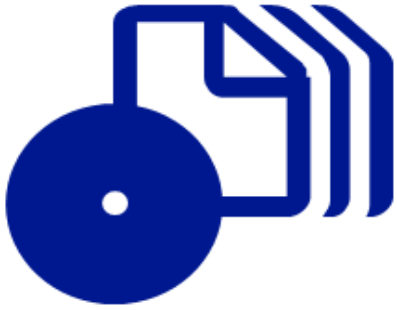
Microsoft

Windows Server 2008 R2 Remote Desktop Services

Christa Anderson and
Kristin L. Griffin with the Microsoft
Remote Desktop Virtualization Team

Resource Kit

How to access your CD files



The print edition of this book includes a CD. To access the CD files, go to http://aka.ms/627376/files, and look for the Downloads tab.

Note: Use a desktop web browser, as files may not be accessible from all ereader devices.

Questions? Please contact: mspinput@microsoft.com

Microsoft Press

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

Library of Congress Control Number: 201093498

Printed and bound in the United States of America.

Microsoft Press books are available through booksellers and distributors worldwide. For further information about international editions, contact your local Microsoft Corporation office or contact Microsoft Press International directly at fax (425) 936-7329. Visit our Web site at www.microsoft.com/mspress. Send comments to msinput@microsoft.com.

Acquisitions Editor: Martin DelRe
Developmental Editor: Karen Szall
Project Editor: Valerie Woolley and Megan Smith-Creed
Editorial Production: Custom Editorial Productions, Inc.
Technical Reviewer: Alex Juschin; Technical Review services provided by Content Master, a member of CM Group, Ltd.
Cover: Cover Design: Tom Draper Design; Illustration: Todd Daman

Body Part No. X17-21601

I dedicate this book to my family, who has always been supportive, always pushes me to do my very best I can do, and always has a "Go team!" waiting when I really need one.

—Christa

I dedicate this book to Elizabeth Nelson-Lyda and Michael B. Smith for taking me under your wing back in the day, and for always believing in me. You were great mentors and are great friends.

—Kristin

Contents at a Glance

Contents

Acknowledgments

This book isn't the work of just two people. We owe many thanks to the combined efforts of a lot of people at Microsoft, our terrific set of editors, and the greater community. (All this said, any errors in this book are the sole responsibility of the authors.)

One of the best things about working at Microsoft is that a lot of very smart (and very helpful) people work there, and we are grateful for the insights of these people. Throughout this book, you'll find Direct from the Source sidebars contributed by members of the product team. We also extend our heartfelt thanks to the members of the product team who sat down with us to explain the finer details of how something worked. From the Remote Desktop Virtualization (RDV) team, we'd like to thank Niraj Agarwala, James Baker, Ara Bernardi, Tad Brockway, Vikash Bucha, Yuvraj Budhraja, Hammad Butt, Rommy Channe, Munindra Das, Silvia Doomra, Samim Erdogan, Rajesh Ganta, Costin Hagiu, Al Henriquez, Travis Howe, Olga Ivanova, Gopikrishna Kannan, Sergey Kuzin, Rob Leitman, Raghu Lingampally, Meher Malakapalli, Benjamin Meister, Ranjana Rathinam, Rajesh Ravindranath, Ray Reskusich, Sriram Sampath, Bhaskar Swarna, and Janani Venkateswaran. Even people from other teams got involved. Many thanks to Kyle Beck, Jeff Heatton, Michael Kleef, Timothy Newton, Mark Russinovich, Tom Shinder, Makarand Patwardhan, Bohdan Velushchak, Paul Volosen, and Jon Wojan for your invaluable assistance. We'd also like to thank Christa's manager, Ashwin Palekar, for his support during this project.

RDS expertise isn't limited to people at Microsoft, either. Remote Desktop Services MVPs as well as MVPs and experts from other disciplines also pitched in to contribute Direct from the Field sidebars and explain the intricacies of related technologies. Many thanks go to Janique Carbone, Brian Ehlert, Ross Harvey, Helge Klein, Russ Kaufmann, Shay Levy, Brian Madden, Patrick Rouse, Greg Shields, Michael Smith, and Mitch Tulloch.

The great team at Microsoft Press had a huge hand in turning this project from an idea into the book you hold in your hands. We'd like to thank Martin Del Re at Microsoft Press for asking us to write the first edition of the book in the first place, Megan Smith-Creed at Custom Editorial Productions, Inc., for great editing and project management on this edition, and Alex Juschin for tech editing the book. The rest of the editorial team at Custom Editorial Productions, Inc., did a terrific job of copyediting and proofing this text. Thank you all!

Finally, we'd like to thank our friends and families for their support during this big project. We couldn't have done it without you. We promise to talk about something else now.

Introduction

Welcome to the *Windows Server 2008 R2 Remote Desktop Services Resource Kit*! This is a detailed technical resource for planning, deploying, and running Microsoft Remote Desktop Services (RDS). Because some features of RDS are brand new, this book is valuable both for those completely new to RDS and those who have used Terminal Services (its former name) in previous versions of Microsoft Windows.

Within this resource kit, you'll find in-depth information about the improvements in RDS introduced in Windows Server 2008 R2. This book combines underlying architectural concepts with practical hands-on instructions that allow you to set up a working RDS ecosystem, understand *why* it's working, and give you some guidance about how to fix it when it's not. You'll also find detailed information and task-based guidance on managing all aspects of RDS, including deploying RD Session Host servers, integrating RDS role services with other key parts of the Windows Server 2008 R2 operating system, and extending the reach of RDS to outside the corporate network. Finally, the companion media includes additional tools and documentation that you can use to manage and troubleshoot RDS role services. Although we mention some third-party tools in the course of this book, this book is fundamentally about running RDS using only the tools found in the operating system. You can do what we've done here using *only* Windows Server 2008 R2. Nor do we get into extensive discussion of any of the third-party tools that many people use with native Remote Desktop Services. For example, many people with high-complexity RDS deployments use management software from Citrix or Quest or other RDS partners, but we don't discuss it here because it's not included with the operating system.

ON THE COMPANION MEDIA **See the team partner page at *http://www.microsoft.com/windowsserver2008/en/us/rds-partners.aspx* for a list of companies that make products complementing or expanding on Remote Desktop Services in Windows Server 2008 R2.**

What's New in Remote Desktop Services in Windows Server 2008 R2?

Remote Desktop Services in Windows Server 2008 R2 took a lot of the improvements added in Windows Server 2008 and added the features people had asked for. Want native support for VDI? It's added to RD Connection Broker. Want

fewer logons, security filtering, simplified discovery of available applications and virtual machines (VMs)? It's in the new version of RD Web Access. Want to address problems discovered via Network Access Policies (NAP), not just shut people out of the network? It's in the new edition of RD Gateway. Want improved application compatibility? See RD Session Host for IP address virtualization and dynamic fair share scheduling that proactively prevents one session from taking all the processor cycles. Want to stop installing printer drivers on both sessions and VMs? Easy Print now works for both virtualization options.

For those who went straight to Windows Server 2008 R2 from Windows Server 2003, let's take a look at what the new features add to the former model of a terminal server and a license server.

Simplified Application Delivery and Display

Terminal Services in Windows Server 2003 presented all remote applications from a desktop, completely separating the display of local and remote applications. RemoteApp programs (introduced in Windows Server 2008) launch from a server, but integrate with the local desktop so they look like they're running locally.

Not only do the applications integrate better with the local desktop, they're easier to find and distribute, thus making it easier to support a larger and more complex deployment. One of the issues in enabling remote access is how to get the most complete and up-to-date set of remote resources to your user base. This is especially true when you're providing access to individual applications, not to a full desktop. Using RDS Web Access, you can present links to individual applications or to entire desktops and know that these links will always be up to date. In Windows Server 2008 R2, RD Web Access can present RemoteApp programs from more than one farm as well as VMs. It also, however, supports security filtering so that you can manage an aggregated source for all remote resources but only display to people the ones they should use.

Improved Farm Support

The Session Directory service in Windows Server 2003 offered the beginning of farm support, but was only available for Enterprise SKUs and didn't include any load balancing—it just kept track of where connections had gone. In Windows Server 2008 R2, RD Connection Broker is available on the Standard SKU, supports load balancing, and can broker connections to both sessions and VMs.

Secure Internet Access

One of the key benefits of Remote Desktop Services is its ability to support mobile workers. We had a great (and extremely itinerant) tech editor, RDS MVP Alex Juschin, for this edition of the book. He's got a great description of how he used Remote Desktop Services while completing his part.

> *In your book you can mention that I have been reviewing your book all over the world using the RDP protocol to connect to my home in Dublin via 3G or WiFi . I've worked while on a smelly Kebap Bus in Poland, in a freezing hotel in Latvia, while being driven in a high-end coach in Estonia, on the ferry to England, in a pub in Ireland, on a train going down the coast from Belfast, while tasting wine in France, sitting in a nice Brasserie on the island of Jersey, eating Belgian chocolate in Brussels, on a plane to Germany, on a bench with a beautiful view in Zurich, in a café near the Berlin Wall, in a prison in Finland (ok, hotel, but it used to be a prison), and on the highest point of Germany (Zugspitze).*

In Windows Server 2003, Terminal Services didn't support secure Internet access except across virtual private networks. In Windows Server 2008 R2, Remote Desktop Services supports connectivity over Secure Sockets Layer (SSL) via RD Gateway. RD Gateway allows you to set up different rules for local and remote access and does not require any client-side setup. Introduced in Windows Server 2008, in R2, RD Gateway now enforces device and resource redirection decisions made at the gateway and supports NAP remediation.

Simpler and Broader Device Redirection

RDS assumes that a lot of people will be working from computers with local resources, and that those people won't want to be cut off from their resources when they're working in their session or VM. It also assumes that the server administrators don't want to spend more time than necessary making these resources available.

Although printer redirection, as it's been known in earlier versions of Terminal Services, still works as it did, Easy Print, introduced in Windows Server 2008, helps simplify printer redirection. Rather than requiring administrators to install printer drivers on the server, Easy Print allows redirected printers to use the drivers already installed on the client computer. In Windows 2008 R2, RD Easy Print works with even more printer types and works from both sessions and VMs.

Part of the rich remote work experience is using local devices. Support for local devices has been expanded through the Plug and Play Device Redirection Framework, introduced in Windows Server 2008.

Simplified License Management

Per-user licensing was introduced in Windows Server 2003 but didn't include any tracking, so you couldn't easily tell if you were in compliance. Windows Server 2008 R2 allows you to track Per-User RDS CAL usage. Additionally, the Licensing Diagnostics feature can help you resolve licensing issues. Windows 2008 R2 RD License servers can now migrate licenses from one server to another without the help of the Microsoft Clearinghouse. This can be done even if a license server is out of commission.

This is only a partial list of new features—Chapter 1, "Introducing Remote Desktop Services," describes the Remote Desktop Services features in Windows Server 2008 R2, and the rest of the book explains how to use them. But these are some of the highlights that show how the role has expanded in management and user experience.

ON THE COMPANION MEDIA **The authors will post data that is relevant to the *Windows Server 2008 R2 Remote Desktop Services Resource Kit* on the book's blog, located at *http://blog.kristinlgriffin.com/*. You can find this link on the companion media.**

How This Book Is Structured

Our goal in writing this book is to help you set up a working Remote Desktop Services farm, as well as VDI pooled and personal VMs using all the pieces in the operating system, while understanding the greater context of the circumstances under which Remote Desktop Services is useful, how it works, and how Windows Server 2008 R2 compares to previous versions. This book has twelve chapters.

- Chapter 1, "Introducing Remote Desktop Services," explains where RDS came from and how it has evolved as a platform, what new features are available in this latest iteration, and what you can accomplish with this new version of the product. It also explains how other services support RDS.
- Chapter 2, "Key Architectural Concepts for Remote Desktop Services," dives into RDS internals and relevant Windows Server 2008 R2 internals. It also shows you how to determine the hardware and software you will need to support this product in your environment.

- Chapter 3, "Deploying a Single Remote Desktop Session Host Server," shows you how RD Session Host servers work, and how to install and configure this role service.
- Chapter 4, "Deploying a Single Remote Desktop Virtualization Host Server," explains what VDI is, how Microsoft VDI works, and how to install and configure a RD Virtualization Host and the supporting roles.
- Chapter 5, "Managing User Data in a Remote Desktop Services Deployment," discusses the different types of profiles that work with RDS and how to deploy and troubleshoot user profile solutions and folder redirection.
- Chapter 6, "Customizing the User Experience," discusses how remoting works, promoting good client experience in the remote session, and how to print from RDS sessions.
- Chapter 7, "Molding and Securing the User Environment," explains why you should lock down the RDS environment and how you should do it, and describes how to provide remote assistance to users from within the user session.
- Chapter 8, "Securing Remote Desktop Protocol Connections," discusses RDP encryption, server and client authentication, and how to configure security settings on the RD Session Host server.
- Chapter 9, "Multi-Server Deployments," introduces key concepts for multi-server deployments, shows how to create RD Session Host farms, and explains how to publish applications and display resources through RD Web Access.
- Chapter 10, "Making Remote Desktop Services Available from the Internet," shows you how to install and configure RD Gateway to provide access to RemoteApps, desktop sessions, and pooled and personal VMs to users located outside the corporate network.
- Chapter 11, "Managing Remote Desktop Sessions," shows you how to monitor and terminate processes and users sessions running on an RD Session Host server, how to provide help with remote control, and how to drain RD Session Host servers for maintenance.
- Chapter 12, "Licensing Remote Desktop Services," discusses the new RDS licensing paradigm, including both RDS and VDI licensing. This chapter explains how licenses are tracked and enforced; how RD License server assign RDS CALs; how to install, configure, and maintain RDS License servers; how to diagnose licensing issues with the Licensing Diagnosis tool; and how to migrate licenses from one server to another.

Document Conventions

The following conventions are used in this book to highlight special features or usage.

Reader Aids

The following reader aids are used throughout this book to point out useful details.

READER AID	MEANING
Caution	Warns you that failure to take or avoid a specified action can cause serious problems for users, systems, data integrity, and so on.
Note	Underscores the importance of a specific concept or highlights a special case that might not apply to every situation.
On the Companion Media	Calls attention to a related script, tool, template, job aid, or URL on the companion CD that helps you perform a task described in the text.

Sidebars

The following sidebars are used throughout this book to provide added insight, tips, and advice concerning different Remote Desktop Services features.

NOTE **Sidebars are provided by individuals in the industry as examples for informational purposes only and may not represent the views of their employers. No warranties, express, implied, or statutory, are made as to the information provided in sidebars.**

SIDEBAR	MEANING
Direct from the Source	Contributed by experts from the product group who provide "from-the-source" insight into how Remote Desktop Services works, best practices, and troubleshooting tips.
Direct from the Field	Contributed by experts external to the product group who have real-world experience working with Remote Desktop Services. Some experts are Microsoft field engineers; others are Microsoft MVPs or other experts.
How It Works	Provides unique glimpses of Remote Desktop Services features and how they work.

Command-Line Examples

The following style conventions are used in documenting command-line examples throughout this book.

STYLE	MEANING
Bold font	Used to indicate user input (characters that you type exactly as shown).
Italic font	Used to indicate variables for which you need to supply a specific value (for example, *file name* can refer to any valid file name).
`Monospace font`	Used for code samples and command-line output.
%VariableName%	Used for environment variables.

Companion Media

In addition to the book itself, you also get a CD that contains some great tools and other resources. System requirements for running the CD are at the back of this book. The CD includes the following resources.

Links

The companion media includes many links to URLs that lead to more information about Remote Desktop Services-related topics, Remote Desktop Services resources, partner web sites, and more. Some of the URLs are referenced throughout the book and some are not

Management Scripts

On the companion media, you will find a collection of scripts illustrating ways to work with Remote Desktop Services using Windows PowerShell and VBScript. We've also included listings in relevant locations in the book so that you can better understand how these scripts support the functionality you're looking for. Although these scripts are intended as samples instead of finished products, they do useful work such as allowing you to easily determine the shadowing permissions on a server or providing application-usage metering not provided in the GUI.

Find Additional Content Online As new or updated material becomes available that complements your book, it will be posted online. The type of material you might find includes updates to book content, articles, links to companion content, errata, sample chapters, and more. This website is available at *http://www.microsoftpressstore.com/title/9780735627376*.

Support for This Book

Every effort has been made to ensure the accuracy of this book. As corrections or changes are collected, they will be added the MS Press Media website. To find Microsoft Press book and media corrections:

1. Go to www.microsoftpressstore.com.
2. In the Search box, type the ISBN for the book, and click Search.
3. Select the book from the search results, which will take you to the book's catalog page.
4. On your book's catalog page, find the Errata & Updates tab. View/Submit Errata.

If you have questions regarding the book or the companion content that are not answered by visiting the book's catalog page, please send them to Microsoft Press by sending an email message to **mspinput@microsoft.com**.

We Want to Hear from You

We welcome your feedback about this book. Please share your comments and ideas via the following short survey.

http://www.microsoft.com/learning/booksurvey

Your participation will help Microsoft Press create books that better meet your needs and your standards.

NOTE **We hope that you will give us detailed feedback via our survey. If you have questions about our publishing program, upcoming titles, or Microsoft Press in general, we encourage you to interact with us via Twitter at *http://twitter.com/MicrosoftPress*. For support issues, use only the email address shown above.**

CHAPTER 1

Introducing Remote Desktop Services

You might be reading this book for any of a number of reasons. Perhaps you're an old hand at Microsoft Terminal Server and are interested in seeing what Remote Desktop Services (RDS) in Microsoft Windows Server 2008 R2 can do for you. You might have installed Windows Server 2008 R2 and are now interested in what all these web accesses, gateways, and Remote Desktop Session Host servers do. Maybe you have heard about RDS and are interested in how you might benefit by incorporating it into your environment. For that matter, you might be wondering how RDS compares to other remote access technologies in Windows Server 2008 R2.

Whichever reason you have to be interested in RDS, this book is for you.

This chapter sets the stage for the rest of the book. To understand the evolution of Microsoft Terminal Services (now called Remote Desktop Services), you have to understand where it came from and the ecosystem in which it operates. To understand what you can do with the roles and role services, you have to understand the essential goals of RDS in Windows Server 2008 R2 and the scenarios that it's designed for. And, because RDS isn't an end in itself but a piece of the broader Windows infrastructure, you'll see how RDS roles interact with other technologies, like Windows Server 2008 Hyper-V and IIS.

After reading this chapter, you'll understand the following.

- Why Terminal Services is now known as Remote Desktop Services
- What Windows Server 2008 R2 includes for supporting a RDS environment
- What scenarios the RDS role services are intended to support
- What kinds of new technology enable those new scenarios
- How RDS role services interact with each other

- How RDS role services depend on other Windows Server roles
- What application programming interfaces (APIs) exist for developers to use, and what are some examples of the kinds of features that developers can add to RDS

Where Did RDS Come From?

If you're looking at RDS for the first time with Windows Server 2008 R2, you'd hardly recognize its earliest incarnations. Like Windows Server itself, RDS has changed a *lot* over the years and has become much more comprehensive. It's not important to go through an exhaustive feature list for each edition, but it's useful to see how multi-user Windows has developed since its inception in the mid-1990s.

Citrix MultiWin

The original MultiWin architecture was designed not by Microsoft but by Citrix, who licensed the Microsoft Windows NT 3.51 source code from Microsoft to create multi-user Windows. [MultiWin was originally going to be based on IBM Operating System/2 (OS/2) when Microsoft was part of the OS/2 project, but Windows won.] Citrix created its own product called WinFrame, which was a multi-user version of Windows NT 3.51 and totally separate from the operating system that Microsoft produced.

A First Experience with Multi-User Windows

Christa first experienced multi-user Windows through WinFrame 1.7 in 1997 at an IBM training center in New York's Hudson River Valley. Training lasted multiple days, so there were hotel rooms in the training center. Originally, the training center provided a PC in each guest room, and staff had to deal with the maintenance headaches of that setup. But by that training session in 1997, they'd moved to setting up thin clients (connected to the WinFrame servers) in all guest rooms so that guests could check email and work from their rooms. When attendees checked in, a script automatically created a user account for that person. This is all common now, of course, but at the time, it was heady stuff and a big change from the desktop-centric model of Windows.

Windows NT, Terminal Server Edition

WinFrame was built on Windows NT 3.51. Microsoft licensed MultiWin back from Citrix in 1995 and plugged this multi-user core into the Windows NT 4.0 base operating system to make a new product: Windows Server with multi-user capabilities. The result was Windows NT 4.0 Terminal Server Edition. Citrix no longer provided a stand-alone product but released MetaFrame, which ran on top of Terminal Server Edition (in much the same way that Citrix XenApp runs on Windows Server now) and added some new features and management tools.

Terminal Server Edition was very much a starting point. The operating system was pretty basic, to put it mildly. Almost every installation of Terminal Server Edition ran MetaFrame on top of it, because the base product did little more than provide a multi-user operating system. Even basic functionality such as clipboard mapping was not included. The fact that Terminal Server Edition and the core operating system were different products wasn't great for either Microsoft or its customers. Microsoft had to deal with two sets of operating system service packs, and customers had to purchase a separate product to test server-based computing *and* juggle two different service packs that were not released at the same time. On the plus side, when there was a problem with Service Pack 6 (SP6) for Windows NT 4.0, it was solved by the time SP6 for Terminal Server Edition was released.

Windows 2000 Server

The first real breakthrough in Terminal Services was in Microsoft Windows 2000 Server. For the first time, Terminal Services was a server role in the base server operating system, not a separate product. Why did this matter? There are several reasons. First, the game of juggling incompatible service packs for single-user and multi-user operating systems was over. Second, there was a fundamental change in the way that server-based computing and remote access were perceived. Before Windows 2000, if you wanted to manage a Windows server from the graphical user interface (GUI), you generally sat down in front of it—there was no capability for remote management using Microsoft Remote Desktop Protocol (RDP). The problem was that there is a limit to the number of servers that you can sit in front of during the day, especially when those servers are in different buildings—or even in different cities. Windows 2000 Server introduced Remote Administration as an optional component, allowing server administrators to manage servers even when they *weren't* sitting in front of them. Not only did this make server administration a lot easier, it also came to the aid of Terminal Services, because it gave people a good use case for remote usage and multi-user computing.

Having Terminal Services in Application Server mode available in the core operating system also meant that trying Terminal Server for users required comparatively little effort—setting up a basic pilot could be done with as little effort as installing the role in Application Server mode and letting people use Notepad. In addition, because RDP in Windows 2000 Server added some basic functionality such as client printer redirection and a shared clipboard between local and remote sessions, trying Terminal Server and getting a feel for how users could benefit from shared computing was possible even with only the tools in the core operating system.

Windows Server 2003

The next big step was Microsoft Windows Server 2003, which took some of the decisions made in the Windows 2000 Server timeframe to their next logical conclusions. If Remote Administration is a good thing, why should it be an optional component? Instead, enable it for all Windows server roles and make it an option for the client. And although the basic functionality in Windows 2000 Terminal Server is useful, it doesn't provide a sufficiently rich

client experience. Let's enable drive mapping, full color, sound, and other features that were previously possible only with third-party products, so that the remote experience can be a lot more like the local desktop experience.

Another big change to Windows Server 2003 was in management. Windows 2000 terminal servers could be managed only singly. You could configure them remotely, but not collectively. Windows Server 2003 introduced some Group Policy settings for configuring and managing terminal servers, and Terminal Server Manager supported management of remote servers.

Windows Server 2008

Microsoft Windows Server 2008 represented a big breakthrough in Terminal Services functionality. Previous versions of Terminal Services had included only two roles: the terminal server and a license server.

> ***NOTE*** **Although Windows Server 2003 included the Session Directory Server for basic farm support, this role was available only in the Enterprise Edition and was not widely deployed.**

If your needs extended beyond remote access to a full desktop on the local area network (LAN), then you needed third-party additions to the role to help you fulfill them. With Windows Server 2008, Terminal Services gained the following advantages.

- Visual integration between locally and remotely running applications
- A web interface for presenting applications on the terminal servers individually
- A secure gateway to enable support for secure access via the Internet
- A session broker to route incoming connections to the most appropriate terminal server
- A printing subsystem that did not require print drivers to be installed on the terminal servers
- Redirection of new types of devices

Windows Server 2008 R2 and RDS

Windows Server 2008 R2 is technically a "minor release" like other R2 releases, but it introduces a lot of changes for RDS. The role service has expanded again to add virtual desktop support (often called VDI, for *Virtual Desktop Infrastructure*). It has also gained some new features, some of the most important being the following.

- Support for connection to Hyper-V based virtual machine (VM) pools of shared VMs and personal VMs assigned to an individual
- Changes to Remote Desktop (RD) Web Access that allow the portal to display resources from multiple RD Session Host servers (formerly known as terminal servers) or farms, and that enable security filtering for RemoteApp programs and VMs

- Improved application compatibility and resource management on RD Session Host
  Support for Aero Glass remoting and other user experience improvements to RDP 7
- Support for forms-based single sign-on through RD Web Access so that users need authenticate only once in the website to get to all the RemoteApp programs assigned to them
- Improvements to Remote Desktop Gateway to enforce drive redirection policies and enable client remediation when clients do not conform to software rules
- Improved discoverability for license servers for a more reliable connection

DIRECT FROM THE SOURCE

Why VDI?

Michael Kleef, Senior Product Manager
Windows Server Marketing

Microsoft added VDI support to Windows Server 2008 R2 to allow customers further desktop delivery choice in thin client computing. Although Remote Desktop Session Host is a mature product and still provides relevant customer value at the right TCO (total cost of ownership) point, there are times when the level of personalization and isolation that VDI with Windows 7 delivers are important for specific use cases. Applications that require elevated permissions are hard to support on an RD Session Host because one elevated-privilege mistake could affect all users of the server. The isolation of VMs makes it possible to support this type of application using VDI. Another example is native application compatibility; this was largely solved by Microsoft App-V, but it can't solve all application issues in which the application requires a Windows client installation. It's for reasons like this that Microsoft invested in delivering a VDI platform in Windows Server 2008 R2 and extended it further in Service Pack 1 with Dynamic Memory and RemoteFX, to increase VM density and improve the rich user experience.

Most obviously, Terminal Services is now called Remote Desktop Services, and all subroles are renamed to go along with the change. The service was renamed to reflect the much broader scope of the server role, including sessions and the role services needed to get people connected to them, but also hosting of VMs and secure wide area network (WAN) access.

NOTE **Because this book is about Windows Server 2008 R2, it uses the current names for the server role and its role services. See Table 1-1 for a list of some of the names you'll come across most often. For a complete mapping of the old and new name for RDS, see *http://technet.microsoft.com/en-us/library/dd560658(WS.10).aspx*.**

TABLE 1-1 Mapping TS Names to RDS Names

FORMER NAME	WINDOWS SERVER 2008 R2 NAME
Terminal Services	Remote Desktop Services
Terminal server	Remote Desktop Session Host server
Terminal Services Licensing (TS Licensing)	Remote Desktop Licensing (RD Licensing)
Terminal Services Web Access (TS Web Access)	Remote Desktop Web Access (RD Web Access)
Terminal Services Gateway (TS Gateway)	Remote Desktop Gateway (RD Gateway)
Terminal Services Client Access License (TSCAL)	Remote Desktop Services Client Access License (RDSCAL)
Terminal Services Manager	Remote Desktop Services Manager
Terminal Services Configuration	Remote Desktop Services Configuration

The pattern is pretty obvious; if any names you see don't make sense, look at the list provided at the link.

The Evolving Remote Client Access Experience

Although this book focuses on the server shared-computer experience, not the client, it is important to know that RDS also changed on the client side as the server-side capabilities evolved. Microsoft Windows 2000 Professional did not support incoming remote access connections (nor did Microsoft Windows 9.*x*), but Microsoft Windows XP, Windows Vista, and Windows 7 all do. Supporting incoming remote connections enabled several new ways to use Windows clients, including

- Remote access to a physical computer from home or another area of the building
- Remote Assistance
- Virtual desktop hosting
- Hosting RemoteApp programs to be displayed in another client operating system (for application compatibility)

Remote access from another computer reflects the reality that many people use more than one computer, and that a home might have more than one computer. Remote Assistance uses the remote control feature of RDS—the ability to permit a second person to see or even take over a remote session—for enabling help desk support, even on desktops. Virtual desktop hosting was one of the chief competitors to session hosting for a long time (and is now part of the service). Features like RemoteApp on Hyper-V allow people to run applications on an older operating system while seeing them on a newer one, even if the application won't run on Windows 7 for some reason.

NOTE **Generally speaking, most 32-bit applications can run on a 64-bit platform as long as these applications don't include drivers and don't have a 16-bit installation routine. Web applications designed to run in Microsoft Internet Explorer 6 are one exception to this rule. Internet Explorer 6 is included with Windows Server 2003, but can't be installed on Windows Server 2008 R2. Therefore, if you have Internet Explorer 6–dependent applications and want to display them as RemoteApp programs, you can host them in VMs using RemoteApp for Hyper-V.**

RDS shows up in the client versions of Windows even when you don't expect it. It's the technology that enables Fast User Switching and Remote Assistance (to name just two), and a version of the RDP protocol is the basis of Live Mesh.

In short, the story of Remote Desktop Services is the story of how multi-user computing has become less of a niche technology and more of a Microsoft strategy for enabling various scenarios that blur the line between the PC and the data center. Even when they're not called RDS, multi-user computing and the Remote Desktop Protocol have become crucial parts of the core Windows platform.

What Can You Do with RDS?

The preceding section provides a (very fast) look at where RDS came from and how it became part of the core Windows platform for both client and server. You will learn about the technology in depth in later chapters. But what do you *do* with it?

Fundamentally, RDS breaks the hard links between location, client operating system, and capability.

