Realtime
publishers

Automating Windows 7 Installation for Desktop and VDI Environments

Greg Shields

Copyright Statement

Chapter 7: Creating a Complete Solution for Automating Windows 7 Installation

You should by now recognize this book's pattern of introducing new technologies and approaches into the Windows deployment story. That story involves plenty of layers, each of which builds on the infrastructure of the layer below. It started in Chapter 1 with the very basics of deploying images—and weren't those basics, indeed! The images back then weren't even customized. They were little more than the default configuration you get by Next, Next, Finishing your way through a manual installation.

The next chapters then led you through greater customization and introduced additional automations. By now, you've learned how to create single images that deploy everywhere. You've automated the installation of drivers, applications, and user state information. You've also discovered the MDT, and how its Task Sequences are the glue that ties your automations together.

Throughout this book, I've shown you how Microsoft's free solutions can accomplish necessary deployment tasks. Those free tools get you pretty far. Using them, you can kick off a relatively-automated Windows installation with little effort and good results.

Yet even with substantial automation in place, there's always an underlying desire to *fully automate everything*. A complete solution for automating Windows 7 installation should take you out of the picture entirely. That solution should be able to upgrade or refresh computers without needing to touch them at all. This topic, the zero-touch approach, is the final layer that creates your complete solution. Just like each of the previous chapters, getting there requires leaning on the infrastructure you've built up to this point.

There is, however, one downside: Taking that final step requires a new piece of software that isn't free. Getting to the complete solution, at least with today's options, requires assistance from Microsoft's System Center Configuration Manager (or ConfigMgr for short).

ConfigMgr isn't free, but it is valuable in ways that go beyond desktop deployment. If this book's exploration of automations for Windows *installation* excites you, then ConfigMgr's many automations for Windows *management* will excite you even more. It's a fantastic administrative tool with a long history, bringing much to the table in addition to OS deployment.

Before it can deploy operating systems (OSs), ConfigMgr must be installed. Its agents must also be distributed to your desktops. Doing so requires an exercise in design work and more than a few initial configurations. Those initial steps by themselves are a big topic. So in the interests of space, this chapter will assume you've already completed the setup activities. If you need assistance, numerous books and guides exist that'll get you started. As I begin this chapter, I'll assume you've already completed the installation of ConfigMgr and have deployed its agents to the computers on your network.

> **Resource**
>
> ConfigMgr's initial installation and agent deployment can be exceedingly complex activities. If you're looking for a little help, Alberto Ortega has written an excellent blog post that I use when building new ConfigMgr servers on Windows Server 2008 R2. Found at http://blogs.southworks.net/aortega/2009/09/16/deploy-sccm-2007-sp2-rc-on-windows-server-2008-r2/, this post does an excellent job of documenting the exact steps that will complete ConfigMgr's initial installation and agent distribution.

Stepping Back: Understanding LTI, ZTI, and UDI

Before we get started, let's spend a minute understanding why ConfigMgr is necessary for full automation. In short, it's the ConfigMgr agent that's needed. Microsoft's acronym-filled MDT documentation refers to three deployment approaches. These they refer to as LTI, ZTI, and UDI. These three acronyms reference Microsoft's *Light-Touch Installation*, *Zero-Touch Installation*, and *User-Driven Installation* approaches.

MDT alone supports only the LTI approach. That's because the "Light" in Light-Touch Installation refers to the fact that some of its activities must be accomplished *at the desktop*. You've seen this in previous chapters where someone at the desktop is needed to launch a PXE boot or run the LiteTouch.vbs script.

But this chapter's goal is full automation. Getting there means not having to physically be present at any desktop for an installation. That said, even if it's not you in person, kicking off that installation requires *something* to exist at the desktop. That's why ConfigMgr, and specifically its agent, is a requirement. The ConfigMgr agent is *that something* at the desktop. Being installed there, the agent is perfectly positioned to facilitate an OS upgrade or refresh, resulting in a ZTI.

> **Note**
>
> Although I won't be discussing it in this chapter, ConfigMgr is similarly well-positioned to handle the extra steps necessary for users to initiate their own deployment. This approach represents Microsoft's UDI, and is an added capability you can layer on top of what you learn in this chapter.

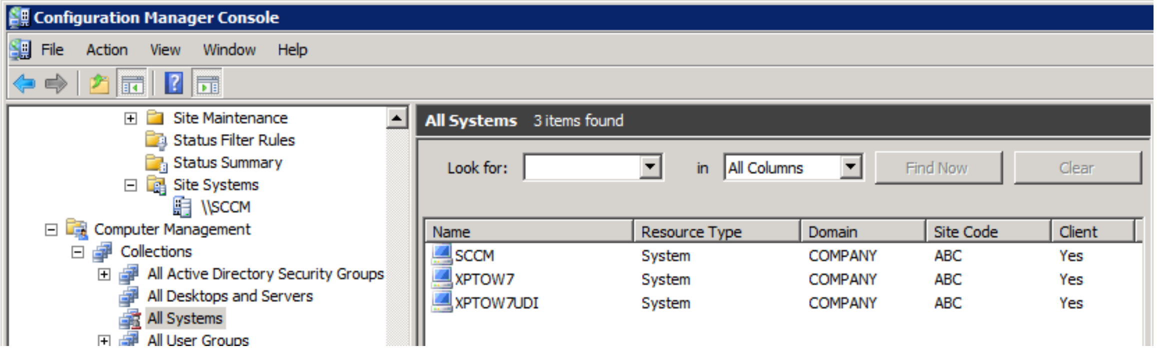
For the ZTI installation approach to work, that ConfigMgr agent must be present at every desktop. As you can imagine, this means that you won't be using ZTI for fresh installs. A fresh install starts with an empty hard drive, which means no ConfigMgr agent is present on the machine. Thus, this chapter will focus on the processes used to upgrade an existing machine (one that already contains an agent) as well as refreshing one with a clean OS.

Step Fourteen: ZTI with ConfigMgr

As I mentioned, getting to complete automation with Microsoft's tools requires the assistance of a ConfigMgr agent. Among its other tasks, this agent must be present to receive the signal from the ConfigMgr server that an installation package has been advertised and is ready for distribution.

I've built my ConfigMgr server on a new computer named \\sccm. Although I'm using a different computer here for ConfigMgr, be aware that there are no technical limitations that prevent you from collocating ConfigMgr with WDS and MDT. If you do intend these services to coexist, make sure to use the full version of SQL Server and not SQL Server Express for all your deployment services. That's because ConfigMgr requires a full version of SQL Server to function.

Upon completing the initial installation of ConfigMgr, my server shows a set of systems with installed and functioning agents. I know this because the Client column in Figure 7.1 shows Yes for each of the computers shown.



All Systems 3 items found

Look for: in All Columns

Name	Resource Type	Domain	Site Code	Client
SCCM	System	COMPANY	ABC	Yes
XPTOW7	System	COMPANY	ABC	Yes
XPTOW7UDI	System	COMPANY	ABC	Yes

Figure 7.1: Three computers, each with a ConfigMgr agent.

A few configurations must be set before you start deploying OSs. These configurations go above and beyond the typical ConfigMgr installation. They integrate ConfigMgr with your existing MDT infrastructure. They add a State Migration Point, which will be used in storing user state information during upgrades. Finally, they create a folder and file share ConfigMgr will use to store package information.

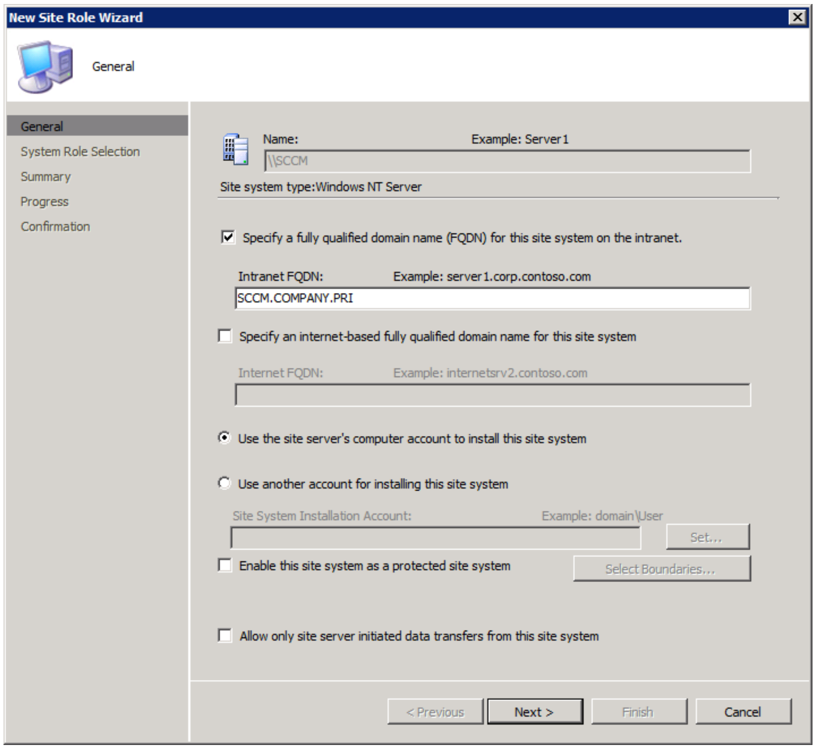
Integrating ConfigMgr with MDT

First up is integrating your ConfigMgr console with MDT. Start by installing the ConfigMgr console to your MDT server. The console is available on the ConfigMgr media by clicking the Install Configuration Manager 2007 SP2 link. Once launched, choose to Install or upgrade an administrator console.

