Chapter No. 4
"Getting Started with Horizon Mirage Layering Technology"
In this package, you will find:

A Biography of the author of the book

A preview chapter from the book, Chapter NO.4 "Getting Started with Horizon Mirage Layering Technology"

A synopsis of the book’s content

Information on where to buy this book

About the Author

Peter von Oven is an experienced technical consultant and has spent the last 20 years of his IT career working with customers and partners in designing technology solutions aimed at delivering true business value. During his career, Peter has been involved in numerous large-scale enterprise projects and deployments presented at key IT events and has worked in senior presales roles for some of the IT giants. Over the last 8 years, he has focused his skills and experience within the desktop virtualization market, and today he leads the End-User Computing SE team at VMware in the UK&I, tasked with delivering the next generation of end-user computing and workforce mobility solutions.

For More Information:
There are a number of people I want to thank for the support they have given me during the writing of this book. Firstly and most importantly, I would like to thank my family for their support while I have spent many evenings and weekends at the keyboard; and especially my two daughters, Eleanor and Charlotte who, at an age where they are both learning to read and write, have been fascinated by daddy writing a story about a computer; to the Packt Publishing team for giving me the opportunity to write this book; and particularly Mohammed Fahad whose advice and guidance to me as a first-time author has been invaluable. Finally, I would like to thank some of my colleagues, particularly Simon Townsend who has supported me through this project and has also given me the encouragement to achieve this, as well as Paul Whyton, Peter Schraml, and Spencer Pitts who have all helped answer questions along the way.
VMware Horizon Mirage Essentials

VMware Horizon Mirage is a desktop software management solution designed to help you manage, support, and protect the software running on Windows-based desktops and laptops. This doesn't just mean the Windows desktop operating system itself, but a deeper and more granular level of management by abstracting the individual component parts of what makes up the entire desktop operating environment. In Horizon Mirage, these are referred to as layers, where a layer can be made up of applications, device drivers, user profiles, and user data.

The Horizon Mirage product is not entirely new and was conceived and developed by a company called Wanova back in 2008, before being acquired by VMware in May 2012. The initial question asked by many, myself included, was: "Why would VMware purchase a company that delivered a solution that provided an alternative to desktop virtualization (VDI) and didn't require any hypervisors at the backend in order to work?"

When you start to look closely at what Mirage delivers you soon start to see the synergies of why this acquisition makes perfect sense. It opens up the scope of delivering a broader solution and ultimately delivering better end user experiences that are easier to manage. It's not just about delivering VDI any longer; it's about delivering a complete end-user computing experience, regardless of whether it's physical, virtual, or a combination of both.

To some people Horizon Mirage will be something new, and to others it may be something they already have some experience with. Whichever camp you fall into, this book is designed to give you a detailed understanding of every aspect of Horizon Mirage, allowing you to successfully deploy and manage your virtual or physical desktop environments.

What This Book Covers

Chapter 1, An Introduction to VMware Horizon Mirage, explains all about what Horizon Mirage is, the terminologies used, and the infrastructure components that make up the solution. There are also details on how to build your own lab environment, which is used throughout the book as working examples.

Chapter 2, Design, Install, and Configure, explains how to design and size a Horizon Mirage infrastructure using best practice. Once designed, we can go on to install the product using step-by-step guides, which include illustrations.

Chapter 3, A Guided Tour of Management Consoles, will take you through a guided tour of both the MMC-based Mirage Management Console and the Web Management Console, helping you gain a better insight into how to perform various Horizon Mirage tasks and where to find them in the consoles.

For More Information:  
Chapter 4, Getting Started with Horizon Mirage Layering Technology, introduces you to the Horizon Mirage layering technology, the foundation for image management with Horizon Mirage. You will learn what a layer is in Mirage terminology and how to manage and build operating system layers and application layers from both a practical and theoretical perspective.

Chapter 5, Manage – Centralized Image Management and Deployment, explains how to start working with the Mirage layering technology to deliver image management, address remote office locations, and repair endpoint software issues.