In many ways, this is a natural extension of networking. If you're using a single computer unconnected to any networks, you're limited to the applications and data stored on that computer. If you attach that computer to a network and enable file sharing, you can use data that is not stored on your laptop, and a systems administrator can both back up that data (impossible for someone else to do on an isolated desktop) and secure it. With RDS, you can use not only data stored somewhere else but also applications stored somewhere else. They don't even have to be capable of running on the client computer as long as they'll run on the host. Presentation remoting improves file sharing because the files you use don't have to be accessible to the client computer as long as they're available to the back-end application.

With an isolated PC, you are absolutely tied to what that computer can do. With presentation remoting, the capabilities are more flexible, because what you see isn't necessarily running on the computer where you're working, or even in the same country. This has benefits for security, location, and device independence.

Improved Security for Remote Users

Totally PC-based computing has problems with data security. More and more people work on laptops, and laptops are meant to be taken places. But laptops with data stored on them are a security risk, even if you password-protect the laptop. Unless you take the laptop with you *everywhere*, including lugging it along to dinner instead of leaving it in the hotel room when you're on the road, the data on your laptop is vulnerable to theft. And if someone *really* wants the laptop, it doesn't matter if you take it with you. This doesn't even address the dilemma of leaving the laptop in a taxi or on a train by accident. It happens. BitLocker technology on Windows 7 and Windows Vista protects against theft but does not protect against loss from a misplaced or broken laptop that wasn't backed up.

If the data is on the laptop and you lose the laptop, the data's gone. The obvious solution is not to keep the data on the laptop—store it in the data center instead. But if you're accessing the data center from a remote location via a virtual private network (VPN) and working with large files (in this day of heavy-duty formatting, what file *isn't* large?), it's tempting to keep the file on the local drive while working on it remotely and then copy it back to the network when you're done with it. However, if you work this way, you're back where you started with the data on the local drive.

Information Insecurity

It's not practical to make sensitive information accessible only to people within the four walls of the office, but it's been shown again and again what happens when that information leaves the data center. In November 2009, the Army Corps of Engineers lost a hard drive containing the names and social security numbers of as many as 60,000 current and former Army service members and some civilians. As of this writing, the drive has not yet been recovered. This isn't the first time that sensitive data has been lost to a misplaced laptop or other portable media.

It's not always feasible to store sensitive information only in the data center, accessible solely via secure connection to a Remote Desktop Session Host server behind the perimeter network. Sometimes, the information must be available even when a network connection isn't. But when it is feasible, it's much more secure to keep information where it's least likely to be compromised, stolen, or lost: in the data center.

One solution to the dilemma of how to secure data while keeping it accessible to the people who need it is to keep *everything* in the data center, including the applications required to edit the data. If both the applications and the confidential data are on the network, then it's either impossible to edit the data locally (because no application for doing the editing is installed locally) or not as desirable to do so because there's no reason to download the remote file to the local computer for a more responsive experience. No sensitive data ends up on the client computer; it all stays within the boundaries of the data center.

NOTE **Given a sufficiently long distance or sufficiently slow Internet connection, the remote connection will also be slow; and if the network connection isn't totally reliable, it can be frustrating as the session disconnects. As you know all too well, even high-speed networks experience some latency when you're working on one continent and the data center is on another one. But these problems apply to any remote-access scenario and have less chance of accidentally corrupting the original document by attempting to write to it over a slow connection. A disconnected session doesn't lead to data loss—it's just there waiting for its user to reconnect to it.**

What if you want people to be able to edit confidential documents when they are in a secure location but not when they're accessing the corporate network from the local coffee shop? Using RDS in Windows Server 2008 R2, you can set up rules that determine which applications a remote user has access to, whether the user has any local drives mapped, and even whether it's possible to cut and paste text between local and remote applications. Security needs can determine the restrictions placed on remote access while still keeping the data easily available when it should be.

Provisioning New Users Rapidly

This is especially useful for temporary workers. If you are providing computer services for someone who will only be around temporarily (for example, a consultant needing a temporary desktop or a temporary worker) then it's good not to need to spend much time on setting up a computer for her, but also good to give her a clean work environment that doesn't require her to work around the detritus left by the previous user of the computer. Through RDS, you can get a new user set up and working almost as quickly as you're able to get her a domain account. In addition, the pooled VM or remote desktop session the person uses will be brand new, with no old settings left from a previous user, which should simplify troubleshooting and training.

Enabling Remote Work

Related to security for mobile workers is remote work. Telecommuting is becoming more common in the workplace. Some help desk suppliers and U.S. government agencies don't even have desks for all their workers, since their workplaces are designed for most people to be working from home most of the time. According to the Status of Telework Report to the Congress (see *http://www.telework.gov/Reports_and_Studies/Annual_Reports/2009teleworkreport.pdf*), over 100,000 people working for the U.S. government teleworked during 2008, with 64 percent of these teleworking at least 1 to 3 days per week. This represents an increase of just under 9 percent since 2007.

Nor is telework a solely North American phenomenon. In 39 percent of western European companies, some people work at home at least part of the time, according to "IT and the Environment," a 2007 paper by the Economist Intelligence Unit.

But working from home has its own set of challenges, not least being the question of how the company can support the desktop environment. Home-based computers can't be easily managed by Group Policy; they can break down with no IT staff immediately available to provide assistance, and people working from home can't always readily talk through a computer-based problem with help desk staff. And how do you update an application when it's time to move from, say, Microsoft Office 2007 to Office 2010? If you've worked remotely for even a brief span of time, you probably have experienced the advantages of mobility and the disadvantages of lack of local support. It's great being able to work from the coffee shop, hotel, or airport lobby; it's not so great acting as your own help desk.

Server-based computing helps enable remote scenarios in several ways. You don't have to worry about home users installing applications that they shouldn't run on the Remote Desktop Session Host servers if you follow basic security procedures (more later on this topic). Since the applications are stored on the RD Session Host servers, they're installed and updated there, not on the clients. And, as discussed in the previous section, "Provisioning New Users Rapidly," using RDS allows the administrator to determine the kind of resource sharing that the local and remote computers should do and which applications are available, depending on the location from which a user is connecting.

Bringing Windows to PC-Unfriendly Environments

Not all the people who need a PC work in an environment that allows them to have one. One example is electronics firms. If you're making circuit boards, you make them within what's called a *clean room*, a room with no dust and which requires a time-consuming process to enter. If you need to use Windows applications in a clean room, you can't use PCs. The fans inside the case kick up dust inside the computer and spread it into the room. In addition, it's not practical to have PCs that might need servicing in any room that takes extensive preparation to enter as a clean room does. Therefore, you need RDS to provide Windows applications to the terminals.

Thin clients are also good for environments where you want access to Windows applications but the circumstances are not PC-friendly, if they've got too much dust or vibration to be good for the PC. Small terminals that can be wall-mounted or carried work better in these circumstances than PCs do. But since these small terminals have very limited memory and CPU power and no disks, you can't run Windows 7 on them. To get access to the latest operating system and applications, you need an RD Session Host server for the terminals to connect to.

PC-less Windows environments include places such as upscale health clubs or city apartment lobbies. Management wants to attract customers by offering the convenience of a personal computer in the lobby or cafe but doesn't want to support computers in these locations. (Bulk can also be an issue when you're trying to squeeze five user work areas into a small counter space.) Windows terminals can connect to an RD Session Host server and present the applications. They're also smaller, cooler, and more reliable than PCs, which can get misconfigured.

It has been said that there's no point to getting thin clients because if you buy PCs, you get more power for the same money. With thin clients, you're not paying for the computing power; you're using very little, comparatively speaking. You're paying for the reduced administration and smaller physical footprint and energy use. This solution is not for everyone, but sometimes thin clients are a better choice than PCs.

Business Continuity and Disaster Recovery

One advantage of RDS is that it enables you to set up user work environments quickly. As long as the servers are available in the data center, they can be made available to users almost as quickly as the user's computer is plugged in and turned on. Using a combination of centralized application installs and Internet access, it's possible to set up a new branch office quickly even if the RD Session Host servers are located offsite. For maximum flexibility and ease of setup, this model assumes that the RD Session Host servers are user-agnostic (that is, all user information, including profiles, is stored elsewhere) and identically configured.

Supporting Green Computing

One of the hot topics (no pun intended) these days is how to make companies and governments greener—how to help them use less energy. IDC, a market-research firm, says that power consumption is now one of systems managers' top five concerns. Companies now spend as much as 10 percent of their technology budgets on energy, says Rakesh Kumar of Gartner, a consultancy. (Only about half of this amount is used to run computers; much of it goes toward cooling them, since for every dollar used to power a server, you spend a dollar to cool it.) Dropping power usage is a win-win situation, really—because companies have to pay for their power, using less energy means that they spend less money on power.

> ***NOTE*** **A December 2007 paper from McKinsey & Company, "Reducing U.S. Greenhouse Gas Emissions: How Much at What Cost?" (*http://www.mckinsey.com/clientservice/ccsi/pdf/US_ghg_final_report.pdf*), shows the marginal costs of reducing carbon dioxide emissions. The cost of reducing the carbon emissions for combined heat and power in commercial buildings is negative. That is, it pays companies to go green.**

There's a *lot* of waste in desktop-centric computing. According to IDC, average server utilization levels range from 15 to 30 percent. Average resource utilization rates for PCs have been estimated at less than 5 percent. Because you have to power the processor and memory whether you're using them or not, this represents a lot of waste. Therefore, depending on the needs of the client, there might be quite a bit of room for people accessing their desktops—or at least their applications—from an RD Session Host server. For companies that can reasonably exchange desktop computers for Windows-based terminals, this can represent a huge savings, both in terms of the power drawn by the full desktops and in terms of the air conditioning required to cool the building heated by hundreds of powerful PCs.

Improved Command-Line Support

Windows Server 2008 had a wide array of programmable interfaces that duplicated—and even extended—the capabilities of the GUI. What it didn't have was the best way to get at them. Windows PowerShell supported Windows Management Instrumentation (WMI) but had no remote access capabilities (and finding the right WMI object isn't trivial unless you already know what you're looking for), so you couldn't use Windows PowerShell to manage settings on a server farm. VBScript did support remote access and WMI, but it required knowing how to script. (You also need to learn to use Windows PowerShell to use it, but it's simpler and a lot of basic tasks have cmdlets already prepared.)

Command-line management is simpler in Windows Server 2008 R2 for two reasons. First, the Windows PowerShell team introduced remote access support in Windows PowerShell 2.0. Second, the RDS team created Windows PowerShell objects to map to its WMI structure. It's now possible to easily find the capability that you want according to server role, and the objects are fully supported by standard Windows PowerShell cmdlets. You'll be reviewing throughout this book how to use Windows PowerShell to manage the RDS farms.

NOTE **For an example of the kinds of things you can do with Windows PowerShell and RDS, see the RDS team's Script Center site at *http://technet.microsoft.com/en-us/scriptcenter/ee364707.aspx*. Some scripts use VBScript for backward compatibility with previous operating systems.**

RDS for Windows Server 2008 R2: New Features

So far, you've seen an overview of some of the ways you might apply server-based computing to meet your company's needs for supporting remote workers or PC-unfriendly environments. Many new features in Windows Server 2008 help you support these scenarios specifically. This book is devoted to letting you know what's new in RDS and how to use it. This section discusses some of the features and how this version of RDS differs from previous versions in ways larger than individual features.

DIRECT FROM THE FIELD

New Features of RDS You Might Not Have Heard Of

Greg Shields, RDS MVP
*Partner and Principal Technologist with Concentrated Technology (*www.ConcentratedTech.com*)*

RDS in Windows Server 2008 R2 gets a lot of press, because there's so much in it that's new and exciting. But in among its heavy-hitter updates are a few that you might not know about.

For example, did you know that its *Dynamic Fair Share Scheduling* ensures that each user on the same server gets an equal amount of processor attention? With it, a lightweight user running Microsoft Word can collocate with a heavyweight user performing a software build, or crunching a database query, or any other CPU-intensive activity. Neither session is impacted by the actions of the other.

***Remote Desktop IP Virtualization* is also new for those finicky applications that require unique IP addresses to function. Without it, all applications running from the same RD Session Host will appear to have the same IP address. With it, an RDS server can virtualize a set of IP addresses so that those applications execute without problems.**

Even *Windows Installer* gets improved with Windows Server 2008 R2. In previous operating system versions, Windows Installer wasn't fully Terminal Services–aware. This limitation made the installation of some applications very difficult as concurrent installs would block each other. That awareness is finally present in R2, improving the success rate of installing applications to RDS. Installing MSI packages on an RD Session Host server is the same as installing them on a client computer—they serialize and don't block.

With R2, your options for connecting users to applications become as important as the application delivery itself. This "feature" isn't so much a feature as a completely new way of thinking about *application delivery*. The incorporation of RemoteApp and Desktop Connection in Windows 7 with the RD Web Access in Windows Server 2008 R2 gives you more options for how you connect users to their applications. Depending on your needs, you can deliver RemoteApp programs and VMs via a web page in Internet Explorer, through an .RDP file delivered to the user, or, for those using Windows 7, you can simply populate your users' Start menu.

The Changing Character of RD Session Host Usage

One RDS change in Windows Server 2008 R2 is in the usage assumptions. Windows Server 2003, for example, assumed that administrators will generally run individual servers from the corporate LAN (and probably only one or two of them) since the session brokering piece is available only in the Enterprise edition of the software. Windows Server 2008 assumed that terminal servers would be hosted in farms, that people would run both locally installed applications and RemoteApp programs, and that at least some people would be accessing the RD Session Host servers from the Internet.

RDS in Windows Server 2008 R2 expands on the assumptions in Windows Server 2008 to assume the following, among other things.

- Many users access the corporate LAN from the Internet at least some of the time.
- Users don't always log on from domain-joined computers.

- Users are more likely to use a PC (with some locally installed applications) than a terminal device.
- Users might work from a branch office but still are connected to the domain.
- Some users will run very demanding applications from the data center.
- Applications will be served from a farm of identical servers more often than a single server.
- Some users will be allowed to install applications even in a hosted workspace.
- Some applications should be isolated for best compatibility.

You will learn about some RDS role services here, but a technical walkthrough of these features is less important right now than understanding the business problems that they're designed to solve. The rest of this book will provide design, deployment, and operations guidance.

Supporting VM Users

Sessions are a good way to enable that a lot of people use the same physical hardware. However, sessions don't work for everyone, especially not if desktop replacement is the goal. A session can't permit its users full administrative access to tweak settings through the Control Panel, isn't always friendly to resource-hungry applications (at least, the resource-hungry applications are not always friendly to the other sessions), and doesn't permit users to install applications to use later in exactly the same environment. Nor can you hibernate a session to easily save not just data, but also the work that you were in the middle of completing when you dropped everything and ran to catch the bus. Using a VM, it is literally possible to save your work state.

One new feature in Windows Server 2008 R2 is native support for Virtual Desktop Infrastructure (VDI), which is a short name for "managed virtual machines." Microsoft VDI supports two kinds of VMs. *Personal desktops* are assigned to an individual and can be customized according to whatever rules are in place in the organization. *Pooled desktops* are generally available to anyone with access to the pool. Although it is possible in some cases to make changes to them, there is no guarantee that a user changing a pooled desktop will get the same one the next time they log in—rolling back changes is often normal, to avoid people contaminating the desktop pool with applications and settings they will never reuse.

Each kind of desktop is designed for a different purpose. Personal desktops are for full desktop replacement. Although accessible only via RDP, a personal desktop is controlled by the user it is assigned to, and if a person has a personal desktop, the RD Connection Broker will always attempt to connect them to it first. A personal desktop can replace a physical computer and even has the advantage of making the machine state easy to back up, so moving to a new physical platform doesn't mean losing all settings.

Pooled desktops are more for supporting people who need to run applications that aren't well hosted on an RD Session Host server, even with the new support for fair share processing

that prevents a single session from using all the processor power. They can be preinstalled with any applications that the people who need the pool will need.

Pooled desktops can also support an application-compatibly feature released after Windows Server 2008 R2 shipped: RemoteApp on Hyper-V. This feature allows you to run RemoteApp programs from a VM rather than from an RD Session Host server. It's designed to allow computers running Windows 7 that need to run an application that can't run on Windows 7 (for example, a web application based on Internet Explorer 6) from a computer running Windows XP located in the data center. Although each VM can still only support one incoming connection at a time, RemoteApp for Hyper-V makes it possible to support these older applications while retaining the features of Windows 7 on the desktop.

How to Get RemoteApp Technology from a Client

Remoting technology is great for displaying applications that can't run on the client. For example, you can run really demanding applications from a session or a VM to integrate with an older operating system or on hardware that won't support them.

Supporting older applications that won't run on an operating system later than Windows Server 2003 and Windows XP is a bit more problematic. Windows Server 2003 didn't include support for RemoteApp technology, so to run the older applications there would mean publishing only from a full desktop. And up until now, Windows XP didn't support RemoteApp connections (although some companies had solutions that did something functionally similar).

Microsoft has several different technologies that support RemoteApp from client operating systems such as Windows XP. They're all intended for different user scenarios.

XP Mode uses Virtual PC technology to run a Windows XP VM on a computer running Windows 7. People with their own computers would run this to enable themselves to run applications locally that will not run on Windows 7. To get XP Mode, go to *http://www.microsoft.com/windows/virtual-pc/download.aspx*.

MED-V is essentially managed XP Mode (see *http://blogs.technet.com/medv/archive/2009/04/30/windows-xp-mode-in-windows-7-how-it-relates-to-future-versions-of-med-v.aspx*). You'd use this to deploy XP Mode in an organization so that you don't rely on individuals to update their own RemoteApp guest machines.

The catch to XP Mode is that it requires the RemoteApp VM to run locally. Not all computers have the hardware to run two full machines at the same time (required with Type 2 hypervisors like Virtual PC). To make it possible to support RemoteApp from Windows XP, there's RemoteApp for Hyper-V. This model runs the Windows XP guest VMs hosting the RemoteApp programs in a data center and uses RDP to

Continued on the next page

display them on a computer running Windows 7. To get the updates required to use RemoteApp for Hyper-V, go to *http://support.microsoft.com/kb/961742*.

MED-V and XP Mode are outside the scope of this book because they do not use the RDS infrastructure, but RemoteApp for Hyper-V is discussed in more detail in Chapter 3, "Deploying a Single Remote Desktop Session Host Server."

Supporting Telecommuters and Mobile Workers Securely

The way that people work in information fields has changed a great deal over the years. At one time, most information workers (the best way to describe people who need regular access to a shared pool of data to do their jobs) went to where the information was: namely, to the office. When they left the office, they stopped working on anything that depended on that central pool of information. Similarly, when they were in the office, they could easily add to this central pool of information—after all, all this information is created by people—and when they left, they could not continue adding to the central pool of information.

Laptops changed this by giving telecommuters a computer that they could easily take with them, but laptops still didn't have access to the central pool of information that people could access at the office. Widespread Internet access combined with the increasing use of email as a personal information store gave additional access, but email doesn't include *everything* your company knows—just that information included within emails you've sent or received.

The next stage was securely connecting to the corporate network, retrieving the information required, and then downloading it to the laptop. This, of course, required both broad access to high-speed networks for downloading the documents to the local computer and also for the application to be installed locally. It also meant that people needed some way for the laptop to access the data center without creating a security breach or spreading a virus on the corporate network.

Much of the industrialized world today has access to the necessary components: laptops and high-speed networks that are available both at home and in public places such as airports and hotels. The tricky problems that arise include how to regulate which computers are allowed access to the network and how to keep sensitive data off computers vulnerable to theft or loss. There's also the problem of gaining access to the data that mobile workers create while on the road. Data stored on a laptop won't make it back to the corporate network until the road warriors get back from the trip, or at least get some free time to upload all their new data to the central data pool.

RDS long held promise in supporting telecommuters and mobile workers, but the solution included with the operating system didn't have all the tools needed to make this work until Windows Server 2008. Windows Server 2008 Terminal Services changed this, introducing Terminal Services Gateway (TS Gateway). TS Gateway enabled authorized users to access authorized corporate resources securely via RDP tunneled through the Internet. Windows Server 2008 R2 added some enhancements for increased security in the new version of TS Gateway, called Remote Desktop Gateway (RD Gateway).

RD Gateway enables users to access the corporate network—and the centralized data pool—securely via SSL from the hotel or airport or even the beach (if you can keep sand out of your laptop). When combined with RDP file signing and server authentication, RD Gateway provides secure Internet access, giving users some assurance that the RDP file that they launch is a legitimate resource and not a spoofed server set up to capture their logon credentials. RD Gateway can also set policy to protect the data center, controlling which people and computers are allowed to access the data center via this path and letting administrators control what resources they have access to once they get there.

NOTE **RD Gateway and SSL aren't the only ways to create a secure connection to the data center from a remote location—VPNs and Direct Access are other access options. But RD Gateway has some advantages, including controlled access to specific resources, which is discussed in detail in Chapter 10, "Making Remote Desktop Services Available from the Internet."**

Using Public Computers Without Storing Connection Data

The previous section discussed personal laptops, and that's what most people use to access the data center while on the road. However, it's not reasonable to expect that people will *never* log on except from a computer that they own. For example, you could be connecting to the corporate RD Session Host servers from a computer at your family's home in Tucson, or from a kiosk at an Internet cafe in Darmstadt. In both cases, you need a way to access work resources without leaving any personal data cached on those computers, including an RDP file used to point to the data center.

Remote Desktop Web Access (RD Web Access) has features that enable you to do this. Rather than storing connection settings in an RDP file that you can get in email or save to a desktop, RD Web Access is a secured website that displays icons representing shared desktops and RemoteApp programs. When a user clicks a link, RD Web Access generates the RDP settings for the resource to which the user is attempting to connect. With the advent of forms-based authentication in Windows Server 2008 R2, users can log onto the website once, then use the same credentials to access all RemoteApp programs displayed in the browser.

RD Web Access and RD Gateway are independent role services, but they can be combined to provide secured Internet access without depending on saved RDP files.

Integrating Locally Installed Applications and RemoteApp Programs

RDS in Windows Server 2008 R2 doesn't require a specific client operating system to work; you can connect to a VM or to an RD Session Host server using clients as old as RDP 5.2. (Previous versions of RDP aren't supported because of security improvements in RDP 5.*x*.) However, you'll definitely get the best experience using RDP 7. This version of the client enables some new visual remoting not possible with previous versions. Like Terminal Services in Windows Server 2008, RDS continues to blur the line between client and server.

One feature of RDS depends on a capability in the client operating system and is available only to clients running Windows 7: RemoteApp and Desktop Connections. (For those using Windows Server 2008 R2 as a client, it's also possible to set up this feature from this operating system.) You will learn about this feature in detail in Chapter 9, "Multi-Server Deployments," but in short, it allows users to add icons automatically from applications running in the data center to their Start menu.

NOTE **For the best user experience, you should use the latest version of RDP (7, as of this writing) but many features are available even to older versions of the RDP client. See Chapter 6, "Customizing the User Experience," for more details.**

Supporting High-Fidelity User Experience over RDP

Early versions of Terminal Services made it very obvious that you were connecting to a remote computer. The color quality was low, you couldn't redirect devices, you couldn't use more than one monitor, the quality of audio redirection wasn't the best, and so forth.

Windows Server 2008 R2 makes it easier to work remotely by supporting the following features.

- True multi-monitor support, including varying layouts and both landscape and portrait orientations.
- Aero remoting for single-monitor sessions on Windows 7.
- Client-side rendering of multimedia and audio Windows Media Player files.
- Improved display of video from Silverlight and Windows Media Foundation.
- Bi-directional audio remoting, including sound recording to a remote session.

Working from Branch Offices

Working remotely isn't a label just for those working from home or while on the road. "Remote" workers might operate in a separate office, but one with resources similar to the corporate office. In this scenario, the network is reliable, the computers are domain-joined . . . but the data center is not in the same physical location as the branch office workers, and onsite IT staffing might be minimal.

Supporting Larger Server Farms

RDS deployments don't consist of just one or two servers anymore, but the tools available in Windows Server 2003 didn't really support farms. (Session Directory Server was available only on the enterprise edition of Windows Server 2003.) Windows Server 2008 R2 RDS is more suited to managing access to multiple servers because it adds additional group policies for server management and the RD Connection Broker enables users to connect to farms instead of single servers.

Other Business Cases for RDS

Administrators benefit from RDS, too.

Regulatory Compliance Requirements

For the IT department, data security and the ability to meet regulatory requirements both remain top priorities. RDS helps secure an application and its data in a central location, reducing the risk of accidental data loss caused by, for example, the loss of a laptop. Key features of RDS, such as RD Gateway and RemoteApp combined with RD Web Access, help ensure that partners, or users, who do not need full access to a company network or computers can be limited to a single application, if needed.

Complex Applications

In an environment with complex applications such as line-of-business (LOB) or customized older software, or in situations in which large and complex applications are frequently updated but are difficult to automate, RDS can help simplify the process by reducing the burden of managing multiple applications across the entire environment. The client machines can access the applications they require from a central source, rather than requiring applications to be installed locally.

Merger Integration or Outsourcing

In the case of a merger, the affected organizations will typically need to use the same LOB applications, although they might be in a variety of configurations and versions. In addition, organizations might also find that they are working with outsourced or partner organizations requiring access to specific LOB applications but not to the full corporate network. Rather than performing a costly deployment of the entire set of LOB applications across the extended infrastructure, these applications can be installed on an RD Session Host server and made available to the employees and business partners who require access, when they need it.

New RDS Technology in Windows Server 2008 R2

New technology in RDS in Windows Server 2008 R2 does a lot to improve the user experience. Part of the goal of this release was to make the remoting unobtrusive so that an application executing remotely should appear to be executing locally. In this section, you will learn about some of the technology in this release that enables this. The rest of this book will go into more detail.

Integration of RemoteApp Programs and Desktops into the Start Menu

Technically, it was possible to integrate RemoteApp icons with the Start menu in Windows Server 2008. To do so, you had to

1. Package the RemoteApp from the RD Session Host server as a Microsoft Windows Installer (MSI) file.
2. Publish this MSI file through Group Policy.
3. Repackage and republish manually as required when the RemoteApp settings changed.

It's not a bad system, and MSI publishing is still the only way that you can support file associations with RemoteApp programs. (It's also the only way you can integrate RemoteApp programs with the Start menu on Windows XP and Windows Vista.) However, it doesn't update automatically, and you can't add more RemoteApp programs to the Start menu without editing Group Policy. Finally, since it requires Group Policy, you can't use this method to publish applications to computers outside the domain.

A new feature called RemoteApp and Desktop Connections avoids these drawbacks. A new application Control Panel item in Windows 7 (and Windows Server 2008 R2) called RemoteApp and Desktop Connections can accept a Uniform Resource Locator (URL) for the *publishing feed* created from the farm. This feed aggregates all the RemoteApp programs, VM pools, and personal desktops available. When a user connects to the URL for the feed and presents their credentials, RD Web Access filters the display so that they get links only to resources that they are permitted to use. These links then populate the client's Start menu.

Using RemoteApp and Desktop Connections has the following advantages.

- It allows users to start locally installed applications and RemoteApp programs in the same way: through the Start menu.
- It does not require the computer running Windows 7 to be connected to the domain.
- It updates automatically whenever RemoteApp programs or VMs are added to or removed from the feed, or when permissions change.
- Users have to log on only once to create the connection.
- Finally, this feed is written in XML, an industry standard, and is available to developers to consume in other ways.

Aero Glass Remoting

One of the visual limitations of Windows Server 2008 was that Windows Vista had this great Aero Glass interface . . . but this wasn't available from terminal server sessions. Today, Aero remoting is available when connecting to Windows 7 VMs and Windows Server 2008 R2 sessions from a client running Windows 7—even if the endpoint can't display Aero itself (for example, if connecting to a headless computer).

Aero Glass remoting from Windows 7 is enabled by default; to enable it from Windows Server 2008 R2 requires turning on desktop composition. The details are discussed in Chapter 6.

NOTE **Although you can get Aero remoting from Windows Vista to Windows Vista, Aero remoting from Windows 7 or Windows Server 2008 R2 requires the Windows 7 client operating system.**

Aero Glass remoting is available for single-monitor sessions only.

Improved Application Compatibility

One of the interesting questions about applications, especially those that are a little fussy, is whether they will work on an RD Session Host server. Three new technologies in Windows Server 2008 R2 RDS seek to address application compatibility problems.

- Changes to the process of installing MSI packages make the installation process work more as it does on client operating systems. Chapter 3 goes into the details, but the impact is to prevent simultaneous first-time uses of applications based on MSI installs from blocking each other.
- Windows Server 2008 has Windows System Resource Manager (WSRM) for preventing single sessions or processes from using up all the processor time. Windows Server 2008 R2 still supports WSRM, but it also introduces a new feature for preventing this problem in a more proactive manner. Whereas WSRM identifies badly behaving applications and scales back their processor time, Dynamic Fair Share Scheduling (DFSS) works with the scheduler to ensure that a single session never starves other sessions for processor cycles. You'll learn about this in more detail in Chapter 3.
- Finally, IP virtualization makes it possible for a session—or only certain applications running in a session—to have a unique IP address. In previous versions of Terminal Services, all applications on a server would have the same IP address: the server's IP. Although this worked much of the time, it prevented applications or security scenarios that required a discrete IP address. Again, you'll find out more about this feature in Chapter 3.

Support for True Multi-monitor Remoting

Version 6 of the Remote Desktop Connection client introduced monitor spanning, so you could use two or more monitors (up to a resolution of 4096 × 2048) to display a remote session. To get this, you connected to the terminal server using the */span* switch. Span was an improvement over being limited to a single monitor but had some drawbacks.

- The monitors had to be arranged in a row.
- The remote session was still a single-monitor session—just one with a *really* big monitor. Because of this, if you had only two monitors, error messages displayed in the middle of your screen sometimes got bisected or obscured. In addition, maximized applications would take up all the monitor space.

Again, the total supported resolution had to be below 4096 × 2048 (for example, 1600 × 1200 + 1600 × 1200 = 3200 × 1200).

