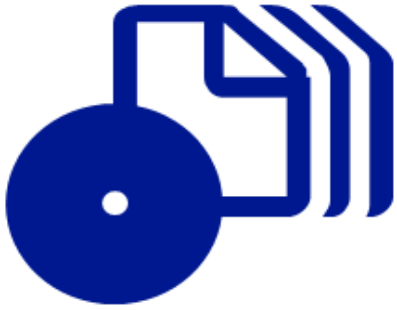
Microsoft

Windows 7

Mitch Tulloch, Tony Northrup,
Jerry Honeycutt, Ed Wilson,
and the Windows 7 Team at Microsoft

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/627000/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: 2009935674

Printed and bound in the United States of America.

1 2 3 4 5 6 7 8 9 QWT 4 3 2 1 0 9

Distributed in Canada by H.B. Fenn and Company Ltd.

A CIP catalogue record for this book is available from the British Library.

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 rkinput@microsoft.com.

Acquisitions Editor: Juliana Aldous
Developmental Editor: Karen Szall
Project Editor: Melissa von Tschudi-Sutton
Editorial Production: Custom Editorial Productions, Inc.
Technical Reviewers: Mitch Tulloch and Bob Dean; Technical Review services provided by Content Master, a member of CM Group, Ltd.
Cover: Tom Draper Design

Body Part No. X15-66448

Contents at a Glance

Contents

Acknowledgments

The authors of the Windows 7 Resource Kit would like to thank the numerous product team members and other experts at Microsoft who contributed hundreds of hours of their valuable time to this project by helping us plan the scope of coverage, providing access to product specifications, reviewing chapters for technical accuracy, writing sidebars that provide valuable insights, and offering their advice, encouragement, and support as we worked on this project. We would particularly like to express our thanks to the following individuals who work at Microsoft:

Aanand Ramachandran, Aaron Smith, Abhishek Tiwari, Adrian Lannin, Alan Morris, Alex Balcanquall, Alwin Vyhmeister, Andy Myers, Anirban Paul, Anjli Chaudhry, Anton Kucer, Ayesha Mascarenhas, Baldwin Ng, Bill Mell, Brent Goodpaster, Brian Lich, Chandra Nukala, Chris Clark, Connie Rock, Crispin Cowan, Darren Baker, Dave Bishop, Denny Gursky, Desmond Lee, Devrim Iyigun, George Roussos, Gerardo Diaz Cuellar, Gov Maharaj, James Kahle, James O'Neill, Jason Grieves, Jason Popp, Jez Sadler, Jim Martin, Joe Sherman, John Thekkethala, Jon Kay, Joseph Davies, Judith Herman, Katharine O'Doherty, Kathleen Carey, Kevin Woley, Kim Griffiths, Kukjin Lee, Kyle Beck, Lilia Gutnik, Lyon Wong, Mark Gray, Michael Murgolo, Michael Niehaus, Michael Novak, Mike Lewis, Mike Owen, Mike Stephens, Narendra Acharya, Nazia Zaman, Nils Dussart, Pat Stemen, Ramprabhu Rathnam, Richie Fang, Rick Kingslan, Scott Roberts, Sean Gilmour, Sean Siler, Sharad Kylasam, Steve Campbell, Thomas Willingham, Tim Mintner, Troy Funk, Varun Bahl, Vikram Singh, and Wole Moses.

Thanks also to Bill Noonan, Mark Kitris, and the CTS Global Technical Readiness (GTR) team at Microsoft for contributing their expertise to this project. The GTR team develops readiness training for Microsoft Commercial Technical Support (CTS) engineers in all product clusters, including Platforms, Messaging, Office Worker, and Developer. GTR creates deep technical content "by engineers, for engineers" with the help of top Subject Matter Experts (SMEs) who are real support engineers from the CTS product clusters.

Finally, special thanks to our outstanding editorial team at Microsoft Press, including Juliana Aldous, Karen Szall, and Melissa von Tschudi-Sutton, for their unflagging energy and tireless commitment to working with us on this challenging

project and making it a success. Thanks also to Jean Findley at Custom Editorial Productions, Inc. (CEP), who handled the production aspects of this book, and to Susan McClung and Julie Hotchkiss, our copy editors, who showed wonderful attention to detail. And thanks to Bob Dean, our tireless technical reviewer.

Thanks everyone!

—*The Authors*

Introduction

Welcome to the *Windows 7 Resource Kit* from Microsoft Press! The *Windows 7 Resource Kit* is a comprehensive technical resource for deploying, maintaining, and troubleshooting Windows 7. The target audience for this resource kit is experienced IT professionals who work in medium-size and large organizations, but anyone who wants to learn how to deploy, configure, support, and troubleshoot Windows 7 in Active Directory Domain Services (AD DS) environments will find this resource kit invaluable.

Within this resource kit, you'll find in-depth information and task-based guidance on managing all aspects of Windows 7, including automated deployment, desktop management, search and organization, software update management, client protection, networking, remote access, and systematic troubleshooting techniques. You'll also find numerous sidebars contributed by members of the Windows team at Microsoft that provide deep insight into how Windows 7 works, best practices for managing the platform, and invaluable troubleshooting tips. Finally, the companion media includes the *Windows 7 Resource Kit PowerShell Pack* and sample Windows PowerShell scripts that you can customize to help you automate various aspects of managing Windows 7 clients in enterprise environments.

Overview of the Book

The six parts of this book cover the following topics:

- **Part I—Overview** Provides an introduction to the features of Windows 7 and an overview of security enhancements for the platform.
- **Part II—Deployment** Provides in-depth information and guidance on deploying Windows 7 in enterprise environments, with particular focus on using the Microsoft Deployment Toolkit 2010 (MDT 2010).
- **Part III—Desktop Management** Describes how to use Group Policy to manage the desktop environment for users of computers running Windows 7 and how to manage specific features such as disks and file systems, devices and services, printing, search, and Windows Internet Explorer.
- **Part IV—Desktop Maintenance** Describes how to maintain the health of computers running Windows 7 by using the eventing infrastructure, monitoring performance, managing software updates, managing client protection, and using Remote Assistance.

- **Part V—Networking** Provides in-depth information concerning core networking, wireless networking, Windows Firewall, Internet Protocol Security (IPsec), remote connectivity using virtual private networking (VPN), Remote Desktop, and Internet Protocol version 6 (IPv6).
- **Part VI—Troubleshooting** Describes how to troubleshoot startup, hardware, and networking issues, as well as how to interpret Stop messages.

Document Conventions

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

Readeraids

The following readeraids are used throughout this book to point out useful details:

READERAID	MEANING
Note	Underscores the importance of a specific concept or highlights a special case that might not apply to every situation
Important	Calls attention to essential information that should not be disregarded
Warning	Warns you that failure to take or avoid a specified action can cause serious problems for users, systems, data integrity, and so on
On the Companion Media	Calls attention to a related script, tool, template, or job aid on the book's companion media 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 features of Windows 7:

SIDEBAR	MEANING
Direct from the Source	Contributed by experts at Microsoft to provide "from-the-source" insight into how Windows 7 works, best practices for managing clients, and troubleshooting tips.
How It Works	Provides unique glimpses of Windows 7 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
%SystemRoot%	Used for environment variables

On the Companion Media

The companion media is a valuable addition to this book and includes the following:

- **Windows 7 Resource Kit PowerShell Pack** A collection of Windows PowerShell modules you can install on Windows 7 to provide additional functionality for scripting Windows administration tasks using Windows PowerShell. For more information, see the section titled "Using the Windows 7 Resource Kit PowerShell Pack" later in this introduction.
- **Sample Windows PowerShell scripts** Almost two hundred sample Windows PowerShell scripts are included to demonstrate how you can administer different aspects of Windows 7 using Windows PowerShell. For more information, see the section titled "Using the Sample Windows PowerShell Scripts" later in this introduction.
- **Additional documentation and files** Additional documentation and supporting files for several chapters are included on the companion media.
- **Additional reading** Sample chapters from other Microsoft Press titles are included on the book's companion media.
- **Windows 7 Training Portal** A link to Windows 7–related products presented by Microsoft Learning.
- **Author links** A page that has links to each author's Web site, where you can find out more about each author and his accomplishments.
- **eBook** An electronic version of the entire *Windows 7 Resource Kit* is also included on the companion media.

Additional information concerning the contents of the companion media can be found in Readme.txt files in various folders.

FIND ADDITIONAL CONTENT ONLINE **As new or updated material becomes available that complements your book, it will be posted online on the Microsoft Press Online Windows Server and Client Web site. The type of material you might find includes updates to book content, articles, links to companion content, errata, sample chapters, and more. This Web site is available at *http://microsoftpresssrv.libredigital.com/serverclient/* and is updated periodically.**

Digital Content for Digital Book Readers: If you bought a digital-only edition of this book, you can enjoy select content from the print edition's companion CD.
Visit *http://www.microsoftpressstore.com/title/9780735627000* to get your downloadable content. This content is always up-to-date and available to all readers.

Using the Windows 7 Resource Kit PowerShell Pack

The Windows 7 Resource Kit PowerShell Pack is a collection of Windows PowerShell modules that you can install on Windows 7 to provide additional functionality for scripting Windows administration tasks using Windows PowerShell. *Modules*—a new feature of Windows PowerShell 2.0—allow Windows PowerShell scripts and functions to be organized into independent, self-contained units. For example, a single module can package together multiple cmdlets, providers, scripts, functions, and other files that can be distributed to users. See the section titled "Disclaimer Concerning Windows PowerShell CD Content" later in this introduction for more information.

The PowerShell Pack contains ten modules that can be installed to add additional scripting capabilities to your Windows PowerShell environment. The additional functionalities provided by these modules are as follows:

- **WPK** Creates rich user interfaces quickly and easily from Windows PowerShell. Features over 600 scripts to help you build quick user interfaces. Think HTML Applications (HTAs), but easy.
- **FileSystem** Monitors files and folders, checks for duplicate files, and checks disk space.
- **IsePack** Supercharge your scripting in the Integrated Scripting Environment (ISE) with more than thirty-five shortcuts.

- **DotNet** Explores loaded types, finds commands that can work with a type, and describes how you can use Windows PowerShell, DotNet, and COM together.
- **PSImageTools** Converts, rotates, scales, and crops images and gets image metadata.
- **PSRSS** Harnesses the FeedStore from Windows PowerShell.
- **PSSystemTools** Gets operating system or hardware information.
- **PSUserTools** Gets the users on a system, checks for elevation, and starts a process as administrator.
- **PSCodeGen** Generates Windows PowerShell scripts, C# code, and Pinvoke.
- **TaskScheduler** Lists scheduled tasks and creates and deletes tasks.

For information on how to install the PowerShell Pack on Windows 7, see the ReadmePP.txt file in the \PowerShellPack folder on the companion media.

Note that the modules and accompanying documentation included in the PowerShell Pack are presented as is, with no warranty, and are entirely unsupported by Microsoft. Do not use these modules in your production environment without testing them first in a nonproduction environment. See the section titled "Disclaimer Concerning Windows PowerShell CD Content" later in this introduction for more information.

Using the Sample Windows PowerShell Scripts

Included on the companion media are almost two hundred sample scripts that demonstrate how you can administer different aspects of Windows 7 using Windows PowerShell. These sample scripts are presented as is, with no warranty, and are entirely unsupported by Microsoft. Do not use these scripts in your production environment without testing them first in a nonproduction environment. You may need to customize some scripts to make them work properly in a production environment. See the section titled "Disclaimer Concerning Windows PowerShell CD Content" later in this introduction for more information.

Before you use these scripts, you must understand how Windows PowerShell execution policy controls how scripts are run on a computer. Windows PowerShell can have five possible values for the script execution policy on a computer:

- **Restricted** This is the default setting and allows no scripts to run.
- **AllSigned** This setting means that scripts need a digital signature before they can be run.

- **RemoteSigned** This setting means that only scripts run from file shares, downloaded using Internet Explorer, or received as e-mail attachments must be signed.
- **Unrestricted** This setting means that all scripts can be run.
- **Bypass** This setting means that nothing is blocked and there are no prompts or warnings.

To view the current script execution policy, open a Windows PowerShell command prompt and type **Get-ExecutionPolicy**. The current execution policy for your system can be changed by typing **Set-ExecutionPolicy <*value*>,** where <*value*> is one of the five values listed previously. Changing the execution policy requires that Windows PowerShell be run as an administrator. Note, however, that if your script execution policy is set by your network administrator using Group Policy, you will not be permitted to change the execution policy on your computer.

Microsoft recommends that the execution policy be configured as *RemoteSigned* within a production environment, unless you have a compelling reason for either a stricter or a less strict setting. For information on how to sign PowerShell scripts, see *http://technet.microsoft.com/en-us/magazine/2008.04.powershell.aspx*. You can also type **Get-Help about_signing** at the Windows PowerShell command prompt for further information about signing scripts.

Remoting, a new feature of Windows PowerShell 2.0, uses the WS-Management protocol to allow you to run Windows PowerShell commands on one or many remote computers. This means that many of the scripts included on the companion media will work on remote computers even though they may not have the *–computer* parameter that allows you to specify a remote computer name. For Windows PowerShell remoting to work, you must have Windows PowerShell 2.0 installed and configured on both the local computer and the targeted remote computer. You must also enable remoting on the targeted remote computers by running the **Enable-PSRemoting** command on them, which configures these computers to receive remote commands. The **Enable-PSRemoting** command must be run with administrative rights. For more information about Windows PowerShell remoting technology, type **Get-Help about_remoting** at the Windows PowerShell command prompt.

Some of these sample scripts use Windows Management Instrumentation (WMI), Active Directory Services Interface (ADSI), or the Microsoft .NET Framework application programming interfaces (APIs) to connect to remote computers. These scripts may work on remote computers even if Windows PowerShell is not installed on those computers. Before some of these scripts can work remotely, however,

you may need to enable remote administration through Windows Firewall on both the host computer and the target computer on the appropriate network connection. You will also need to be a member of the local administrators group on the remote computer.

You can use the *EnableDisableRemoteAdmin.ps1* script to enable remote administration through Windows Firewall. Note that the actions performed by this script are not appropriate for edge-connected machines and may not be appropriate for some enterprise customers. Before you run **EnableDisableRemoteAdmin.ps1** in a production environment, you should evaluate the changes being made by this script to determine if they are appropriate for your environment. For more information about how WMI works through the Windows Firewall, see *http://msdn.microsoft.com/en-us/library/aa389286.aspx*.

All the sample scripts include command-line help. To obtain basic information about a script, type **Get-Help *script_name.ps1,*** where *script_name.ps1* is the name of the script. To see sample syntax for using the script, as well as detailed help information, type **Get-Help *script_name.ps1* –Full**. If you only want to see examples of how to use the script, type **Get-Help *script_name.ps1* –Examples.**

Disclaimer Concerning Windows PowerShell CD Content

The Windows PowerShell scripts included on the companion media are only samples and are not finished tools. These scripts are provided as proof-of-concept examples of how to administer Windows 7 clients using Windows PowerShell. Although every effort has been made to ensure that these sample scripts work properly, Microsoft disclaims any responsibility for any and all liability or responsibility for any damages that may result from using these scripts. The sample scripts are provided to you as is, with no warranty or guarantee concerning their functionality, and Microsoft does not provide any support for them.

The Windows 7 Resource Kit PowerShell Pack included on the companion media is also unsupported by Microsoft and is provided to you as is, with no warranty or guarantee concerning its functionality. For the latest news and usage tips concerning this PowerShell Pack, see the Windows PowerShell Team Blog at *http://blogs.msdn.com/powershell/*.

Be sure to thoroughly familiarize yourself with using these Windows PowerShell scripts and modules in a test environment before attempting to use them in your production environment. Because these sample scripts are provided as proof-of-

concept samples only, you may need to customize them if you intend to use them in your production environment. For example, the scripts as provided include only minimal error handling and assume that the clients they are being run against exist and are configured appropriately. The authors therefore encourage readers to customize these scripts to meet their particular needs.

Resource Kit Support Policy

Every effort has been made to ensure the accuracy of this book and the companion media content. Microsoft Press provides corrections to this book through the Internet at the following location:

http://www.microsoft.com/mspress/support/search.aspx

If you have comments, questions, or ideas regarding the book or companion media content, or if you have questions that are not answered by querying the Knowledge Base, please send them to Microsoft Press by using either of the following methods:

E-mail:

rkinput@microsoft.com

Postal Mail:

Microsoft Press
Attn: Windows 7 Resource Kit Editor
One Microsoft Way
Redmond, WA 98052-6399

Please note that product support is not offered through the preceding mail addresses. For product support information, please visit the Microsoft Product Support Web site at the following address:

http://support.microsoft.com

CHAPTER 22

Supporting Users with Remote Assistance

Remote Assistance (RA) in Windows Vista included improvements in connectivity, performance, usability, and security along with feature enhancements that make it even more useful than Remote Assistance in Windows XP was. The Windows 7 operating system builds on these earlier improvements with Easy Connect, a new feature of Remote Assistance that makes it easier than ever for novice users to request help from expert users and for experts to offer help to novices. With increased Group Policy support, command-line scripting capabilities, session logging, bandwidth optimization, and more, Remote Assistance is now an essential tool for enabling enterprises to support users in Help Desk scenarios. This chapter examines how Remote Assistance works in Windows 7, how to use it to support end users, and how to manage it using Group Policy and scripts.

Understanding Remote Assistance

Supporting end users is an essential function of IT departments and the corporate Help Desk. Unfortunately, conventional technical support provided over the telephone or using chat tools is generally cumbersome and inefficient. As a result, supporting users is often both time-consuming and costly for large enterprises to implement. For example, end users often have difficulty describing the exact nature of the problem they are having. Because of their general inexperience and lack of technical knowledge, end users may try to describe their problem using nontechnical, inexact language. As a result, Help Desk personnel are generally reduced to asking a series of simple questions to try to isolate the problem the user is having. The methodical nature of these questions sometimes causes users to feel as if Help Desk personnel are being condescending, and such

misunderstandings can reduce the effectiveness of the support experience and can make users tend to avoid contacting support personnel when future problems arise.

End users also often have difficulty following instructions given to them by Help Desk personnel who are trying to assist them. Well-trained support personnel will try to avoid using technical jargon when communicating with end users, but although using plain language can improve the support experience, it may also mean that resolution steps become long and tiresome. For example, telling a user how to use Disk Cleanup from System Tools in Accessories can require several sentences or more, and this kind of communication can add time to support incidents, making them more costly to the company.

Remote Assistance solves these problems by enabling support personnel to view the user's desktop in real time. The user seeking assistance can demonstrate the nature of the problem to the support person. This is a quicker and more efficient way to communicate a problem than using words or e-mail. If necessary, the user can also give the support person permission to assume shared interactive control of the user's computer to show the user how to resolve the problem. The result of using Remote Assistance is faster problem resolution, an improved support experience, and a lower Total Cost of Ownership (TCO) for supporting end users in large, corporate environments.

Remote Assistance vs. Remote Desktop

Remote Assistance and Remote Desktop are different features of Windows 7 that have entirely different uses. Remote Desktop is based on Microsoft Terminal Services and is a tool for logging on to remote computers. When you use Remote Desktop to connect to a remote computer, a new user session is established. Remote Desktop can also establish sessions with computers that have no interactive sessions running (no users logged on locally), such as headless servers. For more information on Remote Desktop, see Chapter 27, "Connecting Remote Users and Networks."

Remote Assistance, on the other hand, is a tool for interactively helping users troubleshoot problems with their computers. To use Remote Assistance, both the User (also called the Novice) and the Helper must be present on their computers. Unlike Remote Desktop, Remote Assistance does not create a new session. Instead, Remote Assistance allows the Helper to work in the existing session of the User. The User's desktop gets remoted to the Helper, who can then view the User's desktop and, with the User's consent, share control of the desktop.

Here is another way to summarize the difference between these two features: In Remote Assistance, both users involved are looking at the same desktop using the same logon credentials (those of the interactively logged-on User) and can share control of that desktop; in Remote Desktop, when the remote person logs on, the interactively logged-on user (if one exists) is logged out.

Improvements to Remote Assistance in Windows 7

As mentioned previously, Remote Assistance in Windows 7 builds on the many enhancements introduced earlier for this feature in Windows Vista. These earlier enhancements improved upon the earlier Windows XP implementation of Remote Assistance and included the following:

- Connectivity improvements with transparent Network Address Translation (NAT) traversal using Teredo and IPv6
- An improved user interface (UI) that is easier to start and use
- A stand-alone executable (Msra.exe) that accepts command-line arguments and can easily be scripted
- Improved overall performance with a smaller footprint, quicker startup and connect times, and optimized bandwidth usage for screen updates
- Enhanced security with mandatory password and integration with User Account Control (UAC)
- New Offer RA via IM scenario and an open application programming interface (API) for integration with peer-to-peer (P2P) applications
- Additional Group Policy settings for improved manageability

In addition to these Windows Vista enhancements for Remote Assistance, Windows 7 adds the following new enhancements to Remote Assistance:

- Easy Connect, a method for soliciting Remote Assistance that uses the P2P collaboration infrastructure to simplify Remote Assistance user interactions
- An improved Windows Remote Assistance Wizard that makes it easier than ever for users to solicit or offer help
- New command-line arguments for the Remote Assistance executable (Msra.exe)

Remote Assistance in Windows 7 and Windows Vista deprecates the following features that were available on Windows XP:

- No more support for the MAILTO method of solicited Remote Assistance
- No more support for voice sessions

In addition, Remote Assistance in Windows 7 has deprecated the file transfer feature that was available in Windows XP and Windows Vista. Compatibility with earlier versions is still supported, however—for example, if a file transfer is initiated from a Windows XP or Windows Vista computer, Windows 7 will accept the transfer.

For information on interoperability between the Windows XP, Windows Vista, and Windows 7 versions of Remote Assistance, see the section titled "Interoperability with Remote Assistance in Windows XP" later in this chapter.

How Remote Assistance Works

In Remote Assistance, the person needing help is referred to as the *User* (or *Novice*), and the support person providing assistance is called the *Helper* (or *Expert*). You start Remote Assistance from the Start menu by navigating to All Programs, selecting Maintenance, and then selecting Windows Remote Assistance. You can also start Remote Assistance from a command prompt by typing **msra.exe**.

