Realtime publishers

Automating Windows 7 Installation for Desktop and VDI Environments

Greg Shields

Copyright Statement

Chapter 3: Techniques in Installing Applications During Windows Deployment

Chapter 2 covered the first six steps in creating a fully-automated and universal Windows 7 image. With those six steps, you're ready to start deploying Windows with fully-automated ease. However, you can probably imagine that there's an important piece of this deployment that's still missing: *the applications*.

In this chapter, I'll discuss two very different techniques that you can use to install applications onto Windows 7 desktops during a deployment. I'll start by explaining the *thick approach*, and will then begin the dive into a discussion of the *layered approach*.

The thick approach is probably similar to how you've been handling applications. Using the thick approach, you create one or more *master images* that contain the operating system (OS) along with any applications you want deployed. Once created and configured to your liking, this master image is captured back to your WDS server and later redeployed to computers around your network. The approach is called *thick* because the images grow large as applications and configurations are added.

The thick approach with master images has long been used because it is easy to comprehend. Simply create an image, make any configuration changes, install and configure applications, and finally capture that image back to the deployment server. Although the approach requires extra steps, it is easy to visualize what the user will get when the captured image is redeployed: *If you created it on the master image, the user will get it.*

Step Seven: Creating a Master Image with Applications

The creation of a master image starts with the deployment of a basic image to a master computer. This master computer will be the location where you will perform the image customization work. Deploy that basic image using the process you learned in the previous two chapters.

Once the image is deployed to the master computer, make the changes you want eventually deployed to users. In addition to installing applications, you might want to set a few of their initial configurations to make things easy. You might also want to install any licenses that can be legally replicated throughout your network. Remember that any change you make on this master image will be copied to every other computer when the image is later deployed.

Audit Mode

Windows Vista and Windows 7 now include a special configuration "mode" called Audit Mode. Although you can absolutely make changes to your master computer without using Audit Mode, this special mode enables a set of protections for master images during this customization process.

Using Audit Mode is optional and out of scope for this book. For more information, see http://technet.microsoft.com/en-us/library/dd799305(WS.10).aspx.

In my sample master image, I deployed a copy of Windows 7 using our process thus far and then installed Microsoft Office. For grins, I also changed the theme and configured the power settings for the computer. For our sample solution, these simple changes represent the only custom configurations I'll set in my master image.

Invoking Sysprep

You'll need to Sysprep the computer as the very last step in this customization process. A Sysprep must be completed to prepare the computer for being deployed, as it completes a process called *generalization*. The generalization process eliminates all the data on the OS instance that links it to the hardware on which it was installed. By generalizing, you can later redeploy this image to other hardware.

Older versions of Sysprep were quite painful to use. Luckily, it has gotten quite a bit easier in Windows 7 to complete this task. Sysprep is now available directly on every Windows 7 instance. To launch Sysprep, run the command

```
C:\Windows\System32\sysprep\sysprep.exe
```

Running sysprep.exe will bring forward the dialog box you see in Figure 3.1.

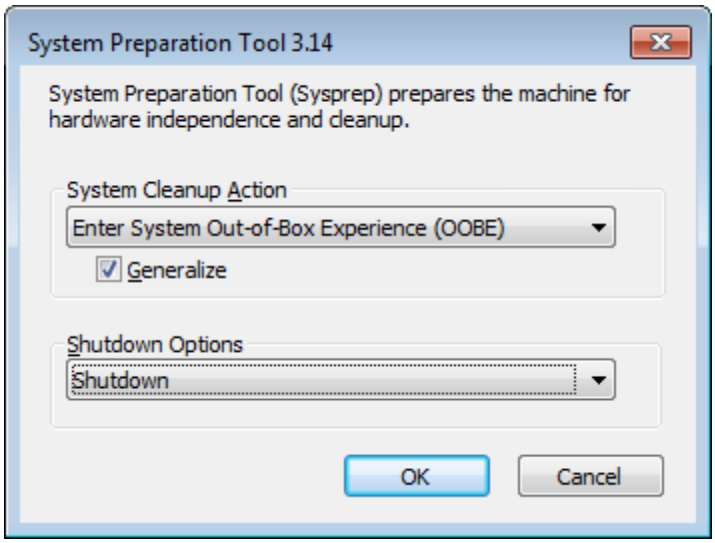


Figure 3.1: The System Preparation Tool dialog box.