Without launching the console after its installation, select the Configure ConfigMgr Integration link under Microsoft Deployment Toolkit in the Start menu. You'll be prompted for a Site server name and Site code. These will correspond to your ConfigMgr site and site server. Now you can launch the ConfigMgr console, and right-click Computer Management | Operating System Deployment | Task Sequences. If you see a selection titled Create Microsoft Deployment Task Sequence, you've got a successful integration.

Adding the State Migration Point

Your next task is to add the State Migration Point role to the ConfigMgr server. Navigate to Site Management | <siteCode> | Site Settings | Site System, then right-click the link for your ConfigMgr server, and choose New Roles. You'll be greeted with a screen that's similar to Figure 7.2.



Name: Example: Server1
\\SCCM

Site system type:Windows NT Server

☑ Specify a fully qualified domain name (FQDN) for this site system on the intranet.

Intranet FQDN: Example: server1.corp.contoso.com
SCCM.COMPANY.PRI

☐ Specify an internet-based fully qualified domain name for this site system

Internet FQDN: Example: internetsrv2.contoso.com

● Use the site server's computer account to install this site system

○ Use another account for installing this site system

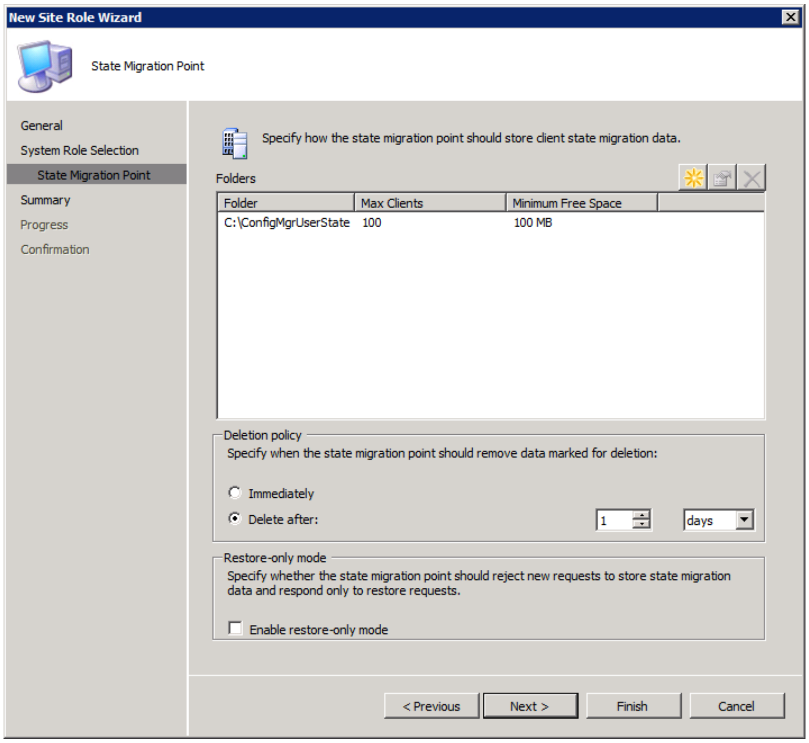
Site System Installation Account: Example: domain\User
Set...

☐ Enable this site system as a protected site system
Select Boundaries...

☐ Allow only site server initiated data transfers from this site system

Figure 7.2: New Site Role Wizard.

This first role assignment is to the ConfigMgr server itself, so you need only click Next to continue. At the next screen (not shown) choose to add the State migration point role, and click Next. The screen that follows (see Figure 7.3) sets the location for storing user state information as well as settings for the deletion policy and whether to enable restore-only mode. Click the upper-right yellow star, and enter a folder path to your storage location. As you can see in Figure 7.3, I've chosen *C:\ConfigMgrUserState* as my path, and limited the number of clients and minimum free space. Click through the wizard's remaining pages to finish the installation.



Specify how the state migration point should store client state migration data.

Folders

Folder	Max Clients	Minimum Free Space
C:\ConfigMgrUserState	100	100 MB

Deletion policy

Specify when the state migration point should remove data marked for deletion:

○ Immediately

● Delete after: 1 days

Restore-only mode

Specify whether the state migration point should reject new requests to store state migration data and respond only to restore requests.

□ Enable restore-only mode

Figure 7.3: Configuring the state migration point.

Creating an OS Deployment Share

The next preparation step is easy. OS deployment with ConfigMgr requires a folder and associated share to store package information. On my server, I created a folder called C:\OSDPackages and shared it as \\sccm\OSDPackages with appropriate permissions. Do the same on your server, as you'll be adding data to subfolders of that share in a moment.

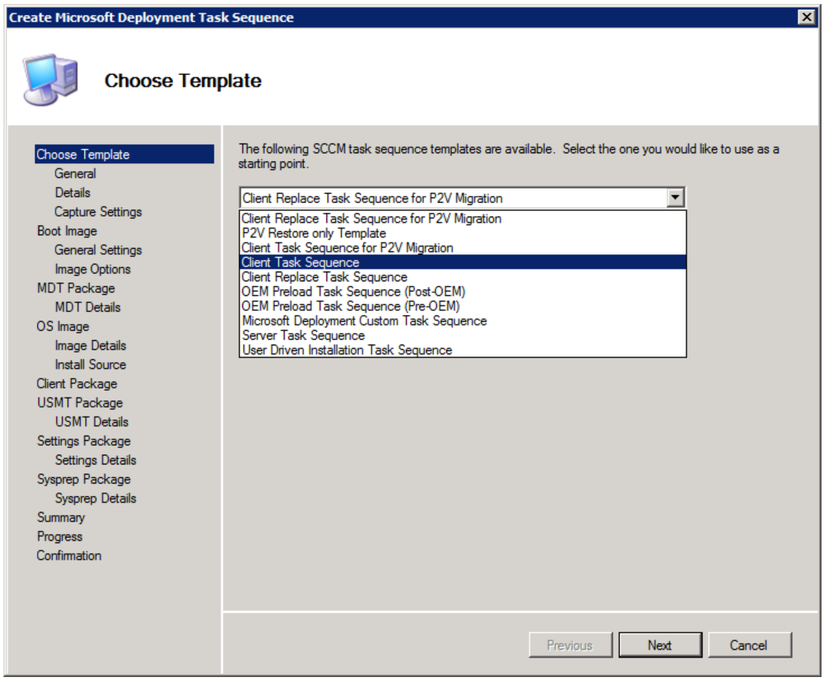
Creating a Deployment Task Sequence in ConfigMgr

Most IT professionals don't realize that WDS and MDT aren't even necessary to deploy OSs with ConfigMgr. That said, there's a reason this book delays ConfigMgr until its second-to-last chapter: *Getting here requires understanding the underlying infrastructure first.* You've developed that understanding through the previous chapters. You now know how WinPE functions. You've also created at least one image that can be deployed. You've gathered a list of drivers and applications that need installation with the deployment, and you're familiar with the complexities of user state migration.

By first knowing each of these tasks, your efforts in working with ConfigMgr are eased because ConfigMgr itself is a complex application. It needs to be, in part because of how it can scale from just a few computers to tens of thousands. Having built your MDT and other infrastructures before now enables you to leverage that experience within ConfigMgr. In fact, thanks to the MDT-to-ConfigMgr integration you just completed, the Create Microsoft Deployment Task Sequence menu item is about to become your best friend.

Right-click Task Sequences in ConfigMgr, and select Create Microsoft Deployment Task Sequence. What appears is a lengthy wizard that requires numerous settings to complete. The first time you run this wizard, you'll be creating many of the items it asks for in its pages. You'll be able to reuse many of those items during subsequent uses of the wizard.

You can consider the Client Task Sequence template (see Figure 7.4) to be the core template for many ZTI deployments. This template will scan for applications and user state, offload user state information, rebuild the computer, reinstall applications, and ultimately replace user state information onto the upgraded or refreshed computer. Select this template in this screen.


Create Microsoft Deployment Task Sequence

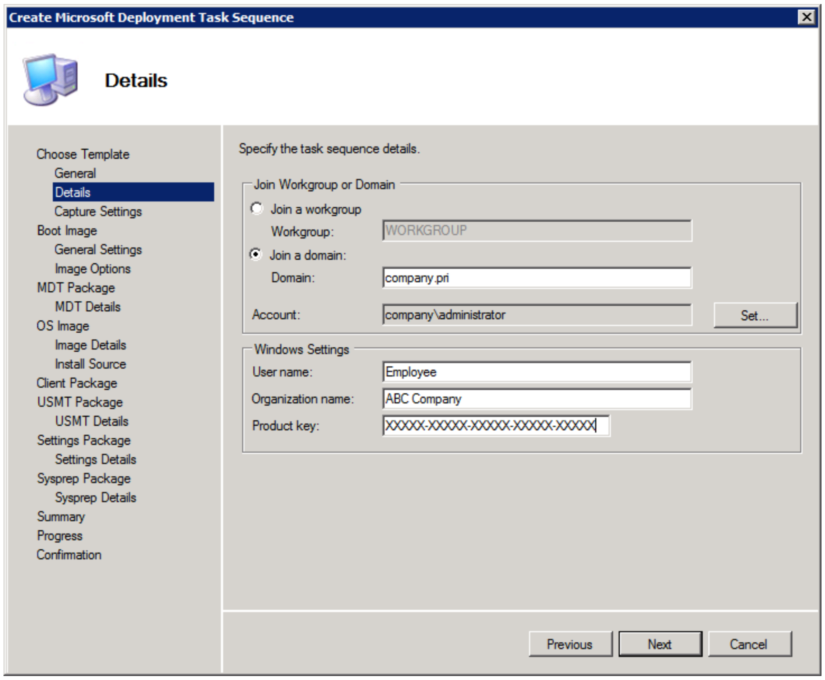
Choose Template

The following SCCM task sequence templates are available. Select the one you would like to use as a starting point.