Remote Assistance has two basic modes of operation:

- **Solicited RA** In *Solicited RA* (also known as *Escalated RA*), the User requests assistance from the Helper by initiating the Remote Assistance session using e-mail, instant messaging (IM), Easy Connect, or by providing the Helper with a saved copy of an invitation file (*.MsRcIncident). Each of these methods uses a different underlying mechanism:
  - **Solicited RA using e-mail** This method requires that the e-mail clients being used by the User support Simple Mail Application Programming Interface (SMAPI). Examples of SMAPI-compliant e-mail clients include Windows Mail, which is included in Windows Vista, and Microsoft Office Outlook 2007. Windows 7 does not have a built-in e-mail SMAPI-compliant client, but you can install Windows Live Mail, which is available for download as part of the Windows Live Essentials suite of applications (at *http://get.live.com*). Web-based e-mail services, such as Windows Live Hotmail, are not SMAPI-compliant and cannot be used for soliciting or offering Remote Assistance using e-mail. In this approach, the User starts the Remote Assistance UI to create an e-mail message that has a Remote Assistance invitation file (*.MsRcIncident) attached to the message. The User must specify a password for the Remote Assistance session, which must be communicated to the Helper using an out-of-band (OOB) method such as calling the Helper on the telephone. When the Helper receives the User's Remote Assistance invitation, she opens the attached ticket, enters the password that was conveyed by the User, and the Remote Assistance session starts. The Helper must respond to the invitation from the User within a specified time limit (the default is 6 hours), or the invitation will expire and a new one will need to be sent. In a domain environment, this ticket lifetime can also be configured using Group Policy. See the section titled "Managing Remote Assistance Using Group Policy" later in this chapter.
  - **Solicited RA using file transfer** This method requires that both the User and Helper have access to a common folder (such as a network share on a file server), or that they use some other method for transferring the file (for example, by using a USB key to manually transfer the file or by uploading the file to an FTP site). The user creates a Remote Assistance invitation file and saves it in the shared folder. The User must provide a password that must be communicated to the Helper using an OOB method such as a telephone call. The Helper retrieves the ticket from the shared folder, opens it, enters the password, and the Remote Assistance session starts. Again, the Helper must respond to the invitation within a specified time, or the invitation will expire and a new one will be needed. (The expiration time is configurable through Group Policy.)

- **Solicited RA using instant messaging** This method for soliciting assistance requires that the IM applications being used by both the User and the Helper support the Microsoft Rendezvous API. An example of an IM application that supports the Rendezvous API is Windows Live Messenger, which is available for download as part of the Windows Live Essentials suite of applications (at *http://get.live.com*). In this approach, the User requests assistance from someone on his buddy list. To ensure that the remote person is really the User's buddy (and not someone masquerading as the buddy), Remote Assistance requires that a password be relayed from the User to the Helper by other means (such as a phone call) before the Helper can connect. For more information on the Rendezvous API, see the Windows Software Development Kit (SDK) on MSDN at *http://msdn.microsoft.com/en-us/library/aa359213.aspx*.
- **Solicited RA using Easy Connect** This method for soliciting assistance is new in Windows 7 and uses Peer Name Resolution Protocol (PNRP) to enable direct P2P transfer of the Remote Assistance invitation using the cloud. To establish the initial Remote Assistance session, the User only needs to communicate a password to the Helper using an OOB method such as by telephone. The Helper uses this password to obtain the Remote Assistance invitation from the cloud and initiate the session. When the initial Remote Assistance connection has been made, a trust relationship is established between the Helper and the User. This trust relationship is established through the exchange of contact and certificate information. Subsequent interactions are simplified because the contact information can be used to pick a Helper who is currently available. For more information on this method for soliciting assistance, see the section titled "Scenario 1: Soliciting Remote Assistance Using Easy Connect" later in this chapter. For information on how Easy Connect works, see the sidebar titled "Direct from the Source: How Easy Connect Works" later in this chapter. For information on how PNRP works, see the sidebar titled "How It Works: PNRP and Microsoft P2P Collaboration Services" later in this chapter.

- **Unsolicited RA** In Unsolicited RA (also known as Offer RA), the Helper offers help to the User by initiating the Remote Assistance session using Distributed Component Object Model (DCOM). Unsolicited RA is a typical corporate Help Desk scenario in which all the users are in a domain. The Helper enters either the fully qualified domain name (FQDN) or IP address of the User's computer to connect to the User's computer. This method requires that the Helper has been previously authorized as a domain administrator to be able to offer Remote Assistance to the Users. (For information on how to authorize Helpers for offering Remote Assistance, see the section titled "Managing Remote Assistance Using Group Policy" later in this chapter.) This method also requires that the Helper either knows the name (the host name on a local subnet; the fully qualified name otherwise) or address (IPv4 or IPv6) of the User's computer.

HOW IT WORKS

PNRP and Microsoft P2P Collaboration Services

Microsoft P2P network and collaboration technologies are designed to enable the next generation of peer-to-peer scenarios, including shared workspaces, distributed computing, and even load balancing. These P2P technologies allow users to securely communicate and share information with each other without requiring a central server to be involved. Because P2P technologies are designed to work in networking environments with transient connectivity—such as an ad hoc wireless network established between several laptops at a coffee shop—they cannot rely on the server-based Domain Name System (DNS) to perform name resolution between peers. Instead, P2P name resolution is based on the PNRP, a mechanism for distributed, serverless name resolution of peers in a P2P network.

PNRP works by utilizing multiple groupings of computers called clouds. These clouds correspond to two different scopes of IPv6 addresses:

- **Global cloud** Any given computer will be connected to a single Global cloud. For computers with IPv6 Internet connectivity, the Global cloud is Internet-wide. In networks where computers do not have IPv6 Internet connectivity but still have Global IPv6 addresses (such as firewalled corporate environments), the Global cloud is network-wide.
- **Link-local clouds** One or more clouds, each corresponding to nodes within the same subnet or network link (link-local addresses and the link-local address scope). Note that Remote Assistance only uses the Global (Internet-wide) cloud; link-local clouds are not used by Remote Assistance.

Peer names in PNRP are static identifiers of endpoints that can be resolved to changing IP addresses, enabling P2P communications. Peer names can be computers, users, devices, groups, services, or anything that can be identified by an IPv6 address and port. Peer names are represented by identifiers (IDs) that are 32 bytes long and can be either unsecured (names that can be spoofed) or secured (names that cannot be spoofed because they are derived from a public/private key pair owned by the publisher).

The underlying name resolution functions on PNRP IDs within a cloud are stored in a distributed fashion in a cache on each peer within the cloud, with each peer's cache containing only a portion of the names for all the peers in the cloud. When the issuing peer wants to resolve the name of the targeted peer to its published address and port number, it follows these steps:

1. The issuing peer first consults its own PNRP cache for this information. If it finds this information, it sends a PNRP Request message to the targeted peer and waits for a response. These Request messages serve the function of enabling peers to communicate to other peers their active involvement within the cloud.

2. If the issuing peer does not find this information, it sends the Request message to the peer whose ID most closely matches (that is, is closest numerically to) that of the targeted peer. The peer that receives this message then consults its own cache. If it finds a closer match or the match itself, it returns this information to the requesting peer. The requesting peer then goes to the returned peer and the process continues until the resolution succeeds or fails.
3. If the peer that receives this message does not find closer information in its cache, it returns the message to the issuing peer, indicating that it does not know the targeted peer. The issuing peer then repeats the previous step by sending a message to the peer whose ID next most closely matches that of the targeted peer. This process continues until the targeted peer is found (if present on the network) or not found (if no longer present within the cloud).

Looping is prevented by including in the Request message the list of peers that have already forwarded requests.

For more information on how PRNP and other Microsoft P2P technologies work, see *http://technet.microsoft.com/en-us/library/bb742623.aspx* on TechNet.

Remote Assistance Operational States

Remote Assistance has three operational states:

- **Waiting For Connect** This state occurs when either:
  - The Helper has offered Remote Assistance to the User, but the User has not yet agreed to allow the Helper to connect to his computer.
  - The User has sent the Helper an invitation but the Helper has not yet responded by opening the invitation, or the Helper has opened the invitation and the User has not yet agreed to allow the Helper to connect to his computer.

  In the Waiting For Connect state, the Helper cannot view or control the screen of the User's computer until a Remote Assistance connection has been established and both computers have entered the Screen Sharing state. After the Remote Assistance application has been started and is running in the Waiting For Connect state, the application should not be closed until the other party responds and establishes the connection. For example, if the User uses the Solicit RA Using E-mail method and sends an invitation file to a Helper, the Remote Assistance application opens on the User's computer and waits for the Helper to accept the invitation. If the User closes Remote Assistance on her computer before the Helper accepts the invitation, the Helper will not be able to connect to the User's computer and the User will need to send a new invitation.
- **Screen Sharing** This state occurs when the User has consented to allow the Helper to connect to his computer—either after the User has sent the Helper an invitation

or the Helper has offered Remote Assistance to the User. In the Screen Sharing state, a Remote Assistance session has been established and the Helper can view—but not control—the screen of the User's computer.

When the User is prompted for consent to allow the Helper to connect to his computer, a warning message appears on the User's computer saying that the Helper wants to connect to his computer. This warning message is customizable using Group Policy. See the section titled "Managing Remote Assistance Using Group Policy" later in this chapter for more information.

- **Control Sharing** This state occurs after the Screen Sharing state when the Helper has requested control of the User's computer and the User has consented to allow the Helper to have shared control of his computer. In the Control Sharing state, the Helper has the same level of access to the User's computer that the User has, and the Helper can use his own mouse and keyboard to remotely perform actions on the User's computer. Specifically:
  - If the User is a standard user on his computer, the Helper will be able to perform only those actions on the User's computer that can be performed by a standard user on that computer.
  - If the User is a local administrator on his computer, the Helper will be able to perform any actions on the User's computer that can be performed by a local administrator on that computer.

  For more information on the level of control that a Helper has on a User's computer, see the section titled "Remote Assistance and the Secure Desktop" later in this chapter.

User vs. Helper Functionality

After a Remote Assistance connection has been established and both computers have entered the Screen Sharing state, the User and Helper are able to perform the tasks listed in Table 22-1.

TABLE 22-1 Tasks That Can Be Performed by User and Helper During a Remote Assistance Session

DESCRIPTION OF TASK	USER?	HELPER?
Chat	Yes	Yes
Save a log of session activity	Yes (default)	Yes (default)
Configure bandwidth usage	Yes	No
Pause (temporarily hide screen)	Yes	No
Request shared control	No	Yes
Give up shared control	Yes	Yes
Disconnect	Yes	Yes

Remote Assistance and NAT Traversal

Remote Assistance works by establishing a P2P connection between the User's computer and the Helper's computer. One challenge this poses is that it can be difficult to establish P2P connections if one or both of the computers involved are behind a gateway or router that uses NAT. NAT is an IP routing technology described by RFC 1631 that is used to translate IP addresses and TCP/UDP port numbers of packets being forwarded. NAT is typically used to map a set of private IP addresses to a single public IP address (or to multiple public addresses). Home networks using a wireless or wired router also use NAT technology.

To overcome this difficulty, Windows 7 and Windows Vista include built-in support for Teredo, an IPv6 transition technology described in RFC 4380 that provides address assignment and automatic tunneling for unicast IPv6 connectivity across the IPv4 Internet. The NAT traversal capability provided by Teredo in Windows 7 and Windows Vista allows Remote Assistance connectivity when one or both of the users involved in a Remote Assistance session are hidden behind a NAT. The Remote Assistance experience is transparent from the perspective of the users involved, regardless of whether or not NAT is being used on either user's network. For most small business and home user environments, Remote Assistance in Windows 7 and Windows Vista will seamlessly traverse a NAT-enabled router with no additional router configuration required. For information on enterprises that need to remotely support users who work from home, see the section titled "Other Possible Remote Assistance Usage Scenarios" later in this chapter.

NOTE **Offering Remote Assistance using DCOM is not usually a Teredo scenario because enterprise users are behind a corporate firewall and are not separated from each other by NATs.**

Remote Assistance can connect across restricted NATs and cone NATs, which generally comprise the large majority of deployed NATs. Beginning with Windows 7, Remote Assistance can also connect across certain types of symmetric NATs, but only if the other computer is not behind a symmetric NAT as well. For more information on NAT traversal support in Windows 7, see Chapter 28, "Deploying IPv6."

Remote Assistance will not connect in certain configurations. Specifically:

- Remote Assistance will not work if the NAT-enabled router is configured to block the specific ports used by Remote Assistance. See the section titled "Remote Assistance and Windows Firewall" later in this chapter for more information.
- Remote Assistance will not work if the User's NAT-enabled router is configured to block all UDP traffic.

NOTE **To determine the type of NAT a network is using, open an elevated command prompt and type netsh interface teredo show state.**

For more information on IPv6 support in Windows 7, including built-in client support for Teredo and other IPv6 transition technologies, see Chapter 28.

To verify whether your NAT supports Remote Assistance, you can use the Internet Connectivity Evaluation Tool at *http://www.microsoft.com/windows/using/tools/igd/default.mspx*. If your NAT supports Universal Plug and Play (UPnP), then Remote Assistance should be able to get a global IPv4 address that allows anyone to connect to you. If your NAT supports Teredo/IPv6 and you are running Windows 7 or Windows Vista, then an RA Helper that is running Windows 7 or Windows Vista and is Teredo-enabled should be able to connect to you.

Remote Assistance and IP Ports Used

The ports used by a Remote Assistance session depend on which version of Windows is running on the two computers involved in the session. Specifically:

- **Windows 7 to Windows 7, Windows 7 to Windows Vista, or Windows Vista to Windows Vista** Dynamic ports allocated by the system in the range TCP/UDP 49152–65535
- **Windows 7 to Windows XP or Windows Vista to Windows XP** Port 3389 TCP (local/remote)

In addition, the Offer RA via DCOM scenario uses port 135 (TCP).

NOTE **If you are concerned about opening the DCOM port (TCP port 135) on your corporate firewall and want to avoid doing this but still want to be able to offer Remote Assistance to remote users, you can do so by using Authenticated IPsec Bypass as described in *http://technet.microsoft.com/en-us/library/cc753463.aspx*.**

Remote Assistance and Windows Firewall

The Windows Firewall is configured with a group exception for Remote Assistance. This group exception has multiple properties that are grouped together as part of the Remote Assistance exception. The Remote Assistance exception properties will change depending on the network location of the computer (private, public, or domain). For example, the default Remote Assistance exception when the computer is in a public location is stricter than when the computer is in a private location. In a public location (such as an airport), the Remote Assistance exception is disabled by default and does not open ports for UPnP and Simple Service Discovery Protocol (SSDP) traffic. In a private network (a home or work network, for example) the Remote Assistance exception is enabled by default and UPnP and SSDP traffic is permitted. In a domain-based enterprise environment, the Remote Assistance exception is typically managed using Group Policy and is enabled by default in Windows 7; it was disabled by default in Windows Vista.

The default configuration of the Remote Assistance exception in Windows Firewall varies depending on the firewall profile. Specifically, note the following:

- **Private profile** The Remote Assistance exception in the Windows Firewall is enabled by default when the computer location is set to Private. It is configured for NAT traversal using Teredo by default so that users in a private networking environment (for

example, the home environment) can solicit help from other users who may also be behind NATs. The private profile includes the appropriate exceptions needed to allow communication with UPnP NAT devices. If a UPnP NAT is in this environment, Remote Assistance will attempt to use the UPnP for NAT traversal. This profile also includes exceptions needed for PNRP. Offer RA via DCOM is not configured in this profile.

- **Public profile** The Remote Assistance exception is disabled by default and no inbound Remote Assistance traffic is permitted. Windows Firewall is configured this way by default to better protect users in a public networking environment (such as a coffee shop or airport terminal). When the Remote Assistance exception is enabled, NAT traversal using Teredo is enabled. However, traffic to UPnP devices is not enabled, and Offer RA via DCOM is not enabled.
- **Domain profile** The Remote Assistance exception when the computer is in a domain environment is geared toward the Offer RA scenario. This exception is enabled by default in Windows 7 and is typically managed via Group Policy.

Table 22-2 summarizes the state of the Remote Assistance firewall inbound exception for each type of network location. The Remote Assistance exception has outbound properties as well; however, outbound exceptions are not enabled in Windows Firewall by default.

TABLE 22-2 Default State of Remote Assistance Firewall Inbound Exception for Each Type of Network Location

NETWORK LOCATION	STATE OF REMOTE ASSISTANCE EXCEPTION	DEFAULT PROPERTIES OF THE REMOTE ASSISTANCE EXCEPTION
Private (Home or Work)	Enabled by default	■ Msra.exe application exception ■ UPnP enabled for communications with UPnP NATs ■ PNRP enabled ■ Edge traversal enabled to support Teredo
Public	Disabled by default; must be enabled by user with Admin credentials	■ Msra.exe application exception ■ Edge traversal enabled to support Teredo
Domain	Enabled by default in Windows 7; disabled by default in Windows Vista	■ Msra.exe application exception ■ RAServer.exe (the RA COM server) application exception ■ PNRP enabled ■ DCOM port 135 ■ UPnP enabled for communications with UPnP NATs

Remote Assistance and the Secure Desktop

When a User consents to having a Helper share control of her computer during a Remote Assistance session, the User has the option of allowing the Helper to respond to UAC prompts (Figure 22-1). Typically, UAC prompts appear on the Secure Desktop (which is not remoted), and consequently the Helper cannot see or respond to Secure Desktop prompts. The Secure Desktop mode is the same mode that a user sees when she logs on to her computer or presses the Secure Attention Sequence (SAS) keystroke (Ctrl+Alt+Delete). UAC elevation prompts are displayed on the Secure Desktop instead of the user's normal desktop to protect the user from unknowingly allowing malware to run with elevated privileges on her computer. The User must provide consent to a UAC prompt to return to her normal desktop and continue working. This consent requires either clicking Continue (if the user is a local administrator on her computer) or by entering local administrative credentials (if she is a standard user on her computer).

FIGURE 22-1 The User has the option of allowing the Helper to respond to UAC prompts when the Remote Assistance session is in the Control Sharing state.

It is important to understand that the Secure Desktop on the User's computer is not remoted to the Helper's computer. In other words, the Helper can respond only to UAC prompts on the User's computer using the User's own credentials. This means that if the User is a standard user on her computer and the Helper is a local administrator on the User's computer, the Helper can have only administrative privileges on the User's computer if the User can first supply those credentials.

Enforcing this limitation is essential to ensure the security of Windows 7 desktops. The reason behind this design decision is that, if Remote Assistance was architected to allow the Helper to remotely elevate the User's privileges, the User would be able to terminate the Remote Assistance session and thus steal local administrative credentials from the Helper.

Remote Assistance Logging

Remote Assistance can generate a session log of Remote Assistance-associated activity. Session logging is enabled by default and consists of timestamped records that identify Remote Assistance-related activities on each computer. Session logs only contain information about activities that specifically relate to Remote Assistance functionality, such as who initiated the session, if consent was given to a request for shared control, and so on.

Session logs do not contain information on actual tasks that the User or Helper performed during a session. For example, if the Helper is given Shared Control privileges, starts an Admin command prompt, and performs steps to reconfigure the TCP/IP configuration on the User's computer during a Remote Assistance session, the session logs will not contain a record of this action.

Session logs do include any chat activity performed during a Remote Assistance session. The log generated during a session is also displayed within the chat window so that both the User and the Helper can see what is being logged during the session. Session logs also include any file transfer activity that occurs during the session, and they also record when the session has been paused.

PURPOSE OF REMOTE ASSISTANCE SESSION LOGGING

Session logs for Remote Assistance are mainly intended for enterprises that are required to maintain records of system and user activity for record-keeping purposes. They are not intended as a way to record every action performed by Help Desk personnel when troubleshooting problems with users' computers. A typical environment in which session logging might be required would be in a banking environment, where a financial institution is required by law to maintain records of who accessed a computer and at what time.

Because the permissions on these session logs grant the User full control over logs stored on her own computer, by default, session logs are generated on both the User's computer and the Helper's computer so that the Helper can archive them and protect them from tampering. The logs created on each side of a Remote Assistance session are similar but not identical. This is because session logs are generated from the perspective of the computer involved—whether the User's computer or the Helper's computer—and therefore complement each other instead of being identical.

In an enterprise environment, Group Policy can be used to enable or disable session logging. If session logging is not configured using Group Policy, both the User and Helper are free to disable session logging on their own computers. For more information, see the section titled "Managing Remote Assistance Using Group Policy" later in this chapter.

SESSION LOG PATH AND NAMING CONVENTION

Session logs are XML-formatted documents so that they can be easily integrated into other data sets—for example, by importing them into a database managed by Microsoft SQL Server 2005. All session logs are stored in the following subfolder of the user's profile:

%UserProfile%\Documents\Remote Assistance Logs

A unique session log file is created for each Remote Assistance session on the computer. Log files stored within this folder are formatted using XML and are named using the convention *YYYYMMDDHHMMSS*.xml, where the time format is 24-hour. For example, a session log created at 3:45:20 P.M. on August 13, 2008, would be named 20080813154520.xml.

The XML content of a typical session log looks like the following:

```
<?xml version="1.0" ?>
<SESSION>
  <INVITATION_OPENED TIME="3:24 PM" DATE="Wednesday, May 07, 2008" EVENT="A Remote
Assistance invitation has been opened." />
  <INCOMING_IP_ADDRESS TIME="3:26 PM" DATE="Wednesday, May 07, 2008">fe80::2856:e5b0:
fc18:143b%10</INCOMING_IP_ADDRESS>
  <CONNECTION_ESTABLISHED TIME="3:26 PM" DATE="Wednesday, May 07, 2008" EVENT="A Remote
Assistance connection has been established.">jdow</CONNECTION_ESTABLISHED>
  <EXPERT_REQUEST_CONTROL TIME="3:27 PM" DATE="Wednesday, May 07, 2008" EVENT="jdow has
requested to share control of the computer." />
  <EXPERT_GRANTED_CONTROL TIME="3:27 PM" DATE="Wednesday, May 07, 2008" EVENT="jdow has
been granted permission to share control of the computer." />
  <EXPERT_CONTROL_STARTED TIME="3:27 PM" DATE="Wednesday, May 07, 2008" EVENT="jdow is
sharing control of the computer." />
  <EXPERT_CONTROL_ENDED TIME="3:27 PM" DATE="Wednesday, May 07, 2008" EVENT="jdow is not
sharing control of the computer." />
  <CHAT_MESSAGE TIME="3:30 PM" DATE="Wednesday, May 07, 2008">jdow: test</CHAT_MESSAGE>
  <CHAT_MESSAGE TIME="3:30 PM" DATE="Wednesday, May 07, 2008">jchen: ok</CHAT_MESSAGE>
  <CONNECTION_ENDED TIME="3:30 PM" DATE="Wednesday, May 07, 2008" EVENT="The Remote
Assistance connection has ended." />
  <INVITATION_CLOSED TIME="3:30 PM" DATE="Wednesday, May 07, 2008" EVENT="A Remote
Assistance invitation has been closed." />
</SESSION>
```

Using Remote Assistance in the Enterprise

The main Remote Assistance scenario within a corporate networking environment is supporting desktop computers that are on the corporate network and joined to a domain. Users' computers must be configured appropriately before they can be offered Remote Assistance. This is done via Group Policy, as explained in the section titled "Managing Remote Assistance Using Group Policy" later in this chapter. Additionally, the Remote Assistance exception in the Windows Firewall must be enabled. For more information, see the section titled "Remote Assistance and Windows Firewall" earlier in this chapter.

Because most corporate networks have a perimeter firewall blocking access from outside the internal network, supporting remote users who are connecting from outside the corporate network can be more difficult. However, most enterprises now use virtual private network (VPN) technologies to allow remote users to connect to their corporate networks over the Internet, and this kind of scenario generally poses no problem to Remote Assistance functionality.