Note in Figure 3.1 that the System Cleanup Action is set to *Enter System Out-of-Box Experience (OOBE)*, and that the *Generalize* check box is selected. These two settings revert the computer back to a point where the Set Up Windows wizard will be displayed upon the next reboot. This action puts the computer into the correct position for deployment. That is the same position we experienced with our basic image in the previous chapters. You'll also note that I've configured the *Shutdown Options* to *Shutdown* the computer once it has completed generalizing. This is important for the next part of this process.

> **Note**
>
> Pre-staging the computer into this mode means that our original unattend files—those we worked on in the previous chapter—should still work in customizing and automatically deploying this computer in the same way as our basic image.
>
> You should still be able to use your original unattend files, even against this customized image. Just make sure that you re-attach them to the new image back in WDS after capturing the image in the next step.

Creating a Capture Image

To deploy this image, you've first got to capture it to your WDS server. Capturing that image requires the use of a special boot image I'll call a *capture image*. This capture image boots the computer to WinPE just like a boot image. But instead of handing off to an install image, the capture image instead gathers the computer's data into a master image.

Let's create that capture image. Go to your WDS server, and click *Boot Images*. Right-click a boot image, and choose *Create Capture Image*. You'll be presented with the *Create Capture Image Wizard* just like you see in Figure 3.2. Give the capture image a name and description as well as a storage path. You'll notice that I'm storing my capture image with the boot image in *C:\RemoteInstall\Boot\x64\Images*. Click Next to create the image, which will take a few minutes to complete.