Client Replace Task Sequence for P2V Migration

Figure 7.4: Choosing a template.

In the next screen, (not shown) enter a Task Sequence name and any comments. Then fill out the Details page with information about your domain and Windows settings (see Figure 7.5). If your organization uses volume license keys for desktops, you can enter that license key into the provided blank.



Create Microsoft Deployment Task Sequence

Details

Specify the task sequence details.

Join Workgroup or Domain

○ Join a workgroup

Workgroup: WORKGROUP

● Join a domain:

Domain: company.pri

Account: company\administrator

Windows Settings

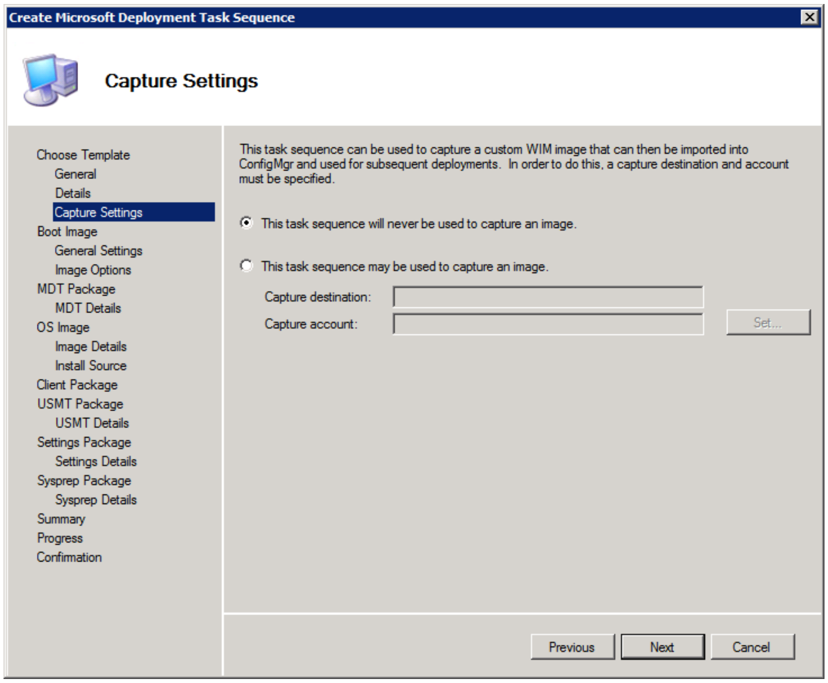
User name: Employee

Organization name: ABC Company

Product key: XXXXX-XXXXX-XXXXX-XXXXX-XXXXX

Figure 7.5: Providing template details.

Figure 7.6 defines how the Task Sequence might be used. A ConfigMgr Task Sequence can be used for deploying images or deploying and subsequently capturing an image. We already have an image to use that was captured back in an earlier chapter, so we won't need to use this image for a later capture. Thus, configure the screen as you see it in Figure 7.6.



Create Microsoft Deployment Task Sequence

Capture Settings

This task sequence can be used to capture a custom WIM image that can then be imported into ConfigMgr and used for subsequent deployments. In order to do this, a capture destination and account must be specified.

● This task sequence will never be used to capture an image.

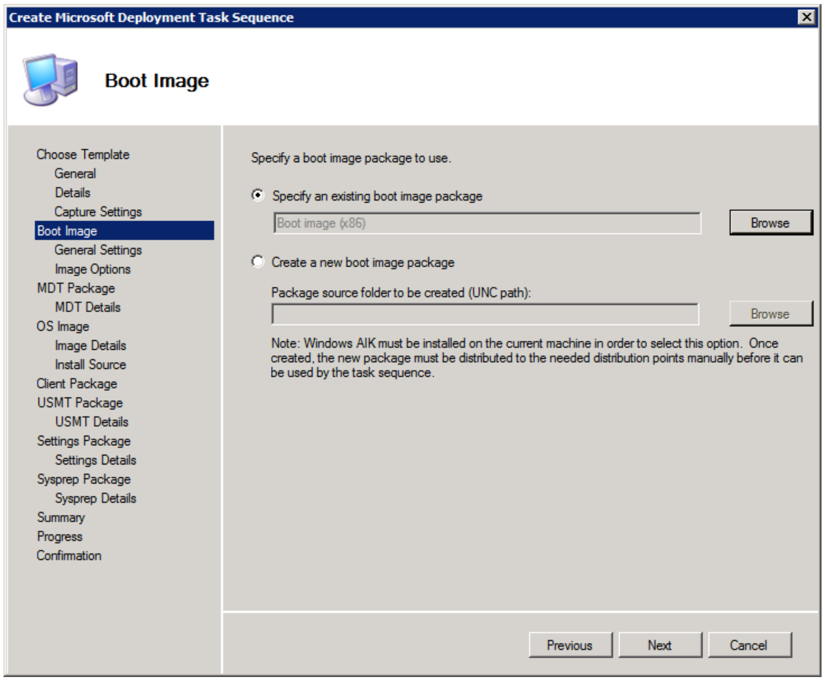
○ This task sequence may be used to capture an image.

Capture destination:

Capture account:

Figure 7.6: Specifying template capture settings.

You'll find Microsoft's wizard verbiage to get a little confusing in the next series of screens. The first is seen in Figure 7.7 where the wizard asks for a boot image. Know that the boot image used by ConfigMgr is slightly different than the one used by WDS and MDT during previous examples.



Specify a boot image package to use.

● Specify an existing boot image package

Boot image (x86)

○ Create a new boot image package

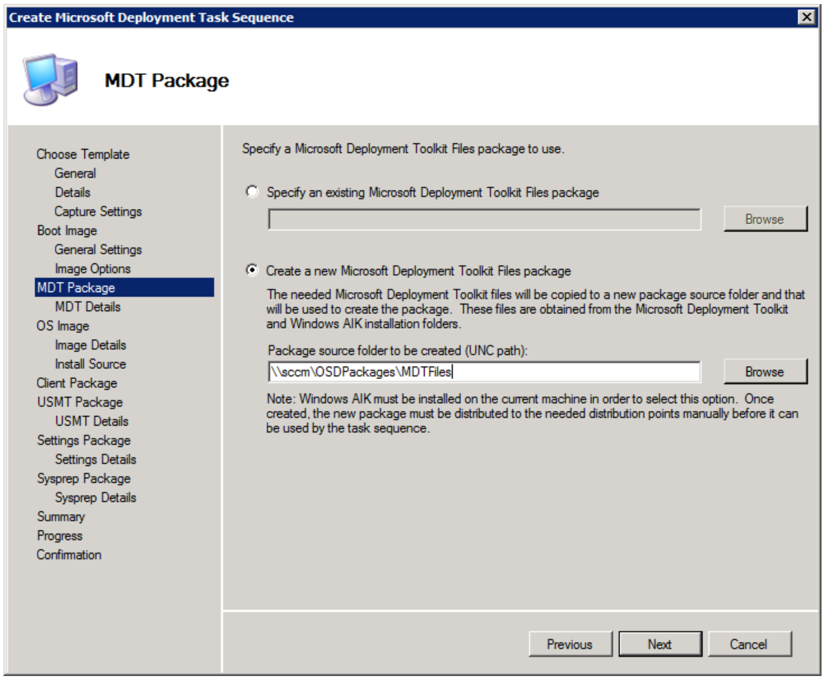
Package source folder to be created (UNC path):

Note: Windows AIK must be installed on the current machine in order to select this option. Once created, the new package must be distributed to the needed distribution points manually before it can be used by the task sequence.

Figure 7.7: Specifying a boot image.

This alternate boot image is configured with extra code that facilitates its integration with ConfigMgr. Thus, you won't be using the same boot image you used in previous chapters. Here, select the *Specify an existing boot image package* option, and click Browse. Two ConfigMgr boot images should be automatically available, one each for x86 and x64. In this example, I'll use the x86 boot image.

Another point of confusion with this wizard is in its next screen (see Figure 7.8) where MDT files package information is required. This package is a collection of scripts and other data from the MDT that is used by ConfigMgr for deploying an OS. This is a new ConfigMgr instance as well as the first execution of the Microsoft Deployment Task Sequence, so the MDT files package does not exist. Thus, it must be created. To do so, select the second radio button and provide a UNC path to a subfolder of your choice within the OS deployment package share created earlier in this chapter. For my environment, I'll use \\sccm\OSDPackages\MDTFiles.



MDT Package

Specify a Microsoft Deployment Toolkit Files package to use.