Using Remote Assistance in the Corporate Help Desk Environment

The standard approach to using Remote Assistance in an enterprise environment is for Help Desk personnel to offer Remote Assistance to users who telephone to request assistance. A typical scenario might be as follows:

1. User Jane Dow (the User) is having problems configuring an application on her computer. She phones Help Desk, explains her problem briefly, and asks for help.
2. A Help Desk person named Jacky Chen (the Helper) asks Jane for the FQDN or IP address of her computer. She responds with the information, which she can get from computer properties or by running *ipconfig*.
3. Jacky starts Remote Assistance on his computer and uses the Offer RA feature to offer help to Jane. This causes a dialog box to appear on Jane's computer, asking her if she would like to allow Jacky to connect to her computer.
4. Jane accepts the offer, and at this point Jane's desktop may temporarily change to conserve network bandwidth used by the Remote Assistance session. The Remote Assistance window that opens on Jane's screen tells her that she is being helped by Jacky.
5. At this point, Jacky can see Jane's screen, but he can't control it. Jane then explains the problem she is having, either by using the Chat feature of Remote Assistance, or more likely over the telephone. Jacky asks Jane to perform a series of steps to correct the problem and watches her screen in his own Remote Assistance window as she does this.
6. If the instructions Jacky provides are too complex or if time is limited, Jacky can ask Jane if he can share control of her computer. If Jane agrees, Jacky clicks the Request Control button at the top of his Remote Assistance window. A dialog box appears on Jane's desktop asking her if she wants to allow Jacky to share control of her desktop. Jane accepts the prompt and also selects the option to allow Jacky to respond to UAC prompts on Jane's computer.
7. Jacky is now connected to Jane's computer using Jane's credentials, and he can both view her screen and interact with it using his own mouse and keyboard. Jacky then proceeds to perform the steps needed to resolve the problem, either correcting the issue or demonstrating to Jane how to fix the problem if it occurs again in the future. If at any time Jane wants to force Jacky to relinquish control of her computer, she can click the Stop Sharing button or the Disconnect button, or she can press the Panic key (Esc).

NOTE **Offer RA needs preconfiguration of the User's computer via Group Policy. See the section titled "Managing Remote Assistance Using Group Policy" later in this chapter for more information.**

Other Possible Remote Assistance Usage Scenarios

Other types of Remote Assistance scenarios are also possible for businesses ranging from large enterprises to Small Office/Home Office (SOHO) environments. Examples of possible usage scenarios include:

- A user who is having a problem configuring an application on her computer can phone the Help Desk for assistance. A support person can then use Offer RA to connect to the user's computer, ask for control of her screen, and show the user how to configure her application. This scenario is the standard one for enterprise Help Desk environments and is described in more detail in the section titled "Using Remote Assistance in the Corporate Help Desk Environment" earlier in this chapter.
- A user who is having trouble installing a printer sends a Remote Assistance invitation to Help Desk using Windows Mail. A support person who is monitoring the Help Desk e-mail alias reads the message, opens the attached invitation file, and connects to the user's computer. The support person asks for control of the user's computer and walks him through the steps of installing the printer.
- A user is on the road and is connected to the internal corporate network using a VPN connection over the Internet. The user is having problems configuring Windows Mail on her computer, so she opens Windows Live Messenger and notices that someone she knows in Corporate Support is currently online. She sends a Remote Assistance invitation to the support person using Windows Live Messenger, and that person responds to the invitation, asks for control, and shows the user how to configure Windows Mail.
- A user who is having problems installing an application uses Easy Connect to request help from a support technician. Because this is the first time he has requested help from this particular support technician, the user must communicate the password for the session to the support technician using an OOB method such as making a telephone call. The next time the user needs help, however, he will not need to provide a password because of the trust relationship that was established during the first Remote Assistance session between them.

The preceding list is not intended to be complete—other corporate support scenarios using Remote Assistance are possible. Generally speaking, however, corporate environments will use Offer RA to provide assistance to users who phone Help Desk when they have problems. Some enterprises may also allow users to submit Remote Assistance invitations either via e-mail or by saving invitation files to network shares that are monitored by support personnel. Others might use IM applications that support Remote Assistance within the corpnet.

NOTE **Helpers can have multiple Remote Assistance sessions open simultaneously—one session for each User they are supporting. However, Users can have only one Remote Assistance session in the Waiting For Connect state. The invitation that was created could be sent to multiple recipients—any of whom may connect. All subsequent connect attempts will be blocked until the first Helper disconnects, after which another Helper may connect. If the User disconnects the session, the Remote Assistance application terminates and no further connections will be allowed.**

Interoperability with Remote Assistance in Windows Vista

Remote Assistance in Windows 7 is fully backward-compatible with Remote Assistance in Windows Vista, except that Windows Vista does not support the new Easy Connect method for soliciting Remote Assistance found in Windows 7. This means that a User on a Windows Vista computer cannot use Easy Connect to solicit Remote Assistance from a Helper on a Windows 7 computer, and a User on a Windows 7 computer cannot use Easy Connect to solicit Remote Assistance from a Helper on a Windows Vista computer. In addition, a Windows 7 user cannot transfer a file with a Windows Vista user during a Remote Assistance session.

Interoperability with Remote Assistance in Windows XP

Remote Assistance in Windows 7 is backward-compatible with Remote Assistance in Windows XP, with the following limitations:

- Offer RA from Windows 7 to Windows XP is supported, but Offer RA from Windows XP to Windows 7 is not supported. This means that enterprises who want to implement Offer RA as a support solution for their Help Desk departments should ensure that computers used by support personnel who will help users running Windows 7 are themselves running Windows 7 (and not Windows XP).
- NAT traversal using Teredo and IPv6 is supported on Windows 7 to Windows 7 Remote Assistance only, and not on Windows 7 to Windows XP.
- Voice support for Remote Assistance in Windows XP is not supported by Remote Assistance in Windows 7, and any attempt by a User on a Windows XP computer to use this feature during a Remote Assistance session with a Helper on a Windows 7 computer will cause a notification message regarding this limitation to appear.
- The MAILTO method of soliciting assistance that is supported by Remote Assistance in Windows XP is not supported by Remote Assistance in Windows 7.
- Windows Messenger (which shipped with Windows XP) does not ship with Windows 7. Users of Remote Assistance with Windows Messenger in Windows XP will need to migrate to an IM application such as Windows Live Messenger that supports Windows 7 Remote Assistance.
- Offer RA via Windows Live Messenger is supported in Windows 7 but not in Windows XP.
- Windows XP does not support the new Easy Connect method for soliciting Remote Assistance found in Windows 7. This means that a User on a Windows XP computer cannot use Easy Connect to solicit Remote Assistance from a Helper on a Windows 7 computer, and a User on a Windows 7 computer cannot use Easy Connect to solicit Remote Assistance from a Helper on a Windows XP computer. A Windows 7 user cannot transfer a file with a Windows Vista user during a Remote Assistance session.

Implementing and Managing Remote Assistance

Remote Assistance is a powerful and flexible feature that can be used in many different ways to support users within large enterprises, medium-sized businesses, and SOHO environments. This section outlines how to initiate Remote Assistance sessions from both the UI and the command line. This section also demonstrates how to use Remote Assistance in an enterprise Help Desk environment involving two common scenarios:

- Helper offers Remote Assistance to a User who telephones the Help Desk with a problem.
- User creates a Remote Assistance invitation and saves it on a network share that is monitored by Help Desk personnel.

For information on other scenarios for implementing Remote Assistance, including sending invitations with Windows Mail and Windows Messenger, search for the topic "Remote Assistance" within Windows Help and Support.

Initiating Remote Assistance Sessions

Remote Assistance sessions can be initiated from either the UI or the command line. A significant usability enhancement, from the perspective of support personnel, is that Offer RA is no longer buried within Help And Support as it is in Windows XP, but instead is easily accessible now from the graphical user interface (GUI).

Initiating Remote Assistance from the GUI

Initiating Remote Assistance sessions from the GUI can be done using the following methods:

- From the Start menu, click Start, point to All Programs, select Maintenance, and then select Windows Remote Assistance.
- Click Start and type **assist** in the Start menu search box. When Windows Remote Assistance appears in the search results under Programs, click it.

Either of these actions will open the initial Remote Assistance screen, shown in Figure 22-2.

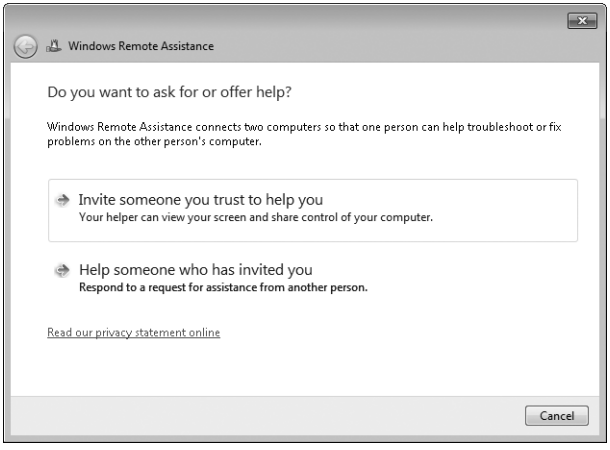


FIGURE 22-2 The initial screen of Windows Remote Assistance

When this initial screen appears, you can do either of the following:

- Solicit Remote Assistance from someone by clicking the Invite Someone You Trust To Help You option, which displays the How Do You Want To Invite Your Trusted Helper? screen, as shown in Figure 22-3.

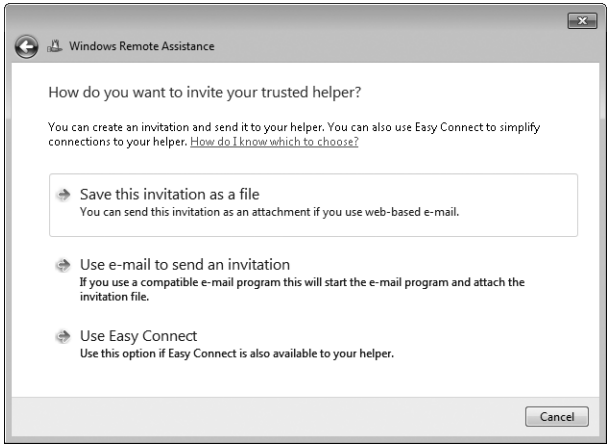


FIGURE 22-3 The screen for soliciting Remote Assistance from someone

- Accept a Remote Assistance invitation from someone or offer Remote Assistance to someone by clicking the Help Someone Who Has Invited You option, which displays the Choose A Way To Connect To The Other Person's Computer screen, as shown in Figure 22-4.

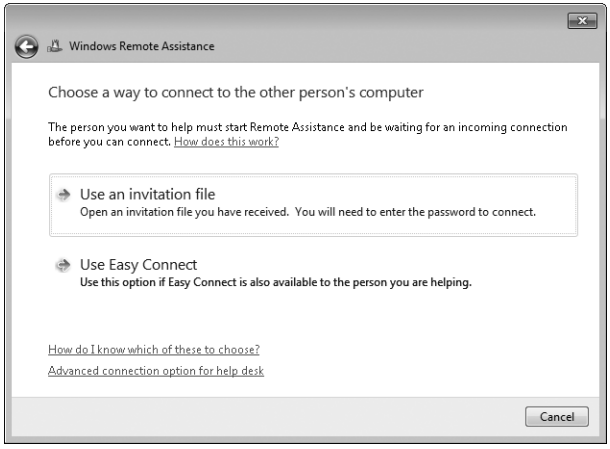


FIGURE 22-4 The screen for offering Remote Assistance to someone

The How Do You Want To Invite Your Trusted Helper? screen (see Figure 22-3) lets you select from the following methods for soliciting Remote Assistance:

- **Save This Invitation To A File** Selecting this option lets you save your Remote Assistance invitation file to a folder on your computer or to an available shared folder on the network.
- **Use E-mail To Send An Invitation** Selecting this option starts your e-mail client application, creates a new message, and attaches the invitation file to the message. Note that if you do not have an SMAPI-compatible e-mail client application on your computer, this option will be unavailable.
- **Use Easy Connect** Selecting this option creates and publishes your Remote Assistance invitation file to the cloud using PNRP and displays a 12-character password that you must communicate OOB to your Helper for him to accept your invitation. If, however, you previously used Easy Connect to establish a Remote Assistance session with the same Helper, the Helper can accept your invitation without any password required.

NOTE **If the computer has IPv6 disabled or is behind a NAT router that blocks Teredo traffic, the Easy Connect option will be unavailable.**

The Choose A Way To Connect To The Other Person's Computer screen (see Figure 22-4) lets you accept a Remote Assistance invitation from someone or offer Remote Assistance to someone. The following options are available on this screen for accepting a Remote Assistance invitation from someone:

- **Use An Invitation File** Selecting this option lets you browse your local file system or network share for the Remote Assistance invitation from someone who needs your help. You will need the password associated with the invitation, which must be provided OOB by the User who needs help.
- **Use Easy Connect** Selecting this option lets you browse the PNRP cloud for the Remote Assistance invitation from someone who needs your help. The first time you use Easy Connect to help this individual, you will need the password associated with the invitation, which must be provided OOB by the User who needs help. For subsequent times that you use Easy Connect to help this individual, the password is not required.

To offer Remote Assistance to someone, click the Advanced Connection Option For Help Desk link at the bottom of Figure 22-4. Additional steps for soliciting and offering Remote Assistance are described in the scenario sections later in this chapter.

HOW IT WORKS

RA Invitation Files

Remote Assistance invitation files (.MsRcIncident) are XML-formatted file documents that include information used by the Helper's computer that will attempt to connect. This ticket information is encrypted to prevent unauthorized users from accessing the information if e-mail or file transfer is used to send the invitation over an unsecured network.

If the e-mail method is used to send the invitation file to the Helper, the invitation file is sent as an e-mail attachment with a filename of RATicket.MsRcIncident. If the file transfer method is used instead, the invitation file is created by default on the desktop of the User's computer, and the filename of the invitation is Invitation.MsRcIncident.

Initiating Remote Assistance from the Command Line

Remote Assistance in Windows 7 and Windows Vista is implemented as a stand-alone executable called Msra.exe. You can initiate Remote Assistance sessions directly from the command line or by using scripts. The syntax and usage for this command is explained in Table 22-3.

TABLE 22-3 Syntax and Usage for Command-Line Remote Assistance (Msra.exe)

OPTION	SUPPORTED ON	DESCRIPTION
/novice	Windows 7 Windows Vista	Starts Remote Assistance as Novice (User) in Solicited RA mode and presents the user with the choice of either sending a Remote Assistance ticket using a SMAPI-enabled e-mail application such as Windows Mail or by saving the invitation as a file. After this choice has been made, Windows Remote Assistance opens on the User's computer in the Waiting For Connect state.
/expert	Windows 7 Windows Vista	Starts Remote Assistance in the Helper mode and presents the choice of either specifying the location of a Remote Assistance ticket to open or specifying the User's computer name or address (Offer RA). The computer name can be either a host name (if the User is on the local subnet) or an FQDN (DNS name), and the address can be either an IPv4 address or an IPv6 address. Unsolicited Remote Assistance without an invitation requires preconfiguration of the remote computer being helped.

OPTION	SUPPORTED ON	DESCRIPTION
/offerRA *computer*	Windows 7 Windows Vista	Starts Remote Assistance as Helper in Unsolicited (Offer) RA mode and uses DCOM to remotely open Remote Assistance on the User's computer and then connect to the User's computer to initiate a Remote Assistance session. The User's computer can be specified using either its computer name or address. The computer name can be either a host name (if the User is on the local subnet) or a FQDN (DNS name), and the address can be either an IPv4 address or an IPv6 address. This method is demonstrated in more detail in the section titled "Scenario 3: Offering Remote Assistance Using DCOM" later in this chapter.
/email *password*	Windows 7 Windows Vista	Starts Remote Assistance as Novice (User) in Solicited RA mode and creates a password-protected RA ticket that is attached to a new Remote Assistance invitation message opened by the default SMAPI-enabled e-mail client (which by default is Windows Mail). The password must be six characters or more and must be relayed separately to the Helper. The e-mail client application launches a window with the invitation file attached. The User must enter the e-mail address of the Helper in the To field to send the message to the Helper.
/saveasfile *path password*	Windows 7 Windows Vista	Starts Remote Assistance as Novice (User) in Solicited RA mode and creates a password-protected Remote Assistance ticket that is saved at the path specified. The path can be either a local folder or network share, and the User must have appropriate permissions on the destination folder to create the file. The path must include a file name for the ticket. (The .MsRcIncident file extension will be automatically added to the file name.) The password must be six characters or more. Use of this method is demonstrated in more detail in the section titled "Scenario 2: Soliciting Remote Assistance by Creating Remote Assistance Tickets and Saving Them on Monitored Network Shares" later in this chapter.

OPTION	SUPPORTED ON	DESCRIPTION
/openfile *path password*	Windows 7 Windows Vista	Starts Remote Assistance as Expert (Helper) in Solicited RA mode and opens a previously created Remote Assistance ticket that was saved within the path specified. The path may be either a local folder or network share, and the Helper must have appropriate permissions on the destination folder to open the file. The path must include the file name of a valid ticket that has the .MsRcIncident file extension. The password must be the same password that was used by the User to secure the ticket when it was created.
/geteasyhelp	Windows 7 only	Starts Remote Assistance as Novice (User) in Solicited RA mode and with the Easy Connect option already selected. After the Remote Assistance invitation has been posted to the PNRP cloud, the User is presented with a 12-character password that she must communicate OOB to the Expert (Helper), which the Helper can then use to accept the invitation and initiate the Remote Assistance session.
/offereasyhelp *address*	Windows 7 only	Starts Remote Assistance as Expert (Helper) in Offer RA mode and with the Easy Connect option already selected. The Helper is presented with a dialog box for entering the 12-character password that was communicated OOB to him by the Novice (User), which is needed by the Helper to accept the invitation and initiate the Remote Assistance session.
/getcontacthelp *address*	Windows 7 only	Starts Remote Assistance as Novice (User) in Solicited RA mode with the Easy Connect option already selected and with the Remote Assistance history contact specified by *address* already selected. You can find *address* for a contact in your Remote Assistance history by opening the RAContacthistory.xml file located in the \Users*Username*\Appdata\Local folder on your computer. The format for *address* is a 40-character hexadecimal string with .RAContact appended to it.

OPTION	SUPPORTED ON	DESCRIPTION
/offercontacthelp *address*	Windows 7 only	Starts Remote Assistance as Expert (Helper) in Offer RA mode with the Easy Connect option already selected and with the Remote Assistance history contact specified by *address* already selected. You can find *address* for a contact in your Remote Assistance history by opening the RAContacthistory.xml file located in the \Users*Username*\Appdata\Local folder on your computer. The format for *address* is a 40-character hexadecimal string with .RAContact appended to it.

NOTE **There is no support for Windows Management Instrumentation (WMI) scripting of Msra.exe.**

Scenario 1: Soliciting Remote Assistance Using Easy Connect

In Windows 7, the simplest way for home users to request assistance from others is to use Easy Connect. (Easy Connect is not intended for enterprise environments because it requires global P2P connectivity to work.) In the following scenario, Tony Allen, a Novice user, requests help from Karen Berg, a friend who is an Expert user. Tony solicits Karen's help for the first time by starting Remote Assistance on his computer and selecting Invite Someone You Trust To Help You followed by Use Easy Connect. At this point, Windows Remote Assistance displays a password, as shown in Figure 22-5.

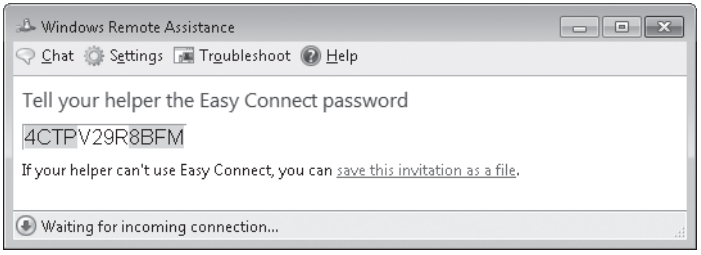


FIGURE 22-5 Tony's computer displays the password needed for Karen to connect using Remote Assistance.

Tony telephones Karen, indicates that he wants her to help him using Remote Assistance, and gives her the password. Karen now starts Remote Assistance on her own computer and selects Help Someone Who Has Invited You. Windows Remote Assistance opens and displays the dialog box shown in Figure 22-6.

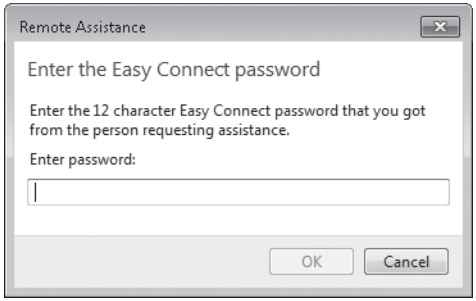


FIGURE 22-6 Karen needs Tony's password to connect to his computer using Remote Assistance.

Karen enters the password Tony has given her and clicks Enter. Karen's computer searches the PNRP cloud for Tony's Remote Assistance invitation and displays Attempting To Connect in the Remote Assistance status bar. When the invitation has been found, the status bar message changes to Waiting For Acceptance. At this point, a dialog box will appear on Tony's computer asking if he would like to allow Karen to connect to his computer and view his desktop (shown in Figure 22-7). Tony has two minutes to respond to this dialog box before the offer times out and the dialog box disappears, which will cause a message saying "The person you are trying to help isn't responding" to appear on Karen's computer.

FIGURE 22-7 Tony must allow the Remote Assistance connection to occur.

Tony clicks Yes and the Remote Assistance session begins. At this point, the desktop properties of Tony's desktop may change (based on configurable settings) to optimize the network bandwidth used by Remote Assistance for screen updates on Karen's computer. Karen can now request control from Tony, send files to Tony or receive files from him, chat with Tony, or disconnect the session. Tony can send and receive files, chat, or pause or disconnect the session.

NOTE **If you are a User and a Helper has shared control of your computer, you can immediately terminate shared control and return the session to Screen Sharing state by pressing the Panic key (Esc).**

If Tony needs help again from Karen on some future occasion, the steps involved are simpler. Tony starts Remote Assistance and selects Invite Someone You Trust To Help You. Remote Assistance displays the history list of recent contacts Tony has used before as Helpers, as shown in Figure 22-8.

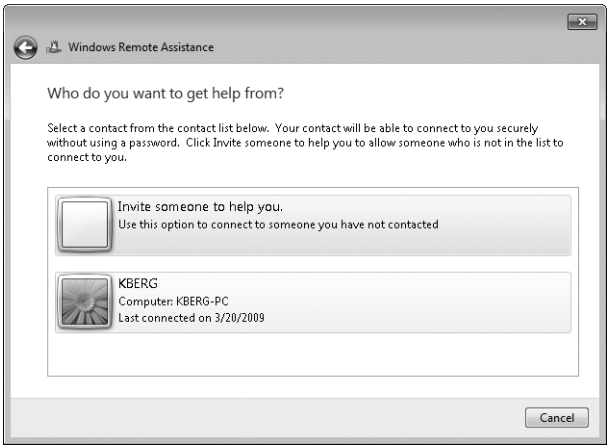


FIGURE 22-8 Karen is listed as a contact in Tony's history list.

Tony clicks Karen's contact info in his history list. This time, instead of a password being displayed, a message appears, indicating that Tony should tell Karen he needs her help (shown in Figure 22-9).

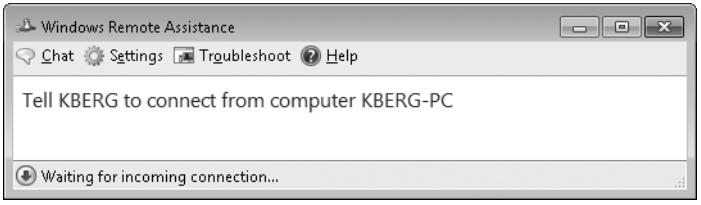


FIGURE 22-9 No password is needed on subsequent requests for help that use Easy Connect.