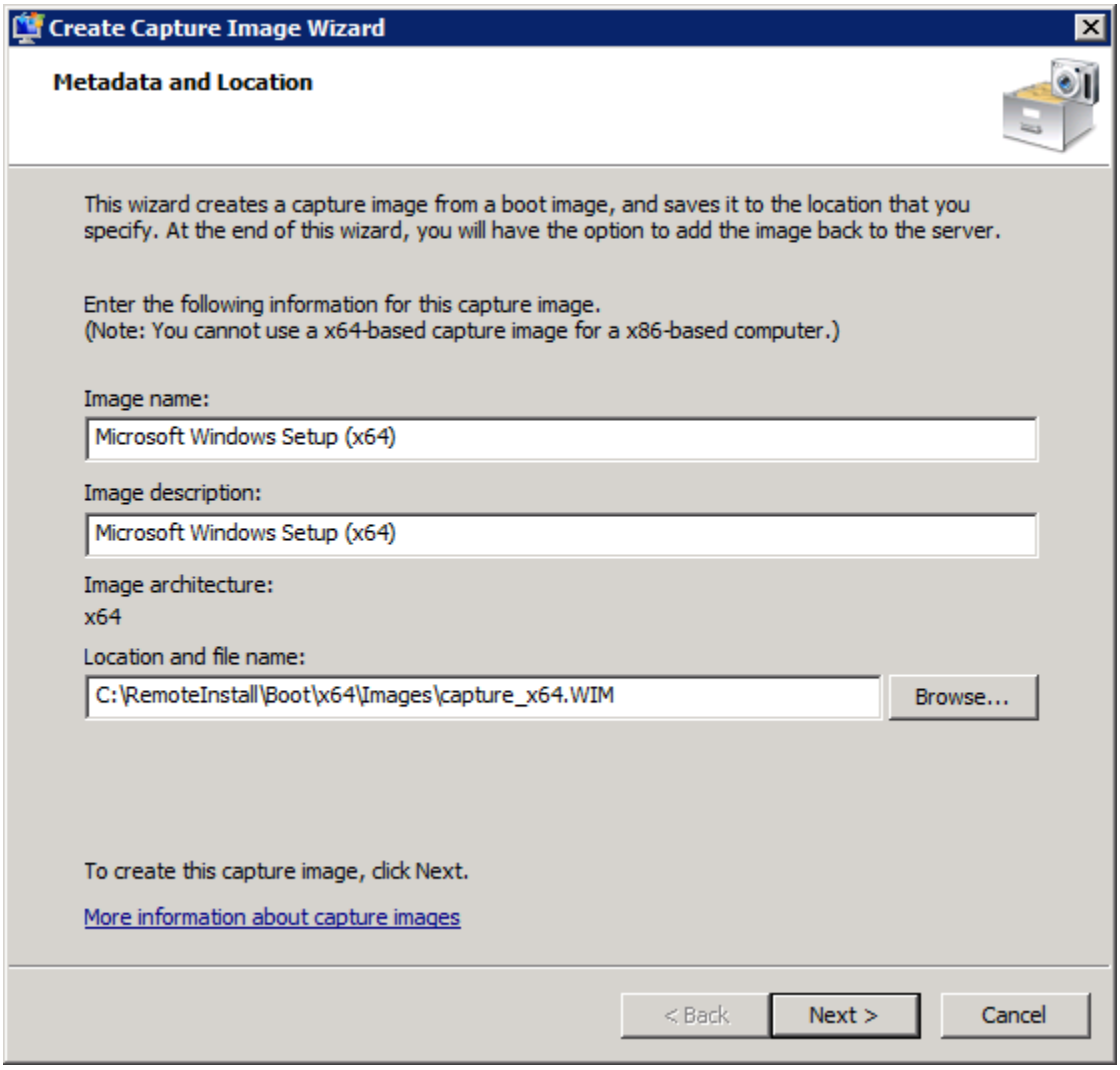


Figure 3.2: The Create Capture Image Wizard.

After the image has been created, you'll be presented with a Task Progress screen (not shown here). There will be an *Add image to the Windows Deployment Server now* check box. Select that box, and click Finish. This step adds the just-created capture image back to the WDS server, enabling the capture image to be deployed over the network just like we've been doing with every image so far.

Run through the resulting *Add Image Wizard* to add the new capture image back to WDS. Figure 3.3 shows how I've given this image a name of *Microsoft Windows Capture (x64)*. Completing this wizard and adding the image takes a few minutes once again, but as a result, you'll be able to capture a master computer's image in the same way that you deploy an image—over the network!

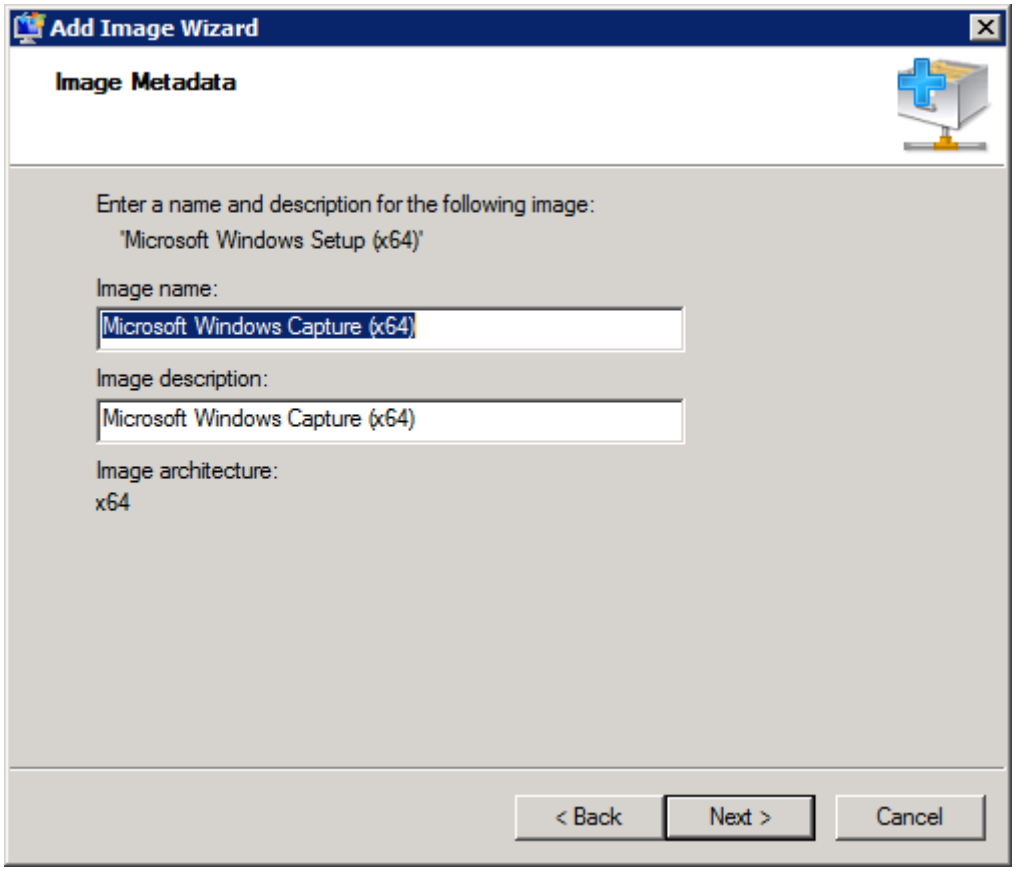


Figure 3.3: The Add Image Wizard.

