to be downloaded in the case in where Adaptiva was installed on a separate server than ConfigMgr.*

1. Once the supporting tools are ready, open the "P2P PXE Settings" editor by clicking the **Edit P2P PXE Policy Settings** item in the "P2P PXE Task Navigator".
2. In the "P2P PXE Settings" editor, check the box to **Enable P2P PXE**.
3. In the "WAIK/ADK toolkit settings" section, specify the following:



Select PXE Toolkit - Select the appropriate toolkit that is installed.

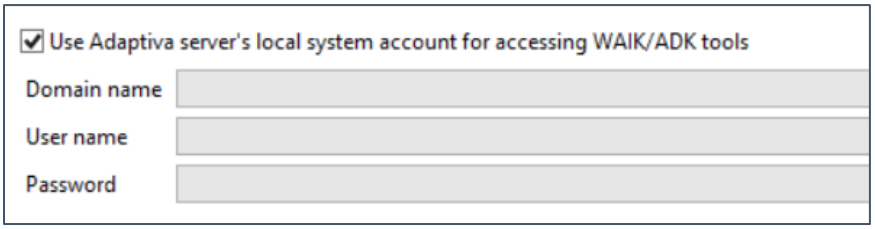
WAIK/ADK location – Enter the path to the toolkit installation folder.

Ex: C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit

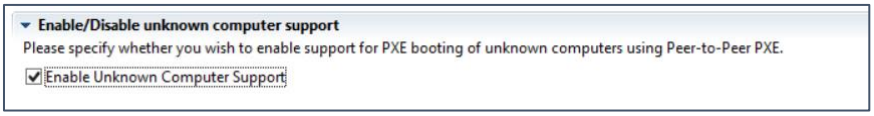
ADK 8.0 BCDEdit.exe location – Enter the path to the where the BCDEdit.exe utility is located.

> ***Warning: If the ADK ε.Ь and higher option is selected, the bcdedit.exe utility from ADK 8.0 must be specified in order to support older Oss that you may deploy. It can be downloaded and extracted from the following link:***
>
> ***http://adaptiva.cloud/builds/private/bcdedit.zip***

4. If the path specified is on a remote server and the Adaptiva server's computer account doesn't have access to the location, uncheck the box: **Use Adaptiva server's local system account for accessing WAIK/ADK tools** and enter the credentials to be used.

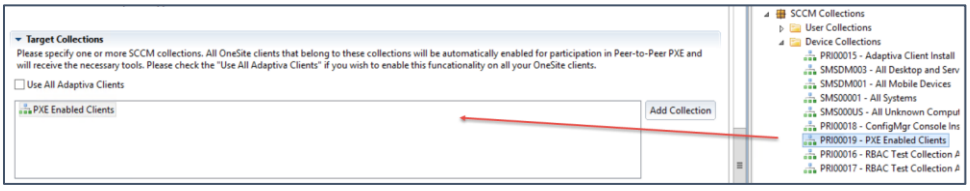


5. In the "PXE Support for EUFI Devices" section, check the box **Enable Support for PXE Booting UEFI Devices** in the case you want to PXE boot EUFI devices.
6. In the "Enable/Disable unknown computer support" section, check the box **Enable Unknown Computer Support** to enable unknown computer support.



NOTE: For OneSite to work with Unknown Computers, it must be enabled within the selected boot images in ConfigMgr.

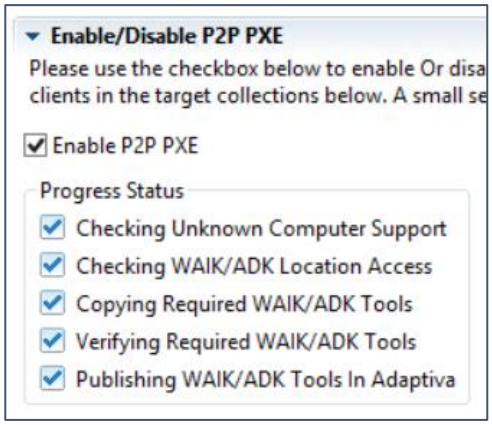
7. In the "Target Collection" section, drag and drop one or more SCCM collections into the **Target Collection** field, or alternatively, check the box **Use All Adaptiva Clients**. All the computers that are part of these collections will automatically receive a small number of WAIK/ADK tools and will become capable of becoming PXE responders and TFTP servers to serve other machines on their respective subnets.



NOTE: It is highly recommended that you enable Peer-to-Peer PXE on All Adaptiva Client computers, and completely do away with PXE servers in your entire SCCM environment.

The remaining sections described below should be considered optional to enable P2P PXE.

8. In the "Mac Exclusion List" section, add any client MAC addresses you would like to not PXE boot to a PXE enabled client.
9. In the "Task Sequence Variables" section, add any task sequence variables and values that OneSiteDownloader may read.
10. In the "P2P PXE Server Workflow" section, you can specify any custom PXE workflow.
11. In the "PXE Approval Workflow" you can specify an approval workflow to approve PXE requests.
12. Once complete, at the top of the editor, check the box, **Enable P2P PXE** and click the **Apply** button at the top of the screen. The Progress Status section will indicate any failures.



P2P PXE Verification

Peer-to-Peer PXE works silently in the background, without requiring any day to day operational supervision from the administrator. It can be reassuring to physically verify that it has actually been enabled and that the required elements are indeed in place.

On the Adaptiva server, you can verify that the WAIK /ADK tools have been published as Adaptiva content

1. On the Adaptiva server, navigate to:
   <AdaptivaInstallFolder>\AdaptivaServer\Data\ContentLibrary
2. Notice the file: **Adaptiva$WAIK$.1.content**. This file contains 10 files which will be sent to clients to enable PXE capabilities. The file size should be about 1.77 MB.

> *NOTE: To view the files inside the .content archive, use a tool such as 7-Zip to open the archive, or temporarily change the extension to .zip and open with Windows File Explorer.*

3. On one or more Adaptiva clients, you can verify that the WAIK / ADK tools have been automatically downloaded by opening the folder: <AdaptivaClientInstallFolder>\AdaptivaClient\data\p2ppxe.

   The following 10 files will be present:

   - Abortpxe.com
   - Adaptivaboot.efi
   - Adaptivaiboot.efi
   - AdaptivaSecureHash.xml
   - Bcdedit.exe
   - Boot.sdi
   - Bootmgr.exe
   - Pxeboot.com
   - Pxeboot.n12
   - Wgl4_boot.ttf

4. On one or more Adaptiva client's registry, you can verify that Peer-to-Peer PXE has been turned on by reviewing the following registry value:

   For Adaptiva 5.5 Clients or Above:

```
Key: HKLM\Software\<WoW6432Node>\Adaptiva\Client
Value Name: p2p_pxe.pxe_enabled
Value: true
```

   For Older versions of the Client:

```
Key: HKLM\Software\<WoW6432Node>\Javasoft\Prefs\Adaptiva\Client
Value Name: p2p_pxe.pxe_enabled
Value: true
```

Using Peer-to-Peer PXE

Adaptiva OneSite's Peer-to-Peer PXE is simple in nature and requires little management. You just do what you'd do for bare metal provisioning - no extra steps are required.

Some common gotchas have been documented here for you:

Make sure you have advertised an appropriate task sequence to an appropriate SCCM collection, including to unknown computers, if you're using unknown computers.

Make sure the computer you're trying to boot isn't showing up under "Unknown computers list" in the Adaptiva workbench

For additional information on the P2P PXE Process and Troubleshooting assistance, see the following KB article; https://support.adaptiva.com/hc/en-us/articles/115002832351-P2P-PXE-Process-Troubleshooting-Guide

Remote PXE Option

In the case where a network device is configured to disallow DHCP traffic from untrusted sources (DHCP Snooping), it is likely that DHCP/ PXE traffic on the network will be blocked. Another scenario to consider is when DHCP / PXE traffic is being routed to a remote network (using IP Helpers or DHCP Options), where a DHCP server resides. If DHCP traffic is being routed to another location, it will interfere with local Adaptiva PXE P2P responders from servicing PXE broadcast traffic. To address either scenario, a configuration option called Remote PXE is available in which Adaptiva P2P PXE clients can provide P2P PXE services to clients on another subnet.

To enable this feature, the following requirements must be met:

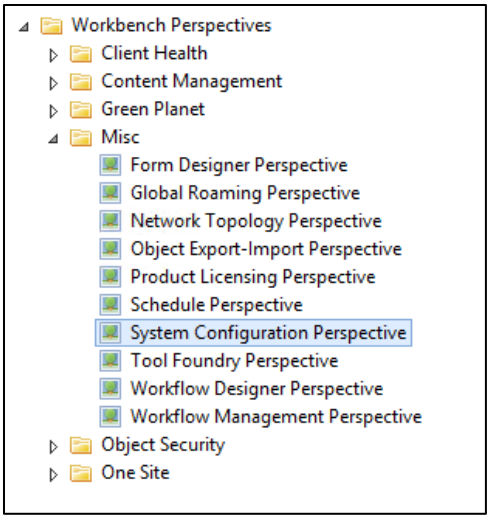
- The subnet hosting the DHCP server, or wherever the DHCP traffic is being routed, should host at least one Adaptiva client.
- If DHCP Snooping is enabled, an IP Helper rule must be configured to forward DHCP traffic to the remote subnet which will host the Adaptiva client.
- P2P PXE must be enabled in the Workbench for machines in the remote subnet where DHCP / PXE traffic is being routed to support the PXE responder.
- P2P PXE must also be enabled for machines in the location where PXE clients would be booting so that the appropriate PXE utilities are available for TFTP boot.
- A System Configuration policy with the Remote PXE setting must be applied to machines in the remote subnet where DHPC / PXE traffic is being routed.