Chapter 6, Migrate – Operating System and Hardware Migration Made Easy, discusses the migration feature of Horizon Mirage and how to move from Windows XP or Windows Vista to Windows 7. It also covers migration between hardware platforms.

Chapter 7, Protect – Backup, Recovery, and DR for Endpoint Devices, discusses how Horizon Mirage can deliver protection to the endpoint devices in your environment. This ranges from full-blown DR and restoration capabilities, to allowing users the ability to restore files themselves or to be able to access them via a web browser from another device.

Chapter 8, Advanced Configuration, References, and Links, covers advanced features such as CVD upload policies, upgrading to a new version, integration with Horizon View, references, and useful links to the information used in this book. This is a bonus online chapter which you can download at http://www.packtpub.com/sites/default/files/downloads/2352EN_Chapter08.pdf.

For More Information:
4

Getting Started with Horizon Mirage Layering Technology

In this chapter we will introduce you to the Horizon Mirage layering technology—the foundation for image management with Horizon Mirage. You will learn what a layer is in Horizon Mirage terminology and how a desktop operating system is put together from its component parts (layers). We will also cover capturing a base layer and an application layer and will look at best practices for building your reference/gold image and applications. Finally, we will look at how Horizon Mirage manages device drivers.

An introduction to layering technology
In this first section, we are going to discuss what layering means in Horizon Mirage terms and how to get started on working with these layers.

What is a Horizon Mirage layer?
The definition of a layer in Horizon Mirage is that it provides a level of abstraction. If you think about what that usually means in traditional VMware terms, it's ESXi running virtual machines, where the ESXi hypervisor provides the abstraction between the physical hardware and the guest operating systems.

If we turn to end-user computing terms, it would mean a traditional VMware VDI solution to virtualize the desktop operating system (a hosted virtual desktop) or it could be virtualizing an individual application, abstracting it from the operating system using VMware ThinApp. These two components would be the core pieces in building a user's desktop. The final piece in this jigsaw would be to deliver a user's profile, as typically we would deploy a floating, nonpersistent desktop, built on demand as a user logs in.

If we take those three individual components—the operating system, applications, and user profile—we deliver them back to the user in the form of their virtual desktop; however, the user has no idea that each component was delivered independently. This would look somewhat like the following diagram:

What we have just described is the traditional VDI model; so how is Horizon Mirage different? Basically, it can deliver these levels of abstraction, or layers as they are called in Horizon Mirage terminology, without the need for a hypervisor or any level of abstraction. It works natively and, as such, addresses the use cases where the traditional VDI model of remotely accessed centrally hosted desktops does not fit, or the business case just doesn't stack up.

Because it doesn't use any virtualization technology, some of the limitations of virtualizing apps and operating systems no longer apply. For example, there are no complicated type-1, client-based hypervisors required in order to make the layers work. The Windows operating systems and applications run natively on the endpoint device. When it comes to delivering applications, not virtualizing them means that you can now include applications that have driver dependencies. Even with VMware ThinApp and Microsoft App-V, you cannot virtualize an application that includes drivers.

Everything runs natively on the endpoint device, so there should be no compatibility issues. If the endpoint runs a supported Windows operating system, it’s pretty much good to go.

**What does a Mirage layer look like?**

To start with, Horizon Mirage looks at separating the way the layers are managed into two different categories. The first category of layer is the one that is controlled and managed by the IT department. This includes drivers, base layers, and application layers and forms the core image of an endpoint device. Bringing these layers together effectively delivers a standard and consistent build.
The other category refers to those layers that are controlled by the end user. These layers include any form of personalization that the user has created, such as their own data, profile changes, and any applications that they have installed themselves in addition to the core applications delivered by IT. It also includes the machine identity layer.

Layers are stored centrally on the Horizon Mirage Servers in the datacenter when a user's endpoint is centralized by the Horizon Mirage Client installed on that endpoint.