Capturing the Master Image

That was the hard part. Your last task in this step is to capture the image you've just generalized and made ready for deployment. At this point, the master computer should be powered down if you configured Sysprep per the instructions. Sysprep needs it to remain powered down so that its power-back-on-into-Set-Up-Windows state is captured by the capture image.

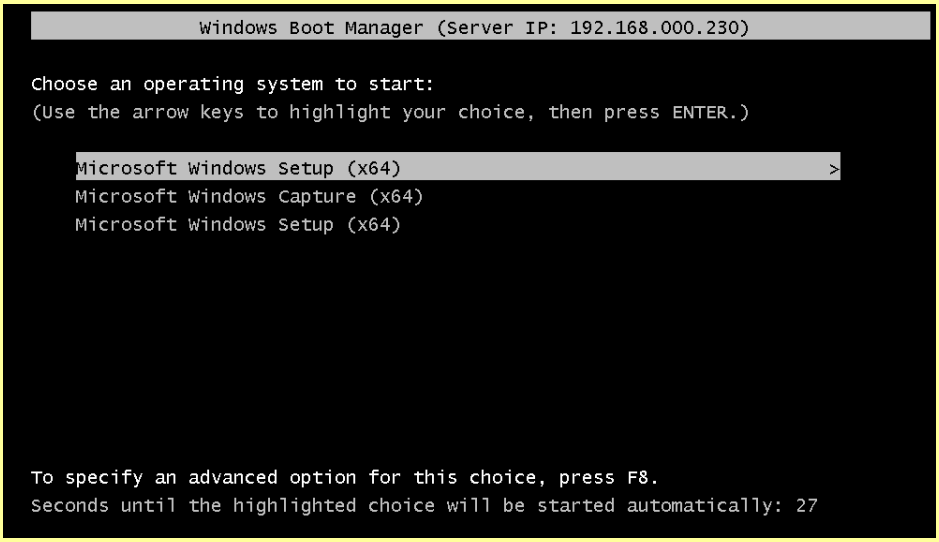
You'll do that now, but be terrifically careful with this step. Power on the powered-down master computer and connect it to your WDS server via PXE boot. *You must make sure not to power on the computer into its Windows installation.* Quickly power it back down if you have any problems with PXE. Assuming you're successful with the PXE boot, you're ready to capture its image.

Note

There's another important change to be made here first. Remember back in Chapter 2 when we attached the *unattend_x64_client.xml* unattend file to our WDS server? That unattend file boots computers and immediately begins deploying our basic Windows 7 boot image. You don't want to deploy an image at this point, you want to capture one. So, the unattend file needs to disappear for a while.

To ensure that you boot to the correct image, right-click to view the properties of the WDS server, and select the *Client* tab. Clear the *Enable unattended installation* check box. You can re-select the check box after you've completed the capture.

If you've done everything correctly, your master computer will boot to a screen similar to the following image. There, you'll want to select the capture image to start the capture process. Be very careful with this process, as booting back to Windows may require needing to re-Sysprep and shut down the computer before trying again.



Don't forget to select the *Enable unattended installation* check box once you've completed your capture. That returns your WDS server back to its normal operation for automatically deploying images.

The next set of steps is a manual process; however, this is OK because you'll be doing them relatively rarely. After booting with the capture image, you'll see an initial welcome screen. Click Next to see the *Directory to Capture* screen, similar to Figure 3.4.

Anyone think it's funny they call this the *directory* to capture? Sorry, I digress…