○ Specify an existing Microsoft Deployment Toolkit Files package

● Create a new Microsoft Deployment Toolkit Files package

The needed Microsoft Deployment Toolkit files will be copied to a new package source folder and that will be used to create the package. These files are obtained from the Microsoft Deployment Toolkit and Windows AIK installation folders.

Package source folder to be created (UNC path): \\sccm\OSDPackages\MDTFiles

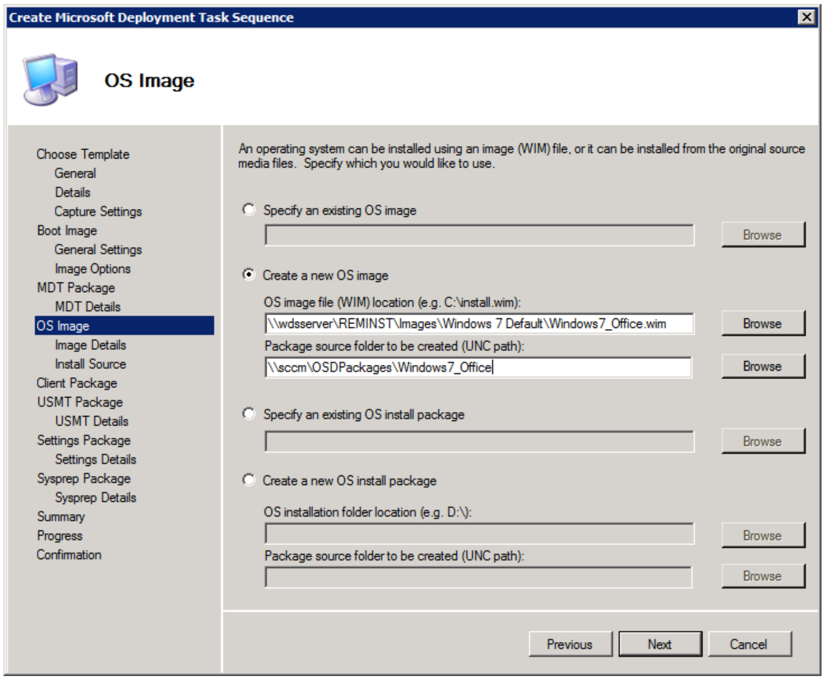
Note: Windows AIK must be installed on the current machine in order to select this option. Once created, the new package must be distributed to the needed distribution points manually before it can be used by the task sequence.

Figure 7.8: Providing an MDT package.

This location will be populated with information from MDT after the wizard is complete and the Task Sequence is generated. In later Task Sequences, you may simply reference an existing ConfigMgr package through the top radio button.

In the following screen (not shown) you will be asked for name, version, language, manufacturer, and comment information for the package that will eventually be created that includes this data. A name is required at minimum; however, the other information will be useful down the road for identifying the characteristics of this package. In my environment I'll name the package *MDTFiles*.

One of the biggest benefits of layering ConfigMgr atop an existing MDT instance is that you've already built the OS image you intend to deploy. You'll see multiple options in Figure 7.9 for creating or specifying the OS image to be deployed. Choosing the first option will select an image that has already been converted into a ConfigMgr package.



OS Image

An operating system can be installed using an image (WIM) file, or it can be installed from the original source media files. Specify which you would like to use.

○ Specify an existing OS image

● Create a new OS image

OS image file (WIM) location (e.g. C:\install.wim): \\wdsserver\REMINST\Images\Windows 7 Default\Windows7_Office.wim

Package source folder to be created (UNC path): \\sccm\OSDPackages\Windows7_Office

○ Specify an existing OS install package

○ Create a new OS install package

OS installation folder location (e.g. D:\):

Package source folder to be created (UNC path):

Figure 7.9: Specifying an OS image.

You haven't created that package yet, so choose to *Create a new OS image*. In doing so, enter the path to the image's WIM file that you created in a previous chapter. You'll see in Figure 7.10 that I have selected Windows7_Office.wim, which is the WIM file created earlier that currently resides on the WDS server. I'm also providing a path to new subfolder of the ConfigMgr OS deployment package share. Doing this copies the WIM file to the new location once the wizard is complete. It at the same time reconfigures the WIM file into a ConfigMgr package for deployment.

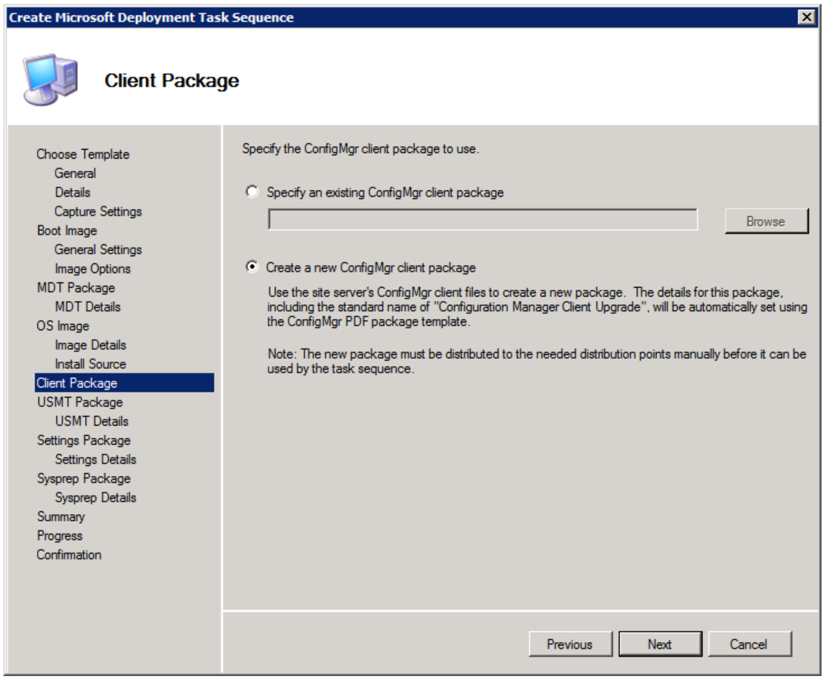
In the next screen (not shown), you'll be asked for name, version, and comment information. As with the MDT package, enter as much detail as possible in these blanks to assist with later identification.

Note

During later uses of this tool, you can select to *Specify an existing OS image* instead. This will direct you to the list of packages currently installed on the ConfigMgr server.

In order for ConfigMgr to manage the client after deployment, it will need to add a ConfigMgr agent to the image as the OS is deployed. That ConfigMgr agent is part of its own package. (Notice how everything that is to be deployed in ConfigMgr must be encapsulated into a ConfigMgr package?) As this is a fresh installation of ConfigMgr, that Client Package does not yet exist.

If yours does, you may choose the top radio button option in Figure 7.10, click Browse, and locate the Client Package you've already created. If yours does not, select the second radio button, and click Next to continue.



Specify the ConfigMgr client package to use.

○ Specify an existing ConfigMgr client package

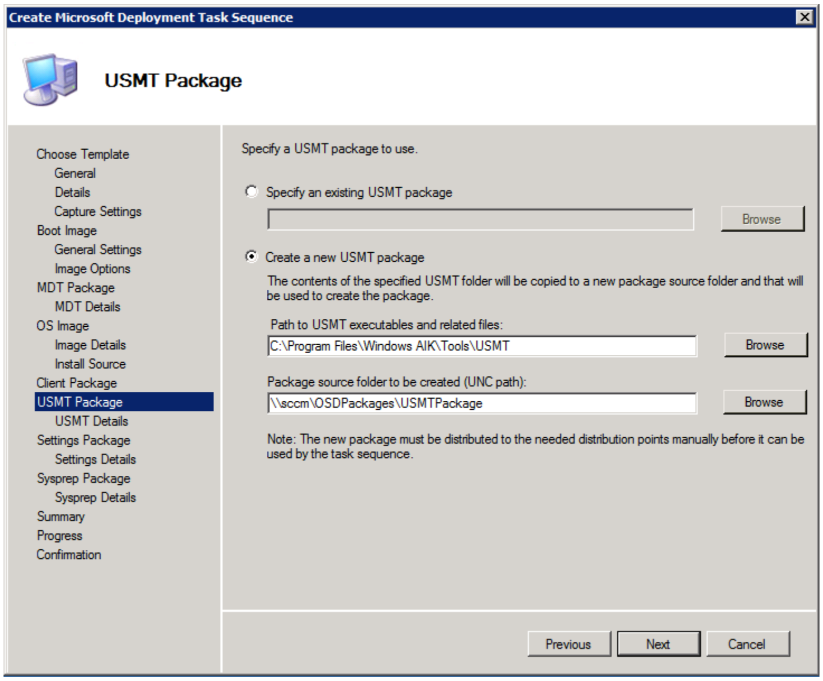
● Create a new ConfigMgr client package

Use the site server's ConfigMgr client files to create a new package. The details for this package, including the standard name of "Configuration Manager Client Upgrade", will be automatically set using the ConfigMgr PDF package template.

Note: The new package must be distributed to the needed distribution points manually before it can be used by the task sequence.

Figure 7.10: Determining which client package to use.

As with the other packages so far, a USMT package must also be created. This USMT package collects the USMT code from the location in the upper path box and copies it to the path in the lower box (see Figure 7.11). The upper path should be automatically filled in for you as this location is supplied from the WAIK. As with the other packages, the lower path will be a subfolder of your choice within the OS deployment package share. The screen that follows (not shown) will ask for the usual package characteristics: name, version, language, manufacturer, and comments.