Horizon Mirage can then manage each layer of the centralized CVD independently, allowing you to build or update an image from its component parts and when complete, synchronize that built image back to the endpoint. The following diagram shows a high level view of the Mirage layers:

Managing the layers independently allows IT to perform actions such as migrating the operating system (base layer) or delivering an individual application or set of applications to an end point (app layers). In the following sections we are going to look in detail at how to create layers.

**How to capture a base layer**

Before we move into the management of images, which we will cover in *Chapter 5, Manage – Centralized Image Management and Deployment*, the first thing we need to do is to make sure we have an image to work with.

For More Information:
In this section we are going to look at how to create a reference machine and then centralize that to create and upload a reference CVD in preparation for capturing a base layer from it. Once we have this base layer, we can use it as the standard gold image to deploy to other endpoints across your environment.

**Stage 1 – building a reference machine**

The first stage in the process, as shown in the following diagram, is to build your reference machine or gold image. This is the image that will be deployed to all the other endpoints in your environment:

Before we start the building part, there are a few things to cover first.

**What is a base layer? – a quick recap**

In Chapter 1, *An Introduction to VMware Horizon Mirage*, we defined what a base layer is, but we are going to spend a short while just playing that back as a quick recap. A base layer is your core image and contains the OS, updates, patches, service packs, and the core applications that the majority of users will get, such as Microsoft Office or antivirus software. It is captured or taken from a reference machine.

**What is a base layer used for?**

Once you have created your base layer, you can start to deploy it by assigning it to the already uploaded CVDs in the datacenter. By doing this, you will effectively be replacing or updating the existing base layer of the uploaded CVD or collection of CVDs. In turn, that synchronizes back to the endpoints. The result is that the endpoints are now running the same base layer that you built and captured from your reference machine. This is how Horizon Mirage delivers centralized image management.

For More Information:

Considerations when creating a base layer

There are a few guidelines and best practices for creating base layers. Ideally, you want as low a number of base layers as possible. If feasible, having one base layer would be the ideal scenario.

Hardware

When you deploy an image, Horizon Mirage is able to include the relevant drivers for that endpoint using the drivers stored centrally and creating a driver profile. However, there are some drivers that might need to be installed once the base layer has been deployed. For example, the driver may be an integrated part of a software package, such as printer/scanner drivers for a multifunction device where the driver gets installed as part of the control center software for that device rather than just the driver.

Applications

When it comes to applications, the best practice that is applied to building any other desktop gold image applies here too, and that is to include the software and applications that cover the majority of use cases within the gold image. So, for example, you could say that most users will get the basic Microsoft Office Suite and have Word, Excel, PowerPoint, and Outlook installed on their desktops. If this is the case, these should be included within the base layer along with other things such as Adobe Reader.

Any additional application that may be deployed on a smaller use case basis or any specialized application that may be deployed for particular departments can be delivered after the initial base layer has been deployed using application layers or may be delivered as a virtual application using ThinApp; for example, applications such as Microsoft Visio or Microsoft Project.

Licensing operating systems and applications

One thing that is important is the way in which the installed applications or operating systems are licensed when deploying them with Horizon Mirage.

The preferred method, and probably the easiest way to license would be to use a Volume Licensed Key (VLK). With a VLK, you would enter license information during the creation of the base layer as you installed the operating system on the reference machine and then activate it. Using a VLK means that, when the image is deployed to endpoints, it neither needs to be activated nor does it need to check in with a licensing server.

For More Information:
If you use a Multiple Activation Key (MAK), this should be managed either manually or via a script after the base layer has been deployed. A MAK has a predetermined number of allowed activations based on your Volume Licensing agreement with Microsoft, so you might need multiple MAKs. That being the case, you would need a base layer for each MAK so it makes it easier to apply this information after base layer deployment.

If you use a KMS server for licensing, you would use the same process as you would for building any other image, ensuring that you activate the operating systems and any applications that reside in the base layer during the installation on the reference machine.

If you use OEM licensing, you could use this with the reference machine; however, with Mirage, you are able to migrate across different hardware platforms and vendors. Hence, if you built your reference machine using HP hardware and an associated HP OEM license, you would not be able to migrate to Dell hardware without updating the license key to either a Dell OEM license or other VLK.