Tony telephones Karen and asks her to start Remote Assistance on her computer. Karen does this and selects Who Do You Want To Help (shown in Figure 22-10).

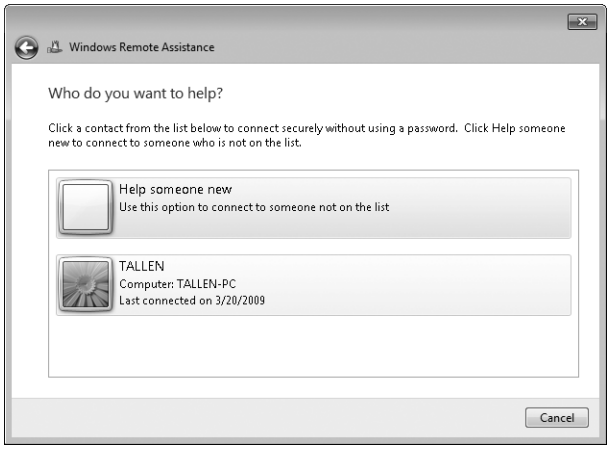


FIGURE 22-10 Tony is listed as a contact in Karen's history list.

Karen clicks Tony's contact info in her history list. Karen's computer searches the PNRP cloud for Tony's Remote Assistance invitation and displays Waiting For Acceptance in the Remote Assistance status bar when the invitation is found. Tony then clicks Yes, and the new Remote Assistance session begins.

You can also use the new command-line switches for Msra.exe in Windows 7 to simplify the Easy Connect experience even further. For example, if Tony frequently needs help from Karen, Karen (the Expert user) could create a shortcut on Tony's desktop that executes the following command:

```
msra.exe /getcontacthelp address
```

Here, *address* is the value of the ADDRESS attribute in Karen's Remote Assistance history contact information on Tony's computer, which is stored as an XML element in the RAContacthistory.xml file located in the \Users\TALLEN\Appdata\Local folder on Tony's computer. The contents of this file might look like the following:

```
<?xml version="1.0"?>
<RAINVITATIONCOLL>
<RAINVITATIONITEM NAME="KBERG" COMPUTERNAME="KBERG-PC" AVATAR="Qk1QgA…[lots of
characters]…"
PUBLICKEY="BgIAAAC…"
ADDRESS="5823b8d7b47af2c1cd94f32535a79d8f0569e7d0.RAContact"
TYPE="1"
TIME="20090320170235.779000"/>
</RAINVITATIONCOLL>
```

Using this example, the shortcut Karen creates on Tony's computer should execute the following command:

```
msra.exe /getcontacthelp 5823b8d7b47af2c1cd94f32535a79d8f0569e7d0.RAContact
```

Karen can then create a similar shortcut on her own computer using Tony's Remote Assistance history contact information, which is stored as an XML element in the RAContacthistory.xml file located in the \Users\KBERG\Appdata\Local folder on Karen's computer. After this is done, Tony can request assistance by simply double-clicking the shortcut on his desktop, and once he has informed Karen of this, Karen then double-clicks the corresponding shortcut on her own computer, and when Tony agrees to allow the connection, the session is started.

DIRECT FROM THE SOURCE

How Easy Connect Works

John Thekkethala, Program Manager
Remote Assistance Team

The new Easy Connect feature simplifies Remote Assistance by enabling a direct P2P transfer of the Remote Assistance invitation using PNRP. When the User starts Remote Assistance and selects Invite Someone You Trust To Help You and then Use Easy Connect, a Remote Assistance invitation is created, encrypted, and published as a payload on a node in the PNRP cloud. This invitation will be retrieved by the Helper from the PNRP cloud and the information is used to establish a Remote Assistance connection to the User.

When the invitation is created, a 12-character alphanumeric password is generated automatically and is displayed in the Tell Your Helper The Easy Connect Password dialog box. The first time the User uses any particular Helper, the password must be relayed OOB to the Helper before the Helper can connect to the User's computer. The password is case insensitive and avoids characters and numbers that could look similar (such as I and 1, 5 and S, and 0 and O).

After the PNRP node has been created in the PNRP cloud, the User's computer waits for an incoming connection from the Helper's computer. This node will exist for 30 minutes before expiring and invalidating the invitation.

The Helper starts Remote Assistance, selects Help Someone Who Has Invited You and then Use Easy Connect, and enters the password relayed OOB from the User. The Helper's computer uses the password to locate the PNRP node containing the User's invitation, grabs the payload (that is, the invitation), and decrypts it. Remote Assistance uses the invitation to connect to the User's computer. Of course, after the Remote Assistance connection has been established, the User must still provide explicit consent before his desktop is remoted.

When a Remote Assistance session has been established using Easy Connect, the User and the Helper become trusted contacts of each other. The Remote Assistance history store on each computer is used to maintain a list of records of trusted contacts that were established using Easy Connect. These records contain the following information for each trusted contact:

- User name
- Computer name
- User graphic (associated with the user logon account)
- Date and time of connection
- Public key of the connected user

Each history record identifies a specific user on a specific computer. A record is created only if each side of the connection has positive confirmation that the other side has received the user's entire contact info. Note that the Remote Assistance contact history does not include the user's role (User or Expert). This means that when trust is established between two user/computer pairs, either one of them may take the role of User and ask the other for assistance.

The next time the User tries to solicit assistance from the same Helper using Easy Connect, the User simply starts Remote Assistance and selects the Helper from the User's Remote Assistance contact list—no password is needed because the Helper is already trusted by the user. The Remote Assistance ticket is exchanged using Secure PNRP. All the User needs to do is notify the Helper that assistance is requested, and this can be done by telephone, IM, or any other OOB method.

After the User has notified the Helper that assistance is requested, the Helper starts Remote Assistance and selects the contact of the user. The Helper's computer uses Secure PNRP to retrieve the Remote Assistance invitation and the Remote Assistance session with the User is established without any password needing to be entered by the Helper.

Scenario 2: Soliciting Remote Assistance by Creating Remote Assistance Tickets and Saving Them on Monitored Network Shares

Another way that you can use Remote Assistance in an enterprise environment is by having users create invitation files and save them on a network share that is monitored by Help Desk personnel. This way, when Help Desk determines that a new ticket has been uploaded to the share, a support person can call the user on the telephone to obtain the password for the ticket and then use the ticket to establish a Remote Assistance session with the user who needs help.

To make the procedure easier, administrators can first deploy a script on users' desktops that uses command-line Remote Assistance (via Msra.exe) to create the invitation file and save it on the network share. For example, let's say that users' invitation files should be uploaded to \\FILESRV3.contoso.com\Support\IncomingTickets, a folder in the Support share on the file server named FILESRV3. The following script, named SubmitTicket.vbs, could be deployed on each user's desktop to accomplish this task.

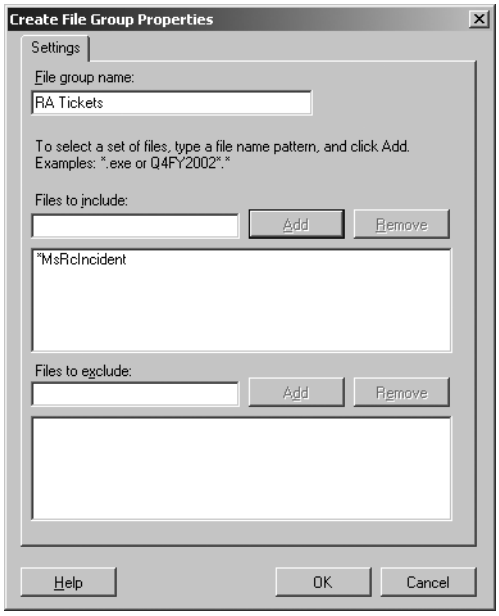
```
dim strPassword
dim strUser
dim strTicketName

strPassword = InputBox("Enter a password for your ticket")
Set WshShell = Wscript.CreateObject("Wscript.Shell")
strUser = WshShell.ExpandEnvironmentStrings("%username%")
strTicketName = strUser & "-" & Year(Now) & "-" & Month(Now) & "-" & Day(Now) & _
   "-" & Hour(Now) & "-" & Minute(Now) & "-" & Second(Now)
strRA = "msra.exe /saveasfile \\FILESRV3\Support\IncomingTickets\" & _
   strTicketName & " " & strPassword
WshShell.Run strRA
```

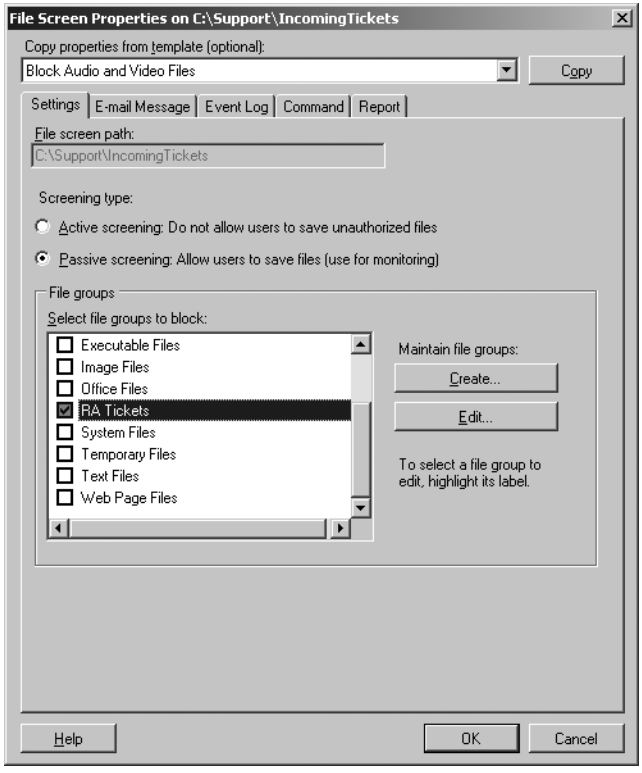
When the user double-clicks this script to run it, an Input box appears asking the user to provide a password to be used to secure the invitation. After the user supplies a password, a new Remote Assistance ticket is created and saved in the target folder on the file server. The name of the ticket is unique and consists of the user's name followed by the date and time, such as *tallen-YYYY-MM-DD-HH-MM-SS.MsRcIncident*. When the support person monitoring the share has obtained the ticket's password using an OOB method such as a telephone call, the support person opens the ticket. After the user grants consent, the Remote Assistance connection is established.

To monitor the IncomingTickets folder in the network share, Help Desk personnel can use the file-screening capabilities of file servers running Windows Server 2008. To do this, perform the following steps to create a passive file screen that monitors the folder and sends an e-mail alert to a Help Desk alias whenever a new ticket is uploaded to the folder:

1. Install or upgrade the File Server role on the Windows Server 2008 computer where the Support folder is located.
2. Start the File Server Resource Manager console from Administrative Tools, right-click the root node, and select Configure Options.
3. Specify the DNS name of the IP address of a Simple Mail Transfer Protocol (SMTP) host that can be used to forward alert e-mails that are generated by the file screen you will create.
4. Click OK to close File Server Resource Manager Options and expand the console tree to select File Screens under File Screening Management.
5. In the Action pane, select the Create File Screen option.
6. Click Browse to select the Incoming folder for the File Screen Path.
7. Select the Define Custom File Screen Properties option and click Custom Properties.
8. Choose the Passive Screening option so that uploaded tickets will only be monitored and not blocked by the screen.
9. Click Create to create a new file group called RA Tickets, and click Add to add files of type *.MsRcIncident to the group.



10. Click OK to return to the properties sheet for the new file screen and select the check box for the RA Tickets file group you just created.



11. Click the E-mail tab and specify a support alias (such as support@contoso.com) that will be notified whenever a new ticket is uploaded to the folder. Configure a suitable subject and body for the message.
12. Click Create to create the new file screen and then choose the option to save the screen without creating a template.
13. Test the new file screen by opening a command prompt on a user's computer and then typing **msra.exe /saveasfile *path password,*** where *path* is the UNC path to the Incoming folder within the Support share on the file server, and *password* is any password of six or more characters that you specify.

MORE INFO **For more information on how to implement file screening in Windows Server 2008, see the topic "Screening Files" in the Microsoft Windows Server TechCenter at *http://technet2.microsoft.com/windowsserver2008/en/library/c16070f8-25f6-4d22-8040-5299b08d6eea1033.mspx?mfr=true*.**

Scenario 3: Offering Remote Assistance Using DCOM

Before you can offer Remote Assistance to other users, your user account must be authorized as a Helper on the User's computer. You should use Group Policy to do this in an enterprise environment. (See the section titled "Managing Remote Assistance Using Group Policy" later in this chapter for information on how to do this.)

After a support person (or group of individuals) has been configured as a Helper for all Windows 7 computers in a domain or organizational unit (OU), the support person can offer Remote Assistance to users of those computers when they need assistance. For this scenario, let's say that Tony Allen (tallen@contoso.com) is a Windows 7 user who needs assistance with an issue on his computer. Tony telephones the Help Desk department, and the call is taken by Karen Berg (kberg@contoso.com), who asks Tony for the name or IP address of his computer. Tony provides Karen with his fully qualified computer name (TALLEN-PC.contoso.com) or IP address. Karen then offers assistance to Tony by starting Remote Assistance on her computer, selecting Help Someone Who Has Invited You, clicking Advanced Connection Option For Help Desk, and entering the name or IP address of Tony's computer (shown in Figure 22-11).

NOTE **Karen could also type msra /offerRA TALLEN-PC.contoso.com at a command prompt to offer assistance quickly to Tony.**

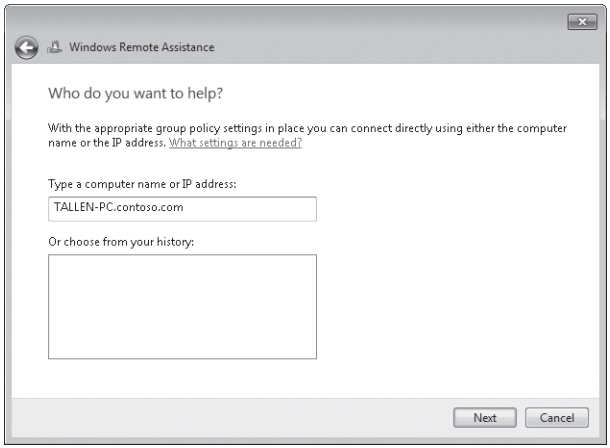


FIGURE 22-11 Karen offers help to Tony using an unsolicited RA.

The experience is even easier if Karen needs to offer help to Tony again on some future occasion. Karen simply starts Remote Assistance on her computer, selects Help Someone Who Has Invited You, and clicks Advanced Connection Option For Help Desk, and the name or IP address of Tony's computer is displayed in her Remote Assistance history list (shown in Figure 22-12).

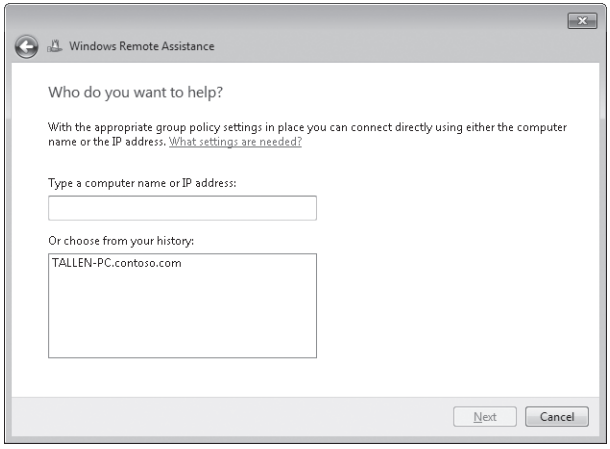


FIGURE 22-12 The history list makes it easy to start Remote Assistance sessions with users that were helped before.

NOTE **Karen can also display the screen in Figure 22-12 quickly by typing msra /offerRA at the command prompt.**

Karen then clicks Tony's computer in her history list and clicks Next, and when Tony accepts the offer, the session begins.

Managing Remote Assistance Using Group Policy

In an enterprise environment, Remote Assistance can be managed using Group Policy. The policy settings for Remote Assistance are all machine settings and are found in the following policy location:

Computer Configuration\Policies\Administrative Templates\System\Remote Assistance

When these policy settings are written to the registry on targeted computers, they are stored under the following registry key:

HKLM\SOFTWARE\Policies\Microsoft\WindowsNT\Terminal Services

Remote Assistance policy settings are summarized in Table 22-4.

TABLE 22-4 Group Policy Settings for Remote Assistance

POLICY	DESCRIPTION
Solicited Remote Assistance	Enabling this policy allows users of targeted computers to use Solicited RA to request assistance using e-mail, file transfer, or IM. Disabling this policy prevents users from using Solicited RA. The default setting is Not Configured, which allows users to change their Remote Assistance settings using the Remote tab of the System item in Control Panel. If the policy is Enabled, you can further configure whether Helpers can be prevented from sharing control of the User's computer, the maximum ticket lifetime, and the method used for sending invitations by e-mail. (Windows 7 does not support the MAILTO method—select SMAPI instead if the targeted computers are running Windows 7.) Ticket lifetime applies only to Remote Assistance invitations sent by e-mail or file transfer. The default ticket lifetime when Group Policy is not being used is six hours. If this policy is Enabled, you must also enable the Remote Assistance exception in Windows Firewall to allow Solicited RA to work. In an unmanaged environment, this setting can also be configured using the Remote tab of the System CPL in Control Panel. This policy is also supported on Windows XP Professional and Windows Server 2003.

POLICY	DESCRIPTION
Offer Remote Assistance	Enabling this policy allows designated Helpers to use Offer RA to offer assistance to users of targeted computers. Disabling this policy or leaving it Not Configured prevents Offer RA from being used to offer assistance to users of targeted computers. If the policy is Enabled, you can further configure whether Helpers can view or control the Users' computers, and you must specify a list of Helpers who are allowed to Offer RA to the users of the targeted computers. Helpers can be either users or groups and must be specified in the form *domain_name**username or domain_name**groupname*. If this policy is Enabled, you must also enable the Remote Assistance exception in Windows Firewall to allow Offer RA to work. (In Windows 7, the Remote Assistance exception is open by default for the domain firewall profile.) This policy is also supported on Windows XP Professional and Windows Server 2003. See the Explain tab of this policy setting for more details.
Allow Only Vista Or Later Connections	The default Windows 7 invitation file includes an XP-specific node for backward compatibility. This node is not encrypted and allows Windows XP computers to connect to the Windows 7 computer that created the ticket. Enabling this policy causes all Remote Assistance invitations generated by users of targeted computers to *not* include the XP node, thereby providing an additional level of security and privacy. Disabling this policy or leaving it Not Configured leaves information such as IP address and port number unencrypted in Remote Assistance invitations This policy setting applies only to Remote Assistance invitations sent using e-mail or file transfer and has no effect on using IM to solicit assistance or on using Offer RA to offer assistance. In an unmanaged environment, this setting can also be configured by clicking Advanced from the Remote tab of the System Properties dialog box. This policy is supported only on Windows Vista and later platforms.
Customize Warning Messages	Enabling this policy causes a specified warning to be displayed on targeted computers when a Helper wants to enter Screen Sharing state or Control Sharing state during a Remote Assistance session. Disabling this policy or leaving it Not Configured causes the default warning to be displayed in each instance. If the policy is Enabled, you can further specify the warning message to be displayed in each instance. This policy is supported only on Windows Vista and later platforms.

POLICY	DESCRIPTION
Turn On Session Logging	Enabling this policy causes Remote Assistance session activity to be logged on the targeted computers. For more information, see the section titled "Remote Assistance Logging" earlier in this chapter. Disabling this policy causes Remote Assistance auditing to be disabled on the targeted computers. The default setting is Not Configured, in which case Remote Assistance auditing is automatically turned on. This policy is supported only on Windows Vista and later platforms.
Turn On Bandwidth Optimization	Enabling this policy causes the specified level of bandwidth optimization to be used to enhance the Remote Assistance experience over low-bandwidth network connections. Disabling this policy or leaving it Not Configured allows the system defaults to be used. If the policy is Enabled, you must specify the level of bandwidth optimization you want to use from the following options: ■ No Optimization ■ No Full Window Drag ■ Turn Off Background ■ Full Optimization If No Optimization is selected, the User's computer will use the Windows Basic theme with full background, and during a shared control session, the Helper will be able to drag full windows across the User's screen. Additional optimization turns off effects to allow a more responsive experience for the Helper. This policy is supported only on Windows Vista and later platforms.

NOTE **In Windows XP, members of the Domain Admins group are granted Helper privileges implicitly even if they are not added to the Helpers list of the Offer Remote Assistance policy setting. This is no longer the case in Windows 7 and Windows Vista, where the Domain Admins group must now be added explicitly to the Helpers list to grant them Helper privileges for Offer RA.**

Configuring Remote Assistance in Unmanaged Environments

Users of unmanaged computers can enable and configure Remote Assistance using the Remote tab of the System CPL in Control Panel (shown in Figure 22-13). Enabling or disabling Remote Assistance and configuring its settings this way requires local administrator credentials on the computer, so a UAC prompt will appear when the user tries to do this.

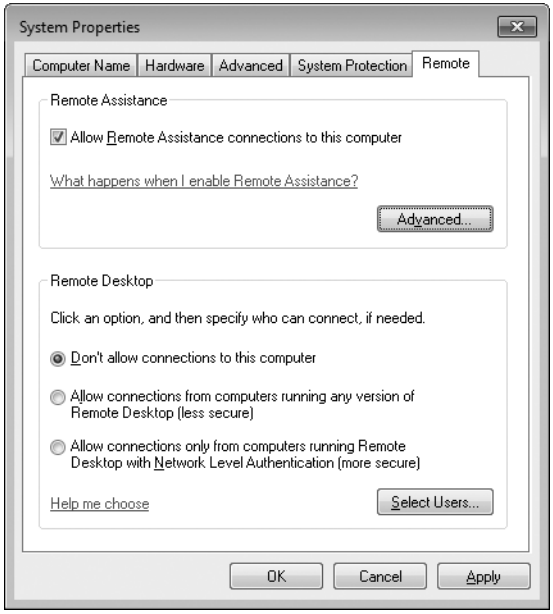


FIGURE 22-13 Configuring Remote Assistance from the Remote tab of System in Control Panel

Note that settings changes made this way will affect all users on the system. Clicking Advanced lets you specify whether remote control of the computer will be allowed during a Remote Assistance session, what the maximum lifetime of a Remote Assistance invitation can be before it times out (the default is six hours), and whether invitations supported only by Remote Assistance in Windows Vista or later versions will be created (see Figure 22-14).