RDS replaces monitor spanning with true multi-monitor support. With multi-monitor support, each monitor on the client machine is redirected individually, so that each monitor (up to 16) is seen as a separate monitor to the remote session. (Group Policy limits it to 10, but it's technically possible up to 16 if you set this value programmatically.) Therefore

- The monitors can be arranged in any configuration that makes sense to the user: a row, a box, an L, and so forth.
- Individual applications will maximize to the size of the monitor they're currently displayed in, not the entire row of monitors.
- Each monitor can have a maximum resolution of up to 4096 × 2048.

True multi-monitor is not supported with Aero Glass remoting. If multi-monitor and Aero Glass remoting are both configured, multi-monitor will take precedence.

Remoting huge and high-resolution displays can take a toll on server performance, so you might want to tweak the maximum supported resolution and maximum supported monitors. For more details, see Chapter 6.

Client-Side Multimedia Rendering

Many modern personal computers, even modest ones, have a lot of power—more than a server does to render all multimedia in a session on the server and then stream it to the client, at any rate.

In Windows Server 2008 R2, the RDS team has improved the media playback experience by efficiently transporting audio/video-based multimedia in a compressed format within the RDP protocol. Rather than being rendered on the server, it's sent to the client to be played back through Windows Media Player. The content will appear to be displaying locally because it is—even though it was originally generated in a remote session. However, it will also be fully integrated with the remote session.

This approach has several advantages.

- It reduces bandwidth usage since data over the wire will be compressed video instead of a succession of bitmaps; the experience is roughly equivalent to running from a file share or video server. Resizing the window won't affect the playback, either.
- It reduces the processing on the server because the server no longer needs to use processor time decoding the video and packaging it on RDP.

To support this, the client must support multimedia redirection and the server must be configured for audio and video playback. This feature is covered in more detail in Chapter 6.

Single Sign-On for Farms

Single sign-on, or having to present a password only once to use resources from your computer, is obviously good for users. Imagine coming to work in the morning and logging on to your computer. Then you click an icon and need to present credentials again. Then you click another icon and need to present credentials again. By 10 A.M., you're probably ready to just

go for coffee and forget about working, since productivity clearly isn't happening if you have to log on every time you start an application.

Single sign-on was introduced in Windows Server 2008, but it was improved in Windows Server 2008 R2 with forms-based authentication. Whereas the previous version allowed you to continue to work without re-presenting your credentials when logging into the same server, the current iteration caches your credentials in a secure web form to present any time you attempt to connect to a RemoteApp program.

Extending Easy Print to Client Platforms and Eliminating .NET Dependency

Printer drivers have long been the bane of the terminal services administrator's life. At first, supporting printer drivers was a gamble in which, if the driver didn't crash the terminal server, you'd won. Supporting client-side printers increased the exposure to error-prone drivers by lessening the administrator's control over the drivers installed. When supporting Windows NT drivers on the terminal servers and non–Windows NT drivers on the client (for example, when using Windows 98 as a client to a Windows 2000 Server terminal server), the drivers might not have the same name. This would require the administrator to create driver mapping files that basically say, "When the system refers to *this* driver from within the client session, *that* driver on the terminal server should be used." Otherwise, the print job would not print.

Over time, the drivers got more reliable as the problem became better understood. When both the client and terminal server were based on Windows NT technology, the driver name mismatch problem ceased to be an issue. Then Windows Server 2003 introduced a new Group Policy that permitted only user-mode drivers by default. This removed the chance of installing a poorly written kernel-mode driver that could crash the server, but it still meant that terminal server administrators had to test, maintain, and support a variety of drivers for both corporate printers and mapped client printers (although some companies stopped supporting mapped client printers just to avoid the driver problems).

Another problem with previous iterations of printing was deciding which printers should be mapped to the remote session. If printer mapping was enabled, then all the client printers would map to the terminal server, regardless of whether this was appropriate. Mapping all these printers could also be time-consuming, not to mention increasing the number of drivers that needed to be installed on a terminal server.

Terminal Services in Windows Server 2008 addressed these problems in several ways. First, and simplest, Group Policy allows administrators to map only the client's *default* printer to a terminal session. Second, Easy Print technology avoids the driver problem for clients running Windows Vista and Remote Desktop Connection 6.1. Basically, Easy Print allows users to print from a remote session without having to install any drivers on the terminal session at all. The remote session gets printer settings from the client and even makes calls to the client-side GUI to show the driver configuration panes for the drivers.

Easy Print had two catches, though: It didn't work when connecting to client operating systems (which eliminated most common VDI scenarios) and it required .NET on the client

operating system to work. In Windows Server 2008 R2, both those limitations are addressed. Whereas .NET is required to convert the XPS of the data stream to the GDI commands required to print, in Windows Server 2008 R2 and Windows 7, the operating system does this.

To learn more about Easy Print, see Chapter 6.

RDS Roles in Windows Server 2008 R2

Users of Terminal Services in Windows Server 2008 will find most of the roles in Windows Server 2008 R2 RDS familiar. RDS is supported by six role services.

- RD Session Host
- RD Virtualization Host
- RD Connection Broker
- RD Web Access
- RD Gateway
- RD Licensing

RD Session Host

The RD Session Host (known as the terminal server in Windows Server 2008) remains the core piece of the Remote Desktop Services architecture for delivering individual applications and for getting the highest user density for full desktops. A RD Session Host server is different from other types of Windows servers in several ways. Fundamentally, a server with this role installed works a lot more like a workstation than a server.

For example, other server roles are designed to serve one general purpose, such as handling email or database queries. Their priorities are clear: Whatever is at the foreground of that server's purpose gets the lion's share of the processor. A shared server is different. Many people are using it at the same time, so it can't just assume that whichever application is in the foreground is the one that should get all the processing time—which foreground of the 40 or so sessions should it pick? Therefore, all user processes on a Remote Desktop Session Host server have the same priority so that they share the processor more or less evenly among all remote users.

> ***NOTE*** **In Windows Server 2008 R2, a new feature called Dynamic Fair Share Scheduling (DFSS) proactively ensures that the scheduler doesn't allocate too much processor time to any single session. This feature is on by default.**

Users connect to an RD Session Host server via the RDP. They make this connection by starting an RDP file that details all the settings for the connection. Users can get to this file from a network share or in email, and it can be automatically generated from a browser or (for clients running Windows 7) the Start menu through RemoteApp and Desktop Connections. When a user starts a remote session, it's protected from other remote sessions running on that computer. Users can't see each other's sessions, and the applications running in those

sessions don't share read/write memory. They can have an impact on each other inadvertently (for example, by using demanding applications that take memory away from other users) but there's minimal security risk in having multiple people running sessions on the same RD Session Host server. To say "no security risk" is, of course, not possible, because there are some exceptional cases that could be exploited by an expert with the right tools, but this is generally true.

BEST PRACTICE **RD Session Host servers have a heavy workload supporting all the remote client sessions, so it's generally best to reserve them only for that use.**

Chapter 2, "Key Architectural Concepts for Remote Desktop Services," talks about how to size an RD Session Host server; information about how to install and set up the role is included in Chapter 3; and how to set up server farms with the RD Connection Broker is covered in Chapter 9.

RD Virtualization Host

Windows Server 2008 R2 introduces a new kind of supported resource: VMs. (VMs, of course, are not new with Windows Server 2008 R2, but support for them within the RDS infrastructure is.) This role service uses Hyper-V to host VMs. VMs can be pooled (generally available to anyone with access to the VM pool) or personal (assigned to a particular user in AD DS).

Why support VMs as well as sessions? The answer is simple: both are valid means of virtualizing the desktop. For higher density, you want sessions: *Many* more people can run sessions on a single computer than can run VMs, because sessions share a lot of basic infrastructure in the operating system (even though they can't see each other). VMs are a virtual manifestation of a physical machine and thus completely separate from each other. This takes many more resources to support. You can run a dozen sessions on a server with 4 GB of RAM and a modern processor, but this same server would have a hard time supporting more than a couple of VMs running at the same time.

NOTE **True story: At one virtualization event, some people said they had heard about virtualized desktops through VMs first. They'd never heard of sessions and were excited by the possibilities of "lightweight VDI."**

The reason why VMs are valuable is related to why they're so resource-intensive: they're a completely isolated environment. A VM is configured with a certain amount of memory and a certain number of processors, reserved for it and not available to other VMs. The operating system is entirely reserved for the use of the VM. That means that whatever happens within the VM does not affect other VMs running on the same physical server. Users can install applications and they will be installed only on that VM. Users can run the most processor-intensive CAD (computer-aided design) software around and they won't drain resources from other VMs. Users can completely misconfigure a VM and cause it to crash, and this will affect only the person currently using it.

In RDS, VMs are often assigned to power users. Those with personal desktops are those who need a complete desktop replacement (albeit one that can be backed up and has all the protection of the data center): those who need to be able to install applications and configure their computers. Personal desktops are also good candidates for applications that require a persistent local data source (that is, they can't store all their data on a network share). Those using pooled desktops are often those who need to run applications that aren't good candidates for virtualization on an RD Session Host for one reason or another—they require a previous version of the browser, are 16-bit (Windows Server 2008 R2 is 64-bit only, and 16-bit applications won't run on that platform), or otherwise just don't fit but will work on a pooled VM.

Chapter 2 covers how to size an RD Virtualization Host server; Chapter 4, "Deploying a Single Remote Desktop Virtualization Host Server," discusses how to set up the role for a single-server installation; Chapter 9 teaches you how to deploy the role in a farm; and Chapter 10 details how to manage larger deployments.

RD Web Access

Remote Desktop Web Access (RD Web Access) integrates with Microsoft Internet Information Services (IIS) to display the icons of authorized RemoteApp programs and VMs in a portal displayed in Internet Explorer and launch the connections. A user authorizes against the portal and can see the icons for all the remote resources allocated to them by the administrator. When he or she clicks an icon, it creates and starts a RemoteApp program in much the same way it would if the RDP file were stored on the user's computer. Using the new forms-based authentication in RDS, after a user authenticates to a portal once, his or her credentials can be used for any resource the user is authorized to access.

When a user starts a RemoteApp program, a session is started on the RD Session Host server that hosts the RemoteApp program, or the VM backing the VM icon. The RD Web Access server does not start the application. As shown in Figure 1-1, it just displays the application icon, creates the RDP file for that application when the user double-clicks that icon (1), and then passes the RDP file to the user to start the application from the RD Session Host (2). RemoteApp programs and desktops started via RD Web Access do not display in the browser but in their own windows (3) and are independent of the browser window. Closing the browser won't disconnect or terminate the connections to the RD Session Host or VM.

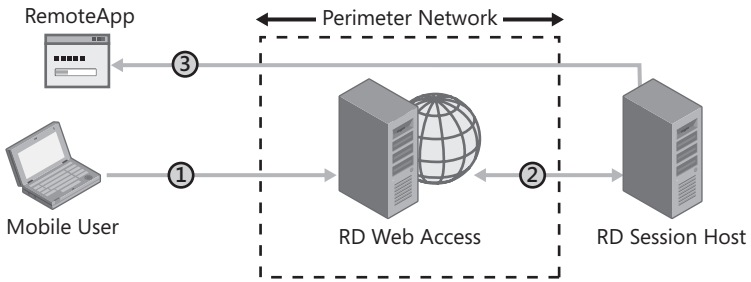


FIGURE 1-1 RD Web Access displays application icons in a browser for the convenience of users.

RD Web Access has many benefits, including the following.

- Users can access RemoteApp programs from a website over the Internet or from an intranet. To start a RemoteApp program, they just double-click the program icon.
- With the new Web SSO feature, after the user authenticates to the website, those credentials are stored and provided for any other connections they initiate—even connections on other servers or other farms.
- RD Web Access can display resources from more than one farm and aggregate them into a single window.
- RD Web Access will display only the resources assigned to a particular person.
- By using RD Web Access, there is much less administrative overhead than that required to maintain and distribute RDP files for connecting to an RD Session Host farm. You can easily deploy programs from a central location and don't have to worry about ensuring that RDP files containing connection information are up to date.
- RD Web Access includes Remote Desktop Web Connection, which enables users to connect remotely to the desktop of any computer where they have Remote Desktop access from the RD Web Access portal.
- RD Web Access works with minimal configuration, but the RD Web Access web page includes a customizable Web Part, which can be incorporated into a customized web page or a Microsoft SharePoint site.

That's how RD Web Access benefits people using a browser . . . but in Windows Server 2008 R2, this role service supports even people connecting without a browser. RemoteApp and Desktop Connections is a new feature in Windows 7 (it's part of the operating system, not the RDP client, so it is not available in previous versions of Windows) that allows RemoteApp and VM icons to be added to a client's Start menu and started from there. The trick is that RD Web Access gets its information about which RemoteApp programs and desktops are available to which users from the publishing service on the RD Connection Broker and makes those resources available through a URL. One URL supports the website you see with a browser, and another supports connections delivered to RemoteApp and Desktop Connections.

Chapter 9 explains how to configure and use RD Web Access and RemoteApp and Desktop Connections.

RD Connection Broker

For the sake of redundancy, it's good practice to have more than one RD Session Host server hosting your remote application set and to load-balance your servers. And it's essentially a given that there will be more than one VM in any deployment using VDI—there might even quite possibly be more than one RD Virtualization Host to run those VMs.

Having multiple endpoints and servers supporting those endpoints allows you to spread out the user load and eliminates the possibility that one server could go down and take out your ability to serve centralized applications. The trouble is that connections are fundamentally made to individual RD Session Host servers, not to groups of them. That is, the final

connection is made to the RD Session Host server named RDSH01 (or whatever other name you've given it).

But if your RDP files include the names of individual RD Session Host servers, the connections won't be load-balanced. Nor will they be flexible enough to determine that a user really should be connecting to another RD Session Host server when starting a new application, because he or she already has an application open there. If you've deployed VMs, it's possible to point an RDP file to a particular VM without making any assignments in Active Directory Domain Services—it's essentially the same thing as using RDP to connect to a physical machine identified by name. But assigning VMs by name doesn't allow you to use pooled VMs. Nor can RDP files automatically wake up a VM that's hibernating and prepare it for the connection. If you attempt to make a direct connection to a hibernating VM, the connection will fail.

HOW IT WORKS

An Introduction to Connection Brokering

The RD Connection Broker role service handles the problem of how to connect user requests for sessions or VMs intelligently to the right endpoint, as shown in Figure 1-2. For RemoteApp connections, RD Connection Broker makes this decision according to several criteria, including

- **Which farm was the incoming request attempting to connect to?**
- **Does the person making the connection request already have an existing (active or disconnected) session on that farm?**
- **If no connection exists, which RD Session Host server has the lowest number of sessions?**

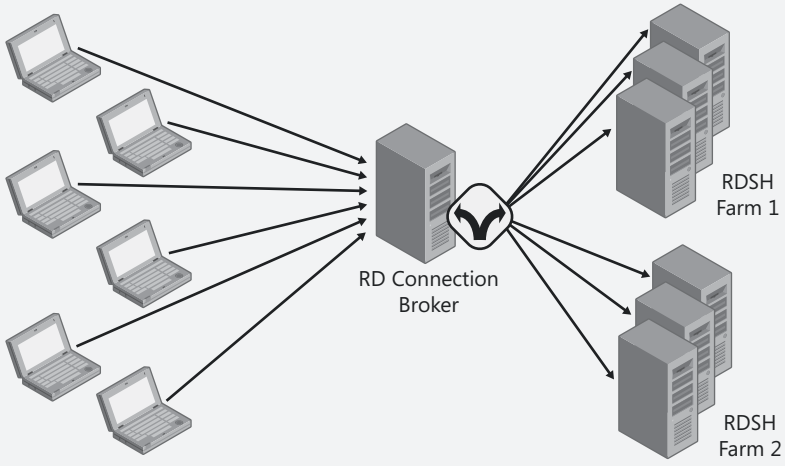


FIGURE 1-2 The RD Connection Broker routes incoming connections to the appropriate RD Session Host server.

For VM connections (see Figure 1-3), the RD Connection Broker makes its decision based on similar criteria.

- **Is the VM request for a personal VM?**
- **If for a pooled VM, does the person requesting already have a disconnected session on a VM?**

If no connection exists, the connection is sent to the RD Virtualization Host server that has the lowest number of currently active VMs, and the RD Virtualization Host server prepares a VM for the connection.

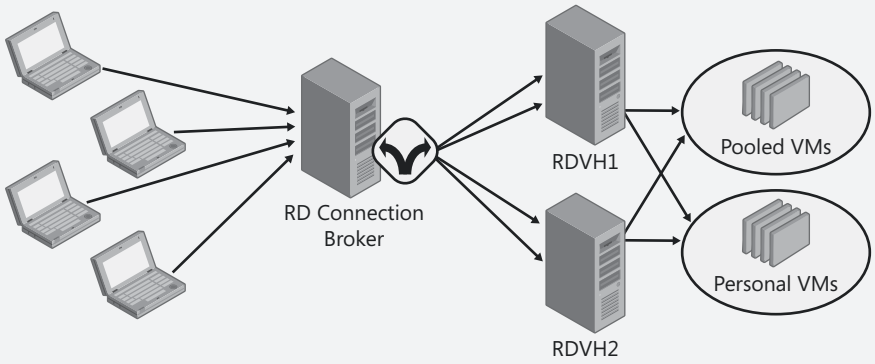


FIGURE 1-3 The RD Connection Broker also brokers connections to VMs on RD Virtualization Host servers.

The RD Connection Broker includes only one form of load balancing—keeping track of how many sessions RD Session Host servers have or how many VMs each RD Virtualization Host is running—but it can be integrated with third-party load balancers that support other criteria such as processor or memory load, time of day, or application.

Chapter 9 explains how to use RD Connection Broker to support RD Session Host farms and pooled and personal VMs.

RD Gateway

In the dark days before Windows Server 2008, if you wanted to connect to a terminal server from the outside world using only the tools in the box, you might have considered opening port 3389 (the port that RDP listens on by default) so that the terminal server could accept incoming connections. Most people didn't do this, however, because of the security hole it opened.

One of the role services of RDS in Windows Server 2008 R2 is Remote Desktop Gateway (RD Gateway). RD Gateway enables authorized remote users to connect to resources on an internal corporate or private network, from any Internet-connected device, whether originally part of

the domain or a public computer or kiosk. As shown in Figure 1-4, the network resources can be RD Session Host servers supporting full desktops or RemoteApp programs, VMs, or computers with Remote Desktop enabled. In other words, people accessing the corporate network from the Internet can use RDP to connect to full desktops, individual applications, or even their own desktop computers—it all depends on what the administrator has set up.

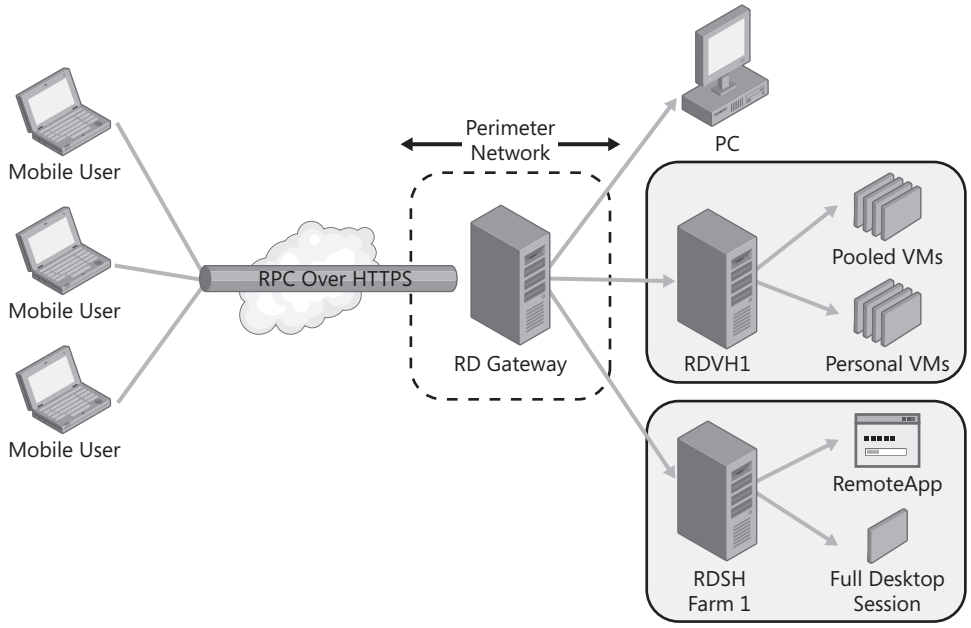


FIGURE 1-4 RD Gateway provides secure access to the corporate network from other networks such as the Internet.

RD Gateway uses RDP over HTTPS to establish a secure encrypted connection between remote users on the Internet and the internal network on which their applications run; this requires only port 443 to be open (which it probably is already for secure Internet connectivity). By doing this, RD Gateway does the following.

- Enables remote users to connect to internal network resources over the Internet by using an encrypted connection, without needing to configure VPN connections.
- Provides a comprehensive security configuration model that enables you to control access to specific internal network resources.
- Provides a point-to-point RDP connection that can be limited, rather than allowing remote users access to all internal network resources.
- Enables most remote users to connect to internal network resources that are hosted behind firewalls in private networks and across Network Address Translators (NATs). With RD Gateway, you do not need to perform additional configuration for the RD Gateway server or clients for this scenario (aside from opening port 443 in the firewall).

The RD Gateway Manager console enables you to configure authorization policies to define conditions that must be met for remote users to connect to internal network resources. For example, you can specify

- Who can connect to RD Gateway (in other words, the users and computers who can connect).
- Which network resources (computers or computer groups) users can connect to.
- Whether device and disk redirection is allowed.
- Whether clients must use smart card authentication or password authentication, or either one.

To enhance security further, you can configure RD Gateway servers and RDC clients to use Network Access Protection (NAP). NAP is a health policy creation, enforcement, and remediation technology included in Windows XP Service Pack 3 (Windows XP SP3), Windows Vista, Windows Server 2008, Windows 7, and Windows Server 2008 R2. Using NAP, system administrators can enforce client computer health requirements, which can include software requirements, security update requirements, required computer configurations, and other settings to connect to RD Gateway.

You can also use RD Gateway server with Microsoft Internet Security and Acceleration (ISA) Server or Forefront Threat Management Gateway (TMG) to enhance security. In this scenario, you can host RD Gateway servers in a private network rather than a perimeter network and host ISA or TMG in the perimeter network. The SSL connection between the RDC client and ISA or TMG Server can be terminated at the Internet-facing server.

The RD Gateway Manager console provides tools to help you monitor RD Gateway connection status, health, and events. With RD Gateway Manager, you can specify events (such as unsuccessful connection attempts to the RD Gateway server) that you want to monitor.

RD Gateway can be used with RDP files stored on clients, with RD Web Access, or with RemoteApp and Desktop Connections. Combined with RD Web Access or RemoteApp and Desktop Connections, you can set up a remote workspace that presents a website with the appropriate application icons and then makes sure that the person connecting or the computer he's connecting from meets the RD Gateway rules.

RD Gateway uses few resources and if sized properly can support hundreds of incoming users, so it can safely be combined with other roles that might be in the perimeter network.

RDS Licensing

The RDS Licensing role service is responsible for keeping track of who has a license to use the RD Session Host servers. Not who's *authorized* to use the RD Session Host server—AD DS user rights or RD Gateway makes that call, depending on what level the administrator is authorizing this connection. RDS Licensing is the license management system that enables RD Session Host servers to obtain and manage RDS client access licenses (RDS CALs) for devices and users that are connecting to an RD Session Host server.

NOTE **RDS Licensing supports previous versions of terminal servers as far back as Windows 2000 Server. Also, the operating system supports two concurrent connections to administer a computer remotely, so you do not need a license server for these connections.**

RD Session Host servers can be configured to require either per-user or per-device RDS CALs. You'll learn more about the details of RDS Licensing in Chapter 12, "Licensing Remote Desktop Services," but the basic story is this: Each RD Session Host server determines if the user or the computer connecting to it has a valid license. If it does (and the user has permission to log on), then the RD Session Host server grants the connection. If it does not, then the RD Session Host server attempts to contact a license server to see if a license for that device or user is available. The license server then either allocates a license to the device (per-device RDS CAL) or edits the properties of the user's account in AD DS to show that a license has been used (per-user RDS CAL). If the RD Session Host server cannot connect to an RDS Licensing server, it will issue a temporary license if the RD Session Host server is within its grace period. Access will be granted for up to 120 days.

Servers supporting the RDS Licensing role maintain a database that tracks how RDS CALs have been issued. For per-device RDS CALs, the license is assigned to a computer. For per-user RDS CALs, the license is not actually assigned but its usage is registered in AD DS and can be tracked.

RD Licensing is a low-impact service, requiring very little processor time or memory for regular operations. Memory usage is less than 10 MB. Its hard disk requirements are small, even for a significant number of clients: The license database grows in increments of 5 MB for every 6,000 RDS CALs issued. The license server is active only when an RD Session Host server is requesting an RDS CAL, and its impact on server performance is very low, even in high-load scenarios. Therefore, in smaller deployments, the RDS Licensing role service can be installed on the same computer as the RD Session Host role service. In larger deployments, the RD Licensing role will often be on a separate computer.

Although only accessing the RD Session Host role will trigger the consumption of an RDS CAL, using any part of the RDS infrastructure requires an RDS CAL (or, for VDI-only deployments, a VDI CAL).

How Other Services Support RDS

The RDS role doesn't exist in a vacuum. Several roles help to support the various role services of RDS, and without them, the solution doesn't work. In addition to the core RDS role services and their relationship with each other, it's important to understand their relationship with other Windows Server roles. This section covers these roles and how they support RDS functionality.

What are the roles and how do they fit together? How do they fit with the other non-RDS parts of the Windows infrastructure (Hyper-V, IIS, certificates, and AD DS, among others)?

The Client Connection

Yes, it might be obvious, but it's still worth looking at: The way the client interacts with the role services of RDS defines what the user experience to a particular endpoint will be.

Whether the endpoint is a session on an RD Session Host server, a VM hosted on RD Virtualization Host, or even a physical machine, the fundamental relationship between client and endpoint has three parts: the RDC client, the RDP connection, and the endpoint.

- The RDC client component initiates the connection to the endpoint and receives the data that the server sends to it.
- The server component on the endpoint interacts with the core operating system and takes the information received (for example, sounds being produced, bitmaps being displayed), converts it to RDP commands, and serializes it to be passed to the client.
- The protocol enables the connection between the client and the endpoint; it defines the kind of information that is passed between them via virtual channels.

NOTE **Why the distinction between RDP and RDC? RDP is the Remote Desktop Protocol, the protocol that passes user input and application output between client and server. RDC is the Remote Desktop Connection, the client component that initiates and manages the RDP connection.**

In short, the client requests the connection, the endpoint formats the calls to the applications and operating system in a way that the client (or server, depending on which way the information flow is going for a particular transaction) can understand, and RDP passes the right information that lets the user communicate with the applications on the server as though they were running locally.

This communication relies on *virtual channels*, bi-directional connection streams provided through RDP. They establish a data pipe between the RDC client and the endpoint to pass specific kinds of information, such as device redirection or sound, between client and server. Virtual channels are a way to extend the functionality of RDP that's been available since Windows 2000 Server, and they are also used by some features of RDS, such as device and sound redirection.

But a lot has changed since Windows 2000 Server, and one of the components that's changed is that the 32 static virtual channels originally made available with RDP 5.1 aren't enough anymore. More kinds of data are now available, and it's clear that there might be more not yet considered. In addition, static virtual channels had a problem: They were created at the beginning of the connection and torn down at the end. If you added a device during the session, it couldn't use virtual channels unless you terminated the connection and then reconnected.

IMPORTANT **Terminating a connection ends it completely on the server. A disconnected session still exists on the server and a user can reconnect to it**

Therefore, RDS supports *dynamic virtual channels*, virtual channels that the client creates on demand and then shuts down when it's done with them. If you're curious about the interfaces to make dynamic virtual channels work for you (or how they work at all), see the PDF titled "Functionality for RDS Scripters and Developers" on the companion CD.

Hosting VMs

For some time, it has been possible to virtualize Terminal Services roles, but Hyper-V was not a required component of a Terminal Services deployment. In RDS, Hyper-V is required to use the VM hosting feature.

Hyper-V is installed automatically if you choose to install the RD Virtualization Host Role service. Because RD Virtualization Host requires Hyper-V, it is the only RDS role service that cannot be virtualized.

Authenticating Servers with Certificates

Although you don't need a Certificate Authority (CA) server to use RDS, you will definitely need certificates from somewhere.

One of the curious things about RDS is the trust required between client and server. Obviously, the server has to trust the client, since the server is a partial porthole to the corporate network. But the client has to trust the server as well. The client is providing the user name and password for the corporate network, so it's important that the server the client is connecting to is a legitimate endpoint and not a rogue server set up to steal logon credentials.

To ensure that an endpoint's identity can be trusted, you can install a certificate on the server and on the client. To do this, you'll need to get certificates from your own in-house PKI solution, or you'll need to purchase certificates from a public CA.

> ***IMPORTANT*** **All RD Session Host servers in the same farm must use the same certificate for certificate-based authentication.**

Certificates are also used to

- Authenticate the identity of an RD Gateway server and allow it to set up a secure channel with the client.
- Sign RDP files
- Provide HTTPS access to the RD Web Access website

Enabling WAN Access and Displaying Remote Resources

Two components of RDS require IIS: RD Web Access and RD Gateway. RD Web Access's need for IIS is pretty apparent: It provides information about the RemoteApp programs and desktops available to a user through two URLs. One URL supports display for RD Web Access and one supports RemoteApp and Desktop Connections.

IIS is also required for RD Gateway. RD Gateway encapsulates RDP traffic over HTTPs, so it requires certain components of IIS.

IIS is installed automatically when you install an RDS role service that requires it.

Updating User and Computer Settings

It's such an obvious choice to use AD DS for a support role that you might not have thought of it, but it's crucial to a functioning centralized computing infrastructure in several ways—not all of which you might have expected. AD DS manages

- The group policies that configure RD Session Host servers and the user sessions running on them.
- Whether or not a user has the right to connect to an RD Session Host server.
- The process of showing that a user has consumed a per-user RDS CAL.