**Working with endpoint encryption solutions**

The following are some of the endpoint encryption solutions:

- **Full disk encryption (FDE):** Horizon Mirage cannot make changes to partitions or boot sectors, so you need to install any full disk encryption that will modify your hard drive before you install Horizon Mirage. Horizon Mirage supports the following:
  - Checkpoint
  - PGPdisk
  - Sophos Safeguard
  - McAfee Endpoint Encryption

- **Microsoft BitLocker:** This is fully supported by Horizon Mirage, but you must first enable it because a base layer does not provide the ability to enable or disable the encryption.

- **Microsoft Encrypted File System (EFS):** Horizon Mirage fully supports EFS. By default, files are uploaded to the Horizon Mirage Server unencrypted, with file encryption being maintained as a new feature of Mirage. Users can only access and decrypt files created by using their AD account credentials.

**Base layer use case examples**

The following examples show some typical use cases for different ways of managing, deploying, and maintaining base layers.

---

For More Information:

The departmental or individual use case scenario

In this example, the base layer consists of the following software components:

- Operating system
- Antivirus and firewall software
- Standard desktop tools and utilities
- Core Office applications
- Specific applications for the department/use case

As you can see from the content of the base layer, everything is handled within a single layer and therefore does not require any additional application deployment.

This approach has some positive aspects because everything gets deployed as a single layer and in one go. However, you will need to maintain more reference machines and more base layers as you will typically need one of each for every department/use case.

The hardware-specific use case

Similar to the previous example, the base layer consists of the core operating system and applications but in this second use case, it also includes specific support for specialized hardware (as shown in the following list), maybe for something such as a design office with CAD/CAM apps and hardware:

- Operating system
- Antivirus and firewall software
- Standard desktop tools and utilities
- Specific hardware support for printers, plotters, and specialized CAD peripherals
- CAD/CAM software and applications

In this use case, the base layer contains everything relevant to the CAD office and has all the hardware support, applications, and drivers prebuilt into the base layer.

As with the previous use case, you would need different base layers for each use case, when maybe the best option would be to deliver the applications as a separate layer.

For More Information:
Best practice and tips
As we discussed at the beginning of this section, by building a reference machine you are building a gold image for all the endpoints in you environment. To this end, you will follow your standard build procedure to create the reference machine, while paying attention to the points highlighted previously.

In addition, you should disable any autoupdating software such as Windows Update or Adobe Reader. The reason is being that you want the endpoints to be the same as the reference machine and you want them to be updated in a controlled manner rather than randomly, which would end up in inconsistent builds. If you start to have differences between a reference machine and endpoints when enforcing base layers and application layers, you could end up with application conflicts.

In case of antivirus software, there are usually specific configuration details that you need to take into account when deploying the software on a gold image or reference machine. Check with your AV vendor what their best practice is for this.

Using the information described so far in this chapter, you should now be able to build a reference machine from which to work. We are going to look at the process of how to capture a base layer and an application layer using our example lab.

Stage 2 – installing the Horizon Mirage Client
We have already covered the installation process for the Horizon Mirage Client in Chapter 2, Design, Install, and Configure, but now we are going to install it on our reference machine. The process is exactly the same for the installation; however, this time we are not going to centralize the endpoint just yet; instead, we are going to upload it as a reference CVD.

In our example lab environment, we will install the Horizon Mirage Client on the REF-PC-WIN7 endpoint.

As we are planning on using this as our reference machine, there are some additional tasks that you need to perform in order to prepare the image for use as a base layer. These are covered in Stage 4 – preparing the image before capture.