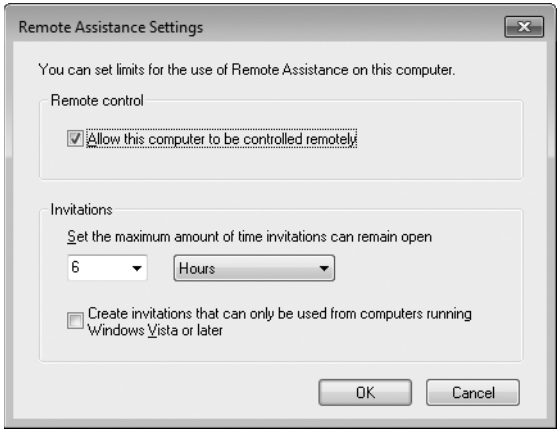


FIGURE 22-14 Advanced configuration settings for Remote Assistance

NOTE **A PNRP invitation is valid for only 30 minutes and does not use this setting. This limitation does not apply to trusted contact-based invitations.**

In managed environments, when the following Group Policy setting is Enabled, the Control Panel settings for configuring Remote Assistance become unavailable (appear dimmed):

Computer Configuration\Policies\Administrative Templates\System\Remote Assistance \Solicited Remote Assistance

Additional Registry Settings for Configuring Remote Assistance

Additional behavior for Remote Assistance can be configured by modifying certain registry settings. Specifically, per-user registry settings for Remote Assistance are found under the following key:

HKCU\Software\Microsoft\Remote Assistance

These settings are changeable when in the Waiting To Connect mode or when in the connected mode from the Settings button.

WARNING **If Group Policy is used to manage Remote Assistance settings and any configured policy settings overlap these registry settings, the policy settings prevail.**

DIRECT FROM THE SOURCE

Troubleshooting Remote Assistance in Windows 7 and Windows Vista

John Thekkethala, Program Manager
Remote Assistance Team

When I attempt to create an invitation with e-mail or save-to-file, I see a warning message stating that Windows Firewall is currently blocking Remote Assistance.

The Remote Assistance firewall exception will change depending on your network location (Private, Public, or Domain). If you are at home, your network location type should be set to Private, which enables the Remote Assistance firewall exception automatically. If your network location is set to Public, the Remote Assistance firewall exception is not enabled automatically for security purposes. It will need to be enabled by an administrator.

If you are connected to a managed network (for example, when you are within a corporate domain), the network location is categorized as Domain, and the Remote Assistance exception is not enabled automatically. It is expected to be configured by Group Policy by your system administrator.

I cannot use Remote Assistance to connect from my home computer to a work computer.

Remote Assistance uses Teredo (IPv6) to traverse NATs. However, Teredo cannot be used to traverse corporate edge firewalls that provide NAT for intranet clients and block dynamic ports or outbound UDP traffic. Because you do not have a globally reachable IPv4 address within the corpnet, Remote Assistance cannot make a connection to you from outside the corpnet.

If I disable the Windows Firewall, I cannot make a Remote Assistance connection in certain cases. This is counterintuitive, because I expect connectivity to be less restrictive with the firewall disabled.

In Windows 7 and Windows Vista, the Windows Firewall is IPv6 aware. The Remote Assistance exception in the Windows Firewall enables Teredo for edge traversal. If the Windows Firewall is disabled, the ability to use Teredo for NAT traversal is also disabled. The Windows Firewall must be running with the Remote Assistance exception enabled for Remote Assistance to be able to traverse NATs using Teredo.

I cannot use Remote Assistance to connect from my work computer to my home computer.

Your corporate firewall may be configured to block outbound P2P connections. In a managed environment (domain-joined computers), which is typically found in a corporate network, the Remote Assistance exception does not enable Teredo (edge traversal), because corporate firewalls typically block outbound UDP traffic. NAT traversal using Teredo is disabled by default in this scenario. If the person you are trying to help is behind a UPnP NAT or is connected directly to the Internet, you should be able to make a connection. Check with your network administrator to see whether outbound P2P connections through the corporate firewall can be enabled.

When I move my laptop (or change my home network location) from a private to a public location, I am not able to connect to certain computers.

If you have a laptop that moves between work and home, the properties of the Remote Assistance firewall exception in the Windows Firewall will change depending on whether your network location is classified as Private, Public, or Domain. In a Private location, the Remote Assistance exception is enabled by default. If you are using a UPnP NAT, the Remote Assistance exception will allow communications with the UPnP NAT to enable Remote Assistance connections that make use of UPnP. In a Public network, the Remote Assistance exception is not enabled by default and will need to be enabled using administrator credentials. In addition, the default

Public profile does not permit UPnP communication for security purposes, thereby restricting Remote Assistance connectivity in certain cases.

I am on a low-bandwidth connection, and the person helping me is experiencing slow screen refreshes.

Under Settings, set the Bandwidth Usage to Low to reduce the bandwidth used during a Remote Assistance connection. Keep in mind that display quality decreases as bandwidth usage is limited.

Why can't I connect to Windows XP computers that are behind a NAT as easily as I can connect to Windows 7 or Windows Vista computers?

Remote Assistance in Windows XP does not support Teredo for NAT traversal. Consequently, a Windows 7– or Windows Vista–to–Windows XP Remote Assistance connection may fail in cases in which both computers are behind non-UPnP NATs.

How does Remote Assistance make a connection?

When the Remote Assistance invitation is created, the User's computer will set itself as a listener on all of its IP addresses (IPv4 and IPv6), including its Teredo address. All of these listeners are waiting for a connection from the Helper's computer. The address and port information associated with these different listeners is relayed to the Helper's computer via the Remote Assistance invitation (which gets transported by Windows Messenger when Messenger is used to launch Remote Assistance). The Helper's computer then tries to connect concurrently on all the address/port pairs in the invitation. The first successful connection that is made is used for the Remote Assistance session and the rest of the connection attempts are terminated.

How do I troubleshoot a connection failure between two home-based Windows 7 or Windows Vista computers that are behind NATs?

Refer to the Remote Assistance Connectivity information in Tables 22-5 and 22-6 to verify that the network configuration you have is supported for Remote Assistance connectivity. Then confirm that the Windows Firewall on the computer of the person that is being helped is running and configured for Remote Assistance as follows:

- The Windows Firewall is IPv6 compatible and must be running to enable NAT traversal using Teredo.
- The network location of the computer must be set to Private or Public because Teredo is not enabled in Domain or Managed settings.
- The Remote Assistance exception in the firewall must be enabled to allow Remote Assistance connections.

Now check that there is no edge firewall between the User and Helper because it may block P2P applications like Remote Assistance.

Finally, confirm that the User and Helper are not behind a symmetric NAT and that Teredo is able to get to the Qualified state on both computers. To determine this, do the following:

1. First, initiate Teredo by forcing Remote Assistance into the Waiting To Connect state. You can do this by typing msra.exe /saveasfile myinvitation mypassword at a command prompt.
2. Next, check to see if Teredo can be activated on both computers and goes into the Qualified state. Open an elevated command prompt window and type netsh interface teredo show state at the command prompt. The output should show Teredo in the Qualified state. If Teredo does not go to the Qualified state on both computers, a Remote Assistance connection may not be possible between these two computers. Teredo will not go into the Qualified state if one of the following two conditions exists:
    - A global Teredo server could not be reached at teredo.ipv6.microsoft.com.
    - The computer is behind a symmetric NAT. To verify this, look at the output of netsh interface teredo show state and check the output on the NAT: line, which specifies NAT type.

When I am helping someone who is a standard user, I cannot run a program that needs administrator privileges even though I have administrator privileges to the User's computer.

Remote Assistance allows a User to share control of his computer with a remote Helper. If the User is a standard user, the remote Helper is given the same privileges as the standard user. If the Helper attempts to start a program that requires administrator credentials, by default these credentials must be entered locally (on the Secure Desktop) by the User and cannot be entered remotely by the Helper. This is required to prevent a security loophole where Admin programs started by a remote Helper could be hijacked by the local User simply by terminating the Remote Assistance session. In managed environments in which client computers are running Windows Vista Service Pack 1 (SP1) or later versions, however, a new Group Policy setting can be enabled that allows Remote Assistance to turn off the Secure Desktop during a Remote Assistance session even if the User is a standard user. As a result, the remote Helper can now enter administrator credentials when a UAC prompt appears during a Remote Assistance session to perform Admin-level tasks on the User's computer. To configure this behavior, enable the following policy setting:

> Computer Configuration\Policies\Windows Settings\Security Settings\Local Policies\Security Options\User Account Control: Allow UIAccess Applications To Prompt For Elevation Without Using The Secure Desktop

Cross-Platform Connectivity for Remote Assistance

For environments in which different versions of Windows are used, Tables 22-5 and 22-6 summarize the Remote Assistance connectivity between Expert and Novice users on computers running Windows XP, Windows Vista, and Windows 7.

TABLE 22-5 Remote Assistance Connectivity for Expert on Windows XP

		EXPERT ON WINDOWS XP			
		Directly Connected	**Behind UPnP NAT**	**Behind non-UPnP NAT**	**Behind Corporate Edge Firewall****
NOVICE (USER) ON WINDOWS XP	**Directly Connected**	Yes	Yes	Yes	Yes
	Behind UPnP NAT	Yes	Yes	Yes	Yes
	Behind non-UPnP NAT	Yes, using Msgr Only	Yes, using Msgr Only	No	No
	Behind Corporate Edge Firewall**	Yes, using Msgr Only	Yes, using Msgr Only	No	Yes, if both are behind same firewall No, if both are behind different firewalls
NOVICE (USER) ON WINDOWS 7 OR WINDOWS VISTA	**Directly Connected**	Yes	Yes	Yes	Yes
	Behind UPnP NAT	Yes	Yes	Yes	Yes
	Behind non-UPnP NAT	Yes, using Msgr Only	Yes, using Msgr Only	No	No
	Behind Corporate Edge Firewall**	Yes, using Msgr Only	Yes, using Msgr Only	No	Yes, if both are behind same firewall No, if both are behind different firewalls

TABLE 22-6 Remote Assistance Connectivity for Expert on Windows Vista and Windows 7

		EXPERT ON WINDOWS VISTA AND WINDOWS 7			
		Directly Connected	**Behind UPnP NAT**	**Behind non-UPnP NAT**	**Behind Corporate Edge Firewall****
NOVICE (USER) ON WINDOWS XP	**Directly Connected**	Yes	Yes	Yes	Yes
	Behind UPnP NAT	Yes	Yes	Yes	Yes
	Behind non-UPnP NAT	Yes, using Msgr Only	Yes, using Msgr Only	No	No
	Behind Corporate Edge Firewall**	Yes, using Msgr Only	Yes, using Msgr Only	No	Yes, if both are behind same firewall No, if both are behind different firewalls
NOVICE (USER) ON WINDOWS 7 OR WINDOWS VISTA	**Directly Connected**	Yes	Yes	Yes	Yes
	Behind UPnP NAT	Yes	Yes	Yes	Yes
	Behind non-UPnP NAT	Yes, using Teredo*	Yes, using Teredo*	Yes, using Teredo*	None
	Behind Corporate Edge Firewall**	No	No	No	Yes, if both are behind same firewall No, if both are behind different firewalls

**Teredo connectivity is not available if both computers are behind Symmetric NATs.*

***Edge Firewall must permit outbound connection (for example, using the Microsoft ISA Firewall Client).*

Summary

Remote Assistance has been enhanced in Windows 7 and Windows Vista to provide better performance, improved usability, NAT-traversal flexibility, and increased security. Best practices for implementing Remote Assistance in an enterprise environment include the following:

- Use Group Policy to enable users of targeted computers in a domain or OU to receive offers of Remote Assistance from Help Desk personnel.
- Use Group Policy to enable the Remote Assistance exception in the Windows Firewall.
- Use Group Policy to deploy scripts to enable users to run the Msra.exe executable if you want to customize how they launch Remote Assistance sessions—for example, to upload an invitation to a network share monitored by support personnel.
- If all of your support computers are running Windows 7 or Windows Vista, use Group Policy to encrypt Remote Assistance tickets to hide sensitive information such as users' IP addresses and computer names.
- If corporate policy requires Remote Assistance records for auditing purposes, use Group Policy to enable Remote Assistance logging on your company's desktop computers and run scripts to periodically move both Helper and User Remote Assistance logs to a safe storage.
- To meet corporate privacy and security requirements, use Group Policy to customize the text message that users see before they allow the Helper to view their screens or share control.

Additional Resources

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

Related Information

- "Windows Remote Assistance: Frequently Asked Questions" at *http://windowshelp.microsoft.com/Windows/en-US/Help/398b5eda-aa7f-4078-94c5-1519b697bfa01033.mspx*.

On the Companion Media

- RemoteAssistanceDiag.ps1

CHAPTER 28

Deploying IPv6

Like the Windows Vista operating system before it, the Windows 7 operating system has a new Next Generation Transmission Control Protocol/Internet Protocol (TCP/IP) stack with enhanced support for Internet Protocol version 6 (IPv6). This chapter provides you with an understanding of why IPv6 is necessary and how it works. The chapter describes the IPv6 capabilities in Windows 7, Windows Vista, and Windows Server 2008 and outlines how to migrate the IPv4 network infrastructure of your enterprise to IPv6 using IPv6 transition technologies, such as Intra-Site Automatic Tunnel Addressing Protocol (ISATAP). Finally, the chapter describes how to configure and manage IPv6 settings in Windows 7 and how to troubleshoot IPv6 networking problems.

Understanding IPv6

The need for migrating enterprise networks from IPv4 to IPv6 is driven by a number of different technological, business, and social factors. The most important of these are:

- The exponential growth of the Internet is rapidly exhausting the existing IPv4 public address space. A temporary solution to this problem has been found in Network Address Translation (NAT), a technology that maps multiple private (intranet) addresses to a (usually) single, public (Internet) address. Unfortunately, using NAT-enabled routers can introduce additional problems, such as breaking end-to-end connectivity and security for some network applications. In addition, the rapid proliferation of mobile IP devices is accelerating the depletion of the IPv4 public address space.

- The growing use of real-time communications (RTC) on the Internet, such as Voice over IP (VoIP) telephony, instant messaging (IM), and audio/video conferencing, exposes the limited support for Quality of Service (QoS) currently provided in IPv4. These new RTC technologies need improved QoS on IP networks to ensure reliable end-to-end communications. The design of IPv4 limits possible improvements.
- The growing threats faced by hosts on IPv4 networks connected to the Internet can be mitigated considerably by deploying Internet Protocol security (IPsec), both on private intranets and on tunneled connections across the public Internet. However, IPsec was designed as an afterthought to IPv4 and is complex and difficult to implement in many scenarios.

IPv6, developed by the Internet Engineering Task Force (IETF) to solve these problems, includes the following improvements and additions:

- IPv6 increases the theoretical address space of the Internet from 4.3×10^9 addresses (based on 32-bit IPv4 addresses) to 3.4×10^{38} possible addresses (based on 128-bit IPv6 addresses), which most experts agree should be more than sufficient for the foreseeable future.
- The IPv6 address space is designed to be hierarchical rather than flat in structure, which means that routing tables for IPv6 routers can be smaller and more efficient than for IPv4 routers.
- IPv6 has enhanced support for QoS that includes a Traffic Class field in the header to specify how traffic should be handled and a new Flow Label field in the header that enables routers to identify packets that belong to a traffic flow and handle them appropriately.
- IPv6 now requires IPsec support for standards-based, end-to-end security across the Internet. The new QoS enhancements work even when IPv6 traffic is encrypted using IPsec.

Understanding how IPv6 works is essential if you plan to benefit from IPv6 by deploying it in your enterprise. The following sections provide an overview of key IPv6 concepts, features, and terminology.

NOTE **For more detailed information on IP concepts, features, and terminology, see the white paper titled "Introduction to IP Version 6" at *http://www.microsoft.com/downloads/details.aspx?FamilyID=CBC0B8A3-B6A4-4952-BBE6-D976624C257C&displaylang=en*. Another good reference for learning IPv6 is the book, *Understanding IPv6, 2nd Edition*, by Joseph Davies (Microsoft Press, 2008).**

Understanding IPv6 Terminology

The following terminology is used to define IPv6 concepts and describe IPv6 features:

- **Node** An IPv6-enabled network device that includes both hosts and routers.
- **Host** An IPv6-enabled network device that cannot forward IPv6 packets that are not explicitly addressed to itself. A host is an endpoint for IPv6 communications (either the source or destination) and drops all traffic not explicitly addressed to it.
- **Router** An IPv6-enabled network device that can forward IPv6 packets that are not explicitly addressed to itself. IPv6 routers also typically advertise their presence to IPv6 hosts on their attached links.
- **Link** One or more LAN (such as Ethernet) or wide area network (WAN, such as Point-to-Point Protocol [PPP]) network segments bounded by routers. Like interfaces, links may be either physical or logical.
- **Neighbors** Nodes that are connected to the same physical or logical link.
- **Subnet** One or more links having the same 64-bit IPv6 address prefix.
- **Interface** A representation of a node's attachment to a link. This can be a physical interface (such as a network adapter) or a logical interface (such as a tunnel interface).

NOTE **An IPv6 address identifies an interface, not a node. A node is identified by having one or more unicast IPv6 addresses assigned to one of its interfaces.**

Understanding IPv6 Addressing

IPv6 uses 128-bit (16-byte) addresses that are expressed in colon-hexadecimal form. For example, in the address 2001:DB8:3FA9:0000:0000:0000:00D3:9C5A, each block of 4-digit hexadecimal numbers represents a 16-bit digit binary number. The eight blocks of four-digit hexadecimal numbers thus equal 8 × 16 = 128 bits in total.

You can shorten colon-hexadecimal addresses by suppressing leading zeros for each block. Using this technique, the representation for the preceding address now becomes 2001:DB8:3FA9:0:0:0:D3:9C5A.

You can shorten colon-hexadecimal addresses even further by compressing contiguous 0 (hex) blocks as double colons ("::"). The address in this example thus shortens to 2001:DB8:3FA9::D3:9C5A. Note that only one double colon can be used per IPv6 address to ensure unambiguous representation.

Understanding IPv6 Prefixes

An IPv6 prefix indicates the portion of the address used for routing (a subnet or a set of subnets as a summarized route) or for identifying an address range. IPv6 prefixes are expressed in a manner similar to the Classless Inter-Domain Routing (CIDR) notation used by IPv4. For example, 2001:DB8:3FA9::/48 might represent a route prefix in an IPv6 routing table.

In IPv4, CIDR notation can be used to represent individual unicast addresses in addition to routes and subnets. IPv6 prefixes, however, are used only to represent routes

and address ranges, not unicast addresses. Unlike IPv4, IPv6 does not support variable-length subnet identifiers, and the number of high-order bits used to identify a subnet in IPv6 is almost always 64. It is thus redundant to represent the address in our example as 2001:DB8:3FA9::D3:9C5A/64; the /64 portion of the representation is understood.

Understanding IPv6 Address Types

IPv6 supports three different address types:

- **Unicast** Identifies a single interface within the scope of the address. (The scope of an IPv6 address is that portion of your network over which this address is unique.) IPv6 packets with unicast destination addresses are delivered to a single interface.
- **Multicast** Identifies zero or more interfaces. IPv6 packets with multicast destination addresses are delivered to all interfaces listening on the address. (Generally speaking, multicasting works the same way in IPv6 as it does in IPv4.)
- **Anycast** Identifies multiple interfaces. IPv6 packets with anycast destination addresses are delivered to the nearest interface (measured by routing distance) specified by the address. Currently, anycast addresses are assigned only to routers and can only represent destination addresses.

NOTE **IPv6 address types do not include broadcast addresses as used by IPv4. In IPv6, all broadcast communications are performed using multicast addresses. See Table 28-2 for more information on multicast addresses.**

Understanding Unicast Addresses

Unicast addresses are addresses that identify a single interface. IPv6 has several types of unicast addresses:

- **Global unicast address** An address that is globally routable over the IPv6-enabled portion of the Internet. Therefore, the scope of a global address is the entire Internet, and global addresses in IPv6 correspond to public (non-RFC 1918) addresses used in IPv4. The address prefix currently used for global addresses as defined in RFC 3587 is 2000::/3, and a global address has the following structure:
  - The first 48 bits of the address are the global routing prefix specifying your organization's site. (The first three bits of this prefix must be 001 in binary notation.) These 48 bits represent the public topology portion of the address, which represents the collection of large and small Internet service providers (ISPs) on the IPv6 Internet and which is controlled by these ISPs through assignment by the Internet Assigned Numbers Authority (IANA).
  - The next 16 bits are the subnet ID. Your organization can use this portion to specify up to 65,536 unique subnets for routing purposes inside your organization's site. These 16 bits represent the site topology portion of the address, which your organization has control over.

- The final 64 bits are the interface ID and specify a unique interface within each subnet.

- **Link-local unicast address** An address that can be used by a node for communicating with neighboring nodes on the same link. Therefore, the scope of a link-local address is the local link on the network; link-local addresses are never forwarded beyond the local link by IPv6 routers. Because link-local addresses are assigned to interfaces using IPv6 address autoconfiguration, link-local addresses in IPv6 correspond to Automatic Private IP Addressing (APIPA) addresses used in IPv4 (which are assigned from the address range 169.254.0.0/16). The address prefix used for link-local addresses is FE80::/64, and a link-local address has the following structure:
  - The first 64 bits of the address are always FE80:0:0:0 (which will be shown as FE80::).
  - The last 64 bits are the interface ID and specify a unique interface on the local link.

  Link-local addresses can be reused—in other words, two interfaces on different links can have the same address. This makes link-local addresses ambiguous; an additional identifier called the zone ID (or scope ID) indicates to which link the address is either assigned or destined. In Windows 7, the zone ID for a link-local address corresponds to the interface index for that interface. You can view a list of interface indexes on a computer by typing **netsh interface ipv6 show interface** at a command prompt. For more information on the zone ID, see the section titled "Displaying IPv6 Address Settings" later in this chapter.
- **Unique local unicast address** Because a site-local address prefix can represent multiple sites within an organization, it is ambiguous and not well suited for intraorganizational routing purposes. Therefore, RFC 4193 currently proposes a new type of address called a unique local unicast address. The scope of this address is global to all sites within the organization, and using this address type simplifies the configuration of an organization's internal IPv6 routing infrastructure. A unique local address has the following structure:
  - The first seven bits of the address are always 1111 110 (binary) and the eighth bit is set to 1, indicating a unique local address. This means that the address prefix is always FD00::/8 for this type of address.
  - The next 40 bits represent the global ID, a randomly generated value that identifies a specific site within your organization.
  - The next 16 bits represent the subnet ID and can be used for further subdividing the internal network of your site for routing purposes.
  - The last 64 bits are the interface ID and specify a unique interface within each subnet.

NOTE **Site-local addresses have been deprecated by RFC 3879 and are replaced by unique local addresses.**

Identifying IPv6 Address Types

As Table 28-1 shows, you can quickly determine which type of IPv6 address you are dealing with by looking at the beginning part of the address—that is, the high-order bits of the address. Tables 28-2 and 28-3 also show examples of common IPv6 addresses that you can recognize directly from their colon-hexadecimal representation.