Functionality for RDS Scripters and Developers

It's crucial to understand that RDS is not just a product—although it's definitely that—but it's also a development platform for both independent software vendors (ISVs) and consultants creating custom solutions. Windows Server 2008 added a lot of new APIs for partners, and Windows Server 2008 R2 adds even more. Although a description of how to use all of these APIs is beyond the scope of this book, information available on the companion media highlights some of the platform extensions available to RDS partners through public interfaces.

ON THE COMPANION MEDIA **For a detailed description of the RDS API, please see "Functionality for RS Scripters and Developers" on the companion media. Detailed instructions for using this API are on MSDN.**

NOTE **Public interfaces (also known as APIs) are interfaces that are, well, publicly available and documented on MSDN so that developers can use them. Private interfaces are not documented. The main difference is supportability. A private interface might change at any time if required by the people who developed it (in this case, Microsoft). An API won't change without notice. Even if you had the option to build solutions based on private interfaces, it would be better to build on the public APIs than on private ones.**

Summary

This chapter introduced you to RDS in Windows Server 2008 R2. At this point, you should understand

- How this role has developed since it became part of Windows 10 years ago.

- What RDS is used for.
- The new business cases that Windows Server 2008 R2 RDS now supports.
- The RDS roles that support these new business cases and how they interact.
- How other Windows roles (and the client) support RDS functionality.
- How RDS is a development platform and some of the functionality that scripters and developers can add to it.

In Chapter 2, you'll find out how Windows architecture supports RDS.

Additional Resources

These resources contain additional information and tools related to this chapter.

- To learn more about some fundamental concepts of the operating system that affect RD Session Host and RD Virtualization Host functionality (and sizing), see Chapter 2, "Key Architectural Concepts for Remote Desktop Services."
- To learn how to set up an RD Session Host server, see Chapter 3, "Deploying a Single Remote Desktop Session Host Server."
- To learn how to set up an RD Virtualization Host server to support pooled VMs and personal desktops, see Chapter 4, "Deploying a Single Remote Desktop Virtualization Host Server."
- To learn how to set up user profiles with RDS, see Chapter 5, "Managing User Data in a Remote Desktop Services Deployment."
- To understand how RDP integrates the client and server operating systems for display, printing, and audio and device redirection, see Chapter 6, "Customizing the User Experience."
- To learn how to lock down the user environment with Group Policy, see Chapter 7, "Molding and Securing the User Environment."
- To learn how RDP connections are secured for LAN connections, see Chapter 8, "Securing Remote Desktop Protocol Connections."
- To learn how to use RD Connection Broker to deploy a farm of RD Session Host servers or a pool of RD Virtualization Host VMs, see Chapter 9, "Multi-Server Deployments."
- To learn how to publish resources to RD Web Access and RemoteApp and Desktop Connections, see Chapter 10, "Making Remote Desktop Services Available from the Internet."
- To learn how to use RDS on the Internet, see Chapter 10, "Making Remote Desktop Services Available from the Internet."
- To learn how to manage sessions on an RD Session Host server, see Chapter 11, "Managing Remote Desktop Session Host Sessions."

- To learn how RDS licensing works and how to use an RD License server, see Chapter 12, "Licensing Remote Desktop Services."
- To learn about RDS life-cycle management, see Chapter 13, "Life-Cycle Management for Remote Desktop Services."
- For more details on the APIs available to developers, see the RDS Reference at *http://msdn.microsoft.com/en-us/library/aa383494(VS.85).aspx* or, for longer documents and source code, see the RDS Code Gallery site at *http://code.msdn.microsoft.com/rdsdev*.
- For in-depth developer resources (including code samples and detailed documents), see the RDS team Code Gallery site at *http://code.msdn.microsoft.com/rdsdev*.

CHAPTER 5

Managing User Data in a Remote Desktop Services Deployment

Thus far in this book, you have learned how to set up a single Remote Desktop (RD) Session Host server or a simple Microsoft Virtual Desktop Infrastructure (VDI) deployment. Those deployments aren't yet production-ready, though: No applications are available, the connections aren't secured, you haven't yet defined the devices and experience to redirect, and the profiles and Folder Redirection aren't yet set up.

Properly configured profiles and Folder Redirection go a long way toward a good user experience for users working via remote connection to the data center. Because profiles weren't originally designed for remote work environments, this can sometimes be tricky. Remote Desktop Services (RDS) independent software vendor (ISV) partners have developed some products to help make a highly flexible system for complex environments. This chapter, however, shows you how best to configure profiles and Folder Redirection using the tools that come with Windows.

The basic elements of a user workspace are the configuration settings in the user's profile and the default locations to save data. After reading this chapter, you will understand the following.

- How roaming, local, and mandatory profiles work
- Why virtualization can complicate implementing profile strategies
- Best practices for storing and managing profiles
- How to use Folder Redirection to unify user default locations between local and remote applications

- The benefits and drawbacks of using mandatory profiles to maintain a consistent look and feel
- How to secure the desktop to prevent users from saving files to it and why this is important
- How to support profiles across servers running both Windows Server 2008 R2 and Windows Server 2003, or Windows 7 and Windows XP virtual machines (VMs)

How Profiles Work

A *profile* is a collection of settings and documents that define a user's work environment, sometimes referred to as a user's "personality." A user's profile includes both configuration data and personal data such as documents and pictures. Personal data in the profile can be stored on the desktop or in one of the folders associated with the user account (for example, My Documents). The profile also includes user specific settings, such as the following.

- Changes that you make to application layouts, such as adding buttons, changing the layout, and adding a default signature
- Changes to system settings that are unique to the user experience, such as changing your desktop background, screen saver, and keyboard layout

Machine-wide settings such as firewall settings are *not* stored in the user profile.

Documents and supporting files that are part of your profile are stored in a unique user profile folder (and subfolders). Local and roaming profile settings are stored as a single file (called NTUSER.DAT), not as a collection of individual settings. NTUSER.DAT is stored in the root of each user's profile folder. Mandatory profile settings are stored in NTUSER.MAN; this file can be shared among multiple users because it is read-only.

> ***NOTE*** **Super-mandatory profiles label the folder where they're stored with the .man suffix, like this: *//servername/sharename/*mandatoryprofile.man/. Super-mandatory user profiles are similar to normal mandatory profiles except that users with super-mandatory profiles cannot log on when the server that stores the mandatory profile is unavailable. Users with normal mandatory profiles can log on with the locally cached copy of the mandatory profile. Use super-mandatory profiles only when you want to have absolute control of the user profile—so much so that you can't take the chance that a cached copy might be out of date.**

While a user is logged in, the NTUSER.DAT file is loaded temporarily into HKEY_CURRENT_USER (HKCU) in the registry of the computer that user is logged on to; the documents are stored in the subfolders within the profile folder, as shown in Figure 5-1. You will find out in detail about the parts of a profile—both the registry and the data folders—later in this chapter. But first let's examine the different types of profiles.

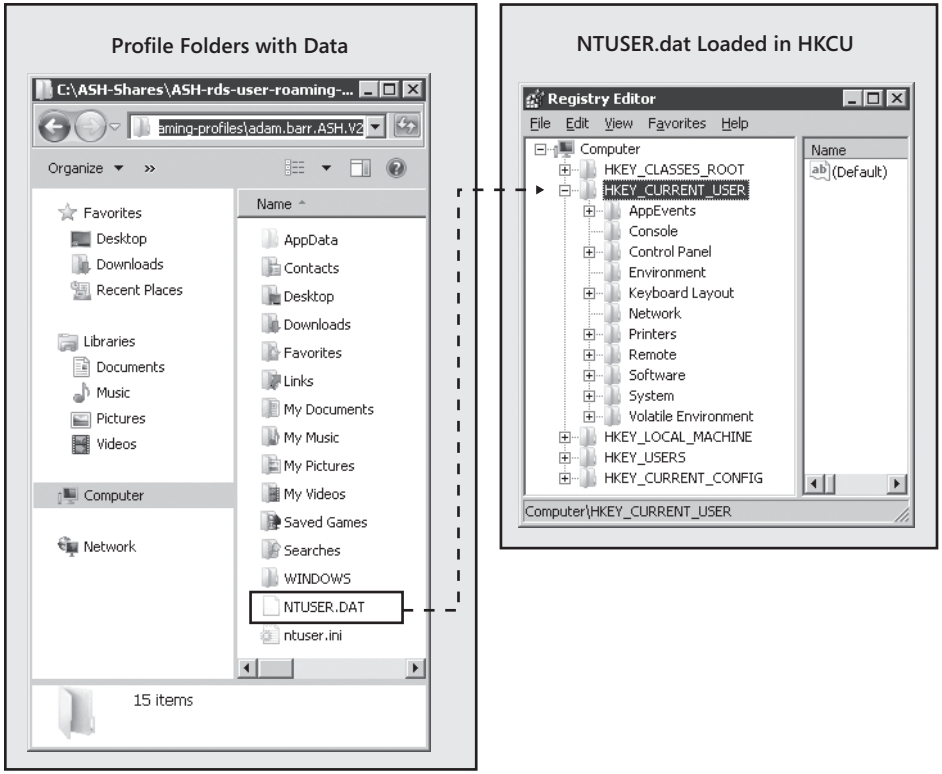


FIGURE 5-1 The user profile contains personal settings and data such as folders and the user-specific registry settings.

Types of Profiles

As alluded to in the previous section, there are three types of profiles: local, roaming, and mandatory. Local profiles are stored on and used from a single computer and store data in NTUSER.DAT. Roaming profiles are stored on and used from a network share, so they're available to any computer that can access that particular network share. They also store data in NTUSER.DAT. Mandatory profiles are often centrally located like roaming profiles, but whereas local profiles and roaming profiles are read-write, mandatory profiles are read-only. They store their settings in NTUSER.MAN.

Local profiles are usually fast to load because they are stored on the computer the user is using. When a user logs on, the local profile will load from its local location on the hard drive and populate HKCU. When the user logs off, the contents of HKCU (including any changes that the user made) will be written back to the local hard disk and overwrite the previous version of the file.

NOTE **Local profiles aren't a good fit for most remoting scenarios because they're stored on a single computer. Personal desktops and single RD Session Host server deployments are possible exceptions to this, but pooled VMs and RD Session Host sessions in a farm larger than one server will quickly find that local profiles lead to an inconsistent user experience. This is because the user would have a unique local profile on each machine she logs onto.**

Roaming profiles afford the most flexibility in a remoting environment because they're stored in a central location accessible to all VMs and RD Session Host servers. They're also read-write, so users can adjust their settings. When a user logs onto a session or VM (or a computer, for that matter), the roaming profile will load from its network location and populate HKCU in the registry. When the user logs off, the contents of HKCU (including any changes that the user made) will be written back to the network location and overwrite the previous version of the file.

Mandatory profiles are loaded to HKCU when a user logs on, just like a roaming profile, but they aren't written back to their storage location at logoff—all changes to the profile are just discarded.

How Profiles Are Created

A user does not start with a user profile. The profile is created the first time that a user logs onto a machine. Mandatory profiles are the exception to this, and even the mandatory profile, which is used by multiple people, has to initially come from somewhere. To fully understand profiles, you need to know how profiles are initially created. This will come in handy later in this chapter, when you learn how to create a mandatory profile and also how to customize a default profile.

All profiles are created from a "default profile." Each RD Session Host—actually, every computer—has a local default user profile (located at C:\Users\Default in Windows Vista and later) for this purpose. Depending on which type of profile will be used and how you have implemented the profile strategy, the process of making user profiles varies slightly.

If your users will use local profiles (for instance, if you have only one RD Session Host), new user profiles will be created by making a copy of the local default profile located on the computer that the user logs on to. This copy will go into a new folder labeled by the login name of the user.

If your users will use roaming profiles, when a new user logs on to a server for the first time, a new profile is created for him by making a copy of a default user profile. Domain joined computers will first look for a network default user profile (stored in the netlogon share on a domain controller and replicated to other domain controllers). If it does not find one in the network share, then it will use the local default profile located on the computer to which the user logged on.

User Profile and the Registry

The registry is organized into sections called *keys,* which align with a particular configuration option. For example, computer-wide settings are stored in HKEY_LOCAL_MACHINE (HKLM), whereas user-specific settings are stored in HKEY_CURRENT_USER (HKCU). As with all versions of Microsoft Windows NT since it was first released, Windows Server 2008 R2 and Windows 7 maintain user-specific settings in HKCU for each user logged on to the computer.

You can see how HKCU works and reflects changes to the user environment by following the process outlined in the following How It Works sidebar, "Observe How Changes to the Environment Are Reflected in the Registry."

HOW IT WORKS

Observe How Changes to the Environment Are Reflected in the Registry

One easy way to watch how HKCU changes as you customize your environment is to make a change and watch the contents of the registry, as follows.

1. Run Regedit.exe and confirm that you want to run it when prompted.
2. Navigate to HKCU\Control Panel\Colors\ and look at the value of the Window key. If you're using the default Windows 7 color scheme, the value of this entry should be 255 255 255. (Full saturation of red, blue, and green values show up as white on a monitor. Values of 0 for all three show up as black. If you ever studied color theory, this is a demonstration that black is the absence of color.)
3. Right-click the Desktop and choose Personalize from the context menu to open the Personalization window.
4. Click Window Color And Appearance. In the Appearance Settings dialog box, click Advanced to open the aptly named Advanced Appearance dialog box. From here, select Window from the Item drop-down list. Change Color 1 to light gray and click OK.
5. Click OK in the Appearance Settings dialog box. The screen will adjust for a moment, and then the background color of windows will turn light gray.
6. If you examine the value of HKCU\Control Panel\Colors\Window, you'll see that it's now 192 192 192.

In Windows Server 2008 R2 and Windows 7, HKCU contains the subkeys described in Table 5-1. Even if you're logging on to a Windows Server 2008 R2RD Session Host server from an earlier operating system such as Windows XP, the profile in the RD Session Host session corresponds to the server platform. These are still the registry keys that apply to the session, not the client computer operating system. There might be additional subkeys in this section; it depends on which applications you have installed. For example, if you install Microsoft Outlook, you'll see an Identities key.

TABLE 5-1 Subkeys of HKCU in Windows 7 and Windows Server 2008 R2

SUBKEY	DESCRIPTION	MAPS TO
AppEvents	Sounds played on system events.	Control Panel\Sounds
Console	Command window settings such as window size, colors, and buffer size.	Command Prompt\Properties
Control Panel	User desktop appearance settings, mouse and keyboard settings, power policy, and accessibility.	Control Panel
Environment	Environment variable definitions.	Control Panel\System\Advanced
EUDC	Customized characters that users install for viewing and printing documents when standard fonts don't support them. Applies to East Asian font sets.	Control Panel\Fonts
Keyboard Layout	Edits the keyboard layout. Useful if your operating system is displaying in one language but you want to use the keyboard layout of another one (for example, displaying in English but arranging the keyboard as though you were in Germany).	Control Panel\Regional and Language Options
Network	Network drive mappings and settings.	Control Panel\Networks
Printers	Printer connection settings.	Control Panel\Printers
Remote (Remote Access in Windows 7)	Contains settings to be applied to remote sessions (for example, ClearType or wallpaper) for each session. The subkey corresponds to the Session ID.	
Session Information	Information about the current session, such as how many applications are open.	Not stored—populated during the session
Software	Personal settings for all software installed for that user.	Individual applications
System	Contains the current control set for that user (drivers and services to run at startup).	Not stored—populated on startup
Volatile Environment	Environment variables for the current logon session.	Not stored—populated for each session

Data is stored in HKCU only for the duration of the session, while data stored in HKLM persists until the reboot. Most pieces of the registry are saved in files called *hives* and are loaded as necessary. When a hive file is opened, it's reloaded into the registry. Therefore, HKCU is stored as a hive in a file called NTUSER.DAT that is loaded at user logon. Each user logged on to an RD Session Host server sees his or her own version of HKCU.

How does this data get loaded? When you log on to a computer, the User Profile Service loads the hive file from the location specified in your user account properties and populates HKCU for that session. When you log off the computer, the hive file is written back to its storage location as NTUSER.DAT. If you happen to be logged on to more than one computer at a time, two copies of your profile will be open, populating the contents of HKCU on each computer.

> ***NOTE*** **Profiles can be cached on the server to speed up logons if you set the corresponding Group Policy. However, even if you enable caching, when a user logs off the RD Session Host server, the corresponding branch of HKCU is cleared. You'll find out more about caching user profiles in the section entitled "Caching Roaming Profiles" later in this chapter.**

In addition to loading HKCU with the contents of your profile, logging on to an RD Session Host server updates two parts of HKLM, the computer-wide section of the registry. HKLM\Software\Microsoft\Windows NT\CurrentVersion\Profile List (Figure 5-2) contains a list of all profiles cached on the computer. It also lists the profiles used by the System account, Network Service account, and the Local Service account. As you can see, machine accounts have profiles just like user accounts do.

The users are identified by security identifiers (SIDs), but you can distinguish them by browsing the keys. The values show the path to both the local cache (the ProfileImagePath key value shown in Figure 5-2) and to the roaming profile folder share (the CentralProfile key value shown in Figure 5-2), so it's not hard to map user names to profiles.

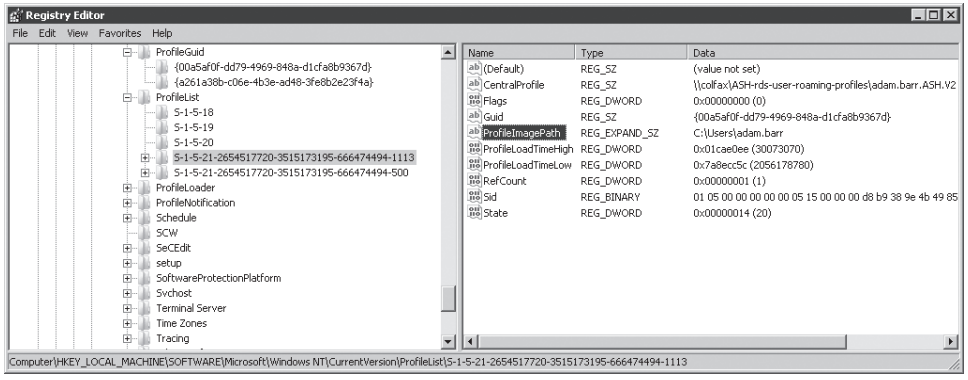


FIGURE 5-2 Loading a profile into a remote desktop session updates the Profile List key for the entire RD Session Host server.

When you log off an RD Session Host server, the two keys with your SID are locked. They don't actually go away, but if you attempt to open the key associated with a user who is currently logged off, you'll get an error message telling you that the system cannot find the file specified. Log on again, and the key with the same SID will be repopulated.

Although loading a profile adds two keys to the registry that never go away, most of the time it doesn't matter. As discussed in the section entitled "The Consequences of Deleting a Profile Folder from Windows Explorer" later in this chapter, it *does* matter should you choose to delete a profile. Deleting the file doesn't delete the registry keys associated with it. Therefore, always use the correct tools to delete profiles; otherwise those users won't be able to load their profiles properly when they log on again.

How Profile Changes Are (Not) Merged

The operating system loads the contents of NTUSER.DAT into HKCU at logon and saves back to NTUSER.DAT at logoff, in the same way that you might open a Microsoft Word document when you log on, type in it for a while, and then save the document when you log off. This has some important implications for a remote environment.

As an example, imagine this scenario: You are logged on to two different computers and you open a new Word document in each session. In Session 1, you type "Every Good Boy Does Fine." In Session 2, you type "All Cows Eat Grass." You save the file in Session 1 as Myfile.docx. Next you save the file in Session 2 as Myfile.docx in the same location, confirming that you want to overwrite the old file when prompted.

The next time you open Myfile.docx, the file will say only "All Cows Eat Grass." The phrase "Every Good Boy Does Fine" has been overwritten. In short, the files are not merged; they're written back to the save location, and the version last written to that location is the only one you'll see.

So it is with profiles, which are just another type of file. If you log on to two sessions, each of which is using the same roaming profile, you will have two copies of your profile open. If you make changes to the open profile, you'll see them at the time, but they won't be saved into NTUSER.DAT until you log off. (Unlike the Word .docx file, the file system won't ask if you want to overwrite the profile file.) As in the previous example, if you have a profile open in Session 1 and in Session 2, log off Session 1 and then log off Session 2, only the changes made to the Session 2 copy of the profile will appear when you log on again and reload that profile. The only difference from the document scenario is that the operating system won't ask you if you want to overwrite the previous version.

CAUTION **One implication of the way profiles work is that you shouldn't use the same profile for local sessions and remote sessions. If you do, then by definition, every time you log on to your computer and then log on to an RD Session Host server, you will be opening two copies of your profile. You will almost certainly lose profile data this way.**

You might be wondering whether opening two RemoteApp programs from a single RD Session Host server opens one or two copies of your profile. The answer depends on the version of Windows Server hosting the session, and how you're starting the applications. On a terminal server running Windows Server 2003, you could create a Remote Desktop Protocol (RDP) session that would open a single application instead of displaying the entire desktop. (As noted in Chapter 1, "Introducing Remote Desktop Services," not many people did this because the experience wasn't very user-friendly, but it was possible.) If you presented individual applications this way, then each time a user opened an application on the same server, he would open a separate session and therefore a separate copy of the profile.

Windows Server 2008 improved on this design in two ways. First, it introduced RemoteApp programs. All RemoteApp programs started from the same server by the same user account run in the same session, so they open only a single copy of your profile. Second, when deciding where to route incoming connections to an RD Session Host server farm, the RD Connection Broker will check to see if a user already has an open session on an RD Session Host server in the farm. If it does, then the user will be routed to the same session to start the application. So, what is the result? You have preference to the server where you already have an open connection, *and*, so long as you're connecting to only a single server, only one copy of the profile will be open because all RemoteApp programs will run in the same session.

Profile Contents External to the Registry

Not all parts of a profile are stored in HKCU. The same folder that contains the NTUSER.DAT file also contains other folders that contain user data as well as application-specific data. In Windows Vista and Windows Server 2008, the profile includes the folders listed in Table 5-2. (More folders might be available, depending on which applications you have installed.)

TABLE 5-2 Folders Associated with a Windows 7 or Windows Server 2008 R2 Profile

FOLDER	DESCRIPTION
AppData	Default root location for user application data and binaries.
Contacts	Used to store contact information and is also the address book for Windows Mail, the successor to Microsoft Outlook Express (Windows Mail is not included in Windows 7 or Windows Server 2008 R2).
Desktop	All items stored on the desktop, including files and shortcuts.
Documents	Default root location for all user-created files (spreadsheets, text documents, and so on).
Downloads	Default location for all files downloaded using Windows Internet Explorer.
Favorites	Bookmarked Uniform Resource Locators (URLs) in Internet Explorer.
Links	File and folder shortcuts; these show up under the Favorites menu on the left side of an Explorer window.
Music	Default root location for all music files.

Continued on the next page

FOLDER	DESCRIPTION
Pictures	Default root location for all image files.
Saved Games	Default location for saved games.
Searches	Default location for saved searches performed from the Search Programs And Files input box on the Start menu.
Videos	Default root location for all video files

Beginning in Windows Vista and Windows Server 2008, the profile structure changed from Windows XP and Windows Server 2003. (Windows 7 and Windows 2008 R2 retain this new profile structure.) The new structure uses more folders to organize the data.

Notice that Windows XP and Windows 2003 were not mentioned in Table 5-2. This is because profiles have evolved over time and the structure of profiles has changed. Windows XP and Windows Server 2003 profiles are called version 1 (V1) profiles; profiles using the structure of Windows Vista and Windows Server 2008 and later are called version 2 (V2) profiles. A V2 user profile folder is distinguished from its predecessors by an added .V2 extension.

Version 2 profiles generally use more folders than those of Windows XP, but V1 top-level folders such as NetHood and PrintHood were moved inside the AppData folder beginning in Windows Vista. Table 5-3 (adapted from the Microsoft document "Managing Roaming User Data Deployment Guide" located at *http://technet.microsoft.com/en-us/library /cc766489(WS.10).aspx*) shows the differences in the default root profile folder structure between V1 and V2 profiles.

TABLE 5-3 Profile Folder Structures of V1 and V2 Profiles

V2 PROFILE FOLDERS (WINDOWS VISTA AND LATER)	V1 PROFILE FOLDERS (WINDOWS XP AND WINDOWS SERVER 2003)
Now AppData\Roaming	Application Data
Contacts	Not Applicable
Desktop	Desktop
Downloads	Not Applicable
Favorites	Favorites
Links	Not Applicable
Documents	My Documents
Music	In My Documents
Pictures	In My Documents
Videos	Not Applicable
Saved Games	Not Applicable

V2 PROFILE FOLDERS (WINDOWS VISTA AND LATER)	V1 PROFILE FOLDERS (WINDOWS XP AND WINDOWS SERVER 2003)
Searches	Not Applicable
Tracing	Not Applicable
Now in AppData folder	My Recent Documents
Now in AppData folder	NetHood
Now in AppData folder	PrintHood
Now in AppData folder	Send To
Now in AppData folder	Start Menu
Now in AppData folder	Templates
Now in AppData folder	Local Settings
Now in AppData folder	Cookies

As you might have noticed in Table 5-3, the Local Settings folder from V1 profiles does not exist in V2 profiles, and many V1 profile folders are now consolidated under the AppData folder in V2 profiles. Why does this reorganization of data matter?

One big accomplishment of the V2 profile reorganization is that machine-specific data is now separated from user-specific data. V1 profiles kept machine-specific and user-specific data scattered through the profile. V2 profiles sort this data and do a better job of separating user-specific data from data that is either too large to roam with the user or is specific to a particular machine and therefore should not roam.

In V2 profiles, the AppData folder now has three subfolders that separate this kind of data.

- **AppData\Roaming** Data that is user-specific and should roam with the user profile
- **AppData\Local** Data that is either machine-specific or too large to roam with a user's profile folder, for example, an Outlook .OST file
- **AppData\LocalLow** Data for "low-integrity" apps (such as browser-based apps) to store data

Table 5-4 (which was adapted from the Microsoft "Managing Roaming User Data Deployment Guide") shows where certain V1 profile data is stored in the V2 profile structure.

TABLE 5-4 Data Storage Reorganization from V1 to V2 Profiles

V2 PROFILE DATA LOCATIONS	V1 PROFILE DATA LOCATIONS
...\AppData\Local	Local Settings\Application Data
...\AppData\Local\Microsoft\Windows\History	Local Settings\History
...\AppData\Local\Temp	Local Settings\Temp
...\AppData\Local\Microsoft\Windows \Temporary Internet Files	Local Settings\Temporary Internet Files

Continued on the next page

V2 PROFILE DATA LOCATIONS	V1 PROFILE DATA LOCATIONS
...\AppData\Roaming\Microsoft\Windows\Cookies	Cookies
...\AppData\Roaming\Microsoft\Windows \Network Shortcuts	NetHood
...\AppData\Roaming\Microsoft\Windows \Printer Shortcuts	PrintHood
...\AppData\Roaming\Microsoft\Windows\Recent	Recent
...\AppData\Roaming\Microsoft\Windows\Send To	Send To
...\AppData\Roaming\Microsoft\Windows\Start Menu	Start menu
...\AppData\Roaming\Microsoft\Windows\Templates	Templates

NOTE **The "Managing Roaming User Data Deployment Guide" is available at** ***http://technet.microsoft.com/en-us/library/cc766489%28WS.10%29.aspx.***

Because V1 profiles and V2 profiles are so different, you can't use the same profiles for Windows Server 2008 R2 RD Session Host servers that you did for terminal servers running Windows Server 2003or Windows XP VMs. The structures of the profiles don't match.

You'll learn later in this chapter how to allow Windows Server 2003 and Windows Server 2008 profiles to coexist. (See the section entitled "Sharing Folders Between Windows Server 2003 and Windows Server 2008 Roaming Profiles" later in this chapter.) This is important both for supporting mixed deployments of terminal servers running Windows Server 2003 and Windows Server 2008 R2 RD Session Hosts, and for supporting Windows 7 VM pools and Windows XP VM pools. (The changes to the profile structure between the operating systems are one reason why you should not combine Windows 7 and Windows XP VMs in the same pool.)

Introduction to Folder Redirection

Although these data folders are stored by default in the user's profile folder, they don't have to be. In fact, in most cases, it's best if some of them aren't. Here's why.

First, keeping user data within the profile folder increases the profile size. Assuming that you're storing profiles on a central share instead of on individual RD Session Host servers (and, for reasons you'll see shortly, this is a good assumption), this can slow logons. A large profile increases the time that it takes for users to log on and log off (because the data in the profile must be cached on the RD Session Host server). In Windows Server 2008 R2, if the profile cache on a server exceeds the quota allocated to the profile cache, it will delete the most recently used profiles, but there's still no reason to fill the cache with user data.

Second, if you're using mandatory profiles and you don't redirect folders outside the profile folder, users will not be able to save files to the standard personal folders such as Documents. The files will look like they're saving, but they won't be retained. This will cause users a great deal of grief and bring you many unsolvable calls to the Help desk.

> ***NOTE*** **The Recycle Bin is a hidden file in the root of the profile folder. You can't redirect it, and even if you're using mandatory profiles, you will still be able to send files to the Recycle Bin.**

The third reason applies to VMs, whether pooled or personal. In the case of a personal desktop, saving files locally preserves them, but it complicates file restore because the files are stored in the VM. To restore the files saved on the local VM, you'd need to restore the VM from backup. Saving the files separately makes it easier to restore them, and the easiest way to do that is to enable Folder Redirection. In the case of *pooled* VMs, Folder Redirection is essential. As with mandatory profiles, saving files to local folders on a pooled VM can lead to lost data. As discussed in Chapter 4, "Deploying a Single Remote Desktop Virtualization Host Server," the most common configuration for pooled VMs is to roll back changes at user logout so the VM remains pristine. That rollback means that any documents saved to the VM would be lost. (Some ISV solutions actually delete the VM on each use and re-create it, which has the same effect.)