How to Enable the Remote PXE System Configuration Setting

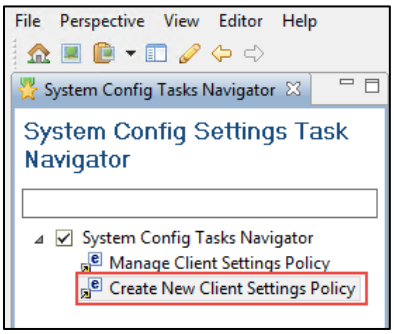
The Remote PXE settings can be enabled using the System Configuration Perspective, or via a registry value, which has the same result.

To enable Remote PXE using the System Configuration Setting, follow the below process:

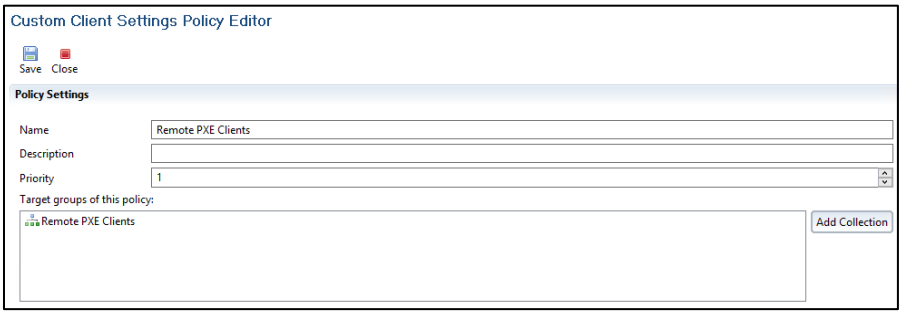
1. In the Adaptiva Workbench, in the "Workbench Perspectives" home, expand the **Home** folder and launch the **System Configuration Perspective**.



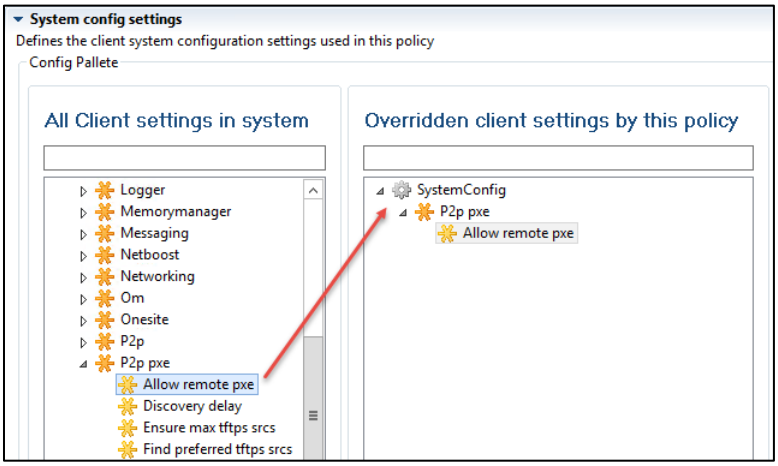
2. In the "System Config Settings Task Navigator", select **Create New Client Settings Policy** which should open a new policy in the editor.



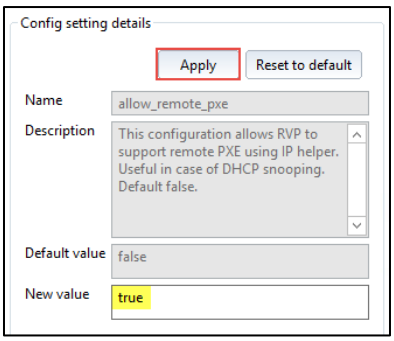
3. In the "Custom Client Settings Policy Editor", in the **Name** field, enter a name for the policy such as "Remote PXE Clients". Optionally, enter a **Description** or change the **Priority** field as appropriate.
4. In the "Target groups of this policy:" field, click the **Add Collection** button to add a collection which contains Adaptiva clients that reside on the subnet in which DHCP / PXE traffic is being routed.



5. At the “System config settings” section, within the “All Client settings in system” field, expand **SystemConfig – P2p pxe**, then select **Allow remote pxe** and drag it into the **SystemConfig** in the “Overridden client settings by this policy” field.



6. To the right, in the “Config settings details” section, in the “New value” field, change the value from false to **true**, then click the **Apply** button.



7. Once complete, scroll to the top of the editor and click **Save** to apply the policy to the target clients. The clients should receive the policy immediately.

The following registry value can be set manually:

```
Key: HKLM\Software\<WoW6432Node>\Adaptiva\Client
Value Name: p2p_pxe.allow_remote_pxe
Value: true
```

adaptiva

Implementing OneSite vSMP

Introduction

Adaptiva OneSite includes a feature called Virtual State Migration Points or Virtual SMP. The Virtual SMP offers an efficient alternative to the ConfigMgr State Migration Point role. Much like OneSite enables the elimination of secondary sites and distribution points, Virtual SMPs make use of the revolutionary OneSite Virtual SAN, the Caching File System, and Peer-to-Peer technologies to enable the elimination of State Migration Points from ConfigMgr environments.

Virtual SMP tasks are integrated directly into the SCCM Task Sequence UI for seamless administration and operation. Deep integration is also provided with SCCM Computer Associations, enabling the use of OneSite's Virtual SMP as a simple replacement for potentially hundreds of physical SMP servers.

Features

Peer state storage - State data is saved in the OneSite Virtual SAN and the Adaptiva Caching File System. By default, state stores are "pinned" to the cache, so that they are not automatically deleted in the event of low disk-space conditions.

State store redundancy - Multiple copies of state data can be stored, so that if one copy disappears or goes offline, another is available. Additionally, if no state stores are online during the restore process, OneSite will wake up a client that has hosted the store.

Storage lifetime control - Allows space allocated for state stores to be freed after a specified lifetime.

Integration with Config Manager Computer Association functionality - Publication of a state store can be linked to a Config Manager Computer Association object.

Zero peer impact - Peer machines which provide hosting for the Virtual SMP state store will not see any disk space utilization or increased CPU utilization because of the caching file system, extensive load leveling, and advanced host selection algorithms performed by Adaptiva.

Concepts

State store

A state store is a combination of free space that exists on participating peer hosts, and all the metadata and settings that were specified during its creation and subsequent maintenance (e.g., expiration and disuse expiration, required host count, desired host count, etc.)

Among other metadata, each state store contains a list of participating hosts, a share name, and a UNC path that may be used to store and retrieve files to and from the state store.

Redundancy and Replication

State stores are fundamentally designed to exist on multiple participating hosts and are geared towards redundancy amongst the participating hosts. Data may be written to one of the participating hosts, usually using the UNC path that is present in the state store's metadata, and then the Virtual SMP may be requested to perform replication, which places copies of data on all the remaining hosts.

> *NOTE: User State is stored in its entirety on each participating host. The data is never divided between hosts.*

Trimming

At the time state stores are created it is not clear exactly how much space will be required to store the USMT data for a particular user, but an initial size must be specified. Once the USMT capture step has completed and the store has been populated with data, if the data is less than the allocation then the store can be "trimmed," so that the store only uses the actual required space.

Pinning

Pinning a state store prevents the Adaptiva client from deleting that content to free up space. The Adaptiva File Caching system is an intelligent system involving clients that are in constant communication with each other. The clients keep track of how many copies of each piece of content exist on their subnet or at their location, and a client will use this information in making decisions about what pieces of content can be deleted from its own cache if the user needs more disk space.

Publication and Discovery

State stores are designed to survive for the entire duration of the expiration interval which was specified during their creation. The machine that creates the state store may reboot multiple times, and may be formatted, renamed, or completely retired, without affecting the longevity of the state store itself.

To allow a state store to be used in scenarios where the store is created on one machine during USMT capture and needs to be accessed from a different machine during USMT restore, an elaborate publication and discovery system has been built into OneSite's Virtual SMP.

This process has also been integrated with SCCM Computer Associations, allowing Virtual SMPs to be used as replacements for physical SMPs.

Publication

Publication is the process by which Adaptiva clients are made aware of OneSite objects in the Adaptiva Peer-to-Peer system. The following data fields are published at the time of creating a state store:

- Source machine's MAC address
- Source machine's SMBIOS ID
- Target machine's MAC address
- Target machine's SMBIOS ID

- Allocation ID

For ease of use, the "OneSite – Create State Store" task sequence step automatically determines and populates the values of all these data fields.

If the "Publish state store for source computer in the computer association" option is selected:

Selecting this option makes the state store discoverable using the SCCM computer association for this source machine. Usually used in a "Replace" scenario moving from one machine to another.

