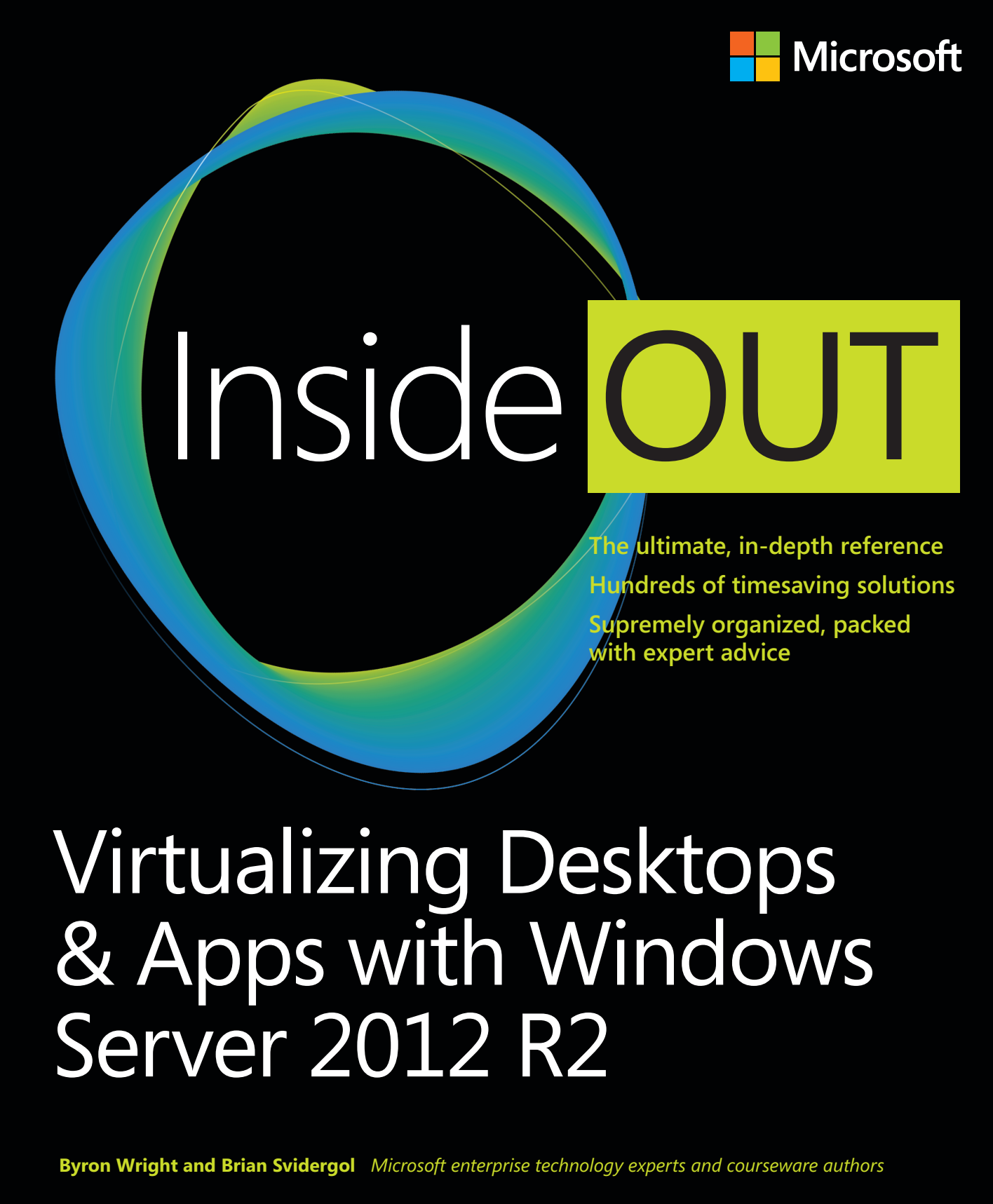
Microsoft
Inside OUT
The ultimate, in-depth reference
Hundreds of timesaving solutions
Supremely organized, packed with expert advice
Virtualizing Desktops & Apps with Windows Server 2012 R2
Byron Wright and Brian Svidergol
Microsoft enterprise technology experts and courseware authors

Microsoft

Virtualizing Desktops & Apps with Windows Server 2012 R2 Inside Out

Byron Wright
Brian Svidergol

PUBLISHED BY
Microsoft Press
A division of Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Library of Congress Control Number: 2014952206
ISBN: 978-0-7356-9721-8

Printed and bound in the United States of America.

First Printing

Microsoft Press books are available through booksellers and distributors worldwide. If you need support related to this book, email Microsoft Press Support at mspinput@microsoft.com. Please tell us what you think of this book at http://aka.ms/tellpress.

Acquisitions Editor: Alison Hirsch
Project Editor: Rosemary Caperton
Editorial Production: nSight, Inc.
Technical Reviewer: Todd Meister; Technical Review services provided by Content Master, a member of CM Group, Ltd.
Copyeditor: Ann Weaver
Indexer: Lucie Haskins
Cover: Twist Creative • Seattle

Contents at a glance

This page intentionally left blank

Table of contents

This page intentionally left blank

Introduction

If you want to learn about using Microsoft technologies to implement application and desktop virtualization, then this book is for you. If you need to support users who roam among multiple computers and platforms, then this book will help you identify, select, and implement options for user state virtualizations. If you are implementing application virtualization by using Microsoft Application Virtualization (App-V), then this book will help you learn how to design and configure an App-V deployment. It also will teach you how to sequence applications for deployment. If you want to implement virtual desktops to simplify hardware upgrades and support mobile users, then this book will teach you about both session-based virtual desktops and virtual machine–based (VM-based) virtual desktops. It also will teach you how to secure remote access to virtual desktops by using Remote Desktop Gateway (RD Gateway). Finally, to ensure that your application and desktop virtualization meets the performance requirements of your users, you will learn about monitoring application and desktop virtualization.

This book assumes that you have a working knowledge of Windows client and server administration or have access to that information. So, we assume that you understand basic facts like how to connect a computer to a domain, how to create and apply Group Policy, and how to work with management tools. This book focuses on implementing the application and desktop virtualization technologies that layer on top of Microsoft Windows Server 2012 R2. Each of the technologies covered in the book is examined from a real-life perspective. This book provides examples and recommendations for implementation.

System requirements

The following are the recommended system requirements necessary to implement the technologies described in this book in a development environment:

- A processor with Second Level Address Translation (SLAT) support
- Windows 8.1 (Enterprise edition recommended, Professional edition minimum)
- Windows Server 2012 R2
- Microsoft Desktop Optimization Pack (MDOP)
- At least 8 GB of random access memory (RAM)
- A hard drive with at least 128 GB of free space
- A CD-ROM or DVD-ROM drive
- A mouse or other pointing device
- A 1024 x 768 or greater monitor display

Acknowledgments

We'd like to thank Alison Hirsch at Microsoft Press for helping us get started on this book and ensuring that we were on the right track. We'd also like to thank Rosemary Caperton at Microsoft Press for getting us to the finish line. Finally, we'd like to thank Michael McMann, Brad Joseph, Phil Helsel, and Tony Jamieson from Microsoft Learning Experiences for working with us on a new development process.

Byron would personally like to thank Tracey, Samantha, and Michelle for tolerating the occasional late night required to get a large project like this done.

Brian would personally like to thank his wife Lindsay, his son Jack, and his daughter Leah for supporting him in another project. He would also like to thank Elias Mereb, Charles Pluta, and Bob Clements for their continued support and expertise.

Free ebooks from Microsoft Press

From technical overviews to in-depth information on special topics, the free ebooks from Microsoft Press cover a wide range of topics. These ebooks are available in PDF, EPUB, and Mobi for Kindle formats, ready for you to download at:

http://aka.ms/mspressfree

Check back often to see what is new!

Errata, updates, & book support

We've made every effort to ensure the accuracy of this book and its companion content. You can access updates to this book—in the form of a list of submitted errata and their related corrections—at:

http://aka.ms/VirtApps/errata

If you discover an error that is not already listed, please submit it to us at the same page.

If you need additional support, email Microsoft Press Book Support at *mspinput@microsoft.com*.

Please note that product support for Microsoft software and hardware is not offered through the previous addresses. For help with Microsoft software or hardware, go to *http://support.microsoft.com*.

We want to hear from you

At Microsoft Press, your satisfaction is our top priority, and your feedback our most valuable asset. Please tell us what you think of this book at:

http://aka.ms/tellpress

The survey is short, and we read every one of your comments and ideas. Thanks in advance for your input!

Stay in touch

Let's keep the conversation going! We're on Twitter: *http://twitter.com/MicrosoftPress*.

This page intentionally left blank

CHAPTER 4

Planning and implementing App-V

Virtualization usage has been expanding throughout the last several years. Although many IT administrators are now familiar with desktop and server virtualization, application virtualization is a lesser-known technology that is expanding rapidly. Simply put, application virtualization is a technology that delivers applications to computing devices in isolated containers without the need to perform a standard application installation on each computing device.

App-V provides a new way to deliver applications to users, a new way to centrally manage applications, and additional application capabilities for complex environments. Virtualizing applications, when appropriately planned and implemented, is an effective supplement to desktop and server virtualization and can help organizations move closer to cloud computing.

Overview of App-V

Microsoft Application Virtualization (App-V) is Microsoft's application virtualization technology. It is part of the Microsoft Desktop Optimization Pack (MDOP). MDOP is available to Microsoft Software Assurance customers and to Microsoft MSDN subscribers. It is a suite of desktop optimization applications that help IT administrators deploy, manage, and troubleshoot desktop computing environments. The newest version, at the time of this writing, is App-V 5.0 Service Pack 3.

Benefits of App-V

You know that App-V is Microsoft's application virtualization technology, and you have an idea of how to access it. But, to fully appreciate it, you should have a good understanding of its capabilities. The core capabilities that you can deliver with App-V are the following:

- **Run multiple versions of applications at the same time** You can use App-V to run different versions of applications concurrently on the same computer. For example, it's possible to run Internet Explorer 9, Internet Explorer 10, and Internet Explorer 11 concurrently if they all are set up as App-V applications; otherwise, you can't run multiple versions of Internet Explorer side by side on the same computer. It's also possible to

use App-V in conjunction with Remote Desktop Services (RDS). This allows users to run applications side by side on Remote Desktop Session Host (RD Session Host) servers.

- **Minimize application conflicts** Sometimes two or more applications conflict with one another because of dynamic-link library (DLL) or application programming interface (API) conflicts. When you deliver these applications as App-V applications, however, there isn't a conflict because each App-V application runs in its own isolated environment.

- **Simplify application removal** App-V applications do not install locally, which makes it easier to remove them. Clean removal isn't always possible with applications that install directly on Windows-based clients, even if an application has been designed to remove all files and settings when it uninstalls. Virtualized applications are removed easily after a user signs out from a computer.

- **Simplify application upgrades** Instead of upgrading a locally installed application on all computers in your organization with a hotfix, service pack, or new version, the modular nature of virtualized applications means that you can replace one version of an application with an updated version with less effort. You only need to update an application on the App-V server(s), and clients receive the latest version of the application from the publishing server the next time they launch the application.

- **Minimize license-compliance risks** App-V has application metering functionality that enables you to detect every use of a virtualized application to ensure license compliance.

- **Scale infrastructure** You can add publishing servers to an App-V deployment as necessary to ensure that service levels are maintained as demand grows.

- **Take advantage of client hardware resources** Even though App-V applications do not install locally, they can use a local computer's processor and RAM resources. In environments where client computers have adequate hardware resources, this can create a better experience for users than running applications on an RD Session Host server would.

- **Allow users to use roaming applications** If applications stream with App-V rather than install locally, users can sign in to any computer that has an App-V client and quickly access their applications. You also can configure App-V to work with user virtualization to allow users to have application settings and data for App-V applications that roam across client computers.

- **Give users quick access to their applications** Imagine a scenario in which a user needs to have an application installed. In many environments, a routine application installation requires several steps, such as copying the installation files to the client

computer, manually performing the installation, manually performing application updates, and cleaning up the installation files. With App-V, you can just add the users to a group, and they can immediately run the application through App-V.

Differences between standard and virtualized applications

In a traditional IT environment, applications are installed on each computing device. For this discussion, we'll refer to the traditional installed application as a standard application. During a standard application installation, the computing device is modified to meet the application's requirements. For example, a common installation of a standard application would include the following steps:

The application's files are copied During the installation, the installer copies the necessary application files to a specified folder. During this process, the files often are uncompressed first.

The registry is modified to support the application Most applications add to the registry to support the installation and running of the application. In addition, it's common for existing registry keys and values to be modified to support automatic application startup on boot and to set any required dependencies.

DLL files are registered Often, .dll files must be registered during the installation.

Permissions are configured You often need to set permissions to configure which user(s) can run the application, which profiles the shortcuts are added to, and which users can modify configuration files.

Shortcuts are added to the device Many applications add a shortcut to the Start screen, desktop, and taskbar. Additionally, it is common for applications to add a tray icon for quick user access.

App-V applications, which are virtualized, aren't installed onto computing devices. Instead, they are packaged for deployment. In App-V, packaging an application for deployment is known as sequencing an application. Later in this chapter, we'll discuss sequencing further. For now, let's examine the differences between an App-V application deployed to a computing device and a standard application installed locally on a computing device. For applications deployed with App-V, the following characteristics highlight the different methods by which applications interact with the computing device:

App-V applications run in their own isolated environment This reduces application conflicts and application crashes impacting other applications or the operating system and provides the foundation to allow multiple versions of applications to run at the same time.

App-V applications use a virtual registry This reduces installation difficulties and application conflicts and improves the stability of a client computer because applications aren't sharing a single registry.

App-V applications use virtual file systems This reduces conflicts by ensuring that applications don't overwrite shared files.

App-V applications use virtual services This also reduces conflicts because virtual services do not have dependencies on other non-virtual services and provide isolation from other virtual services.

App-V applications are installed by being packaged and delivered to computing devices. App-V applications only need to be updated once, by updating the application's package on the App-V server. By reducing the number of times you have to install and update an application, you can greatly reduce the number of hours required to manage your application infrastructure.

Placing and functionality of the virtualization engine

Virtual applications require access to resources on a host computer. Access typically uses a system request that a virtualization engine needs to intercept. The engine provides functions for capturing an application's system call and manipulating it where needed.

Interception in user mode

Applications run in user mode and perform operations on system services that reside in kernel mode. When an interception occurs in user mode, the virtualization engine must be placed over the native API layer. Some applications call the functions of the native API directly instead of by using Windows API.

One advantage of the user mode strategy is that the virtualization engine doesn't have to filter between system calls of different applications. The virtualization engine only works with an application that it built for the virtual environment.

Interception in kernel mode

In contrast, if the virtualization engine operates in kernel mode, it can intercept all system calls before they reach Windows executive services. The advantage is that it is easy to loosen the isolation of applications and let them share the same virtual environment. This enables you to create dependencies between packages, similar to working with middleware or plug-ins. However, the agent that builds the virtual environment must install natively on a computer to gain the required privileges for an interception in kernel mode. In this case, all applications that run on the host machine are affected. Furthermore, the virtualization layer captures system calls from all applications and must filter the processes to respond to every call correctly.

App-V implements a hybrid approach by identifying the locations of resources that the application requires. While the application is running, the virtualization engine ensures that function calls are modified only if they request a path inside the virtual file system or a key inside the virtual registry. When the engine runs in kernel mode, it also must check which package to consult to find the rules, because it will receive function calls from several running applications that belong to different packages.

The virtualization layer must intercept and redirect requests to the file system and registry to virtual counterparts that contain the files and keys that belong to a certain application. To decide which function calls to intercept and which to handle as usual, the virtualization layer needs rules. These rules generate when the App-V Sequencer, which collects all the files and registry keys that the installer creates or modifies, monitors an application's installation. Then, at the application's runtime, the virtualization engine ensures that function calls are modified only if they request a path inside the virtual file system or a key inside the virtual registry.

Application virtualization infrastructure

The infrastructure technologies of an App-V deployment are extensive, based on the deployment model that you choose. They work together to provide the complete suite of App-V technologies. It is important to familiarize yourself with all of the technologies, the typical life cycle of a virtual application, the deployment models, and some of the characteristics of packages and content packages.

App-V application life cycle

To effectively manage your virtual application infrastructure, you need to plan for the life cycle of your virtual applications. Without an effective life cycle, you may end up with application sprawl—a situation in which you have too many applications to manage. To avoid this, you should spend ample planning time designing and documenting an operational framework for your virtual application life cycle. In this section, we'll discuss the four phases of the virtual application life cycle: sequencing, publishing and deployment, updating, and termination.

Application sequencing

The App-V Sequencer is one of the primary applications of an App-V deployment. You use it to create virtual application packages. Then, you deploy the packages to your App-V clients. It is important to consider the following before you begin deploying the sequencer and sequencing applications:

- **Prerequisites** If the computer that runs the sequencer isn't running Windows 8 or newer or Windows Server 2012 or newer, then it must have the following software installed prior to installing the sequencer. Note that the App-V client installation automatically will install the Visual C++ prerequisites.
  - Visual C++ Redistributable Package for Visual Studio 2013
  - Visual C++ 2005 Redistributable
  - Microsoft .NET Framework 4
  - Windows PowerShell 3.0
  - Microsoft KB2533623 hotfix
- **Windows 8 or newer or Windows Server 2012 or newer** If the computer that runs the sequencer is running Windows 8 or newer or Windows Server 2012 or newer, it already has the prerequisite software.
- **Match the hardware and software** The computer that runs the sequencer should have a hardware and software configuration that matches the App-V client computers. For example, if all of your App-V client computers run Windows 8.1, you should install the sequencer on similar computer hardware that runs Windows 8.1.
- **Use a virtual machine** When possible, use a virtual machine (VM) as the computer that runs the sequencer. This allows you to take a snapshot of the VM prior to sequencing an application. Then, after you finish sequencing an application, you should revert the VM to the snapshot. This allows you to sequence an application with the same baseline configuration, which minimizes issues. Although the sequencer will allow you to sequence multiple applications without reverting the sequencer to a baseline configuration, it will warn you that you may encounter issues.
- **Multiple sequencers** If you have multiple operating system versions running the App-V client, you seriously should consider having multiple sequencers. This enables you to sequence applications on the same operating system to which you will deploy or stream the applications.

Application publishing and deployment

After you sequence an application, you need to publish it and deliver it to the clients. Publishing a virtual application makes the application available to App-V clients. Before the actual publishing process, you need to be aware of the different methods that you can use based on the type of App-V deployment you have.

Stand-alone deployment model

In the stand-alone model, you need to add the App-V package of the application to clients. You can do this by using the Add-AppVClientPackage Windows PowerShell cmdlet. For example, if you have an App-V package named 7-Zip.appv located at \\tt-util-01\share\7-Zip.appv, you can run the following command to add the package:

```
Add-AppVClientPackage –Path \\tt-util-01\share\7-Zip.appv
```

After running the command, the output will show the details of the package added. In fact, the output is the same as if you were to run the `Get-AppVClientPackage –Name 7-Zip` Windows PowerShell command. The output is shown in Figure 4-1.

```
PackageId            : 30252a91-5cc5-419a-bcfb-645c942df001
VersionId            : 5c57388f-d692-49f0-a609-25dcb6e3e690
Name                 : 7-Zip
Version              : 0.0.0.1
Path                 : \\tt-util-01\share\7-Zip.appv
IsPublishedToUser    : False
UserPending          : False
IsPublishedGlobally  : False
GlobalPending        : False
InUse                : False
InUseByCurrentUser   : False
PackageSize          : 4872031
PercentLoaded        : 10
IsLoading            : False
HasAssetIntelligence : False
```

Figure 4-1 Adding a package

In the output, notice that the IsPublishedToUser property is set to False. This is an important detail because while it is set to False, the user won't see or be able to use the virtual application. After you've added the package, the client will begin receiving the files that make up the package. The data will be stored locally on the client. You can look at the %ProgramData %\App-V directory to see data from the package.

The next step is to publish the application. However, you can't publish an application until the application has been added to the client. Once you are ready to publish, you can use the `Publish-AppVClientPackage` Windows PowerShell cmdlet. For example, if you added a package named 7-Zip, you can publish it to the client by running the following Windows PowerShell command:

```
Publish-AppVClientPackage –Name 7-Zip
```

Once you run that, the output will be similar to when you added the package. The key difference is that the IsPublishedToUser property will be updated to a value of True, as shown in Figure 4-2.

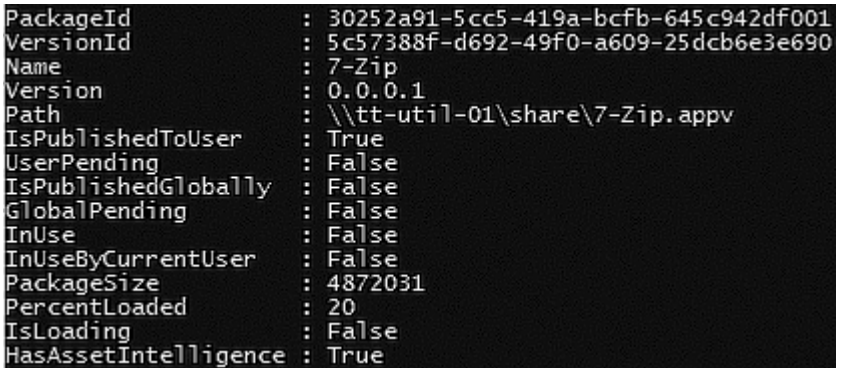

Figure 4-2 Publishing a package

After you publish the package, the application becomes available to the user. If shortcuts are configured for the package, they will begin to be displayed after publishing the package.

Full infrastructure model

In the full infrastructure model, the publishing process is a bit more automated, especially when you are dealing with a large number of virtual applications. The high-level process to publish an application is shown below. Note that the first step involving Group Policy is a one-time step in a new full infrastructure model and would not need to be performed for each application that you want to publish.

1. Create a new Group Policy Object (GPO) and modify the App-V–related GPO settings for your environment. Link it to the computers that have the App-V client software. At a minimum, you should configure an App-V publishing server in the GPO so that App-V clients will automatically be configured for a publishing server.

2. Add the application package on the App-V management portal. Configure the settings based on your environment. At a minimum, you need to ensure that the users have access to the application and that the application is published. You can configure access by right-clicking the application and then clicking Edit Active Directory Access.

3. Publish the application. To publish an application from the management portal, right-click it and then click Publish. You also can publish an application by using Windows PowerShell. For example, to publish an App-V package named 7-Zip, you can run the `Publish-AppvServerPackage –Name 7-Zip` command.

4. Sync the clients or wait for the next automatic sync. To immediately sync a client with an App-V publishing server named TT-UTIL-01, run the `Sync-AppvPublishingServer –Name TT-UTIL-01` Windows PowerShell command.

Configuration Manager model

If you use App-V and Configuration Manager to manage and deliver applications, then the steps to publish an application are different from other App-V models. The following high-level steps describe the process of publishing by using Configuration Manager.

1. In the Configuration Manager console, create a new application.
2. On the General page of the Create New Application Wizard, configure the application type to be Microsoft Application Virtualization (App-V) Client 5.0. Then, browse to the location of the .appv package that you want to publish. Also, ensure that the user and deployment .xml configuration files are in the same location as the .appv file. By default, the name of the user configuration file is <app>_UserConfig.xml. For example, if the name of the .appv file is 7-Zip.appv, then the name of the user configuration file is 7-Zip_UserConfig.xml. By default, the name of the deployment configuration file is <app>_DeploymentConfig.xml. For example, if the name of the .appv file is 7-Zip.appv, then the name of the deployment configuration file is 7-Zip_DeploymentConfig.xml.
3. Complete the Create New Application Wizard by specifying application details or maintaining the default values.
4. Distribute the application to Configuration Manager distribution points so that clients can obtain the application from the nearest distribution point.
5. Deploy the application to clients. You can deploy the application as a streaming application if you have a full infrastructure App-V deployment model. Otherwise, you can opt for the download and execute method. Each method has pros and cons. See *http://technet.microsoft.com/en-us/library/jj822982.aspx* for more information on the two deployment methods.

Application update

One ongoing maintenance task that you'll need to perform is updating applications. An application update, sometimes called an application upgrade, occurs when a software company releases a newer version of the application. Often, companies release newer versions of software to fix security issues or provide new or enhanced functionality. In a standard application deployment, in which applications are installed on every computer, you need to perform the update on every computer. In a virtualized application deployment, in which applications are packaged and delivered by App-V, you only need to update the packaged application on the App-V Sequencer and then update the distribution method with the updated package.

To update an existing App-V application with the App-V Sequencer, perform the following steps:

1. Run the App-V Sequencer on the client computer that you use for sequencing.

2. Click the Modify An Existing Virtual Application Package option, shown in Figure 4-3.

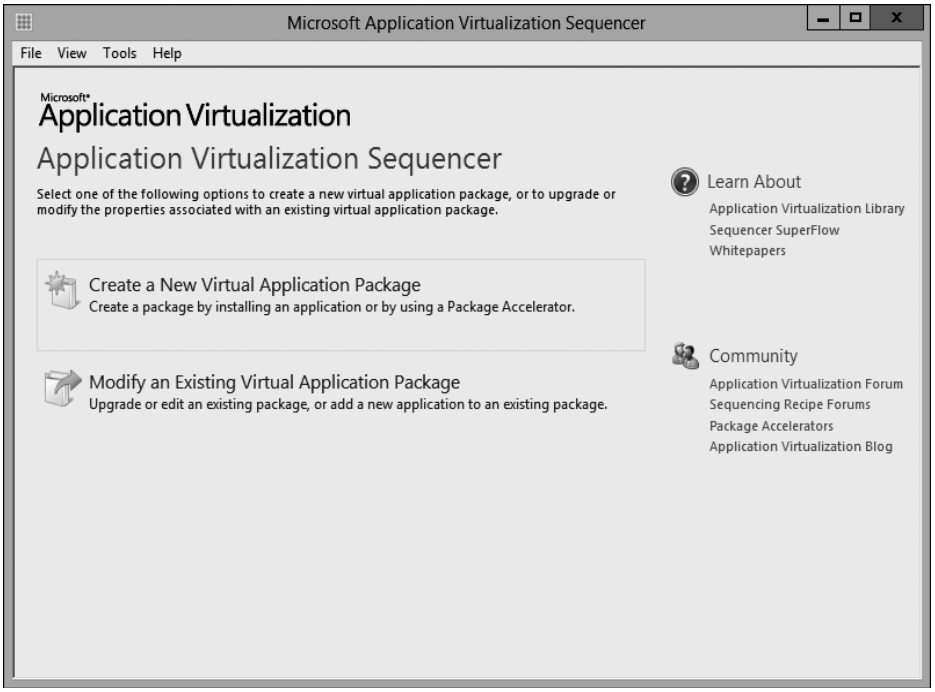

Figure 4-3 App-V Sequencer

3. On the Select Task page, shown in Figure 4-4, keep the Update Application In Existing Package option selected and then click Next.

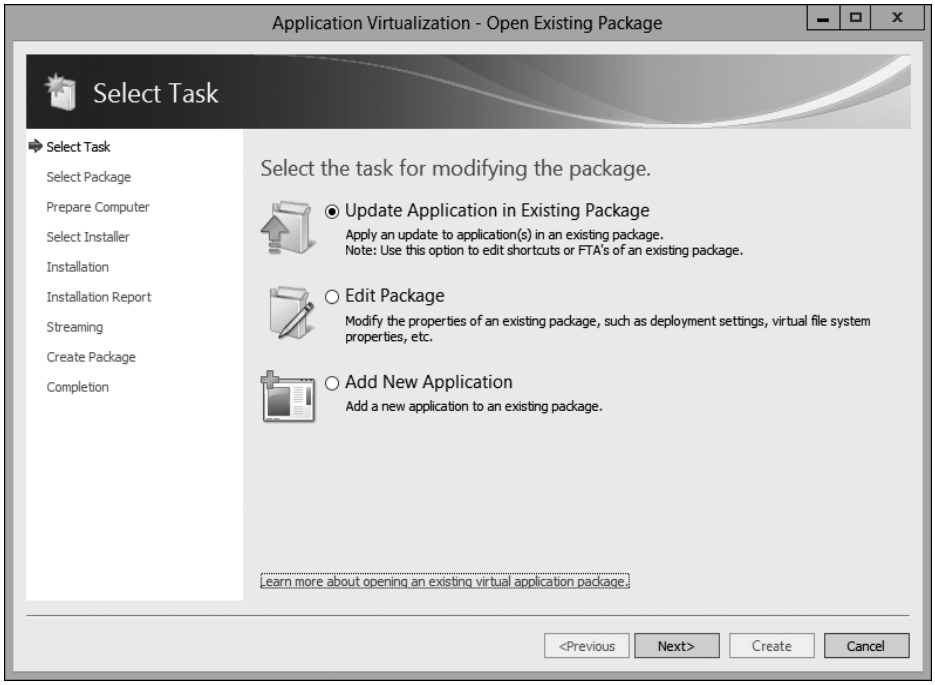

Figure 4-4 App-V Sequencer Select Task page

4. On the Select Package page, shown in Figure 4-5, click Browse and navigate to the existing App-V package file (.appv), click the file, click Open, and then click Next.

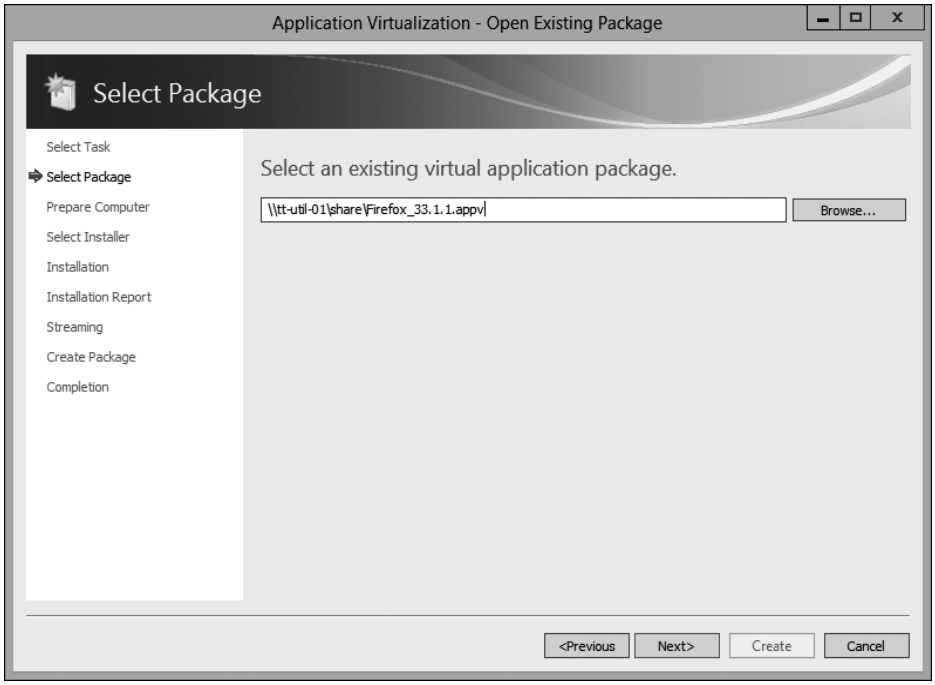

Figure 4-5 App-V Sequencer Select Package page

5. On the Prepare Computer page, shown in Figure 4-6, if the computer is ready to create a package, click Next. If issues are listed, remediate the issues if necessary and then click Refresh until the computer is ready to create a package. Click Next.