For More Information:
Stage 3 – centralizing the reference machine

Now that we have built the reference machine, it's time to upload it to the Horizon Mirage Server as a reference CVD. The following is a diagram of the process:

As we have seen, the process of centralizing an endpoint is managed from the Horizon Mirage Management Console; however, this time it is slightly different from the normal centralization process because it isn't one of the tasks on the Common Wizards screen. The process of centralizing an endpoint now is as follows:

1. Navigate to the Pending Devices list and highlight the REF-PC-WIN7 device (labeled as 1). The device should now appear in this list as the Horizon Mirage Client has been installed:

2. Right-click on REF-PC-WIN7 and, from the menu that appears, select the Create a new Reference CVD option.

For More Information:
3. Select the **Create Reference CVD for Base Layer capture** radio button (labeled as 3) on the **Activate Device** window and click on **Next >** as shown in the following screenshot:

4. Select the **Horizon Mirage default CVD policy** option (4) and click on **Next >**.

5. Select the **Don't use a Base Layer** option (labeled as 5) as we are going to create a new base layer in this chapter. Click on **Next >** to continue:

---

For More Information:
6. Select the **Automatically choose a volume** option (labeled as 6) as shown in the next screenshot. In our example lab, we only have one volume but, if you have created a different one, use that instead. Click on **Next >** to continue:

![Automatically choose a volume screenshot]

7. The **Summary** screen gives an overview of the options you have selected from the previous menus. Verify whether the details are correct (labeled as 7) and click on **Finish**. If not, click on **< Back** to return to the previous screens:

![Summary screenshot]

The endpoint will now start uploading to the Horizon Mirage Server as a reference CVD. You can check the progress from the **Reference CVDs** section in the Management Console or by looking at the endpoint and monitoring the **Status** dialog box.
Stage 4 – preparing the image before capture

Before you can capture a base layer, there is another part to the process in preparing the image on the reference machine. This basically moves the BCD boot store from the Windows 7 partition and puts it in to the partition on which Windows is placed.

1. On the endpoint device REF-PC-WIN7, right-click on the Mirage icon on the taskbar, select Tools (labeled as 1) and then select Windows 7 Image Setup (labeled as 2). This is shown in the following screenshot:

   ![Image of Mirage icon and Windows 7 Image Setup]

2. A dialog box opens with some instructions on the image preparation process. Click on Setup to start the image preparation process:

   ![Image of Windows 7 Image Setup dialog box]

3. You will briefly see a command prompt dialog box open; a script will execute, and then you will see a dialog box saying that the setup has completed successfully. Click on OK to close the box and exit the setup tool. Allow the endpoint to synchronize the changes back to the reference CVD by clicking on Sync Now.

We are now ready to capture the base layer; we will cover this in the next stage of the process.

For More Information:
Stage 5 – capturing the base layer

We now have a centralized reference machine uploaded as a reference CVD. The next step is to capture a base layer from that CVD in preparation for using it to deploy or migrate our endpoints.

In the following sections, we will work through the steps required to capture the base layer from this new reference machine.

The next stage of the process is shown in the following diagram:

The base layer capture process

You can initialize the base layer capture process by right-clicking on the reference CVD and then choosing the option from there or by using the Common Wizards option. We are going to use the Common Wizards option in this example thus, from the Common Wizards screen, click on Capture Base Layer as shown in the following screenshot:
Perform the following steps to capture a base layer:

1. Click on the **Use an existing reference CVD** radio button (labeled as 1) as shown in the following screenshot:

![Capture Base Layer](image1)

2. Highlight the **REF-PC-WIN7** CVD (labeled as 2) from the Select a Reference CVD screen and click on **Next >** to continue:

![Capture Base Layer](image2)

3. Click on the **Create a new layer** radio button. You then need to enter a name for this base layer (labeled as 3). In our example, we will call it **Win7 Core**. Then give the base layer a description (labeled as 4). We will call this example **Windows 7 core build**.

For More Information:
4. Finally, we can enter a version number for this base layer (labeled as 5). As this is the first version, we will call this 1.0. This is shown in the following screenshot:

![Capture Base Layer window](image)

5. The next screen is the **Validation** screen. This will show you any compatibility issues in your base layer that could cause an issue when you deploy this base layer out to your endpoints.

6. On the **Summary** screen, ensure that the options you have selected and information entered are shown. Click on **Finish** to start the capture process.