- Source machine's MAC address - the MAC address of the machine where the "Create" custom task sequence step is running is stored in this field.
- Source machine's SMBIOS ID - the SMBIOS ID of the machine where the "Create" custom task sequence step is running is stored in this field.
- Target machine's MAC address – the target machine's MAC address is automatically obtained from the computer association whose source MAC address and source SMBIOS ID match the values for the machine where the "Create" custom task sequence step is running. If no such computer association exists, this value is left blank.
- Target machine's SMBIOS ID - target machine's SMBIOS ID is obtained automatically from the computer association whose source MAC address and source SMBIOS ID match the values for the machine where the "Create" custom task sequence step is running. If no such computer association exists, this value is left blank.
- Allocation ID - an automatically generated GUID is used during creation, and a blank value is used for matching during find.

If the "Publish state store for target computer in the computer association" option is selected

Selecting this option makes the state store discoverable using the SCCM computer association for this target machine. Usually used when performing a "Wipe & Load" on the same machine.

- Source machine's MAC address - source machine's MAC address is obtained automatically from a matching computer association, if one exists, else left blank.
- Source machine's SMBIOS ID - source machine's SMBIOS ID is obtained automatically from a matching computer association, if one exists, else left blank.
- Target machine's MAC address - local machine's MAC address is used.
- Target machine's SMBIOS ID - local machine's SMBIOS ID is used.

- Allocation ID - an automatically generated GUID is used during creation, and a blank value is used for matching during find.

If the "Publish state store ID specified in the following task sequence variable" option is selected:

Selecting this option makes the state store discoverable via the Allocation ID field, which will be populated with the value stored in the specified task sequence variable. With this option, the Allocation ID field is the only field whose value will be populated during publication.

- Source machine's MAC address - left blank.
- Source machine's SMBIOS ID - left blank.
- Target machine's MAC address - left blank.
- Target machine's SMBIOS ID - left blank.
- Allocation ID - the value stored in the specified task sequence variable is used for publication. This method must be used during the find operation.

Discovery

At the point in the task sequence when user state data will be restored, the state store containing that data must be discovered.

At the time it was created, the state store was published in the Adaptiva Peer-to-Peer system using the publication fields described above. It is possible that some of the publication fields were left blank, depending on which publication option was specified in the "Create" custom task sequence step and whether an SCCM computer association was found.

For this reason, the Search Parameter option that is specified in the "Find" custom task sequence step should correspond with the Publication option that was specified in the "Create" custom task sequence step.

For example, if the if the "Publish state store for source computer in the computer association" option was used during the "Create" custom task sequence step, then the "Find state store for source computer in the computer association" option should be specified during the "Find" custom task sequence step.

Deletion

Once user state data has been restored, the state stores containing that data can be safely deleted. Deletion can take place in one of two ways:

- Explicitly, by use of the "OneSite – Delete State Store" custom task sequence step.
- Automatically, by means of the Storage Expiration parameter specified in the "Create" custom task sequence step. For automatic deletion the "Delete" custom task sequence step would not be used, and the state store would be allowed to expire.

Installing and Enabling OneSite Virtual SMP

Minimum Requirements

Adaptiva OneSite Virtual SMP works with all currently supported versions of Configuration Manager.

Installation

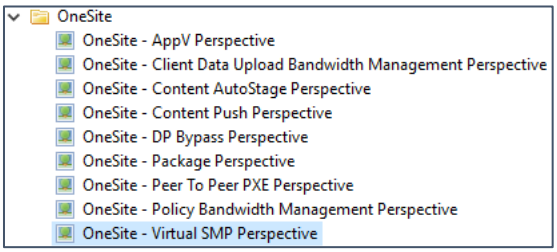
Installation of Virtual SMP is part of the OneSite installation process. By installing OneSite Server, you have automatically expanded the options that are available to you in the Config Manager Task Sequence Editor. If the ConfigMgr console is open during the installation of OneSite, the tasks may not be available until the console is relaunched.

> *NOTE: To use the OneSite Virtual SMP custom task sequence extensions in a ConfigMgr console running on a machine other than the site server, simply install the Adaptiva Workbench on that machine.*

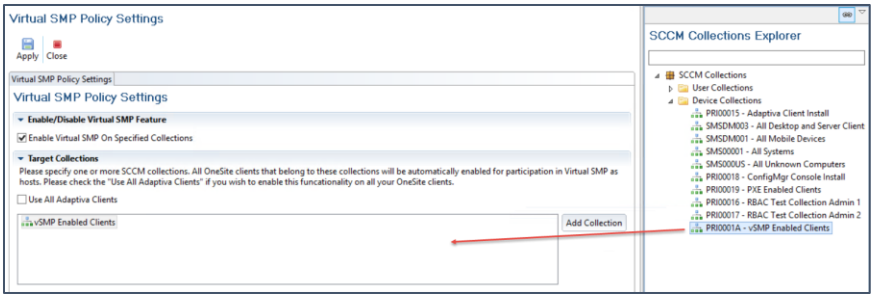
Enablement

To enable the Virtual SMP feature:

1. In the Adaptiva Workbench, open the "Home" perspective by clicking on the **Home** icon in the toolbar.
2. In the "Workbench Perspectives" pane, expand the **OneSite** folder and open the **OneSite – Virtual SMP Perspective**.



3. To enable the Virtual SMP feature, check the box **Enable Virtual SMP On Specified Collections**.
4. In the "Target Collections" section, drag a collection of machines you want to serve as Virtual SMPs or select **Use All Adaptiva Clients**.



5. Click **Apply** to enable the Virtual SMP feature on the target systems. This will enable all targeted clients to participate as state store hosts for OneSite Virtual SMP.

Using Adaptiva OneSite Virtual State Migration Point

Using Microsoft's USMT tool in a ConfigMgr Task Sequence usually involves four task sequence steps for capturing and restoring user files and settings:

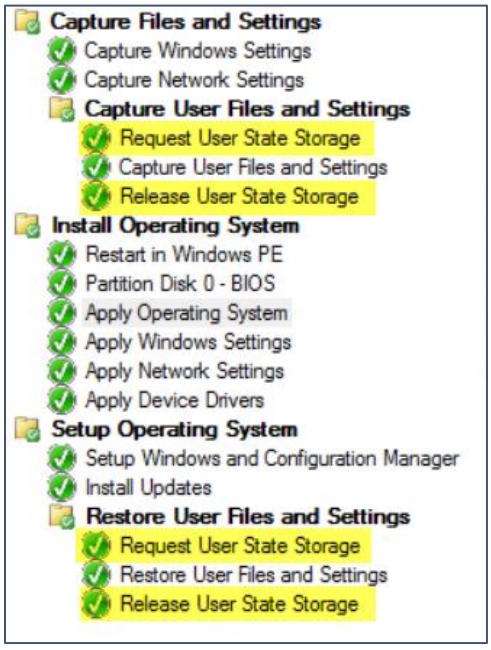
- Request State Store
- Capture User State
- Restore User State
- Release State Store

Of these, **Request State Store** and **Release State Store** can utilize a physical State Migration Point server and are the steps that will be replaced with the custom task sequence steps supported by OneSite Virtual SMP.

> *NOTE: **Capture/Apply Windows Settings** and **Capture/Apply Network Settings** are not part of USMT, and do not require a State Migration Point.*

Sample Config Manager Task Sequence for use with physical State Migration Points

A task sequence that includes USMT state migration steps may look something like the image below. The steps that will be replaced in this task sequence by OneSite Virtual SMP custom task sequence steps are highlighted:



Using Virtual SMP Custom Task Sequence Steps in a Task Sequence

OneSite extends the ConfigMgr Task Sequence editor by making eight new custom task sequence steps available. All of the OneSite Virtual SMP Custom Task Sequence steps can be run in either a standard Windows operating system environment or in the Windows Pre-installation (WinPE) Environment.

The new OneSite Virtual SMP steps are:

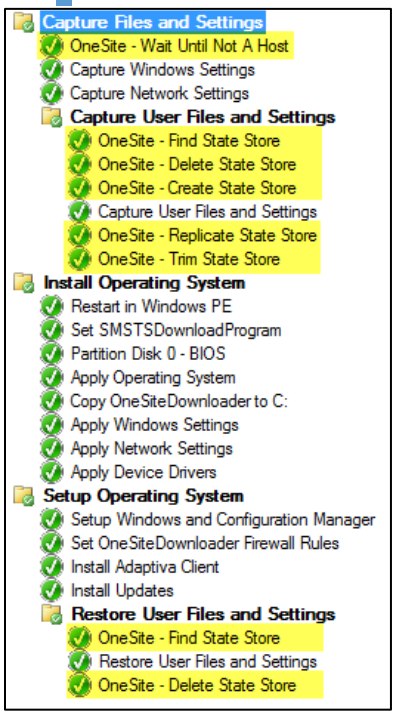
- Create State Store
- Pin State Store
- Replicate State Store
- Trim State Store
- Find State Store
- Delete State Store
- Is Machine a Host
- Wait Until Not a Host

Sample Config Manager Task Sequence for Use with OneSite Virtual SMP

The sample task sequence shown earlier, modified to use OneSiteDownloader and OneSite's Virtual SMP, might look something like the example below.

The OneSite Virtual SMP steps are highlighted.