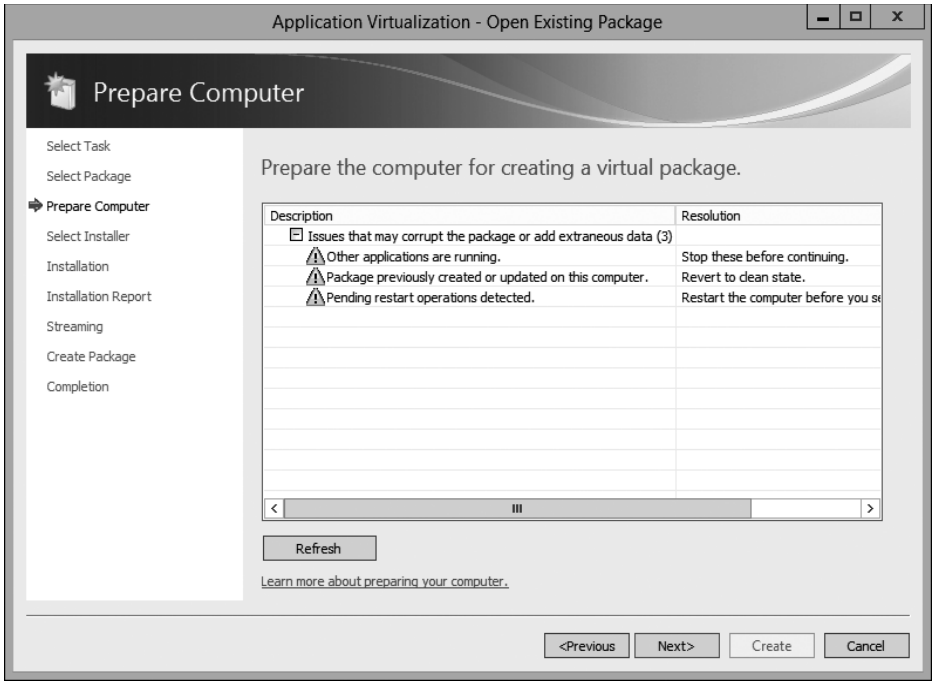

Figure 4-6 App-V Sequencer Prepare Computer page

6. On the Select Installer page, shown in Figure 4-7, click Browse and navigate to the install file for the application update. Click the installer file, click Open, and then click Next.

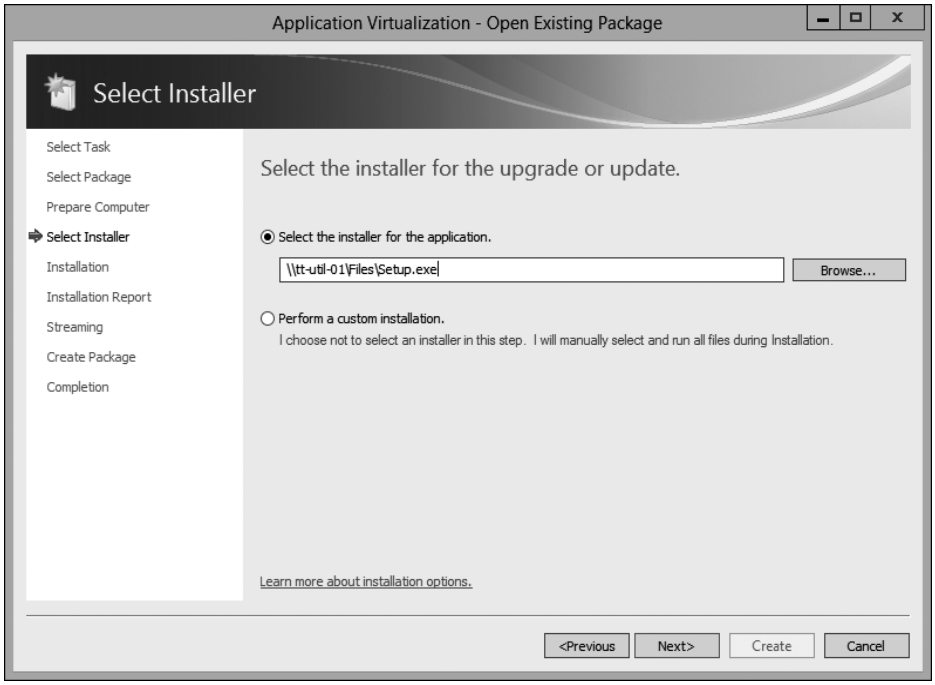

Figure 4-7 App-V Sequencer Select Installer page

7. The application update installation will begin. Update the application based on the installation program. When finished, select the I Am Finished Installing check box, as shown in Figure 4-8, and then click Next.

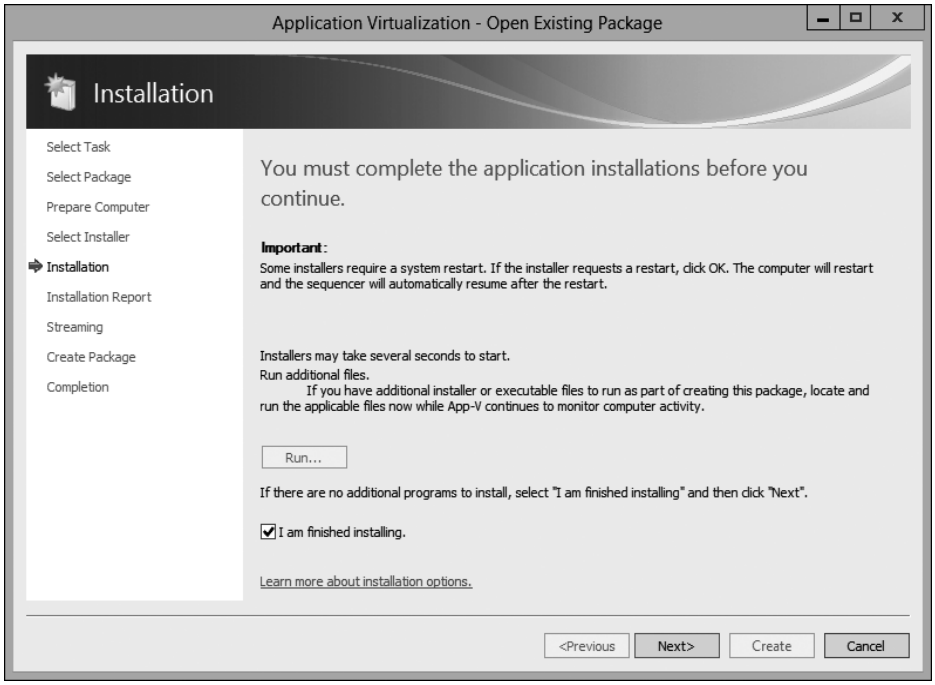

Figure 4-8 App-V Sequencer Installation page

8. On the Installation Report page, shown in Figure 4-9, if the update was successful, the App-V wizard should report that there weren't any issues detected. Click Next.

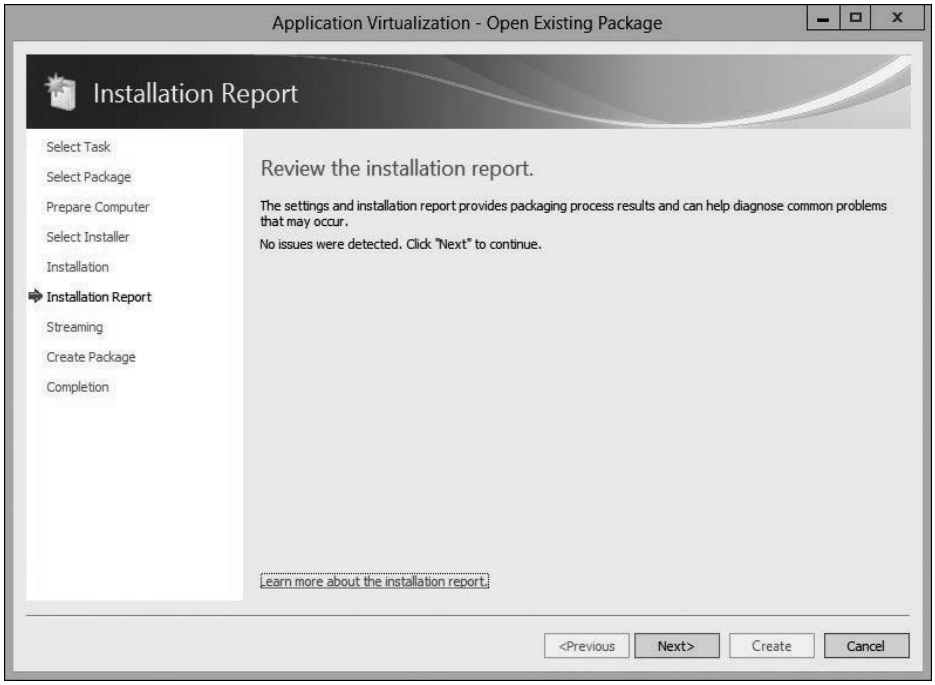

Figure 4-9 App-V Sequencer Installation Report page

9. On the Prepare For Streaming page, shown in Figure 4-10, click Run All. Then, perform any needed first-run application configurations. When finished, close the updated application and then click Next in the App-V wizard.

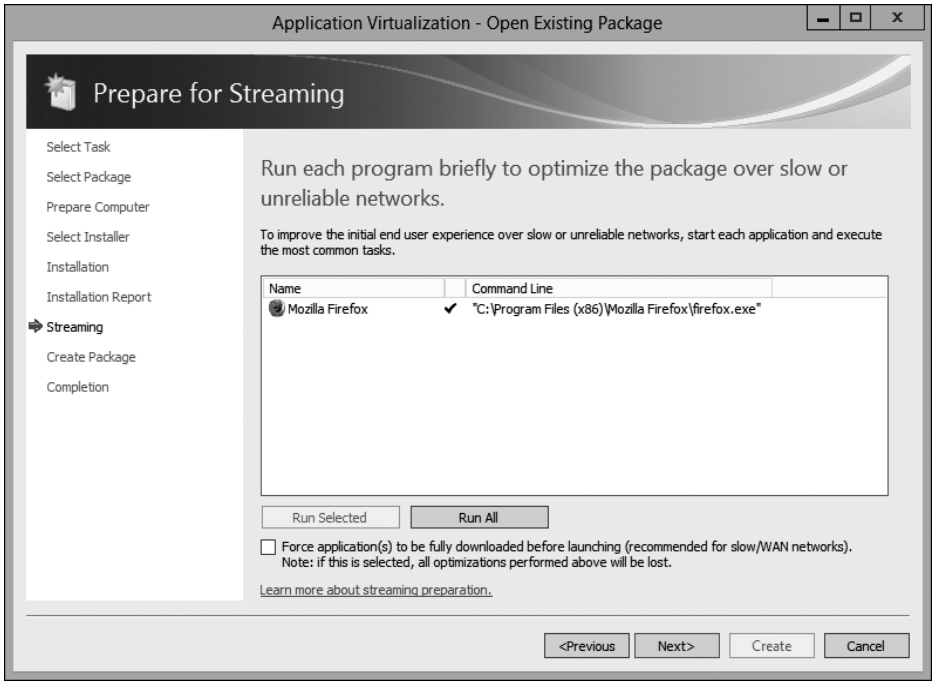

Figure 4-10 App-V Sequencer Prepare For Streaming page

10. On the Create Package page, shown in Figure 4-11, enter a location to save the updated application package and then click Create.

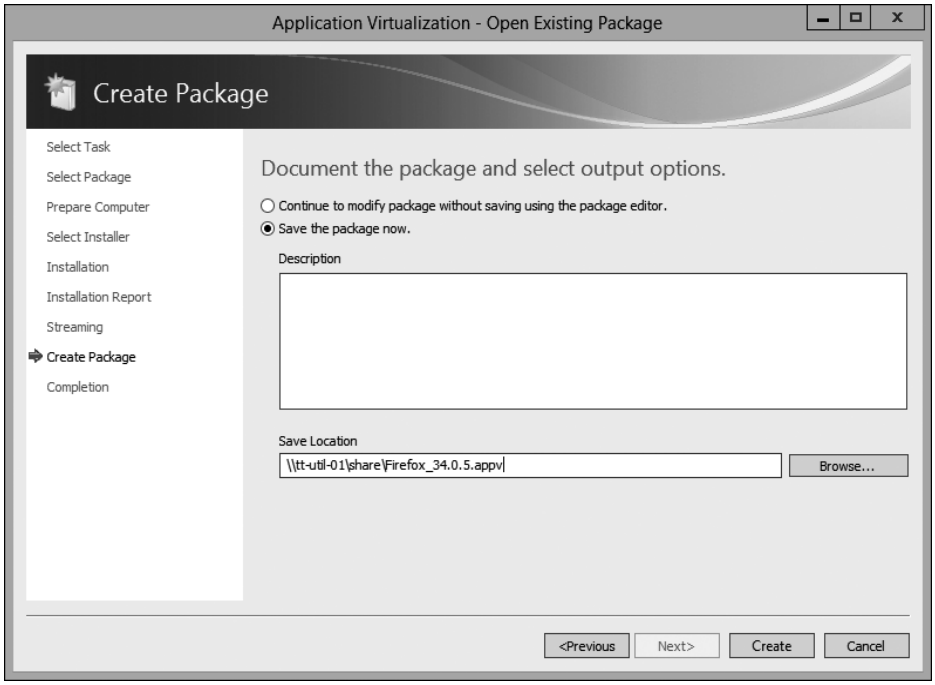

Figure 4-11 App-V Sequencer Create Package page

11. After the package creation completes, click Close. Next, you need to make the updated .appv package available to publishing servers, App-V clients, or Configuration Manager, depending on your App-V deployment model.

Application termination

As part of your routine application maintenance, you'll need to temporarily and permanently remove applications from your environment. Sometimes, you'll need to remove applications from specific users. Other times, you'll need to permanently remove an application from all of your users. You should understand the options available to you for handling these tasks. Let's look at a few scenarios and walk through the process.

Removing an application from a small number of users

There are many situations in which you will need to remove an application from one user or a small group of users. For example, if you were running end user pilot testing for a new application, you may want to remove users after they've completed their testing. There are multiple methods for doing this, but we'll focus on one of the most common methods in an environment with the App-V full infrastructure deployment model. In this model, you should be assigning application access based on Active Directory Domain Services (AD DS) security

groups. When it is time to remove one or more users, the simplest method is to remove those users from the security group. The virtual application remains available in App-V, but only for users with the appropriate access rights. In situations in which you are completely and permanently removing an application from all users, you have a few options:

In the App-V Management Console, you can delete the application by right-clicking it and then clicking Delete, as shown in Figure 4-12.

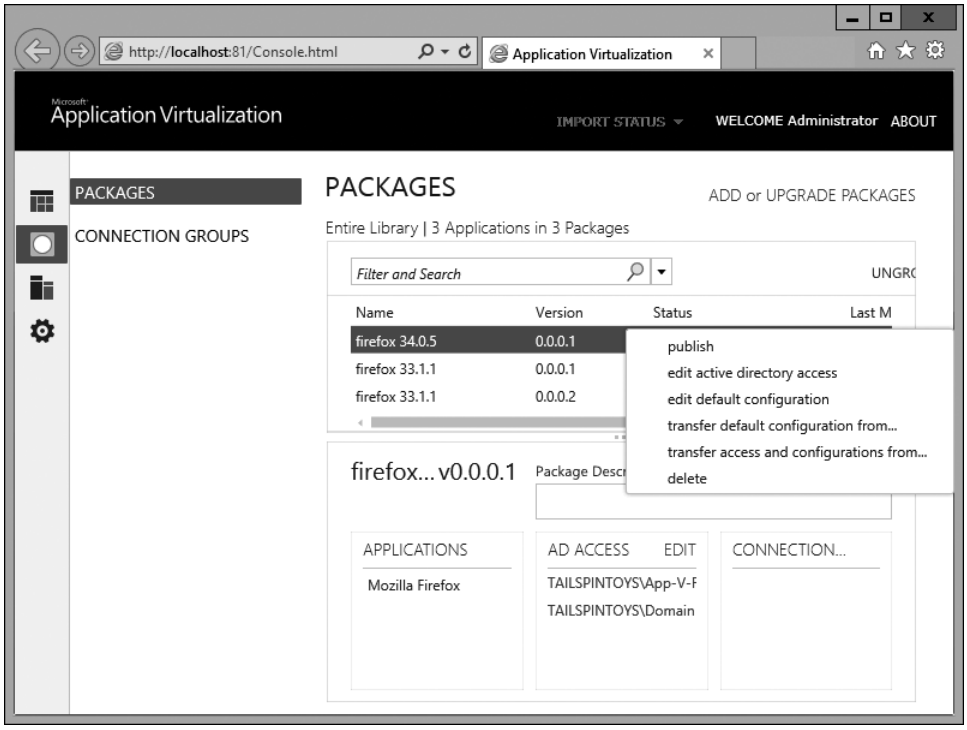

Figure 4-12 App-V Management Console showing Packages workspace with two published applications

From that point on, new App-V clients can't launch the deleted application. Users who already have the application or have used the application will still see it and be able to launch it. You should be aware of a couple of things that cause this. You delete apps from the App-V Management Server. The publishing server, by default, refreshes the list of published applications every 10 minutes. Clients get the list of applications from the publishing server. This 10-minute refresh occurs even if the management server and publishing server run on the same server! Even if you sync the App-V client with the publishing server immediately following an application deletion, it won't matter because the publishing server isn't yet aware that the application was deleted. In most production environments, this 10-minute refresh cycle is

okay. However, if you need to reduce the amount of time between refreshes or manually perform an immediate refresh, you have the following options:

On the publishing server, stop the AppVPublishing application pool, shown in Figure 4-13, in Internet Information Server (IIS).

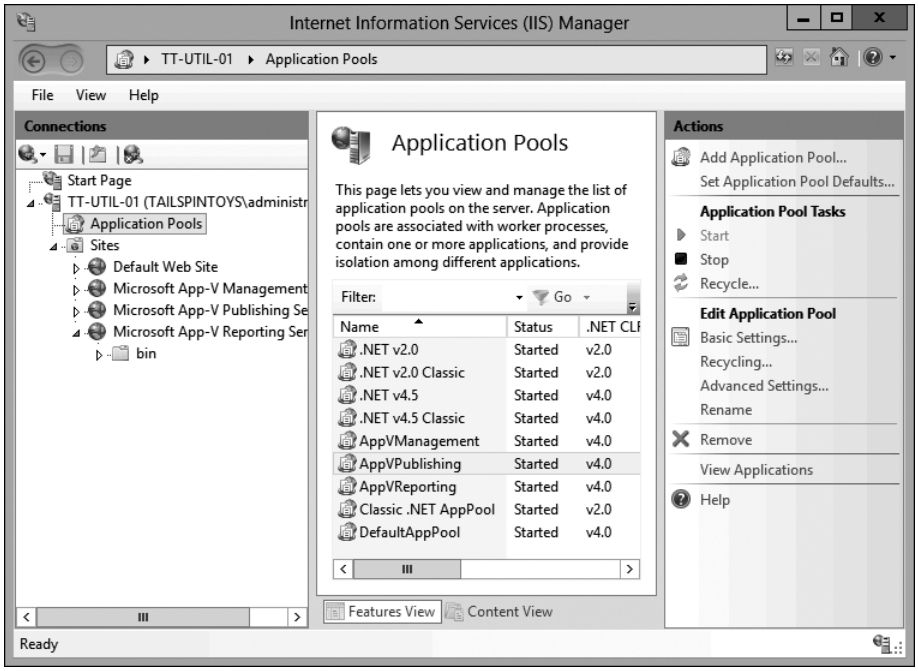

Figure 4-13 IIS application pools for App-V

Then, start the AppVPublishing application pool. This kicks off an immediate refresh of the published apps. Thereafter, you should perform a client sync to complete the removal process. Otherwise, you must wait until the next client sync.

Alternatively, you can adjust the refresh intervals in the registry on the publishing server. To do this, go to the HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\AppV\Server\PublishingService key and reduce the value of the PUBLISHING_MGT_SERVER_REFRESH_INTERVAL entry. The default is 600 seconds. You can reduce it to something much smaller, such as 10 seconds, as shown in Figure 4-14.

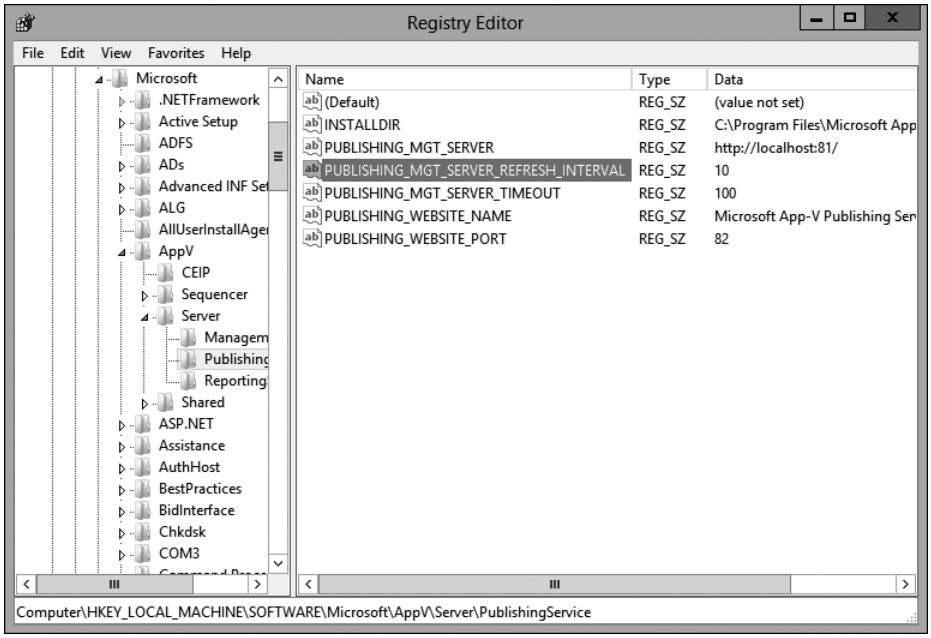

Figure 4-14 App-V publishing registry values

App-V technologies

An App-V 5.0 deployment includes a number of technologies, some of which must be present in all App-V 5.0 models, and some of which are used by only specific App-V deployments. These technologies are the management server, publishing server, management server database, reporting server, reporting server database, Sequencer, and App-V client.

From the Feature Selection page of the Microsoft Application Virtualization (App-V) Server 5.0 Setup Wizard, you can select which servers and databases you want to install, as shown in Figure 4-15.

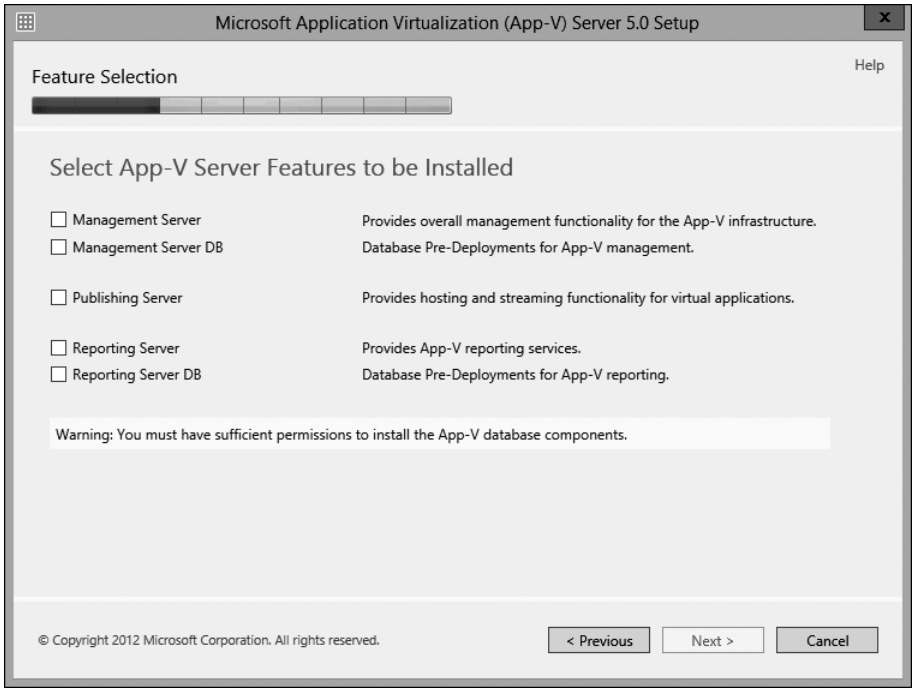

Figure 4-15 The Feature Selection page showing the available App-V server features

Management server and management server database

The management server hosts a Microsoft Silverlight–enabled web application that App-V administrators use to manage an App-V infrastructure. When you use App-V 5.0, all management occurs through the web console or Windows PowerShell, unlike previous versions of App-V. A computer must have Silverlight installed to access the console.

You can use the management server to work with the following objects:

- **Packages** You can import packages in the App-V file format, which uses the .appv file extension. You then can publish the packages to App-V publishing servers. You also can use the console to configure package security. In Figure 4-16, the management console shows the Packages workspace with a couple of published applications.
- **Connection groups** Connection groups make it possible for virtual applications to interact with one another. You also can use the console to configure security for connection groups.

- **Publishing servers** You can authorize publishing servers. You perform this task from the Servers node. You must specify publishing servers in the *domain\computername* format.
- **Administrators** You can add and manage App-V administrators. Administrators are able to import and publish packages, configure connection groups, and add publishing servers.

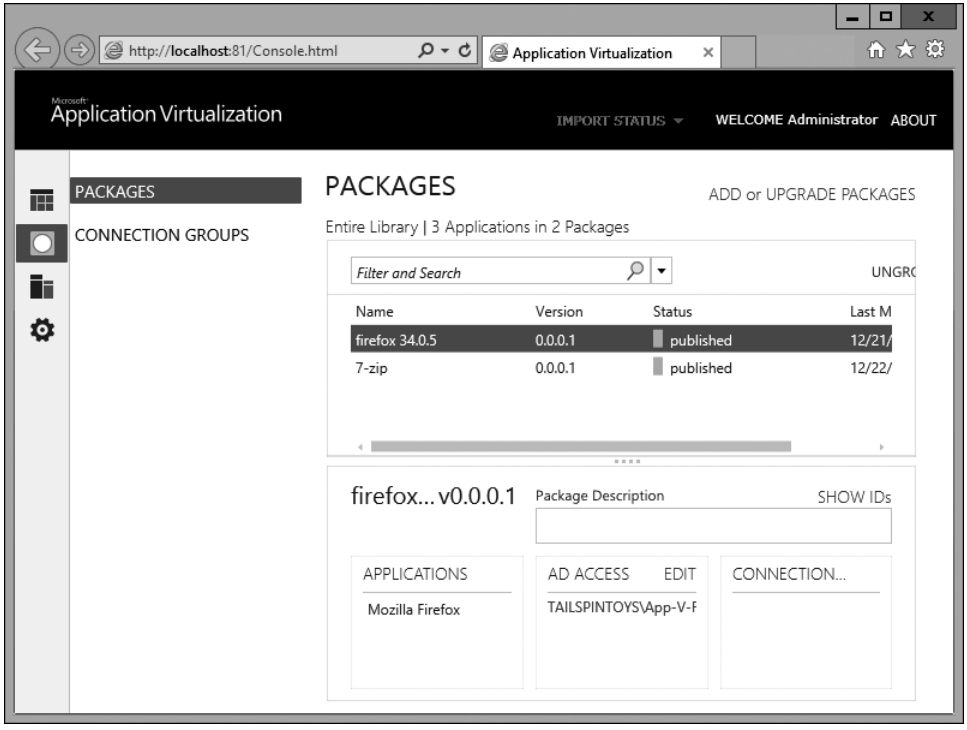

Figure 4-16 App-V Management Console showing Packages workspace with two published applications

The management server database stores the App-V configuration and data settings. The database is hosted on a Microsoft SQL Server instance that is running SQL Server 2008 Service Pack 2 (SP2), SQL Server 2008 R2, or SQL Server 2012. You should install the management server database separately from the management server, and if you do, you need to deploy the database first and then specify its location when you deploy the management server.

Publishing server, sequencer, and App-V client

The publishing server is a web server, built on IIS, which hosts and deploys applications for App-V clients. In previous versions of App-V, a publishing server was called a streaming server.

You deploy App-V applications to a publishing server in the App-V full infrastructure model by using the App-V Management Server console or by using Windows PowerShell.

You can deploy a management server and a publishing server on the same computer. You also can deploy a publishing server on a separate computer as long as you already have deployed an existing management server. This is different from previous versions of App-V, in which it was possible to deploy App-V streaming servers without having deployed a management server.

Previous versions of App-V streamed applications by using the Real-Time Streaming Protocol (RTSP). App-V 5.0 applications stream from a publishing server by using HTTP, HTTPS, or SMB. When you configure a publishing server, you specify a TCP/IP port that is used to stream applications. You subsequently use this port address when you configure Group Policy for use with App-V clients so that clients can access published App-V applications.

Publishing servers are useful in environments that have multiple geographic locations and are connected by wide area network (WAN) links. If you have the App-V full infrastructure model with a management server, publishing server, and database server at the headquarters office, clients outside the headquarters office likely would have a degraded experience compared to having App-V publishing servers locally at their office. In this case, you should deploy publishing servers at branch offices so that clients at those branch offices would be able to receive applications directly from the publishing server in the branch office, rather than across a WAN link from a publishing server in another office.

You can configure a client with the addresses of up to five publishing servers when you use Group Policy, as shown in Figure 4-17.