7. You will then see a dialog box asking if you want to switch to the Task View. Click on **Yes** to switch to this view to monitor the progress of the capture process (labeled as 6):

![Task Monitoring window](image)

---

For More Information:
Updating an existing base layer

If you update your reference machine and in turn the reference CVD, you will need to update the base layer to reflect this. To do this, first update your reference machine and then ensure that all of the changes made are synchronized to the reference CVD on the Horizon Mirage Server:

1. From the Reference CVD section in the Management Console, select the reference CVD that you want to update. In our example, we will again use the REF-PC-WIN7 CVD. Right-click on it and choose the Capture Base Layer… option.
2. Click on the Update an existing layer radio button (labeled as 1). From the drop down box (labeled as 2), select the layer that you want to update. In our example, we will use the Win7 Core base layer to update.
3. Give the new base layer a description (labeled as 3) and finally click on the radio button (labeled as 4) to reflect what type of update this is, either minor or major. In our example, we will make this a minor point release and so will select a minor version (1.1):

Capturing an application layer

In the previous sections of this chapter we have discussed how to create a reference machine, centralize it to the Horizon Mirage Server as a reference CVD, and capture a base layer to be used as the core image.

For More Information:
In the following sections, we are going to look at how we capture an application layer so that we can deploy applications independently of the base layer. Not only does this allow us to update applications without the need to rebuild or change the reference machine, we can also deploy applications on a departmental or use case basis by building individual layers that are delivered together to build the complete desktop image.

The application capture process

If you are familiar with VMware ThinApp application virtualization technology, the process to capture an application layer with Mirage is similar. The following diagram shows an overview of the capture process:

As shown in the preceding diagram, there are three steps to capturing an application layer in Horizon Mirage. Each stage is explained in more detail in the following sections along with the practical side of the process of capturing an application layer.

Stage 1 – performing a pre-scan

The first step is to get a base line of the reference machine, that is, what the machine state looks like before the application is installed. So, before we start, we need a reference machine for building the application layers. We could use the same reference machine that we used for capturing our base layer and then restore it each time we want to capture a new application. Or, as we are using VMware Workstation, we could take snapshots and then roll back afterwards. The key thing is that the endpoint needs to appear in the Pending Devices list.

For More Information:
In the example lab, however, we have a prebuilt endpoint called REF-PC-APPS that we will use for application capture. This machine is effectively a copy of the Windows 7 reference machine with the Mirage client installed and connected to the Mirage Server. It appears in the Pending Devices section as it doesn't get centralized.

The following steps will help you to perform a pre-scan:

1. Navigate to the Common Wizards screen in the Horizon Mirage Management Console and click on Capture App Layer:

2. From the Select Pending Device dialog box (labeled as 2), click on the REF-PC-APPS device and click on Next >:

3. From the Select Upload Policy dialog box (labeled as 3), click on the Horizon Mirage default option and click on Next >:

For More Information:
4. Click on the **Automatically choose a volume** radio button (labeled as 4) and click on **Next >**:

5. Horizon Mirage will now check the compatibility of the task and will report back on any issues it finds or detects relating to capturing an application. If there are no issues, click on **Next >** on the **Check Compatibility** dialog box.

6. Finally, you will see the summary window that contains the details of the task Mirage is about to start. Check if the details are correct and click on **Finish** to start the process.

7. If you now switch to the **Task Monitoring** view (labeled as 5), you will see the process running:

8. If you switch to the endpoint device, you will see what is happening at the client end during the pre-scan phase of the capture process. This is shown in the following screenshot:

---

For More Information:
Horizon Mirage will now complete the pre-scan. Once completed, we can move on to the next section and install the application.

**Stage 2 – installing the application**

Once Horizon Mirage has completed the pre-scan, the next step is to install the application that you want to capture as a layer. The endpoint will show that Horizon Mirage is now in the **Recording App Layer** mode, as shown in the following screenshot:

![Recording App Layer Screenshot](image)

The installation process for the application is no different from the way you would normally install the application. In this example, we are going to install Google Chrome.