> *NOTE: It is recommended using the "Connect to Network Folder" task sequence step before the "Capture User State" and "Restore User State" steps. In the "Connect to Network Folder" task sequence step, set the 'Path' property to %OSDStateStorePath% and set the 'Account' to the Network Access Account.*

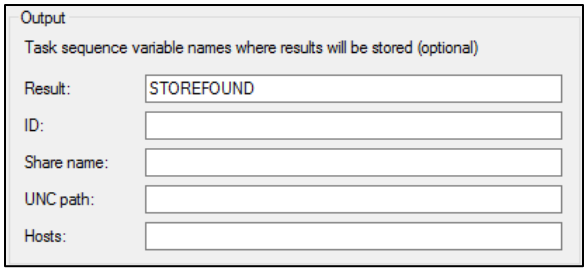


In this example, the first task is the "OneSite - Wait Until Not A Host" task so that the machine does not get imaged while it is serving as a host of state data for a different migration. This enables use cases which involve imaging a large number of machines concurrently.

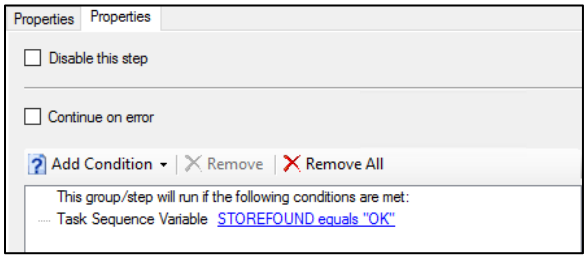
Before creating the state store, this task sequence uses the "OneSite - Find State Store" task to locate previous state stores which should be deleted before capturing another state store.

You must set this step to Continue on Error, as if it fails to find a State Store, this is a good thing and means we can continue.

Also, In the Properties for the "OneSite - Find State Store" the Result field can include a value of STOREFOUND. This is optional and only needed when there is the possibility that a capture was done without a completed restore.



A conditional "OneSite - Delete State Store" task can be added right after, with the Condition defined, Task Sequence Variable: STOREFOUND equals OK. If a state store is found, this task will remove any existing state stores for this machine. This would only be used with the previous setting.



Next the task, "OneSite – Create State Store" will be added to create the target for storing the user's data and settings.

After the state store is created and the user files and settings are captured, USMT data is replicated to all the hosts using the "OneSite - Replicate State Store" task.

The "OneSite - Trim State Store" task then reduces the size of the store from the estimated size to the actual size that is in use, if the initial size was larger than the actual data.

Once the operating system has been installed and configured, the "OneSite - Find State Store" task uses the Configuration Manager Computer Association to find the correct state store. The settings and data are restored to the newly imaged machine.

In this example, we have added the step "OneSite – Delete State Store" to immediately remove this store from any and all machines that housed a replica of this data. This could be for security purposes. Otherwise, we would likely skip this step and allow the data to be deleted once it expires, allowing us the safety net of being able to restore this again should something happen to our newly imaged machine.

Various combinations of OneSite's Virtual SMP custom task sequence tasks can be incorporated into task sequences. The next section provides details about the use of each task.

Virtual SMP Task Sequence Steps

OneSiteDownloader in a Virtual SMP Task Sequence

All of the OneSite Virtual SMP steps make use of the Adaptiva OneSiteDownloader utility to carry out their tasks. Each of the steps includes an input parameter that points to the location of OneSiteDownloader.exe.

> *NOTE: To use OneSite Virtual SMP on 64-bit systems, OneSiteDownloader64.exe needs to be renamed OneSiteDownloader.exe and reference it in the task sequence by providing the folder location in the input parameter for each OneSite Virtual SMP step.*

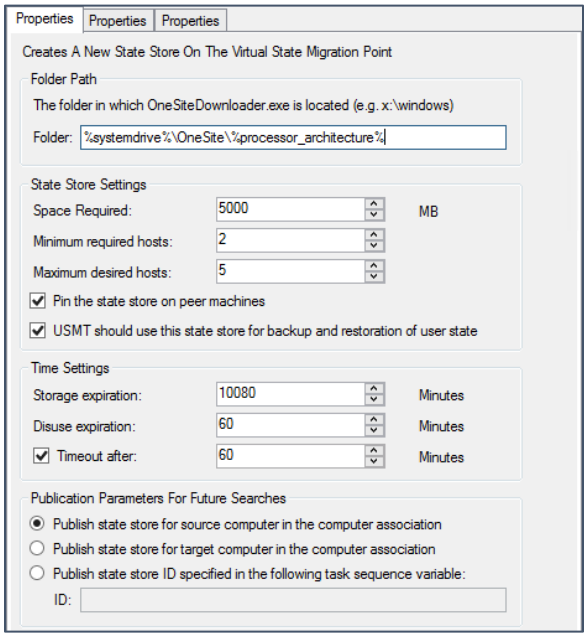
OneSite - Create State Store

The **OneSite - Create State Store** task allocates space for USMT data to be stored on the Adaptiva Virtual SAN. This step is a replacement for the "Request State Store" step in ConfigMgr.

Input parameters are provided via the Config Manager Task Sequence editor when the step is created.

Properties

Properties Tab One



- **Folder Path** - specifies the folder where the OneSiteDownloader.exe file is located. This can be a UNC path, or a local path, and must be available to the environment in which the custom task sequence step is executed. For steps that occur within the Windows PE section of the task sequence, this property specifies the OneSiteDownloader.exe file location in the boot image, e.g. %systemdrive%\OneSite\%processor_architecture%. (See Note above)

- **Space required** - specifies the estimated amount of storage space required, in megabytes, for the user state stores. This amount of space will be allocated on each machine that is determined to be a qualified host for this state store. If the user's data is larger than this allocated space, then it will dynamically expand to capture all of the user's data. However, this does add some delay to the process so tuning this to match the average store size is an important part of the design. A separate step can be included in the task sequence to trim the store after the user state data is copied, to preserve space.

  Range: 10 - 1024000; Default: 1000MB.

- **Minimum required hosts** - specifies the required number of peers that must be available to store redundant copies of the user state associated with the task sequence. The task sequence will not continue unless at least this number of peer hosts has the required available free space. If you want to ensure that there is a host available to restore from, you will likely want to increase this value.

  Range: 1 - 100; Default: 2.

- **Maximum desired hosts** - specifies the greatest number of peers that will store copies of the user state. If available, this is the number of hosts that will store the user state data associated with the task sequence. The task sequence will continue if this number of qualified peers is not available, as long as the "Minimum required hosts" parameter is satisfied.

  Range: 1 - 100; Default: 5.

- **Pin the State Store on Peer Machines** - specifies whether the state store should be pinned by default during this step. If the Adaptiva cache driver needs to make room on the host's caching file system for other content, pinned state stores will not be considered for deletion. State stores which have not been pinned are subject to automatic deletion in the event of low disk space conditions on the host.

  Default: checked.

- **USMT should use this state store for backup and restoration of user state** - this checkbox must be checked to direct USMT to use the store that is being created by this task sequence step.

This is provided as an option so that a single task sequence may be used in multiple scenarios, some of which may not use the Adaptiva OneSite Virtual SMP.

Default: checked.

- **Storage Expiration** - specifies the amount of time, in minutes, that the state store will remain in the cache on each host. The state stores will be automatically removed from the caches within one hour of this limit being reached, and all content in that state store will be irretrievably lost.

  Range: 60 - 100800; Default: 10080 minutes (7 days).

- **Disuse Expiration** - is the amount of time, in minutes, that the Adaptiva cache system will wait for content to be copied to a store that has been allocated, before deleting the store. If the state store is found to be completely empty after this amount of time, the state stores will be automatically removed from the caches within one hour of this limit being reached, and future operations on the state store will fail.

  Range: 5 - 100800; Default: 60 minutes.

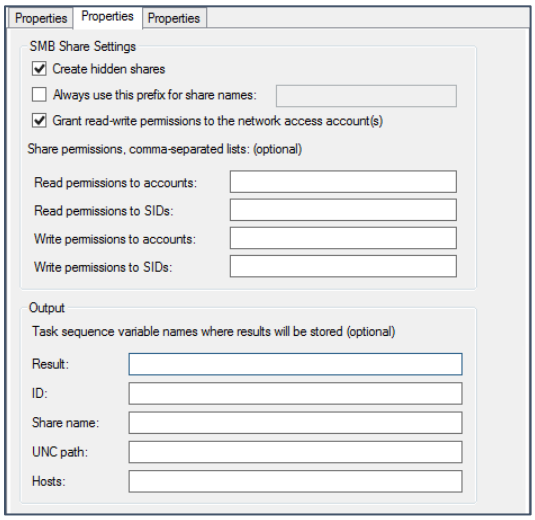
- **Timeout after** - If the box is unchecked the task sequence will wait indefinitely for the storage to be allocated. If the box is checked, the task sequence will time out and fail after the specified number of minutes, if the state store has not been created on the required number of hosts.

  Range: 5 - 14400; Default: checked, 60 minutes.

- **Publication Parameters for Future Searches** - There are three options for this setting, only one of which can be selected. These parameters control how the share for the state store will be published in the Adaptiva P2P system for future discovery. This information may be used later, when a Find step is executed for the state store:
  - **Publish state store for source computer in the computer association** - allows the source computer information from the ConfigMgr computer association to be used in the state store to be published. This is the default publication parameter option.
  - **Publish State Store for target computer in the computer association** - allows the target computer information from the ConfigMgr computer association to be used in the share to be published.

- **Publish State Store ID specified in the following task sequence variable** - allows a state store GUID to be used and stored the specified task sequence variable. The state store GUID is automatically generated by OneSite when the store is created. If the specified task sequence variable does not exist, it will be created automatically.