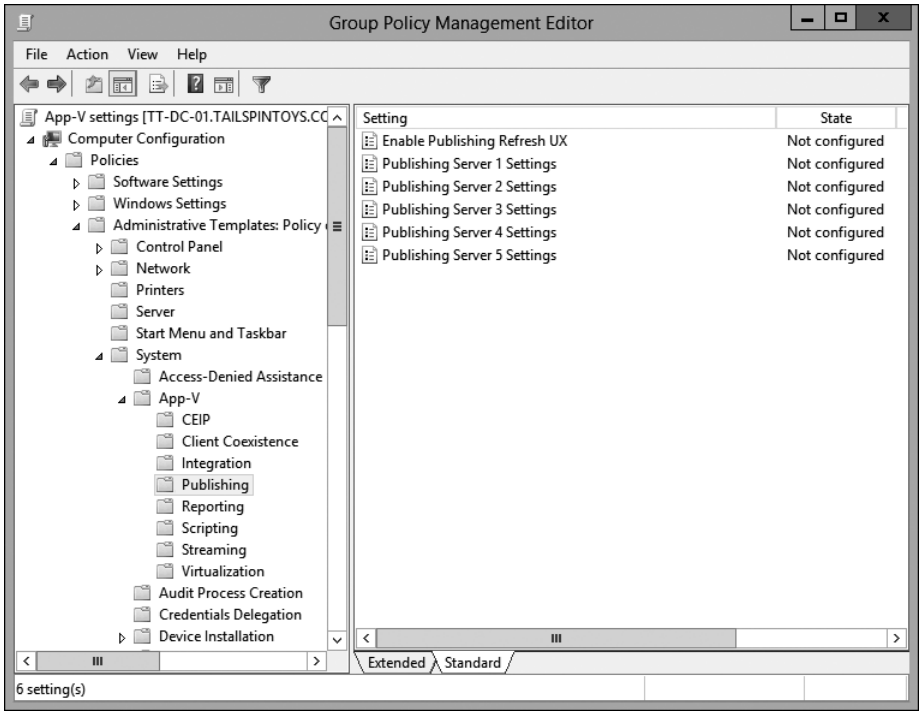

Figure 4-17 GPO settings for publishing

When configuring Group Policy to support clients in branch offices, provide the address of both the local publishing server and a second publishing server to use in case the local publishing server fails. The diagram in Figure 4-18 represents an App-V environment with a headquarters office and two branch offices. Each App-V client is configured to use the local App-V publishing server in its local office and also is configured with a secondary App-V publishing server in an alternate site.

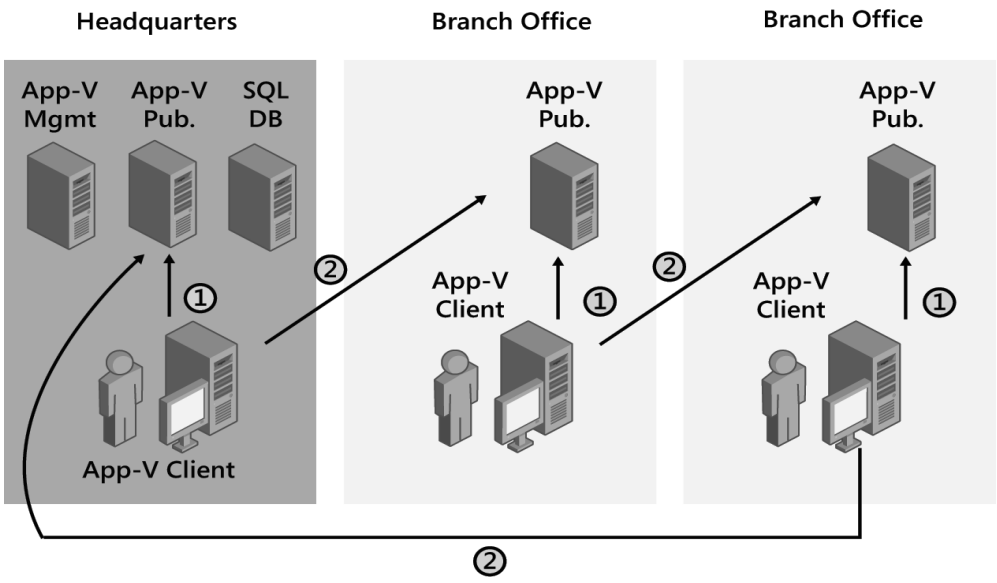

Figure 4-18 App-V environment with multiple publishing servers

You can use the Sequencer to create applications that can run under the App-V client. You should sequence an application on a Sequencer that is configured on the operating system on which the application will run. This helps minimize compatibility issues for your virtualized applications.

The App-V 5.0 Sequencer can install only on computers that run Windows 7 Service Pack 1 (SP1), Windows Server 2008 R2 SP1, Windows 8, Windows 8.1, Windows Server 2012, or Windows Server 2012 R2. In most cases, you should configure a client operating system as a sequencer. This is because most of your virtualized apps likely will be on a client operating system.

The App-V Sequencer has the following requirements:

- Windows 7 Service Pack 1 (SP1), Windows Server 2008 R2 SP1, Windows 8, Windows 8.1, Windows Server 2012, or Windows Server 2012 R2
- Microsoft .NET Framework 4 or 4.5
- Windows PowerShell 3.0 or newer
- Update for Windows KB2533623

The App-V client must be installed on computers before those computers can run virtualized applications. The App-V client supports both x86 and x64 operating system architectures:

- On x86 operating system architecture, only x86 applications are supported.
- On x64 operating system architecture, both x86 and x64 applications are supported.

You need to deploy a separate App-V client on RD Session Host servers. We look at that client in an upcoming section in this chapter titled "App-V for Remote Desktop Services client."

Reporting server and reporting server database

The App-V reporting server, an optional feature, is the built-in reporting feature that you can use for reporting on virtual application usage. The reporting server records the following information:

- Application use, including launch status, startup times, and shutdown times
- Client information such as the host name, client version, operating system version and type, processor architecture, and operating system service pack level
- Package information, such as the package name, version, source, and the percentage cached

You configure the address of the reporting server when you use App-V Group Policy settings, as shown in Figure 4-19. Clients forward data to this address, which the reporting server then forwards to the reporting server database. You can install the reporting server separately from the reporting server database, though if you do this, you already must have deployed the database on another server.

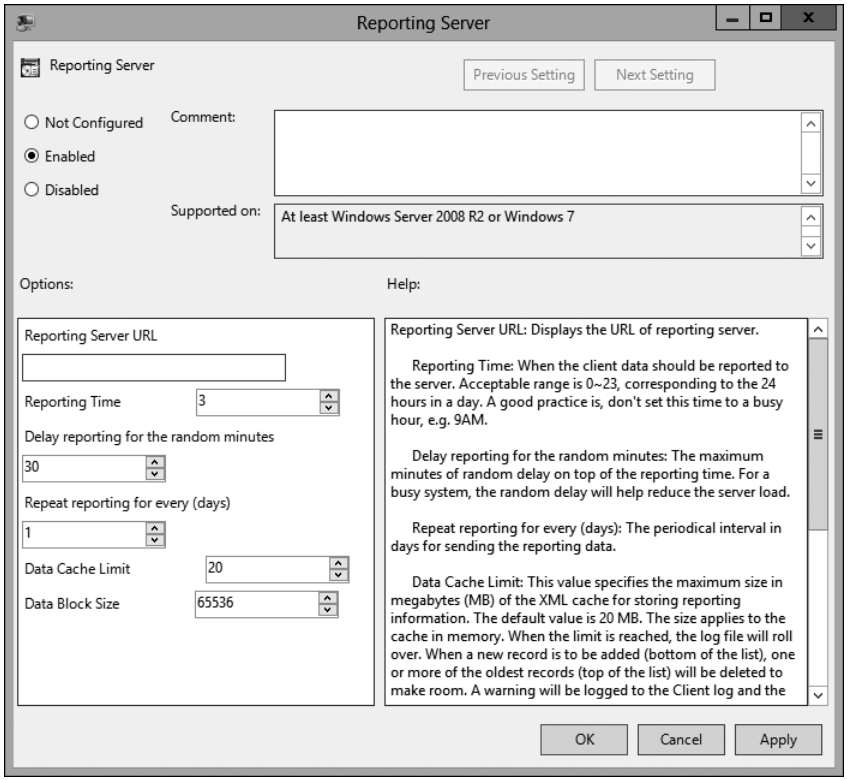

Figure 4-19 GPO settings for reporting

The reporting server database stores all the information that is forwarded to the reporting server. The instance that hosts the reporting server database must meet the same requirements as the instance that hosts the management server database. You can host both databases on the same server. You don't have to install SQL Server Reporting Services to deploy an App-V reporting server, but it is helpful because the App-V product doesn't include report generation.

App-V deployment models

There are three deployment models that you can use to deploy App-V. Each model and its characteristics are described below:

- **Full infrastructure model** This is the most complete deployment of App-V and also is the most commonly deployed model. It offers the most services and functionality and has the largest footprint. The key technologies of the full infrastructure deployment are the App-V Management Server, the App-V Publishing Server, the App-V client, and an

App-V Sequencer. SQL is required on the back end, and the solution ties into AD DS. Optional technologies are the reporting server and associated reporting database. This model offers streaming of applications without requiring a System Center Configuration Manager environment. An example of a typical full infrastructure model deployment is shown in Figure 4-20.

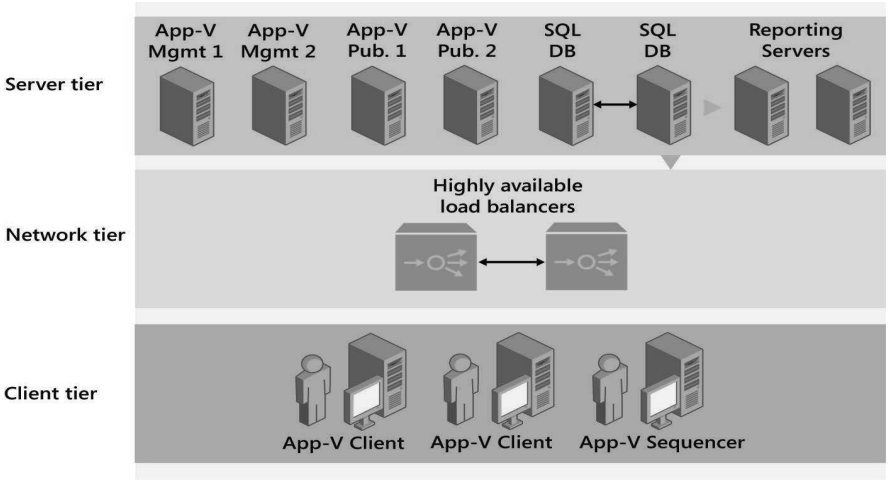

Figure 4-20 Technologies of a typical full infrastructure model

- **App-V integration with Configuration Manager model** This is an implementation of App-V that, at a minimum, includes App-V clients, an App-V Sequencer, and System Center Configuration Manager 2012 or later. The App-V Management Server isn't part of this deployment model, the App-V Publishing Server isn't part of this deployment model, there isn't a desktop configuration service, and you can't track licensing or perform metering. The desktop configuration service is used by App-V clients to find out information about available virtual applications. This deployment model doesn't require SQL or AD DS, but it requires System Center Configuration Manager, which enables you to stream App-V applications from Configuration Manager distribution points and automatically deploy the App-V client to computers.

- **Stand-alone deployment model** This is a small implementation of App-V. This deployment model has only App-V clients and an App-V Sequencer, which enables you to create .appv files or .msi files for delivery via a separate application delivery solution such as Group Policy or network file shares. The .appv files and the .msi files can be run only by the App-V client. This model isn't used often but can be valuable for a test environment or an environment with a lot of users who do not routinely connect to the network.

App-V packages

An App-V package comprises several files that have specific functionality. The primary package file is the .appv file that contains the captured assets and state information. Additional files provide custom integration information for publishing applications, detailed sequencing reporting, and, optionally, sequencing templates and package accelerators. You can use the following files to provide custom integration:

- The .appv file contains the captured files and state from the sequencing process in a single file. This file includes the architecture of the package file, publishing information, and registry settings in a tokenized form that can reapply to a machine and to a specific user on delivery.
- The .msi file is used in stand-alone deployments or, optionally, when deploying by using Configuration Manager or other deployment platforms.
- The _DeploymentConfig.xml file contains default publishing parameters for all applications in the package, and it can be modified to support customization.
- The _UserConfig.xml file allows customization of publishing parameters for specific domain users. You can customize items such as shortcuts and file associations with this config file.
- The Report.xml file contains diagnostic information, how sequencing is done, and what files are excluded from a package. It also contains the messages that are displayed in the Sequencer after you finish sequencing an application.
- The .cab file is an optional package accelerator file that speeds up the creation of sequenced virtual application packages.
- The .appvt file is an optional Sequencer template file that retains commonly reused Sequencer settings.

Contents of an .appv package

An .appv file is a compressed file that contains the contents of a virtual application package. It is based on the Open Packaging Conventions standard. It is used to store a combination of XML and non-XML files in a single entity. You can view .appv file contents by renaming the file with a .zip extension and exploring its contents.

The following list describes the primary .appv file contents:

- **StreamMap.xml** Contains Feature Block 0, also named the Publishing Feature Block.
- **PackageHistory.xml** Contains information about the origin of a package, for example, which user sequenced the package, on which machine, and at what time.

- **FilesystemMetadata.xml** Contains a list of the files that are part of the application and were captured during the sequencing.
- **AppxManifest.xml** Metadata for a package that contains everything that is needed to publish.
- **Registry.dat** A mountable .dat file containing the registry that was captured as part of the package.
- **Root** Contains the file system for the virtualized application that was captured during sequencing.

Planning App-V infrastructure

When introducing a new technology, such as App-V, planning and designing the infrastructure are fundamental to a successful implementation. As we'll discuss in this section, a reliable App-V environment depends heavily on the design and infrastructure. The process for implementing application virtualization is flexible and scalable, with larger deployments requiring more planning and different technologies.

Some key areas of interest when planning your App-V infrastructure include the following:

- The App-V infrastructure requirements
- The various App-V deployment models
- Sizing and performance
- High availability and disaster recovery

App-V infrastructure requirements

Before deploying App-V in your environment, you must ensure that the supporting infrastructure is in place and configured. App-V 5.0 has the following infrastructure requirements:

- **Active Directory Domain Services** AD DS is required for authentication and authorization of applications and connection groups. AD DS is needed only if you plan to deploy an App-V server, such as in a full infrastructure deployment model.
- **Installation service account** A service account in AD DS is required for the initial installation of the App-V server, presuming that your deployment is a full infrastructure model. This account needs Read permission to query AD DS and local Administrators group access on the server on which you perform the App-V installation. Following the installation of the management server, you can transition this to a security group in

AD DS, allowing you to easily add users who require administrative access to the management console.

- **Package repository** This is the location where package files will be stored for delivery to App-V clients.

The servers in an App-V environment have the following requirements::

- **Management server** Supported on Windows Server 2008 R2 with SP1 and newer. It requires the following technologies:
  - Microsoft .NET Framework 4.0 or newer
  - Windows PowerShell 3.0
  - Microsoft Visual C++ 2010 SP1 Redistributable Package (x86/x64)
  - Microsoft SQL Server 2008 Standard, Datacenter, or Developer edition (32-bit or 64-bit) or newer
- **Reporting server** Supported on Windows Server 2008 and newer. It requires the following:
  - Microsoft .NET Framework 4.0 or newer
  - Microsoft Visual C++ 2010 SP1 Redistributable Package (x86/x64)
  - Windows Web Server with the IIS role installed
  - Common HTTP Features (static content and default document)
  - Application Development (ASP.NET, .NET Extensibility, ISAPI Extensions, and ISAPI Filters)
  - Security (Windows Authentication, Request Filtering)
  - Management Tools (IIS Management Console)
  - 64-bit ASP.NET
  - Microsoft SQL Server 2008 Standard, Datacenter, or Developer edition (32-bit or 64-bit) or newer
- **Publishing server** Supported on Windows Server 2008 and newer. It requires the following:
  - Microsoft .NET Framework 4.0 or newer
  - Microsoft Visual C++ 2010 SP1 Redistributable Package (x86/x64)
  - Windows Web Server with the IIS role installed
  - Common HTTP Features (static content and default document)

- Application Development (ASP.NET, .NET Extensibility, ISAPI Extensions, and ISAPI Filters)
- Security (Windows Authentication, Request Filtering)
- Management Tools (IIS Management Console)
- 64-bit ASP.NET

Although the design of an App-V environment is very flexible, certain scenarios are not supported:

- Installation on domain controllers isn't supported for any App-V server technology.
- Installation isn't supported on Server Core installations of Windows Server.
- App-V 5.0 can't be installed on a system that has a previous version of the App-V Management Server.
- Microsoft SQL Server Express as a database engine isn't supported.

App-V deployment possibilities

Distributing virtual applications requires the App-V client software on the target computer. As you design your server infrastructure, you'll need to review the four main deployment models that we introduced earlier in this chapter. Each model has its own strengths, and the model you choose will determine which type of server infrastructure you deploy.

App-V full infrastructure model

The full infrastructure model provides all of the management server capabilities that App-V offers, including application streaming, authentication, security, licensing, and metering.

When planning for the full infrastructure model, you'll need AD DS and Microsoft SQL Server. The App-V Management Server should be on the same LAN segment hosting the database. Publishing servers are used in this model to publish content from a file server share to a distributed environment's remote locations by providing streaming capabilities close to the clients that are using the applications. This reduces latency and improves the end-user experience.

System Center 2012 Configuration Manager–integrated model

If you have an existing System Center 2012 Configuration Manager infrastructure, or you are looking to implement one, you can leverage Configuration Manager to distribute virtual applications in the same way that you distribute traditional application packages. You can add virtual applications to a Configuration Manager environment by using the same Create Application Wizard, as shown in Figure 4-21.

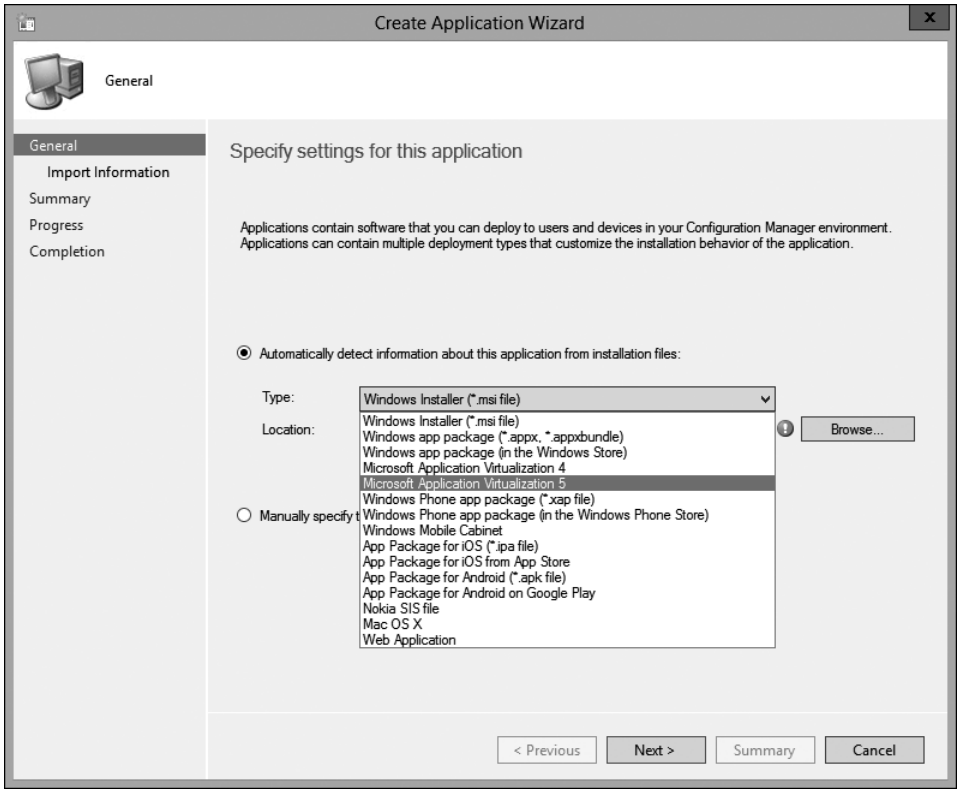

Figure 4-21 Configuration Manager Create Application Wizard

Many of the advanced capabilities that are available for managing a traditional application—such as using task sequences and building queries in collections to define which devices are targeted—also are available for a virtual application. You can target both users and computers to deliver an application in a more intelligent way, expanding on capabilities of the App-V full infrastructure model. For example, when you use a primary device as one of the possible rule requirements, you can identify which deployment type is used based on whether the user is working on his or her primary device.

The Configuration Manager–integrated model requires both the App-V client and the Configuration Manager client on each managed system. It doesn't use any server technologies of the full App-V infrastructure to deliver virtual applications; instead, it uses existing Configuration Manager distribution points to deliver the virtual application to client devices. Note that some reporting capabilities aren't available in the integrated model when compared to the full infrastructure model. For example, if you use local delivery where clients download and execute the application, you only can report if the application has been used and the

last application usage time. In the full infrastructure model with reporting, you can report the number of times an application has been used.

Application delivery to a Configuration Manager client works differently from the App-V full infrastructure scenario. In the full infrastructure model, the App-V client manages its own content, and it can refresh instantly against the publishing server. In the Configuration Manager–integrated scenario, the Configuration Manager client manages the App-V client.

Configuration Manager supports two types of delivery methods for virtual applications:

- **Streaming delivery** You can enable streaming delivery on Configuration Manager distribution points. This option streams a virtual application to a client through HTTP or HTTPS.
- **Local delivery** This delivery first uses the Configuration Manager client to download all the files needed for the application through Background Intelligent Transfer Service (BITS). After downloading the files, the package fully loads into the App-V client cache.

Electronic software distribution model

The electronic software distribution (ESD) model is ideal for environments in which you prefer to leverage an existing software distribution solution. In this case, most distribution systems can use the virtual .msi file produced by the App-V Sequencer for delivery with an .appv package.

Planning considerations for the ESD model include the following:

- **Existing software distribution system** An existing software distribution system that can recognize and distribute .msi packages to client devices.
- **App-V Sequencer** A system deployed in your environment with the App-V Sequencer installed for building and managing virtual applications.
- **Windows PowerShell** The ability to deploy a script that contains the App-V client module for Windows PowerShell cmdlets. This provides the ability to add and publish packages in ESD mode.
- **Connection groups** Designating connection groups (grouping one or more App-V packages to enable interaction with one another) requires manually creating a connection group XML file and deploying it by using a custom Windows PowerShell script.
- **Group Policy** Having Group Policy available simplifies the task of configuring the App-V client. Alternatively, a manual or scripted configuration is possible through the Windows Registry. In Figure 4-22, a GPO named App-V settings provides several App-V settings to computers.

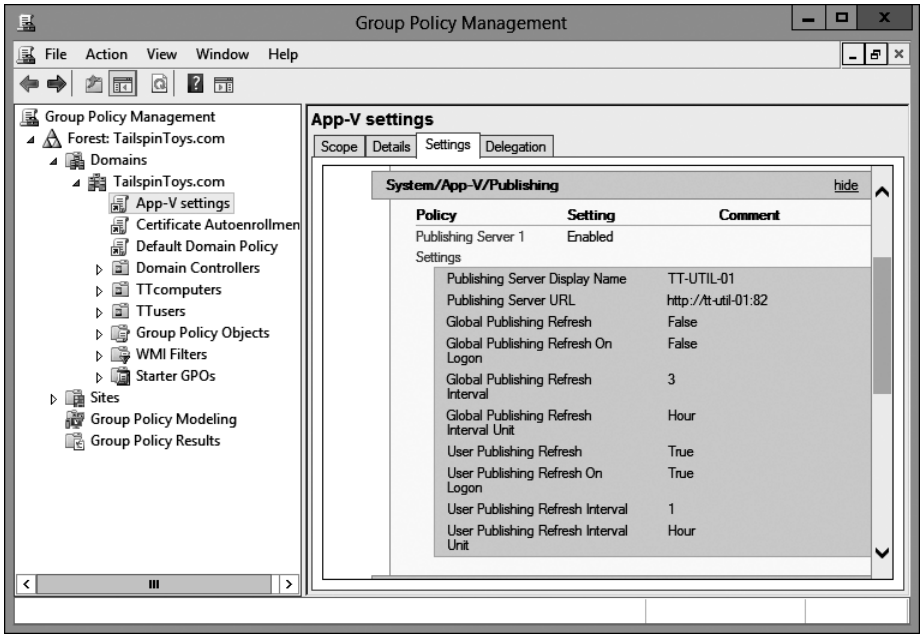

Figure 4-22 App-V GPO settings

Stand-alone deployment model

The App-V stand-alone model consists of the App-V Sequencer and an App-V client, and it requires no additional App-V infrastructure. The Sequencer has an option to create a virtual .msi file during the sequencing process. The virtual .msi file invokes Windows PowerShell commands and then publishes and loads the application to the App-V client cache.

The App-V Sequencer packages publication information, shortcuts, and the installation routines into an .appv file package, and the Sequencer generates virtual .msi files that you can execute manually. When executed, the installer adds the virtual application package to the App-V client and configures publication information to load applications from a local location rather than stream them across a WAN.

Stand-alone deployments require an App-V client on the computers, which allows a virtual .msi file to publish and load virtual applications or enables management through Windows PowerShell. You don't configure an App-V client to connect to any App-V server.

The stand-alone delivery scenario enables an organization to deploy virtual applications in situations where no servers are available to support other deployment methods for virtual applications. Use stand-alone deployments in the following scenarios:

- There are remote users who can't connect to an App-V infrastructure.
- Software management systems, such as Configuration Manager or another electronic software distribution system, already are in place.
- Network bandwidth limitations prevent electronic software distribution. In this case, you can use virtual application delivery on physical media.

Because the stand-alone model employs an .msi file, you can distribute the file if you use an existing software distribution infrastructure, such as GPOs, shared folders, optical media such as CDs and DVDs, USB flash drives, or others.

Service disruption impact

One of the common design steps in implementing App-V is to make the infrastructure highly scalable, which limits the impact of service disruption. The App-V infrastructure is highly dependent on AD DS. Therefore, it is recommended that you carefully plan your AD DS architecture to avoid unwanted service disruptions.

It's important to point out that from a client perspective, once an application is loaded on a computer, that device can run the application independently from the server. A previously published package can have different states on client computers:

- **Not Available** In this state, the package isn't registered or isn't available on the client.
- **Registered** In this state, the package is registered to the computer, but it still is not registered for the user.
- **Published** In this state, the application is registered and published on the client, and the user can start using it.
- **Partially Loaded** In this state, the application can be started because the client already has downloaded the initial feature block. Depending on which portion is missing, the rest of the files can download over the network, so the file server repository is the critical technology that provides that functionality.
- **Fully Loaded** In this state, the application downloads and extracts entirely onto a client machine, and it can be used in an offline scenario.

The following areas will be a concern if the virtual applications aren't configured to fully load on the client machine or aren't already published on the publishing server:

- **File server repository** The most critical technology that influences an application's functionality will be the file server repository. Storage availability and AD DS will need extra planning considerations in this scenario.
- **Management server** If the management server or the management database is down, adding new packages, updating existing packages, or managing connection groups won't be possible.
- **Publishing server** Publishing server failure affects the ability to make changes to the publishing list that clients previously received, which is for non-persistent Virtual Desktop Infrastructure (VDI) and RDS scenarios.
- **Reporting server** Reporting server or reporting database failure isn't critical to running App-V applications; the only functionality that might not work is reports that a client sends about usage statistics for virtual applications, which are stored in the reporting database.

Functional and physical placement

Organizations might plan different App-V infrastructure deployments based on their needs. When you start to plan for your App-V environment, you should try to answer the following questions to help with your design and implementation:

- Are there requirements that all roles must live on a single server? Decide whether you want to combine or cohost functionality.
- Do you need centralized or decentralized roles in a distributed environment?
- What are the requirements for high availability?
- Are the virtualized application users located in all of your office locations?
- Is your virtualization environment able to virtualize your entire App-V deployment?

Based on the answers to these questions, there are several design scenarios:

- **Small and midsize deployments** For small and midsize scenarios, which commonly address an environment with a small number of users and few packages in a single geographical site, you might cohost all of the roles on a single server.
- **Midsize and large deployments** For midsize and large deployments, which commonly require a flexible and scalable environment, you might consider a more complex design in which all services implement individually. In this scenario, every connection addresses a virtual IP address and machine name, and no services cohost on any given computer.

- **Distributed deployments** For distributed deployments, which commonly need to support a large number of users in different locations with many different requirements, you should implement a scenario that can address locations with no datacenter and weak Internet connectivity. For this type of design, all configuration data that is stored in a management database should be located in a major datacenter. Because the management server communicates with the management database, it should be located close to the SQL server because SQL communication is time-sensitive and network-sensitive. The file server repository share that holds the application should be located close to clients, and interval refreshes can be adjusted according to the actual network capabilities.

- **High availability deployments** In this scenario, you must have two identical machines (physical or virtual) that are configured in NLB mode (or behind a third-party hardware load balancer), where the following services are installed:
  - App-V management
  - App-V publishing
  - App-V reporting

Even if you don't start with a highly available environment, you should consider using load balancing. It can simplify scaling out later and provide some additional capabilities such as drainstopping a server.

It is recommend that you host the SQL Server database separately from the App-V services. This consideration is made for performance, security, and scalability. For highly available designs, you should consider implementing a SQL Server cluster.

Sizing and performance

Actual sizing and performance planning depends on multiple factors, such as scaling an App-V infrastructure properly to lower the round-trip response time and providing proper package optimization for streaming across slow networks.

Round-trip response time on a publishing server is the time that is needed for the publishing server to receive a successful package metadata update from the management server. Round-trip response time on a client is the time the App-V client computer takes to receive a successful notification from the publishing server.

If you have increased internal demand, you can implement an additional management server behind your load balancers.

Often, users might demand external scalability based on the location you must support. A design should include a content repository in each location to provide conveniently located packages to clients. Additionally, you might consider implementing a publishing server and a management server to lower the round-trip response time on clients. Capacity planning

should be included to evaluate future demands in planned growth to meet expected performance levels.

A few factors influence round-trip response time on a publishing server. Some of these include the number of:

- Publishing servers that make simultaneous requests.
- Connection groups that are configured on a management server.
- Access groups that are configured on a management server.
- A single management server can simultaneously respond to up to 320 publishing servers with a round-trip response time of approximately 40 seconds; a single management server with fewer than 50 publishing servers results in a round-trip response time of less than 5 seconds.
- The number of connection groups starts to influence round-trip response time after more than 400 are created.
- The number of access groups increases the round-trip response time as it grows.

The number of publishing servers that simultaneously connect to a management server does not influence central processing unit (CPU) utilization and SQL database transactions per second; batches per second are identical, regardless of the number of publishing servers.

For App-V, reporting server capacity planning should focus on the number of clients that simultaneously send reporting information to a reporting server. Round-trip response time increases linearly with an increased number of clients. For example, round-trip response time is 2.6 seconds with 500 clients and 5.2 seconds with 1,000 clients.

Capacity planning for the publishing server influences the round-trip response time on an App-V 5.0 client computer to send a publishing refresh request and to receive a response.

The following are the main factors that influence capacity planning of an App-V publishing server:

- The number of clients that simultaneously connect to a single publishing server.
- The number of packages in each refreshThe network bandwidth between clients and the publishing server.
- A publishing server with a dual-core processor can respond to up to 5,000 clients that simultaneously request refreshes. From 5,000 through 10,000 clients, a publishing server should have a quad-core processor at minimum. Increasing the number of packages increases response time by 40 percent, and network bandwidth has a major influence on

response time. For example, clients that run on slow networks—less than 1.5 megabits per second—will have a significantly slower response time than the same number of clients that run on LAN networks.