For these reasons, it's good practice to use Folder Redirection with RDS, whether connecting to VMs or sessions. You'll learn how to do this in the section entitled "Centralizing Personal Data with Folder Redirection" later in this chapter. For now, just know that redirecting profile folders means just that: storing profile subfolders and the data within them, outside the main root profile folder.

How Virtualization Complicates Storing User Configuration and Files

This topic will be discussed a lot in this chapter, but to begin, you need to be very clear about why virtualization complicates user profiles and the way users store data. Fundamentally, it's because profiles were originally designed for logging into one place at a time, and when using RDS, you might be logged into more than one remote session.

RDS supports five remoting work scenarios.

- RemoteApp programs running from an RD Session Host server and displayed alongside locally running applications
- RemoteApp programs running from a VM (most often a Windows XP VM)
- A full desktop session on an RD Session Host server
- A pooled VM, which might be running any version of a Windows client operating system
- A personal VM, which might be running any version of a Windows client operating system

Figure 5-3 shows the intricate matrix of user profiles and redirected folders for users who access multiple desktop and RDS environments.

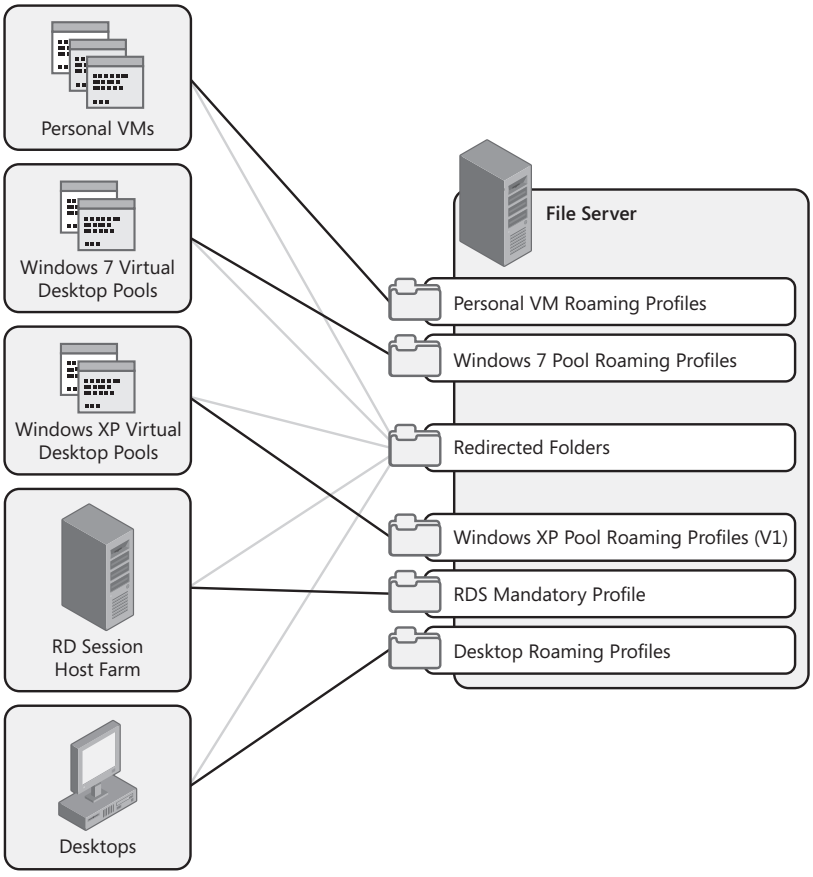


FIGURE 5-3 Providing a consistent environment for RDS environments becomes more complicated with virtualization.

So what does it mean to have all these virtualization environments available?

Using more than one or two types of virtualization can lead to profile proliferation. It's relatively simple if you use one type of virtualization. For example, if you normally work from a desktop running Windows 7 and use RemoteApp for Hyper-V to run a couple of Windows XP applications as RemoteApp programs, then you will have two profiles—one for the RemoteApp session and one for local use. Add a session to that and you could potentially have three profiles to manage. Similarly, the more server farms that a person will need to access to run RemoteApp programs, the more likely that she will have multiple copies of her profile open at once. This is a good argument against farm proliferation.

Operating systems that use V1 profiles can technically use the same V1 profile (and the same goes for operating systems that use V2 profiles). Whether this is a good idea depends on whether the settings in the profiles are appropriate to both local and remote sessions. Also, keep in mind that if you have a copy of your profile open in two sessions, then you might lose changes if you edit both copies.

Storing Profiles

By default, when you log on to a computer running Windows 7 for the first time (unless you've set up roaming profiles), you'll create a new profile in its local profile directory (%SystemRoot%\Users). This profile directory will have your name as a logon alias; it will contain your folders and NTUSER.DAT (which is a hidden file, so you won't see it unless you've enabled viewing hidden files). If left alone, thereafter you'll store everything in that location. Documents will default to Documents, images will default to Pictures, and where music is stored by default is left as an exercise for the reader. All will be well . . . so long as that's the only computer you use. If it's *not* the only computer you use, however, life gets somewhat more complicated.

Thus far, you have learned how to set up only a single RD Session Host server. However, to provide redundancy and better scale, you'll need to have multiple RD Session Host servers organized into a farm. When a user logs on to an RD Session Host server farm, the connection is passed from an RD Session Host server to the RD Connection Broker. If the user trying to connect has no current sessions, the RD Connection Broker picks the RD Session Host server with the lowest number of active sessions and sends the user there, as shown in Figure 5-4. Each time a user connects, the RD Connection Broker decides anew which server the user should connect to, based on the number of connections that each server is actively supporting and whether the user already has a session open somewhere. The user connects to the server with the fewest active connections or the one where the user already has an open session. It is likely (and highly recommended) that users will log off when not using their RD Session Host server session, so if you use local profiles for RD Session Host server sessions, then over time, a user will have a local profile on all the servers in the farm.

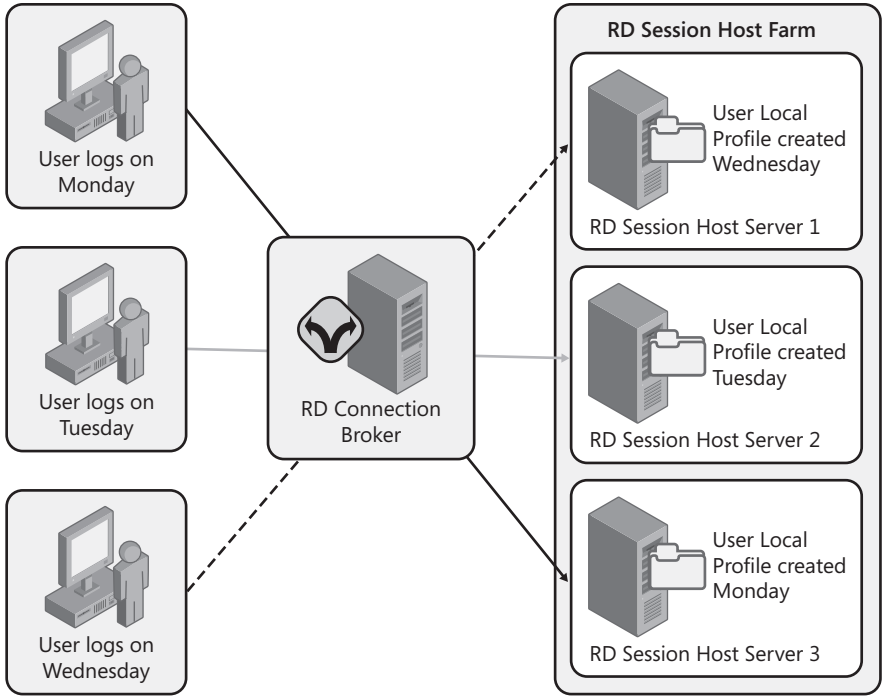


FIGURE 5-4 If you use local profiles with RD Session Host or pooled VMs, a user could eventually have local profiles on every server in the farm or every VM.

This might not sound so bad. The user's logons will occur quickly because the profile isn't loaded from the network but rather from the local computer. But when the user makes a change here and there, over time, her desktop will look completely different depending on which RD Session Host server (or pooled VM) she logs on to. (If user data is part of the profile—if you haven't redirected profile folders—the user will be even more confused because the data that she saved in one local My Documents folder won't be in another one.) If she makes a *bad* change, that change could well lead to a Help desk call that can be tricky to figure out until you determine to which RD Session Host server she is connected. This is especially true because the problem might vanish if the user logs off and then logs back on and the RD Connection Broker sends her to a different RD Session Host server.

To avoid this scenario, all the RD Session Host servers should use the same copy of the profile, which means that you need to use roaming (or mandatory) profiles stored on a network share. When a user logs on, the User Profile Service looks at the user account properties to see where the profile reserved for RD Session Host server sessions is kept and loads it from there.

When a user logs off, the profile is either deleted from the RD Session Host server or retained in the local cache, depending on the Group Policy settings applied to the RD Session Host servers. For faster logons, cache the profile. Just ensure that there's enough space on the hard disk holding the cache to support everyone who might need to cache their profile there.

Providing a Consistent Environment

The ways in which you can provide applications to users has grown, and keeping the user experience consistent across these different environments has become even more complicated. Now you must design and implement a profile strategy that takes into account the following.

- Users can use more than one endpoint type at the same time.
- Microsoft VDI can include both V1 (in Windows XP) and V2 profiles (in Windows Vista and later).
- One user can have multiple profiles.

Expect Multiple Profiles

As you offer more ways to present applications to users, delivering user configuration data in the profile gets more complicated. For example, instead of having users logging onto a single desktop and doing all of their work on that local machine, you can now offer full desktops in a session, RemoteApp programs, personal VMs, pooled VMs, and even RemoteApp programs *from* VMs. Each of these application delivery solutions has a unique environment, and therefore, when using the RDS, we recommend implementing different user profiles for each of these unique environments. The problem with this is that users expect to have the same experience wherever they log on. This is not really possible when users have multiple unique environments.

The Last Write Wins

The benefits of having multiple profiles far outweighs the profits of not having them. Implementing a unique profile for each environment helps to overcome the "Last Write Wins" problem. This is exactly what it sounds like: If a user logs on to multiple places (multiple RDS farms, for example) and those farms have all been set up so that the user utilizes a single roaming profile, then that single roaming profile gets overwritten each time the user logs off each farm. Each time the profile used in a session is copied back to the roaming profile share, it overwrites what was previously there.

The user profile is made of both folder data and registry data. You might not experience much data getting overwritten in the folder areas because you can open only certain files in certain environments (as shown in Figure 5-5). However, the user profile stored in HKCU is all contained in one file: NTUSER.DAT. As Figure 5-5 shows, if the user has a profile open in two different sessions, the second logoff will overwrite any changes saved to the profile at the first logoff.

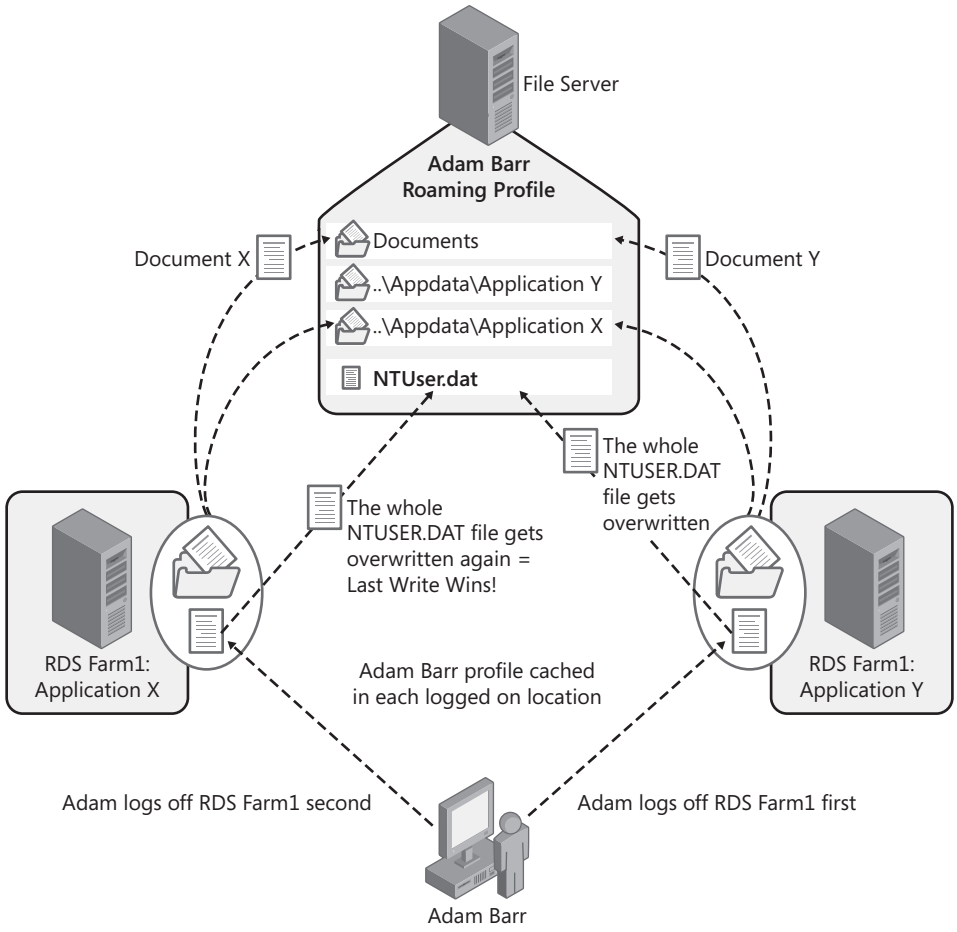


FIGURE 5-5 The Last Write Wins.

For this reason, we recommend creating multiple farms only when necessary.

> ***NOTE*** **There are third-party products that will allow you to use the same profile in multiple environments and will still avoid the Last Write Wins problem. Please see the RDS Partner page at *http://www.microsoft.com/windowsserver2008/en/us/rds-partners.aspx* for more details on these products on partner offerings.**

Design Guidelines for User Profiles

Each of the following affects how you save user-specific configuration settings and data for use with RDS.

- Local profiles generally aren't suited to deployments of more than one RD Session Host server because the user experience will be different on every RD Session Host server.
- Large roaming profiles can increase logon and logoff times. The User Profile Service must copy the files to the endpoint and then copy them back to the profile when storing files on a personal VM can complicate backups and restoring data.
- Rollback reverts all changes to a pooled VM to the state when you took the snapshot.
- Profile settings are stored as a flat file written back to the profile storage location at logoff.

The following sections explain how these facts affect your design.

Balance Flexibility and Lockdown

Local profiles aren't a good fit for RDS deployments larger than a single server. Storing local profiles on RD Session Host servers in a multi-server environment will cause the following problems.

- It leads to an inconsistent user experience and can create problems that are hard to troubleshoot because they're linked to logging onto a specific RD Session Host server.
- It fills up an RD Session Host server hard disk with duplicate copies of a profile (that is, the profile will be stored on each RD Session Host server that a user logs on to).
- It requires that you back up the RD Session Host server because it now holds user data.

You have two remaining choices: roaming profiles and mandatory profiles. Neither choice is always appropriate. The option that you pick depends on the amount of control you want and have authority to implement.

Roaming profiles can be freely edited by their owners within the limits defined by Group Policy (discussed in Chapter 6, "Customizing the User Experience"). That is, if you've defined the wallpaper for a user group via Group Policy, that will be the wallpaper every time anyone in that user group logs on. If you haven't specified the wallpaper using Group Policy, anyone is welcome to change the wallpaper when connecting to the RD Session Host server. Like local profiles, roaming profiles store user configuration data in NTUSER.DAT.

Mandatory profiles differ from roaming profiles in that their owners can edit them, but any changes that they make will not be saved to the profile. This can speed up logoff times because nothing is written back to the network share where you've stored the mandatory profiles. More insidiously, mandatory profiles don't save any data to folders stored within the profile folder. You *must* use Folder Redirection if using mandatory profiles, if you want users to be able to save data to their personal folders. In fact, that's worth highlighting in a cautionary note.

CAUTION **If you use mandatory profiles or pooled VMs with rollback enabled, you *must* configure Folder Redirection to allow users to save files to their personal folders that are part of their profiles.**

The core choice between mandatory and roaming profiles is the tradeoff of flexibility versus control. Mandatory profiles eliminate the chance of a user making a bad change that can't be fixed by logging off and logging back on again. Mandatory profiles also speed logoff times because they don't need to be written back to the share.

However, mandatory profiles don't allow users the degree of personalization that many people have come to expect from Windows. In addition, mandatory profiles don't allow other applications to save data to the profile either. This means that some security applications that require giving users a private key [such as the encrypted file system (EFS)] don't work with mandatory profiles. The choice will depend on your corporate culture, your need to use applications that require private keys, and the ability of the IT department to control the desktop.

ON THE COMPANION MEDIA **One solution to the choice between roaming profiles and mandatory profiles is not to choose. Use mandatory profiles and combine them with a mechanism that allows users to save selected settings and have them applied at logon. Windows Server 2008 does not include this functionality, but several RDS ISVs or consulting partners do. You can find an example of this functionality—a tool named Flex Profiles—from the following link on the companion media:** ***http://www.immidio.com/flexprofiles*****.**

Use Folder Redirection

Whether you're using roaming profiles or mandatory profiles, it's best practice to use Folder Redirection with sessions or pooled or personal VMs.

If you're using roaming profiles, Folder Redirection will ensure that the profile stays small. A large profile will slow both logon and logoff times. The fastest approach is to use local profiles, but for reasons already discussed, you don't want to combine local profiles with RD Session Host servers.

If you're using mandatory profiles, then use Folder Redirection selectively. Any folders stored in the profile folder will become read-only. For some folders, this is very bad news because people won't be able to save their documents or pictures in their personal folders. But for some folders, this is exactly what you want. For example, if you don't want people to remove icons from the Start menu permanently, leave the Start Menu folder in the profile folder. See the section entitled "Centralizing Personal Data with Folder Redirection" later in this chapter for how to implement Folder Redirection.

Compartmentalize When Necessary

It is generally best practice to maintain different profiles for different environments because different types of virtualization can have different user configuration requirements. Don't go crazy creating different profiles for every possible occasion, but make sure your profile plan supports the various ways people use RDS. Compartmentalizing can also help avoid accidental overwrites.

- You might need V1 profiles to access terminal servers running versions of Windows earlier than Windows Server 2008, and V2 profiles to access RD Session Host servers.
- Implement roaming profiles for use with VM pools to keep the user experience consistent and avoid losing profile changes to rollback.
- Personal VMs can use a local profile for faster logons.
- To avoid the Last Write Wins problem, avoid users opening the same profile on multiple machines at the same time.

Prevent Users from Losing Files on the Desktop

There are a couple of cases where it's really important to prevent users from saving files to the desktop.

Users can lose, or misplace, data when using RemoteApp programs if you're not careful about Folder Redirection. Here's why: The Desktop folder contains everything that you can see on the desktop—files and shortcut icons. Many users are used to saving documents to the desktop. This is acceptable if you're actually seeing the full desktop, but if you're using RemoteApp programs, users don't *see* their desktop in the RD Session Host server session. Users could save data to the desktop and then not know where that data actually is because they can't see it. (They could open a document if they moved to the Desktop path when opening a file, but just double-clicking a document on the session desktop is not possible in this scenario.) To prevent users from saving files to the desktop, you can make the desktop read-only and trigger an error message if the user tries to save files to the desktop. To do this, you'll need to do the following.

- Redirect the Desktop folder to an external share.
- Set the permissions on this external share to read-only.

> ***NOTE*** **For instructions on how to create a read-only desktop, read the section entitled "Creating a Safe Read-Only Desktop" later in this chapter.**

If you keep the Desktop folder in the profile folder and use mandatory profiles, then people can save files to the desktop . . . as long as they are logged on. When the user logs off, however, no changes are saved, including saved files on the desktop. The same thing will happen to users of VM pools with rollback enabled; anything saved by the user to the VM during each session will be discarded once the VM snapshot is invoked.

In both cases, redirect the desktop to a folder so users can save data there without it being discarded at logoff.

> ***NOTE*** **For instructions on implementing Folder Redirection, see the section "Centralizing Personal Data with Folder Redirection" later in this chapter.**

Upload Profile Registry Settings in the Background

NTUSER.DAT is updated only when a user logs off. A user who does not log off isn't saving changes. This can lead to data loss. A new policy in Windows Server 2008 R2 enables this file to be uploaded while the user is logged on, as follows.

Computer Configuration |Administrative Templates | System | User Profiles | Background upload of a roaming user profile's registry file while user is logged on

Configure the setting to upload NTUSER.DAT on a set schedule (at a certain time of day) or at a set interval, designated in hours.

> ***NOTE*** **This setting does not upload any other profile data, just the contents of HKCU.**

Speed Up Logons

People are sensitive to the amount of time it takes to log on to a session. If it takes too long, you'll have problems with people leaving their sessions open rather than logging off. This is a security risk, has the potential to lock files that more than one person might need to edit, and keeps processes open on the RD Session Host server. You can disconnect and terminate sessions forcibly using Group Policy, but this has other drawbacks.

To encourage people to log off, make the logon process as painless as possible. You've already learned about using Folder Redirection to minimize the size of a profile. To speed things up, you can also employ Group Policies to do the following.

- Cache roaming profiles.
- Limit the amount of time an RD Session Host server or VM will try to load the user profile before using a temporary profile.
- Set an upper limit on the size of a user profile.
- Process group policies asynchronously.

New to Windows Server 2008: Speeding Up Logoffs

Speeding up logons is important, but when it's Friday afternoon and you want to get out of the office, logoffs are just as important. There are two ways in which Windows Server 2008 and later help logoffs take less time.

You can limit the size of a profile using Group Policy (and help this limit by redirecting the folders out of the policy). This policy, Limit Profile Size, is set per user and is located in User Configuration | Policies | Administrative Templates | System | User Profiles.

Prior to Windows Server 2008, there was a nasty catch when it came to profile quotas: Windows was serious about enforcing this limit. If you made your roaming

profile larger than Group Policy allowed, Windows would prevent you from logging off until you made the profile smaller. In Windows Vista and later, you can log off, but if the profile is larger than the size permitted by Group Policy, the profile changes won't get written back to the roaming profile storage area.

Before Windows Server 2008, another issue that could delay logoffs (or prevent you from unloading your roaming profile altogether) was applications or drivers that left handles to the registry open (in other words, they started to use it but never broke the connection). Microsoft had a separate tool called the User Profile Hive Cleanup Service (in an application called UPHClean) that checked for these open handles and closed them so users could log off. In Windows Server 2008 and later, UPHClean functionality is handled by the User Profile Service.

Caching Roaming Profiles

To reduce the time that it takes to log on to an RD Session Host server, the server will cache the roaming profiles. Ordinarily, RD Session Host servers attempt to retrieve the roaming profile from its central location. In cases when the network connection to the profile server is too slow or not working, however, being able to log on with a locally cached copy of your profile can at least speed things up. Caching stores a copy of the profile on the RD Session Host server. This profile cache isn't used if the original roaming profile is available, but it can speed up logons in the case of slow or absent network connections.

Caching profiles is not without its drawbacks. It consumes hard disk space on the RD Session Host server. It can also prevent new users from logging on if the space allocated to cached profiles gets filled up. If you do cache profiles, make sure that you've got sufficient space for your user base and use Group Policy to delete profiles that aren't being used.

CAUTION **Don't delete user profiles from the RD Session Host server using Windows Explorer or the delete command-line tools, because this does not clean up the registry entries associated with the profile and can affect the user's ability to log on again. Configure the RD Session Host servers with Group Policy to delete any profiles unused for a given period.**

Process Group Policy Asynchronously

Caching user profiles also means that you can use asynchronous processing of Group Policy, a policy processing model introduced in Windows Server 2008. You can apply Group Policy synchronously or asynchronously. If you apply it synchronously (the default model for a server), logon doesn't complete until the Group Policy settings that apply to that user are applied. If

you apply Group Policy asynchronously (the default action for a desktop), the user can log on while Group Policy is being applied. Asynchronous processing can lead to changes in the user environment after users have logged on but will speed up logon times if Group Policy processing is slowing things down. For a review of the connection process, see Chapter 3, "Deploying a Single Remote Desktop Session Host Server."

Allow asynchronous Group Policy processing by enabling the following Group Policy setting.

> Computer Configuration | Policies | Administrative Templates | System | Group Policy | Allow Asynchronous User Group Policy Processing When Logging On Through Remote Desktop Services

This policy works only when logging on to an RDS session host. It's not needed when logging on to desktop pools, because a desktop operating system already processes Group Policy asynchronously by default.

Deploying Roaming Profiles with Remote Desktop Services

This section discusses managing roaming profiles in an RDS environment, including the following.

- Creating roaming profiles
- Converting an existing local profile to a roaming profile
- Creating a default network profile
- Using Group Policy to set up the roaming profile storage area automatically
- Implementing a Group Policy infrastructure that supports these policies, including security filtering and loopback policy
- Managing roaming profiles cached on the RD Session Host servers

Creating a New Roaming Profile

To implement roaming profiles, you will need to

1. Create a network share in which to store the roaming profiles.
2. Configure the user accounts (through Active Directory Users And Computers or Group Policy) to use roaming profiles.
3. Have each user log on and create the roaming profile.

First, create a shared network location to store the roaming profiles. On the file server, create a new folder and set the appropriate NTFS and share permissions, using the guidelines in Table 5-5.

TABLE 5-5 Recommended Share and NTFS Permissions for an RDS Roaming Profiles Storage Folder

USER ACCOUNT	PERMISSION TYPE	NTFS PERMISSIONS
Authenticated Users group	Share	Full Control
Creator Owner	NTFS	Full Control, subfolders and files only
Local System	NTFS	Full Control on this folder, subfolders, files
User/Group whose profiles will be stored in the folder	NTFS	List Folder Content/Read, Create Folders/Append Data, all on this folder only

DIRECT FROM THE SOURCE

How Profile Folders Are Named

Sergey Kuzin
Software Development Engineer II

The way that a user's profile folder is named depends on the circumstances in which it's created. The user My Name (with user name Myname) with an account in Domain1 will store his profile in one of two places: \RDS-Roaming-Profiles\Myname or \RDS-Roaming-Profiles\Myname.Domain1.

The best case is to add the domain name to the profile path; this disambiguates the path when there are two (or more) users with the same name living in different domains. For example, in a large corporate network, you might have Domain1\Myname (that's me) and Domain2\Myname (some other user). When Domain1\Myname logs on to a legacy terminal server the profile created for him will be ...\Myname. If Domain2\Myname later wants to store his profile on the same server, he will have a problem. That's why you add *.domain* to the profile path, so that users with the same name but from different domains would have different profiles. So ideally, you always want to add *.domain* to the profile path.

But then, what do you do with profiles that were created before you made this change and don't have *.domain* in the name? Leave them as is. But in this case, how do you know which user this particular profile belongs to? You use permissions to determine that. When the User Profile Service creates a new profile, it gives full control to the user whom this profile is created for. So, if Domain1\Myname has explicit full control permission to the ...\Myname folder, then this profile belongs to me and not to Domain2\Myname. That's why you have this logic when creating profile names.

Here is the logic you use to create the profile path.

Continued on the next page

1. **Attempt to locate the ...*username.domain* path. If it exists and the user has explicit permissions to it, then use it.**
2. **If the user does not have explicit Full Control access to ...*username.domain* or this folder does not exist, then try to access ...*username*.**
3. **If ...*username* exists and the user has explicit permissions to it, then use it.**
4. **If the user does not have explicit Full Control access to ...*username* or the folder does not exist, then use ...*username.domain*.**

As you can see, by default you always create the folder with ...*username.domain*. Only when the ...*username* folder exists and the user has explicit Full Control access to it do you use it. Again, it's always best to include the domain name in the profile path so that two people with the same user name with accounts in different domains can store their profiles in the same central share.

When you've set up the profile location, configure the user account to use roaming profiles. This process varies slightly for profiles used with RD Session Host servers and for profiles used with pooled and personal VMs. You will see these differences as you step through this process. It's easiest if you configure this via Group Policy, but you will also see how to do it on a per-user basis.

Remote Desktop Session Host

To configure a user account to use roaming profiles, perform the following steps.

1. Open Active Directory Users And Computers, right-click a user's account, and choose Properties.
2. For Remote Desktop Session Host situations, navigate to the Remote Desktop Services Profile tab and type the Profile Path location using the format ***\\servername\sharename*\%username%.*DomainName***, as shown in Figure 5-6.

The variable %username% inserts the user account name into the profile path, so you don't have to customize the path for each person when adding new accounts manually or through a script. You don't need to add the .V2 extension to this path, either; it will be added automatically because the profile will be a 2008 version profile. The next time the user logs on to the RD Session Host server, he will use the roaming RDS profile.

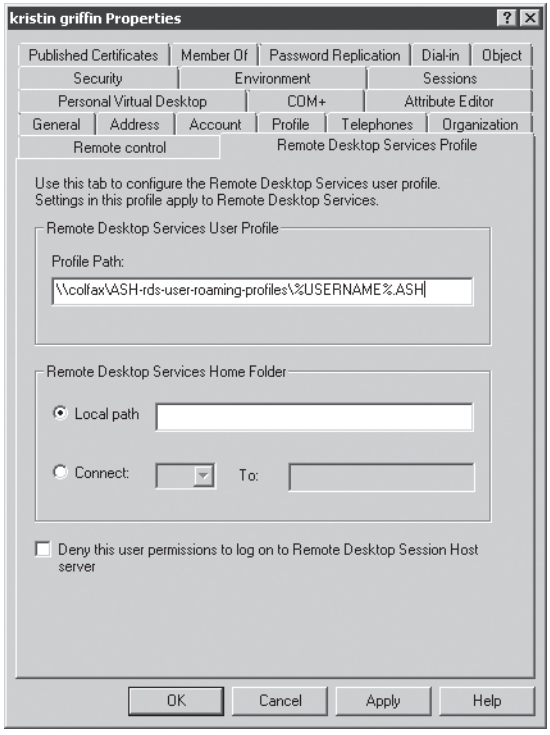


FIGURE 5-6 Enter the Remote Desktop Services profile path.