Once installed, you should ensure that the application is updated and that any patches or updates are applied. If you want to make any configuration changes, you should do this after installation so that those changes get recorded too. Remember that these changes are computer-wide settings and do not apply to a specific user. User settings are not part of an app layer. We are now going to install the application.

As part of the lab environment, we downloaded the standalone installer for Google Chrome and copied it onto the file share on the domain controller. From the endpoint device, navigate to that folder and launch the installer. Follow all the steps to install the Chrome browser. Once completed, we will perform the final stage to complete the capture.

**Stage 3 – post-scan and application layer capture**

Now that the application has been installed, the final stage is for Horizon Mirage to perform a post-scan. The post-scan process takes another snapshot of the reference machine state so that it can compare this to the state captured during the pre-scan.
By comparing the two states of the reference machine, before and after application installation, Horizon Mirage can work out what has changed. Those changes are all of the components that are required to run the captured application. This could be the actual executable files, .dll files, filesystem changes, or registry settings.

Perform the following steps to perform a post-scan and application layer capture:

1. On completion of the application installation, switch back to the Horizon Mirage Management Console and the Task Monitoring window. Click on the Capture App Layer task that is currently running (labeled as 1) and right-click on it. Click on the Finalize App Layer Capture option (labeled as 2) as shown in the following screenshot:

2. You will see a dialog box pop up as Horizon Mirage finishes the post-scan process. Once that process has finished, you will see the Review Recorded Applications dialog box displayed with Google Chrome listed (labeled as 3) along with the version number and publisher, as shown in the following screenshot:

For More Information:
3. Click on the Create a new layer radio button (labeled as 4). The details will already have been prepopulated with the information captured from the application. Click on Next > to continue:

4. Horizon Mirage will now carry out the compatibility checks before capturing the application layer. I just want to draw your attention to the following example, which shows one of the most common errors encountered (labeled as 5) when capturing an application layer:

This message comes up if you haven't rebooted the capture machine after the application installation has finished.

5. You need to go back and reboot the machine even if the application install did not request a reboot. Once the endpoint has rebooted, return to the Check Compatibility dialog box and click on the Refresh button (labeled as 6). You should then see the message No compatibility issues detected. Click on Next > to continue.
6. On the Summary screen, check the task details and click on Finish. The application is now captured. Switch back to the endpoint to watch the process complete, as shown in the following screenshot:

![App Layer Capture](image)

7. When the task has completed, you will see that the app layer is available in the App Layers view in the Horizon Mirage Management Console, as shown in the following screenshot:

![App Layers](image)

The recording process only captures changes made to the machine and discards everything that is specific to a user. So you are not able to preconfigure an application if this data is saved in the user profile, for example, `%AppData%` or `HKCU`.

**App layer capture best practice**

The best way to capture app layers is to use a virtual machine as you would in ThinApp, where you have a setup capture machine. Having a VM allows you to take a snapshot of the machine before you start the application installation process, meaning that, after you have captured the app, you can roll back to the snapshot and start the next app capture from a clean machine.

It is always good practice to keep the app layer capture machine to the bare minimum. By this I mean it should have just the OS and latest patches and no other applications installed. It should also have all autoupdates switched off and not be joined to the domain. This ensures there are no unnecessary files, folders, or registry entries.

For More Information:

Managing device drivers

In the previous sections of this chapter we have discussed base layers and application layers. In this section, we will cover the final piece of building an image—device drivers. Although not explicitly called a layer, the device drivers are effectively treated as an IT-managed layer.

What is a Horizon Mirage driver library?

Horizon Mirage uses the concept of a driver library to store all the device drivers for the endpoints within your environment.

Best practice is to use the drivers supplied by the hardware vendor for each endpoint by either using restored media or, for the latest drivers, the vendor’s website.

Driver library folder structure

The easiest way to store and manage drivers is to create a folder structure with drivers grouped by hardware model, operating system, and the type of driver.