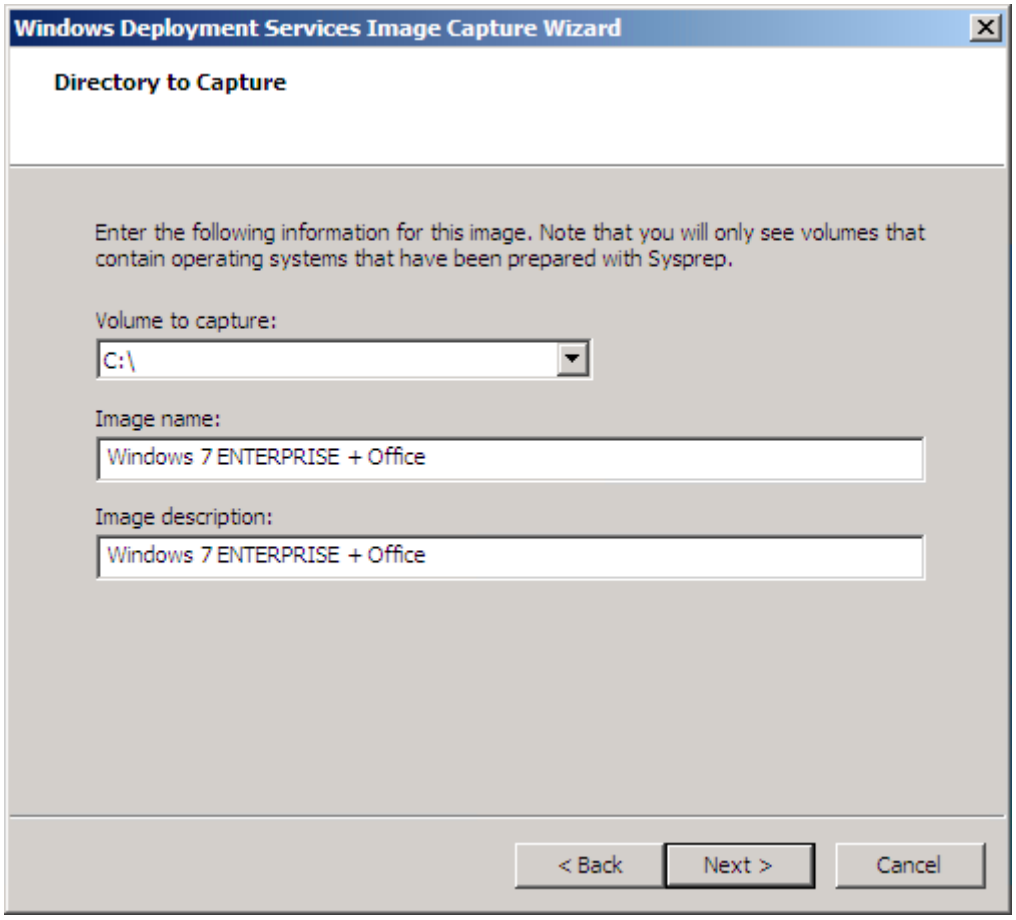


Figure 3.4: Directory to Capture.

The drop-down box on this screen is exceptionally important. In it should be the volume of the computer you'll want to capture back to WDS. You can see in my example that C:\ is the volume I'm interested in capturing. *If you do not see a volume here, your Sysprep did not complete successfully.* If you don't see a volume, log out of the capture mode before attempting any other action, power on the computer back to its installed OS, manually answer the Set Up Windows questions, make any necessary changes, and re-Sysprep the computer before starting this process again.

In Figure 3.4 are also *Image name* and *Image description* boxes. These are intended for the name and description of the image as will be seen in WDS. I've named mine to show that this image is comprised of my basic Windows 7 ENTERPRISE image plus Microsoft Office. Click Next to continue.

The next screen, shown in Figure 3.5, identifies where you want to store the install image. Be aware that this *New Image Location* screen behaves a little strangely. Follow along closely because I experienced some problems the first few runs through.