TABLE 28-1 Identifying IPv6 Address Types Using High-Order Bits and Address Prefix

ADDRESS TYPE	HIGH-ORDER BITS	ADDRESS PREFIX
Global unicast	001	2000::/3
Link-local unicast	1111 1110 10	FE80::/64
Unique local unicast	1111 1101	FD00::/8
Multicast	1111 1111	FF00::/8

TABLE 28-2 Identifying Common IPv6 Multicast Addresses

FUNCTION	SCOPE	REPRESENTATION
All-nodes multicast	Interface-local	FF01::1
All-nodes multicast	Link-local	FF02::1
All-routers multicast	Interface-local	FF01::2
All-routers multicast	Link-local	FF02::2
All-routers multicast	Site-local	FF05::2

TABLE 28-3 Identifying Loopback and Unspecified IPv6 Addresses

FUNCTION	REPRESENTATION
Unspecified address (no address)	::
Loopback address	::1

NOTE **For information on IPv6 address types used by different IPv6 transition technologies, see the section titled "Planning for IPv6 Migration" later in this chapter.**

Understanding Interface Identifiers

For all the types of unicast IPv6 addresses described in the preceding sections, the last 64 bits of the address represent the interface ID and are used to specify a unique interface on a local link or subnet. In previous versions of Windows, the interface ID is uniquely determined as follows:

- For link-local addresses, such as a network adapter on an Ethernet segment, the interface ID is derived from either the unique 48-bit media access control (MAC)–layer address of the interface or the unique Extended Unique Identifier (EUI)–64 address of the interface as defined by the Institute of Electrical and Electronics Engineers (IEEE).
- For global address prefixes, an EIU-64–based interface ID creates a public IPv6 address.
- For global address prefixes, a temporary random interface ID creates a temporary address. This approach is described in RFC 3041; you can use it to help provide anonymity for client-based usage of the IPv6 Internet.

In Windows 7, however, the interface ID by default is randomly generated for all types of unicast IPv6 addresses assigned to LAN interfaces.

> ***NOTE*** **Windows 7 randomly generates the interface ID by default. You can also disable this behavior by typing netsh interface ipv6 set global randomizedidentifiers=disabled at a command prompt.**

Comparing IPv6 with IPv4

Table 28-4 compares and contrasts the IPv4 and IPv6 addressing schemes.

TABLE 28-4 IPv4 vs. IPv6 Addressing

FEATURE	IPv4	IPv6
Number of bits (bytes)	32 (4)	128 (16)
Expressed form	Dotted-decimal	Colon-hexadecimal
Variable-length subnets	Yes	No
Public addresses	Yes	Yes (global addresses)
Private addresses	Yes (RFC 1918 addresses)	Yes (unique local addresses)
Autoconfigured addresses for the local link	Yes (APIPA)	Yes (link-local addresses)
Support for address classes	Yes, but deprecated by CIDR	No
Broadcast addresses	Yes	Multicast used instead
Subnet mask	Required	Implicit 64-bit address prefix length for addresses assigned to interfaces

NOTE **For detailed specifications concerning IPv6 addressing, see RFC 4291 at *http://www.ietf.org/rfc/rfc4291.txt*. There are also other differences between IPv4 and IPv6, such as how the headers are structured for IPv4 versus IPv6 packets. For more information, see the white paper, "Introduction to IP Version 6," at *http://www.microsoft.com/downloads/details.aspx?FamilyID=CBC0B8A3-B6A4-4952-BBE6-D976624C257C&displaylang=en*.**

Understanding IPv6 Routing

Routing is the process of forwarding packets between connected network segments and is the primary function of IPv6. An IPv6 network consists of one or more network segments, also called *links* or *subnets*. These links are connected by IPv6 routers, devices that forward IPv6 packets from one link to another. These IPv6 routers are typically third-party hardware devices, but you can also configure a multihomed computer running Windows Server 2008 as an IPv6 router if needed.

How IPv6 Routing Works

The header of an IPv6 packet contains both the source address of the sending host and the destination address of the receiving host. When an IPv6 packet arrives at a host, the host uses its local IPv6 routing table to determine whether to accept the packet or forward it to another host or network.

Each IPv6 node (host or router) has its own IPv6 routing table. A *routing table* is a collection of routes that store information about IPv6 network prefixes and how they can be reached, either directly or indirectly. On IPv6 hosts, such as computers running Windows 7, Windows Vista, or Windows Server 2008, the IPv6 routing table is generated automatically when IPv6 initializes on the system. Local administrators can use the *netsh interface ipv6* commands to manage these tables by viewing them and by manually adding or removing routes. The use of this command is discussed further later in this section.

When an IPv6 packet arrives at a physical or logical network interface on an IPv6 host, such as a multihomed computer running Windows Server 2008, the host uses the following process to determine how to forward the packet to its intended destination:

1. The host checks its destination cache to see whether there is an entry that matches the destination address in the packet header. If such an entry is found, the host forwards the packet directly to the address specified in the destination cache entry and the routing process ends.
2. If the destination cache does not contain an entry that matches the destination address in the packet header, the host uses its local routing table to determine how to forward the packet. Using the routing table, the host determines the following:
   - **Next-hop address** If the destination address is on the local link, the next-hop address is simply the destination address in the packet header. If the destination

address is on a remote link, the next-hop address is the address of a router connected to the local link.

- **Next-hop interface** This is the physical or logical network interface on the host that should be used to forward the packet to the next-hop address.

3. The host then forwards the packet to the next-hop address using the next-hop interface. The host also updates its destination cache with this information so that subsequent packets sent to the same destination address can be forwarded using the destination cache entry instead of using its local routing table.

IPv6 Route Determination Process

In step 2 of the preceding procedure, the host determines the next-hop address and next-hop interface by using its local routing table. The details of this process are as follows:

1. For each routing table entry, the first *N* bits in the route's network prefix are compared with the same bits in the destination address in the packet header, where *N* is the number of bits in the route's prefix length. If these bits match, the route is determined to be a match for the destination.
2. The list of all matching routes is compiled. If only one matching route is found, this route is chosen and the route determination process is ended.
3. If multiple matching routes are found, the matching route having the largest prefix length is chosen and the route determination process is ended.
4. If multiple matching routes having the largest prefix length are found, the matching route having the lowest metric is chosen and the route determination process is ended.
5. If multiple matching routes having the largest prefix length and lowest metric are found, one of these routes is selected and the route determination process is ended.

The effective result of this IPv6 route determination process is as follows:

1. If a route can be found that matches the entire destination address in the packet header, then the next-hop address and interface specified in this route are used to forward the packet.
2. If a route of the type described in step 1 is not found, the most efficient (that is, lowest-metric) route that has the longest prefix length matching the destination address is used to forward the packet.
3. If a route of the type described in step 2 is not found, the packet is forwarded using the default route (with network prefix ::/0).

IPv6 Routing Table Structure

IPv6 routing tables can contain four different types of routing table entries (that is, routes):

- **Directly attached network routes** These typically have 64-bit prefixes and identify adjacent links (network segments connected to the local segment via one router).

- **Remote network routes** These have varying prefixes and identify remote links (network segments connected to the local segment via several routers).
- **Host routes** These have 128-bit prefixes and identify a specific IPv6 node.
- **Default route** This uses the network prefix ::/0 and is used to forward packets when a network or host route cannot be determined.

On a computer running Windows 7, Windows Vista, or Windows Server 2008, you can use the *netsh interface ipv6 show route* command to display the IPv6 routing table entries. The following is a sample routing table from a domain-joined computer running Windows 7 that has a single LAN network adapter, no IPv6 routers on the attached subnet, and no other configured network connections.

```
Publish  Type      Met  Prefix                    Idx  Gateway/Interface Name
-------  --------  ---  ------------------------  ---  -------------------------
No       Manual    256  ::1/128                     1  Loopback Pseudo-Interface 1
No       Manual    256  fe80::/64                  15  Teredo Tunneling Pseudo-
                                                       Interface
No       Manual    256  fe80::/64                  12  Local Area Connection
No       Manual    256  fe80::100:7f:fffe/128      15  Teredo Tunneling Pseudo-
                                                       Interface
No       Manual    256  fe80::5efe:172.16.11.131/128  14  isatap.{9D607D7D-0703-4E67-
                                                       82ED-9A8206377C5C}
No       Manual    256  fe80::5da9:fa1d:2575:c766/128  12  Local Area Connection
No       Manual    256  ff00::/8                    1  Loopback Pseudo-Interface 1
No       Manual    256  ff00::/8                   15  Teredo Tunneling Pseudo-
                                                       Interface
No       Manual    256  ff00::/8                   12  Local Area Connection
```

Each route in this table is specified using the following fields:

- **Publish** If set to Yes, the route is advertised in a routing Advertisement message; otherwise No.
- **Type** If set to Autoconf, the route was configured automatically using the IPv6 routing protocol; if Manual, the route has been configured by the operating system or an application.
- **Met** Indicates the metric for the route. For multiple routes having the same prefix, the lower the metric, the better the match.
- **Prefix** Specifies the address prefix for the route.
- **Idx** Specifies the index of the network interface over which packets matching the route's address prefix are reachable. To display a list of interfaces and their indices, use the *netsh interface ipv6 show interface* command.
- **Gateway/Interface Name** For directly attached network routes, specifies the name of the interface; for remote network routes, specifies the next-hop address of the route.

NOTE **For more information about IPv6 routing and routing tables, see The Cable Guy article titled "Understanding the IPv6 Routing Table" at *http://technet.microsoft.com/en-ca/library/bb878115.aspx*.**

Understanding ICMPv6 Messages

Internet Control Message Protocol (ICMP) for IPv4 (ICMPv4) is used in IPv4 networks to allow nodes to send and respond to error messages and informational messages. For example, when a source node uses the *ping* command to send ICMP Echo Request messages (ICMP type 8 messages) to a destination node, the destination node can respond with ICMP Echo messages (ICMP type 0 messages) indicating its presence on the network.

On IPv6 networks, ICMP for IPv6 (ICMPv6) fulfills the same functions as ICMPv4 on IPv4 networks—namely, to provide a mechanism for exchanging error messages and informational messages. ICMPv6 also provides informational messages for the following:

- **Neighbor Discovery (ND)** The process by which hosts and routers discover each other on the network so that they can communicate at the data-link layer. (ND serves the same purpose as Address Resolution Protocol [ARP] does in IPv4 networks.)
- **Multicast Listener Discovery (MLD)** The process by which membership in multicast groups is determined and maintained.

NOTE **For more information about ND, see the next section titled "Understanding Neighbor Discovery." For more information about ICMPv6 message types and header formats and about MLD, see the white paper, "Introduction to IP Version 6," at *http://www.microsoft.com/downloads/details.aspx?FamilyID=CBC0B8A3-B6A4-4952-BBE6-D976624C257C&displaylang=en*.**

Understanding Neighbor Discovery

ND is the process by which nodes on an IPv6 network can communicate with each other by exchanging frames at the data-link layer. ND performs the following functions on an IPv6 network:

- Enables IPv6 nodes (IPv6 hosts and IPv6 routers) to resolve the link-layer address of a neighboring node (a node on the same physical or logical link)
- Enables IPv6 nodes to determine when the link-layer address of a neighboring node has changed
- Enables IPv6 nodes to determine whether neighboring nodes are still reachable
- Enables IPv6 routers to advertise their presence, on-link prefixes, and host configuration settings

- Enables IPv6 routers to redirect hosts to more optimal routers for a specific destination
- Enables IPv6 hosts to discover addresses, address prefixes, and other configuration settings
- Enables IPv6 hosts to discover routers attached to the local link

To understand how ND works, it helps to first compare it with the similar processes used in IPv4. In IPv4, you use three separate mechanisms to manage node-to-node communication:

- **Address Resolution Protocol** A data link-layer protocol that resolves IPv4 addresses assigned to interfaces to their corresponding MAC-layer addresses. This enables network adapters to receive frames addressed to them and send response frames to their source. For example, before a host can send a packet to a destination host whose IPv4 address is 172.16.25.3, the sending host first needs to use ARP to resolve this destination address (if the host is on the same LAN) or the IP address of the local gateway (if the host is on a different LAN) to its corresponding 48-bit MAC address (such as 00-13-20-08-A0-D1).
- **ICMPv4 router discovery** These ICMPv4 messages enable routers to advertise their presence on IPv4 networks and enable hosts to discover the presence of these routers. When router discovery is enabled on a router, the router periodically sends router advertisements to the all-hosts multicast address (224.0.0.1) to indicate to hosts on the network that the router is available. When router discovery is enabled on hosts, the hosts can send router solicitations to the all-routers multicast address (224.0.0.2) to obtain the address of the router and assign this address as the host's default gateway.
- **ICMPv4 Redirect** Routers use these ICMPv4 messages to inform hosts of more optimal routers to use for specific destinations. ICMPv4 Redirect messages are needed because hosts typically cannot determine the best router on their subnet to send remote traffic for a given destination.

On IPv4 networks, these three mechanisms enable nodes on a network segment to communicate on a link. On IPv6 networks, these three mechanisms are replaced by the five ICMPv6 message types shown in Table 28-5.

NOTE **The solicited-node multicast address, which is used as the destination address for ICMPv4 Neighbor Solicitation messages (ICMPv6 type 135 messages) when address resolution is being performed, is a special type of multicast address composed of the prefix FF02::1:FF00:0/104 followed by the last 24 bits of the IPv6 address that is being resolved. IPv6 nodes listen on their solicited-node multicast addresses. The advantage of using this multicast address for address resolution in IPv6 is that typically only the targeted host is disturbed on the local link. By contrast, the ARP messages used in IPv4 for address resolution queries are sent to the MAC-layer broadcast address, which disturbs all hosts on the local segment.**

TABLE 28-5 ICMPv6 Message Types Used for ND

MESSAGE TYPE	ICMPV6 TYPE	DESCRIPTION
Router Solicitation	133	Sent by IPv6 hosts to the link-local scope all-routers multicast address (FF02::2) to discover IPv6 routers present on the local link.
Router Advertisement	134	Sent periodically by IPv6 routers to the link-local scope all-nodes multicast address (FF02::1), or sent to the unicast address of a host in response to receiving a Router Solicitation message from that host. (Windows Vista and later versions use multicast for optimization.) Router Advertisement messages provide hosts with the information needed to determine link prefixes, link maximum transmission unit (MTU), whether to use DHCPv6 for address autoconfiguration, and lifetime for autoconfigured addresses.
Neighbor Solicitation	135	Sent by IPv6 nodes to the solicited-node multicast address of a host to discover the link-layer address of an IPv6 node, or sent to the unicast address of the host to verify the reachability of the host.
Neighbor Advertisement	136	Sent by an IPv6 node to the unicast address of a host in response to receiving a Neighbor Solicitation message from the host, or sent to the link-local scope all-nodes multicast address (FF02::1) to inform neighboring nodes of changes to the host's link-layer addresses.
Redirect	137	Sent by an IPv6 router to the unicast address of a host to inform the host of a more optimal first-hop address for a specific destination.

Understanding Address Autoconfiguration

On IPv4 networks, addresses can be assigned to hosts in three ways:

- Manually, using static address assignment
- Automatically, using Dynamic Host Configuration Protocol (DHCP) if a DHCP server is present on the subnet (or a DHCP relay agent is configured on the subnet)
- Automatically, using APIPA, which randomly assigns the host an address from the range 169.254.0.0 to 169.254.255.255 with subnet mask 255.255.0.0

On IPv6 networks, static addresses are generally assigned only to routers and (sometimes) servers, but hardly ever to client computers. Instead, IPv6 addresses are almost always

assigned automatically using a process called *address autoconfiguration*. Address autoconfiguration can work in three ways: stateless, stateful, or both. Stateless address autoconfiguration is based on the receipt of ICMPv6 Router Advertisement messages. Stateful address autoconfiguration, on the other hand, uses DHCP for IPv6 (DHCPv6) to obtain address information and other configuration settings from a DHCPv6 server.

> ***NOTE*** **The DHCP Server service of Windows Server 2008 supports DHCPv6. The DHCP Server service of Windows Server 2003 does not support DHCPv6.**

All IPv6 nodes (hosts and routers) automatically assign themselves link-local addresses (addresses having the address prefix FE80::/64); this is done for every interface (both physical and logical) on the node. (6to4 interfaces are an exception—they might not have link-local addresses automatically assigned.) These autoconfigured link-local addresses can be used only to reach neighboring nodes (nodes on the same link). When specifying one of these addresses as a destination address, you might need to specify the zone ID for the destination. In addition, link-local addresses are never registered in DNS servers.

> ***NOTE*** **Manual assignment of IPv6 addresses is generally needed only for IPv6 routers and for some servers. You can configure a computer running Windows 7 with multiple interfaces to be used as a router. For more information on configuring IPv6 routers, see the Cable Guy article titled "Manual Configuration for IPv6" at *http://technet.microsoft.com/en-us/library/bb878102.aspx*. For a description of the IPv6 routing table, see the Cable Guy article titled "Understanding the IPv6 Routing Table" at *http://technet.microsoft.com/en-us/library/bb878115.aspx*.**

An autoconfigured IPv6 address can be in one or more of the states shown in Table 28-6.

TABLE 28-6 Possible States for an Autoconfigured IPv6 Address

STATE	DESCRIPTION
Tentative	The uniqueness of the address is still being verified using duplicate address detection.
Valid	The address is unique and can now send and receive unicast IPv6 traffic until the Valid Lifetime expires.
Preferred	The address can be used for unicast traffic until the Preferred Lifetime expires.
Deprecated	The address can still be used for unicast traffic during existing communication sessions, but its use is discouraged for new communication sessions.
Invalid	The Valid Lifetime for the address has expired and it can no longer be used for unicast traffic.

NOTE **The Valid and Preferred Lifetime for stateless autoconfigured IPv6 addresses is included in the Router Solicitation message.**

For detailed descriptions of how address autoconfiguration, address resolution, router discovery, redirect, duplicate address detection, and neighbor unreachability detection processes are performed, see the white paper, "Introduction to IP Version 6," at *http://www.microsoft.com/downloads/details.aspx?FamilyID=CBC0B8A3-B6A4-4952-BBE6-D976624C257C&displaylang=en*.

NOTE **To display the state for each autoconfigured IPv6 address on a Windows 7 computer, open a command prompt and type netsh interface ipv6 show addresses at a command prompt.**

Understanding Name Resolution

The Domain Name System (DNS) is fundamental to how name resolution works on both IPv4 and IPv6 networks. On an IPv4 network, host (A) records are used by name servers (DNS servers) to resolve fully qualified domain names (FQDNs) like *server1.contoso.com* into their associated IP addresses in response to name lookups (name queries) from DNS clients. In addition, reverse lookups—in which IP addresses are resolved into FQDNs—are supported by using pointer (PTR) records in the *in-addr.arpa* domain.

Name resolution works fundamentally the same way with IPv6, with the following differences:

- Host records for IPv6 hosts are AAAA ("quad-A") records, not A records.
- The domain used for reverse lookups of IPv6 addresses is *ip6.arpa,* not *in-addr.arpa*.

NOTE **The enhancements to DNS that make IPv6 support possible are described in the draft standard RFC 3596 at *http://www.ietf.org/rfc/rfc3596.txt*.**

Understanding Name Queries

Because the dual-layer TCP/IP stack in Windows 7 means that both IPv4 and IPv6 are enabled by default, DNS name lookups by clients running Windows 7 can involve the use of both A and AAAA records. (This is true only if your name servers support IPv6, which is the case with the DNS Server role for Windows Server 2008 and Windows Server 2003.) By default, the DNS client in Windows 7 uses the following procedure when performing a name lookup using a particular interface:

1. The client computer checks to see whether it has a non–link-local IPv6 address assigned to the interface. If it has no non–link-local addresses assigned, the client sends a single name lookup to the name server to query for A records and does not query for AAAA records. If the only non–link-local address assigned to the interface is a Teredo address, the client again does not query for AAAA records. (The Teredo client in Windows Vista and later versions is explicitly built not to automatically perform AAAA lookups or register with DNS to prevent overloading of DNS servers.)
2. If the client computer has a non–link-local address assigned to the interface, the client sends a name lookup to query for A records.
   - If the client then receives a response to its query (not an error message), it follows with a second lookup to query for AAAA records.
   - If the client receives no response or receives any error message (except for Name Not Found), it does not send a second lookup to query for AAAA records.

NOTE **Because an interface on an IPv6 host typically has multiple IPv6 addresses, the process by which source and address selection works during a name query is more complex than when DNS names are resolved by IPv4 hosts. For a detailed description of how source and address selection works for IPv6 hosts, see the Cable Guy article titled "Source and Destination Address Selection for IPv6" at *http://technet.microsoft.com/en-us/library/bb877985.aspx*. For additional information on DNS behavior in Windows 7 and Windows Vista, see "Domain Name System Client Behavior in Windows Vista" at *http://technet.microsoft.com/en-us/library/bb727035.aspx*. For information about the different types of IPv6 addresses usually assigned to an interface, see the section titled "Configuring and Troubleshooting IPv6 in Windows 7" later in this chapter.**

NOTE **Issues have arisen with poorly configured DNS name servers on the Internet. These issues, which are described in RFC 4074 (*http://www.ietf.org/rfc/rfc4074.txt*), do not cause problems on Windows Vista or later versions because Microsoft has altered the DNS client behavior specifically to compensate for them. However, administrators of DNS servers should make sure these issues are fixed, because they can cause problems with DNS name resolution for most IPv6 networking stacks, including stacks found in earlier Windows platforms such as Windows XP.**

Understanding Name Registration

DNS servers running Windows Server 2003 can dynamically register both A and AAAA records for clients running Windows 7. Dynamic registration of DNS records simplifies the job of maintaining name resolution on networks running the Active Directory Directory Service.

When a client running Windows 7 starts up on a network, the DNS Client service tries to register the following records for the client:

- A records for all IPv4 addresses assigned to all interfaces configured with the address of a DNS server
- AAAA records for all IPv6 addresses assigned to all interfaces configured with the address of a DNS server
- PTR records for all IPv4 addresses assigned to all interfaces configured with the address of a DNS server

NOTE **AAAA records are not registered for link-local IPv6 addresses that have been assigned to interfaces using address autoconfiguration.**

PTR Records and IPv6

Clients running Windows 7 do not try to register PTR records for IPv6 addresses assigned to interfaces on the computer. If you want to enable clients to perform reverse lookups for Windows 7 computers using IPv6, you must manually create a reverse lookup zone for the ip6.arpa domain on your DNS servers and then manually add PTR records to this zone. For detailed steps on how to do this, see "IPv6 for Microsoft Windows: Frequently Asked Questions" at *http://www.microsoft.com/technet/network/ipv6/ipv6faq.mspx*.

However, PTR records for reverse lookups using IPv6 are not often used, because the namespace for reverse queries is formed by using each hexadecimal digit in the colon-hexadecimal representation of an IPv6 address as a separate level in the reverse domain hierarchy. For example, the PTR record associated with the IPv6 address 2001:DB8::D3:00FF:FE28:9C5A, whose full representation is 2001:0DB8:0000:0000:00D3:00FF:FE28:9C5A, would be expressed as A.5.C.9.8.2.E.F.F.F.0.0.3.D.0.0.0.0.0.0.0.0.0.0.0.8.B.D.0.1.0.0.2.IP6.ARPA. The performance cost of resolving such a representation is generally too high for most DNS server implementations.

By default, DNS servers running Windows Server 2003 do not listen for DNS traffic sent over IPv6. To enable these DNS servers to listen for IPv6 name registrations and name lookups, you must first configure the servers using the **dnscmd /config /EnableIPv6 1** command. By default, DNS servers running Windows Server 2008 listen for DNS traffic sent over IPv6. You must then configure each client running Windows 7 with the unicast IPv6 addresses of your DNS servers using DHCPv6, the properties of IPv6 (TCP/IPv6) in the Network Connections folder, or the *netsh interface ipv6 add dns interface=NameOrIndex address=IPv6Address index=*PreferenceLevel command where *PreferenceLevel* specifies the index for the specified

DNS server address. (DHCP servers running Windows Server 2003 do not support stateful address assignment using DHCPv6.)