NOTE **Windows Server 2008 and later and Windows Vista profiles have a .V2 extension. Older operating systems use V1 profiles, which have no extension associated with the profile folder name.**

Virtual Machines

Pooled and personal VMs do not use Remote Desktop Services profiles. A pooled or personal VM is really a virtualized client desktop and acts accordingly—that is, it uses regular profiles. For these VM scenarios, enter the profile share's UNC path on the Profiles tab of the user account Properties dialog box, shown in Figure 5-7.

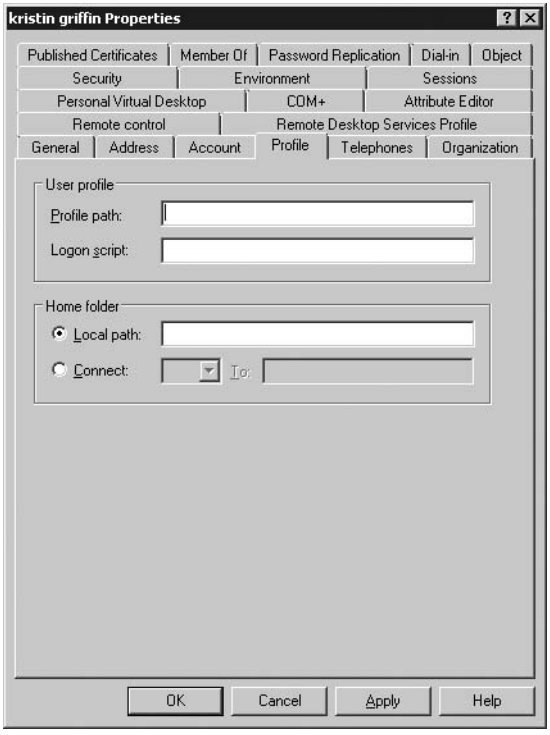


FIGURE 5-7 Specify the profile used for pooled and personal VMs on the Profile tab, not the Remote Desktop Services Profile tab.

When the user is configured to use roaming profiles, it's time to create the profile. This happens when the user first logs on to the RD Session host server (or the pooled/personal VM). When the user first logs on, the following happens.

1. The User Profile Service creates a profile folder for the user in the specified path.
2. The User Profile Service copies the default profile on the RD Session Host server or VM to give the user a profile.
3. When the user logs off, the User Profile Service copies the profile to its storage location in the specified network share. The user will be the owner of the folder and therefore will be the only one to have access to the folder and its contents.

Although a user profile folder is for the user, if Administrators also have permissions they can delete a corrupted profile or perform other maintenance easily. To permit this, give the Domain Admins group Full Control NTFS rights to the parent folder, and pre-create roaming profile folders for each user in the roaming profiles share. Make sure that the user has full control of his profile folder, subfolders, and files and that the user is also the owner of the folder. The simplest way to do this is to use Group Policy; if you keep your RD Session Host servers or pooled VMs in their own organizational unit (OU), you can also create a computer Group Policy object (GPO) with Loopback Processing enabled and give administrators access to profile contents by enabling the following GPO setting.

Computer Configuration | Policies | Administrative Templates | System | User Profiles | Add The Administrators Security Group To The Roaming User Profile Share

For more information on Loopback Processing and using Group Policy to create and manage RDS roaming profiles, see the section entitled "Using Group Policy to Manage Roaming Profiles" later in this chapter.

DIRECT FROM THE FIELD

Managing Roaming Profiles Without Admin Access to the File Server

Bohdan Velushchak
Operations Engineer, MSIT

To use roaming profiles, you need a file server to store them on. In a smaller deployment, you can have administrative rights to the file server as well as the terminal servers, but enterprise deployments often segregate ownership. If you aren't an administrator of the file server, you can't manage the folders directly—you'll need to ask the file server administrator. Even the Group Policy setting Add The Administrators Security Group To Roaming User Profiles will not help if the RDS administrator is not a member of the Administrators group on the file server. You could lobby to become a member of the Administrators group on the file server, but this is counter to Least Privilege Access principles.

You can resolve this situation with a logoff script. Use Icacls.exe to include RDS administrators to the user profile's permissions during logoff from user's security context. This works because the user has full access permissions to her profile, so she can add necessary permissions for RDS Administrators. For example, the Logoff script might look like this.

```
Icacls.exe //<profile root>/%username%.%userdomain%.v2 /grant
   <RDS Admins group>:
F /T /Q
```

Add this script to each user through Group Policy: User Configuration | Windows Settings | Scripts | Logoff Script. Now you can manage that profile folder.

There are two reasons to do this at logoff, not logon. First, if the user is logging on for the first time, the profile folder might not yet exist, so the settings wouldn't apply until the second time. If the user never logged in again, you couldn't delete her profile without the help of the file server administrators. Second, if the profile is large, it takes some time for Icacls.exe to go through the whole tree. Users do not like long logon times, so why make them wait to start working? Let the script process permissions when they're done working and are less concerned about time.

Converting an Existing Local Profile to a Roaming Profile

Sometimes you will want to convert existing local profiles to roaming profiles. This can apply if you are converting a traditional desktop deployment to an all-RDS deployment, and you are willing to risk that the local profile settings are appropriate for the remote work environment.

> ***NOTE*** **It's often unwise to convert a local profile that a user has been using on a personal desktop to a Remote Desktop Services roaming profile. The user might have administrative access to her personal computer and could have installed numerous applications and made many customizations that don't apply to the shared (and more locked-down) world of RD Session Host servers.**

Converting local profiles to roaming profiles is really simple. Configure all user accounts to use roaming policies as described earlier, and specify that cached copies of the profile should be deleted. When users log on to the server where their local policy resides and then log off, their local profile will be copied to the network share that you specified. The cache on the server will be deleted and only the roaming profile in the network share will remain.

You might have done this conversion in Windows Server 2008 using the Copy To button in the User Profile Properties dialog box. This is no longer possible on a server running Windows 2008 R2 or a client running Windows 7—the button has been disabled.

DIRECT FROM THE SOURCE

Why the Copy To Button Is Disabled

Kyle Beck
Program Manager, Microsoft

The Copy To button is now disabled, because even though this button was used to overwrite a profile with another profile, it was unsupported to use it to edit the default profile. It was unsupported because the source profile was just copied wholesale into the default profile—the Copy To button performed a complete copy of everything in the source profile over the default profile. This could lead to errors in the registry because references to the source user would persist on any new user created from the new default profile. Because it was an unsupported method, its behavior was updated; the default profile is now the only one that is copyable using this button.

The removal of this functionality doesn't prevent you from converting local profiles to roaming profiles or even overwriting one user's profile with another's. Removing the functionality prevents you from overwriting the default user profile with another user profile. People often overwrote the default user profile with a customized one from another user to deploy customized profiles to new users. As described in the Direct from the Source sidebar entitled "Why the Copy To Button Is Disabled," doing this was unsupported (although popular) as far

back as Windows XP, because although this "worked" for many people, it actually was not a clean process. It could lead to problems if that profile had been used at all, and it would also "tattoo" the profile with inappropriate settings and naming, such as the following.

- A list of that user's frequently run programs.
- The user's documents folders will be incorrectly called Administrator's Documents.
- The user might have access to Administrative Tools (this is incorrect for regular users).
- Windows 7 libraries will be broken.

ON THE COMPANION MEDIA **There are other implications to overwriting the default user profile with a user profile by way of the Copy To button. See this article (also on the companion media) for more information: *http://blogs.technet.com/deploymentguys/archive/2009/10/29/configuring-default-user-settings-full-update-for-windows-7-and-windows-server-2008-r2.aspx*. This article also discusses some options for customizing the default profile in Windows 7.**

Customizing a Default Profile

Customizing the default profile is one way to ensure that all new RDS users start with the same settings. The only supported method for customizing the default profile is to use the Sysprep.exe tool (built into Windows 7 and Windows Server 2008 R2) to overwrite the default profile with the profile that you are logged onto when you run Sysprep.exe. Here are the steps.

1. Log on as an administrator and customize the profile as needed. This is the profile that will be copied over the default user profile.
2. Create an Unattend.xml file and add a line of code to it to tell it to copy the profile of the user logged on over the default profile when the system reboots. The line you add is

   ```
   <CopyProfile>true</CopyProfile>
   ```

   The following is example code for a 64-bit version Unattend.xml file with the extra line of code added.

   ```
   <?xml version="1.0" encoding="utf-8"?>
   <unattend xmlns="urn:schemas-microsoft-com:unattend">
   <settings pass="specialize">
   <component name="Microsoft-Windows-Shell-Setup" processorArchitecture="amd64"
   publicKeyToken="31bf3856ad364e35" language="neutral" versionScope="nonSxS"
      xmlns:wcm="http://schemas.microsoft.com/WMIConfig/2002/State"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <CopyProfile>true</CopyProfile>
   </component>
   </settings>
   <cpi:offlineImage cpi:source="catalog:e:/clg files/64-bit/install_windows 7
      ultimate.clg" xmlns:cpi="urn:schemas-microsoft-com:cpi" />
   </unattend>
   ```

3. Save this Unattend.xml file to C:\Windows\System32\Sysprep.
4. After you have the Unattend.xml file in place, open a command prompt and type the following command.

```
sysprep.exe /oobe /reboot /generalize /unattend:unattend.xml
```

> ***NOTE*** **The article at *http://support.microsoft.com/kb/973289* explains how to do this, but at the time of this writing, the syntax is incorrect. Use the one provided here.**

After you run this command, the server will reboot. When it comes back up, the default profile will be overwritten with the one that was logged in when you ran Sysprep. *Now* you can highlight the default profile and use the Copy To button to copy the profile to a network share to be used for roaming profiles.

> ***CAUTION*** **Don't run Sysprep on a production machine. The Sysprep command resets the computer SID as well as eliminating system-specific data like the computer name and the domain affiliation. It can also remove unique hardware drivers and can reset the Windows activation key. If you are using VMs, then one workaround is to take a snapshot of the VM before running Sysprep. After you are done running Sysprep, rebooting, and copying the default profile to another location, apply the snapshot and the VM will be rolled back to its prior state.**

Creating a Default Network Profile

You have already learned (in the section titled "How Profiles Are Created" earlier in this chapter) when a network default user profile would be used to create new user profiles. Using a default network profile to create new roaming profiles might benefit your roaming profiles implementation because it ensures that when new profiles are created, they all stem from the same source.

Reasons Not to Create a Network Default Profile

Creating a network default profile can work well to deploy customized profiles in low-complexity environment. But it's not always the best solution.

First, there is no way to distinguish when a network default profile should be used to create a new roaming user profile. As discussed earlier in this chapter, in complex remoting scenarios, it's possible for people to have more than one remoting profile, and if you point them to the same starting point, they will start with the same profile in all scenarios. For example, a new profile created when the user logs on to a Windows 7 pooled VM would stem from the same network default user profile that is used to create a new user roaming user profile for use in an RD session host server

environment. Depending on how you implement profiles, this might or might not be acceptable.

In short, Windows doesn't allow you to specify more than one default profile location. So unless it's okay to use the same default profile to build all roaming profiles, we recommend applying customizations through Group Policy or scripting.

Assuming that you can use a network default profile for all your scenarios, on Windows 2008 (and Windows 7) you can copy a local default profile to the NETLOGON share on a domain controller, following these steps.

1. Log on to the server with an admin account.
2. From the Run box, browse to the domain controller: \\DOMAIN CONTROLLER\NETLOGON
3. Create a folder in the NETLOGON share and name it Default User.v2.
4. From Server Manager, click Change System Properties, navigate to the Advanced tab, and then click the Settings button in the User Profiles section.
5. Select the Default Profile from the list of profiles stored on the server and click Copy To.
6. Browse to or type the network path \\DOMAIN CONTROLLER\NETLOGON Default User.v2.

BEST PRACTICE **Ensure that the profile doesn't contain any unnecessary data. A large default network profile will slow down the initial profile creation process because new profiles have to pull this large amount of data across the network.**

Using Group Policy to Manage Roaming Profiles

You've seen how to dictate who uses roaming profiles by settings this up on a per user basis in Active Directory Users And Computers. If you have more than a few users, it's easiest to create a GPO that dictates the RDS roaming profile location for everyone who logs on to a farm. This section explains how to do this and how to set up the Group Policy infrastructure that you'll need.

The single most important part of successfully using roaming profiles with RD Session Host servers is to set up the RD Session Host server environment OU and create the GPOs correctly. Group Policy has many different uses, but it all comes down to making changes to many computers or many users all at once.

There are two broad categories of Group Policy: computer settings and user settings. Computer settings are applied at boot time, or on an RD Session Host server (see Chapter 2, "Key Architectural Concepts for Remote Desktop Services," for more details), when a session starts (to apply the settings to the session). User settings are applied when the user logs on

to the session. Because settings are applied to users at logon, they don't have to be saved as part of a user's account properties. Because they're applied second, settings applied to a user will control when there's a conflict.

Because of the order in which user and computer Group Policy is applied, when managing RD Session Host server settings, you'll almost always use an additional GPO to enforce *loopback policy processing*. In short, loopback policy reapplies the user-specific settings that are placed on the OU where Loopback Processing is enabled after the normal user GPOs are applied. The result is that settings placed on the RD Session Host server OU will always take precedence in case of a conflict. If you have blocked GPO inheritance on the RDS OU, then only the user policies that you place on the OU will be implemented for your users. You'll find out more about loopback policies in the section entitled "The Ins and Outs and Ins of Loopback Policy Processing" later in this chapter.

There's some overlap between the computer- and user-specific settings in Group Policy, but you'll generally find that you'll need both to configure the users' working environment. When setting up an RD Session Host server environment, where it's important not just that you are logging on but that you're using an RD Session Host server, you'll *definitely* need both.

ON THE COMPANION MEDIA **The following explanations assume that you have permission to manage Group Policy for your RD Session Host servers. If this is not the case, you'll need to provide the instructions to the administrator controlling Group Policy for your organization and let him or her fit them into corporate management policy. This is one way to organize your RD Session Host server GPOs, but it is not the only possible model. GPO architecture is unique to the particular situation. For example, for some organizations, blocking inheritance might not be an option for business policy reasons. For more information on Group Policy modeling, see "Design Considerations for Organizational Unit Structure and Use of Group Policy Objects," located at *http://technet2.microsoft.com/windowsserver/en/library/2f8f18cf-a685-48db-a7be-c6401a8fb6341033.mspx?mfr=true*. (This article was written for Windows Server 2003, but it still applies.) You can also find the link on this book's companion media.**

Controlling Group Policy Processing for an RDS Environment

When you have multiple users working on one computer, you need to control the environment as much as possible. The easiest way to do this is to perform the following steps.

1. Put RD Session Host server farms and all VMs pools into their own OUs.
2. Block inheritance of all GPOs that are not specifically enforced. (You might not have this option, depending on company policy.)
3. Place computer and user GPOs on these OUs to specify the settings to be implemented for each pool and farm.

Here's how to do all this.

ORGANIZE FARMS AND POOLS INTO OUS

First, create an OU for each RD Session Host farm or VM pool. (Because all members of a farm or pool are homogenous, they should all be in the same OU.) Open Active Directory Users And Computers, right-click the domain, and choose New, Organizational Unit. Name it after the farm (for example, RDSH Farm1) and then drag all computer objects in the farm or pool into the OU (see Figure 5-8).

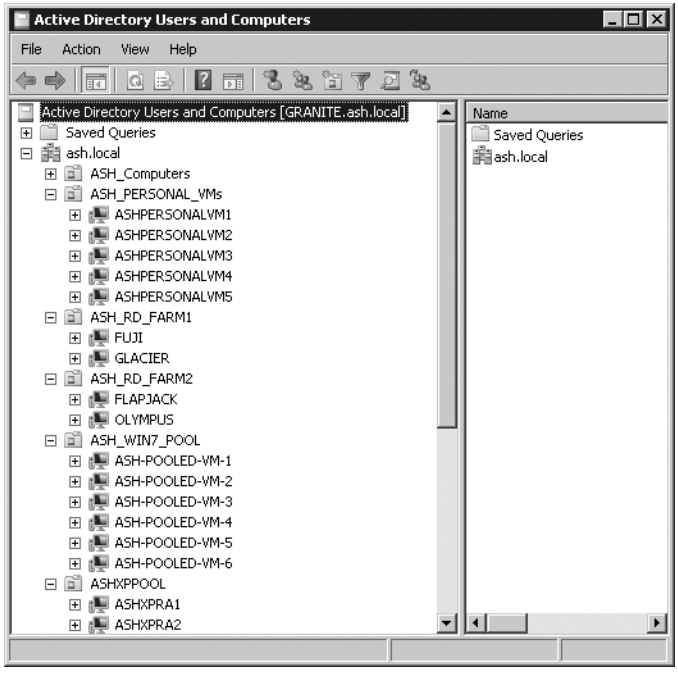


FIGURE 5-8 Create OUs for your RD Session Host server farms and VM pools.

BLOCK GPO INHERITANCE

Next, if possible in your organization, block GPO inheritance for this OU. This ensures that only computer settings set by GPOs linked to this OU will apply to the computers in this OU. It also ensures that with Loopback Processing enabled, only user settings set by GPOs linked to this OU will be applied to users logging on to the computers in this OU; other GPOs set at the domain or site level will not be applied.

To block inheritance for a farm or pool OU, open the Group Policy Management console (GPMC; do this by clicking Start, Programs, Administrative Tools, and Group Policy Management), right-click the RD Session Host server's OU, and choose Block Inheritance. If possible, also do this for your pooled VM OUs. Personal VMs can be controlled like this, but more likely they will act as regular desktops in your environment and will treated as such in the case of Group Policy processing.

IMPORTANT **Company policy might prevent you from blocking inheritance. You can still know exactly what policies are going to be applied to the users and computers in your OUs; it will just take more effort because you will have to know about all Group Policies applied at higher levels.**

CREATE GPOS FOR USER AND COMPUTER SETTINGS

There are multiple ways to set up policies, but it is usually easiest if you separate computer- and user-specific settings into different policies. Although one policy might contain both user- and computer-specific settings, it's simplest to isolate the two types of settings unless your environment is very small or your user base is very homogenous. This allows you to create a consistent model of RD Session Host server management while still allowing you the flexibility to apply different policies to different groups of users and computers (that is, using a GPO on multiple OUs if the functionality is needed in multiple places). Create two different types of GPOs: a computer GPO and user GPOs, as shown in Figure 5-9.

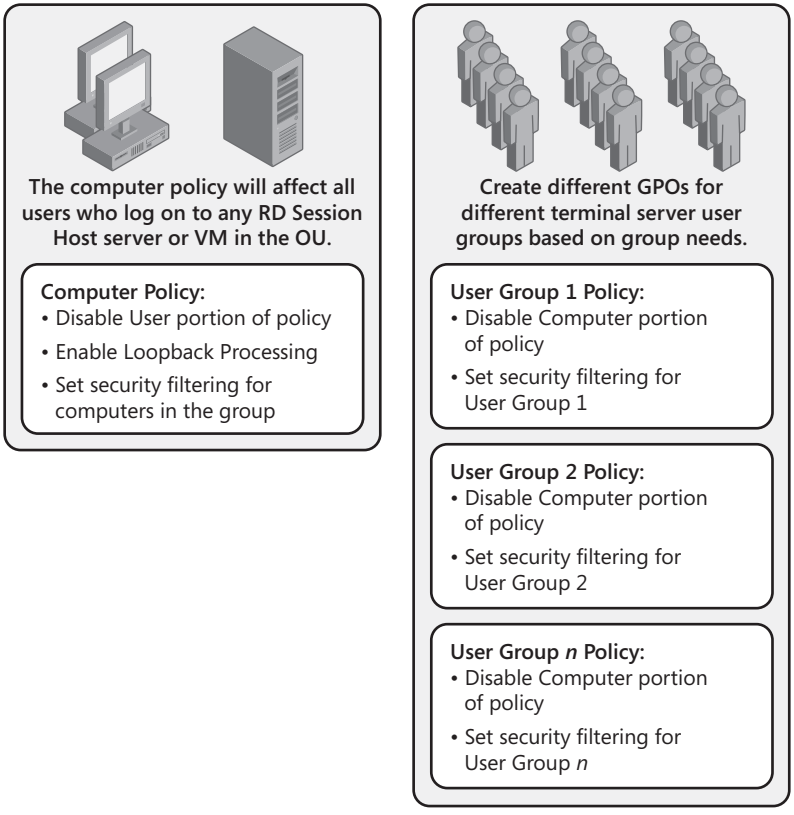


FIGURE 5-9 Create separate user and computer GPOs for the RDS environment.

To create the GPOs, open the GPMC (by clicking Start, Programs, and Administrative Tools). Right-click the Group Policy Objects folder in the left pane, found under your domain folder, and choose New to open the dialog box shown in Figure 5-10.

Name the computer policy something descriptive, such as RDS Computer GPO, and then click OK.

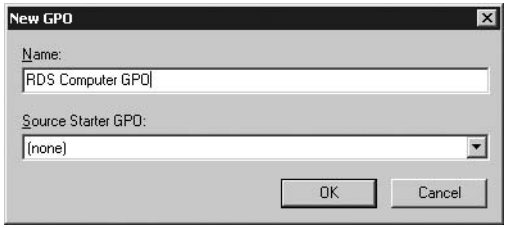


FIGURE 5-10 Create an RD Session Host server computer policy.

Next, create another policy that will hold user-specific settings, naming it something like RDS User GPO. Click OK, and you will be back in the GPMC, with a list of available policy objects that includes the ones you just created, as shown in Figure 5-11.

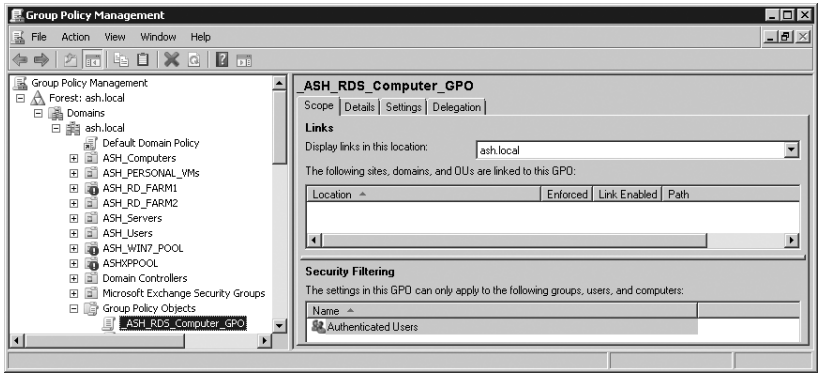


FIGURE 5-11 Create computer- and user-specific GPOs.

Next, ensure that each GPO is specific to one type of settings—computer or user. This is optional, but this will give you more control over your RDS environment.

Click the Details tab in the upper portion of the right pane. Here, there's a GPO Status drop-down list with four options: All Settings Disabled, Computer Configuration Settings Disabled, Enabled, and User Configuration Settings Disabled. For your computer-specific GPOs, make sure that no user-specific settings will be applied by setting the Status to User Configuration Settings Disabled. Follow the same process to create a new user-specific GPO. For the User GPO, navigate to the drop-down menu on the Details tab and set the GPO Status to Computer Configuration Settings Disabled.

Updating Group Policy

Active Directory Domain Services (AD DS) does not immediately send user Group Policy changes down to the computers to which they apply. The Group Policy engine on the computer actually pulls the GPO changes from AD DS at specific intervals, called the *refresh interval*. By default, the refresh interval is 90 minutes (plus a random time ranging from 0 to 30 minutes). To immediately see the effects of changes that you make to GPOs, you can force this refresh. Open a command prompt on your RD Session Host server and type gpupdate /force. Most computer policies can be updated just by doing this; a few (like Folder Redirection) will require a reboot.

The Ins and Outs and Ins of Loopback Policy Processing

Outside an RD Session Host server environment, you often apply Group Policy based on the persona of the user logging on. If you don't want Adam Barr to open Control Panel, for example, you probably feel much the same way about this whether Adam Barr is logged on to his desktop computer or his laptop. Similarly, if you don't care whether he is running Control Panel, then you continue not to care whether he's logged on to his desktop or his laptop. It's his space—let him mess it up. (The Help desk might feel differently about this, but that's another matter.)

As discussed in "Using Group Policy to Manage Roaming Profiles" earlier in this chapter, the computer policy will always be applied first, then the user policy. If a user policy and a computer policy conflict, the user policy will "win," because it's applied last. Any Group Policy stored locally on the computer is applied first. Next, policies placed at these levels are applied in order (local, Site, Domain, OU), as shown in Figure 5-12.

In case of conflicts, the policy applied last wins. For example, computer policies set on a computer OU will override conflicting policies set at the domain level. And user policies will overwrite computer policies in conflicting situations (some settings can be set for a computer and also for a user) because they are applied after computer policies.

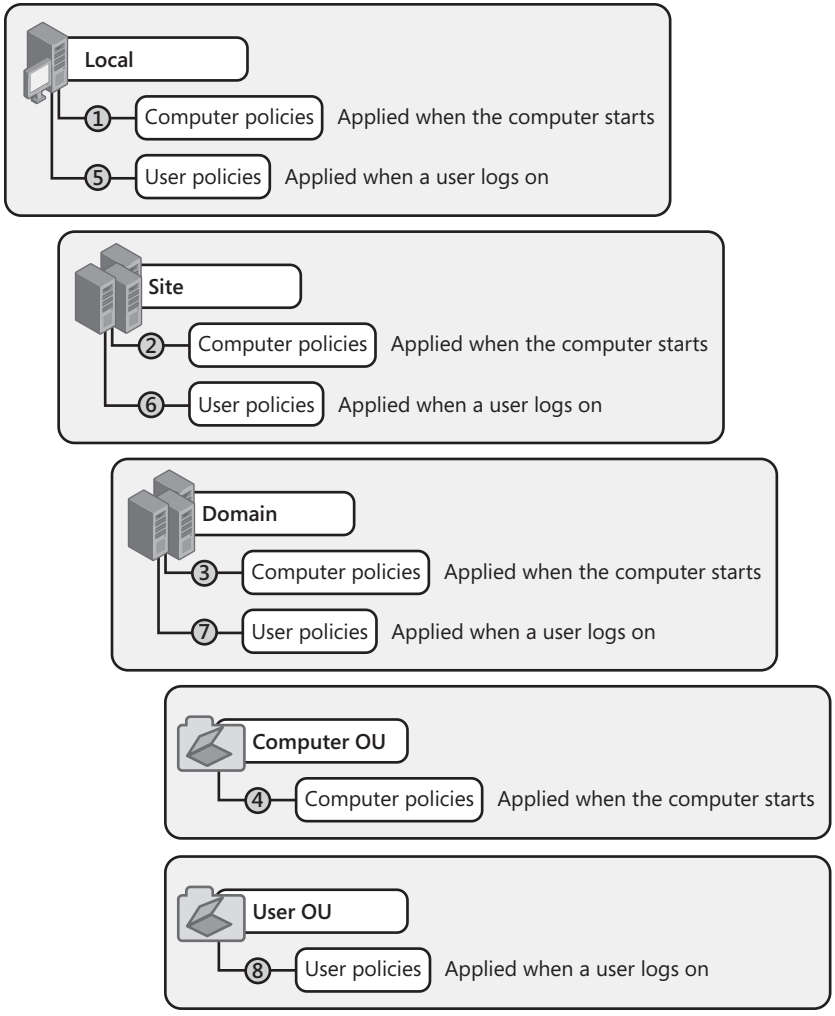


FIGURE 5-12 Group Policies get applied from the top down.

On a personal computer, it's perfectly acceptable to have the identity of the person logging on define the final settings for Group Policy. But RD Session Host server farms and pooled VMs are location-specific or context-specific situations in which *where* you are matters even more than *who* you are. For example, you might decide that it's acceptable for users to use clipboard redirection when connecting to personal VMs, but for security reasons, you don't want them using clipboard redirection when connecting to an RDS server farm hosting sensitive data. You need policies applied based on which computer you are logged on to. In this case, you will apply loopback policy processing to tell the Group Policy engine to apply the user GPOs that are applied to a computer OU (for example, to an RDS farm OU) after applying the user GPOs that are normally applied during logon. With loopback policy processing enabled, GPO processing will now work as shown in Figure 5-13.