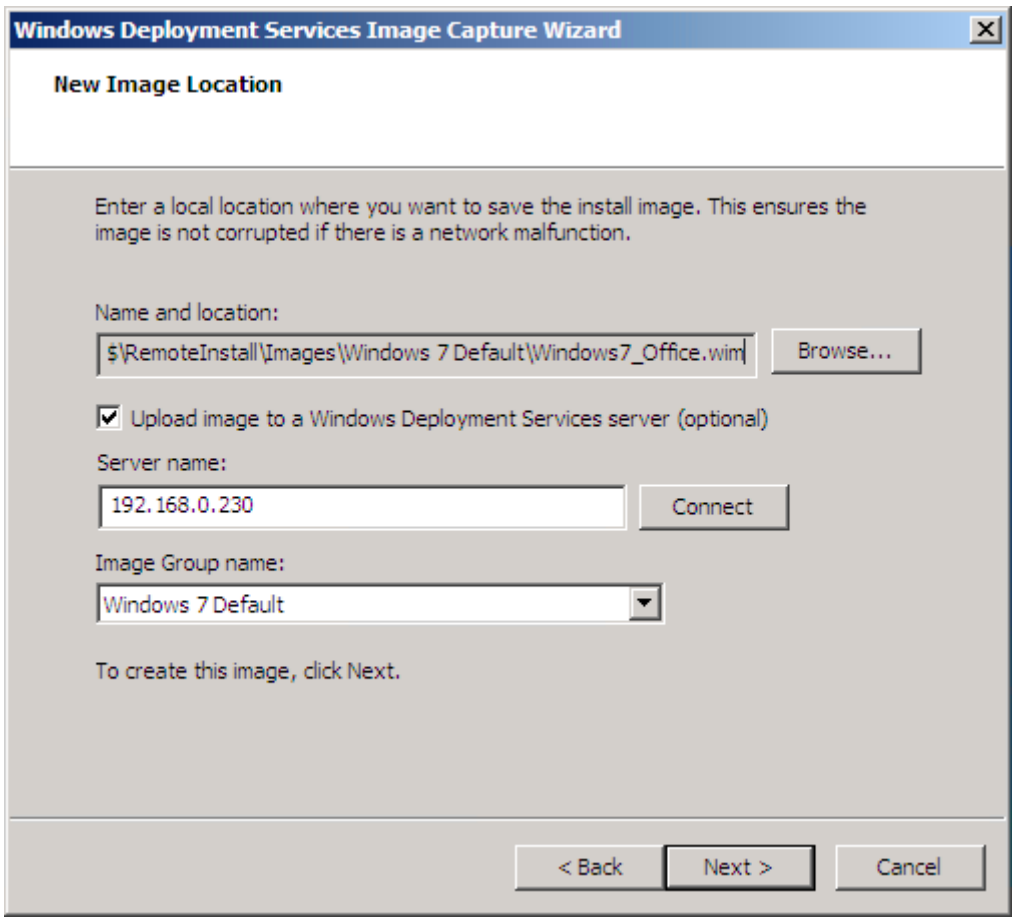


Figure 3.5: New Image Location.

Remember how creating a capture image required two steps? In the first step, the image was created, then it was uploaded to the WDS server. This rather-confusing window needs to accomplish both of those tasks. Problem is that you don't necessarily want to store your image on the master computer; you want to store it on your WDS server.

It's here where things get tricky. Click Browse in the *Name and location* box to browse to the Images folder on your WDS server. For me, the correction location was *\\wdsserver\c$\RemoteInstall\Images\Windows 7 Default*. There, give the image a name, which in my case was *Windows7_Office.wim*. You will be prompted to enter credentials to make this connection.

After completing this step, select the *Upload image to a Windows Deployment Services server (optional)* check box. Enter the name of your WDS server under *Server name*, and click Connect. You will be prompted again for credentials. After successfully authenticating to your WDS server, a list of image groups will appear under *Image Group name*. Select the correct image group, and click Next to continue.

Note

I've seen some very strange behavior at this configuration screen. Namely, that the authentication session used to connect to the WDS server's disk could not be shared by the WDS server connection. To get around this odd limitation, I used the server name for the disk connection and its IP address for the WDS connection.

You might experience different results. If your connection attempt results in an error message, consider using my dual name and IP address trick to connect.

You're done! The capture process should begin. Capturing an image takes a bit longer than deploying one, so plan on quite a few minutes passing. At its conclusion, you should find your newly-created image in the WDS console. Consider performing a test deployment to see how well it works and whether your configurations deploy correctly.

Your configuration should not have affected many of the questions and answers in your unattend file. If you have configured a specific image to be deployed in your Client file, you'll want to change its name and filename. The other settings should remain effective and should correctly answer the questions required by this custom image's Set Up Windows prompt. If you've done everything correctly, you should be able to deploy your custom images using exactly the same process as your basic images from Chapters 1 and 2.

Stepping Back: Application Delivery Mechanisms

Step Seven seems pretty involved, particularly when you pair it with the realization that every little change to a Windows image will require a deployment, a configuration change, and a capture. That's a lot of steps when you consider how often changes can be necessary.

An interesting thing about the Windows XP-to-Windows 7 upgrade in comparison with previous OS upgrades are the new mechanisms now available to actually deliver applications to users. Before I get into any step-by-step processes, I want to step back a minute and really focus on the concept of *application delivery*. Remember back not too many years ago when we didn't have many options for delivering applications. For most of us, when a user needed an application, we arrived at their desk with a DVD and installed it locally.

That "DVD" application delivery mechanism suited us well for many years, but over time, we grew tired of needing to show up at every desk after every request. It is at that point where other application delivery mechanisms grew compelling:

- Group Policy Objects (GPOs) began delivering applications through our Active Directory (AD) infrastructure.
- Application management solutions like System Center Configuration Manager, System Center Essentials, and others from third parties gained widespread use.
- Application virtualization solutions like Microsoft's App-V, among other third-party solutions, can now stream an application through what can only be called a glorified file copy.