High availability for App-V

You should plan for a highly available App-V infrastructure in organizations where App-V is important. The high availability strategy for the App-V infrastructure depends on the App-V deployment model, because different procedures and settings for high availability are needed for different App-V deployment models.

Stand-alone deployment model

The stand-alone deployment model only requires an App-V Sequencer and client computers that have an installed App-V client. In the stand-alone deployment model, the App-V Sequencer is used only when a new application needs to be sequenced. Because the App-V Sequencer installs only when a new application needs to be sequenced, it isn't necessary to make the Sequencer highly available. If you stream from a central share, this share can deploy on a clustered file system or on an NLB web farm. In cases where you require access to sequencing, even in a disaster recovery (DR) scenario, you can deploy multiple sequencing computers.

App-V full infrastructure model

From a planning perspective, the App-V full infrastructure model requires the most attention. Because there are a multitude of technologies with differing high availability models, you should spend time looking at the options available and decide which one makes the most sense for your environment. The following are some questions that you should answer to help you plan for high availability:

- Does the reporting service need to be highly available?
- Does the sequencing computer need to be highly available?
- Are there infrastructure technologies outside App-V that may impact the high availability of App-V, such as load balancers, switches, virtualization servers, or storage?
- Which secondary site should each office use in the case of a publishing server failure at the office?

The answers to the above questions will help you plan the services, the number of servers required, and the high-level design of your highly available environment.

The App-V full infrastructure model stores all configuration and application information in the management server database and stores all utilization data in the reporting server database,

so each of these databases is a single point of failure in this model. Therefore, when you are configuring the App-V full infrastructure model to be highly available, you need to ensure that the management server database and the reporting server database remain accessible. You can do this by deploying these two databases on an instance that is installed on a highly available VM or on a clustered SQL Server instance.

App-V 5.0 supports multiple management, reporting, and publishing servers. You can configure the App-V 5.0 management and reporting server databases to work with multiple management servers and reporting servers by using a security group when specifying the computer account location during setup. At any time, you can add publishing servers to an existing App-V full infrastructure model deployment.

Consider Figure 4-23, which shows what can happen when the application team provides high availability for its services but another team isn't engaged in the project and is unaware of potential impacts.

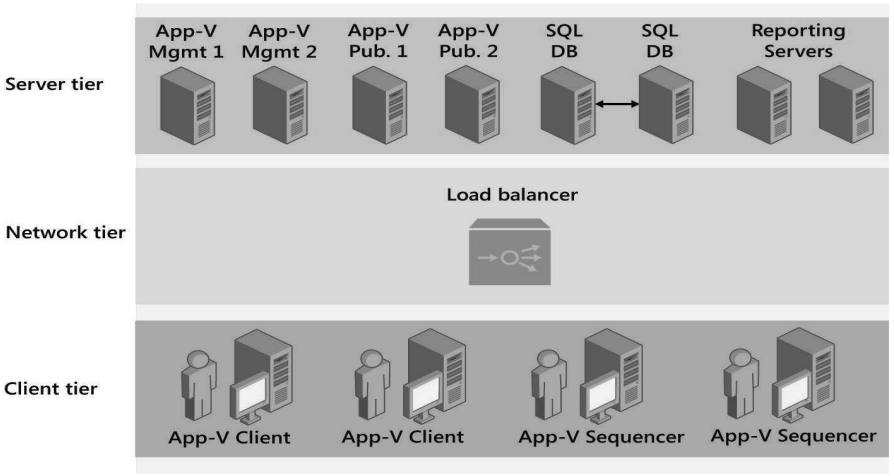

Figure 4-23 A diagram of an App-V full infrastructure deployment model

In the diagram above, although all of the App-V technologies are highly available, there is only one load balancer. It represents a single point of failure. Instead of this scenario, the availability of all services on which App-V is dependent should match the App-V availability.

Integrated Configuration Manager model

For the integrated Configuration Manager model to be made highly available, you should look at all of the options to meet your high availability requirements and figure out if any infrastructure changes are required:

- Are highly available VMs available in your environment? If so, you can use one or more for your Configuration Manager servers.
- Is your existing SQL environment highly available?
- Is the storage that the virtual environment uses highly available?
- Do you have distribution points in all of the locations to which you plan to deliver virtual applications? If so, you need to plan for scenarios in which a distribution point becomes unavailable. In environments with an existing Configuration Manager deployment, it isn't unusual to have sites with a single distribution point. You should consider multiple distribution points if your requirements include immediate access to virtual applications, especially for new App-V clients.
- You also may want to look at the overall high availability of the Configuration Manager environment. Is there an existing hierarchy with a central administration site?

Similar to other App-V deployment models, you should look at all aspects of your environment and ensure that all of the services involved are configured for high availability.

➤ **For more information about high availability in Configuration Manager, see the "Planning for High Availability with Configuration Manager" website at *http://technet.microsoft.com/en-us/library/hh846246.aspx*.**

Disaster recovery

A disaster recovery plan should include a proper backup of critical technologies to respond to a service outage. In addition, regular testing of restore operations will help ensure that the backups are functional and that the operational procedures are adequate to recover from a disaster. At a minimum, an App-V infrastructure backup should protect the management database and the package repository that contains the App-V packages. Outside App-V, you should ensure that the services on which App-V depends also are backed up and restored first after a disaster. For example, you need to ensure that AD DS, SQL, Server virtualization, and the core networking services are up and running before you can successfully recover App-V.

In an App-V backup and recovery scenario, each role has different requirements:

- A management server does not hold any unique data other than the registry configuration of the database source, so you can easily re-create this role in case of a disaster.
- A publishing server contains a registry value that indicates the host name of the management server to contact and a cached copy of the latest publishing data, which you need to back up.
- The reporting server has a registry value that indicates the name of the reporting database.

There are two different scenarios for recovery procedures:

- If a server that contains all of the roles fails, administrators should perform standard image recovery procedures that are defined by organizational policy. This often means restoring the VM or physical server to the most recent backup.
- If an App-V service fails, administrators should perform a recovery by installing the App-V technologies and prerequisites, such as installing and configuring Internet Information Services (IIS) and installing SQL Server on the database server.

When you restore the management server, it enables this service to become operational as soon as the service can contact the App-V database. When you complete the restoration of publishing servers, client requests will be serviced as soon as the service contacts the publishing servers (from 1 to 10 minutes). After the App-V services are restored, the reporting server starts accepting client connections.

Deploying App-V infrastructure

After you determine which application virtualization model to use, you need to deploy the appropriate roles to support that model. In most production environments, you should host the management server database and the reporting server database separately from the management server and the reporting server. It isn't uncommon to see the management database and reporting database also separated, depending on the database infrastructure you have. Before a deployment, you need to understand the order in which to deploy App V technologies. In this section, we will explain the following:

- App-V infrastructure requirements
- Installing management databases
- App-V Management Server configuration
- App-V Publishing Server deployment and configuration
- App-V for Remote Desktop Services clients
- Integrating App-V with System Center Configuration Manager

App-V infrastructure requirements

The technologies of App-V should be deployed in a specific order. If you deploy all roles on the same server when you deploy the App-V full infrastructure model, the installation wizard automatically deploys them in the correct order. If you deploy the roles on more than one server, you should deploy them in the following order:

1. Management server database
2. Management server
3. Publishing server
4. Reporting server database (optional)
5. Reporting server (optional)

When you configure management server database settings, you need to specify the security account of the computer that will access the database. This can be a security group account or the computer account of the server that will function as the management server. When you install a management server separately, you need to specify the instance location and credentials that will be used to access the management server database.

When you deploy a publishing server, you need to specify the network address of the management server. You can't deploy a publishing server without having a management server already deployed, unless you deploy all of the roles at the same time.

You must deploy a reporting server database before deploying a reporting server. The reporting server doesn't have dependencies on any other services in the App-V full infrastructure model.

You don't need to install the App-V Sequencer when deploying the App-V full infrastructure model. However, it is a good idea to deploy a sequencer as soon as possible to begin testing the deployment. The App-V client usually is the last application to be deployed.

Installing management databases

The management server stores all of the configuration data in an App-V management database, which includes all application metadata, the deployment configuration, the relationships, and the security assignments. The management server only communicates with the management database, and it is the first technology that should be installed in the App-V full infrastructure model. When you add management servers for a scalable deployment, you only need to allow Read and Write permissions to the database. You don't have to provide additional configurations.

The minimum supported database platform is Microsoft SQL Server 2008 R2 Standard, Enterprise, Datacenter, or Developer edition (32-bit or 64-bit). The Developer edition should not be used in a production environment. Additional prerequisites include the installation of the following:

- .NET Framework 4 (Full Package)
- Microsoft Visual C++ 2010 SP1 redistributable package (x86)
- Windows Server 2008 or newer

In the deployment scenario in which you install all of the technologies on the same computer, the App-V server setup GUI-based installation first installs the App-V management database and then installs the management server and the publishing server. Finally, if selected, the reporting database and reporting server are installed.

If you are implementing scalable deployment, you should run the GUI installation on a server that hosts the management database because remote SQL database creation isn't supported in the installer.

As an alternative, you can install the SQL database when you execute SQL Server scripts that are extracted from the server setup, as shown in Figure 4-24. SQL Server scripts extract from the setup with the following command: `appv_server_setup.exe /layout /layoutdir=c:\ extract`.

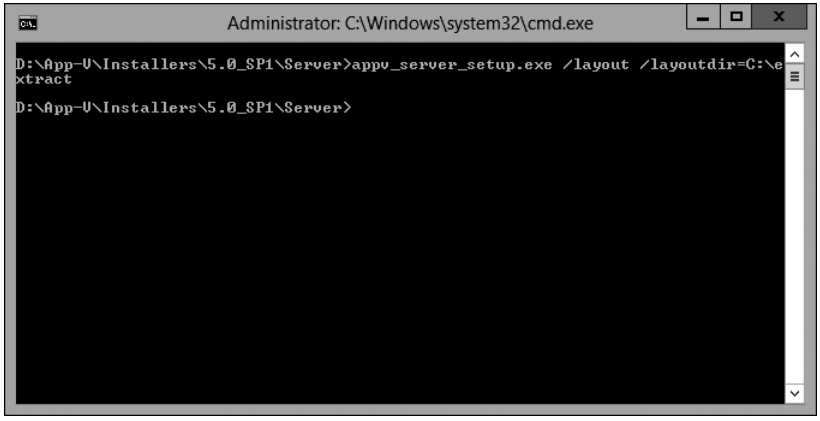

Figure 4-24 Example appv_server_setup.exe command

In the destination folder, you must modify two of the six scripts to provide the appropriate Read and Write permissions for the domain accounts or domain groups that you need to manage an App-V infrastructure. The first modification must be done to the Permissions.sql

script to replace the entry for [ManagementDBWriteAccessAcountSid] and for [Management-DBWriteAccessAcountName] with the security identifier (SID) and the name for the domain group that requires Write permissions to the database. This group should include the App-V administrator account and all management servers in the environment. If you use the same account for installation and App-V administration, then you should use the same entries for [ManagementDBPublishAccessAcountSid] and [ManagementDBPublishAccessAcountName]. Otherwise, you should enter the correct SID and name for the installation account. In Figure 4-25, the Permissions.sql file has been modified and is ready for use.

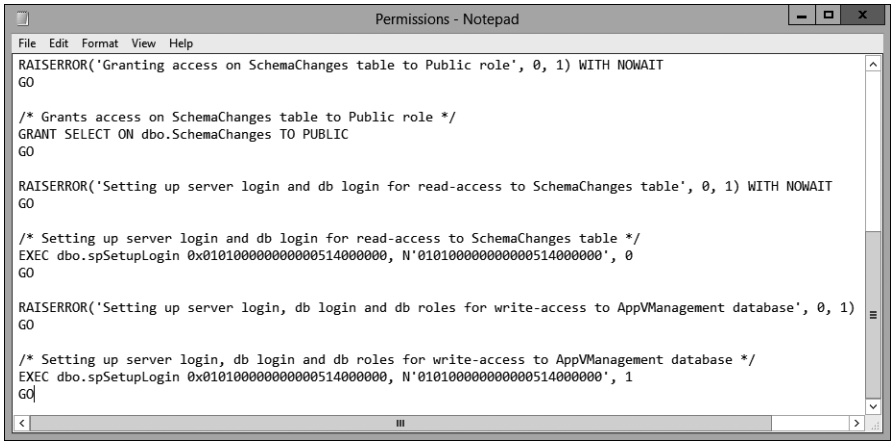

Figure 4-25 An example Permissions.sql file

Modification of the second script, Database.sql, is optional and has to be done only if you plan to replace the default database name AppVManagement with a unique name. In Figure 4-26, the Database.sql file has been updated to create a database named CustomDB.

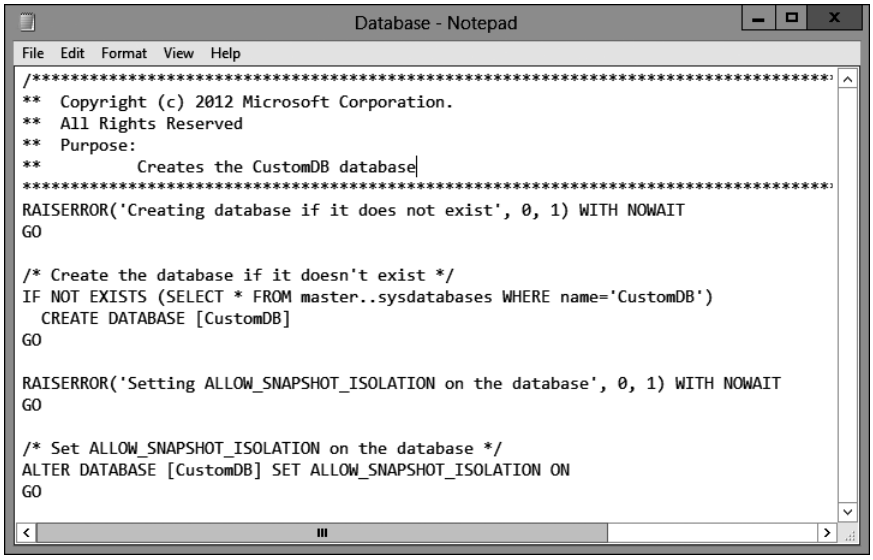

Figure 4-26 An example Database.sql file

A SQL Server administrator must run prepared SQL scripts against a computer that is running SQL Server that will host the database. SQL Sysadmin permissions are required. You can run the script if you first open the SQL Server Management Studio console and run it as a query, but you need select the proper database. The second method that you can use is the **OSQL** command-line application. "The switches /E, /i, and /d are case-sensitive. Reporting database setup is identical to a management database and can be done with the App-V setup installer, or it can be pre-created with SQL scripts. The following commands can be used:

```
OSQL –E –i database.SQL

OSQL –E –d MS_Appv5_Management –i CreateTables.sql

OSQL –E –d MS_Appv5_Management –i CreateStoredProcs.sql

OSQL –E –d MS_Appv5_Management –i UpdateTables.sql

OSQL –E –d MS_Appv5_Management –i insertversionlnfo.sql

OSQL –E –d MS_Appv5_Management –i Permissions.sql
```

App-V Management Server configuration

An App-V Management Server provides a centralized location to manage an App-V 5.0 infrastructure for delivering virtual applications to both an App-V client and an RDS (formerly Terminal Services) client. Unlike previous versions of App-V, a web application that runs on Silverlight manages the App-V 5.0 infrastructure. You configure this web application's address

when you install a management server. The installation of an App-V Management Server creates a dedicated IIS website, for which you can specify the name during the installation setup. By default, it is called the Microsoft App-V Management Service. The App-V Management Service will be configured to listen on a dedicated port number, which can be provided during setup.

App-V server features can install on multiple servers to provide scalability and high availability; however, all App-V server features would need a common way to be accessed, such as by using a load balancer. Each management server node needs connectivity to the database on the computer that is running SQL Server. If a single server hosts multiple technologies, they can use different ports, or you can configure them to share a single port.

Preinstallation tasks include configuring appropriate user and administrative groups that can install and administer a management server. A management server requires that an IIS server is installed and configured to be trusted for delegation. If you plan to support Secure Sockets Layer (SSL) for connectivity to a management server, you also need a server certificate that is issued from either an internal or a public certification authority.

Installing an App-V Management Server at the command line requires elevated privileges. You can display the installation parameters, shown in Figure 4-27, by running the following command:

```
appv_server_setup.exe /?
```


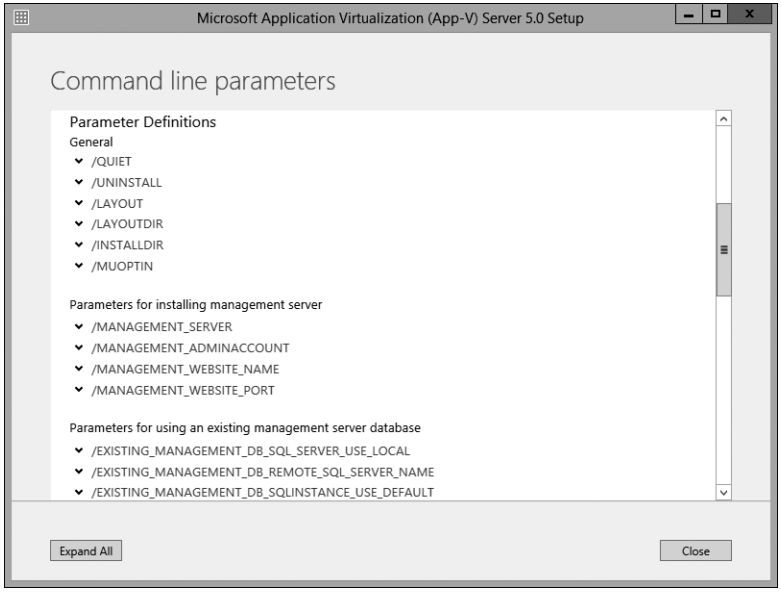

Figure 4-27 App-V setup command-line parameters

The following commands provide an example of an App-V Management Server installation, as shown in Figure 4-28. You can verify the output of the installation in the log file appv_server _datatime.log in the %temp% directory.

```
appv_server_setup.exe /quiet /management_server /MANAGEMENT_ADMINACCOUNT="Adatum
\AppVAdmin" /MANAGEMENT_WEBSlTE_NAME="Microsoft App-V Management
service" /MANAGEMENT_WEBSITE_PORT="80" /EXISTING_MANAGEMENT_DB_REMOTE_SQL_SERVER
_NAME="SQLSRV.adatum.com " /EXISTING_MANAGEMENT_DB_SQLIN STANCE_USE_DEFAULT
/EXISTING_MANAGEMENT_DB_NAME="AppVManagement"
```

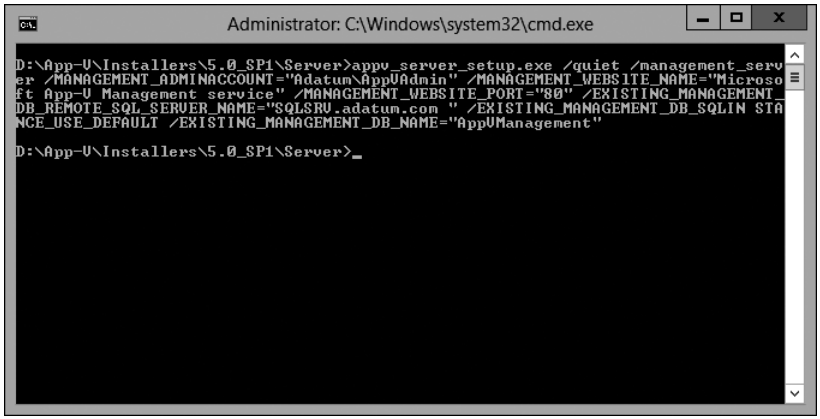

Figure 4-28 Command-line setup of an App-V server

Common postinstallation tasks include sharing the content folder that is used to store the App-V package. It also is common to enable firewall rule exceptions.

You can install a management server by using a very basic GUI, but for enterprise deployment, we recommend script-based installation. Management servers and publishing servers have a dependency on IIS with the following features:

- Common HTTP features: static content and default document
- Application development features: Microsoft ASP.NET, Microsoft .NET Extensibility, and Internet Server API (ISAPI) extensions and filters
- Security features: Windows authentication and request filtering
- Management tools features: IIS Manager

One common method for proper installation of IIS and all required services is to use the Deployment Image Servicing and Management (DISM) tool, which you can use to create a script by saving the following commands in a text editor with the .cmd `extension:dism /Online /Enable-Feature /FeatureName:llS-ApplicationDevelopment ^`

```
/FeatureName:IIS-ASPNET /FeatureName:HS-commonHttpFeatures ^
/FeatureName:Iis-DefaultDocument /FeatureName:Iis-DirectoryBrowsing ^
/FeatureName:Iis-HealthAndDiagnosti cs ^
/FeatureName:HS-Httpcompressionstatic ^
/FeatureName:HS-HttpErrors /FeatureName:HS-HttpLogging ^
/FeatureName:HS-HttpTracing /FeatureName:HS-lSAPiExtensions ^
/FeatureName:HS-ISAPiFilter ^
/FeatureName:Iis-LoggingLibraries /FeatureName:HS-ManagementConsole ^
/FeatureName:Iis-Managementservice /FeatureName:HS-NetFxExtensibi1ity ^
/FeatureName:IIS-Performance /FeatureName:ns-RequestFiltering ^
/FeatureName:HS-RequestMonitor /FeatureName:HS-Security ^
/FeatureName:Iis-staticcontent /FeatureName:Iis-webserver ^
/FeatureName:HS-webserverManagementTools ^
/FeatureName:HS-webserverRole /FeatureName:Iis-windowsAuthentication ^
/FeatureName:WAS-ConfigurationAPl /FeatureName:WAS-NetFxEnvi ronment ^
/FeatureName:WAS-ProcessModel /FeatureName:WAS-WindowsActivationservice ^
```

A management server has the following requirements:

- 1-gigahertz (GHz) or faster x64 processor; two cores Intel Xeon 2.0 GHz or faster recommended
- 2 gigabytes (GB) or more of RAM; 4 GB of RAM recommended
- 200 megabytes (MB) of free disk space (does not include content); 40 GB recommended
- Windows Server 2008 R2 SP1 or newer.NET Framework 4 Extended
- .NET Framework 3.5.1 Features (or 4.5)
- Visual C++ 2010 SP1 Redistributable Package (64-bit)
- Visual C++ 2010 SP1 Redistributable Package (32-bit)
- Silverlight
- Windows PowerShell 3.0

App-V publishing server deployment and configuration

When you deploy an App-V publishing server, you must specify the location of an existing App-V Management Server. This is different from previous versions of App-V, in which it was possible to deploy a stand-alone streaming server without having to configure a management server.

Publishing servers function as distribution points for virtualized applications when you use the App-V full infrastructure model. Applications stream from these servers to clients. The entire application doesn't need to stream before a user can start interacting with it; therefore, you won't need as much bandwidth as you would with other deployment methods. Nonetheless, you still need to provision adequate bandwidth for the connection between a publishing server and the client.

To install a publishing server by using a GUI installer, you must follow the same steps as installing the management server. You have to point to an existing management server, and if these two roles coexist on the same computer, you must choose a different port for the website.

When you perform a command-line installation, you can use the help that the installer provides, which presents examples and definitions that construct the following command:

```
appv_server_setup.exe /?
```


You can use the following commands to perform publishing server installation at the command line:

```
appv_server_setup.exe /quiet /publishing_server

/PUBLlSHlNG_MGT_SERVER=http://lon-svr1.adatum.com

/PUBLlSHlNG_WEBSlTE_NAME="Microsoft Appv Publishing service"/PUBLISHING_WEBSITE_PORT="80
"/EXISTING_MANAGEMENT_DB_REMOTE_SQL_SERVER_NAME="SQLSRV.adatum.com" /EXISTING_MANAGE-
MENT_DB_SQLIN STANCE_USE_DEFAULT /EXISTING_MANAGEMENT_DB_NAME="AppVManagement"
```

As a best practice, when you install on the same server as the management server, use port 80 for the publishing server and an alternate port for the management server.

After you deploy both the management server and the publishing server, you need to configure them with appropriate firewall rules to provide management and client connectivity.

A publishing server is a web application that is hosted on IIS, and any configuration changes can be done through IIS Manager or by using the HKLM\Software\Microsoft\APPV\Server\PublishingService\PUBLISHING_MGT_SERVER registry settings to point to the protocol and port that establish connectivity with the App-V Management Server. The PUBLISHING_MGT_SERVER_REFRESH_INTERVAL registry setting specifies how often a publishing server queries a management server for packages. The default value is 600 seconds (10 minutes), and for testing purposes, you can shorten the interval to propagate changes to

clients more quickly. Any registry changes will become effective when the application pool restarts or IIS restarts.

You also can configure management server settings in IIS and the registry, as shown in Figure 4-29.

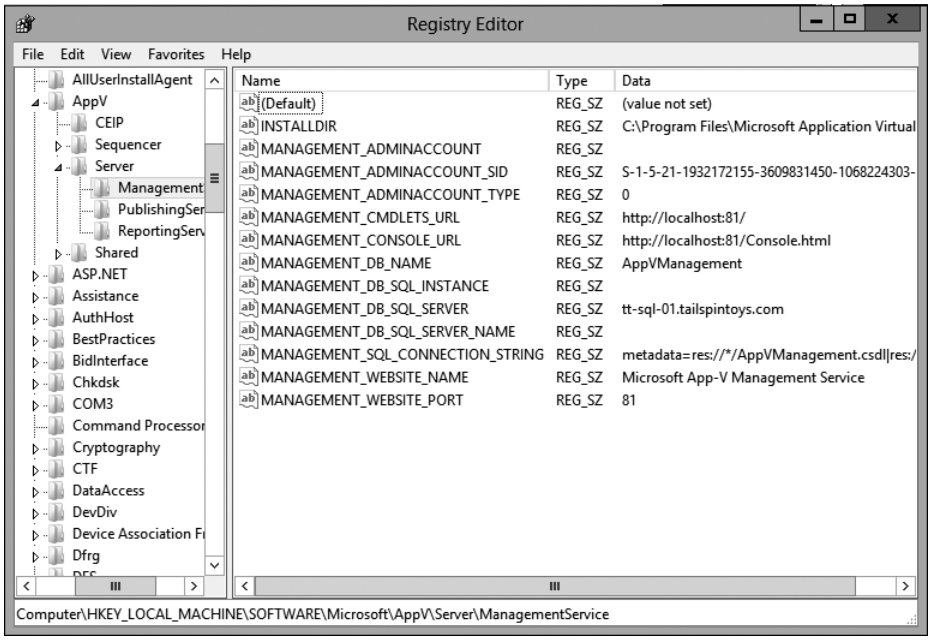

Figure 4-29 App-V registry settings for the management server

`HKLM\Software\Microsoft\APPV\Server\ManagementService` contains configuration data for a management server. From this registry location, you can identify or change a connection string to the management database *(MANAGEMENT_SQL_CONNECTION_STRING) or identify the port and name for the management website.*

In addition to the registry, some configuration settings are stored in the files in the *INSTALLDIR*. The AdminGroup.xml file contains information to recover access to an App-V console when you remove the last administrator from the console.

A publishing server has the following requirements:

- 1 GHz or faster x64 processor; two cores Intel Xeon 2.0 GHz or faster recommended
- 2 GB or more of RAM; 4 GB of RAM recommended
- 200 MB of free disk space (does not include content); 40 GB recommended

- Windows Server 2008 R2 SP1 or newer.NET Framework 4 Extended
- Visual C++ 2010 SP1 Redistributable Package (32-bit)
- Windows PowerShell 3.0
- The Web Server role with the following features:
  - Common HTTP features: static content and default document
  - Application development features: ASP.NET, .NET Extensibility, ISAPI extensions and filters
  - Security features: Windows authentication and request filtering
  - Management tools features: IIS Manager

App-V for Remote Desktop Services client

App-V 5.0 has a separate, special client that makes it possible to run virtualized applications on RD Session Host servers. With this client, you can run applications on RD Session Host servers that might not otherwise run on an RD Session Host server.

The App-V for RDS client has the following system requirements:

- 1.4 GHz or faster x86 or x64 processor
- Windows Server 2008 R2 SP1 or Windows Server 2012
- .NET Framework 3.51 and 4 (Full)
- Windows PowerShell 3.0
- Microsoft KB2533623 (Windows Server 2008 R2)
- Visual C++ 2008 redistributable (if installing by using an executable file)

You must configure Windows Server 2008 R2 or Windows Server 2012 as an RD Session Host server before you install the App-V for RDS client.

You can use the App-V for RDS client with the App-V full infrastructure, stand-alone, and Configuration Manager–integrated models. The App-V for RDS client uses the same Group Policy settings as the normal App-V client.

Integrating App-V with System Center Configuration Manager

The Configuration Manager–integrated model requires that you have an existing Configuration Manager or newer deployment. This model allows you to deploy sequenced App-V applications as one of many different application deployment types.

Before deploying sequenced App-V applications, you should configure App-V client software as an application that you can deploy. You then can specify the App-V client as a requirement when deploying any sequenced App-V application.

You can create the App-V client as an application by performing the following procedure:

1. Copy the App-V client installation file, corecli_amd64.msi or corecli_i386.msi, to a shared folder. In the Configuration Manager console, in the Software Library workspace, under the Application Management node, click Applications.
2. On the ribbon, click Create Application.
3. On the General page of the Create Application Wizard, set the type to Windows Installer (*.msi file) and then click Browse.
4. Browse to the shared folder where you copied the App-V client installation file. Finish the wizard and then click Close.

To create an App-V application in Configuration Manager, perform the following procedure:

1. In the Configuration Manager console, in the Software Library workspace, under the Application Management node, click Applications.
2. On the ribbon, click Create Application.
3. On the General page of the Create Application Wizard, set the type to Microsoft Application Virtualization 5 and then click Browse to go to the network location that hosts the file in .appv file format.
4. Finish the wizard and then click Close.

This page intentionally left blank

This page intentionally left blank

CHAPTER 8

Planning and deploying session-based virtual desktops

Session-based virtual desktops are widely used by organizations to provide remote access to data and applications in a centralized and controlled environment. In Windows Server 2012 R2, Remote Desktop Services (RDS) provides the infrastructure to implement session-based virtual desktops and virtual machine (VM)–based virtual desktops.

In older versions of Windows Server, session-based desktops were provided by a feature named Terminal Services. Terminal Services had the same basic functionality for session-based desktops as RDS, but RDS has been extended with additional functionality to improve the user experience and manageability.

Understanding RDS

RDS is a Windows Server role that provides much more than just remote desktops. RDS includes six role services that enable you to create a scalable and fault-tolerant RDS deployment. You can manage an RDS deployment centrally and in the same way, regardless of the number of servers in an RDS deployment. This makes RDS very scalable.