Specify a USMT package to use.

○ Specify an existing USMT package

● Create a new USMT package

The contents of the specified USMT folder will be copied to a new package source folder and that will be used to create the package.

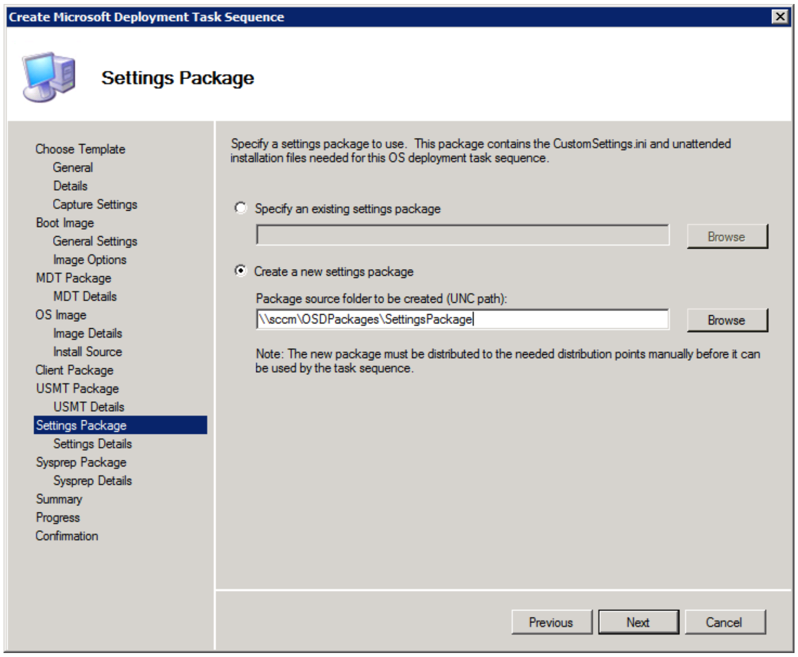
Path to USMT executables and related files: C:\Program Files\Windows AIK\Tools\USMT

Package source folder to be created (UNC path): \\sccm\OSDPackages\USMTPackage

Note: The new package must be distributed to the needed distribution points manually before it can be used by the task sequence.

Figure 7.11: Specifying a USMT package.

One final package is required for Task Sequences that deploy Windows 7 computers. This package, called the Settings Package, includes a set of configurable settings that define how the OS is deployed as well as the behavior of the installation. Choose at this time to create a new one, as one has not yet been created (see Figure 7.12). Give it a subfolder path to your OS deployment share, and (not shown) provide name, version, language, manufacturer, and comment information.



Specify a settings package to use. This package contains the CustomSettings.ini and unattended installation files needed for this OS deployment task sequence.

○ Specify an existing settings package

● Create a new settings package

Package source folder to be created (UNC path): \\sccm\OSDPackages\SettingsPackage

Note: The new package must be distributed to the needed distribution points manually before it can be used by the task sequence.

Figure 7.12: Creating a settings package.

You may click through the remaining screens, choosing the defaults to complete the Microsoft Deployment Task Sequence creation. Notice that no Sysprep package is required because this Task Sequence deploys Windows 7, an OS that natively includes the needed Sysprep code.

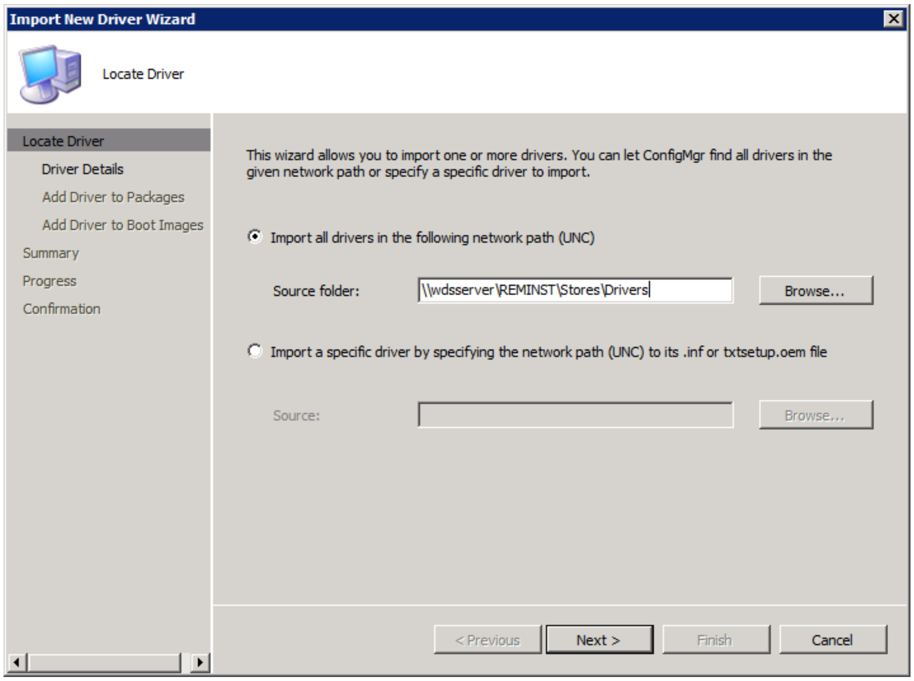
Once you've completed the wizard, the process to create the Task Sequence will take a period of time. This process transfers a large quantity of data from various locations to your OS deployment share.

> **Note**
>
> Once the creation is complete, spend a minute reviewing the contents of the OS deployment share. You'll find that a number of subfolders have been created, with some containing information that will be used during the installation.
>
> This data collection is stored in file format so that it can be modified or otherwise customized to change the behaviors of the installation or the OS that is delivered. Although this topic is out of scope for this chapter, know that you can edit any of the configurations found in this share to modify the behavior of the installation. As you'll discover later, any changes here must be uploaded to a distribution point to be used.

Importing Drivers

ConfigMgr handles driver injection using the same plug-and-play approach seen with WDS and MDT. Thus, the driver store you created during the previous chapters can be automatically imported into ConfigMgr. Do this now by right-clicking the Drivers node, and choosing Import.

If you've been following along throughout this guide, you probably have a folder of drivers on your WDS server in a subfolder of the C:\RemoteInstall folder. Mine is at\\wdsserver\REMINST\Stores\Drivers. You can see in Figure 7.13 that the wizard can interrogate drivers within a stated path and its subfolders to automatically import those drivers into ConfigMgr.



This wizard allows you to import one or more drivers. You can let ConfigMgr find all drivers in the given network path or specify a specific driver to import.

● Import all drivers in the following network path (UNC)

Source folder: \\wdsserver\REMINST\Stores\Drivers

○ Import a specific driver by specifying the network path (UNC) to its .inf or txtsetup.oem file

Source:

Figure 7.13: Importing drivers.

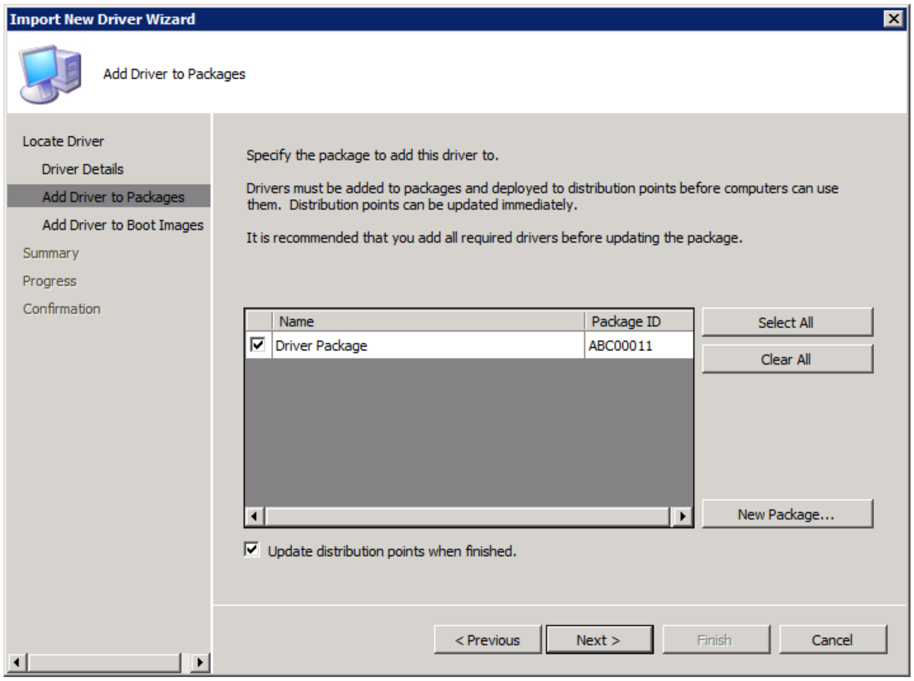
Once imported, drivers can be added (not shown) to categories for filtering. This prevents similar-looking drivers from accidentally being installed onto incorrect equipment. It also prevents drivers from one OS from being inadvertently added to another, which can cause conflicts. Be particularly careful with your drivers, as driver mismatches will cause problems with OS deployment.

Note

In fact, a driver problem that warrants additional attention is caused by drivers used for networking. The WinPE instance used by ConfigMgr requires a functioning network driver so that its installation can communicate with the ConfigMgr server to download images.