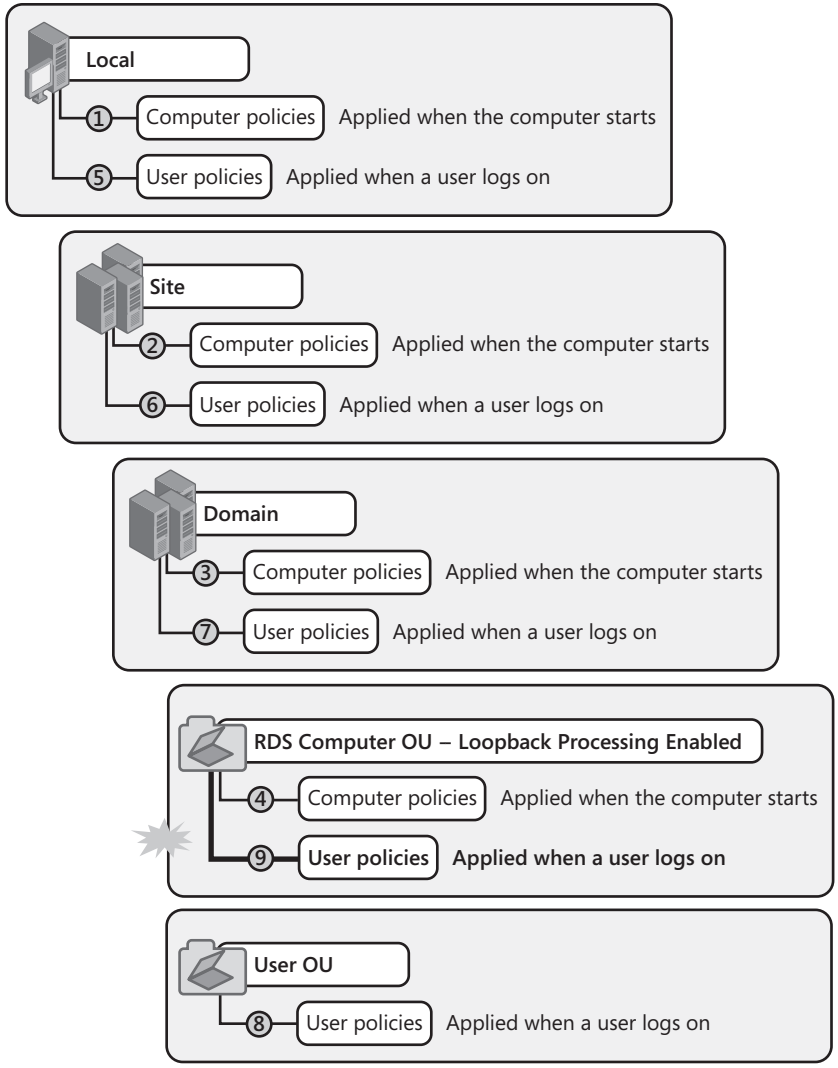


FIGURE 5-13 Loopback Processing changes the effective Group Policy results.

When the RD Session Host server starts, computer GPOs are applied. When the user logs on to the RD Session Host server, the User GPOs are applied to the session. Then, because loopback policy processing is enabled, User GPOs that are applied to the RD Session Host server OU are applied last. In addition, if you have blocked inheritance, it's possible that the *only* GPOs that will be applied are computer and user GPOs that are placed specifically on the OU.

To enable Loopback Processing, right-click the Computer GPO applied to the RD Session Host server OU and choose Edit. The Group Policy Management Editor opens the GPO. Go to Computer Configuration, Policies, Administrative Templates, System, and Group Policy and

find the User Group Policy Loopback Policy Processing Mode node in the pane on the right. Double-click it and you will see the dialog box shown in Figure 5-14.

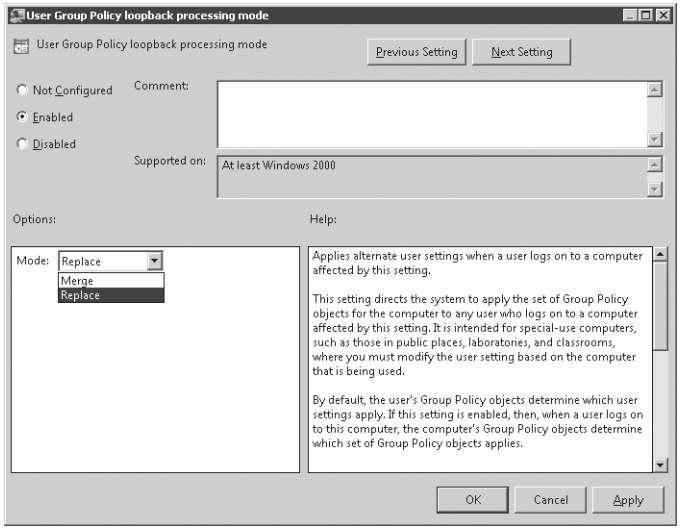


FIGURE 5-14 Enable loopback policy processing from the User Group Policy Loopback Processing Mode Properties dialog box.

HOW IT WORKS

Applying Loopback Policy

Loopback policy can apply to users in one of two ways: Merge Mode and Replace Mode.

- In Merge Mode, loopback policy processing will apply the user GPOs placed on the RD Session Host server OU along with the other normal user GPOs applied from the OU where the user account resides. If there is a conflict, then the user GPOs applied to the RD Session Host server OU will prevail.
- In Replace Mode, the Group Policy engine ignores all other user GPOs from the User OU and applies only the user GPOs applied to the RD Session Host server OU.

Merge Mode and Replace Mode affect only GPOs placed on the OU where the user account resides. User GPOs placed at higher levels (for example, at the domain level) will still be applied unless you have specifically blocked inheritance on the OU where the computers reside.

Whether you choose Merge Mode or Replace Mode depends on your goals and how you've set up the rest of your environment. If users are using the same GPOs to

Continued on the next page

log on to the RD Session Host servers and to their local desktops, their user settings might not mesh well with a shared environment. If that's the case, then you'd pick Replace Mode. If you want the user experience to be as similar as possible for both local and remote logons, then Merge Mode might be more appropriate because it will preserve user-specific policies. The main thing you'll need to watch out for is that GPO settings from the GPOs applied to the user do not cause problems for your user when she is logged on to an RD Session Host server (or pooled VM). Using Merge Mode is more work because it requires a lot of considering of individual policies and their effect on a remote workspace.

Fine-Tuning GPOs with Security Filtering

A GPO works because by default, anyone in the Authenticated Users group can use it, and Authenticated Users means "anyone who is logged on to the domain." (Computers also log on to the domain, so they're also members of Authenticated Users.)

If you have groups of users with specific needs controlled by Group Policy, you can create a User Policy for each user group and then use Security Filtering to apply each User GPO to a specific user group. For example, this technique could come in handy if you give access to multiple applications in one farm but only have licensing enough for a subset of users. You could block certain users from running that application, thus meeting software licensing compliance requirements. To narrow the scope of to whom (or to what) these policies will apply, double-click the GPO in the Group Policy Objects folder and navigate to the Scope tab in the right pane. In the Security Filtering section on this tab, modify Security Filtering to include the specific users group for which you want settings in the GPO to apply, as shown in Figure 5-15.

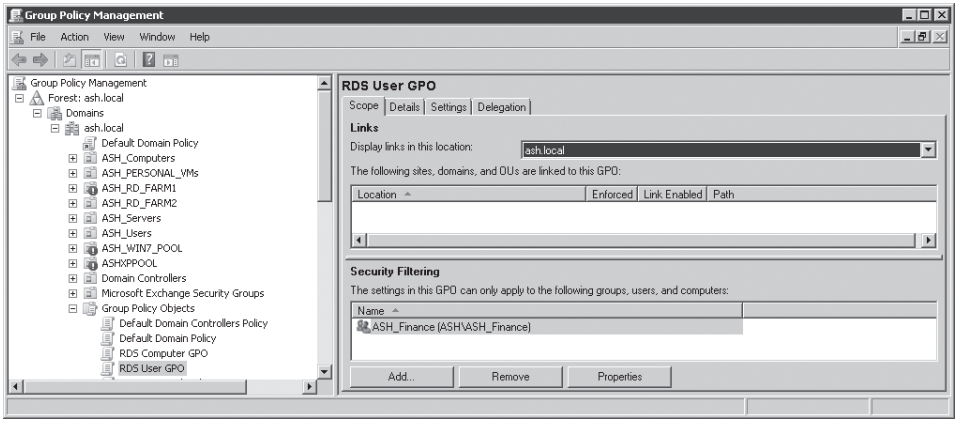


FIGURE 5-15 Add users to the GPO Security Filtering section of the ASH TS Users Policy.

Using Group Policy to Define the Roaming Profile Share

After you have a Group Policy infrastructure set up, you can create a policy to create roaming profile folders in the proper folder share location automatically.

The Group Policy setting to set the path for RDS roaming profiles is a computer setting. Right-click your Computer Policy GPO and choose Edit. Expand the GPO to Computer Configuration | Policies | Administrative Templates | Windows Components | Remote Desktop Services | Remote Desktop Session Host | Profiles. In the pane at right, double-click Set Path For Remote Desktop Services Roaming User Profile, shown in Figure 5-16.

> ***NOTE*** **It might seem counterintuitive to set the RDS roaming profile path for computers, not for users. But the RD Session Host servers must know where to find the roaming profile so the User Profile Service can load it when a user logs on.**

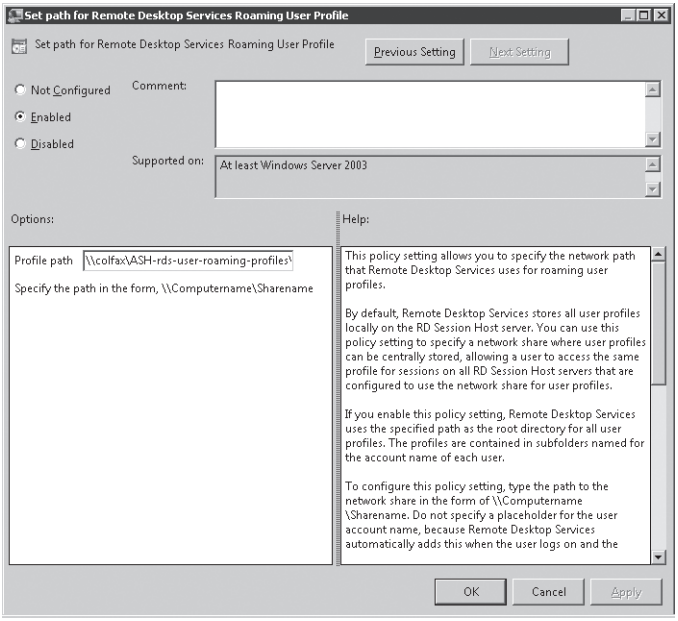


FIGURE 5-16 Set the path for Remote Desktop Services Roaming User Profile storage.

Select the Enabled option and type the RDS roaming profile share location in the Profile Path text box. If you use Group Policy to set the RDS roaming profile path, then the profile folders that are created take the form of *username.domainname*.V2; you do not need to add the %username% variable, the domain name, or the .V2 extension. This is in contrast to defining the path to the Remote Desktop Services profile folder by editing the user account properties through scripting or through Active Directory Users And Computers, where you must specify the *username* and *domainname* variables to create the folder properly.

NOTE **If you already have profiles stored in the profile path and the profile folders do not include the domain name (perhaps they take the form of *username*.V2), change the names to include the domain name. Otherwise, the server will not see the existing profile, and the service will create a new one in the format *username.domainname*.V2.**

If the profile folders are created automatically when the user logs on, then the user gets sole access to the profile and is also set as the owner of the profile folder. To permit administrators to access the profile, enable the following GPO setting: Computer Configuration | Policies | Administrative Templates | System | User Profiles | Add The Administrators Security Group To Roaming User Profiles. With this GPO setting enabled, the following permissions are placed on newly created user folders.

- **User** Full Control, owner of folder
- **SYSTEM** Full Control
- **Administrators** Full Control (This is the local administrators group of the server where the profiles are stored, which also contains the Domain Admins group.)

You can also pre-create user profile folders and set permissions as required. For more information about profile folder permissions, see the section entitled "Converting an Existing Local Profile to a Roaming Profile" earlier in this chapter.

With this GPO setting configured, users accessing the RD Session Host servers in this OU now have a roaming profile created and stored in the designated share.

Configuring Roaming Profile Paths for VMs

Pooled and personal VMs will run client operating systems. Setting an RDS roaming profile path on these machines simply won't work. They are client machines, and for the most part, they should be treated as such. To configure the roaming profile path for client machines, use this GPO setting: Computer Configuration | Policies | Administrative Templates | System | User Profiles | Set Roaming Profile Path For All Users Logging On To This Computer.

Enter the share name where your profiles are stored and add the %username% variable to the end of the path so that each user gets a unique profile folder, as follows.

```
\\servername\sharename\%username%
```

Speeding Up Logons

One of the biggest challenges that IT professionals face in an RDS environment is to provide a user experience that feels as much like a local computer as possible. Users want to log on quickly, work steadily, get their job done, and get out. If they find that they have to wait longer to log on than they like, the Help desk will hear about it, or people will look for ways to circumvent the data center.

Roaming profiles are usually the best choice for RDS. Centralizing the profile on a network share makes it possible to always have the same experience no matter what RD Session Host server or VM a user is logged into—even new ones that were just added. Centralizing also simplifies backups. However, if you don't take steps to avoid it, profiles grow over time. By default, a profile contains not only configuration data but also user documents. Assuming that a user saves files to the folders there for that purpose, the profile will grow. Big profiles slow down logons and logoffs due to the massive amounts of data that must be copied to the remote location.

There are several things you can do to speed logons.

- Take advantage of the new behavior of Group Policy caching among servers in a farm to reduce the time needed for the first login.
- Enable Folder Redirection.
- Manage policy caching.
- Limit profile size.

Let's start with the one that requires no configuration.

Roam Group Policy Cache Between RD Session Host Farm Servers

Group Policy is cached on a computer to speed up logon times. The first time someone logs on to an RD Session Host server, her Group Policy settings won't be cached there. A new feature of Windows Server 2008 R2 copies the Group Policy cache to all servers in a farm. That way, once a user has logged on to one member of the farm, her GP cache will be available on all servers in the same farm.

Enable Folder Redirection

When a user logs on to an RD Session Host server, his roaming profile has to be copied to that RD Session Host server. When the user logs out, the changed profile must be copied back to the roaming profile storage location. Note that you are writing the entire profile back, not just the changes to the profile. Imagine if one of your users saved 30 GB of data in his Documents folder. He would log on to the RD Session Host server and then go get a cup of coffee (or even go to lunch) while waiting for the profile to copy itself to the server. Now imagine if all your users had that much data stored in their Documents folder. If they all come in at 9 A.M. and try to log on to the RD Session Host server, logons could quickly consume all your network bandwidth. Soon the water cooler or break room would be very popular, and no one would get any work done.

Profile caching also suffers if you experience profile bloat. *Profile caching* saves a copy of the user profile on the RD Session Host server so that, if the network is slow to retrieve the saved profile from its file share, the user can still log on using the cached copy. (When you log on to an RD Session Host server, a copy of your profile is saved there as a matter of course. If you enable profile caching, the profile isn't deleted when you log off.) However, if the profiles in the cache are too large, the space allocated for them will fill up, and people won't be al-

lowed to log on because there's no room to store their profiles. There are Group Policies to remove older data in the cache if room runs out, but it's better if you can avoid this problem entirely.

The simplest step that you can take to avoid profile bloat is to enable Folder Redirection. Folder Redirection has two advantages: it keeps user data out of the profile to keep the profile smaller, and it allows differential synching (so that if only part of a file is changed, that part will be saved to the central location, rather than copying the entire file). You'll learn how to set up Folder Redirection in the section "Centralizing Personal Data with Folder Redirection" later in this chapter.

Limit Profile Size

One way to reduce the impact of caching profiles on the RD Session Host servers is to limit the size of the profiles. Although too many profiles can still fill up the hard disk, smaller cached profiles have less impact. To limit profile size, open your RDS User GPO and browse to User Configuration | Policies | Administrative Templates | System | User Profiles. Locate the policy Limit Profile Size and enable it.

If you're redirecting folders, the size of the profile shouldn't be a major concern. NTUSER.DAT is a fairly small file. The exact size depends on the profile, but it's not much; check the size of some representative NTUSER.DAT files to gauge the space needed to allocate space for profiles.

Manage the Profile Cache on RD Session Host Servers

Another way to keep the size of the cache on the RD Session Host servers from getting too large is to delete old copies of the user roaming profiles. You can also limit the profile cache size if you're concerned about running out of room on the servers.

PROGRAMMATICALLY MANAGING THE CACHE

You can use two computer Group Policy settings to delete unused cached profiles on RD Session Host servers in the RD Session Host Farm OU automatically. Both policies are located in Computer Configuration | Policies | Administrative Templates | System | User Profiles.

- **Delete Cached Copies Of Roaming Profiles** Enabling this setting deletes a user's cached profile when the user logs off. This setting ensures that the loaded profile is always the most recent. However, the cached profile provides a fallback configuration to load if the actual profile isn't available for some reason. If you delete cached profiles, then if the actual profile can't be loaded, the user will get a temporary profile and any changes he makes to it will be discarded when the user logs off.
- **Delete Unused Profiles** Windows Server 2008 R2 has a new Group Policy setting that limits the size of the overall roaming profile cache (located in the %SystemDrive%\Users directory). If the size of the profile cache exceeds the configured size, RDS deletes the least recently used copies of roaming profiles until the overall cache goes

below the quota. The policy setting is found in Computer Configuration | Administrative Templates | Windows Components | Remote Desktop Services | Remote Desktop Session Host | Profiles | Limit The Size Of The Entire Roaming User Profile Cache.

NOTE **Although you can apply the Delete Cached Copies Of Roaming Profiles GPO setting to pooled and personal VMs, it doesn't accomplish anything useful. Pooled VMs get rolled back (if set up to do so) when a user logs off, so the user profile cache is cleared as part of the rollback function. And personal VMs are, well, personal. They will have one profile cached on the machine. You will have enough room for one user profile cache in this instance. Deleting the profile cache on a personal desktop will just increase logon time and has no advantages.**

Another way to make sure that your servers do not run out of disk space due to an overgrown profile cache is to put a cap on the cache size. If the size of the entire cache exceeds the limit set by this policy, the server will delete the oldest profile in the cache until the overall size drops below the threshold you set. The GPO setting is located at Computer Configuration | Administrative Templates | Windows Components | Remote Desktop Services | RD Session Host | Profiles | Limit The Size Of The Entire Roaming User Profile Cache.

Enable this setting and enter the following numbers.

- A monitoring Interval (in minutes): The interval at which the profile cache size is checked.
- Maximum cache size (in GB): This is the threshold. If the cache grows beyond this number, the oldest profiles start getting deleted.

DELETING CACHED PROFILES MANUALLY

Deleting cached profiles manually sounds too simple to bother explaining, but it's more subtle than it might appear. Cached profiles are kept in the %SystemDrive%\Users directory. However, the obvious approach doesn't work. If you do the obvious—look at the profiles, check the dates, note that some profiles haven't been used in a while, and delete them—you will prevent the owners of those deleted profiles from being able to log on to the RD Session Host server and load their roaming profiles, at least without some help from you. See the section entitled "The Consequences of Deleting a Profile Folder from Windows Explorer" later in this chapter for more information. For now, let's see how you can avoid extra work.

The problem is that cleaning up old profiles isn't just a matter of deleting some old directories. The registry maintains a list of profiles in HKLM\Software\Microsoft\Windows NT\CurrentVersion\ProfileList. Sort through that key (see Figure 5-17), and you'll see entries for everyone who currently has a profile cached on the server. Although the keys themselves are identified by the SIDs of the user accounts, you can see the names of the profile paths by examining the contents of the keys.

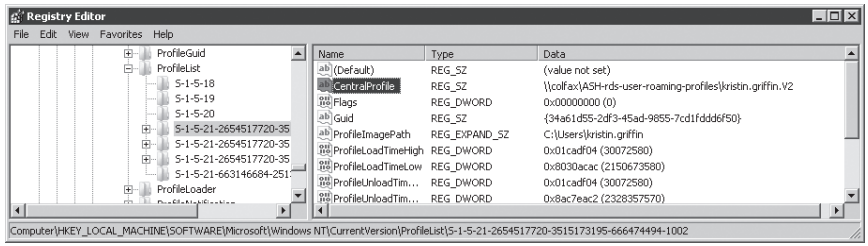


FIGURE 5-17 When you cache a profile on a server, it automatically creates a corresponding registry entry.

> ***NOTE*** **Examining this key can also help you troubleshoot profile problems. If a user seems to be getting his standard profile to log on to the RD Session Host server, check the contents of CentralProfile (see Figure 5-17). If this entry is blank, that person is using a local profile.**

If you just delete the profile from Windows Explorer, the entries in the registry remain, which confuses the server, as explained in the next section.

The cleanest way to delete unused profiles is to let Group Policy delete the old and unused profiles. You can also delete cached roaming user profiles from the User Profiles section of System Properties on the RD Session Host server. Log on to the RD Session Host server as an administrator. Go to Start, Control Panel, System, and click Change Settings. The System Properties dialog box will appear. Select the Advanced tab. In the User Profiles section, click Settings... to open the User Profiles dialog box, shown in Figure 5-18.

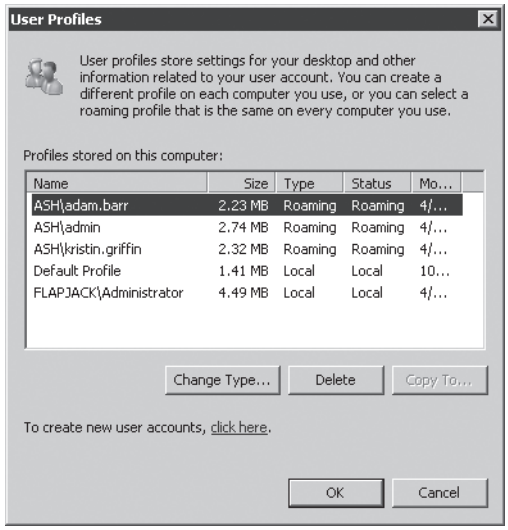


FIGURE 5-18 The User Profiles dialog box displays the profiles stored on the computer.

Highlight the roaming profile that you want to delete and then click Delete. When you see a dialog box confirming that you want to delete the profile, click Yes and the roaming profile cache is deleted. Click OK.

THE CONSEQUENCES OF DELETING A PROFILE FOLDER FROM WINDOWS EXPLORER

Just in case you decide to try deleting a profile folder from Windows Explorer, here's what will happen. If you delete an unused profile folder from Windows Explorer, the next time that user with that folder logs on, he will be unable to load his roaming profile. A temporary roaming profile will be created for him, profile changes that he makes will be discarded at logoff, and Event ID 1511 is logged in the Windows Application event log stating that Windows cannot find the local profile and is logging him on with a temporary profile.

Deleting that directory caused a problem because you didn't clean up the cached profile completely. For each cached profile stored in %SystemDrive%\Users\%UserName%, the User Profile Service creates a registry entry for this profile at HKLM\Software\Microsoft\Windows NT\CurrentVersion\ProfileList, shown in Figure 5-19. This registry key is named according to the user SID.

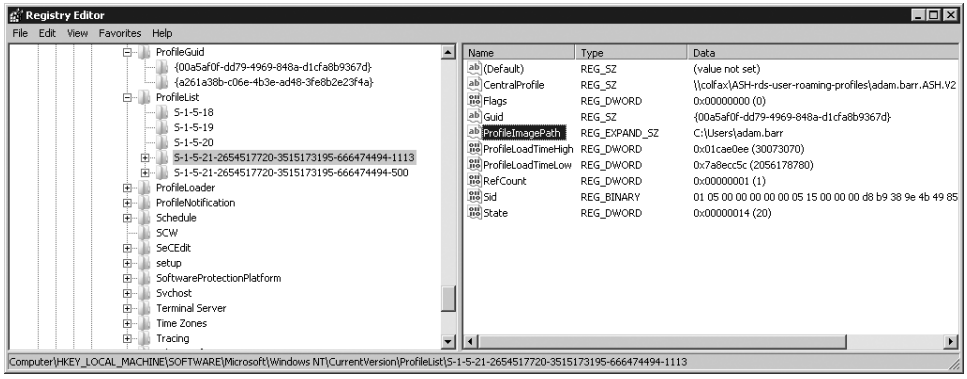


FIGURE 5-19 The RDS roaming profile cache registry entry for user Adam Barr

The ProfileImagePath key in this folder indicates the cache location, which by default is %SystemDrive%\Users\%UserName%. (The network location where the roaming profile is stored is in the CentralProfile key.)

If you delete the user's locally cached profile folder and that user starts a session on that RD Session Host server, he will get a temporary profile. The registry entry corresponding to the user's cached profile is renamed. The SID part stays the same, but it is given an extension of .bak, as shown in Figure 5-20.

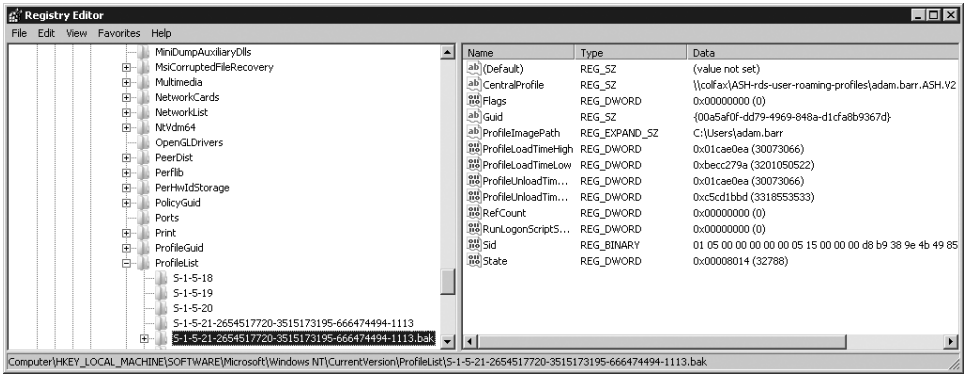


FIGURE 5-20 The old registry key for the profile that was deleted incorrectly now has a .bak extension.

In addition, a new key is created in its place. The newly created registry entry is named after the user SID just as before. However, the ProfileImagePath key inside the new folder now points to %SystemDrive%\Users\TEMP, as shown in Figure 5-21.

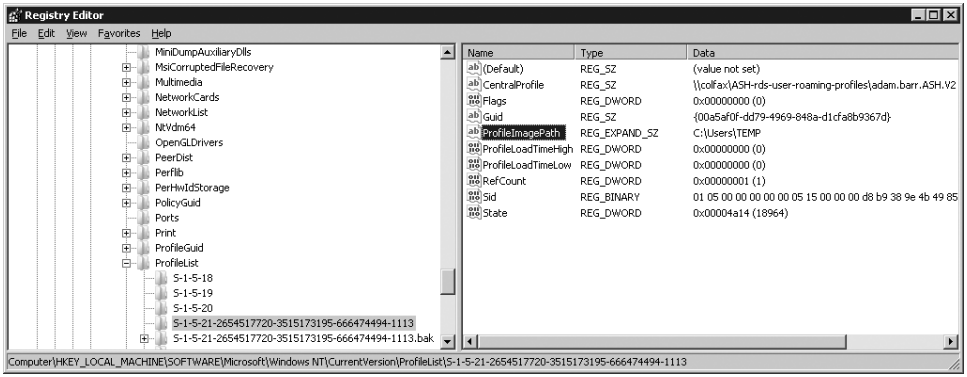


FIGURE 5-21 A new registry entry is created, but the ProfileImagePath key points to %SystemDrive%\Users\TEMP.

Therefore, the entry that used to work now has a .bak extension and is not usable, and the profile actually being used is a temporary profile. When the user logs off, his temporary profile is not copied back to the central profile storage location on the fileserver.

Deleting the profile from the System Properties dialog box User Profiles section no longer works either. Most likely, the profile will not even be listed in the dialog box. If it is, it most likely means that the user has not logged off completely. If you do manage to select it and click Delete, you get an error message: "Profile not deleted completely. Error – The system cannot find the file specified."

To rectify this, you must manually delete the abandoned registry entry that has the .bak extension. You might also need to reboot the server. Only then can the user log on to the RD Session Host server and have his roaming profile correctly cached once again on the server.

Centralizing Personal Data with Folder Redirection

The single biggest thing that you can do to affect profile size, simplify backups, and speed logons and logoffs is to redirect user-specific storage out of the user profile. By default, user data folders such as Documents are in the profile, but they don't have to be. Instead you can create a pointer to a network share where the data actually lives. Users will still store files in their personal folders, but the user data won't be roamed, so it will not affect the time required to load the profiles at logon.

Folder redirection is fundamentally very simple. If you go to HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders, you'll see every folder in your profile and the current location of that folder. If Folder Redirection is not turned on, then all entries will look like this: %USERPROFILE%\Music. The goal is to get rid of the %USERPROFILE% variable and replace it with a new location.

You can't redirect all folders, but you can redirect the ones with the biggest impact on profile size. These folders are

- **AppData(Roaming)** Contains a user's application settings that are not computer-specific and therefore can roam with the user
- **Desktop** Contains any items a user places on his desktop
- **Start Menu** Contains a user's Start menu
- **Documents** Contains documents saved to the default location
- **Favorites** Contains a user's Internet Explorer favorites
- **Music** Contains a user's music files saved to the default location
- **Pictures** Contains a user's pictures saved to the default location
- **Video** Contains a user's video files saved to the default location
- **Contacts** Contains a user's contacts saved to the default location
- **Downloads** Contains a user's downloads saved to the default location
- **Links** Contains a user's Favorite links from Internet Explorer
- **Searches** Contains a user's saved searches
- **Saved Games** Contains a user's saved games

Before you redirect these folders, you need a place to redirect them to. Create a shared folder on the server where you want to store the redirected folders and set permissions on this folder according to the user profile folder permissions that were described in Table 5-5.

To redirect the folders to this share, open the GPMC, create or select an existing user GPO, right-click it, and choose Edit. Go to User Configuration | Policies | Windows Settings | Folder Redirection, as shown in Figure 5-22.

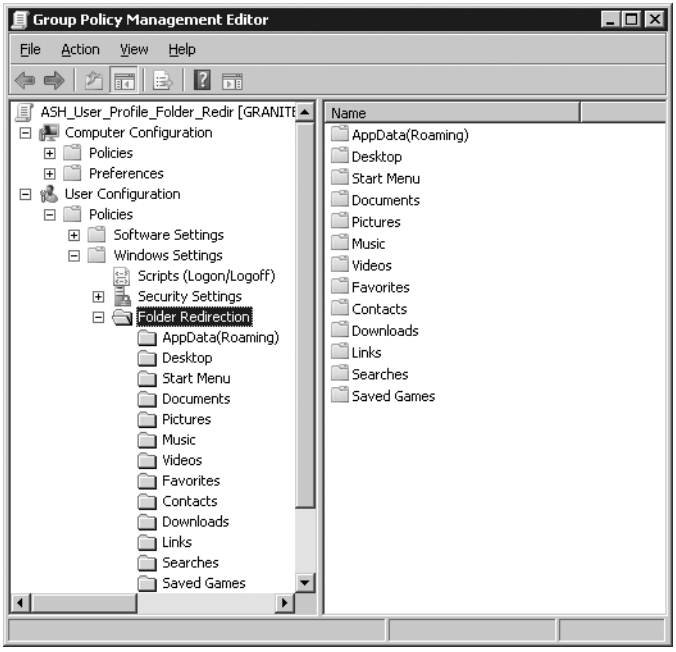


FIGURE 5-22 Set the Folder Redirection policy.