Properties Tab Two



- **Create hidden shares** - when this box is checked, the SMB shares that are created for the state store will be hidden shares.

  Default: checked.

- **Always use prefix for share names** - this option allows a consistent prefix to be used in the names of the SMB shares that are created. If this option is checked, a prefix name is required. SMB shares will be created with this naming syntax:

  <MyPrefix>**sa***StoreID* <$>

  Where <MyPrefix> is the string provided in this parameter. The acronym **sa** is inserted to identify this as a "Store Allocation" share. The *StoreID* is the GUID that is automatically generated when the task sequence executes, and the trailing "$" is included if the option to create hidden shares is enabled.

  The entire share name with a prefix of VirtSMP, an allocation ID of deeb05f7-e86c-4d1a-813d-30ebffee7b20, and the "Hidden Shares" option selected will be:

  VirtSMPsadeeb05f7-e86c-4d1a-813d-30ebffee7b20$

  Default is unchecked, no prefix.

- **Grant read-write permissions to the network access account(s)** – specifies whether read-write permissions to the state store SMB shares should be granted to the Config Manager Network Access Account.

  Default: checked.

- **Share permissions, comma-separated lists (optional)** - read and write permissions to the state stores can be granted to accounts and/or SIDs in these fields.
  - **Read permissions to accounts –** optionally enter a comma-separated list containing names of domain accounts that will be granted read permissions to the state store SMB share. Names must be entered in "domain\user" format.
  - **Read permissions to SIDs** – optionally enter a comma-separated list containing SIDs of domain accounts that will be granted read permissions to the state store SMB share.
  - **Write permissions to accounts –** optionally enter a comma-separated list containing names of domain accounts that will be granted write permissions to the state store SMB share. Names must be entered in "domain\user" format.
  - **Write permissions to SIDs –** optionally enter a comma-separated list containing SIDs of domain accounts that will be granted write permissions to the state store SMB share.
- **Output** - The "OneSite - Create State Store" task sequence step returns information which can optionally be stored in task sequence variables. The following parameters are the names of the variables where the output will be stored. In all cases, if the variable provided does not exist it will be automatically created for you:
  - **Result –** contains the result of the OneSite - Create State Store task sequence execution. If the step is successful, the value of this variable will be set to "OK." If the step fails, the value will contain an error message related to the failure.
  - **ID –** if the step is successful, this variable will optionally contain the allocation ID of the state store. This value will be a GUID, e.g. deeb05f7-e86c-4d1a-813d-30ebffee7b20.
  - **Share name –** if the step is successful, this variable will optionally contain the name of one of the SMB shares that is created by the OneSite - Create State Store task sequence step. The value stored in this variable will be the name of the share that is most desirable at the time, based on factors such as chassis type (desktop is more desirable than laptop), operating system (servers are more desirable than workstations), etc.

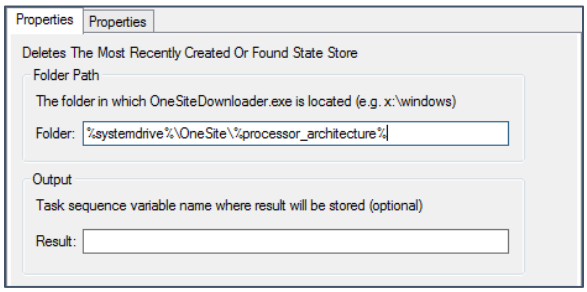
- **UNC path –** if the step is successful, this variable will optionally contain the full UNC path to one of the peers that will host the state store copies associated with this task sequence. This string can be used to access the state store.
- **Hosts –** if the step is successful, this variable will optionally contain a comma-separated list of the short names of machines hosting state store copies for this task sequence.

OneSite - Delete State Store

Once the user state data has been successfully restored, the state store can be deleted using **the OneSite - Delete State Store** step. Successful execution of this step results in the most recently discovered or allocated virtual SAN allocation being deleted, in which case all data stored in the allocation will be removed and all resources used by the allocation will be released.

Stores that are not explicitly deleted using the OneSite - Delete State Store step can be automatically deleted by means of the Storage Expiration parameter in the **OneSite - Create State Store** step described earlier in this document.

Properties



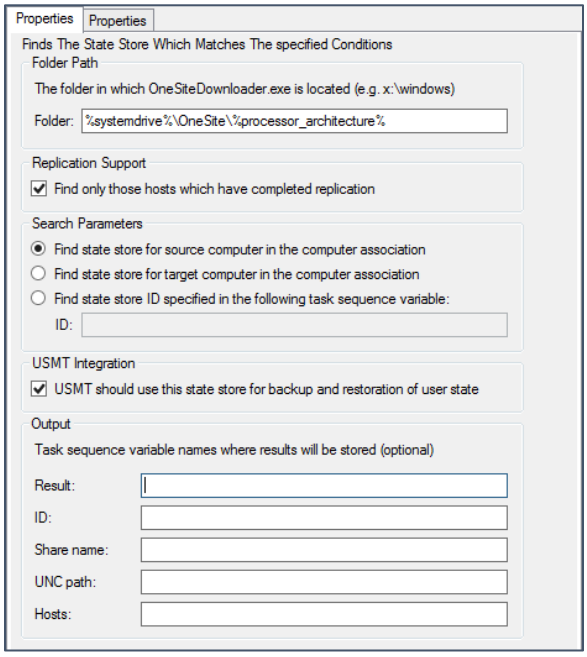
- **OneSiteDownloader Folder Path** - specifies the folder where the OneSiteDownloader.exe file is located. This can be a UNC path, or a local path, and must be available to the environment in which the custom task sequence step is executed. For steps that occur within the Windows PE section of the task sequence, this property specifies the OneSiteDownloader.exe file location in the image, e.g. %systemdrive%\OneSite\%processor_architecture%.
- **Output** - the "OneSite - Delete State Store" task sequence step returns information which can optionally be stored in task sequence variables. If the variable provided does not exist it will be automatically created for you
  - **Result** - contains the result of the OneSite - Delete State Store task sequence execution. If the step is successful, the value of this variable will be set to "OK." If the step fails, the value will contain an error message related to the failure.

> *NOTE: If a result was defined for the "Find State Store" task, this could be referenced as a condition to this task so that the delete operation would only occur if a state store was found. An example would be to set the condition if the Task Sequence Variable: StoreFound equals OK.*

OneSite - Find State Store

Once the migration has taken place, and the target machine is ready to accept the user state data that was saved in the previous task sequence steps, the **OneSite - Find State Store** step is used to locate the stores containing that data. To help ensure that there will be a host available from which to restore state data, this step will attempt WakeOnLAN.

Properties



- **OneSiteDownloader Folder Path** - specifies the folder where the OneSiteDownloader.exe file is located. This can be a UNC path, or a local path, and must be available to the environment in which the custom task sequence step is executed.
- **Find only those hosts which have completed replication** - with this option enabled, the **OneSite - Find State Store** step will not finish until a state store is located that contains a complete copy of the user state replica.

  Default: checked.

- **Search Parameters** – There are three options for this setting, only one of which can be selected:

- **Find state store for source computer in the computer association** - allows the Config Manager computer association for the source computer to be used in discovery of the state store for this task sequence. This is the default Search Parameter option.
- **Find State Store for target computer in the computer association** - allows the Config Manager computer association for the target computer to be used in discovery of the state store for this task sequence.
- **Find State Store ID specified in task sequence variable** - allows a task sequence variable containing the state store GUID to be used for the discovery of the state store for this task sequence.

- **USMT should use this state store for backup and restoration of user state** - this checkbox must be checked to direct USMT to use the Adaptiva store that was created for this task sequence. This is provided as an option so that a single task sequence may be used in multiple scenarios, some of which may not use the Adaptiva OneSite Virtual SMP.

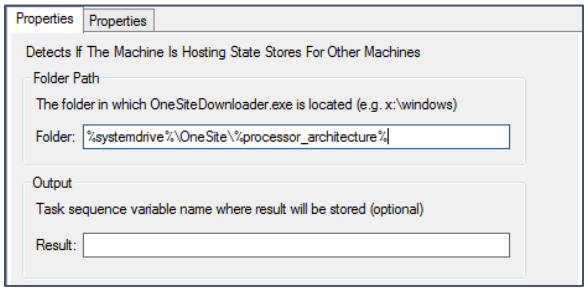
  Default: checked.

- **Output** - the "OneSite - Find State Store" task sequence step returns information which can optionally be stored in task sequence variables. In all cases, if the variable provided does not exist it will be automatically created for you:
  - **Result** - contains the result of the OneSite - Find State Store task sequence execution. If the step is successful, the value of this variable will be set to "OK." If the step fails, the value will contain an error message related to the failure. An example would be to set the Result to StoreFound. This variable can be referenced in another task.
  - **ID** - if the step is successful, this variable will optionally contain the allocation ID of the state store. This value will be a GUID, e.g. deeb05f7-e86c-4d1a-813d-30ebffee7b20.
  - **Share name** - if the step is successful, this variable will optionally contain the SMB share name that was used by the OneSite - Find State Store task sequence step.
  - **UNC path** - if the step is successful, this variable will optionally contain the full UNC path to the share from which the state store associated with this task sequence was restored.
  - **Hosts** - if the step is successful, this variable will optionally contain a comma-separated list of the short names of machines hosting state store copies for this task sequence.