These nifty solutions enable us to evolve towards a kind of *layered approach* in managing our desktops—and, indeed, our desktop images. Rather than installing applications and other customizations directly into images—the *thick approach*—smart administrators know that a *thin approach* is a lot more manageable.

With the thin approach, deployed images remain relatively basic, containing few applications that are directly installed into the image. Pairing down the image to a configuration that is as simple as possible reduces the need for updates over time. Instead, applications are installed (layered!) on top of the basic image using one of the other mechanisms discussed in the previous bulleted list. You'll obviously have a bit more setup in constructing that layering infrastructure, but the result will be a much cleaner and more elegant solution. More importantly, after the up-front work is complete, your day-to-day operations became far, far easier.

Figure 3.6 shows a graphical representation of this layering that I first created for *TechNet Magazine* back in December 2009[1]. In it, take a look at the *Core Operating System* and its *Drivers*. These elements, which we worked on in Chapter 2, are only part of the giant stack that makes up every desktop.

[1] http://technet.microsoft.com/en-us/magazine/ee835710.aspx

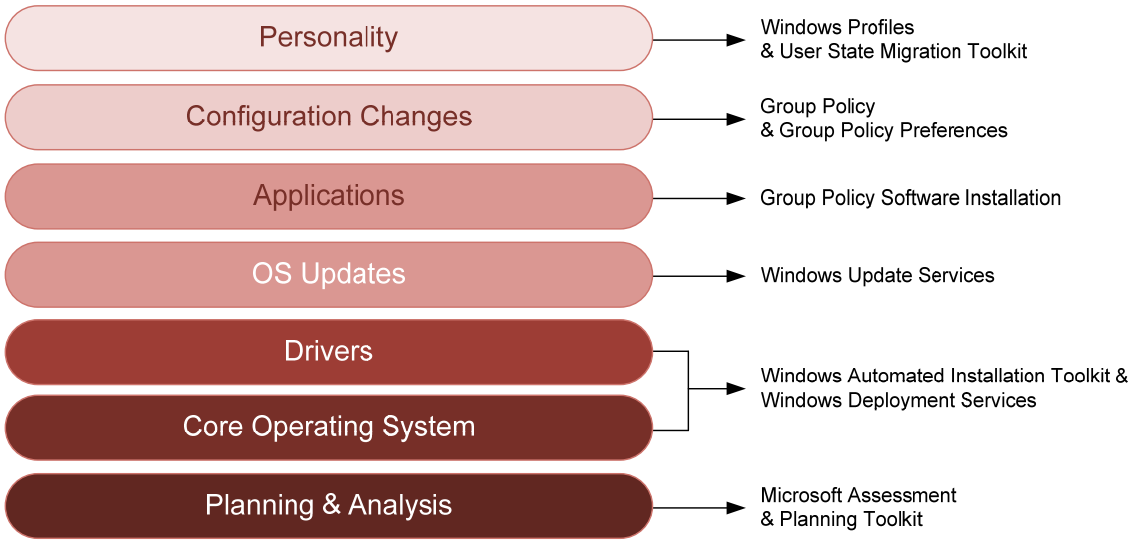


Figure 3.6: Layering Windows OS deployment.

I'll spend some time with the other elements in this stack in other chapters. For now, however, I want to continue the conversation by talking more about the layer marked *Applications*.

From Thick to Thin

Think about how the thin approach will dramatically change your management of desktop images. Let's compare "thin" with "thick": How many desktop images do you have today? Ten? Dozens? Hundreds? You've needed each of those desktop images because they're slightly different in composition. One may have Office 2010 on it, while another still holds Office 2007 or even 2003. Even others may be different only in their drivers.

We fixed the driver problem in the previous chapter. There, we used WDS' *Drivers* node to create a pool of drivers that any deployment can pull from when Plug and Play finds a hardware match. Accomplishing this means we can now get rid of every image that's only different because of its driver composition.

This chapter's goal is to get you started down the road of eliminating the rest of your images. Those images exist because their application composition is different. This chapter won't yet give you the complete solution, but it starts the discussion. Actually getting to that thin nirvana will take many of the remaining chapters.

Thick to Thin: What Do You Need?

Think about the only real way to get to this single-image nirvana: *Stopping the practice of installing applications and customizations directly inside the image.* That's kind of a neat thought, yes? If I deploy an image that doesn't contain applications—or that contains only the applications I want everywhere—I can then layer on top of that image everything else during subsequent steps.