One of the most common uses for RDS is the deployment of session-based virtual desktops. In a session-based virtual desktop, all processing is performed on a Remote Desktop Session Host (RD Session Host) server, and the results are displayed on a Remote Desktop client. The communication between the client and the RD Session Host server uses Remote Desktop Protocol (RDP).

RDP is a very efficient protocol and sends a limited amount of data over the network. This makes it possible to use RDS to provide desktops and applications for users over a LAN, from branch locations over a WAN, or over the Internet.

RDS includes the following functionality:

- **Provides users with a full desktop** Whether you use session-based virtual desktops or VM-based virtual desktops, you can provide users with access to a full remote

desktop that you can access from almost anywhere if you configure the necessary infrastructure.

- **Provides users with access to applications** You can use RemoteApp to provide users with access to applications running on an RD Session Host server. These applications run in a window just as regular applications do on users' desktops. From the user's perspective, applications delivered by RemoteApp function as if they are installed locally.
- **Allows secure remote access without using a virtual private network** The Remote Desktop Gateway (RD Gateway) role service is used as a proxy for accessing session-based virtual desktops or VM-based virtual desktops. This is suitable for securing access from the Internet.

The Terminal Services functionality found in older versions of Windows Server had only session-based virtual desktops and applications. In Windows Server 2012 R2, you also can use RDS to deploy VM-based virtual desktops. Connectivity to the VMs is done by using RDP, just as in a session-based deployment.

Some benefits of using RDS for virtual desktops and applications include the following:

- **Easier application deployment and updates** A typical application deployment requires you to install and update the application on each client computer. In all but the smallest environments, this requires you to implement some type of automated deployment tools for applications. With RDS, you only need to install and update applications on the central servers. This is significantly less work than installing and updating applications on individual client computers.
- **Simplified access to data and applications** When you implement RDS, applications and their data can be accessed from anywhere. You can allow users to use applications from a computer in the office, a home computer, and mobile devices.
- **Faster access to remote data** Access to data over a virtual private network (VPN) or WAN links often results in poor application performance. For example, an application that requires access to a SQL server may be very slow if the connectivity to the SQL server has high latency. When you use RDS, you place the central servers with the application installed close to the application data, and network latency is removed as a performance problem.
- **Higher data security for mobile users** Without RDS, mobile users copy data onto a mobile computer and take it with them. Or, in some cases, they use a VPN to access data remotely while offsite. In both cases, there is a risk that the mobile computer could be lost or stolen and the data accessed by unauthorized users. When you use RDS for remote access to data, there is no need to copy data to the remote device. This mitigates the risk that your organization will lose control of the data.

- **Simplified client hardware management** Using RDS to provide virtual desktops reduces the effort to manage client device computers because the devices are performing much less work. Computers used to access virtual desktops become essentially disposable because the only configuration information they contain is the connection information to the remote desktop. In some cases, you may be able to extend hardware life because the client device is performing very little work.

Comparing RDS and the Remote Desktop feature

Remote Desktop is a feature in Windows 8.1 and Windows Server 2012 R2 that enables you to connect to a computer remotely and to view its desktop, just as when you sign in to that computer locally. The primary intention of the Remote Desktop feature is remote administration. That is why, when you enable the feature, by default only the administrator who enables it can connect to the remote desktop. Other users can connect to the remote desktop only if you grant them permission.

RDS is a Windows Server role that is available only in the Windows Server operating system. To deploy RDS, you need to install at least three role services and perform an additional configuration. RDS provides a similar experience to the Remote Desktop feature, but the primary intention of RDS is to enable users to have a standard remote environment that is available from any device and to use remote resources while integrating remote applications on the local user desktop. Table 8-1 compares RDS and the Remote Desktop feature.

Table 8-1 Comparing RDS and the Remote Desktop feature

RDS	Remote Desktop Feature
Can support many simultaneous users.	Desktop operating systems are limited to one simultaneous user. Server operating systems are limited to two simultaneous users.
Proper licensing must be purchased and configured.	No additional licensing is required.
Used to access a full remote desktop or remote applications (RemoteApp).	Used to access the full remote desktop.
Supports advanced features such as RemoteFX USB Redirection and multimedia redirection.	Does not support advanced features.
Requires an infrastructure of multiple servers that has been properly planned and deployed.	Is enabled on a single computer.

RDS architecture

There are six RDS service roles that can be included in an RDS deployment. At minimum, you need to have the Remote Desktop Connection Broker (RD Connection Broker) role service, the Remote Desktop Web Access (RD Web Access) role service, and either the RD Session Host or Remote Desktop Virtualization Host (RD Virtualization Host) role service. You can install individual RDS role services, but you won't be able to manage them unless they are part of an RDS deployment. Depending on your implementation goals, an RDS deployment can include additional RDS role services, and RDS role services can be installed on multiple servers for scalability and high availability.

Windows Server 2012 R2 RDS includes the following role services:

- **RD Session Host** This role service configures a server to provide session-based desktops and applications. Users can connect to an RD Session Host server and then run applications and use the network resources that the RD Session Host offers. RD Session Host is a required role service in a session-based desktop deployment of RDS.

- **RD Virtualization Host** This role service integrates with the Hyper-V role in Windows Server 2012 R2 to provide VMs that can be used as virtual desktops. The RD Virtualization Host role service also monitors and reports on established client sessions to the RD Connection Broker role service. This role service is responsible for managing the VMs that function as pooled and personal virtual desktops. If VMs are in a saved state, the RD Virtualization Host role service starts the VMs to prepare them for a user connection. For pooled virtual desktops, the RD Virtualization Host role service reverts the VMs to their initial state when users sign out. The RD Virtualization Host role service is required in a VM-based deployment of RDS.

- **RD Connection Broker** This role service manages connections to RemoteApp programs and virtual desktops, and it directs client connection requests to an appropriate endpoint. The RD Connection Broker role service also provides session reconnection and session load balancing. For example, when a user disconnects from a session and later establishes a connection, the RD Connection Broker role service ensures that the user reconnects to his or her existing session. This role service is mandatory in all RDS deployments, but it does not require large amounts of server resources.

- **RD Web Access** This role service provides a web-based interface to RemoteApp programs, session-based virtual desktops, or VM-based virtual desktops. A webpage provides each user with a customized view of all RDS resources that have been published to that user. This role service supports organizing resources in folders, which enables administrators to group remote applications in a logical manner. It also publishes

available RDS resources in an RDWeb feed, which can integrate with the Start screen on client devices. RD Web Access is a mandatory role service for each RDS deployment.

- **Remote Desktop Licensing (RD Licensing)** This role service manages RDS client access licenses (RDS CALs) that are required for each device or user to connect to an RD Session Host server. You use RD Licensing to install, issue, and track RDS CAL availability on an RD Licensing server. You are not required to install this role service during an initial RDS deployment, but an RDS deployment without proper licensing ceases to function after 120 days.

- **RD Gateway** This role service allows authorized remote users to connect securely to RemoteApp programs and virtual desktops from outside the organization over the Internet. An RD Gateway server acts as a proxy for external users to connect to internal RDS resources. To increase compatibility with firewalls in public locations such as hotels, RDP traffic is encapsulated in Hypertext Transfer Protocol Secure (HTTPS) packets. Access is controlled by configuring Remote Desktop connection authorization policies (RD CAPs) and Remote Desktop resource authorization policies (RD RAPs). An RD CAP specifies who is authorized to make a connection, and an RD RAP specifies to which resources authorized users may connect.

All deployment and management of RDS is done by using Server Manager, as shown in Figure 8-1. Server Manager provides an overview of all servers in an RDS deployment and a management interface for each server. RDS in Server Manager uses a discovery process to detect the role services that are installed on each machine that is added to Server Manager.

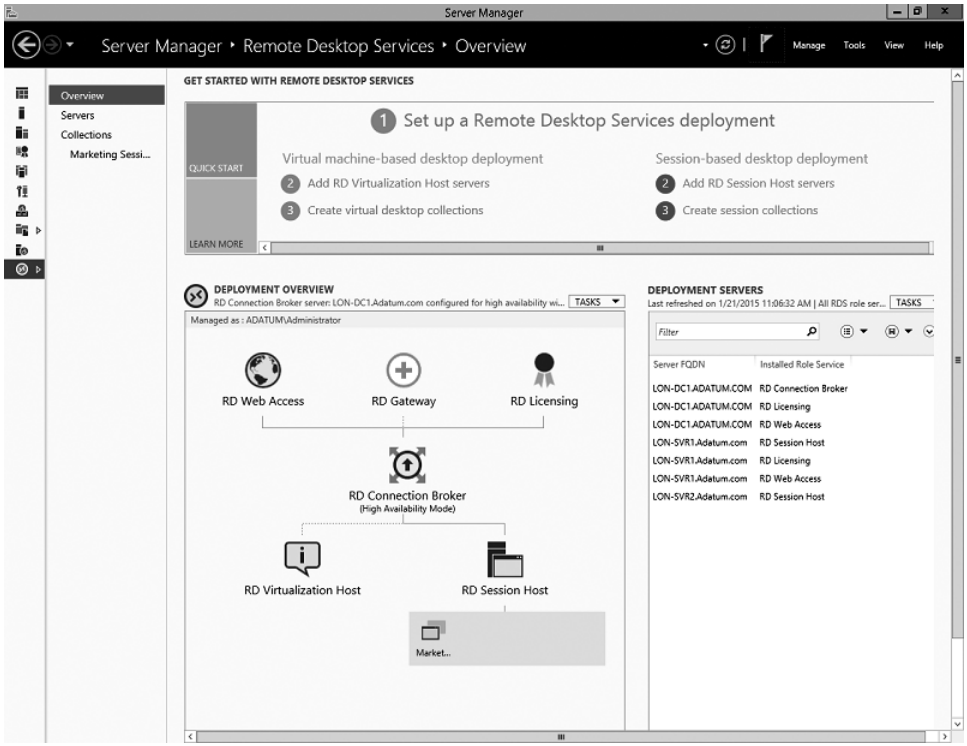

Figure 8-1 RDS configuration in Server Manager

NOTE

Legacy Remote Desktop administration tools such as Remote Desktop Services Manager and RD Session Host Configuration, which were used for configuring and administering RDS in Windows Server 2008 R2, are replaced with RDS in Server Manager in Windows Server 2012 and Windows Server 2012 R2.

Connecting to virtual desktops and RemoteApp programs

Windows client operating systems include Remote Desktop Connection (RDC), which is used to connect to virtual desktops and applications. Microsoft also provides Microsoft Remote Desktop for iOS and Android devices. All of these applications use RDP to connect to virtual desktops and RemoteApp programs.

When you use RDC to access a computer with the Remote Desktop feature enabled, you enter the IP address or DNS name of the remote computer, as shown in Figure 8-2. This type of direct connectivity doesn't work when connecting to RDS because you are connecting through

the RD Connection Broker and need to be directed to a specific collection for the RemoteApp program or virtual desktop.

Figure 8-2 Remote Desktop Connection (RDC)

After you implement servers for the RDS infrastructure, you need to create collections that define what the clients are connecting to and how it is configured. There are two types of collections:

- **Virtual desktop collections** This type of collection contains VMs hosted on RD Virtualization Host servers.
- **Session collections** This type of collection contains RD Session Host servers that provide session-based virtual desktops or RemoteApp programs.

To connect to collections in RDS, you need to have an .rdp file with the correct connectivity information for the RD Connection Broker and the collection to which you are connecting. RDC uses the connectivity information in the .rdp file.

You can create an .rdp file manually and make it available to users. When the user opens the .rdp file, RDC launches and connects to the RD Connection Broker. This method is functional but relatively complex because you need to learn the syntax for creating .rdp files and need to update them if your infrastructure changes.

The simplest way to provide user connectivity to RDS is by using RD Web Access, shown in Figure 8-3. When users connect to RD Web Access, they are provided with a list of collections to which they have access. When they click the appropriate collection, an .rdp file with the correct configuration information is generated, and RDC launches using the information in the .rdp file. This provides a consistent access method even if the RDS deployment is modified.

Figure 8-3 RD Web Access

The following process, shown in Figure 8-4, is used when clients connect to a session collection by using RD Web Access:

1. Users connect to the RD Web Access portal and identify the RDS resource to which they want to connect.

2. Users click the link on the RD Web Access portal for the RDS resource they want to access. This downloads the .rdp file, which contains information about the resource to which the user wants to connect.

3. RDC is launched, and it uses the information in the .rdp file to initiate a connection with the RD Connection Broker role service. After users authenticate to the RD Connection Broker role service, the RDC passes the request about the RDS resource to which the user wants to connect.

4. The RD Connection Broker role service examines the request to find an available RD Session Host server in the desired collection and sends the connection information back to the RDC client. If the request matches a session that already is established for the associated user, RD Connection Broker redirects the client to the server in the collection where the session was established. If the user doesn't have an existing session in the collection, the client redirects to the server that is most appropriate for the user

connection, based on the RD Connection Broker load balancing algorithm—for example, weight factor, fewest connections, and least utilized.

5. The RDC client establishes a session with the RD Session Host server that RD Connection Broker provided.

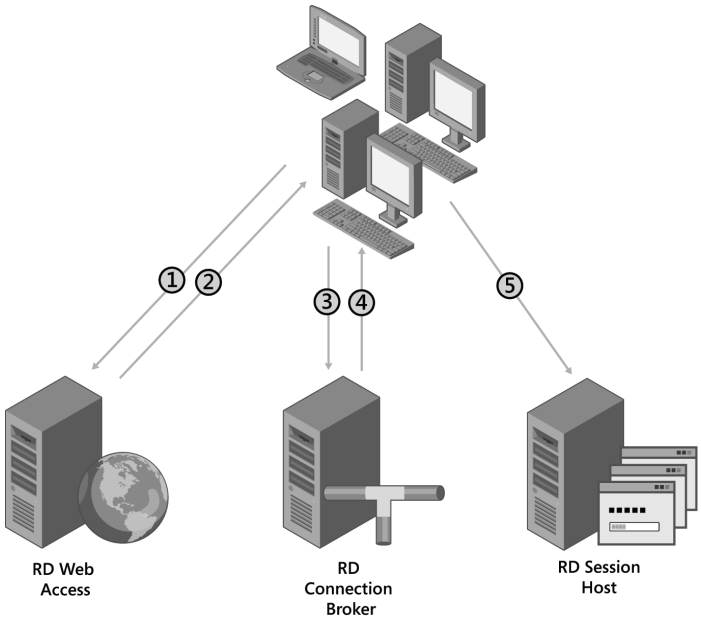

Figure 8-4 Connectivity for session collections

RDS functionality that enhances the client experience

RDC uses the RDP protocol to connect to RDS servers. The following are some of the specific features available that enhance the client experience:

- **Bandwidth reduction features** When an RDP connection is established, various methods to reduce network bandwidth are used, such as data compression and caching. Caching enables an adaptive user experience over LANs and WANs. Clients can detect available bandwidth and adjust the level of graphic detail that is used.

- **Full desktop or application window only** When a client connects to RDS, it can display either a full remote desktop or only the window of a remotely running application (RemoteApp program). With full desktops, users can perform remote administration or run multiple applications. However, the user must deal with two desktops: local and

remote. RemoteApp programs integrate with local desktops, but they still require network connectivity to RDS.

- **RemoteApp programs that look and feel like locally installed applications** The window displayed when you connect to a RemoteApp program looks like a locally installed application. Links to RemoteApp programs can be added to a client's Start screen. RemoteApp program icons support pinning, tabbed windows, live thumbnails, and overlay icons. RemoteApp windows can be transparent, and the content of a RemoteApp window displays while you are moving it.

- **Reconnection to existing sessions** If a user disconnects from a remote desktop session, the user can reconnect to the session and continue to work from the point at which he or she disconnected. The user can connect from the same device or from a different client device. If a session disconnects for a different reason, for example, because network connectivity is lost, the user automatically reconnects to the disconnected session when network connectivity is restored.

- **Redirection of local resources** Client resources such as drives, printers, the Clipboard, smart card readers, and USB devices can redirect to a remote desktop session. This enables you to use locally attached devices while working on RDS and to use the Clipboard to copy content between a local and remote desktop. You even can redirect USB devices that you plug in when the remote desktop connection already is established.

- **Windows media redirection** This feature provides high-quality multimedia by redirecting Windows media files and streams from RDS to a client. When Windows Media Player is used in a session-based virtual desktop, the multimedia file is not rendered on the RD Session Host. Instead, the multimedia stream is redirected to the RDC client and is rendered on the client. This reduces load on the RD Session Host and provides higher quality audio and video playback on the client. If the RDC client does not have the necessary codec for the multimedia content, then the content is rendered on the RD Session Host.

- **Multi-monitor support** This feature enables support for up to 16 monitors of any size, resolution, and layout. Applications function just as they do when you run them locally in multi-monitor configurations.

NOTE

Multi-monitor support requires RDC version 7.0 or later. This software is included with Windows 7 and later. If you are connecting to a computer running Windows 7, multi-monitor support is available only for the Ultimate and Enterprise editions. If you are connecting to a computer running Windows 8 or Windows 8.1, multi-monitor support is available only for the Professional and Enterprise editions.

- **Single sign-on (SSO)** When users connect to RDS, they have to provide their credentials again. With SSO, a user can connect to a remote desktop or start a RemoteApp program as the user who signed in to the local computer, without reentering credentials.

- **CPU, disk, and network Fair Share** Fair Share features are enabled by default on RD Session Host servers to ensure even resource distribution among users. One user can't monopolize resources or negatively affect the performance of other users' sessions. Fair Share can distribute network, disk, and CPU resources dynamically among user sessions on the same RD Session Host server. You can control Fair Share settings through Group Policy.

RemoteFX

RemoteFX introduces a set of enhancements to RDP that enables rich graphics and video capabilities within a remote desktop session, regardless of whether you are connecting to a session-based virtual desktop, running a RemoteApp program, or connecting to a VM-based virtual desktop. In all three cases, the user experience is almost identical to using a local physical desktop. RemoteFX is included in RDS, and you don't need to enable it explicitly unless you want to use the RemoteFX virtual graphics processing unit (vGPU) on a VM-based virtual desktop. In that case, you must add hardware to the VMs that are used for the virtual desktop.

The following is a list of some RemoteFX features:

- **RemoteFX for WAN** This feature delivers an improved user experience over lower-speed networks, such as at a branch office, on a wireless device, or working from home over a WAN connection. RemoteFX for WAN combines the RemoteFX Adaptive Graphics feature with intelligent WAN-aware transports. TCP and UDP can be used for remote desktop connections. The protocol that is better suited for the current connection is selected automatically, and automatic detection of network conditions to adjust the encoding of content is available.

- **RemoteFX Adaptive Graphics** This feature dynamically adapts to changing network conditions and optimizes encodings based on the content delivered. RemoteFX Adaptive Graphics use multiple codecs, which are optimized for different types of content, such as text, images, and video.

- **RemoteFX Media Streaming** This feature provides redirection of multimedia content. When a user attempts to play multimedia content in a remote session, the content is intercepted and redirected to the client. The client receives the compressed content, decodes the content, and plays it back locally.

- **RemoteFX Multi-Touch** This feature extends the Windows 8.1 touch experience to devices on which multi-touch is the primary means of user interaction. Windows 8.1 users are able to interact with remote desktop sessions in the same way as

a local desktop, including support for multi-touch gestures and the ability to navigate between local and remote sessions by using touch.

- **RemoteFX USB Redirection** This feature enables devices to redirect at the USB level. Because of this, no device drivers are required on the client computer, and any USB device—including audio, storage, all-in-one printers, and scanners—can be redirected.

Inside OUT

vGPU for VM-Based Virtual Desktops

An RD Virtualization Host server can suffer from high processor utilization due to graphics processing in the VMs it's hosting. This occurs because the processors in the RD Virtualization Host server do the graphics processing for each VM. This can limit the scalability of the RD Virtualization Host server.

One solution for reducing processor load on the RD Virtualization Host server is to use the vGPU functionality that is available in Windows Server 2012 R2. When you use the vGPU functionality, the VMs can use a dedicated graphics processor in the RD Virtualization Host server for graphics processing. From a performance perspective, this is like putting a more advanced video card in a desktop computer.

To use vGPU on an RD Virtualization Host server, you must meet the following requirements:

- **Second Level Address Translation (SLAT) support** The processor in the RD Virtualization Host server must include support for SLAT. For Intel processors, this is called Extended Page Tables. For AMD processors, this is called Nested Page Tables.
- **Supported video adapter** The video adapter in the RD Virtualization Host server must be DirectX 11-capable with a Windows Display Driver Model (WDDM) 1.2–compatible driver. WDDM 1.2 was introduced with Windows 8.
- **Windows 7 Enterprise with SP1 or Windows 8 Enterprise** Only the Enterprise editions of Windows 7, Windows 8, and Windows 8.1 support the use of vGPU in the VM.
- **Generation 1 VMs** The vGPU functionality isn't supported for generation 2 VMs in Hyper-V. You must configure VMs as generation 1 VMs.

If you meet the requirements for using vGPU, then you can add a RemoteFX 3D Video Adapter to VMs. In the configuration for the RemoteFX 3D Video Adapter, you can configure a maximum number of monitors and maximum monitor resolution.

Remote Desktop Connection configuration options

When you connect to a virtual desktop through RDS, RDC is configured automatically by using an RDP file that is provided by the RD Web Access server. When you use RDC to connect to a server or client with the Remote Desktop feature enabled, you can configure the connectivity settings manually. The configuration options are grouped on several different tabs. Microsoft Remote Desktop for iOS and Android have similar configuration options but different user interfaces.

On the General tab, you can specify the computer to which you want to connect by using RDC and user credentials. You also can save RDC settings in a text file with an .rdp file name extension to initiate a connection later without configuring RDC settings again.

The Display tab is shown in Figure 8-5. On this tab, you can choose the size of the remote desktop window, including the option to run the remote desktop in full-screen mode. You can select to use all local monitors for a remote session, select color depth, and enable a connection bar when the remote desktop is running in full-screen mode.

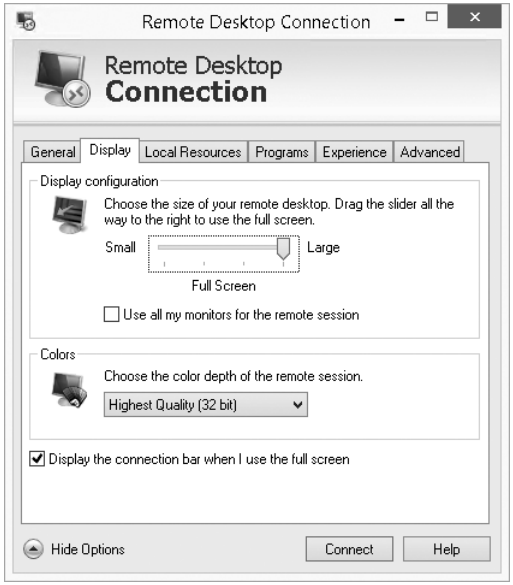

Figure 8-5 Remote Desktop Connection, Display tab

The Local Resources tab is shown in Figure 8-6. On this tab, you can set remote audio settings, such as whether you want to enable remote audio playback and recording. You also can specify a location where Windows key combinations, such as Alt+Tab, are applied and whether local devices and resources in remote sessions are available. For example, you can enable the

option to make the Clipboard, local drive, printers, and devices that you plug in later available in a remote session.

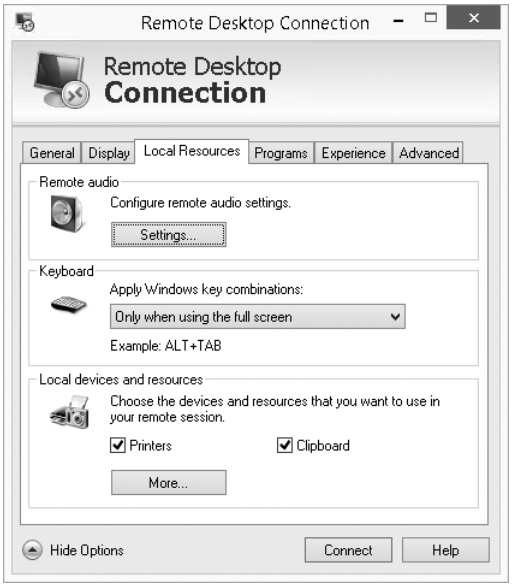

Figure 8-6 Remote Desktop Connection, Local Resources tab

On the Programs tab, you can specify a program that starts automatically in a remote desktop session when you connect to a remote computer. If you configure this option, when you close the program, your session is signed out automatically.

On the Experience tab, you can select a connection speed to optimize performance. You can enable different features, such as the following:

- Desktop background
- Font smoothing or visual styles in RDC
- Show window contents while dragging

By default, RDC automatically detects connection quality and configures connection quality–dependent features accordingly. On this tab, you also can configure persistent bitmap caching and automatic reconnection if a connection drops.

RDC displays the bandwidth with an icon on the connection bar (top of the window) that is similar to a signal strength meter. The meter is based only on bandwidth and does not take

latency into account. The number of bars in the icon identify the bandwidth, as show in Table 8-2.

Table 8-2 RDC bandwidth values

Icon	Bandwidth
4 bars	10 megabits per second (Mbps) and higher
3 bars	2000–9999 kilobits per second (Kbps)
2 bars	512 Kbps - 19999 Kbps
1 bar	Less than 512 Kbps
No icon shown	No bandwidth detected or older remote desktop host

On the Advanced tab, you can configure server authentication and Connect From Anywhere settings. The server authentication options allow you to define what should be done if the certificate provided by the server during authentication isn't valid. By default, a warning is displayed and you have the option to continue. If desired, you can configure this setting to connect without warning or prevent connections.

The Connect From Anywhere settings allow you to configure connectivity through an RD Gateway server. You can configure the alternate credentials for authentication to the RD Gateway server and the location of the RD Gateway server.

RDS licensing

If you want to use RDS, you need to purchase additional RDS CALs for each user or device that uses RDS. This is in addition to the typical licensing that is required for desktop computers. For example, in an environment where users have desktop computers and some applications are delivered by RemoteApp, you would need the following licenses:

- Operating system license for the desktop computer
- Server licenses for the Windows-based servers that deliver the RemoteApp programs
- Windows CALs for each user or computer that accesses the Windows servers
- RDS CALs for each user or desktop that uses RemoteApp programs
- Application licenses for each user or desktop that uses RemoteApp programs

RDS CALs provide users with access to session-based virtual desktops or RemoteApp programs. Licensing for VM-based virtual desktops is slightly more complex because the operating system for the VM also needs to be licensed. If you connect to a VM-based virtual desktop from a device that is covered by a Microsoft Software Assurance agreement, then the license

includes rights to use that same operating system in a VM-based virtual desktop. If the device isn't covered by a Microsoft Software Assurance agreement, then you need to purchase Windows Virtual Desktop Access (Windows VDA) licenses.

➤ **For more information about licensing VM-based virtual desktops, see Chapter 10, "Planning and implementing pooled and personal virtual desktops."**

When a client attempts to connect to an RDS deployment, the server that accepts the connection determines if an RDS CAL is needed. If an RDS CAL is required, then the server requests the RDS CAL on behalf of the client that is attempting the connection. If an appropriate RDS CAL is available, it is issued to the client, and then the client can connect to RDS.

RD Licensing manages the RDS CALs that are required for each device or user to connect to an RD Session Host server. You use RD Licensing to install, issue, and track the availability of RDS CALs on an RD Licensing server. At least one RD Licensing server must be deployed in the environment. The role service can be installed on any server, but for large deployments, the role service should not be installed on an RD Session Host server.

After an RDS installation, there is an initial grace period of 120 days. This grace period begins after the RD Session Host accepts the first client connection. If you have not installed valid licenses by the time the grace period expires, clients will not be able to sign in to the RD Session Host.

Inside Out

Licensing modes

Each RD Session Host server is configured with a licensing mode. The licensing mode determines the type of RDS CALs that an RD Session Host server requests from an RD Licensing server on behalf of a client that is connecting to an RD Session Host server. There are two licensing modes:

- **Per User** **This gives one user the right to access any RD Session Host server in an RDS deployment from an unlimited number of client computers or devices. You should use RDS Per User CALs when the same user connects to RDS from many devices.**
- **Per Device** **This gives any user the right to connect to any RD Session Host server in an RDS deployment from a specific device. When a client connects to an RD Session Host server for the first time, a temporary license is issued. When the client computer or device connects to an RD Session Host server for the second time, if the license server is activated and enough RDS Per Device CALs are available, the license server issues the client computer or device a permanent RDS Per Device CAL. You should consider RDS Per Device CALs when multiple users use the same device for connecting to RDS, for example, a point-of-sale device that is used by different clerks.**

A single RDS deployment can be configured with only one licensing mode. If you need a mix of Per User and Per Device RDS CALs, then you need to implement two RDS deployments.

> NOTE
>
> **A permanent RDS Per Device CAL is valid for a randomly selected number of days between 52 and 89. If the RDS Per Device CAL isn't renewed, then it is returned to the available licenses on the RD Licensing server. You can revoke RDS Per Device CALs before they expire and return them immediately to the available licenses. You are limited to revoking a maximum of 20 percent of the RDS Per Devices CALs.**

If you need to provide access to RDS for multiple external users who are not employees of your organization, then you should consider using an RDS External Connector License. An RDS External Connector License allows an unlimited number of nonemployees to connect to a specific RD Session Host. If you have multiple RD Session Host servers, you need multiple RDS External Connector Licenses in addition to any required Windows Server External Connector Licenses.

Planning infrastructure for session-based desktops

The planning for implementing RDS for session-based desktops can be fairly complex compared to other Windows-based role services. Most Windows-based role services require only one server. RDS requires at least three role services and, in most cases, the role services are spread across multiple servers. You should be aware of the functionality that each role service provides. You also should be aware of how an RDS deployment uses each role service. You need to know role service requirements and which hardware resources are most critical for each role service.

Assessing RDS infrastructure requirements

Before you implement RDS, you must determine your organization's requirements. To do so, you first must evaluate if RDS is the appropriate solution for your needs, and then you must choose between session-based and VM-based desktop deployments. If necessary, an RDS deployment can include both session-based and VM-based desktop deployments. You also must evaluate the existing server infrastructure and estimate the required server hardware, network bandwidth, client types and requirements, and connectivity needs for a successful RDS deployment.

Determine your RDS needs

To determine if RDS is an appropriate solution for your needs, you should assess and analyze the types of users, hardware, and applications in your organization. Areas of consideration include the following:

- **User types** Do you have users in remote locations, single-task users, contractors, and other types of users who would benefit from remote applications or virtual desktops?
- **Hardware** What client hardware currently is deployed in your organization? Would it be beneficial to move from traditional desktops to thin clients for some users? Do you allow users to bring their own devices into the organization's network? Do users wish to use mobile devices to run certain applications?
- **Application compatibility** Can the applications run in a multiuser environment? If not, will the applications run in a virtual environment?
- **Application performance** How do the applications perform in a remote or virtual environment? Keep in mind that many applications perform better as RemoteApp programs on RDS because processing takes place on a server.
- **Application support** Do vendors support the applications in a virtual or multiuser environment? Do vendors provide support to multiple users?
- **Licensing** Can the applications be licensed for a virtual or multiuser environment?
- **Business benefits** Are there justifiable business reasons to implement this solution? Potential benefits include cost savings, reduced deployment time, centralized management, and reduced administration costs.
- **Legal requirements** Because of financial and legal requirements, some organizations mandate that applications and data remain on-premises. RDS enables users to connect to a standard virtual desktop to use familiar applications and to work with data from almost any device, while organizational data stays in the data center.