OneSite - Is Machine a Host

Machines serving as hosts of OneSite State Stores should be treated with care, so that the stored data will be protected. The purpose of the "OneSite - Is Machine a Host" step is to provide a method of checking a machine before executing a task sequence which would put a state store at risk.

Properties



- **OneSiteDownloader Folder Path** - specifies the folder where the OneSiteDownloader.exe file is located. This can be a UNC path, or a local path, and must be available to the environment in which the custom task sequence step is executed.
- **Output** - the "OneSite - Is Machine a Host" task sequence step returns information which can optionally be stored in task sequence variables. If the variable provided does not exist it will be automatically created for you:
    - Result - contains the result of the "OneSite - Is Machine a Host" task sequence execution. If the step is successful, the value of this variable will be set to "OK." If the step fails, the value will contain an error message related to the failure.

OneSite - Pin State Store

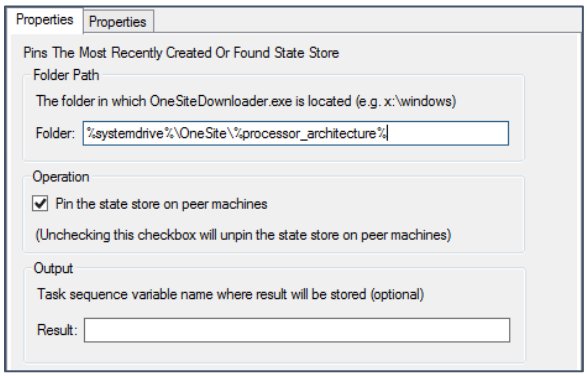
The "OneSite - Pin State Store" task sequence step allows a state store to be pinned or unpinned. Pinned state stores are treated differently from other content by the Adaptiva Cache system, in that they do not get deleted if the host requires more free space. State stores which have not been pinned are subject to automatic deletion by the Adaptiva Cache.

If you selected "Pin the state store on peer machines" in the OneSite – Create State Store task, you do not need to run this step unless you wish to change the setting.

Scope: This task sequence step affects the most recently created or most recently discovered state store.

> ***Caution: Execution of this step with the "Pin the state store on peer machines" option unchecked results in the store becoming unpinned. By default, State Stores are created as "pinned" during the OneSite - Create State Store task sequence step.***

Properties



- **OneSiteDownloader Folder Path** - specifies the folder where the OneSiteDownloader.exe file is located. This can be a UNC path, or a local path, and must be available to the environment in which the custom task sequence step is executed.
- **Pin the state store on peer machines** - specifies whether the state store should be pinned or unpinned. Checking this box will pin the state store, unchecking it will unpin the state store.

  Default: checked.

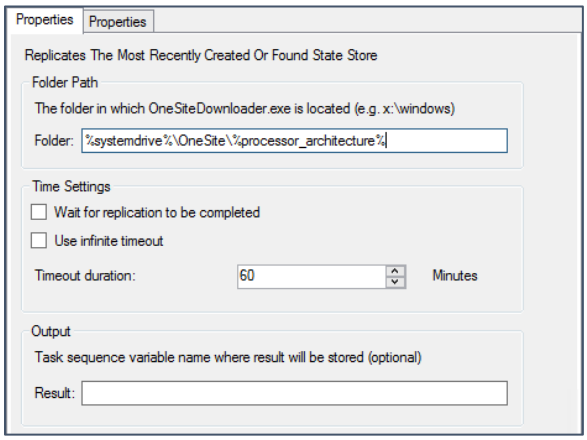
- **Output** - the "OneSite - Pin State Store" task sequence step returns information which can optionally be stored in task sequence variables. If the variable provided does not exist it will be automatically created for you:
  - **Result** - contains the result of the OneSite - Pin State Store task sequence execution. If the step is successful, the value of this variable will be set to "OK." If the step fails, the value will contain an error message related to the failure.

OneSite - Replicate State Store

For the purposes of data protection and availability, it is vitally important that multiple copies of state data be created and stored. If only one copy of a store is created, and then somehow lost prior to completion of migration, there may be no fallback. Additionally, if a store is offline or off the network, a replicated copy can be used for restoring user state.

The number of redundant copies of the state data will be determined by the settings provided in the "OneSite - Create State Store" task described earlier in this document. The "OneSite - Replicate State Store" step is when the redundant copies are created.

Properties



- **OneSiteDownloader Folder Path** - specifies the folder where the OneSiteDownloader.exe file is located. This can be a UNC path, or a local path, and must be available to the environment in which the custom task sequence step is executed.
- **Wait for replication to be completed** - specifies whether the task sequence should continue while the state store is being replicated. Checking this box will cause the OneSite - Replicate State Store task to wait until replication succeeds or fails, or timeout occurs, whichever happens first.

  Default: unchecked.

- **Use infinite timeout** - if this box is checked the OneSite - Replicate State Store task sequence step will wait infinitely for a success or failure result. Unchecking the box enables a timeout value, in minutes, to be entered.

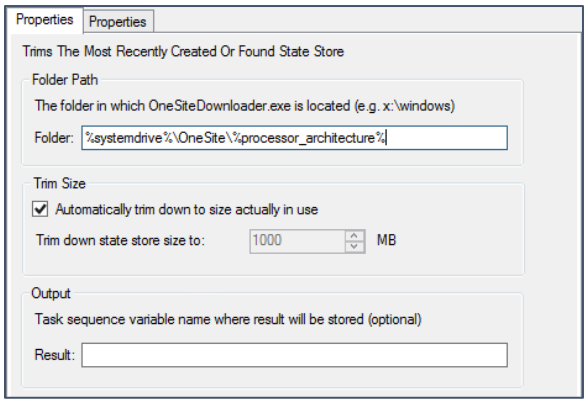
  Range: 5 - 14400; Default: unchecked, 60 minutes.

- **Output** - the "OneSite - Replicate State Store" task sequence step returns information which can optionally be stored in task sequence variables. If the variable provided does not exist it will be automatically created for you:
  - **Result –** contains the result of the OneSite - Replicate State Store task sequence execution. If the step is successful, the value of this variable will be set to "OK." If the step fails, the value will contain an error message related to the failure.

OneSite - Trim State Store

When the "OneSite - Create State Store" task sequence step is added to a task sequence, a value is provided specifying the amount of space required. Once the state data has been copied in to the allocated space, the store can be trimmed to either a specified size, or to the amount of space that is being used by the data.

Properties



- **OneSiteDownloader Folder Path** - specifies the folder where the OneSiteDownloader.exe file is located. This can be a UNC path, or a local path, and must be available to the environment in which the custom task sequence step is executed.
- **Automatically trim down to size actually in use** - specifies that the OneSite - Trim State Store task sequence step should trim the size of the stores to the size that is being used by the user's data and settings.

  Default: checked

- **Trim down state store size to** - if the "Automatically trim down to size actually in use" box is unchecked a value can be entered, in megabytes, for the size to which the store should be trimmed.

  Range: 10 - 102400 MB.

- **Output** - the "OneSite - Trim State Store" task sequence step returns information which can optionally be stored in task sequence variables. If the variable provided does not exist it will be automatically created for you:

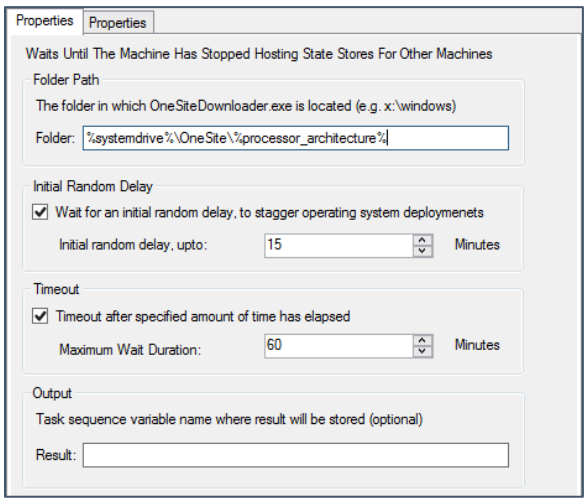
- **Result** - contains the result of the OneSite - Trim State Store task sequence execution. If the step is successful, the value of this variable will be set to "OK." If the step fails, the value will contain an error message related to the failure.

OneSite - Wait Until Not a Host

The "OneSite - Wait Until Not A Host" step provides a method of suspending a task sequence until the machine is no longer hosting a OneSite State Store. Similar to the, "OneSite - Is Machine a Host" task sequence step, the purpose of this step is to protect user state data from unintended deletion.

Once the machine has stopped hosting state stores for peers on the network, a random delay can be specified to "stagger" the start of the next OS migration on the network.

Properties



- **OneSiteDownloader Folder Path** - specifies the folder where the OneSiteDownloader.exe file is located. This can be a UNC path, or a local path, and must be available to the environment in which the custom task sequence step is executed.
- **Initial Random Delay** - this option allows a random delay to be specified, along with the maximum length of the delay, in minutes.

  Range: 1 - 1440; Default: enabled with a maximum 15-minute delay.

- **Timeout** - this option allows a maximum time to be specified, in minutes, for the step to complete. If the option is enabled and the timeout value is reached before completion of the step, the step will fail.

  Range: 1 - 1440: Default: enabled with a 60-minute timeout.