What would I need to do this?

1. I need some mechanism to automatically install applications to desktops after they're deployed.
2. I need some mechanism for identifying which applications should be installed onto which desktops.
3. It would be nice to have another way to inventory applications on my existing desktops and hold that information in reserve. I could then use it to redeploy the correct applications after the desktop is upgraded.
4. I need a way to repackage those applications so that they can be installed via this magical automatic installation system. As with the Windows installation, fully-automated application installation needs to ask no questions during an installation. It needs to start, finish, and be done.
5. I need to recognize that the applications themselves are only part of the job. Along with applications, most users have data that needs to be transferred with the application. Users don't like it when their data isn't available after their computer is upgraded, and the manual steps to make that data available are painful for them and us. As a result, any upgrade project should probably automatically manage my user data as well.

That's a pretty tall order of features. And you're right in thinking that we'll need to add capabilities into our automated solution if we're to get there.

What's great is that you can actually fulfill these needs using Microsoft's free solutions. You can also fulfill these needs using for-cost solutions. As I mentioned in Chapter 1, although Microsoft's no-cost solutions might not be the slickest ones around, they're free.

If your deployment project is small, low in complexity, or suffers under a really small budget, Microsoft's no-cost solutions integrate perfectly to create most of this single-image nirvana. If you've got a few more bucks, use them to expand that solution to include some of the costlier bits—either from Microsoft or third parties. Those additional spendy bits bring greater administrative control, more reporting, and better granularity in the actions they perform.

The Microsoft Deployment Toolkit

But this book is all about the free tools, so that's where we'll focus. I'm sure there are plenty of third-party solution salespeople who'll be happy to explain how their products make all of this simpler.

Once you've implemented what I've told you in Steps One through Seven, you have everything you need to create a fully-automated and universal system for deploying Windows 7. You can actually put down this book and start deploying Windows with success. In fact, if your upgrade project is really small, these seven steps are probably all that you need.

If your upgrade project is more than really small, the steps to this point probably don't fulfill your needs. You've got to be pretty curious at this point how to get to this layered approach for applications, user data, and other customizations. Even if you're small, after going through that painful deploy, reconfigure, and capture process a few more times, this layered concept will probably become compelling.

Unfortunately, *we've pretty much come to the end of the capabilities that we can squeeze out of WDS alone*. In order to take that next step in evolving our solution towards real automation, we need to layer another of Microsoft's acronyms (and applications) on top of WDS. That new acronym and application is called the MDT or the *Microsoft Deployment Toolkit*.

If you've been around the block for any period of time, you know that Microsoft's deployment tools have for years been…well…awful. They worked, but they were very technology focused in the capabilities they presented. What the world has really wanted is a more process-focused solution; one that could truly automate all the usual workflows that make up an upgrade project: user data collection, application inventory, application compatibility checking, application installation, and user data injection—all those "other" tasks that had for so long not been there in RIS, WDS, and the other tools alone. What the world wanted was a way to create sequences of the tasks that go on in an upgrade project. Those sequences would be codified instructions that automatically did all those nifty things on behalf of the administrator. What a neat idea, eh? Well, Microsoft listened, and the MDT was born.

Coming Up: Folding Your Deployment Infrastructure into the MDT

The next chapter will start our exploration into the MDT. As you'll discover, the MDT arrives as a kind of procedural wrapper around all the technologies that we've already explored to this point. You won't necessarily be throwing away anything you've already learned. In fact, by navigating your learning in this way, you're particularly prepared to do well with the MDT's task sequence method of deploying Windows desktops.

The next chapter continues the conversation on techniques in installing applications, focusing on the ways in which the MDT streamlines the process and moves you towards a more layered approach. I'll start by helping you get the MDT installed and initially configured. Then, I'll show you the steps you'll need to implement to incorporate application installations within your Windows deployment.

You also know that applications aren't everything. Applications by themselves are only of limited assistance. You also need to return user data back to a computer after it's been upgraded. Following Chapter 4, I'll continue by discussing the all-important topic of user data and show how the MDT will assist in preserving that data during an upgrade.

Download Additional eBooks from Realtime Nexus!

Realtime Nexus—The Digital Library provides world-class expert resources that IT professionals depend on to learn about the newest technologies. If you found this eBook to be informative, we encourage you to download more of our industry-leading technology eBooks and video guides at Realtime Nexus. Please visit http://nexus.realtimepublishers.com.