Right-click the AppData(Roaming) folder and choose Properties to open the dialog box shown in Figure 5-23.

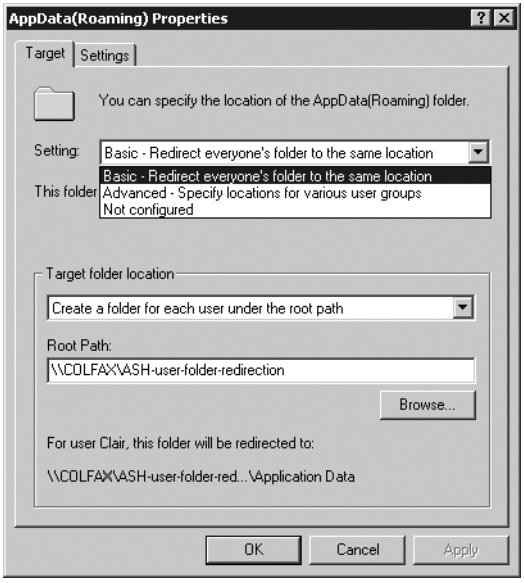


FIGURE 5-23 AppData(Roaming) Folder Redirection properties dialog box

To specify the location of the AppData(Roaming) folder, choose between two options in the Setting drop-down menu.

- **Basic Redirect Everyone's Folder To The Same Location** This means just what it says; all AppData(Roaming) folder data for every user will go to the same location.
- **Advanced Specify Locations For Various User Groups** To store user data in different locations based on user group membership, choose this option.

The menu contents will vary depending on the type of folder redirection you choose. If you choose Basic, then you get a Target folder location drop-down menu with three choices.

- **Create A Folder For Each User Under The Root Path** Choose this option to put each user's profile data into a folder under the root path named according to the user name. In the Root Path text box, specify the location of your designated Folder Redirection share. In most cases, this is the best option.
- **Redirect To The Following Location** Choose this option to redirect all user data to the same location. You'd do this if you wanted all users to use the same Desktop or Start Menu folder. Choose this option only if you want everyone to write to the same user-specific folders.
- **Redirect To The Local Profile Location** Don't choose this option. Your profiles roam, and you want your profile folders redirected to the network share.

Click the Settings tab, as shown in Figure 5-24.

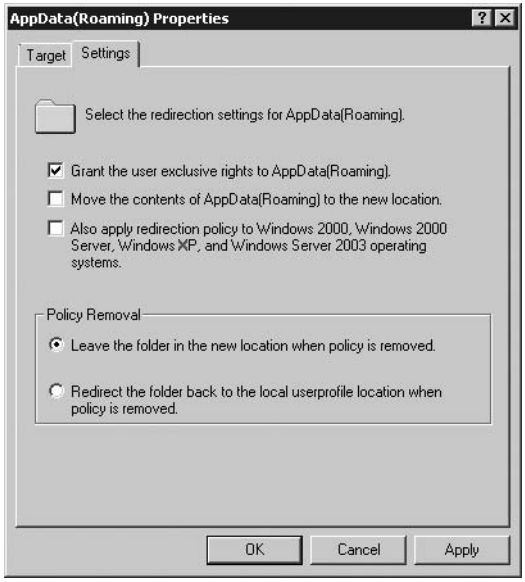


FIGURE 5-24 Grant The User Exclusive Rights To AppData(Roaming) is enabled by default. Clear this check box to let administrators manage the redirected folder.

By default, Grant The User Exclusive Rights To AppData(Roaming) is enabled. If you leave it this way, then the user will own this folder, and only she will be able to access this data. To enable managing this folder, clear this box so that the rights from the parent folder will be inherited. For example, if you give Domain Admins full control of the parent folder, then this group will have access to the redirected user folders as well.

If your users already have these folders before you set up Folder Redirection, then you must set up the existing folders in one of two ways (otherwise, Folder Redirection will fail).

- The user needs to be the owner of the folder and can be granted exclusive rights to the folder.
- If the user does not need to be the owner of the folder, clear this box.

All the folders listed in this GPO section have the same choices to pick from, except for the Pictures, Music, and Video folders. These folders have an extra setting that you can choose for the location of the folder: Follow The Documents Folder. This means that these folders will be stored in the user's Documents folder, wherever that folder is redirected.

To move the contents of the existing folder to the new folder outside the profile, select the Move The Contents Of "The Name Of The Folder Being Redirected" check box to the new location.

ON THE COMPANION MEDIA **When redirecting a folder using Group Policy, one of the options is Move The Contents. Unless you select this option, a duplicate link will be left behind, even when that folder is completely empty, meaning that users will see two Documents folders, two Music folders, and so forth. For tips on how to avoid the "duplicate link" problem, see *http://blogs.technet.com/deploymentguys/archive/2008/05/01/dealing-with-duplicate-user-profile-links-in-windows-vista.aspx*. You can also find the link on this book's companion media.**

Sharing Personal Folders Between Local and Remote Environments

Because the RemoteApp programs are designed to blur the line between the remote computer and the local computer, it might make sense for you to help this along by using the same folder to store user-specific documents. This eliminates the problem of having to remember whether you were saving a file from a local or a remote application to know where the file would be stored.

Sharing Folders Between Windows Server 2003 and Windows Server 2008 R2 Roaming Profiles

The easiest profile environment to manage is homogenous: All users work only in RD Session Host servers, and all servers of sessions are running Windows Server 2008 R2. However, there are good reasons why you might need to support both V1 and V2 profile structure at the same time.

- Some users work both on the RD Session Host server and on VMs running Windows XP (perhaps because they're using RemoteApp on Hyper-V).
- You're migrating to Windows Server 2008 R2 RDS from Windows Server 2003 Terminal Services, and some of the older servers are still in use as you convert.

V1 profiles and V2 profiles are not compatible. Therefore, if you have some active 2003 RD Session Host servers, you will need to keep two sets of profiles for your users—one to log on to the 2003 servers and one to log on to the 2008 servers. And you might need even more profiles if users are also using pooled and personal VMs, and/or RemoteApp programs on Hyper-V. However, Folder Redirection can be used to bridge the gap.

Not all 13 folders that can be redirected in Windows Server 2008 R2 can be redirected in Windows Server 2003, but some can. You can share the data in these folders between the 2003 profiles and the 2008 profiles. On the Settings tab of each folder in the Folder Redirection container is an option called Also Apply Redirection Policy To Windows 2000, Windows 2000 Server, Windows XP And Windows Server 2003 Operating Systems. For some folders, this option is available, but on others (the ones that will not redirect for downlevel operating systems), it appears dimmed and is unavailable. Table 5-6 shows which of the folders can be redirected for Windows 2000, Windows XP, and Windows Server 2003.

TABLE 5-6 Profile Folder Redirection Capabilities for Various Versions of Windows

FOLDER	CAN THE FOLDER BE REDIRECTED FOR EARLIER OPERATING SYSTEMS?	DETAILS
AppData(Roaming)	Yes	If you enable the setting Also Apply Redirection Policy To Windows 2000, Windows 2000 Server, Windows XP, And Windows Server 2003 Operating Systems**,** the following folders within AppData(Roaming) are not redirected: Start Menu, Network Shortcuts, Printer Shortcuts, Templates, Cookies, and Sent To. These folders are redirected if you do not enable this setting.
Desktop	Yes	

Continued on the next page

FOLDER	CAN THE FOLDER BE REDIRECTED FOR EARLIER OPERATING SYSTEMS?	DETAILS
Start Menu	Yes	In Windows Server 2003, the contents of the Start Menu folder are not copied to the redirected location. It is assumed that the Start Menu folder has been pre-created. Therefore, if you do not pre-create the Start Menu folder and place it in the redirected location, the default Start Menu folder located in the user's Windows Server 2003 roaming profile location is used instead.
Documents	Yes	
Pictures	Depends	If the check box for Documents is selected, this folder will follow the Documents folder for earlier operating system profiles. If Documents is not redirected, however, then this folder cannot be redirected.
Music	Depends	If the check box for Documents is selected, this folder will follow the Documents folder for earlier operating system profiles. If Documents is not redirected, then this folder cannot be redirected.
Video	Depends	If the check box for Documents is selected, this folder will follow the Documents folder for earlier operating system profiles. If Documents is not redirected, then this folder cannot be redirected.
Favorites	No	NA
Contacts	No	NA
Downloads	No	NA
Links	No	NA
Searches	No	NA
Saved Games	No	NA

ON THE COMPANION MEDIA **For more information on Windows Server 2003 and Windows XP Profiles and Folder Redirection, see *http://technet2.microsoft.com/windowsserver/en/library/06f7eebc-2ebb-47c5-8361-1958b58078cc1033.mspx?mfr=true*. You can also find the link on this book's companion media.**

NOTE **Some custom applications might not respond well to having the AppData folder redirected. But not redirecting AppData could lead to profile bloat, especially if your applications write a lot of data to this location. For situations like this, consider using App-V to deploy the problem application. For technical resources on sequencing with App-V, see *http://www.microsoft.com/systemcenter/appv/dynamic.mspx*.**

Setting Standards with Mandatory Profiles

One issue with roaming profiles is that users can change them. On the one hand, that's the point. On the other hand, changes can cause problems. If users can change their profiles, they can delete icons, accidentally resize their toolbar so that it disappears, add wallpaper that slows their logon time, and so on.

One way to avoid this is to set policies controlling what users can and cannot do, and Chapter 7, "Molding and Securing the User Environment," explains how to do this. Another way to prevent users from making permanent changes to their profile is to make the user profile read-only. A user can change settings, but those settings will not be saved when the user logs off the RD Session Host server.

Profiles that don't change are called *mandatory profiles*. Mandatory profiles on a central store are copied to the RD Session Host server at logon, but they are not copied back at logoff. Any profile changes that occur are discarded at the end of the user session. Many companies will not implement mandatory profiles because users find them too constricting, but combined with Folder Redirection, they might give your users enough flexibility. Some third-party profile solutions also require the use of mandatory profiles—it depends on how the products are implemented.

Although it's possible to give every user a unique mandatory profile, it's not ideal. One of the best things about mandatory profiles is that because the profile will never be changed, all users can use a single mandatory profile, creating much less maintenance work for administrators. If a change needs to happen to the profile, there is only one place to make the change, instead of many if every user had his or her own individual profile.

Mandatory profiles are great in many respects, but you need to be careful when implementing them to make sure each user who logs on will not be susceptible to registry changes from other users. See the Direct from the Field sidebar that follows for more details.

DIRECT FROM THE FIELD

Mandatory Profiles: Insecure By Default?

Helge Klein
IT Architect, sepago

Mandatory profiles are generally considered fast and secure because they usually are small in size and cannot be modified by the user. Although that is true—mandatory profiles stay pristine indefinitely—there is more to security than read-only access.

Mandatory profiles are a variant of roaming profiles: A master copy on a file server is copied to the RDS session host during logon. The resulting local copy is secured with file system ACLs that grant full access to the user, but to no one else (except administrators and SYSTEM). All is safe and secure—except in the case of mandatory profiles.

A user profile consists not only of file system data, but also of a registry hive (stored in the file NTUSER.MAN) that is mounted to HKU\<SID> and accessible from within a session via the well-known name HKCU. In contrast to the file system, registry permissions are not changed during logon because that is not necessary—at least with roaming profiles where the master copy of each hive already has the correct permissions.

Not so with mandatory profiles. The creation of a mandatory profile involves changing registry permissions on the master copy to full access for "Everyone." And because many users are logged on simultaneously to an RDS session host, each server's registry consists of many users' hives that are readable and writeable by everyone, not just the owner of the individual user profile.

So on an RD Session Host server where mandatory profiles are used, a user can simply open Regedit (if not blocked from doing so), navigate to HKU\<Some other user's SID>, and read/write at will.

Consequences

Users being able to read/write somebody else's HKCU hive poses a potentially grave security problem. At least two types of attacks can be envisioned: eavesdropping and damaging. Here are some simple examples.

Many applications store a list of most recently used (MRU) files in HKCU (for example, Word: HKCU\Software\Microsoft\Office\12.0\Word\File MRU). By reading such lists, attackers can gain information about which documents another user is editing.

Applications and the operating system itself need and expect write access to HKCU. Because a user always has write access to HKCU, programs do not handle

the absence of such permissions well. By changing permissions on another user's hive (for example, removing write access), an attacker could effectively break another user's session, making it impossible to start and use even the most trivial programs—most applications that store their settings in HKCU would be affected.

How to Fix

The following workarounds can help fix this security vulnerability.

1. **Make sure that remote registry editing is limited to administrators.**
2. **Block access to the registry via software restriction policies. This includes, but is not limited to, Regedit.exe, Cmd.exe, Reg.exe, scripts and batch files, and other custom (downloaded) tools. In essence, in order to avoid this problem exclusive white-listing is required.**
3. **Re-ACL (change the security permissions on) each registry hive after it is loaded and replace "Everyone" with the current user.**

Converting Existing Roaming Profiles to Mandatory Profiles

Setting up mandatory profiles is very similar to setting up roaming profiles using Group Policy. To convert a roaming profile to a mandatory profile, you first need to have roaming profiles working, either by setting the RDS Roaming Profile path in the user's account properties in Active Directory Users and Computers, or by using Group Policy. For information on how to set up roaming profiles, see the section entitled "Using Group Policy to Manage Roaming Profiles," earlier in this chapter.

Assuming you have roaming profiles implemented, when a user logs on, her profile is stored in a subdirectory of the designated roaming profile share. To make the user's profile mandatory, in the user's profile folder, locate NTUSER.DAT and change its extension to .man (see Figure 5-25). Then change the NTFS permissions for the user from Full Control to Read & Execute (so she can't change the extension back). The next time the user logs on, she will be using a mandatory profile.

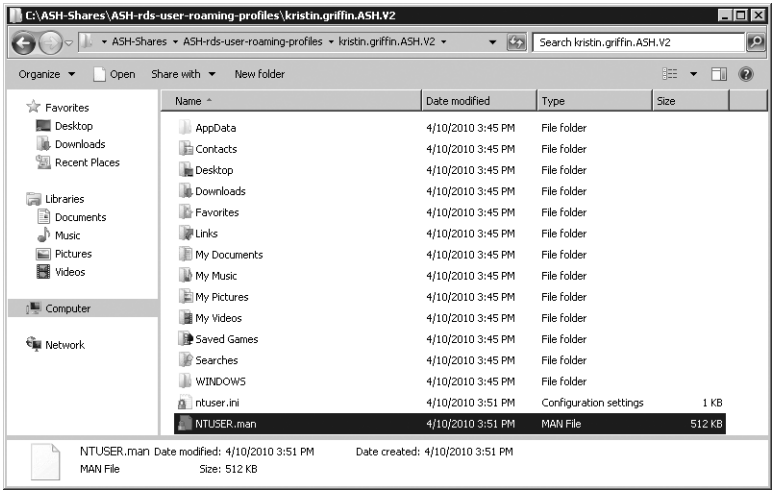


FIGURE 5-25 To convert a roaming profile to a mandatory profile, change its extension.

No changes that the user makes to the profile will be saved. But combining mandatory profiles with Folder Redirection will give users some control over their session and allow them to change their Favorites, Documents, Desktop, and other settings without compromising the configuration data loaded in HKCU.

Creating a Single Mandatory Profile

If you have many users, you probably won't want to convert each roaming profile to a mandatory one—that would negate one of the main reasons to implement mandatory profiles: less configuration and maintenance. To give everyone the same experience, you can create one mandatory profile for everyone to use. Here are the steps to do so.

1. Create a network share to store the mandatory profile (for example: //Colfax/ASH-Mandatory-Profile). Make sure to configure the permissions on this folder correctly. Table 5-7 and Table 5-8 outline the necessary share and NTFS permissions that need to be set on this folder.

TABLE 5-7 Share Permissions for a Mandatory Profile Storage Folder

USER ACCOUNT	SHARE PERMISSIONS
Administrators	Full Control
Authenticated Users	Read

TABLE 5-8 NTFS Permissions for User Accounts for a Mandatory Profile Storage Folder

USER ACCOUNT	NTFS PERMISSIONS
SYSTEM	Full Control, this folder, subfolders, files
Administrators	Full Control, this folder, subfolders, files, Owner
Authenticated Users	Read & Execute, this folder, subfolders, files

2. Create a folder within the folder created in Step 1, name it something appropriate to indicate it is a mandatory profile, and append the .V2 extension (for example: ASH.RDS.MAN.V2).
3. Because using the Copy To button now works only for the Default user profile, this is the profile you will copy to the share you created in Step 1. On the RD Session Host server, from Server Manager, click Change System Properties and select the Advanced tab. In the User Profiles section, click Settings. Highlight the Default User, and click Copy To. In the Copy To dialog box, type or browse to the shared folder location that you created in Step 1. Click Permitted To Use, add Everyone, and click OK.

NOTE **If you choose to create a customized mandatory profile, use Sysprep to overwrite the Default User profile on the machine that you will copy from. For more on customizing the default user profile and using the Copy To button, and how to use Sysprep to customize the Default User Profile, see the sections earlier in this chapter entitled "Converting an Existing Local Profile to a Roaming Profile" and "Customizing a Default Profile."**

4. Rename NTUSER.DAT in the resulting profile (in the file share created in Step 1) to NTUSER.MAN. You will need to change the folder options to show hidden files and folders to see this file.
5. Create appropriate GPOs by doing the following.
   - Edit the Computer GPO setting as follows: Computer Configuration | Policies | Administrative Templates | Windows Components | Remote Desktop Services | Remote Desktop Session Host | Profiles | Set Path For Remote Desktop Services Roaming User Profile to point to the share created in Step 2, for example: //colfax/ash-rds-mandatory-profile/ASH.RDS.MAN). Do not include the .V2 extension.
   - Enable the Computer GPO policy setting as follows: Administrative Templates | Windows Components | Remote Desktop Services | Remote Desktop Session Host | Profiles | Use Mandatory Profiles On The RD Session Host Server
   - Enable the Computer GPO settings as follows: Computer Configuration | Policies | Administrative Templates | System | User Profiles | Add The Administrators Security Group To Roaming User Profiles

6. Apply the GPOs to the RD Session Host Server OU (in Group Policy Manager on a domain controller).
7. Reboot the RD Session Host servers and test by logging in as a regular user.

Creating a Safe Read-Only Desktop

One curious side effect to not being able to save anything to a mandatory profile is that any folders remaining in the profile (that is, not redirected) will not save changes either. For example, if you do not redirect the Desktop folder and if users save files to the desktop, those files will be discarded when they log off. There won't be any error, and the file will be on the desktop during the session, but the files won't be there when the users log on again. To put it mildly, this could be confusing. However, if you're using Remote App programs, you don't really want people saving files to the desktop because not being able to see the desktop will make those files hard to find.

To keep the desktop read-only but make sure people *know* it is read-only, redirect the desktop to a read-only folder as described in the section entitled "Centralizing Personal Data with Folder Redirection" earlier in this chapter. This will both prevent users from saving files to the desktop (which you want) and alert them to the fact that they can't save files to the desktop (which you also want). If they try, they will get an error. They still can't save anything to the desktop, but at least they will *know* that they can't.

Decrease Logon Times with Local Mandatory Profiles

The main reason to house a mandatory profile on a network share is to make it easier to update when you have a farm environment. But it's also worth noting that logon times can be decreased significantly by keeping a mandatory profile local to the server because the profile doesn't get pulled down from the network share when the user logs on.

Maintaining local mandatory profiles is more work, because any changes to the mandatory profiles will need to be made to the mandatory profile on each server. But the increase in logon speed might make this worthwhile to you, especially if you have only a few RD Session Host servers in a farm or you don't often need to change the profile. Again, testing this fully in your environment will tell you if it makes sense for your setup.

To use local mandatory profiles, perform the following steps.

1. Create a folder on each machine called something like "Mandatory_Profile.V2" and set the appropriate NTFS profile folder permissions as specified in Table 5-8.
2. Copy a default profile to the new Mandatory Profile folder, giving Everyone permission to use it when you perform the copy.
3. Convert this local profile to a mandatory profile by changing the extension of NTUSER.DAT to make it NTUSER.MAN.

4. Enable the GPO setting as follows: Computer Configuration | Policies | Administrative Templates | Windows Components | Remote Desktop Services | Remote Desktop Session Host | Profiles | Use Mandatory Profiles On The RD Session Host Server.
5. Enable the Computer GPO setting as follows: Computer Configuration | Policies | Administrative Templates | Windows Components | Remote Desktop Services | Remote Desktop Session Host | Profiles | Set Path For Remote Desktop Services Roaming User Profile. Point to the local mandatory profile location, such as C:\Mandatory_Profile. Do not include the .V2 extension.
6. Do this on each machine in the farm or pool.

Profile and Folder Redirection Troubleshooting Tips

Many people find the combination of RD Session Host servers and profiles daunting. And it's true—things don't always work the way you expect them to. Table 5-9 describes some common errors, possible solutions, and the sections in the chapter where you'll learn how to fix each problem.

TABLE 5-9 Profiles and Folder Redirection Troubleshooting Tips

PROBLEM	SOLUTION	ADDITIONAL INFORMATION IN THIS CHAPTER
Policies appear to be set correctly, but aren't being applied.	Force a policy update by using Gpupdate or by rebooting.	See the sidebar entitled "Updating Group Policy."
Folders are not being redirected to the proper location or roaming profiles are not being loaded.	Check event logs to make sure that share is available on the network and has appropriate permissions.	See the sections entitled "The Consequences of Deleting a Profile Folder from Windows Explorer" and "Centralizing Personal Data with Folder Redirection."
Group Policy settings aren't being applied to the right computers, groups, or users.	Check the security filters and make sure that you've included the correct groups.	See the section entitled "Fine-Tuning GPOs with Security Filtering."
Folders from profiles from earlier operating systems aren't redirecting properly, but Windows 7 and Windows Server 2008 R2 profile folders are redirecting.	Make sure you've enabled earlier Folder Redirection for that GPO.	See the section entitled "Sharing Folders Between Windows Server 2003 and Windows Server 2008 Roaming Profiles."

Continued on the next page

PROBLEM	SOLUTION	ADDITIONAL INFORMATION IN THIS CHAPTER
Users cannot load their roaming profiles when they log on, and they see a message that they will be logged on with a temporary profile.	You might have deleted the cached profile manually using Windows Explorer. Delete the old registry keys and use tools such as the profile management utility or Delprof to delete profiles.	See the section entitled "Deleting Cached Profiles Manually."
Testing Mandatory Profiles returns the error "Access is denied."	Make sure you set the Everyone group to be permitted to use the profile when you use the Copy To button to create the mandatory profile. If necessary, delete the profile that is not working and redo it.	

Summary

Although roaming profiles (read-write or read-only) are often the best model for storing user profiles in an RDS environment, the complications involved in making them work *well* can be daunting. This chapter has explained how profiles work, including how the User Profile Service loads and saves configuration data. You've learned about best practices, including how to keep profiles manageable in size to speed user logons and how Folder Redirection and profile caching also contribute to faster logons. You've seen how to set up Group Policy to enable automatic profile creation and how to use security filtering and loopback policy processing to ensure that the policies are applied correctly with RDS. Finally, you've learned how to set up and use mandatory profiles with RDS and how to prevent users from losing files when using mandatory profiles.

- There are three types of profiles: local, roaming, and mandatory (including super-mandatory).
- Combining roaming profiles with Folder Redirection is generally the best way to store user data in remote environments. Folder Redirection is very important for keeping logon times short and profile sizes small.
- Mandatory profiles work best when you don't want to save any changes to the profile and have prevented users from writing files to profile folders.
- Profiles don't merge—they overwrite. For best results, open only one copy of the user profile at a time. For this reason, you should generally not use the same roaming profile for both local logons and RD Session Host server logons.

- Implementing Group Policy correctly from the beginning is key to making roaming profiles work.
- Folder Redirection is very important to making profiles work properly, as follows.
  - Folder Redirection keeps profiles small.
  - Folder Redirection reduces the data that must be written back to a file stored in a profile folder.
  - Using Folder Redirection, you can share folders between two profiles for better integration of local and remote user experiences.
  - If using mandatory profiles, you must use Folder Redirection to allow users to save files to any of their normal document storage locations (for example, Documents and Favorites).

Additional Resources

The following resources will extend your knowledge of topics addressed in this chapter. All links are available to you on this book's companion media.

- For more information on user profile management (with or without RDS), read the following.
  - "Managing Roaming User Data Deployment Guide," available online at *http://technet.microsoft.com/en-us/library/cc766489%28WS.10%29.aspx* and for download from *http://go.microsoft.com/fwlink/?LinkId=73760*.
  - "Using User Profiles in Windows Server 2003," located at *http://technet2.microsoft.com/windowsserver/en/library/23ee2a30-5883-4ffa-b4cf-4cfff3ff8cb71033.mspx?mfr=true*.
- For more information about how to configure device redirection, see Chapter 6, "Customizing the User Experience."
- To learn how to lock down the server, see Chapter 7, "Molding and Securing the User Environment."
- For more information about publishing RemoteApp programs, see Chapter 9, "Multi-Server Deployments."
- For more information about enabling RD Session Host server farms with RD Connection Broker and multi-server management, see Chapter 9.

Index

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

About the Authors

KRISTIN GRIFFIN was born in California and grew up a military brat, part of a loving and happy family. She has worked with Terminal Services/Remote Desktop Services since Windows 2000 and has implemented RDS for a diverse set of customers, including distributors, law offices, and commercial contracting firms. Formerly a senior IT consultant for a Virginia-based Internet and application service provider, she is now a Seattle-based independent consultant and author. Kristin was honored to receive the Microsoft MVP award for Remote Desktop Services beginning in 2009. You can find her answering questions on the Microsoft RDS Technet Forum (*http://social.technet.microsoft.com/Forums/en/winserverTS/threads*). She also keeps a blog concentrated on RDS tips, setup, and troubleshooting advice at blog.kristinlgriffin.com. In her spare time Kristin enjoys photography, computer graphics, camping, traveling, stained glass, woodworking, and buying more tools from the hardware store. Most of all she enjoys being with her family. She takes her German shepherd dog with her wherever she goes.

A former military brat, **CHRISTA ANDERSON** lived in various places in the western United States until a visit to Virginia ended in a 20-year stay on the East Coast. She returned to Seattle in 2007, where she enjoys the arts and outdoors in a city with a lot of both. Christa's interest in travel and environmental issues contributed to her enthusiasm for presentation remoting, beginning with Citrix WinFrame in the middle 1990s. A former Terminal Services MVP and freelance technical author and speaker for over a decade, she is now a program manager on the Remote Desktop Virtualization team at Microsoft. She promises to talk about something other than the book now.

System Requirements

To use this book's companion CD-ROM, you need a computer equipped with the following minimum configuration:

- Microsoft Windows Server 2008 R2, Windows Server 2008, Windows 7, Windows Vista, Windows Server 2003, or Windows XP
- An appropriate processor depending on the minimum requirements of the operating system)
- At least 2 GB of system memory (depending on the minimum requirements of the operating system)
- A hard disk partition with at least 1 GB of available space
- Appropriate video output device
- Keyboard
- Mouse or other pointing device
- Optical drive capable of reading CD-ROMs

Some items on the companion media have specific requirements. The companion CD-ROM contains numerous links to scripts, tools, Knowledge Base articles, and other information. To view these links, you will need a Web browser and Internet access.

The companion CD-ROM also includes scripts that are written in VBScript (with a .vbs file extension), Windows PowerShell (with a .ps1 file extension) and a few batch files. The Windows PowerShell scripts require that you have Windows PowerShell 2.0 installed. To run these scripts, your system must meet the following additional requirements:. Windows Server 2008 R2 and Windows 7 include Windows PowerShell 2.0. For Windows XP SP3, Windows Vista SP1, and Windows Server 2003 you must download and install Windows PowerShell 2.0. The Windows PowerShell 2.0 download is located at *http://support.microsoft.com /kb/968929*.

- Scripts intended for execution on the local server that depend on specific counters and interfaces will not execute correctly unless the appropriate Remote Desktop Services role service is installed. (For example, a script that queries RD Gateway interfaces will not return results unless the RD Gateway role service is installed.)

The scripts on the CD are not signed. To run them on your computer, we recommend setting the Windows PowerShell Execution Policy to "RemoteSigned." To do this, start Windows PowerShell and type **Set-ExecutionPolicy RemoteSigned**.

This setting will allow you to run the scripts on the CD, and it is more secure than setting this policy to"Unrestricted.

> ***NOTE*** **For more information on using the Set-ExecutionPolicy cmdlet see:** *http://www.microsoft.com/technet/scriptcenter/topics/msh/cmdlets/set-execution-policy.mspx*.

When you run a Windows PowerShell script, you need to provide the full path to the script. To use the VBScript scripts and batch files, double-click them, or execute them directly from a command prompt.

Finally, the CD contains a few files created in Visio 2010, so you will need to have the Visio 2010 viewer to view these files. It also contains a few PDF files so you will need a PDF reader to view these files.

What do you think of this book?

We want to hear from you!

To participate in a brief online survey, please visit:

microsoft.com/learning/booksurvey

Tell us how well this book meets your needs—what works effectively, and what we can do better. Your feedback will help us continually improve our books and learning resources for you.

Thank you in advance for your input!

Microsoft® Press

Stay in touch!

To subscribe to the *Microsoft Press® Book Connection Newsletter*—for news on upcoming books, events, and special offers—please visit:

microsoft.com/learning/books/newsletter