> ***NOTE*** **For more information on enabling Windows Server 2003 DNS server support for IPv6, see Chapter 9, "Windows Support for DNS," in the online book *TCP/IP Fundamentals for Microsoft Windows,* which you can download from *http://www.microsoft.com/downloads/details.aspx?FamilyID=c76296fd-61c9-4079-a0bb-582bca4a846f &displaylang=en*. For further details on the DNS name query and registration behavior in Windows 7 and Windows Vista, see the article titled "Domain Name System Client Behavior in Windows Vista" on Microsoft TechNet at *http://technet.microsoft.com/en-us/library/bb727035.aspx*.**

IPv6 Enhancements in Windows 7

The TCP/IP networking stack in the Windows XP and Windows Server 2003 platforms had a dual-stack architecture that used separate network and framing layers for IPv4 and IPv6 based on separate drivers: Tcpip.sys and Tcpip6.sys. Only the transport and framing layers for IPv4 were installed by default, and adding support for IPv6 involved installing an additional IPv6 protocol feature through the Network Connections folder.

By contrast, in Windows 7, Windows Vista, and Windows Server 2008, the TCP/IP stack has been completely redesigned and now uses a dual–IP-layer architecture in which both IPv4 and IPv6 share common transport and framing layers. In addition, IPv6 is installed and enabled by default in these new platforms to provide out-of-the-box support for new features such as the Windows Meeting Space application, which uses only IPv6. Finally, the dual IP layer architecture means that all of the performance enhancements of the Next Generation TCP/IP stack that apply to IPv4 also apply to IPv6. These performance enhancements include Compound TCP, Receive Window Auto-Tuning, and other enhancements that can dramatically improve performance in high-latency, high-delay, and high-loss networking environments.

> ***NOTE*** **For more information about the performance enhancements in the Next Generation TCP/IP stack, see Chapter 25, "Configuring Windows Networking."**

Summary of IPv6 Enhancements in Windows 7

Windows 7 builds on the many IPv6 enhancements introduced earlier in Windows Vista and Windows Server 2008. These earlier enhancements include the following:

- **Dual–IP-layer architecture** A new TCP/IP stack architecture that uses the same transport and framing layers for both IPv4 and IPv6.
- **Enabled by default** Both IPv4 and IPv6 are installed and enabled by default, with the stack giving preference to IPv6 when appropriate without impairing the perfor-

mance of IPv4 communications on the network. For example, if a DNS name query returns both an IPv4 and IPv6 address for a host, the client will try to use IPv6 first for communicating with the host. This preference also results in better network performance for IPv6-enabled applications.

- **User interface configuration support** In addition to being able to configure IPv6 settings from the command line using the *netsh interface ipv6* command context, you can also configure them in Windows 7 using the user interface. For more information, see the section titled "Configuring IPv6 in Windows 7 Using the User Interface" later in this chapter.
- **Full IPsec support** IPv6 support in previous versions of Windows offered only limited support for IPsec protection of network traffic. In Windows 7 and Windows Vista, however, IPsec support for IPv6 is the same as for IPv4, and you can configure IPsec connection security rules for IPv6 in the same way as IPv4 by using the Windows Firewall With Advanced Security console. For more information on configuring IPsec in Windows 7, see Chapter 26, "Configuring Windows Firewall and IPsec."
- **LLMNR support** The implementation of IPv6 in Windows 7 and Windows Vista supports Link-Local Multicast Name Resolution (LLMNR), a mechanism that enables IPv6 nodes on a single subnet to resolve each other's names in the absence of a DNS server. LLMNR works by having nodes send multicast DNS name queries instead of unicast queries. Computers running Windows 7 and Windows Vista listen by default for multicast LLMNR traffic, which eliminates the need to perform local subnet name resolution using NetBIOS over TCP/IP when no DNS server is available. LLMNR is defined in RFC 4795.
- **MLDv2 support** The implementation of IPv6 in Windows 7 and Windows Vista supports MLD version 2 (MLDv2), a mechanism described in RFC 3810 that enables IPv6 hosts to register interest in source-specific multicast traffic with local multicast routers by specifying an include list (to indicate specific source addresses of interest) or an exclude list (to exclude unwanted source addresses).
- **DHCPv6 support** The DHCP Client service in Windows 7 and Windows Vista supports DHCPv6 as defined in RFCs 3736 and 4361. This means that computers running Windows 7 and Windows Vista can perform both stateful and stateless DHCPv6 configuration on a native IPv6 network.
- **IPv6CP support** The built-in remote access client functionality in Windows 7 and Windows Vista supports IPv6 Control Protocol (IPv6CP) (RFC 5072) to configure IPv6 nodes on a PPP link. This means that native IPv6 traffic can be sent over PPP-based network connections, such as dial-up connections or broadband PPP over Ethernet (PPPoE) connections, to an ISP. IPv6CP also supports Layer 2 Tunneling Protocol (L2TP), and for Windows Vista with Service Pack 1 (SP1) or later, Secure Socket Tunneling Protocol (SSTP)–based virtual private network (VPN) connections. For more information on IPv6CP support in Windows 7, see Chapter 27, "Connecting Remote Users and Networks."

- **Random interface IDs** By default, Windows 7 and Windows Vista generate random interface IDs for non-temporary autoconfigured IPv6 addresses, including both public addresses (global addresses registered in DNS) and link-local addresses. For more information, see the section titled "Disabling Random Interface IDs" later in this chapter.
- **Literal IPv6 addresses in URLs** Windows 7 and Windows Vista support RFC 2732–compliant literal IPv6 addresses in URLs by using the WinINet application programming interface (API) support in Windows Internet Explorer 8.0. This can be a useful feature for troubleshooting Internet connectivity with IPv6-enabled Web servers.
- **New Teredo behavior** The Teredo client in Windows 7 and Windows Vista remains dormant (inactive) until it spins up (is activated by) an IPv6-enabled application that tries to use Teredo. In Windows 7 and Windows Vista, three things can bring up Teredo: an application trying to communicate using a Teredo address (the outbound instantiated scenario), a listening application that has the Edge Traversal rule enabled in Windows Firewall (any IPv6-enabled applications that need to use Teredo can easily do so by setting the *Edge Traversal* flag using the Windows Firewall APIs), and the *NotifyStableUnicastIpAddressTable* IP Helper API. For more information about Windows Firewall rules, see Chapter 26.

In addition to these earlier enhancements, Windows 7 and Windows Server 2008 R2 introduce the following new IPv6 improvements:

- **IP-HTTPS** This stands for Internet Protocol over Hypertext Transfer Protocol Secure (IP over HTTPS), a new protocol that enables hosts located behind a proxy or firewall to establish connectivity by tunneling IP traffic inside an HTTPS tunnel. HTTPS is used instead of HTTP so that proxy servers will be prevented from looking inside the data stream and terminating the connection if traffic seems anomalous. Note that HTTPS does not provide data security—you must use IPsec to provide data security for an IP-HTTPS connection.

  In the Windows 7 implementation of DirectAccess described in the following More Info box, IT-HTTPS is used whenever a firewall or proxy server blocks a client computer from using 6to4 or Teredo to establish an IPv6-over-IPv4 tunnel with an IPv6-enabled DirectAccess server on the corporate intranet.

> ***MORE INFO*** **For more information about IP-HTTPS, see the article, "IP over HTTPS (IP-HTTPS) Tunneling Protocol Specification," on MSDN at *http://msdn.microsoft.com/en-us/library/dd358571.aspx*.**

- **DirectAccess** This is a new feature of Windows 7 and Windows Server 2008 R2 that provides users with the experience of being seamlessly connected to the corporate network whenever they have Internet access. Using DirectAccess, remote users who attempt to access corporate intranet resources, such as e-mail servers, shared folders, or intranet Web sites, can access these resources without the need to connect to a VPN.

By providing users with the same connectivity experience both inside and outside the office, DirectAccess can increase the productivity of your mobile users. DirectAccess also enables administrators to keep the computers of mobile uses in a managed state even when they are off-site by allowing Group Policy changes to be propagated over the Internet.

DirectAccess is implemented as a client/server architecture in which remote IPv6-enabled client computers communicate with IPv6-enabled servers located on the corporate network. DirectAccess can work over existing IPv4 networks, such as the public IPv4 Internet, by using IPv4/IPv6 transition technologies such as 6to4, Teredo, and ISATAP. DirectAccess also supports native IPv6 connectivity for clients that have been assigned native IPv6 addresses.

DirectAccess uses IPsec tunneling to provide security for authentication and resource access. DirectAccess can be implemented in different ways ranging from providing client computers with secure access to intranet resources via an IPv6-enabled IPsec gateway to providing them with secure end-to-end connectivity with each IPv6-enabled application server located on the intranet. DirectAccess requires the use of IPv6 so that client computers can have globally routable addresses.

MORE INFO **For more information about DirectAccess, see Chapter 27 in this resource kit. Also see the article, "DirectAccess Technical Overview for Windows 7 and Windows Server 2008 R2," at *http://technet.microsoft.com/en-us/library/dd637827.aspx*.**

HOW IT WORKS

Teredo Behavior in Windows 7 and Windows Vista

Michael Surkan
Program Manager for TCP and IPv6

Teredo is default-enabled but inactive in both workgroup and domain scenarios. Teredo becomes active in two main scenarios:

- **An application tries to communicate with a Teredo address (for example, by using a URL with a Teredo address in a Web browser). This is outbound-initiated traffic, and Teredo will go dormant again after 60 minutes of inactivity. The host firewall will allow only incoming Teredo traffic corresponding to the specific outbound request, ensuring that system security isn't compromised. This is really no different than the way in which any outbound-initiated traffic works with the host firewall with IPv4. (In other words, all outbound traffic is allowed by default, and a state table allows responses that match the outgoing requests.)**

- **An application or service is authorized to use Teredo with the advanced Windows Firewall *Edge Traversal* flag. If an application has the Edge Traversal option, it is allowed to receive any incoming traffic over Teredo from any source (such as unsolicited traffic). Windows Meeting Space and Remote Assistance automatically set this flag for themselves, but users can do it manually for other Windows services if they prefer, such as with a Web service.**

Configuring and Troubleshooting IPv6 in Windows 7

Although IPv6 is designed to allow IPv6-enabled nodes, such as computers running Windows 7, to automatically configure their interfaces with link-local addresses, these autoconfigured addresses are not registered in DNS servers and can be used only for communicating with other nodes on the local link. Alternatively, by using a DHCPv6 server, you can automatically assign global, site-local, or unique local IPv6 addresses to IPv6-enabled interfaces of link-attached nodes. This is the preferred scenario for end-to-end IPv6 connectivity in enterprises that have a native IPv6-only network infrastructure.

However, you can also use two methods to configure IPv6 settings manually on computers running Windows 7:

- Using the new IPv6 graphical user interface
- Using the *netsh interface ipv6* command context

In addition, it is important to understand the different kinds of IPv6 addresses assigned to computers running Windows 7 so that you can troubleshoot IPv6 connectivity when problems arise.

Displaying IPv6 Address Settings

To display the IPv4 and IPv6 address configuration of the local computer, open a command prompt window and type **ipconfig /all**. The following is an example of the information displayed by running this command on a managed (domain-joined) computer running Windows 7 with a single LAN network adapter, no IPv6 routers on the attached subnet, and no other configured network connections.

```
Windows IP Configuration

   Host Name . . . . . . . . . . . . : KBERG-PC
   Primary Dns Suffix  . . . . . . . : contoso.com
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No
   DNS Suffix Search List. . . . . . : contoso.com
```

```
Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . : contoso.com
   Description . . . . . . . . . . . : Broadcom NetXtreme 57xx Gigabit Controller
   Physical Address. . . . . . . . . : 00-13-D4-C2-50-F5
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::3530:6107:45a2:a92c%8(Preferred)
   IPv4 Address. . . . . . . . . . . : 172.16.11.13(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : Tuesday, March 17, 2009 9:01:24 AM
   Lease Expires . . . . . . . . . . : Wednesday, March 25, 2009 9:01:29 AM
   Default Gateway . . . . . . . . . : 172.16.11.1
   DHCP Server . . . . . . . . . . . : 172.16.11.32
   DHCPv6 IAID . . . . . . . . . . . : 201331668
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-11-50-8C-A7-00-17-31-C5-D2-8E
   DNS Servers . . . . . . . . . . . : 172.16.11.32
   NetBIOS over Tcpip. . . . . . . . : Enabled

Tunnel adapter isatap.contoso.com:

   Media State . . . . . . . . . . . : Media Disconnected
   Connection-specific DNS Suffix  . : contoso.com
   Description . . . . . . . . . . . : Microsoft ISATAP Adapter
   Physical Address. . . . . . . . . : 00-00-00-00-00-00-00-E0
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
```

The preceding command output displays two interfaces on this computer:

- Local Area Connection (the installed network adapter)
- ISATAP tunneling interface

The Local Area Connection interface is an Ethernet network adapter and has both an IPv4 address (172.16.11.13) assigned by DHCP and a link-local IPv6 address (fe80::3530:6107:45a2:a92c) that has been automatically assigned using IPv6 address autoconfiguration. (You can recognize the link-local address by its address prefix, FE80::/64.)

The *%8* appended to this address is the zone ID (or scope ID) that indicates the connected portion of the network on which the computer resides. This zone ID corresponds with the interface index for the Local Area Connection interface. To view a list of interface indexes on a computer, type **netsh interface ipv6 show interface** at a command prompt. For the example computer, the output of this command is the following code.

```
Idx  Met   MTU        State        Name
---  ---  -----       -----------  -------------------
  1   50  4294967295  connected    Loopback Pseudo-Interface 1
  9   25  1280        connected    isatap.contoso.com
  8   20  1500        connected    Local Area Connection
```

Here the Idx column indicates the interface index. The zone ID might be needed when testing network connectivity with this computer from other computers using the *ping* and *tracert* commands. See the section titled "Troubleshooting IPv6 Connectivity" later in this chapter for more information.

Returning to the output of the *ipconfig /all* command, the state of the link-local address assigned to the LAN connection is Preferred, which indicates a valid IPv6 address that you can use to send and receive unicast IPv6 traffic.

The media state of the ISATAP tunneling interface isatap.contoso.com is Media Disconnected. You can enable the ISATAP tunneling interface by opening an elevated command prompt and typing the **netsh interface isatap set state enabled** command. After you have enabled the ISATAP interface, the ISATAP portion of the *ipconfig /all* output will look something like this.

```
Tunnel adapter isatap.contoso.com:

   Connection-specific DNS Suffix  . : contoso.com
   Description . . . . . . . . . . . : Microsoft ISATAP Adapter
   Physical Address. . . . . . . . . : 00-00-00-00-00-00-00-E0
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::5efe:172.16.11.13%9(Preferred)
   Default Gateway . . . . . . . . . :
   DNS Servers . . . . . . . . . . . : 172.16.11.32
   NetBIOS over Tcpip. . . . . . . . : Disabled
```

NOTE **If the computer is unmanaged (not domain-joined), the ISATAP adapter will be enabled automatically and will be displayed with a GUID, for example isatap.{9D607D7D-0703-4E67-82ED-9A8206377C5C}.**

The above ISATAP adapter has an autoconfigured link-local address of fe80::5efe:172.16.11.13. The format for an ISATAP address is:

- The first 64 bits are a unicast prefix that can be a link-local, global, or unique local unicast IPv6 address prefix. This example uses the link-local address prefix because no ISATAP router is present on the network. This means that the resulting ISATAP address can be used only for communicating with other ISATAP hosts on the IPv4 network, and this ISATAP address is not registered in DNS servers.
- The next 32 bits are either 0:5EFE (for a private IPv4 address) or 200:5EFE (for a public IPv4 address) in an ISATAP address. (RFC 4214 also allows 100:5EFE and 300:5EFE in this portion of an ISATAP address.)
- The final 32 bits consist of the 32-bit IPv4 address of the host in dotted-decimal form (172.16.11.13 in this example).

MORE INFO **For more information on ISATAP addressing, see the white paper, "IPv6 Transition Technologies," at *http://www.microsoft.com/downloads/details.aspx?FamilyID=afe56282-2903-40f3-a5ba-a87bf92c096d &displaylang=en,* and the white paper, "Intra-site Automatic Tunnel Addressing Protocol Deployment Guide," at *http://www.microsoft.com/downloads/details.aspx?FamilyID=0f3a8868-e337-43d1-b271-b8c8702344cd &displaylang=en*. Also see the section titled "Understanding ISATAP" later in this chapter.**

The output of the *ipconfig /all* and *netsh interface ipv6 show interface* commands does not show a Teredo adapter on the computer because the computer is managed (domain joined). On an unmanaged computer, the Teredo adapter is enabled (in online mode) by default and the *ipconfig /all* output will look something like this.

```
Tunnel adapter Teredo Tunneling Pseudo-Interface:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Microsoft Teredo Tunneling Adapter
   Physical Address. . . . . . . . . : 02-00-54-55-4E-01
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   IPv6 Address. . . . . . . . . . . : 2001:0:4136:e37c:4e8:3426:7c94:fffe(Preferred)
   Link-local IPv6 Address . . . . . : fe80::4e8:3426:53ef:f4f2%10(Preferred)
   Default Gateway . . . . . . . . . : ::
   NetBIOS over Tcpip. . . . . . . . : Disabled
```

The above Teredo tunneling pseudo-interface displays the IPv6 address of the Teredo client as 2001:0:4136:e37c:4e8:3426:53ef:f4f2. The format for a Teredo client address is:

- The first 32 bits are always the Teredo prefix, which is 2001::/32.
- The next 32 bits contain the public IPv4 address of the Teredo server that helped in the configuration of this Teredo address (here 4136:E37C hexadecimal, which converts to 65.54.227.124 in dotted-decimal format). By default, the Teredo client in Windows 7, Windows Vista, and Windows Server 2008 automatically tries to determine the IPv4 addresses of Teredo servers by resolving the name *teredo.ipv6.microsoft.com*.
- The next 16 bits are reserved for various Teredo flags.
- The next 16 bits contain an obscured version of the external UDP port number that corresponds to all Teredo traffic for this Teredo client. (The external UDP port number is obscured, XORing it with 0xFFFF, and, in this example, is 0x3426 XOR 0xFFFF = 0xCBD9 or decimal 52185, meaning UDP port 52185.)
- The final 32 bits contain an obscured version of the external IPv4 address that corresponds to all Teredo traffic for this Teredo client. (The external IPv4 address is obscured, XORing it with 0xFFFF FFFF, and, in this example, is 0x7C94 FFFE XOR 0xFFFF FFFF = 0x836B 0001 or dotted-decimal 131.107.0.1.)

NOTE **IANA has allocated the IPv6 address prefix 2001::/32 for Teredo as of January 2006. (See RFC 4830 at *http://www.rfc-editor.org/rfc/rfc4380.txt* for details.) Windows XP–based clients originally used the 3FFE:831F::/32 Teredo prefix. Windows XP–based clients with the Microsoft Security Bulletin MS06-064 at *http://www.microsoft.com/technet/security/Bulletin/MS06-064.mspx* now use the 2001::/32 prefix.**

Another way to display the IPv6 settings on a computer running Windows 7 is to type the **netsh interface ipv6 show address** command. The results for the computer in the preceding example are as follows.

```
Interface 1: Loopback Pseudo-Interface

Addr Type  DAD State   Valid Life Pref. Life Address
---------  ----------- ---------- ---------- ------------------------
Other      Preferred     infinite   infinite ::1

Interface 9: isatap.{9D607D7D-0703-4E67-82ED-9A8206377C5C}

Addr Type  DAD State   Valid Life Pref. Life Address
---------  ----------- ---------- ---------- ------------------------
Other      Preferred     infinite   infinite fe80::5efe:172.16.11.13%9

Interface 10: Teredo Tunneling Pseudo-Interface

Addr Type  DAD State   Valid Life Pref. Life Address
---------  ----------- ---------- ---------- ------------------------
Public     Preferred     infinite   infinite 2001:0:4136:e37c:1071:3426:31d2:bfce
Other      Preferred     infinite   infinite fe80::1071:3426:31d2:bfce%10

Interface 8: Local Area Connection

Addr Type  DAD State   Valid Life Pref. Life Address
---------  ----------- ---------- ---------- ------------------------
Other      Preferred     infinite   infinite fe80::3530:6107:45a2:a92c%8
```

NOTE **An advantage of displaying IPv6 address settings using the *netsh interface ipv6 show address* command instead of *ipconfig* is that you can execute Netsh.exe commands remotely against a targeted computer by using the *–r RemoteComputerName* option.**

MORE INFO **For more information on how to use *ipconfig,* Netsh.exe, and other tools to display IPv6 configuration information, see the article, "Using Windows Tools to Obtain IPv6 Configuration Information," on Microsoft TechNet at *http://technet.microsoft.com/en-us/library/bb726952.aspx*.**

DIRECT FROM THE SOURCE

Explanation of Teredo States

Kalven Wu, Software Design Engineer in Test
Windows Core Networking

With netsh int teredo show state, you can see the current state of Teredo, which can be one of the following:

- **Offline state** In this state, something has failed and Teredo cannot be activated (cannot be in the Qualified state) to be used by applications. Teredo enters this state in three ways:
  - When the Administrator disables it via netsh int teredo set state disabled.
  - When Teredo detects that the computer is on a managed network (detects the presence of a domain controller on the network—see the section in this sidebar titled "Teredo in Enterprise Networks" for more information), it will go offline if its type is not set to "enterpriseclient".
  - When some internal mechanism has failed in Teredo, such as suddenly being unable to reach the Teredo server or being unable to resolve teredo.ipv6.microsoft.com. In only this case, Teredo will attempt to move into the Dormant state using an exponential back-off time-out as follows: wait 5 seconds, try again; wait 10 seconds, try again; wait 20 seconds, try again; and continue until it tries every 15 minutes.
- **Dormant state** This is the state when Teredo is "enabled but not active." IPv6 traffic cannot flow over Teredo, but applications can trigger to activate Teredo. No edge traversal will occur in this state. No traffic is sent to the Teredo servers.
- **Probe state** This is the transition state from Dormant to Qualified. In this state, Teredo will try to establish communication with the Teredo server. If this succeeds, Teredo moves to the Qualified state. If this fails, Teredo will go to the Offline state.
- **Qualified state** In this state, IPv6 traffic can flow into and out of the system over Teredo and possibly traverse the edge firewall/NAT.

Teredo in Enterprise Networks

Whether a computer is domain joined or in a workgroup doesn't matter to Teredo. Teredo looks only at the environment that the computer is in. If Teredo detects the presence of a domain controller, it will assume that the network is managed. In this case, Teredo will go offline and stay offline unless it was administratively set to "enterpriseclient" using the command *netsh interface teredo set state enterpriseclient*. Hence, Teredo will go to the Offline state on a workgroup computer that is connected to a network with a domain controller to avoid traversing the edge of

a corporate network. Conversely, if you take a domain-joined laptop home, Teredo will detect that it is no longer in a managed network and will go to the Dormant state.

Note that if you disable Teredo via the DisabledComponents registry key, it will override all the Teredo netsh settings.