Work on a structure that makes sense for you and your own environment. You can have as many driver folders as you need to make the management of drivers easier. An example folder structure is shown in the following diagram where we have different hardware vendors, models, operating systems, and driver categories:

Driver library components and structure

The Mirage driver library is made up of the driver folders and driver profiles.

A driver profile can contain multiple folders and would typically relate to a particular endpoint from a particular hardware vendor. For example, you could have a driver profile for an HP 2510p laptop with individual device drivers located in a number of folders.

For More Information:
An example of the driver library components and how they are used is shown in the following diagram:

In this example, the driver folder that contains the generic Intel drivers is relevant to both the HP and Dell models and can therefore be a part of each of the individual driver profiles.

**Creating a driver library**

Before we start, we need to download the drivers for our endpoint devices. For the example lab environment, we are using an HP 2510p laptop and have downloaded the drivers from the HP website [http://tinyurl.com/ke5e5cu](http://tinyurl.com/ke5e5cu). I have shortened this URL to a TinyURL due to the length of the original.

In the example lab, we have copied the downloaded drivers into a shared folder on our domain controller. These would typically be stored in a file share on a file server.

For most manufacturers, the drivers are contained in a self-executing file or install program. For Horizon Mirage to work, we need just the drivers, hence the .dll, .ini, and .inf type files. In our HP example, the drivers are downloaded in a SP123.exe format. When you launch the SP... file, it creates a directory called SwSetup that contains the individual folders and files. These are the files we need.
The next thing we need to do is upload our device drivers onto the Horizon Mirage Server. In the previous example, we created a folder structure to match some of the endpoints in our environment. So, in this example, we are going to import drivers for the HP 2510p laptop.

1. Highlight the HP Laptops folder, right-click on it, and choose Import Drivers....
2. Enter the path to where you copied the drivers on the file share. In our example, the path is `\dc1\file share\drivers\hp` and is shown in the next screenshot. There is also a Keep original folder hierarchy checkbox. This means that the import will take the folder structure from the file share as is and copy it into Horizon Mirage, so it’s probably worthwhile creating your structure in your downloaded driver repository first.
3. In our example, it will take the structure below the `\hp\` directory level. Click on OK to start the import:

4. Check if the drivers have imported correctly. Click on the All folder under the Folders option. You will now see all the drivers that have been imported as shown in the following screenshot:

We have now imported the drivers; in the next section, we will look at how to build our driver profile.

For More Information:
Building a driver profile

With the drivers imported, we can build a driver profile and create a set of rules for how to apply this profile. To build the profile, perform the following steps:

1. Click on the Profiles option under the Driver Library category. Either right-click and select Add or click on the green + button at the top of the screen to add a profile.

2. In the Add Profile dialog box, enter a name for the profile (labeled as 1) and a description of the profile (labeled as 2). The next task is to tick the checkbox next to all the drivers (labeled as 3) that you want to add to this driver profile. This is shown on the following screenshot:

3. As part of the profile creation, you also need to build some rules that tell Horizon Mirage where to apply the driver profile to. Click on the Rules tab at the top. The rule we will build in this example will apply to the HP 2510p laptop.

4. In the Column field (labeled as 1), choose Model from the drop-down menu. In the Condition field (labeled as 2), choose Contains and in the Value field (labeled as 3), type 2510p. Then click on Apply (labeled as 4). In the Results field (labeled as 5), you will see that 2510p is now listed. Click on OK to finish:

For More Information:
Getting Started with Horizon Mirage Layering Technology

We now have a driver profile that we can use when deploying images. We will cover how we use this in more detail in *Chapter 5, Manage – Centralized Image Management and Deployment*.

**Summary**

In this chapter we have learnt what a Horizon Mirage layer is and how to describe the different types of layers that Mirage uses—base layer, app layer and driver profile.

You should now be able to build a reference machine and capture a base layer from it using best practices, capture an app layer, and finally build a driver profile.

In the next chapter we will learn how to put these layers to use in delivering single-image management with Horizon Mirage.

For More Information:

Where to buy this book


Free shipping to the US, UK, Europe and selected Asian countries. For more information, please read our shipping policy.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.