Choosing between session-based and VM-based desktop deployments

RDS has two deployment types:

- **Session-based virtual desktop deployment** This provides users the ability to connect to an RD Session Host and use a full desktop or run remote applications and present them on a client as if they were installed locally.
- **VM-based virtual desktop deployment** This provides users with access to a full Windows client operating system that runs on a VM, for example, Windows 7 or Windows 8.1.

You need to decide which RDS deployment type is best for your environment based on various requirements. For example, you must consider if users must be completely isolated or if they must have administrative access. You should consider whether the applications work properly in a multiuser environment. In addition, you must consider whether you can install and run

applications on Windows Server. Remember that a VM-based virtual desktop deployment typically requires a more powerful server infrastructure and more disk storage than a session-based virtual desktop deployment for the same number of users. For some applications, VM-based virtual desktops might be the only viable solution.

Generally, you should choose session-based virtual desktops if possible. Session-based virtual desktops support a larger number of users than VM-based virtual desktops on the same hardware.

Determine server hardware and network resource requirements

Once you determine the RDS deployment benefits for your organization, you must consider the hardware requirements to support your users, including the following:

- **Number of users** How many users will use RDS, and where are they located?
- **User types** How many users run CPU-intensive and bandwidth-intensive applications? Will you have to provide more bandwidth and server hardware to support expected usage?
- **Connection characteristics** How many concurrent connections do you expect? Can your server and bandwidth resources handle peak usage times?
- **Application silos** Will you have to create multiple server collections to support different applications that might not be able to run on the same server?
- **Load balancing** Will you have to include multiple servers in a collection to spread the load among the servers? This increases available resources and provides redundancy.
- **High availability** What is the organization's tolerance for downtime? Do you need close to zero downtime, or could your organization tolerate the time it would take to restore from backups?
- **Expansion considerations** What are the growth expectations? At what point will new resources need to be brought online?

Determine user requirements

Another aspect to consider is user requirements. A large organization with multiple locations might have a number of mitigating factors to consider, such as the following:

- **Languages** Organizations with a global presence need to support multiple languages. You might need to install language packs on all of your RDS servers.

- **Profile management** How will you store user states? Do users require the same user state when they sign in locally and to an RDS session? Which type of Windows user state virtualization will be used?
- **Printing** Will existing printers function properly in a remote desktop environment? Will there be problems finding printer drivers to support existing printers? Is there a budget to replace older printer models?

Determine how clients access RDS

Clients can connect to RDS in various ways. You probably will need to provide different access methods for different groups of users. Areas to consider include the following:

- Will you allow users to connect over the Internet from remote locations? If so, you will need to set up an RD Gateway and obtain certificates.
- How will you handle Secure Sockets Layer (SSL) certificates—by using certificates from non-Microsoft certification authorities (CAs) or by using certificates that an internal CA issues?

Based on your assessment results, start designing your RDS deployment. You should identify RDS role services that are required and that you will deploy. You also should determine the number and hardware configuration of servers that are required, in addition to planning required storage, connectivity, and firewall configuration.

Planning for the RD Session Host role service

The RD Session Host role service provides Windows-based apps or full Windows desktops for RDS clients. This role service is mandatory for every RDS deployment that provides users with session-based desktops or RemoteApp programs. An RD Session Host server accepts incoming RDP requests, and after a client authenticates, it provides a desktop-based or application-based session to the client. An RD Session Host server is the central location where remote applications are installed, accessed, and maintained.

To plan the deployment of an RD Session Host server, you must consider the number of installed applications, the type of applications, resource use, the number of connected clients, and the type of user interaction. While connected to one RD Session Host, users might run a simple application that has low resource utilization and rarely runs, for example, an old data entry application. On another RD Session Host, users often might run a resource-intensive graphical application that requires many CPU resources, a considerable amount of RAM, intensive disk I/O operations, and that causes a lot of network traffic. If the hardware configuration on both of the RD Session Hosts is the same, the second server is considerably more utilized and can accept fewer user connections.

RD Session Host planning focuses on the number of concurrent users and the workload they generate. A server with a particular hardware configuration might support many simultaneous users or only a few, depending on their usage patterns and the applications that they are running on the RD Session Host.

The following are the main resources that you should consider when estimating RD Session Host utilization:

- **CPU** Each remote application that users start runs on an RD Session Host and utilizes CPU resources on the RD Session Host. In an environment where many users are connected to the same host, CPU and memory typically are the most critical resources.
- **Memory** Additional memory must be allocated to an RD Session Host for each user who connects to the RD Session Host, whether connecting to a full Windows desktop or running a RemoteApp program.
- **Disk** Because user state typically isn't stored on an RD Session Host, disk storage usually isn't a critical resource. However, many applications run simultaneously on an RD Session Host, and the disk subsystem should be able to meet their disk I/O needs.
- **Network** The network should provide enough bandwidth for connected users and for the applications that they run. For example, applications that use a SQL database use the network to connect to that SQL database. Also remember to consider the network bandwidth required to support the user connectivity to the RD Session Host server.
- **GPU** Applications that are graphically intensive, especially those that include three-dimensional graphics, might require vGPU support and RemoteFX to perform well. Without such support, graphics render on the server's CPU and may limit the number of users on the RD Session Host to a relatively small number.

When estimating the required resources for an RD Session Host, you can use one of the following methods:

- **Pilot deployment** This is a common and a simple approach. You first need to deploy RDS in a test environment and capture its initial performance. After that, you start increasing server load by increasing the number of users and monitoring response times and user feedback. You can find out how many users can connect to an RD Session Host and still have an acceptable user experience based on the number of users and the system response time. Based on the findings, you can estimate the number of servers that are needed for a production environment. This approach is reliable and simple, but it requires initial investments for the pilot deployment.
- **Load simulation** This method also uses an initial RDS deployment in a test environment. You need to gather information about applications that users operate and how

users interact with the applications. After that, you can use load simulator tools to generate various levels of typical user loads against an RDS deployment. When a load simulator tool runs, you need to monitor server utilization and responsiveness. This method is similar to the pilot deployment method, but it uses a load simulation tool instead of real users to generate user load. It also requires an initial investment, and its results depend on the initial estimation of actual user usage.

- **Projection based on single-user systems** This method uses data that is collected from a single-user system for projecting expected utilization on an RD Session Host with multiple user sessions. This method requires detailed knowledge of applications that are used, and it usually is not very reliable because a single-user system has a different overhead than a multiuser system.

It is critical that you plan for future scalability of an RDS deployment. User needs for applications will change over time, and you need to be ready to expand your RDS deployment to meet those needs. In some cases, you may be able to scale up the capacity of the individual servers with additional processors or additional memory. Scaling up by using more powerful servers tends to be expensive. Scaling out by adding servers generally is less expensive.

Fortunately, you can scale out an RDS deployment for session-based virtual desktops and RemoteApp programs by adding RD Session Host servers. For example, if you have an RDS deployment for session-based virtual desktops that uses two RD Session Host servers, and those two servers are experiencing frequent peaks of 100 percent CPU utilization, you can add a third RD Session Host server. The RD Connection Broker then automatically load balances the connections across three servers instead of two and reduces the CPU utilization on the two existing servers.

Planning for the RD Connection Broker role service

During RDS deployment planning, you must designate a server on which to install the RD Connection Broker role service. The RD Connection Broker role service is required in each RDS deployment. It provides users with access to RemoteApp programs, session-based virtual desktops, and VM-based virtual desktops. The RD Connection Broker role service manages all aspects of session connectivity. Functions performed by the RD Connection Broker role service include the following:

- **Routes connection requests** Determining the most appropriate RD Session Host or virtual desktop to which to send a connection request, based on a user's identity and the current load on RD Session Host or RD Virtualization Host servers.

- **Stores information about connections to VMs and sessions** By default, connection information is stored in the Windows Internal Database (WID) on an RD Connection Broker server. By storing this information, the RD Connection Broker role

service can reconnect users to the same session in an RDS deployment with multiple RD Session Host servers.

- **Configures RDS servers in the same group (collection)** You configure settings—for example, session settings or certificates—once, and RD Connection Broker applies the settings to servers in the collection.
- **Manages VM creation and deletion** In VM-based desktop deployments, RD Connection Broker manages VM creation and deletion for managed collections, and it assigns personal virtual desktops to users.
- **Provides information to RD Web Access servers** The RD Connection Broker role service gathers collection information about RemoteApp programs, session-based virtual desktops, and VM-based virtual desktops.

When a user initiates a session, the session request is received by the RD Connection Broker role service, which queries the database to determine if there is an existing disconnected session for that user. If so, the user is directed to the disconnected session. If not, the RD Connection Broker role service determines the server in the collection that is best able to handle the new connection, based on the load-balancing algorithm.

Inside OUT

RD Connection Broker scalability

Performance for an RD Connection Broker server depends on the number of requests it receives in a given time frame. Users generate a load on the RD Connection Broker server only when they perform the initial connection to a RemoteApp program or a session-based virtual desktop. This makes it different from an RD Session Host server, for which performance depends on the number of simultaneous users.

A single RD Connection Broker server with four processor cores and four gigabytes (GB) of RAM can process approximately 10 connections per second with a maximum processing time of 1 second. If you are willing to increase the maximum processing time to 12 seconds, that same server can process approximately 60 connections per second. For more detailed information about RD Session Broker scalability, see the white paper "RD Connection Broker Performance and Scalability" at *http://go.microsoft.com/fwlink/?LinkID=510038&clcid=0x409*.

When you consider the number of connections per second to which your RD Connection Broker server will be subject, you need to consider peak utilization times. First thing in the morning when users arrive and after lunch likely are peak times when users are connecting.

A single RD Connection Broker server can handle a large number of connection requests, and for performance, your RDS deployment may require only one. A more critical consideration for the RD Connection Broker role service is availability.

The RD Connection Broker role service is an entry point to an RDS deployment, and it is critical that it is available all the time. If the RD Connection Broker role service isn't available, then clients can't connect to RemoteApp programs or virtual desktops, but existing connections to RemoteApp programs and virtual desktops continue to function properly. When an RDS deployment only has one RD Connection Broker server, the server represents a single point of failure. To make the RD Connection Broker role service highly available or to increase scalability, you can add RD Connection Broker servers.

➤ **Configuring high availability for the RD Connection Broker role service is covered in more detail later in this chapter in the section titled "Understanding high availability for RDS."**

Planning for the RD Web Access role service

The RD Web Access role service is a mandatory part of each RDS deployment, and it installs the Web server role, Internet Information Services (IIS), as its prerequisite. The benefits of RD Web Access include the following:

- From almost anywhere, authorized users quickly can access a list of available RemoteApp programs, remote desktops, and virtual desktops on a webpage.
- A list of available RDS resources publishes automatically via an RDWeb feed, and it can integrate with the Start screen on the client.
- Changes in available RDS resources update automatically on clients that have subscriptions to an RDWeb feed.
- Users can launch the RDC client from the RD Web Access portal, which enables them to connect remotely to the desktop of any computer on which they have Remote Desktop access.
- RD Web Access and RDWeb feeds are personalized and show only RDS resources for which users have permissions.
- Administrators can customize an RD Web Access portal without programming.

➤ **More information about customizing RD Web Access is provided in Chapter 9, "Configuring RemoteApp programs and client connectivity."**

It's important to remember that the RD Web Access role service only provides a link to launch RemoteApp programs or to connect to a Remote Desktop session. The RD Web Access role

service doesn't proxy client requests. When a user connects to a RemoteApp program or a virtual desktop, the client establishes a direct connection to the target server.

Performance considerations for an RD Web Access server are similar to those for an RD Connection Broker server because the RD Web Access role service provides only initial connectivity to RemoteApp programs and virtual desktops. After users are connected to requested resources, the RD Web Access role service is no longer used. Therefore, RD Web Access server performance needs to be designed to accommodate usage at peak times like morning arrivals and after lunch. If required for high availability or scalability, you can implement multiple RD Web Access servers and load balance them.

Planning for preserving user state

In a session collection with multiple RD Session Host servers, the connections from clients are load balanced across the RD Session Host servers by the RD Connection Broker server. By default, when a user connects to a specific RD Session Host server, a local profile is created for that user on the RD Session Host server. The next time a user connects, the RD Connection Broker may direct the client to a different RD Session Host server, where a different local profile is created. Each time users sign in, they may be using a different profile on a different RD Session Host server. This means that user state information such as application configuration, Desktop configuration, Favorites, and Documents are not the same across sessions. To provide a consistent user experience, you should preserve user state across multiple RD Session Host servers.

If users have desktop computers and session-based virtual desktops, you also need to consider whether you want user state to be preserved between desktop computers and the virtual desktops. This can be complicated by the fact that session-based virtual desktops may not have the same configuration as the desktop computers, and, consequently, it may not make sense to synchronize all of the user state information. For example, synchronizing Desktops may result in desktop shortcuts appearing that point to applications that are not available on the RD Session Host servers.

Roaming profiles

Roaming user profiles can be used to synchronize user state, but they synchronize entire user profiles. This typically is not desired for session-based desktops because not all user state information needs to be synchronized between desktop computers and RD Session Host servers. If you use roaming profiles for the desktop computers in your organization and you want to ensure that roaming profiles are not used on the RD Session Host servers, then you can configure the msDS-PrimaryComputer attribute for users and enable the Download Roaming Profiles On Primary Computers Only Group Policy setting.

You also can set user properties for roaming user profiles that are specific to RD Session Host servers, as shown in Figure 8-7. If you configure the Profile Path, then a user connecting to

a session-based virtual desktop uses the specified profile path rather than a roaming profile configured on the Profile tab. Effectively, the RDS user profile becomes a roaming profile used only when connected to an RD Session Host server.

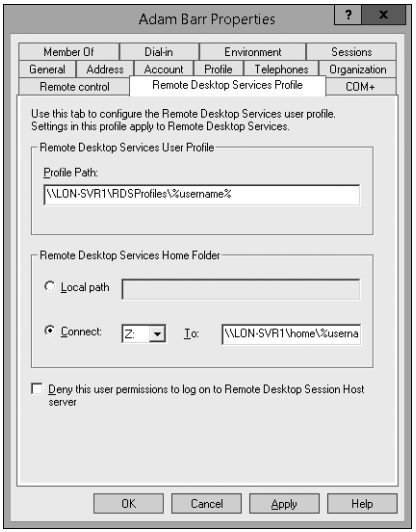

Figure 8-7 User Properties, Remote Desktop Services Profile tab

Instead of configuring individual user accounts with RDS-specific profiles, you can use Group Policy. In a Group Policy object that applies to the RD Session Host servers, you can configure settings in Computer Configuration\Policies\Administrative Templates\Windows Components\Remote Desktop Services\Remote Desktop Session Host\Profiles. There are two relevant settings:

- **Set Path For Remote Desktop Services Roaming User Profile** Specify a UNC path for storing all user profiles. A subfolder for each user is created automatically.
- **Use Mandatory Profiles On The RD Session Host Server** Indicates that the path specified in the Set Path For Remote Desktop Services Roaming User Profile setting contains a mandatory profile that can't be modified. When this setting is enabled, the UNC path for profiles does not contain subfolders for each user.

Folder redirection

Folder redirection also is an option for users with session-based virtual desktops. You can redirect only the folders that are suitable for use on the virtual desktops and desktop computers. Commonly redirected folders include Documents, Favorites, and AppData\Roaming.

If you use folder redirection for desktop computers and don't want folder redirection used when users sign in to the RD Session Host servers, you can use the msDS-PrimaryComputer attribute in user accounts just as you can for roaming profiles. In addition to configuring the attribute, you need to enable the Redirect Folders On Primary Computers Only Group Policy setting.

User profile disks

RDS in Windows Server 2012 and newer offers the option to configure user profile disks to preserve user state across sessions. A user profile disk is a VHDX file that is mounted to the user's profile path at C:\Users\%username% on the RD Session Host. The user profile disk is mounted during sign in. During a user's session, all changes to the profile write in his or her VHDX file, and when the user signs out, his or her profile disk is unmounted. The administrator specifies the maximum size of user profile disks and can limit which folders in a user profile are included in or excluded from a user profile disk.

User profile disks are configured individually for each session collection and can't be shared among collections. A share is specified in the collection configuration to store the user profile disks. All RD Session Host servers in the collection have access to the user profile disks in the share. This provides users with consistent user state from any RD Session Host server in the collection.

User profile disks can be used in conjunction with folder redirection and roaming user profiles. Folder redirection will reduce the size of user profile disks and allow the redirected folders to be accessed from desktop computers. Roaming user profiles are synchronized with the user profile disk.

From a server management perspective, one benefit of user profile disks is controlling the amount of data stored on the C drive of RD Session Host servers. Large user profiles stored on RD Session Host servers can cause the C drive to run low on space and cause performance issues. Because user profile disks are stored on a network share and mounted in C:\Users, the C drive never is used to store profile data.

The primary consideration when planning user profile disks is ensuring that the necessary disk space is available for network storage. To ensure that network storage is sufficient, you need to determine the average user profile size. The amount of storage that you need to allocate for user profile disks is the average user profile size times the number of users plus an allowance for growth in both the number of users and the average profile size.

User profile disks are dynamically expanding VHDX files. By default, the maximum size of a user profile disk is 20 GB, but you can set this to be larger or smaller depending on the needs of your users.

When you configure the share for user profile disks, all RD Session Host servers need to have Full Control permissions. This allows the RD Session Host servers to create and manage the user profile disks. When you configure a collection with user profile disks, these permissions are assigned automatically.

Inside OUT

Infrastructure testing prior to rollout

After you assess RDS infrastructure requirements and familiarize yourself with RDS and its role services, you should perform a proof-of-concept (POC) deployment. POC deployment is critical for a successful RDS deployment. It enables you to evaluate whether all the requirements are met and to perform a load simulation, which simulates typical user actions and validates your estimates for capacity, application workloads, and usage patterns by performing a test run in a controlled environment. During testing, you should find answers to the following questions:

- **How many users can connect, and what is an average response time?** Can POC deployment support the expected number of RD users, and is the response time acceptable? How utilized are servers, how long do user sign-in and sign-out take, and is the user experience as expected?
- **How does the application consume system resources?** Does it do so in accordance with documented estimates? If the application uses hardware as expected, the rest of the deployment can continue based on initial estimates. If it doesn't use hardware as expected, you must recalculate capacity requirements to ensure accurate estimates.
- **Are all of the potential user environment scenarios being tested?** You should test the application by accessing it in all the ways a user might use it. If there are access methods that you can't replicate in a POC environment, these access methods should be implemented in a controlled manner when performing the final deployment.
- **Are the applications and hardware running as expected?** Is additional performance tuning required? Do you need to perform any additional application configuration to run as expected in an RDS environment? Also, confirm that hardware performance is within estimated parameters.
- **Are there any unexpected changes in usage or access?** If any part of the presentation virtualization POC deployment does not reflect your production environment, alter the POC deployment so that it is as similar as possible to your final, planned infrastructure.

Using testing to eliminate errors in a deployment is important because problems with a presentation virtualization environment are much easier to resolve during testing than during full deployment.

Windows Server 2012 R2 includes a Best Practices Analyzer (BPA) for the Remote Desktop Services server role. BPA for RDS can analyze an RDS environment and check for changes that need to be made for RDS to perform optimally. You can access BPA in Server Manager or by running the Invoke-BpaModel cmdlet.

Deploying session-based virtual desktops

RDS includes multiple role services. If you use Server Manager for RDS deployment, you should be aware that if you use role-based or feature-based installation, you can install individual RDS role services. However, if you install an RDS role service in this way, you can't manage it. If you want to manage RDS, a deployment must have at least three role services: RD Connection Broker, RD Web Access, and either RD Session Host or RD Virtualization Host. Individual RDS role services can't be managed if they are not part of an RDS deployment.

Understanding the session-based desktop deployment process

You can deploy RDS by using Server Manager or Windows PowerShell. Server Manager has the ability to install the necessary server roles, role services, and features on multiple servers that are part of an RDS deployment. All management of RDS also can be done from Server Manager.

Inside Out

Adding servers to Server Manager for RDS deployment

Server Manager can be used to manage the local server and remote servers, but you need to add the remote servers manually before they can be managed. A typical RDS deployment has multiple servers, and you should add each of the servers to Server Manager before you begin the deployment process.

To add a server to Server Manager, perform the following steps:

1. **In Server Manager, click Manage and click Add Servers.**
2. **In the Add Servers window, on the Active Directory tab, in the Name (CN) box, type the name of the server and click Find Now.**
3. **Double-click the server you want add and then click OK.**

The high-level steps for deploying session-based virtual desktops are as follows:

1. **Start the RDS installation** In Server Manager, use the Add Roles And Features Wizard to select the Remote Desktop Services Installation option, shown in Figure 8-8. This option configures the wizard to collect the information necessary to perform a deployment of RDS across multiple servers.

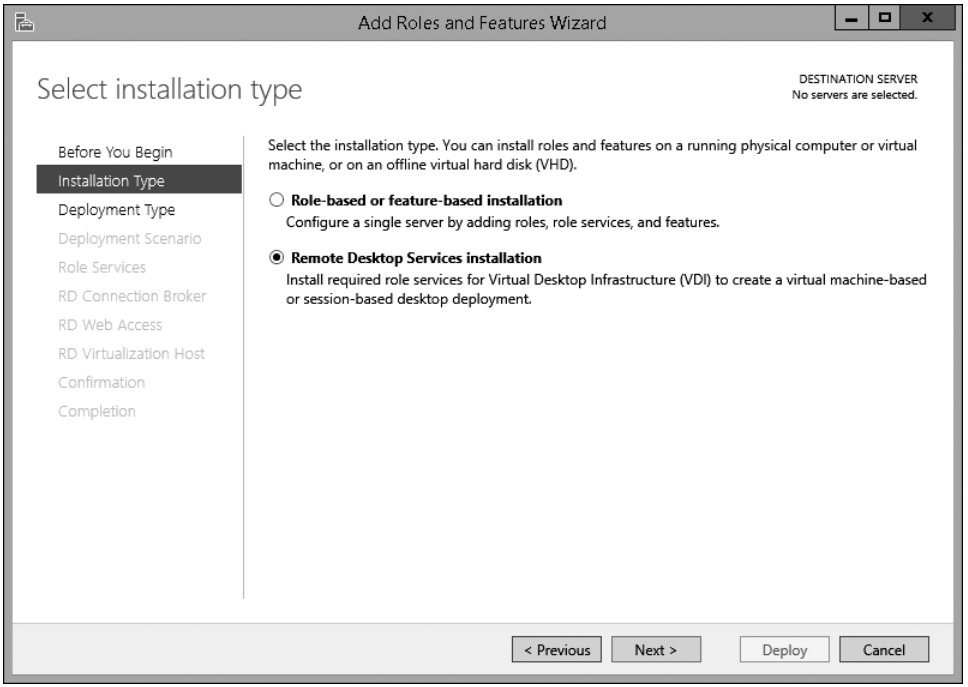

Figure 8-8 Add Roles And Features Wizard, Select Installation Type page

2. **Select the RDS deployment type** On the Select Deployment Type page, shown in Figure 8-9, select the appropriate deployment type. The Quick Start option installs the required role services on a single server and creates a session collection with several sample RemoteApp programs (Calculator, Paint, and WordPad). You only should use the Quick Start option for testing. In most cases, you want to select the Standard Deployment option because this allows you to customize the deployment for your environment.

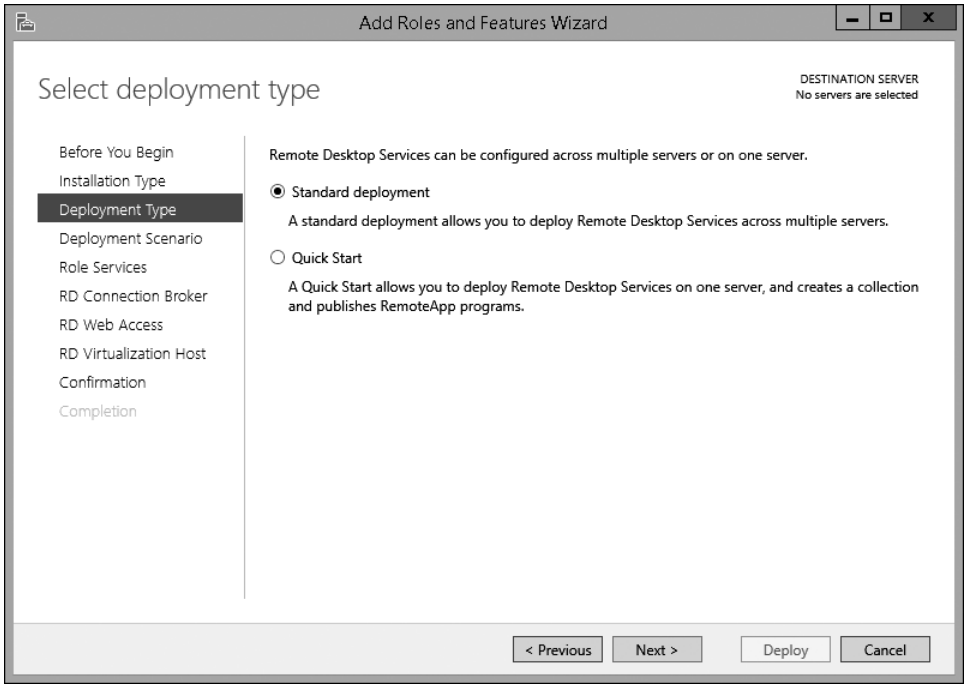

Figure 8-9 Add Roles And Features Wizard, Select Deployment Type page

3. **Select the RD deployment scenario** On the Select Deployment Scenario page, shown in Figure 8-10, select the Virtual Machine–Based Desktop Deployment option or the Session-Based Desktop Deployment option. A VM-based desktop deployment is used to deploy personal and pooled virtual desktops on computers running Hyper-V. A session-based desktop deployment uses RD Session Hosts.

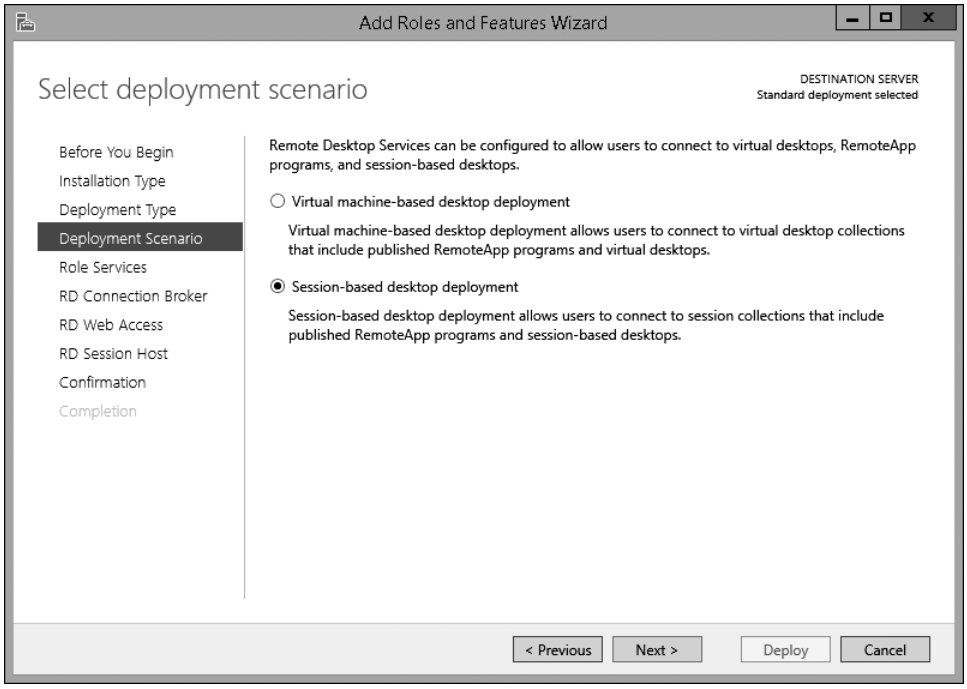

Figure 8-10 Add Roles And Features Wizard, Select Deployment Scenario page

4. **Select servers for RDS role services** In the Add Roles And Features Wizard, select the servers on which you want to install the RD Connection Broker, RD Web Access, and RD Session Host role services. As part of making RDS highly available, you can install each role service on multiple servers. In most RDS deployments, the RD Session Host role service isn't combined with other role services. The RD Connection Broker and RD Web Access role services can be combined in smaller RDS deployments.

During the deployment, the servers on which you installed the RD Session Host role are restarted. After the installation, you can perform initial configuration of the RDS deployment. You also can add servers to the deployment. At minimum, you should add RD Licensing, because you can't connect to an RD Session Host without valid RDS CALs after the initial grace period of 120 days expires. You also should consider installing multiple instances of the RDS role services for high availability.

Inside OUT

Using Windows PowerShell to deploy RDS

You can use Windows PowerShell to deploy RDS on Windows Server 2012 or newer. In Windows Server 2012, use the New-SessionDeployment cmdlet. In Windows Server 2012 R2, use the New-RDSessionDeployment cmdlet.

- **You can use Windows PowerShell to manage all aspects of deploying and managing RDS. For more information about cmdlets available for deploying and managing RDS in Windows Server 2012 R2, see the Remote Desktop Cmdlets in Windows PowerShell page on TechNet at *http://technet.microsoft.com/en-us/library/jj215451.aspx*.**

To install a session-based deployment of RDS, perform the following steps:

1. In Server Manager, click Manage and then click Add Roles And Features.
2. In the Add Roles And Features Wizard, on the Before You Begin page, click Next.
3. On the Select Installation Type page, click Remote Desktop Services Installation and click Next.
4. On the Select Deployment Type page, click Standard Deployment and click Next.
5. On the Select Deployment Scenario page, click Session-Based Desktop Deployment and click Next.
6. On the Review Role Services page, click Next. This page provides a brief description of each role service, but there is nothing to configure. The currently logged-on account is being used to create the deployment and is displayed here as a reminder.
7. On the Specify RD Connection Broker Server page, shown in Figure 8-11, in the Server Pool box, double-click the server on which you wish to install the RD Connection Broker role service and click Next.