- **Output** - the "OneSite - Wait Until Machine Not A Host" task sequence step returns information which can optionally be stored in task sequence variables. If the variable provided does not exist it will be automatically created for you:
    - **Result** - contains the result of the "OneSite - Wait Until Machine Not A Host" task sequence execution. If the step is successful, the value of this variable will be set to "OK." If the step fails, the value will contain an error message related to the failure.

OneSiteDownloader and vSMP

One or more Adaptiva OneSite clients must be present on the subnet where OneSiteDownloader is used, to respond to requests from the tool.

Arguments are provided as values only (no argument name is required). If an argument value will not be provided, then a semicolon placeholder must still be included for that argument.

Internal Task Sequence Variables

Three task sequence variables are used by OneSiteDownloader in a Virtual SMP environment. If OneSiteDownloader is executed from a command line outside of a task sequence, environment variables with the same names are used and must be pre-populated.

Variable	Description and Usage
AdaptivaSiteCodes	(Optional) If present, this variable contains a comma-separated list of one or more site codes which is included with all messages sent by OneSiteDownloader. A receiving client receiving a OneSiteDownloader message containing this attribute will only reply if it is running a Config Manager client assigned to one of the specified site codes.
AdaptivaVsID	(Required) Contains the allocation id of the most recently allocated or discovered virtual SAN allocation. The value is automatically set by the -vsmpCreate and -vsmpFind operations. It is automatically read by all the other operations whenever needed.
AdaptivaVsHosts	(Required) Contains marshaled host information for all the hosts that will participate in the most recently allocated or discovered virtual SAN allocation. The value is automatically set by the -vsmpCreate and -vsmpFind operations. It is automatically read by all the other operations whenever needed. The value is formatted as three semicolon-separated strings: • **Host client ids** - string; comma-separated list of client ids of all hosts which are participating in this allocation.

	• **Host machine names** - string; comma-separated list of FQDN machine names of all hosts which are participating in this allocation. • **Host MAC addresses** – String; comma-separated list of MAC addresses of all hosts which are participating in this allocation, formatted as: E0-06-E6-25-3A-2F.

Command Line Options

The following table describes command line options for OneSiteDownloader which are applicable to Virtual SMP functions.

Option	Description
-vsmpCreate	The vsmpCreate operation broadcasts a message to the local subnet requesting space allocation that meets the requirements of the options provided. If at least the minimum number of peers responds, an allocation ID is stored in the AdaptivaVsID variable. **Syntax:** OneSiteDownloader.exe -vsmpCreate <parameters> **Arguments:** the -vsmpCreate option accepts a single parameter, which is a semicolon-delimited string of 21 values. Some values are required, some are optional. A placeholder for each of the values listed, including the optional values, must be present in the string. If a value is not provided the default will be used. **SpaceRequired** - required integer between 10 & 1024000; Default: 1000MB **MinHosts** – required; integer between 1 & 100; Default: 2 **MaxHosts** – required; integer between 1 & 100; Default: 2 **Pin** – required; true or false; Default: true **USMTUse** – required; true or false; Default: true **StorageExpire** –required; integer between 60 & 100800; Default: 10080 minutes **DisuseExpire** – required; integer between 60 & 100800; Default: 10080 minutes

	Timeout – required; integer between 5 & 14400; Default: 60 minutes **Publish** – required; source, target, or task sequence variable containing a GUID. **Hidden** – required; true or false; Default: true **Prefix** – required; string value **NetwkAccess** - required; true or false; Default: true **ReadAccts** – optional; comma-separated list of accounts in domain\user format **ReadSIDs** – optional; comma-separated list of SIDs **WriteAccts** – optiona; comma-separated list of accounts in domain\user format **WriteSID** – optional; comma-separated list of SIDs **CreateResult** – optional; string value listing the name of the task sequence variable where the step result will be stored **CreateID** – optional; string value listing the name of the task sequence variable where the store id will be stored **CreateShare** – optional; string value listing the name of the task sequence variable where the store share name will be stored **CreateUNC** – optional; string value listing the name of the task sequence variable where the store share UNC path will be stored **CreateHosts** – optional; string value listing the name of the task sequence variable where the machines that will be host stored **Example:** `OneSiteDownloader.exe -vsmpCreate 1000;2;5;true;true;10080;60;60;source;true;;true;MyDomain\MyUser;; MyDomain\MyUser;;CreateResult;CreateID;CreateShare;CreateUNC;CreateHosts`
-vsmpPin	The vsmpPin operation causes a "Pinned" property to be set on the state store. If this property is set the Adaptiva cache driver will not delete this store from the cache when space is needed. **Syntax:** OneSiteDownloader.exe -vsmpPin <parameters>

	Arguments: The -vsmpPin option accepts a single string parameter, which is a semicolon-delimited string of 2 values. Some values are required, some are optional. A placeholder for each of the values listed, including the optional values, must be present in the string. If a value is not provided, the default will be used. **Example:** `OneSiteDownloader.exe -vsmpPin true;PinResult`
-vsmpReplicate	The vsmpReplicate operation is responsible for replicating the state store to retain multiple copies. **Syntax:** OneSiteDownloader.exe -vsmpReplicate <parameters> **Example:** `OneSiteDownloader.exe -vsmpReplicate true;60;ReplicateResult`

-vsmpTrim	The –vsmpTrim operation trims the state store to a specified value or the actual size of the state store.
	
	Syntax: OneSiteDownloader.exe -vsmpTrim <parameters>
	
	Examples:
	
	`OneSiteDownloader.exe -vsmpTrim 1000;TrimResult`
	`OneSiteDownloader.exe -vsmpTrim 0;TrimResult`

-vsmpFind	The –vsmpFind operation is used to locate state stores for the system. **Syntax:** OneSiteDownloader.exe -vsmpFind <parameters> **Example:** `OneSiteDownloader.exe -vsmpFind true;source;true;FindResult;FindID;FindShare;FindUNC;FindHosts`
-vsmpDelete	The –vsmpDelete operation is responsible for deleting state stores for the system. **Syntax:** OneSiteDownloader.exe -vsmpDelete <parameters> **Example:** `OneSiteDownloader.exe -vsmpDelete DeleteResult`

-vsmpIsMachineAHost	The –vsmpIsMachineAHost operation is used to check if the machine is currently hosting state stores. **Syntax:** OneSiteDownloader.exe -vsmpIsMachineAHost <parameters> **Example:** `OneSiteDownloader.exe -vsmpIsMachineAHost IsMachineResult`
-vsmpWaitUntilNotAHost	The –vsmpWaitUntilNotAHost operation will cause the task sequence to wait until the machine is not a host prior to proceeding. **Syntax:** OneSiteDownloader.exe -vsmpWaitUntilNotAHost <parameters> **Example:** `OneSiteDownloader.exe -vsmpWaitUntilNotAHost 15;60;WaitResult`

ConfigMgr State Migration References

Useful links to reference material:

Task Sequence Steps in Configuration Manager

Introduction to operation system deployment in System Center Configuration Manager

User State Migration Tool (USMT) Technical Reference

Introduction to the Data that USMT Migrates

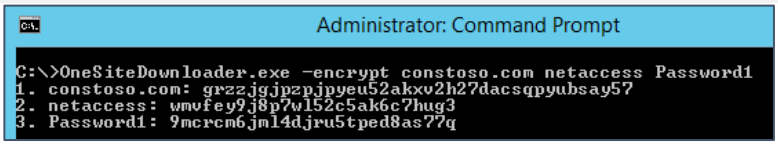
Appendix A: ConfigMgr 2012/Current Branch OneSiteDownloader Task Sequence Variables

The primary role of OneSiteDownloader in ConfigMgr 2012 is to allow packages referenced by a Task Sequence to be acquired using the OneSite peer-to-peer ACP rather than a DP. In ConfigMgr 2012, beginning with SCCM 2012 CU1, Microsoft added the ability to set a Task Sequence Variable that specifies a tool for downloading referenced content. This capability was added in KB2744420, which requires SCCM 2012 CU1. SCCM 2012 CU2 and later versions include this capability without the need of a KB hotfix.

The following table lists all of the Task Sequence variables that can be used by OneSiteDownloader.