If you are following along using VMware Workstation, you may have difficulties with the VMXNET driver used by some versions of VMware Workstation. One workaround is to alter the driver used by your Windows XP desktops by adding the line ethernet0.virtualDev = "e1000" to your VM's VMX file. This line forces the virtual machine to use the more-common Intel e1000 NIC driver, which must be subsequently downloaded from the Internet and installed to the Windows XP desktop.

Drivers must be added to at least one driver package and distributed to distribution points before they can be used by computers during deployment. Figure 7.14 shows how the set of drivers specified in Figure 7.13 have been gathered into a package and sent to the ConfigMgr distribution points.



Specify the package to add this driver to.

Drivers must be added to packages and deployed to distribution points before computers can use them. Distribution points can be updated immediately.

It is recommended that you add all required drivers before updating the package.

	Name	Package ID
☑	Driver Package	ABC00011

☑ Update distribution points when finished.

Figure 7.14: Adding drivers to packages.

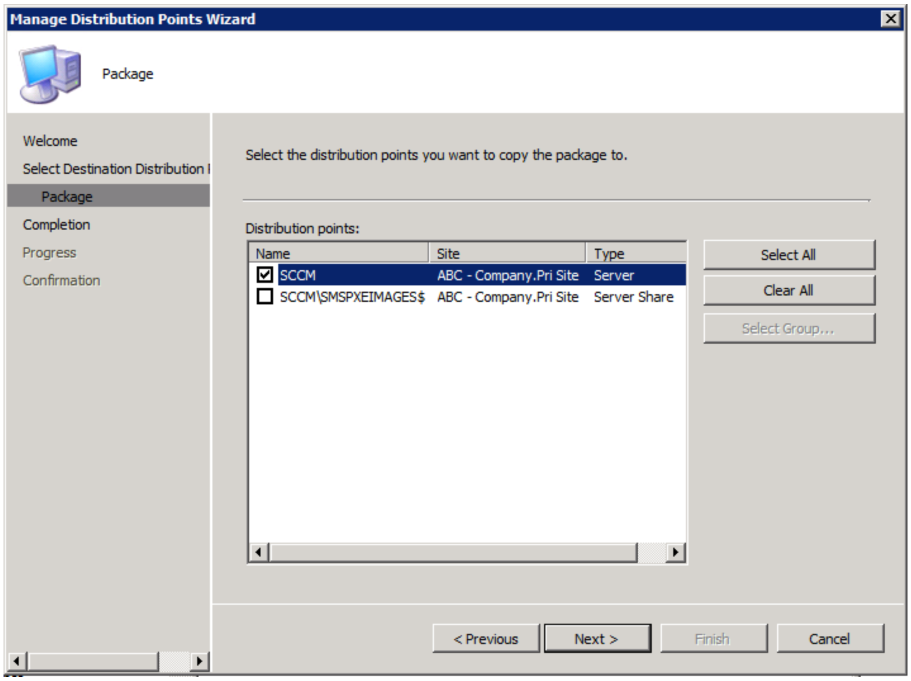
The next screen (not shown) provides a location to add drivers to boot images. If candidate desktops require non-native drivers to boot properly into WinPE, make sure to add them to the correct boot image or the boot and/or deployment process will fail. Click through to finish the wizard.

Updating Distribution Points

At this point, you have the data you need to begin your first deployment. The next step is to add each of the newly-created packages to a ConfigMgr distribution point. This distribution point is different than the file share in which you've been storing data up until now. It is the location agents will query for package data.

Separating your working file share from the "production" distribution point allows you to work with the data in your file share, only updating it to the distribution point when it is ready for deployment. Distribution points are not updated through any of the previous wizards. You'll need to manage them separately. The process to accomplish this task is the same for all packages. I'll show you how to complete this step with one package, then point you towards the series of packages that require attention.

Start by clicking the Packages node. Right-click any of the recently-created packages in the middle pane, and choose Manage Distribution Points. Click Next, then choose to Copy the package to new distribution points. Figure 7.15 shows an example of the distribution points where package data can be uploaded. Select one, and click Next to update that distribution point with the new software.



Select the distribution points you want to copy the package to.

Distribution points:

Name	Site	Type
☑ SCCM	ABC - Company.Pri Site	Server
□ SCCM\SMSPXEIMAGES$	ABC - Company.Pri Site	Server Share

Figure 7.15: Selecting distribution points.

Note

Remember that you can at any point alter the data that is stored within your OS deployment share; however, when you do make changes to that data, the distribution point must always be updated with a new copy of the data. This can be done by selecting the package, and choosing to Update distribution points.

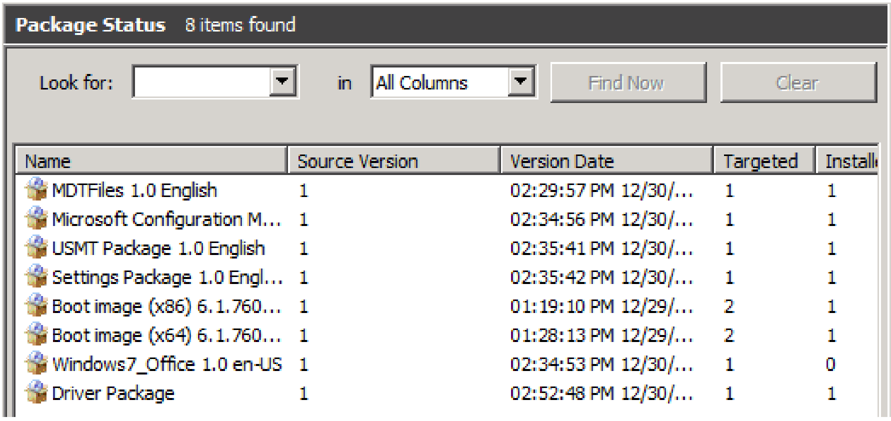
At this point, you can copy the remaining packages to the distribution point. This will be the MDT Files package, Settings package, USMT package, Microsoft Configuration Manager Client Upgrade package, Boot image package, Driver package, and the Windows7_Office Operating System Image package.

Note

Some of these packages are not found under the Packages node. They will be found elsewhere in the Computer Management hierarchy.

One caution is important as you update distribution points with new data. Be aware that the process to copy packages to distribution points takes time. Although smaller packages will be updated almost immediately, larger packages such as your OS images will take much longer. You should not begin advertising packages until they have completed their installation to distribution points.

Figure 7.16 shows how to verify installation status by navigating to System Status | Package Status. Notice here that the Windows7_Office 1.0 en-US package has been targeted to one distribution point but has not completed its installation there. This Package Status view will help you ensure that ConfigMgr has completed its transfer of data to the distribution points prior to creating an advertisement.



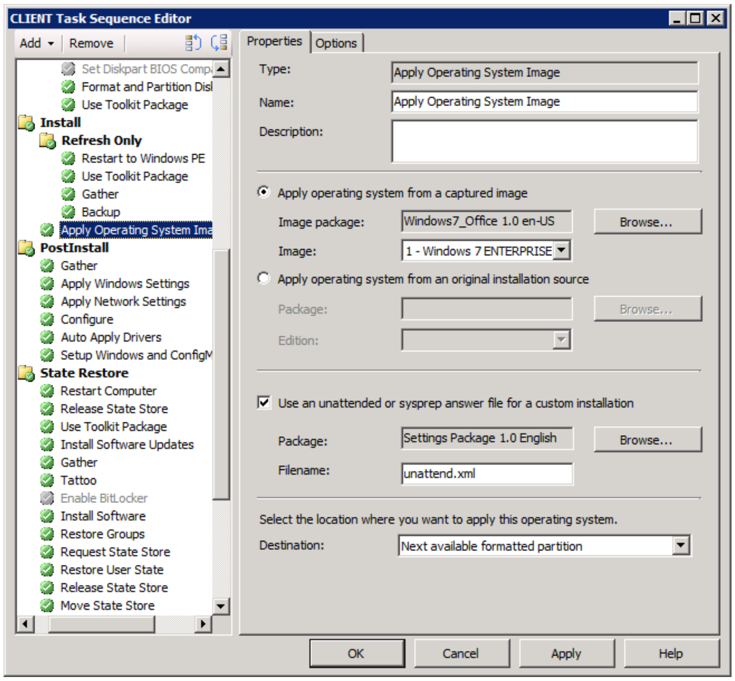
Name	Source Version	Version Date	Targeted	Install
MDTFiles 1.0 English	1	02:29:57 PM 12/30/...	1	1
Microsoft Configuration M...	1	02:34:56 PM 12/30/...	1	1
USMT Package 1.0 English	1	02:35:41 PM 12/30/...	1	1
Settings Package 1.0 Engl...	1	02:35:42 PM 12/30/...	1	1
Boot image (x86) 6.1.760...	1	01:19:10 PM 12/29/...	2	1
Boot image (x64) 6.1.760...	1	01:28:13 PM 12/29/...	2	1
Windows7_Office 1.0 en-US	1	02:34:53 PM 12/30/...	1	0
Driver Package	1	02:52:48 PM 12/30/...	1	1

Figure 7.16: Package status.

Viewing the Task Sequence and Creating the OS Deployment Advertisement

You’re nearly ready for deployment. Prior to creating the advertisement that announces the availability of the OS upgrade, you might want to review the Task Sequence created in the earlier step. Navigate to Task Sequences, right-click, and choose to Edit the Task Sequence you just created.