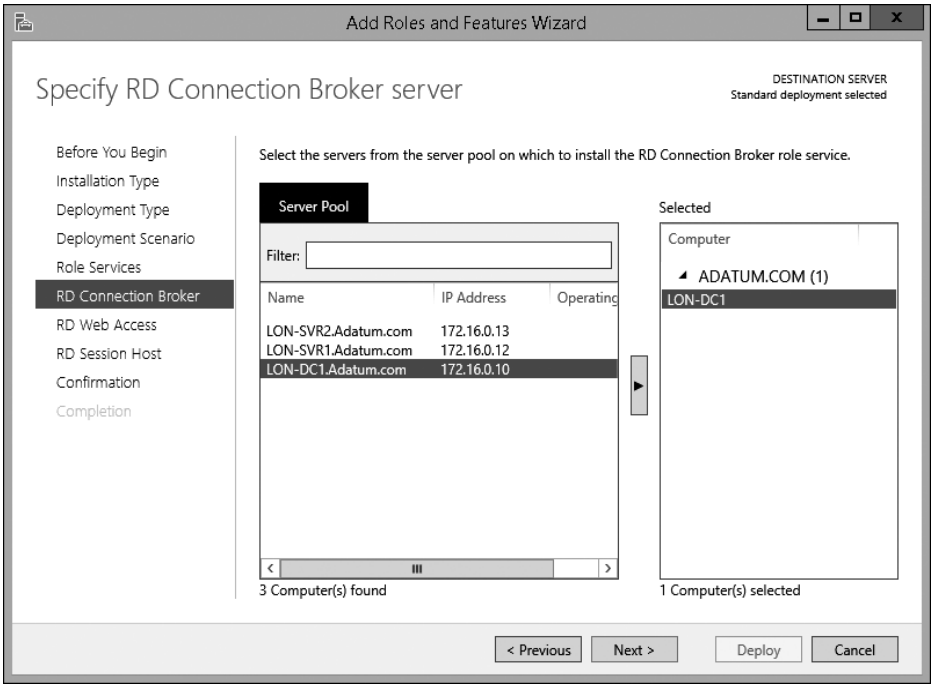

Figure 8-11 Add Roles And Features Wizard, Specify RD Connection Broker Server page

8. On the Specify RD Web Access Server page, shown in Figure 8-12, select the Install The RD Web Access Role Service On The RD Connection Broker Server check box and click Next. Alternatively, you can select another server on which to install the RD Web Access role service.

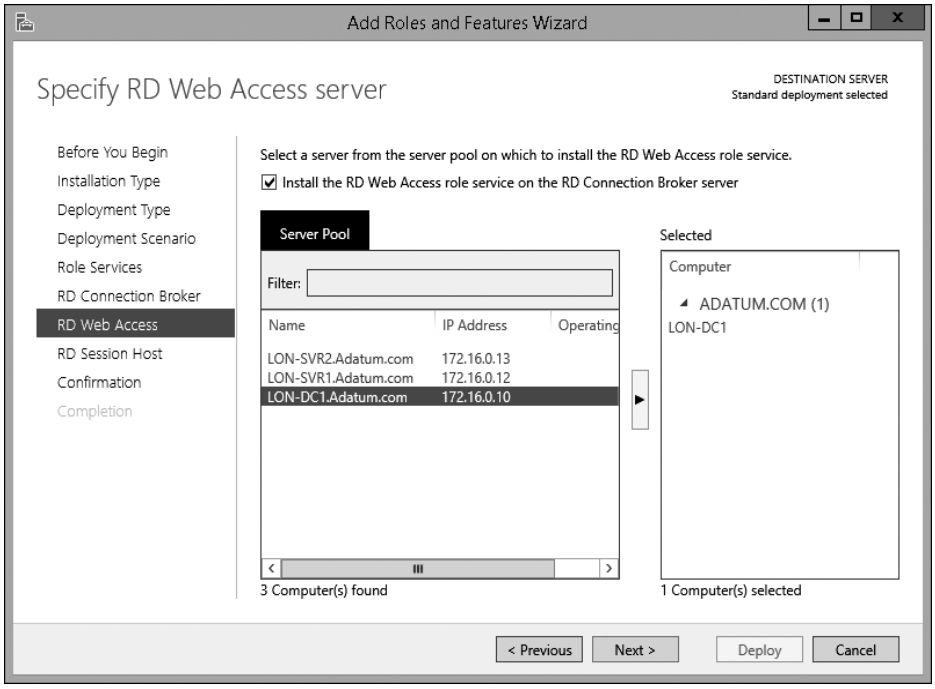

Figure 8-12 Add Roles And Features Wizard, Specify RD Web Access Server page

9. On the Specify RD Session Host Servers page, shown in Figure 8-13, double-click the server on which you wish to install the RD Session Host role service and click Next.

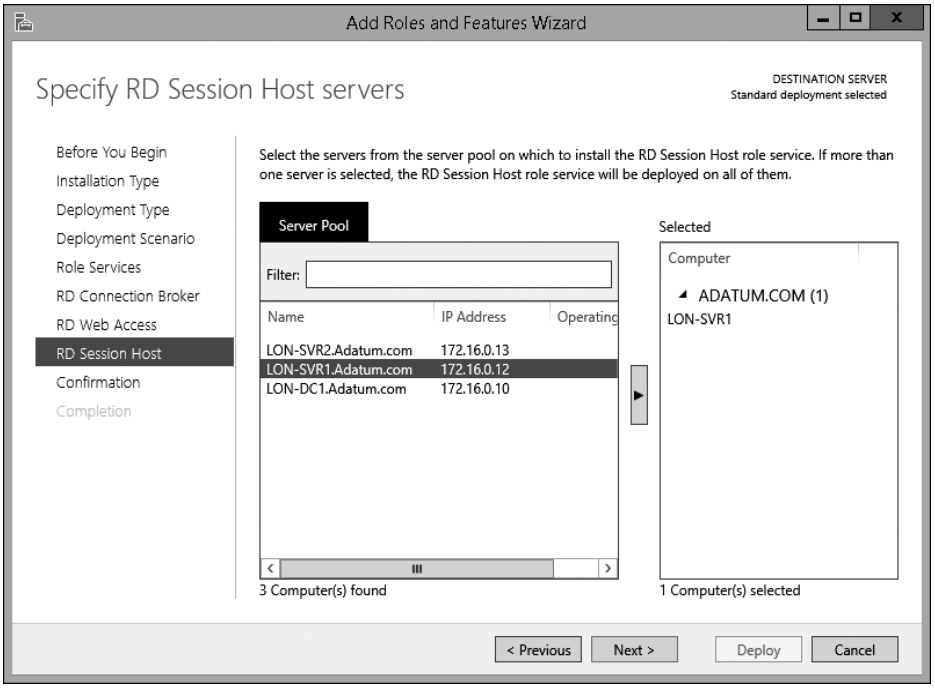

Figure 8-13 Add Roles And Features Wizard, Specify RD Session Host Servers page

10. On the Confirmation page, review the selected servers, select the Restart Destination Server Automatically If Required check box, and click Deploy.

11. On the Completion page, wait for the installation of the RDS role services to complete and click Close. If you are installing roles on the server from which you started the installation, the server may restart and require you to sign in again.

Understanding session collections

Session collections enable you to organize and control user connectivity to RDS. Each session collection contains either RD Session Host servers for session-based virtual desktops or VMs on Hyper-V for pooled or personal virtual desktops.

Collections simplify the administration process by enabling you to manage all collection members as a unit instead of managing them individually. For example, after you configure a collection with session settings, those settings automatically apply to all the servers in the collection. If you add a server to a collection, session settings also automatically apply to the added server.

When you add multiple RD Session Host servers to a collection, connections automatically are load balanced among them. The RD Connection Broker server uses the collection configuration information to identify that there are multiple RD Session Host servers and connects an equal number of clients to each. If an RD Session Host server in a collection fails, the RD Connection Broker connects all users to the remaining RD Session Host servers in the collection.

When there are multiple RD Session Host servers in a collection, they need to be configured with identical applications. Users expect the same applications to be available each time they sign in. If RD Session Host servers have different applications installed, it will appear to users that applications are randomly appearing and disappearing with each connection.

To create a session collection, perform the following steps:

1. In Server Manager, in the navigation pane, click Remote Desktop Services.
2. In Remote Desktop Services > Overview, click Create Session Collections.
3. In the Create Collection Wizard, on the Before You Begin page, click Next.
4. On the Name The Collection page, in the Name box, type the name of the collection and click Next. Make the name something that accurately describes how the collection will be used. You also can type in a more detailed description.
5. On the Specify RD Session Host Servers page, shown in Figure 8-14, double-click the RD Session Host server you want to add to the collection and click Next. Only RD Session Host servers already added to the RDS deployment appear in the Server Pool box. An RD Session Host server can be added to only one collection.

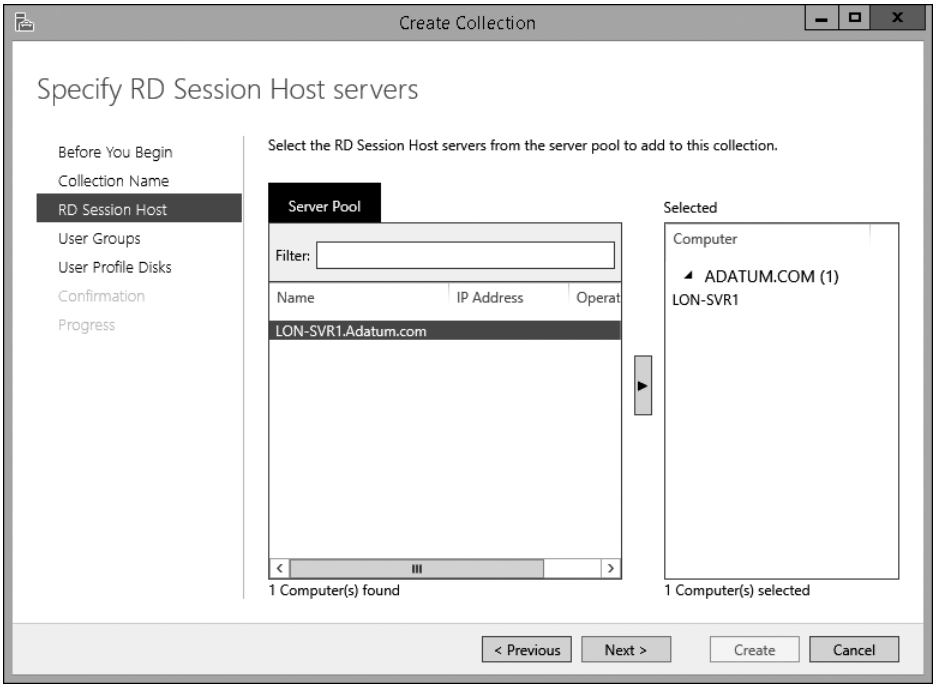

Figure 8-14 Create Collection Wizard, Specify RD Session Host Servers page

6. On the Specify User Groups page, shown in Figure 8-15, remove the Domain Users group, add the groups you want to have access to the collection, and then click Next. The Domain Users group is listed by default and would allow any user in your organization to access the collection. In most cases, you want to restrict collection access to a specific group of users.

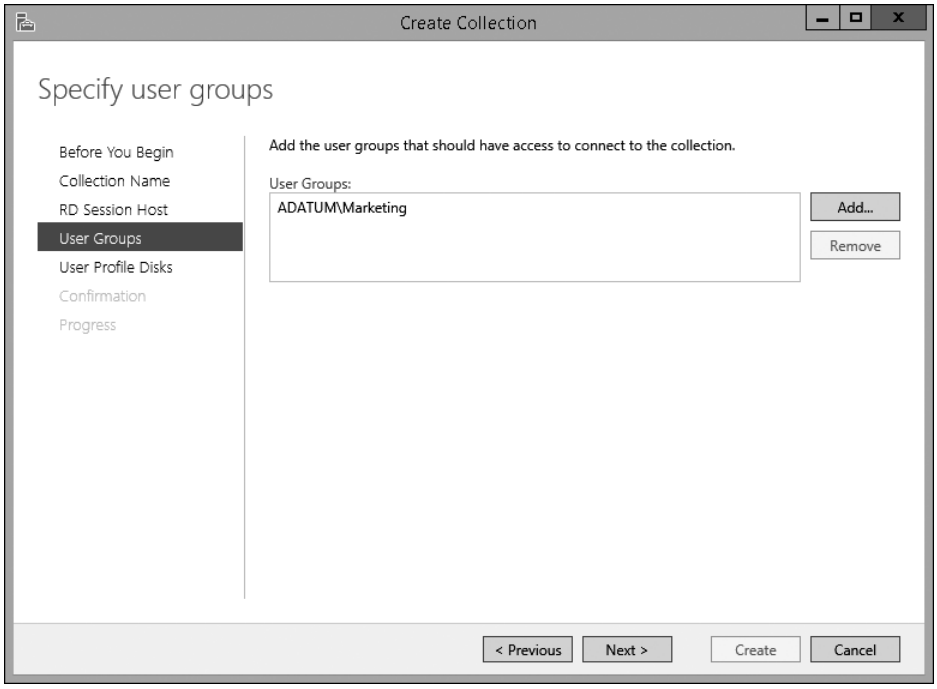

Figure 8-15 Create Collection wizard, Specify User Groups page

7. On the Specify User Profile Disks page, shown in Figure 8-16, select the Enable User Profile Disks check box if you have decided to implement user profile disks for users. If you select this option, you need to enter the UNC path where the user profile disks will be stored in the Location Of User Profile Disks box. You also need to specify a size in the Maximum Size (In GB) box.

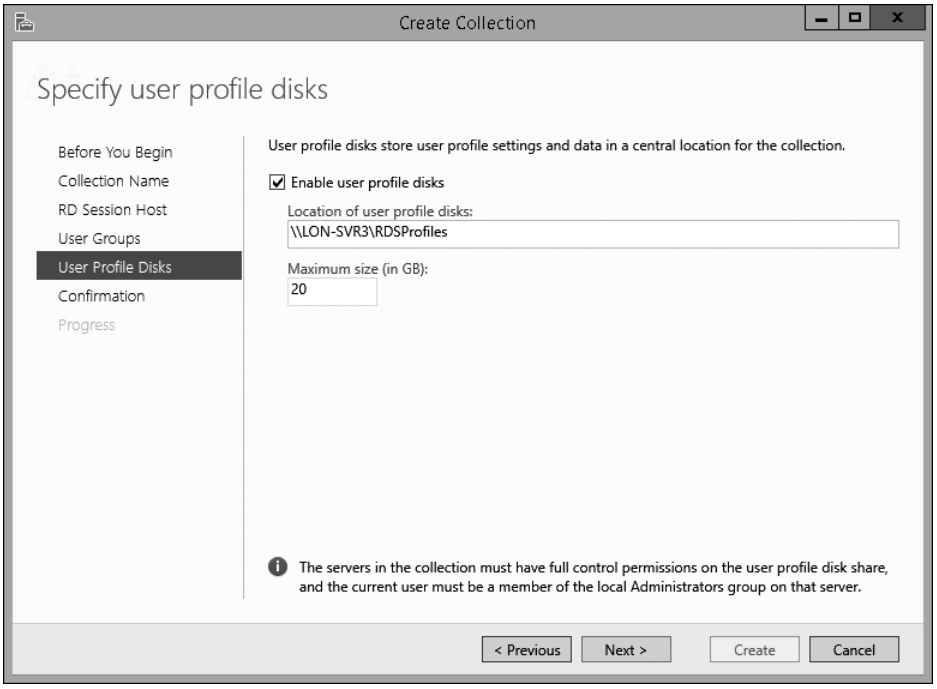

Figure 8-16 Create Collection Wizard, Specify User Profile Disks page

8. On the Confirm Selections page, click Create.
9. On the View Progress page, wait until all tasks are complete and then click Close.

Configuring session collections

The user interface for creating a session collection allows you to configure only a few of the configuration options for a session collection. After the session collection is created, you can edit the session collection and configure many more options.

To edit a session collection, perform the following steps:

1. In Server Manager, in the navigation pane, click Remote Desktop Services.
2. In Remote Desktop Services, in the navigation pane, click the collection you want to edit.
3. While viewing the collection, next to the Properties box, click Tasks and click Edit Properties.
4. In the *CollectionName* Properties window, edit the properties as required and click OK.

When you are editing the properties of a session collection, the editing window is divided into pages with groups of related options. The General page, shown in Figure 8-17, has the Name and Description that you entered during creation. The Show The Session Collection In RD Web Access check box was not available during creation. It is selected by default. Consider disabling this option during scheduled outages when you are performing maintenance on a session collection, for example, when you are upgrading an application on the RD Session Hosts in the collection.

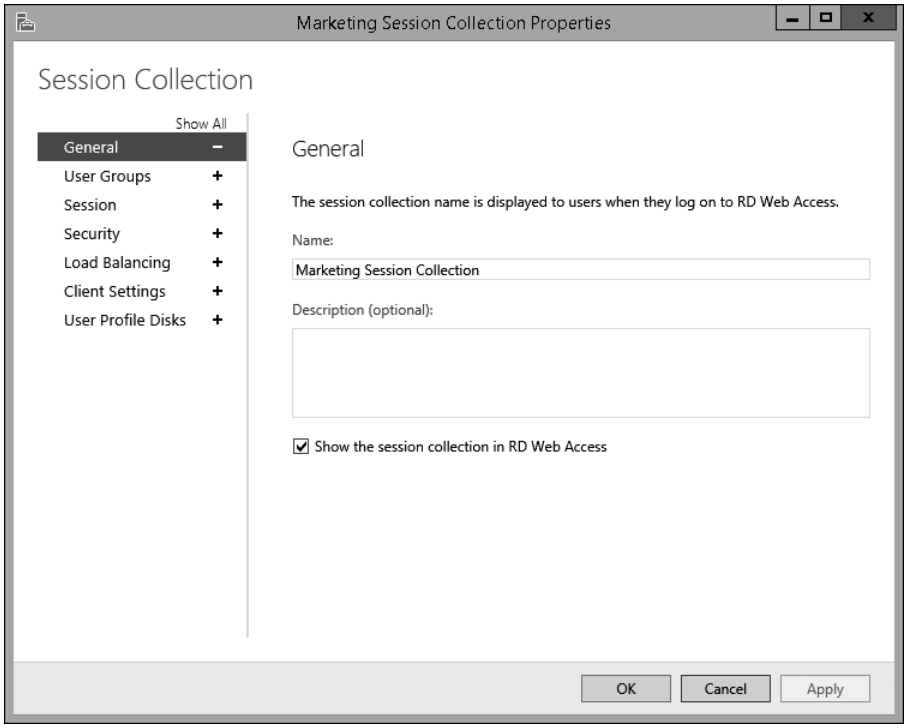

Figure 8-17 Properties of a session collection, General page

The User Groups page in the properties of a session collection allows you to configure which groups of users can connect to the session collection. This is the same as the user groups configured during creation.

The Session page, shown in Figure 8-18, has a number of settings that control session limits and temporary folders.

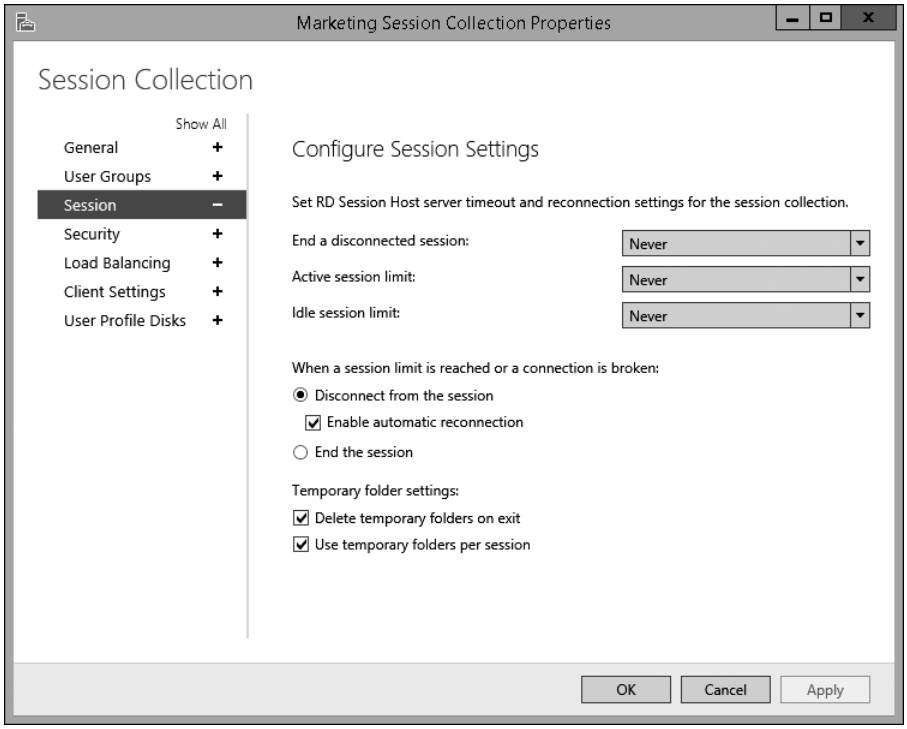

Figure 8-18 Properties of a session collection, Session page

Table 8-3 describes the session settings available on the Session page.

Table 8-3 Session settings for a session collection

Session setting	Description
End A Disconnected Session	Controls when a session is ended after a user disconnects. You can select to never end disconnected sessions or select a value ranging from one minute to five days. The default value is Never. A session is disconnected when there is a network connectivity issue or when a user closes the Remote Desktop Connection window without signing out. A disconnected session keeps all of the applications open and continues to use memory on the RD Session Host. Users can reconnect to the session and resume working where they left off, but if there are too many disconnected sessions, the RD Session Host may not have enough memory.

Active Session Limit	Controls how long an active session can be before it is disconnected or ended. To allow active sessions with no limit, select Never. To limit active sessions, select a time value ranging from one minute to five days. The default value is Never. Users receive a warning two minutes before the active session limit is reached. This provides users with time to save their work. An active session is one in which the user is performing a task. An active session is identified by mouse movement or keyboard input. There is seldom a need to limit active sessions, but you could limit them if you are concerned that unauthorized users are accessing a session. This will force the user to reconnect and provide authentication credentials when the limit is reached.
Idle Session Limit	Controls how long an idle session can be idle before it is disconnected or ended. To allow idle sessions with no limit, select Never. To limit idle sessions, select a time value ranging from one minute to five days. The default value is Never. Users receive a warning two minutes before the idle session limit is reached. This provides users with an opportunity to move the mouse or press a key to make the session active and avoid the idle session limit. An idle session is one in which the user isn't performing a task. An idle session is identified by a lack of mouse movement or keyboard input. Most organizations configure an idle session limit. This has a similar effect on security as having the screen lock on a desktop computer. If a session is connected but unused, it may mean that the user has left his or her connection unattended.
When A Session Limit Is Reached Or A Connection Is Broken	Controls the action that is taken when the active session limit is reached, the idle session limit is reached, or a network problem disconnects a client. You can choose Disconnect From The Session or End The Session. In most cases, you will select Disconnect From The Session to prevent users from losing their work when they are disconnected. When you select this option, you also can select Enable Automatic Reconnection. This allows the RDC client to reconnect automatically after short network interruptions. If you do not select this option, the users must provide authentication credentials to reconnect to their disconnected session. By default, Disconnect From The Session and Enable Automatic Reconnection are selected.
Delete Temporary Folders On Exit	Configures temporary folders to be deleted when a session ends. This ensures that temporary files do not consume unnecessary disk space. This option is enabled by default.

Use Temporary Folders Per Session	Configures each session for a user to have separate temporary folders on RD Session Host servers where a single user account is allowed to have multiple simultaneous sessions. This option is enabled by default to ensure that multiple sessions on an RD Session Host server do not conflict. However, it isn't relevant in most deployments because users typically are limited to a single session.

The Configure Security Settings page, shown in Figure 8-19, allows you to configure the Security Layer and the Encryption Level to use for the session. The Security Layer defines encryption methods that are used to encrypt communication between the RDC client and the RD Session Host. The available options for security layer are as follows:

- **RDP Security Layer** This is the weakest option for the security layer. It is available to support older RDP clients. This security layer does not support the use of Network Level Authentication.
- **SSL (TLS 1.0)** This is the strongest security layer. This security layer supports the use of network-level authentication. When this security layer is used, a certificate on the RD Session Host is used to establish the encryption channel. If the name on the certificate does not match the name used when connecting to the RD Session Host, then a warning is displayed on the client. This is supported by Windows XP SP3 and newer operating systems.
- **Negotiate** This is the default selection for security layer. SSL (TLS 1.0) is used if available on the server and client. If SSL (TLS 1.0) can't be used, then RDP Security Layer is used.

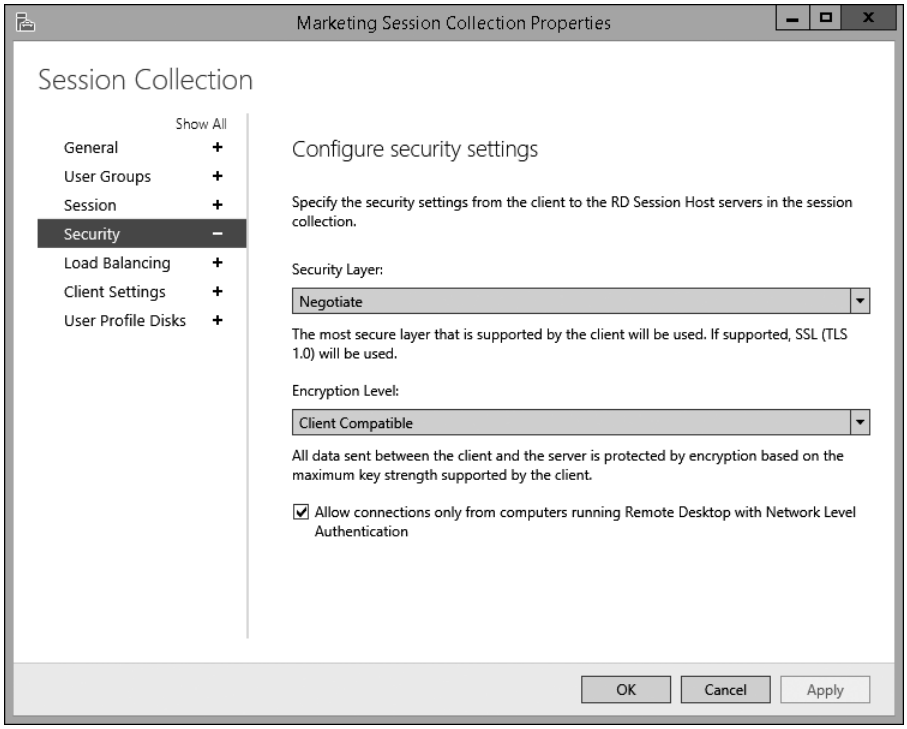

Figure 8-19 Properties of a session collection, Configure Security Settings page

Network Level Authentication is an authentication method that requires clients to enter authentication credentials before they are connected to the RD Session Host server. The credentials are passed by the RDC client to the RD Session Host server, and if the credentials are valid, the sign-in process is performed. When Network Level Authentication isn't used, clients can connect to the RD Session Host server and interact with the sign-in screen on the RD Session Host before they are authenticated. This is a security risk because it is possible for unauthenticated clients that have access to RD Session Host servers to see recently used user names and the operating system version.

You can force all clients to use Network Level Authentication by selecting the Allow Connections Only From Computers Running Remote Desktop With Network Level Authentication check box. This is enabled by default.

The Encryption Level setting allows you to configure the number of bits used for encryption. This setting applies for both security layers, and more bits provide stronger encryption. The options for Encryption Level are as follows:

- **Low** Uses 56-bit encryption for data sent from the client to the server. Data sent from the server to the client isn't encrypted. This option is provided to support older clients and typically isn't required.
- **High** Uses 128-bit encryption for all data sent between the client and server. This option can be used by Windows XP and newer operating systems. This is the preferred option.
- **FIPS Compliant** Uses encryption algorithms that are FIPS 140-1 or FIPS 140-2 compliant for all data sent between the client and server. Federal Information Processing Standards (FIPS) is a United States government standard for data encryption. This option typically isn't used unless requested specifically by an organization that needs to meet FIPS requirements.
- **Client Compatible** Negotiates the highest level of encryption supported by the client and uses that. This is the default configuration, but it can be considered a security risk because it allows 56-bit encryption for clients that request it. Use this option only if you need to support clients that can't use 128-bit encryption.

➤ **The Configure Load Balancing Settings page in the properties of a session collection is covered later in this chapter in the section titled "High availability for RD Session Host servers."**

The Configure Client Settings page has settings for device redirection and monitors. By default, redirection is enabled for all available options. If desired, you can select to enable or disable redirection for the following:

- Audio And Video Playback
- Audio Recording
- Smart Cards
- Plug And Play Devices
- Drives
- Clipboard
- Printers

➤ **You can find more information about the client settings in Chapter 9.**

The User Profile Disks page, shown in Figure 8-20, allows you to configure all of the information entered during collection creation and to define what data is stored on the user profile disks. There are two options for user profile disks data settings:

- **Store All User Settings And Data On The User Profile Disk** Specifies that the complete user profile is stored on the user profile disk. You can add specific folders and files within the profile to exclude.
- **Store Only The Following Folders On The User Profile Disk** Specifies that only selected folders in the user profile are stored on the user profile disk. The folders available for selection are Contacts, Desktop, Documents, Downloads, Links, Music, Pictures, Roaming User Profile Data, and User Registry Data. You also can add specific files and folders within the profile to include.

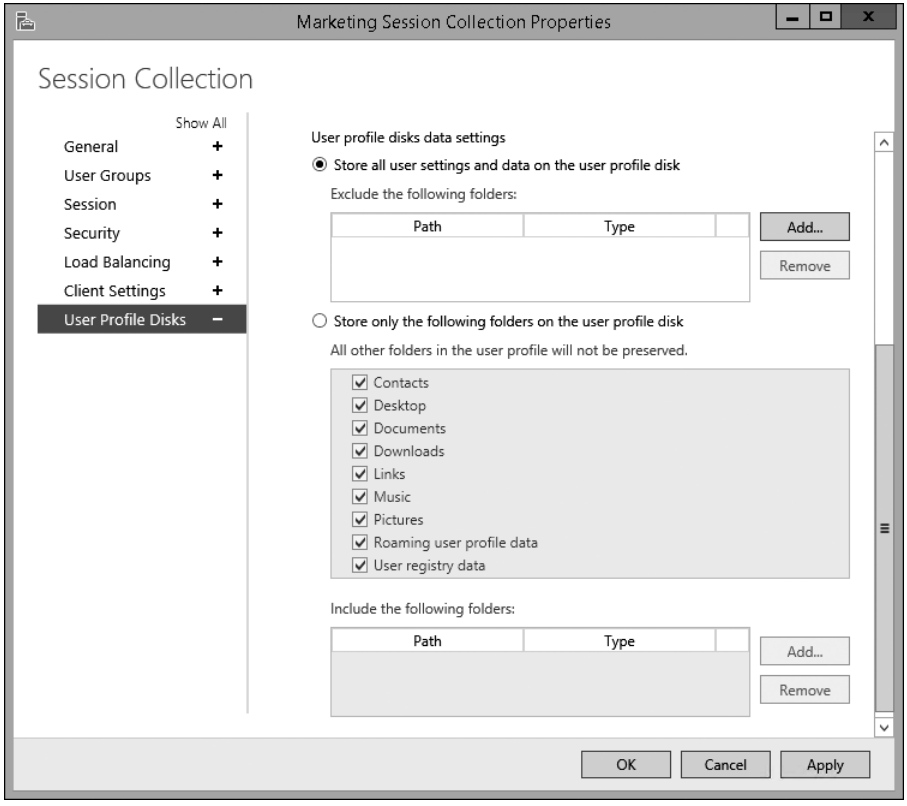

Figure 8-20 Properties of a session collection, User Profile Disks page

Configuring RD Licensing servers

The initial configuration of RDS doesn't configure licensing. However, a functional RDS deployment in production must have licensing properly configured to ensure that users can connect. To configure licensing for RDS, you need to complete the following tasks:

- Set the licensing mode
- Install an RD Licensing server
- Activate an RD Licensing server
- Install and activate CALs

To install an RD Licensing server, perform the following steps:

1. In Server Manager, in the navigation pane, click Remote Desktop Services.
2. On the Overview page, in the Deployment Overview area, click RD Licensing.
3. In the Add RD Licensing Servers Wizard, on the Select A Server page, double-click the server you want to configure as an RD Licensing server and click Next.
4. On the Confirmation page, click Add.
5. Wait until the installation is complete and click Close.