Variable	Description and Usage
SMSTSDownloadProgram	This variable should be applied at the beginning of a task sequence and after every system restart. If you are using a path which contains spaces or will result in using "Program Files", you must surround the value with quotes (") **Value:** The full path in which the OneSiteDownloader.exe is located. **Example in this Guidance:** `%systemdrive%\OneSite\%processor_architecture%\OneSiteDownloader.exe`
OneSiteNoFallback	In some environments it is necessary to ensure that content will never be downloaded from a DP. If this variable has been defined and set to "true", and the OneSiteDownloader tool fails to download content, then the task sequence will not fall back to a DP, and will fail. By default, OneSiteDownloader will always fall back to a DP in case content download fails due to any reason. **Values:** true or false
OneSiteRandomize	When OneSiteDownloader requests content, there will typically be multiple potential sources offered from the peer network. Adding the "OneSiteRandomize" task sequence variable and setting it to "true"

	randomizes the order of the sources that are returned, for load balancing purposes. By default, OneSiteDownloader will not randomize sources. **Values:** true or false
OneSiteDiscoveryTimeOut	When OneSiteDownloader performs content discovery, it waits for 600 seconds and then fails if no response is received from the peer to peer coordinator machine (the Rendezvous Point, or RVP). By setting this variable, this behavior can be overridden. **Value:** <number of seconds before timeout>
OneSiteEnforceMax	When OneSiteDownloader performs content discovery and content is discovered in the client's local subnet, only those sources are returned even if only 1 source is found. To force OneSiteDownloader to always return the maximum (3) sources, set a task sequence variable to "true". This forces OneSiteDownloader to continue discovery until 3 sources are found within the office. **Values:** true or false
OneSiteUsePreferred	Enforces OneSiteDownloader to look for preferred sources. In the case where the local subnet has one or more preferred sources, they are returned along with other non-preferred sources up to 3. In case the local subnet does not have any preferred sources, discovery is spanned into remote subnets and discovery terminates on discovery first preferred source. **Values:** true or false
OneSiteUseSMSTSLogPath	Causes OneSiteDownloader to create its log file, OneSiteDownloader.log in the _SMSTSLogPath folder. **Values:** true or false
AdaptivaServerGUIDs	If defined, only responses from RVPs belonging to these Adaptiva sservers are accepted. This is beneficial when migrating from one

	SCCM version to another version in which there may be Adaptiva clients on the same subnet which report to different Adaptiva servers. To find the Adaptiva server GUID, check the registry of a client which reports to the desired Adaptiva server: `HKLM\Software\[Wow6432Node\Adaptiva\Client client_data_manager.server_guid` **Value:** <Adaptiva Server GUID>
OneSiteNacDomain **OneSiteNacAccount** **OneSiteNacPassword**	Used in the case where OneSiteDownloader should use the ConfigMgr Network Access Account for authentication. The three variables contain encryption keys which provide the domain, username, and password to OneSiteDownloader. These variables should be set at the beginning of the task sequence. This is supported in OneSiteDownloader version 4.0.614.1 and above. To generate the appropriate values for these variables: 1. Gather the domain, username, and password for the Network Access Account. 2. On a standard Windows client, open a command prompt from a folder which contains the 4.0.614.1 version of the OneSiteDownloader utility use the following command: **Example:** `OneSiteDownloader.exe -encrypt <NACDomain> <NACUsername> <NACPassword>`  The utility will output three encryption keys for the domain name, username and password and the task sequence variables should be set to these values respectively.
OneSiteServerNameOrIP	Used in the case where OneSiteDownloader is being invoked on a subnet which does not support broadcasts, such as Wi-Fi. OneSiteDownloader will attempt to communicate with the server

	directly to acquire the RVP list. Once the RVPList is received from the server, OneSiteDownloader uses a unicast protocol to communicate with the local RVP instead of broadcasting. Also, this option can be used in the scenario in which there are no other Adaptiva clients in the same subnet, but there must be another Adaptiva client in the same office. **Value:** <Adaptiva Server FQDN> or <Adaptiva Server IP>

adaptiva

Appendix B: ConfigMgr 2007 OneSiteDownloader Command Line Options

Unlike in ConfigMgr 2012, ConfigMgr 2007 is unable to invoke OneSiteDownloader using Task Sequence variables. In 2007 OneSiteDownloader is invoked using command line parameters which can be used in a Run Command Line task in an SCCM 2007 Task Sequence, or from a command prompt.

The following table lists the options and parameters available for OneSiteDownloader.exe. Only one of these options may be specified at a time, although the tsref or ref options contains parameters which may be combined.

Option	Usage	Description
Tsref Option and Parameters		
-tsref	OneSiteDownloader.exe -tsref [concurrency count] [fallback] [nodownload]	The tsref option resolves all packages and images referenced in a Task Sequence to their local SMB paths, allowing them to be downloaded locally using OneSite P2P download instead of pulling them down from a DP over the WAN.
-tsref <concurrency count>	OneSiteDownloader.exe –tsref 20	The concurrency count parameter is optional and manages the asynchronous behavior of OneSiteDownloader while performing a large number of reference resolutions. By default, 10 references are resolved concurrently. The concurrency count may be increased to make resolutions faster, up to a maximum allowed value of 50. A smaller concurrency count may be specified to slow down content searches, e.g. in cases where a large number of machines are being imaged simultaneously.

-tsref fallback	OneSiteDownloader.exe –tsref fallback	The fallback parameter is an optional setting. If specified, it allows the task sequence to fall back to DPs if OneSite clients are not online.
-tsref nodownload	OneSiteDownloader.exe –tsref nodownload	The nodownload parameter is an optional parameter which prevents content from being downloaded from remote sources. Instead it simply skips contents for which local sources are not discovered. If specified, only those sources should be returned for which Adaptiva content is already available locally. If only packed content is available, then unpacking should take place, but no download across WAN should occur.
-tsref random	OneSiteDownloader.exe –tsref random	When OneSiteDownloader requests content, there will typically be multiple potential sources offered from the peer network. The random parameter is an optional parameter which randomizes the order of the sources that are returned, for load balancing purposes.
-tsref fail	OneSiteDownloader.exe -tsref fail	If OneSiteDownloader fails to resolve any content sources, a non-zero error code will be returned: 1812 ERROR_RESOURCE_DATA_NOT_FOUND Supported in version 5.5.657 and above.
Ref Option		
-ref	-ref [concurrency count] [fallback] [nodownload] <PKGID> < PKGID> ... **Example:**	The -ref option is the same as the -tsref option, but replaces only those references whose SCCM content id's have been specified.

	`OneSiteDownloader.exe -ref 20 fallback PRI00234 PRI00256`	
Other Options (Must be called individually)		
-sccm	-sccm <PKGID> <target folder> **Example:** `OneSiteDownloader.exe –sccm PRI00234 C:\LocalDownload` *NOTE: Only one package can be called per command.*	The sccm option enables downloading of any SCCM Package, Deployment Package or Image to a local folder on the machine using Adaptiva OneSite P2P download. If the requested package has been pre-staged to the local office or has previously been downloaded by any client, it is immediately downloaded from a local peer using Adaptiva OneSite P2P. If the package has never been downloaded to that location before, the package is downloaded over the WAN using Adaptiva OneSite P2P and WAN bandwidth management, cached on other local clients, and then becomes available for copying using Adaptiva OneSite P2P discovery. This can be used from inside Windows PE through a recovery console, through a "Run Command Line" Task in a Task Sequence or directly in all production versions of Windows from XP SP2 onwards, whether or not an Adaptiva OneSite client is installed on that machine.
-adaptiva	-adaptiva <Adaptiva content ID> <target folder> **Example:** `OneSiteDownloader -adaptiva Adaptiva$WAIK "C:\LocalDownloadFolder"`	The -adaptiva option is used to download Adaptiva content on demand, such as Workflows, Policy objects or WAIK Tools. Typically, Adaptiva content is self-downloading and you should never need to download manually. This

		option works just like the -sccm option and is provided for completeness. If you are using this command regularly, please contact Adaptiva so we may automate your use case through an enhancement to the OneSite product.
-get	-get <TaskSequenceVariableName> **Example:** `OneSiteDownloader.exe -get _SMSTSMachineName`	The -get option displays the value of the specified Task Sequence variable. This command can only be used when a Task Sequence is currently executing on the machine. The get command can be used independently of the Task Sequence. The get command doesn't have to be running as a task within the Task Sequence, but a Task Sequence must be running at the time the command is executed. For example, you may open a recovery console in Windows PE using the F8 function key, and then use this command to monitor Task Sequence variables as the Task Sequence progresses.
-set	-set <TaskSequenceVariableName> <TaskSequenceVariableValue> **Example:** `OneSiteDownloader.exe -set OSDCOMPUTERNAME MYPCNAME`	The -set option sets the value of the specified Task Sequence variable or adds a new Task Sequence variable with that name if it doesn't already exist. This command can only be used when a Task Sequence is currently executing on the machine. The set command can be used independently of the Task Sequence. The set command doesn't have to be running as a Task within the Task Sequence, but a Task Sequence must be running at the time the command is executed.

		For example, you may open a recovery console in Windows PE using the F8 function key, and then use this command to modify Task Sequence variables as the Task Sequence progresses.
-list	-list **Example:** `OneSiteDownloader.exe -list`	The -list option displays the names and values of all Task Sequence variables in the currently executing Task Sequence. This command can only be used when a Task Sequence is currently executing on the machine. The list option can be used independently of the Task Sequence. It doesn't have to be running as a Task within the Task Sequence, but a Task Sequence must be running at the time the command is executed. For example, you may open a recovery console in Windows PE using the F8 function key, and then use this command to monitor Task Sequence variables as the Task Sequence progresses.
-logfolder	-logfolder <UNC path of remote share> <domainname> <username> <password> **Example:** `OneSiteDownloader.exe -logfolder \\srv\Logshare company.com svcacct password`	The -logfolder option overrides the default location of the log file generated by the OneSiteDownloader tool. A UNC path containing the location of the folder share where the log file is to be created must be specified, along with the domain name, user logon name and password of an account that has write permissions for that share. This redirects logging to the specified folder share, using a log file that bears a randomly generated file name, e.g.

		{2B837C23-D3ED-44a6-8A83-1E11F97CE635}.LOG
-locallogfolder	-locallogfolder <local path of folder> **Example:** `OneSiteDownloader.exe –locallogfolder C:\Windows\Logs`	The -locallogfolder option causes the OneSiteDownloader log file to be generated in the specified folder. The folder must already exist.