You’ll immediately see the incredible number of steps that are automatically generated as part of the Task Sequence. One of those steps, the Apply Operating System Image step, is shown in Figure 7.17. In this step, the captured image is selected to be deployed as part of the Task Sequence.



- Set Diskpart BIOS Comp
- Format and Partition Disl
- Use Toolkit Package
- **Install**
  - **Refresh Only**
    - Restart to Windows PE
    - Use Toolkit Package
    - Gather
    - Backup
  - Apply Operating System Ima
- **PostInstall**
  - Gather
  - Apply Windows Settings
  - Apply Network Settings
  - Configure
  - Auto Apply Drivers
  - Setup Windows and ConfigM
- **State Restore**
  - Restart Computer
  - Release State Store
  - Use Toolkit Package
  - Install Software Updates
  - Gather
  - Tattoo
  - Enable BitLocker
  - Install Software
  - Restore Groups
  - Request State Store
  - Restore User State
  - Release State Store
  - Move State Store

Type: Apply Operating System Image

Name: Apply Operating System Image

Description:

● Apply operating system from a captured image

Image package: Windows7_Office 1.0 en-US

Image: 1 - Windows 7 ENTERPRISE

○ Apply operating system from an original installation source

Package:

Edition:

☑ Use an unattended or sysprep answer file for a custom installation

Package: Settings Package 1.0 English

Filename: unattend.xml

Select the location where you want to apply this operating system.

Destination: Next available formatted partition

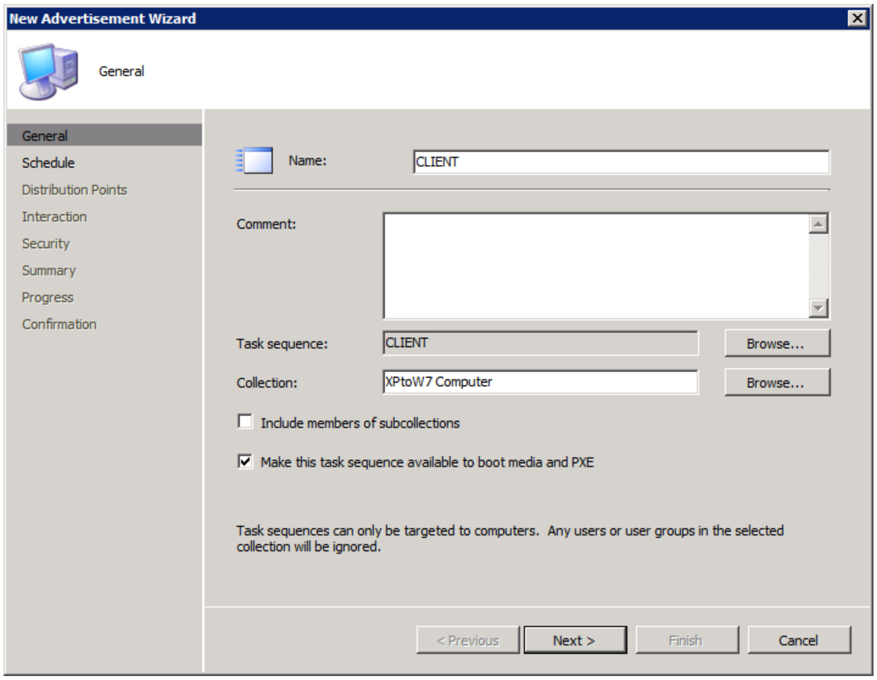
Figure 7.17: Task Sequence.

You can at this point adjust any of the characteristics of this Task Sequence as well as add steps by clicking the upper-left Add button. By default, this Task Sequence will automatically install the selected image to a targeted computer after gathering user state information. It will restore that user state information to the new computer once the installation is complete. Considering the very large number of steps and possible outcomes that exist in this sequence, I'll leave the exploration to you for how you will customize your Task Sequences.

> **Note**
>
> Remember that ConfigMgr is an application installation solution in addition to its job in deploying OSs. Thus, it can install applications on top of managed systems once those applications have been packaged. That packaging process is similar to what was discussed in Chapter 3.
>
> You'll notice that there is a step titled Install Software, which is one mechanism that can be used to install packaged software during an OS deployment.

You're now ready to deploy an OS. Start that process by right-clicking the Task Sequence, and choosing Advertise. Doing so will bring forward the New Advertisement Wizard, similar to Figure 7.18. In this wizard you'll schedule the Task Sequence you want to deploy and connect it to a collection of computers that will be upgraded.

That collection of computers is seen in Figure 7.18. The actual process to create and populate collections is out of scope for this chapter; however, notice that I have created a collection called XPtoW7 Computer that contains only my test computer.



New Advertisement Wizard

General

Name: CLIENT

Comment:

Task sequence: CLIENT

Collection: XPtoW7 Computer

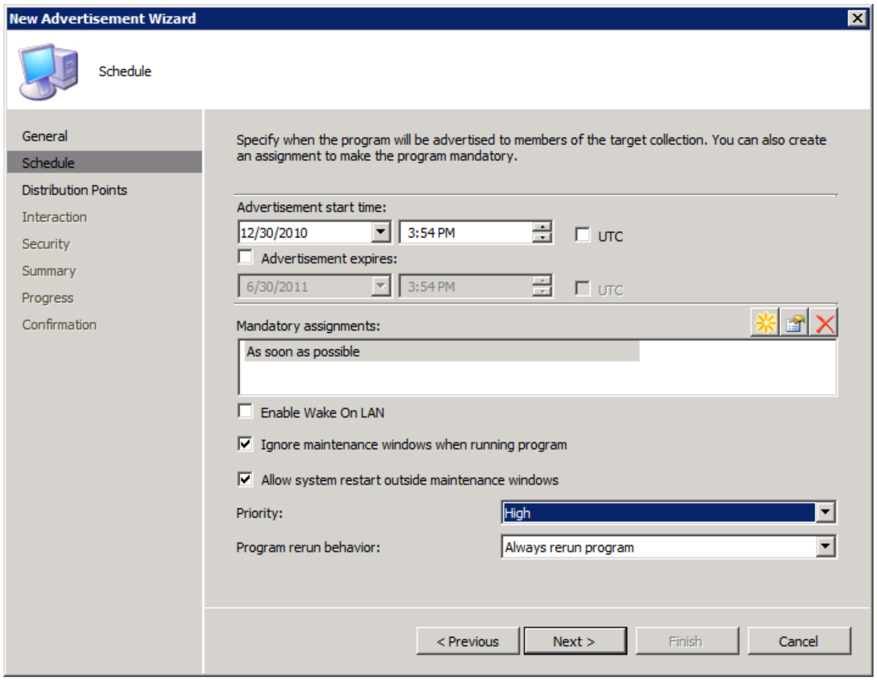
☐ Include members of subcollections

☑ Make this task sequence available to boot media and PXE

Task sequences can only be targeted to computers. Any users or user groups in the selected collection will be ignored.

Figure 7.18: New Advertisement Wizard.

All advertisements run on a schedule. Notice in Figure 7.19 that the advertisement has a start time but is also configured with a mandatory assignment to begin as soon as possible. In ConfigMgr parlance, "as soon as possible" can sometimes involve a large amount of time, so be prepared for a short delay even after the advertisement is created.



New Advertisement Wizard

Schedule

Specify when the program will be advertised to members of the target collection. You can also create an assignment to make the program mandatory.

Advertisement start time: 12/30/2010 3:54 PM ☐ UTC

☐ Advertisement expires: 6/30/2011 3:54 PM ☐ UTC

Mandatory assignments:

As soon as possible

☐ Enable Wake On LAN

☑ Ignore maintenance windows when running program

☑ Allow system restart outside maintenance windows

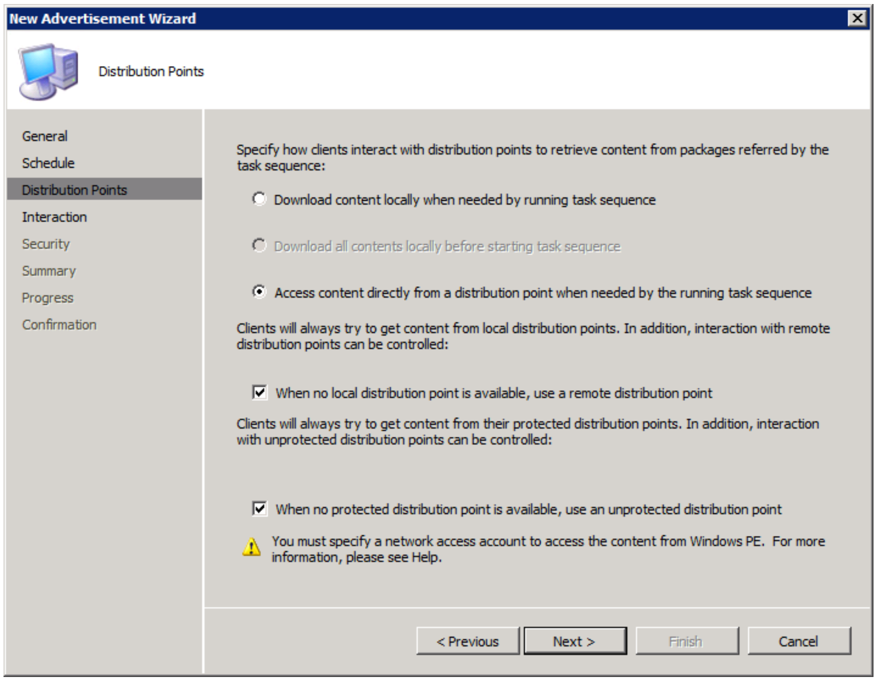
Priority: High

Program rerun behavior: Always rerun program

Figure 7.19: Scheduling the advertisement.