To set the licensing mode for an RDS deployment, perform the following steps:

1. In Server Manager, in the navigation pane, click Remote Desktop Services.
2. On the Overview page, in the Deployment Overview area, click Tasks and click Edit Deployment Properties.
3. In the Deployment Properties window, in the navigation pane, click RD Licensing.
4. On the RD Licensing page, select Per Device or Per User and click OK.

The Microsoft Clearinghouse is the service that is used to activate RD Licensing servers and RDS CALs. When you install an RD Licensing server, you need to activate it before it can be begin servicing clients. To do this, you use Remote Desktop Licensing Manager (RD Licensing Manager), shown in Figure 8-21. RD Licensing Manager is installed on each RD Licensing server.

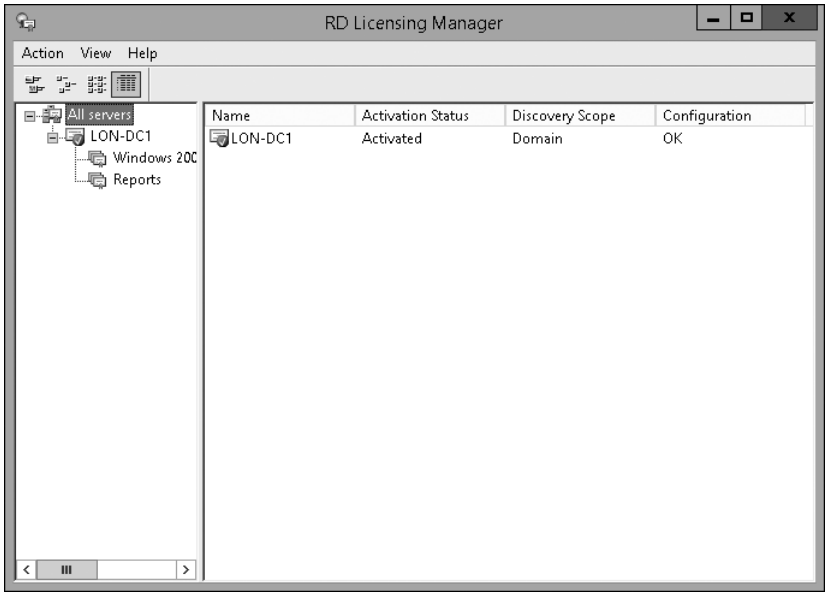

Figure 8-21 RD Licensing Manager

When you activate an RD Licensing server, you need to exchange information with the Microsoft Clearinghouse. You can choose from the following connection methods:

- **Automatic Connection (Recommended)** Transfers the necessary information between the RD Licensing server and the Microsoft Clearinghouse over the Internet. The RD Licensing server must have connectivity to the Internet.
- **Web Browser** Requires you to enter a Product ID at the website https://activate.microsoft.com. Then, you type the license server ID provided by the website into the Activate Server Wizard. Use this connection method if the RD Licensing server does not have access to the Internet.
- **Telephone** Requires you to phone the Microsoft Clearinghouse and provide the Product ID for your server. You are then give a license server ID, which you need to enter into the Activate Server Wizard. Use this connection method if you have no access to the Internet.

Installing RDS CALs is a similar process to activating an RD Licensing server. The same connectivity methods to the Microsoft Clearinghouse are supported. The installation process automatically uses the method that you used when activating the server. You can change the connectivity method in the Properties of the server if required.

To activate an RD Licensing server over the Internet, perform the following steps:

1. In Server Manager, click Tools, point to Terminal Services, and click Remote Desktop Licensing Manager.
2. Right-click the licensing server and click Activate Server.
3. In the Activate Server Wizard, on the Welcome To The Activate Server Wizard page, click Next.
4. On the Connection Method page, in the Connection Method box, select Automatic Connection (Recommended) and click Next.
5. If you have not already configured the company information for your server, you are prompted to do so. On the Company Information page, enter the required company information and click Next.
6. On the next Company Information page, if desired, enter the optional information and click Next.
7. On the Completing The Activate Server Wizard page, deselect the Start Install Licenses Wizard Now check box and click Finish.
8. Right-click the server and click Review Configuration.
9. In the Server Configuration window, shown in Figure 8-22, click Add To Group.

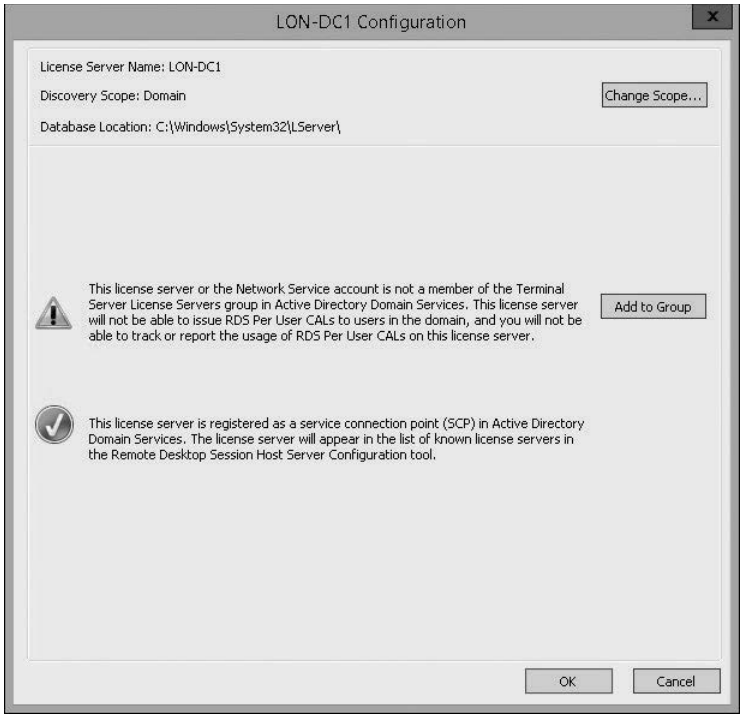

Figure 8-22 License server configuration

10. In the RD Licensing Manager window, click Continue to acknowledge the warning about requiring Domain Admin privileges.

11. In the RD Licensing Manager dialog box, click OK to acknowledge that the server has been added to the Terminal Server License Servers Group.

12. In the Server Configuration window, click OK.

To install RDS CALs over the Internet, perform the following steps:

1. In Server Manager, click Tools, point to Terminal Services, and click Remote Desktop Licensing Manager.

2. Right-click the license server and click Install Licenses.

3. In the Install Licenses Wizard, on the Welcome To The Install Licenses Wizard page, click Next.

4. On the License Program page, select the license program used to purchase your RDS CALs and click Next. Available license programs include Open License, Enterprise Agreement, Campus Agreement, and more.

5. Enter the information requested and click Next. The information requested varies depending on the licensing program used, but it will include either a license code or an agreement number.

6. On the Product Version And License Type page, enter the product version, license type, and number of RDS CALs based on your license and click Next.

7. Wait while the Microsoft Clearinghouse processes the request and the RDS CALs are installed and then click Finish.

Understanding high availability for RDS

A highly available service is one that is available almost all of the time. High availability often is expressed numerically as the percentage of time that a service is available. For example, a requirement for 99.9 percent availability allows 8.75 hours of downtime per year, or approximately 40 minutes of downtime every four weeks. With 99.999 percent uptime, the allowed service downtime is reduced to only 5 minutes per year.

To achieve high availability for a service such as RDS, you need to identify single points of failure in the infrastructure and work to eliminate them. For example, if you have only one RD Session Host server, that is a single point of failure. There will be a service outage if that server has hardware problems or is taken offline for maintenance.

To make infrastructure highly available, you need to make it redundant. For example, within a server, using mirroring (RAID 1) for disks ensures that that failure of a single hard disk does not cause the server to fail. This principle also can be applied at other levels of infrastructure such as networking, network services, and data center power. To make RDS highly available, you need multiple servers for each of the RDS role services, but not all RDS role services automatically become highly available just because you add more servers running the role service. Some RDS role services require you to implement load balancing.

Inside OUT

Redundant hardware vs. redundant servers

Within a server, two common hardware components for high availability are dual power supplies and hardware RAID cards. Combined, these options can add $1,000 to the cost of a server.

When you have highly available infrastructure with multiple servers, you can balance the high availability of components within a server with the high availability provided by having multiple servers. Instead of providing hardware redundancy in servers, you can provide the redundancy with additional servers. This can be a cost-effective way to provide high availability.

In the previous example, you could save $1,000 per server by not putting in dual power supplies and hardware RAID cards. In a large RDS deployment, if you are implementing eight RD Session Host servers, this would save $8,000 in hardware costs, which you could use to purchase two or three additional servers to provide high availability.

Most administrators want their individual servers to be highly available, but it's really the services that need to be highly available. The key to making this strategy work is providing enough redundant servers and monitoring so that you can resolve issues quickly when a server fails

Understanding load balancing

Load balancing is a technology that you can use to achieve high availability and scalability primarily for stateless services. The term *stateless* refers to workloads that respond to each request independently from previous requests and without keeping client state. For example, when a client requests a webpage, a Web server gathers all of the necessary information from the request and then returns a generated webpage to the client. When the client requests another webpage, it might request the webpage from the same Web server or from any other identically configured Web server in a load-balancing cluster.

The servers that are part of a load balancing cluster are referred to as nodes. Each node is configured with the same software so that clients can connect with any node and obtain the intended service. For example, each node would have the same website configured.

Clients connect to a virtual IP address that is used to access all nodes in the cluster. To the clients, the virtual IP address behaves the same way that an IP address configured on a physical server would. Only one cluster node responds to each client request.

If a node in the cluster fails, the remaining nodes continue to service clients. This makes a load balancing cluster highly available. Adding nodes to the cluster increases the capacity of the cluster. Adding nodes is scaling out.

Windows Network Load Balancing

Windows Network Load Balancing (NLB) is a software solution for load balancing that is included in Windows Server operating systems. NLB creates a virtual IP address that all of the

nodes in the load balancing cluster share. When a request comes in to virtual IP, the request is received by all nodes, but only one node responds. The nodes determine the appropriate node to respond based on an algorithm that they all use.

The most common reason organizations consider using NLB is the cost. Because it is included in Windows Server operating systems, it is effectively free. However, there are a few drawbacks to using NLB:

- **It is not service aware** NLB is capable of identifying when a server is no longer responding but not when a service is no longer responding. This means that some types of failures result in clients being directed to a nonresponsive service on a partially functional node.
- **Scalability is limited** NLB supports up to 32 nodes in a cluster, but performance peaks at 8 nodes.
- **Network hardware configuration may be required** Some network switches need additional configuration to work with NLB. This is required because multiple devices are sharing the same virtual IP address but are connected to different switch ports.

Hardware-based load balancing

Most large organizations use specialized hardware load balancers instead of using NLB. Hardware load balancers are more scalable than NLB, but they also are significantly more expensive. The least expensive hardware load balancers are about $2,000, and they can cost more than $40,000.

The configuration of a hardware load balancer varies depending on the vendor, but all of them provide the same basic functionality. The virtual IP address for the load balancing cluster is assigned to the load balancer, and the load balancer receives requests from the clients. The load balancer then forwards each request to a single node.

The load balancer is responsible for identifying failed nodes. Node failure can be identified at the node level, as NLB does, or at the service level. If service-level failure is used, the load balancer monitors the service on each node and stops sending client requests if the service stops responding.

Inside OUT

DNS round robin

An alternate method for load balancing is DNS round robin. This method is implemented entirely by using DNS records and does not require any additional software or hardware. The main concern with DNS round robin is that it can be less reliable than NLB or hardware-based load balancing.

In DNS, a host (A) record is used to identify the IP address to which a name should resolve. For example, a host (A) record identifies the IP address to which www.microsoft.com resolves. If you create multiple host (A) records for a name, then it resolves to multiple IP addresses. When a client resolves the name, the DNS server provides IP addresses from all of the host (A) records for that name. This configuration is called DNS round robin.

When a client receives multiple IP addresses from a DNS server, the typical behavior is to contact the first IP address in the list. If the first IP address in the list doesn't respond, then the client contacts the second IP address in the list. This process continues until the client successfully connects to an IP address or the list is exhausted.

The main drawback to DNS round robin is that it's unpredictable. The client is in control of the process for managing server failure and the list of IP addresses. If the software developer for the client software does not manage server failure well, then users experience poor performance. For client software that is designed to use DNS round robin, it is a simple and effective load balancing mechanism.

High availability for RD Session Host servers

When an RDS deployment has a single RD Session Host server, that server becomes a single point of failure. When it fails, the failure will affect all users who are connected to it and who run RemoteApp programs on that server. You must consider the possibility of failure or the lack of RD Session Host server availability in your disaster recovery plan.

You can take several steps to improve RD Session Host availability. You can use reliable and redundant hardware from respected vendors to minimize the probability of hardware failure. You also should make sure that the network is reliable and that there are multiple network paths to an RD Session Host server. You should be aware, however, that failures are unavoidable, and no single server can always be available without downtime. For example, after you install Windows updates, computer restart is often required, which causes server downtime.

To make the RD Session Host server role highly available, you should have multiple RD Session Host servers in each collection. The RD Connection Broker role service automatically load balances connections to the RD Session Host servers. If the RD Connection Broker role service identifies that an RD Session Host server is unavailable, clients are not directed to the failed RD Session Host server. Clients are directed only to the remaining functional RD Session Host server.

As a best practice, all RD Session Host servers should have a similar hardware configuration. This ensures that all RD Session Host servers have similar performance and can handle a similar

number of clients. The default configuration of load balancing for a collection, shown in Figure 8-23, is best suited for this scenario.

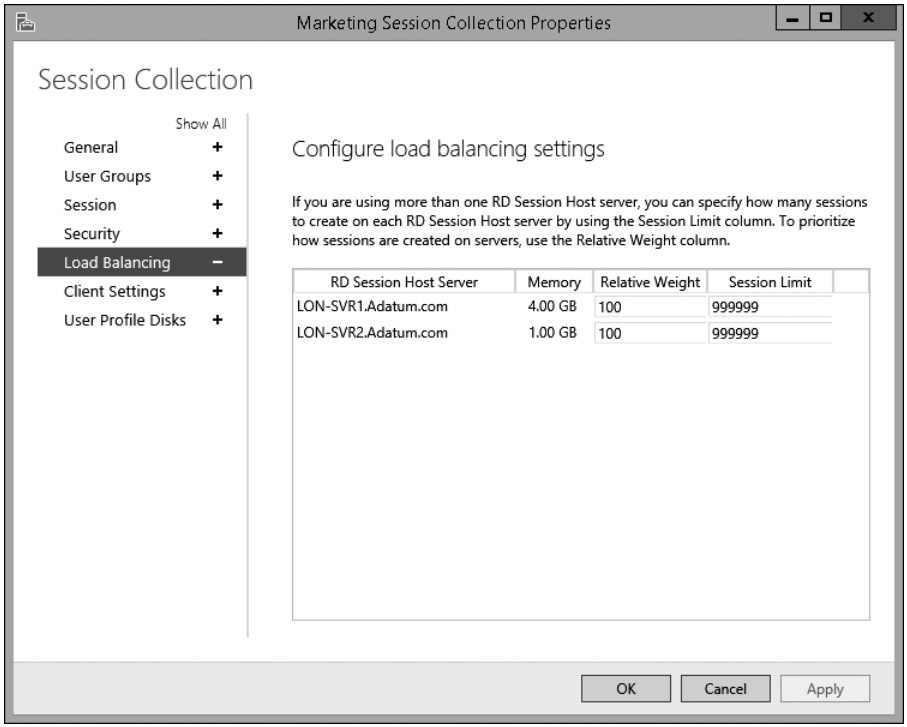

Figure 8-23 Session Collection Properties page, Configure Load Balancing Settings page

You can adjust the ratio of sessions allocated to an RD Session Host server by adjusting the Relative Weight value for that server. The default value for all servers is 100. When all servers have the same value, they all receive the same number of sessions. If you have one server with significantly better hardware and give that server a Relative Weight of 200, then it will receive twice as many sessions as a server with a Relative Weight of 100.

You also can set a Session Limit for each server. The default value for the Session Limit is 999,999, which effectively is unlimited. If you have determined that users experience performance issues when more than 80 clients are connected, then you can set a session limit of 80 to ensure that performance is satisfactory for all users.

To add a second RD Session Broker server to an RDS deployment, perform the following steps:

1. In Server Manager, in the navigation pane, click Remote Desktop Services.

2. On the Overview page, click Add RD Session Host Servers.
3. In the Add RD Session Host Server window, on the Select A Server page, in the Server Pool box, double-click the server you want to configure as an RD Session Host server and click Next.
4. On the Confirm Selections page, select the Restart Remote Computers As Needed check box and click Add.
5. Wait until the RD Session Host role service is installed on the server and click Close.

To add an RD Session Host server to a session collection, perform the following steps:

1. In Server Manager, in the navigation pane, click Remote Desktop Services.
2. In the navigation pane, click the session collection.
3. Scroll down to the Host Servers box, click Tasks, and click Add RD Session Host Servers.
4. In the Add Servers To Collection Wizard, on the Specify RD Session Host Servers page, double-click the RD Session Host server that you want to add to the session collection and click Next.
5. On the Confirm Selections page, click Add.
6. Wait until the task is complete and then click Close.

High availability for the RD Connection Broker role service

The RD Connection Broker role service is responsible for directing clients to an available RD Session Host server. If the RD Connection Broker role service is unavailable, then users are not able to access session-based virtual desktops. Having a single RD Connection Broker server creates a single point of failure.

To make the RD Connection Broker role service highly available, you need to have multiple RD Connection Broker servers. The RD Connection Broker role service uses a SQL Server database to track sessions that have been allocated to RD Session Host servers. For multiple RD Connection Brokers servers to work together, they need to share a single SQL Server database.

> NOTE
>
> **High availability for the RD Connection Broker role service in Windows Server 2012 and newer is active/active. This means that multiple RD Connection Broker servers can respond to client requests at the same time. Older implementations of the RD Connection Broker and Terminal Services (TS) Connection Broker high availability**

were active/passive. A second server was used only when the first server failed. This provided high availability but not scalability.

To prepare the RD Connection Broker role service for high availability, you need to do the following:

- Configure a server running Microsoft SQL Server 2008 R2 or newer. The RD Connection Broker servers must have permission to create a database on the server.
- Install the SQL Server Native Client on all RD Connection Broker servers. The RD Connection Broker servers use this to connect to the SQL database.
- Configure a static IP address on all RD Connection Broker servers. This is required to implement DNS round robin for load balancing.
- Configure a DNS round robin record for the RD Connection Broker servers. Select a name that is meaningful, such as rds.adatum.com.

Inside OUT

Configuring SQL permissions for RD Session Broker high availability

To assign the necessary permissions for RD Session Broker servers on the SQL server, perform the following steps:

1. **Create a security group in Active Directory Domain Services (AD DS) and add the computer accounts for the RD Connection Broker servers.**
2. **Restart the RD Connection Broker server so that the new group membership takes effect.**
3. **On the SQL server, open SQL Server Management Studio.**
4. **In the Connect To Server window, verify that the correct instance of SQL server is listed and click Connect.**
5. **In SQL Server Management Studio, in Object Explorer, expand Security, and expand Logins.**
6. **Right-click Logins and click New Login.**
7. **In the Login – New window, on the General page, in the Login Name box, type Domain/GroupName.**
8. **Click the Server Roles page, select the Dbcreator check box, and click OK.**
9. **Close SQL Server Management Studio.**

When you configure the RD Connection Broker role service for high availability, its database moves from a local WID to a computer that is running SQL Server. Even when an RDS deployment has multiple RD Connection Broker servers, SQL Server still can be a single point of failure. You should make sure that SQL Server is highly available by running it in a failover cluster.

When you configure high availability for the RD Connection Broker role service, you need to provide a Database Connection String that the RD Connection Broker servers use to connect to the SQL server. The Database Connection String has the following format:

```
DRIVER=SQL Server Native Client 11.0;SERVER=LON-SQL.Adatum.com;Trusted_
Connection=Yes;APP=Remote Desktop Services Connection Broker;Database=RDS-DB
```

There are several things to note about the Database Connection String:

- **A SQL native client version is specified** In this example, the SQL native client version is 11.0. This is used when your SQL server is SQL Server 2012. If your SQL server is SQL Server 2008 R2, then the SQL native client version is 10.0.
- **A server name is specified** In this example, the server name is LON-SQL.Adatum.com. In your deployment, you should specify the name of the SQL server that will be hosting the database for the RD Connection Broker servers.
- **A database name is specified** In this example, the database name is RDS-DB. This is the name of the database that will be created for the RD Connection Broker servers to use. You can select an alternate name, but it should be a meaningful name to make it easy to identify.

To configure the RD Connection Broker role service for high availability, perform the following steps:

1. In Server Manager, in the navigation pane, click Remote Desktop Services.
2. On the Overview page, in the Deployment Overview area, right-click RD Connection Broker and click Configure High Availability.
3. In the Configure RD Connection Broker For High Availability Wizard, on the Before You Begin page, click Next.
4. On the Configure RD Connection Broker For High Availability page, shown in Figure 8-24, in the Database Connection String box, type the appropriate Database Connection String for your environment.

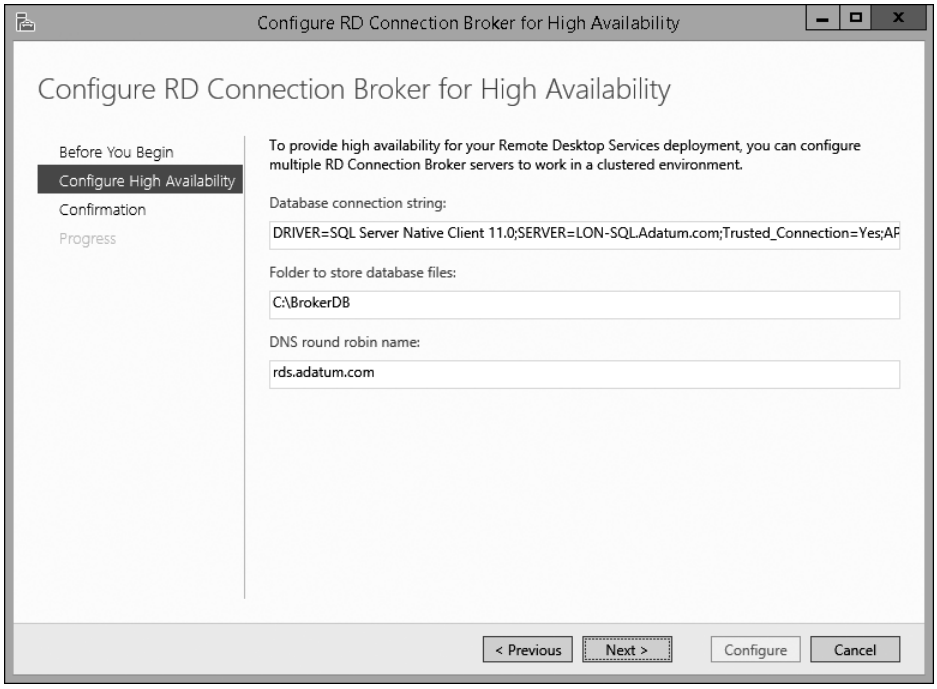

Figure 8-24 Configure RD Connection Broker For High Availability Wizard, Configure RD Connection Broker For High Availability page

5. In the Folder To Store Database Files box, type the path for the database on the SQL server. The database will be created in this location. This folder must already exist.

6. In the DNS Round Robin Name box, type the name of the DNS round robin record that you created for the RD Connection Broker servers and then click Next.

7. On the Confirmation page, click Configure.

8. On the Progress page, click Close.

After you have configured high availability for the RD Connection Broker role service, the RD Connection Broker icon in the Deployment Overview area is updated with the text (High Availability Mode). Now you can add another RD Connection Broker server by right-clicking the RD Connection Broker icon and clicking Add RD Connection Broker server. The new RD Connection Broker server will use the central SQL database that you have configured.

High availability for the RD Web Access role service

RD Web Access servers are a critical part of an RDS deployment. The RD Web Access servers are responsible for providing clients with an .rdp file that contains connectivity information to collections. If RD Web Access isn't available, then clients can't obtain the necessary configuration information to connect to session-based virtual desktops. You should make RD Web Access servers highly available.

Load balancing is used to make RD Web Access servers highly available. You can use NLB, hardware-based load balancing, or DNS round robin. If you are using NLB or hardware-based load balancing, you'll need to create a DNS record for the virtual IP address used by the load balancing cluster. For example, you could create a host (A) record for RDWeb.adatum.com that resolves to the virtual IP address. If you are using DNS round robin, then you need to create multiple host (A) records for RDWeb.adatum.com that resolve to the IP addresses of the RD Web Access servers.

To add an RD Web Access server, perform the following steps:

1. In Server Manager, in the navigation pane, click Remote Desktop Services.
2. On the Overview page, right-click RD Web Access and click Add RD Web Access Servers.
3. In the Add RD Web Access Servers Wizard, on the Select A Server page, double-click the server you want to configure as an RD Web Access server and click Next.
4. On the Confirmation page, click Add.
5. Wait until the installation is complete and click Close.
6. Configure your load-balancing solution with the IP address of the new RD Web Access server.

High availability for the RD Licensing role service

The effect of the RD Licensing role service for an RDS deployment varies, depending on the licensing mode that has been selected. When an RDS deployment is configured for Per User licensing, the RD Session Host servers contact an RD Licensing server each time a client connects. If an RD Licensing server isn't available, then users can't connect.

Per User licensing isn't enforced by RD Licensing servers. If an RD Session Host server can contact an RD Licensing server, that is sufficient to allow a connection. You are responsible for ensuring that you are in compliance with licensing requirements, but they are not enforced.

To make an RDS deployment with Per User licensing highly available, you need to install multiple RD Licensing servers. If the first RD Licensing server is unavailable, then the second is contacted.

Allocation of RDS User CALs among the RD Licensing servers does not matter because they are not enforced. To simplify license management, you can install and activate all RDS User CALs on a single RD Licensing server.

High availability for an RDS deployment configured for Per Device licensing also requires multiple RD Licensing servers, but configuration is more complex because RDS Device CAL usage is enforced. If an RDS Device CAL isn't available, then connectivity can be blocked. Because of this, you need to consider how CALs are allocated among the RD Licensing servers.

RDS client behavior for Per Device licensing varies, depending on the state of the client:

- **First connection** The first time a device connects, it is issued a temporary CAL that can be used only once. If an RD Licensing server is unavailable, the temporary CAL can't be issued, and new devices are unable to connect. The temporary CAL can be issued by any RD Licensing server even if no RDS Device CALs are available on that server.

- **Temporary license** The second time a device connects, it is issued a permanent RDS Device CAL. For a device to be issued a permanent RDS Device CAL, an RD Licensing server with unallocated Per Device CALs must be available. If an RD Licensing server with unallocated Per Device CALs isn't available, then the temporary CAL remains valid for 90 days.

- **Permanent CAL** Devices with a permanent CAL can connect to an RD Session Host when no RD Licensing server is available. Permanent RDS Device CALs are valid for 52 to 89 days and can't be renewed if no RD Licensing server is available.

- **Permanent CAL expired** If the permanent CAL has expired and an RD Licensing server isn't available, the connection is blocked. An RD Licensing server with unused Per Device CALs must be available to issue a new permanent CAL.

The simplest way to configure high availability for the RD Licensing role service when using Per Device licensing is to put all RDS Device CALs on a single RD Licensing server. The second RD Licensing server has no CALs installed and issues only temporary licenses. In this configuration, failure of the RD Licensing server with CALs has no effect on devices with a permanent or temporary license, which typically are the majority of devices. Devices connecting for the first time are issued a temporary license from the remaining RD Licensing server without CALs. The only clients unable to connect are devices with an expired license, which should be a small number of devices.

A slightly more complex way to configure high availability for the RD Licensing role service when using Per Device licensing is to split RDS Device CALs among RD Licensing servers. Most CALs are installed on the primary RD Licensing server, but some are installed on a secondary RD Licensing server. This configuration is better because if the primary RD Licensing server fails, then CALs can be issued by the secondary RD Licensing server, and no devices should be prevented from connecting.

Splitting CALs between two RD Licensing servers is slightly more expensive because you need to purchase additional CALs for the secondary RD Licensing server. In a large deployment of RDS, this likely is worth the additional cost to avoid outages. In a small deployment of RDS, it may not be worth the cost because very few users would be affected.

When you have multiple RD Licensing servers, it is critical that you configure the RDS deployment to use the RD Licensing server you have configured with the CALs as the primary RD Licensing server.

To configure the order of RD Licensing servers, perform the following steps:

1. In Server Manager, in the navigation pane, click Remote Desktop Services.
2. On the Overview page, in the Deployment Overview area, click Edit Deployment Properties.
3. In the Deployment Properties window, in the navigation area, click RD Licensing.
4. Select the server you want to be primary, click Move Up until it is at the top of the list, and click OK.

To add an RD Licensing server, perform the following steps:

1. In Server Manager, in the navigation pane, click Remote Desktop Services.
2. On the Overview page, in the Deployment Overview area, right-click RD Licensing and click Add RD Licensing Servers.
3. In the Add RD Licensing Servers Wizard, on the Select A Server page, double-click the server you want to configure as an RD Licensing server and click Next.
4. On the Confirmation page, click Add.
5. Wait until the installation is complete and click Close.

This page intentionally left blank

Index

B

C

D

G

H

I

K

L

M

N

Q

R

T

U

V

W

X

Z