This is a test environment, so you will also want to select the *Ignore maintenance windows when running program* and *Allow system restart outside maintenance windows* check boxes. Lastly, because you may rerun this package later, consider setting the Program rerun behavior to Always rerun program. Click Next to continue.

Next up is the Distribution Points page (see Figure 7.20) with only a few settings. Set the bottom radio button to instruct the client to gather its data directly from a distribution point rather than downloading the content. An OS deployment will wipe a system disk; this setting prevents the situation where downloaded data is deleted by the Task Sequence. You may also select the two bottom check boxes to ensure that any available distribution point is used.



Specify how clients interact with distribution points to retrieve content from packages referred by the task sequence:

○ Download content locally when needed by running task sequence

○ Download all contents locally before starting task sequence

● Access content directly from a distribution point when needed by the running task sequence

Clients will always try to get content from local distribution points. In addition, interaction with remote distribution points can be controlled:

☑ When no local distribution point is available, use a remote distribution point

Clients will always try to get content from their protected distribution points. In addition, interaction with unprotected distribution points can be controlled:

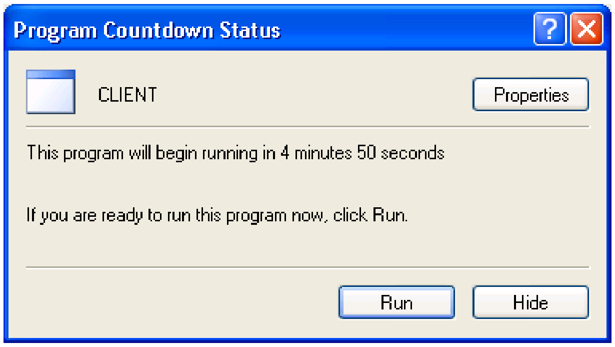
☑ When no protected distribution point is available, use an unprotected distribution point

You must specify a network access account to access the content from Windows PE. For more information, please see Help.

Figure 7.20: Specifying distribution point settings.

The next five screens of the New Advertisement Wizard are not shown, as most deployments use their defaults. Continue through the wizard to create the advertisement.

If you've done everything correctly, a balloon notice will appear on desktops after a few minutes (but sometimes as much as an hour later) after the advertisement is created. Clicking that balloon notice will bring forward the Program Countdown Status message that Figure 7.21 shows. This notice alerts the users that their computers are about to be upgraded, and provides a 5-minute countdown timer prior to beginning the OS deployment.

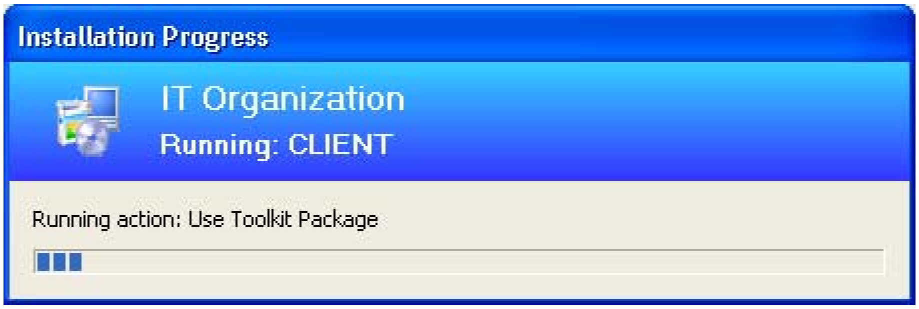


This program will begin running in 4 minutes 50 seconds

If you are ready to run this program now, click Run.

Figure 7.21: The Program Countdown Status balloon.

As with the MDT, a set of pre-installation activities must occur on the client before it boots into WinPE for the actual OS installation. Figure 7.22 shows the Installation Progress bar that shows which steps in the Task Sequence are in progress.

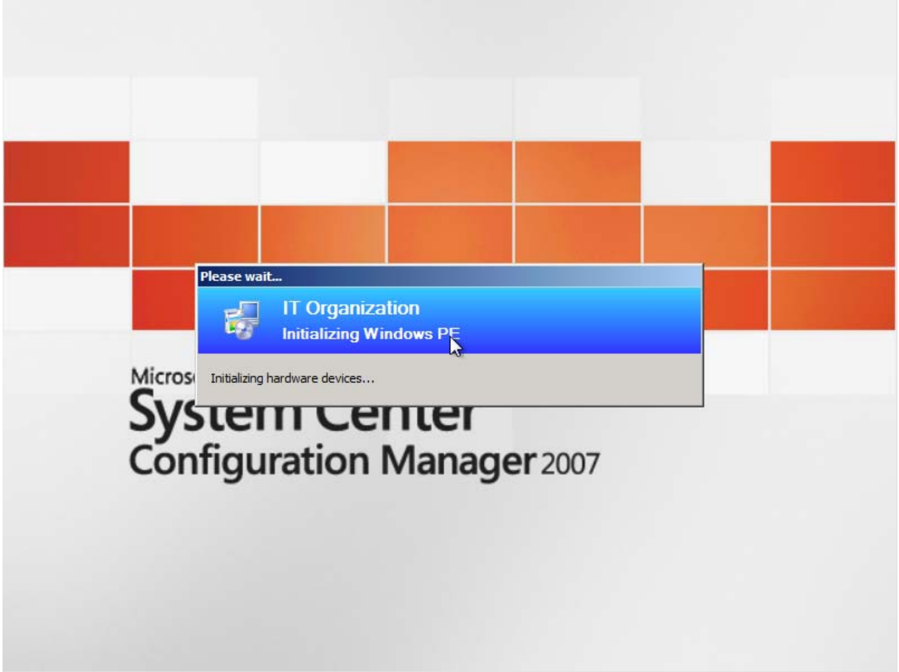


IT Organization
Running: CLIENT

Running action: Use Toolkit Package

Figure 7.22: The Installation Progress bar.

Once the pre-installation activities are complete, the computer will reboot directly into WinPE to begin the installation. ConfigMgr uses a different background screen to denote its version of WinPE. You can see an example of that screen in Figure 7.23. If you've done everything correctly, the OS will install and eventually return control back to the user with the new OS. Viola! You've completed your first zero-touch deployment of an OS!



Please wait...

IT Organization
Initializing Windows PE

Initializing hardware devices...

Figure 7.23: The WinPE Phase.

Stepping Back: What Haven't You Seen? What's Left?

Getting this far with the zero-touch deployment approach is an accomplishment all by itself. The multitude of components required to get this far requires no small effort in designing and integrating a range of technologies. Congratulations!

In fact, this simple example is only a small portion of what ConfigMgr brings to the table in terms of OS deployment. This walk-through represents barely the simplest of installation use cases, and it is written to get you started down the path of even greater automation. Once you understand the basics, you become well-prepared for adding to that knowledge with additional automations, many of which you only gain through your ConfigMgr infrastructure.

ConfigMgr itself comes equipped with a significantly-greater level of intelligence about the software that is deployed and managed in your environment. However, most of its functionality works best only when your IT processes subscribe to using it for pretty much everything. That means coordinating the packaging efforts of your deployed software, updates, OSs, and even desired configurations. By centralizing all facets of Windows configuration control within a ConfigMgr infrastructure, you gain a configuration management database that becomes useful for even greater levels of automation.

Consider additional capabilities that might come through centralizing control in this manner. For example, software installation can be slipstreamed into OS deployments. As ConfigMgr is as much a software installation solution as an OS deployment solution, it stands to reason that any software package created for installation can be automatically added to an image as well. Adding a bit of extra intelligence to your Task Sequences, it becomes possible for your deployment sequences to query computers for installed application packages with the goal of automatically reinstalling those packages after an upgrade.

ConfigMgr also includes support for PXE booting desktops for image deployment, similar to how WDS and MDT use the same technology. ConfigMgr includes a PXE service point role that can handle deployment of WinPE to PXE clients for a streamlined LTI experience.

A third deployment approach is also possible with ConfigMgr that transfers control to users entirely. The UDI approach gives individual users the ability to refresh their computers on their own and without IT involvement. With instrumentation for reinstalling appropriate applications and migrating user data, the UDI approach enables users to fix their own problems without requiring intervention by IT.

The Automations Virtually Never Stop

This chapter has intended to serve as the capstone for this book's discovery on automating Windows deployment. However, the conversation isn't done. The title of this book asserts that it is a definitive guide for desktop *and VDI environments*, but we haven't gotten to the virtual world just yet—I purposely decided to hold on that discussion, giving you the opportunity first to understand fully the Windows deployment process. With that information firmly in hand, this book's final chapter concludes with a look at how this information can be directly translated into the desktop virtualization world. In the next chapter, I'll show you how the information you've learned so far will bring a significant assist should you take the road of virtualizing your desktops.

You'll be happy you've stuck around this long. The information in the last chapter is not to be missed.

Download Additional eBooks from Realtime Nexus!