Configuring IPv6 in Windows 7 Using the User Interface

To configure the IPv6 settings for a network connection in Windows 7 using the user interface, follow these steps:

1. In Control Panel, open Network And Sharing Center.
2. Click Manage Network Connections and then double-click the connection you want to configure.
3. Click Properties and respond to the User Account Control (UAC) prompt.
4. Select Internet Protocol Version 6 (TCP/IPv6) and click Properties to open the Internet Protocol Version 6 (TCP/IPv6) properties sheet (see Figure 28-1).

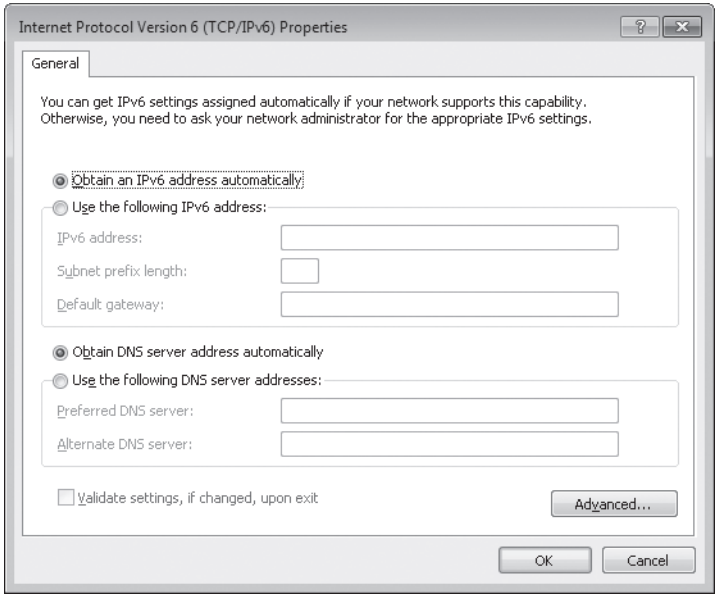


FIGURE 28-1 IPv6 properties of a network connection

5. Configure the IPv6 settings for the network connection as desired.
6. If you want, validate the new TCP/IP settings using the Windows Network Diagnostics Troubleshooter.

By default, the IPv6 settings for a network connection are configured as follows:

- **Obtain An IPv6 Address Automatically** This specifies that the physical or logical interface associated with this connection uses stateful or stateless address autoconfiguration to obtain its IPv6 address.
- **Obtain DNS Server Address Automatically** This specifies that the physical or logical interface associated with this connection uses stateful address autoconfiguration (DHCPv6) to obtain the IPv6 addresses of preferred and alternate DNS servers.

By selecting Use The Following IPv6 Address, you can manually configure the IPv6 address settings for a network connection by specifying the following:

- **IPv6 Address** Type the unicast IPv6 address you want to assign to the physical or logical interface associated with this connection in colon-hexadecimal form. If you need to assign additional unicast IPv6 addresses to the interface, click Advanced and then click the IP Settings tab.
- **Subnet Prefix Length** Type the subnet prefix length for the IPv6 address you assigned to the physical or logical interface associated with this connection. For unicast IPv6 addresses, the subnet prefix length should almost always be specified as 64.
- **Default Gateway** Type the unicast IPv6 address of the default gateway for the local IPv6 subnet in colon-hexadecimal form. If you need to specify additional default gateways, click Advanced and then click the IP Settings tab.

By selecting Use The Following DNS Server Addresses, you can manually specify IPv6 addresses for a preferred and an alternate DNS server to be used by your connection. If you need to specify additional alternate DNS servers, click Advanced and then click the DNS tab. The remaining settings on the DNS tab have similar functionality to those used for configuring IPv4 address settings.

> ***NOTE*** **The Advanced TCP/IP Settings dialog box does not have a WINS tab because IPv6 does not use NetBIOS for name resolution.**

Configuring IPv6 in Windows 7 Using Netsh

To configure the IPv6 settings for a network connection in Windows 7 using the Netsh.exe command, open a Command Prompt window with local administrator credentials and type the appropriate Netsh.exe command from the *netsh interface ipv6* context. Some examples of IPv6 configuration tasks that can be performed from this context include:

- To add the unicast IPv6 address 2001:DB8::8:800:20C4:0 to the interface named Local Area Connection as a persistent IPv6 address with infinite Valid and Preferred Lifetimes, type the following command.

```
netsh interface ipv6 add address "Local Area Connection" 2001:DB8::8:800:20C4:0
```

- To configure a default gateway with unicast IPv6 address 2001:DB8:0:2F3B:2AA:FF:FE28:9C5A for the interface named Local Area Connection, add a default route with this address specified as a next-hop address by typing the following command.

  ```
  netsh interface ipv6 add route ::/0 "Local Area Connection" 2001:DB8:0:2F3B:2AA:FF:FE28:9C5A
  ```

- To configure a DNS server with unicast IPv6 address 2001:DB8:0:1::1 as the second (alternate) DNS server on the list of DNS servers for the interface named Local Area Connection, type the following command.

  ```
  netsh interface ipv6 add dnsserver "Local Area Connection" 2001:DB8:0:1::1 index=2
  ```

For more information on using the *netsh interface ipv6* context, type **netsh interface ipv6 ?** at a command prompt.

Other IPv6 Configuration Tasks

The following section describes some additional IPv6 configuration tasks that network administrators may need to know how to perform with computers running Windows 7.

Enabling or Disabling IPv6

You cannot uninstall IPv6 in Windows 7, but you can disable IPv6 on a per-adapter basis. To do this, follow these steps:

1. In Control Panel, open Network And Sharing Center.
2. Click Manage Network Connections and then double-click the connection you want to configure.
3. Clear the check box labeled Internet Protocol Version 6 (TCP/IPv6), and then click OK (see Figure 28-2).

Note that if you disable IPv6 on all your network connections using the user interface method described in the preceding steps, IPv6 will still remain enabled on all tunnel interfaces and on the loopback interface.

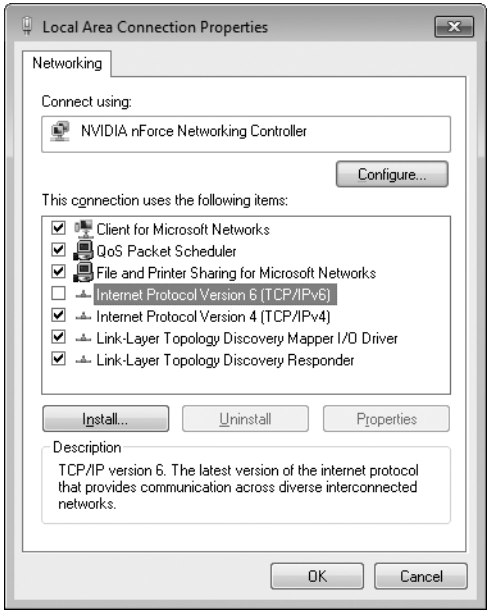


FIGURE 28-2 Disabling IPv6 for a network connection

As an alternative to using the user interface to disable IPv6 on a per-adapter basis, you can selectively disable certain features of IPv6 by creating and configuring the following DWORD registry value:

HKLM\SYSTEM\CurrentControlSet\Services\tcpip6\Parameters\DisabledComponents

Table 28-7 describes the flag values that control each IPv6 feature. By combining these flag values together into a bitmask, you can disable more than one feature at once. (By default, DisabledComponents has the value 0.)

TABLE 28-7 Bitmask Values for Disabling IPv6 Features in Windows 7

FLAG LOW-ORDER BIT	RESULT OF SETTING THIS BIT TO A VALUE OF 1
0	Disables all IPv6 tunnel interfaces, including ISATAP, 6to4, and Teredo tunnels
1	Disables all 6to4-based interfaces
2	Disables all ISATAP-based interfaces
3	Disables all Teredo-based interfaces
4	Disables IPv6 over all non-tunnel interfaces, including LAN and PPP interfaces
5	Modifies the default prefix policy table* to prefer IPv4 over IPv6 when attempting connections

**For more information concerning the IPv6 prefix policy table, see the Cable Guy article, "Source and Destination Address Selection for IPv6," at http://technet.microsoft.com/en-us/library/bb877985.aspx.*

For example, by setting the value of DisabledComponents to 0xFF, you can simultaneously disable IPv6 on all your network connections and tunnel interfaces. If you do this, IPv6 still remains enabled on the loopback interface, however.

> ***NOTE*** **For some examples of common flag combinations that can be used to enable or disable different aspects of IPv6 functionality in Windows 7 and Windows Vista, see the Cable Guy article, "Configuring IPv6 with Windows Vista," at *http://technet.microsoft.com/en-us/library/bb878057.aspx*.**

Depending on your scenario, there are other ways of effectively disabling IPv6 on computers running Windows 7, including the following:

- **Disable the IP Helper service** This service must be running for IPv6 transition technologies such as ISATAP, Teredo, and 6to4 to function on the computer. This service provides automatic IPv6 connectivity over an IPv4 network, and if the service is stopped, the computer will have only IPv6 connectivity if it is connected to a native IPv6 network. Therefore, if your network is not native IPv6, disabling this service on Windows 7 computers effectively disables IPv6 on them. You can use Group Policy to disable this service on targeted Windows 7 computers.
- **Use *netsh* to disable all IPv6 interfaces** For example, the following commands will disable all IPv6 transition technologies (Teredo, 6to4, and ISATAP).

  ```
  netsh interface teredo set state disabled

  netsh interface ipv6 6to4 set state state=disabled undoonstop=disabled

  netsh interface ipv6 isatap set state state=disabled
  ```

  You can include these commands in a script and send them inside a Microsoft System Center Configuration Manager (SCCM) package to disable transition technologies on targeted computers.
- **Configure Windows Firewall to block IPv6 traffic** You could block incoming and outgoing IPv6 protocol 41 (for ISATAP and 6to4) and UDP 3544 (for Teredo) traffic using the Windows Firewall, and you can use Group Policy to push this out to targeted computers. Businesses that implement perimeter firewalls may want to do this as a best practice for safeguarding their networks.

Disabling Random Interface IDs

You can disable the default behavior of generating random interface IDs for non-temporary autoconfigured public addresses (global addresses registered in DNS) and link-local addresses by using the following command.

```
netsh interface ipv6 set global randomizeidentifiers=disabled
```

To re-enable the generating of random interface IDs, use the following command.

```
netsh interface ipv6 set global randomizeidentifiers=enabled
```

> ***NOTE*** **Disabling random interface IDs causes link-local addresses to revert to using 48-bit MAC-layer (or 64-bit EUI) addresses for generating the interface ID portion of the address. In Windows, this happens immediately and does not require a reboot.**

Resetting IPv6 Configuration

To remove all user-configured IPv6 settings and restore the IPv6 configuration of a computer to its default state, type the following command.

```
netsh interface ipv6 reset
```

You must reboot the computer for this command to take effect.

Displaying Teredo Client Status

To verify the current state of the Teredo client on your computer, open a Command Prompt window using local administrator credentials, and then type the following command.

```
netsh interface teredo show state
```

For a computer running Windows 7 on which Teredo is currently inactive, the typical output for this command looks like this.

```
Teredo Parameters
---------------------------------------------
Type                    : default
Server Name             : teredo.ipv6.microsoft.com.
Client Refresh Interval : 30 seconds
Client Port             : unspecified
State                   : dormant
Client Type             : teredo client
Network                 : managed
NAT                     : none (global connectivity)
```

> ***NOTE*** **If your command output doesn't contain all the preceding information, you probably started your command prompt session using standard credentials instead of administrator credentials.**

If you now start an IPv6-enabled application that uses Teredo, such as Windows Meeting Space or Windows Remote Assistance, and then type the same *Netsh* command, the command output typically now looks like this.

```
Teredo Parameters
------------------------------------------
Type                    : default
Server Name             : teredo.ipv6.microsoft.com.
Client Refresh Interval : 30 seconds
Client Port             : unspecified
State                   : qualified
Client Type             : teredo client
Network                 : managed
NAT                     : restricted
```

Comparing these two command outputs shows that starting an application that uses Teredo changes the Teredo client state from Dormant (inactive) to Qualified (active).

> ***NOTE*** **The output of the *netsh interface teredo show state* command also tells you the type of NAT your computer is behind (if any). In the preceding example, the computer is behind a restricted NAT. Teredo works well behind restricted and cone NATs and can even work behind symmetric NATs, but communication between certain types of NATs doesn't work. If you plan to purchase a Small Office/Home Office (SOHO) router for broadband Internet connectivity, the best choice is a router that supports 6to4. For more information on how Teredo works and on the different types of NATs, see "Teredo Overview" at *http://technet.microsoft.com/en-us/network/cc917486.aspx*.**

Troubleshooting IPv6 Connectivity

The standard approach for troubleshooting TCP/IP network connectivity issues on IPv4 networks is to follow these steps:

1. Type **ipconfig /all** at a command prompt to verify the IPv4 configuration of the computer that is experiencing the problem.
2. If verifying the computer's IPv4 configuration doesn't resolve the issue, try using the *ping* command to test for network connectivity, beginning with the local computer and working outward until the cause of the problem is determined. Specifically, follow these steps in the order listed:
   a. Ping the IPv4 loopback address 127.0.0.1 to verify that TCP/IP is installed and configured properly on the computer.
   b. Ping the IPv4 address of the local computer.
   c. Ping the IPv4 address of the default gateway.
   d. Ping the IPv4 address of an IPv4 host on a remote subnet.

Other TCP/IP troubleshooting steps you can use on IPv4 networks include:

- Use the **route print** command to verify the configuration of the local computer's routing table.

- Use *tracert* to verify that intermediate routers are configured properly.
- Use the *pathping* command to identify packet loss over multihop paths.
- Clear the ARP cache by typing **netsh interface ip delete arpcache** at a command prompt.
- Verify the computer's DNS configuration, clear the DNS client resolver cache, and verify DNS name resolution.

> ***NOTE*** **For more information on how to systematically troubleshoot IPv4 connectivity problems, read Chapter 31, "Troubleshooting Network Issues."**

Troubleshooting IPv6 network connectivity issues requires many of the same tools you use when troubleshooting IPv4. However, you use some of these tools in a different way because of the nature of IPv6 addressing and the way IPv6 is implemented in Windows 7 and Windows Vista. The differences include:

- You might need to specify a zone ID when attempting to verify IPv6 network connectivity with a target host using the *ping* command. The syntax for using *ping* with IPv6 is **ping *IPv6Address%ZoneID***, where *ZoneID* is the zone ID (or scope ID) of the sending interface. For example, if the target host has the link-local unicast IPv6 address FE80::D3:00FF:FE28:9C5A and the sending interface has a zone ID of 12, to verify IPv6 connectivity with this host, you type **ping FE80::D3:00FF:FE28:9C5A%12** at a command prompt. To determine the zone ID for an interface, you can either use the *ipconfig /all* command or type **netsh interface ipv6 show interface** at a command prompt. Note that because the zone ID is locally defined, a sending host and a receiving host on the same link may have different zone IDs. (Global and unique local unicast IPv6 addresses do not need a zone ID.)
- You should view and clear the neighbor cache on your computer before attempting to use ping to verify IPv6 network connectivity. The neighbor cache contains recently resolved link-layer IPv6 addresses; you can view it by typing **netsh interface ipv6 show neighbors** and flush it by typing **netsh interface ipv6 delete neighbors** at an elevated command prompt.
- You should also view and clear the destination cache on your computer before attempting to verify IPv6 network connectivity using *ping*. The destination cache contains next-hop IPv6 addresses for destinations. You can view the cache by typing **netsh interface ipv6 show destinationcache**; you can flush it by typing **netsh interface ipv6 delete destinationcache** at an elevated command prompt.
- You should use the *–d* option when attempting to trace the route to a remote IPv6 host using *tracert* or the *–n* option when using *pathping*. These options prevent these commands from performing DNS reverse queries on every near-side router interface along the routing path. Using these options can help speed up the display of the routing path.

NOTE **For more help on troubleshooting IPv6 network connectivity issues, see the Cable Guy article, "Troubleshooting IPv6," at *http://technet.microsoft.com/en-us/library/bb878005.aspx*. See also Chapter 12, "Troubleshooting TCP/IP," in the online book *TCP/IP Fundamentals for Microsoft Windows,* which you can download from *http://www.microsoft.com/downloads/details.aspx?FamilyID=c76296fd-61c9-4079-a0bb-582bca4a846f&displaylang=en*.**

NOTE **Disabling IPv4 can also be a useful troubleshooting technique for developers who need to verify that their applications are IPv6-capable.**

Planning for IPv6 Migration

Migrating your existing IPv4-based network infrastructure to IPv6 requires an understanding of different IPv6 transition technologies that you can use to achieve your goal. Windows 7, Windows Vista, and Windows Server 2008 support three transition technologies in particular:

- **ISATAP** An address assignment and automatic tunneling technology defined in RFC 4214 that you can use to provide unicast IPv6 connectivity between IPv6/IPv4 hosts (hosts that support both IPv6 and IPv4) across an IPv4-based intranet (a private network whose infrastructure hardware, such as routers, supports only IPv4, not IPv6).
- **6to4** An address assignment and automatic tunneling technology defined in RFC 3056 that you can use to provide unicast IPv6 connectivity between IPv6/IPv4 hosts and sites across the IPv4-based public Internet. 6to4 enables you to assign global IPv6 addresses within your private network so that your hosts can reach locations on the IPv6 Internet without needing a direct connection to the IPv6 Internet or an IPv6 global address prefix obtained from an IPv6-supporting ISP. (Communication between a 6to4 site and a node on the IPv6 Internet requires the use of a 6to4 relay, however.)
- **Teredo** An address assignment and automatic tunneling technology defined in RFC 4380 that you can use to provide unicast IPv6 connectivity between IPv6/IPv4 hosts across the IPv4 public Internet, even when the IPv6/IPv4 hosts are located behind zero or more NATs. Teredo provides similar functionality to 6to4 but without needing edge devices that support 6to4 tunneling.

NOTE **For more information on IPv4/IPv6 transition technologies, see the white paper, "IPv6 Transition Technologies," at *http://www.microsoft.com/downloads/details.aspx?FamilyID=afe56282-2903-40f3-a5ba-a87bf92c096d&DisplayLang=en*.**

These three IPv6 transition technologies are supported by Windows 7, Windows Vista, Windows Server 2008, Windows XP SP2, and Windows Server 2003 SP1. Of the three, ISATAP is the primary transition technology that you should use for migrating an existing IPv4-based intranet to IPv6; it is discussed further in the following sections. Teredo is primarily useful in SOHO networking environments, where NAT-enabled broadband routers provide Internet connectivity for users. (Think of Teredo as a transition technology of last resort, because as IPv6 connectivity becomes ubiquitous, the need for NAT traversal will decline until Teredo is no longer needed.)

HOW IT WORKS

Blocking Teredo

Teredo is intended to be a consumer technology and has generally not been recommended for enterprises because Teredo requires the edge device to allow all outbound UDP traffic. For example, because of security reasons, many enterprise administrators do not want client computers on the corporate network to be directly accessible from the Internet, and in that case turning off Teredo is a good idea.

If administrators want to disable Teredo on their client computers or simply prevent it from working, they can do so in one of three ways:

- **Block all outbound UDP traffic by default. (This is the only reliable "external" method.)**
- **Block name resolution of the Teredo DNS host name, which by default on computers running Windows 7 is teredo.ipv6.microsoft.com. (This method, however, leaves an easy workaround, because the user can hard-code IP addresses.)**
- **Use Group Policy or a script to create the following DWORD registry value, which turns off Teredo on targeted computers running Windows 7. (This registry setting is not exposed by default in Group Policy but can be pushed down using a custom ADMX file.)**

  HKLM\SYSTEM\CurrentControlSet\Services\Tcpip6\Parameters\DisabledComponents

  You can specify the following settings for this value:
  - **0x10** Setting this value will disable Teredo only on the computer.
  - **0x01** Setting this value will disable all tunnel interfaces on the computer.

If administrators want to support only native IPv6 in their networks or if they don't want to support any IPv6 traffic until they deploy native IPv6, they can choose to turn off all tunneling technologies using the second choice in the preceding list.

Understanding ISATAP

By default, the IPv6 protocol in Windows 7 automatically configures a link-local unicast IPv6 address of the form FE80::5EFE:*w.x.y.z* (for private IPv4 addresses) or FE80::200:5EFE:*w.x.y.z* (for public IPv4 addresses). This address is a link-local ISATAP address, and it is assigned to the ISATAP tunneling interface. Using their link-local ISATAP addresses, two ISATAP hosts (such as computers running Windows 7) can communicate using IPv6 by tunneling across an IPv4-only network infrastructure (such as a network whose routers forward only IPv4 packets and not IPv6 packets).

NOTE **In Windows 7 and in Windows Vista SP1 or later versions, link-local ISATAP addresses are automatically configured only if the name "ISATAP" (the ISATAP router name) can be resolved. Otherwise, the ISATAP interface will be media disconnected. However, if you administratively enable ISATAP by using the *netsh interface isatap set state enabled* command, the link-local address will be configured regardless of whether the ISATAP router name can be resolved.**

With the addition of one or more ISATAP routers (IPv6-enabled routers that advertise address prefixes, forward packets between ISATAP hosts and other ISATAP routers, and act as default routers for ISATAP hosts), a variety of transition topologies become possible, including:

- Connecting ISATAP hosts on an IPv4-only intranet to an IPv6-capable network.
- Connecting multiple "islands" of ISATAP hosts through an IPv6-capable backbone.

These configurations are possible because ISATAP routers advertise address prefixes that enable ISATAP hosts (such as computers running Windows 7) to autoconfigure global or unique local unicast IPv6 addresses.

NOTE **Without the presence of an ISATAP router, ISATAP hosts running Windows Vista RTM could only autoconfigure link-local unicast IPv6 addresses, which limited IPv6 communications to those between hosts on the IPv4-only intranet. This was changed in Windows Vista SP1 so that without an ISATAP router, the interface will show media disconnected. In other words, Windows Vista SP1 won't configure a link-local ISATAP address when no ISATAP router is configured. The behavior in Windows 7 is the same as in Windows Vista SP1.**

NOTE **For more information on how ISATAP works, see the white paper, "IPv6 Transition Technologies," at *http://www.microsoft.com/downloads/details.aspx?FamilyID=afe56282-2903-40f3-a5ba-a87bf92c096d &displaylang=en*.**

DIRECT FROM THE SOURCE

ISATAP Interface Name

Xinyan Zan, Technical Lead
IPv6 Transition Technology

The ISATAP interface name is based on the DNS setting of the primary IPv4 interface of this ISATAP interface. For example, if the DNS suffix assigned to the primary IPv4 interface of this ISATAP interface is contoso.com, the ISATAP interface name will be isatap.contoso.com.

An alternate form of the ISATAP interface name is isatap.{GUID}, where GUID is a globally unique identifier. However, this GUID form is used to name the ISATAP interface only if there is no DNS suffix setting on the primary IPv4 interface.

Migrating an Intranet to IPv6

Best practices for migrating existing IPv4-based network infrastructures to IPv6 are still evolving. Therefore, this section presents a general outline on how to migrate an intranet to IPv6 and provides references to more detailed information on the subject for interested readers.

The ultimate goal of IPv4 to IPv6 migration is to achieve an IPv6-only network infrastructure that has IPv6-only hosts. From a practical standpoint, however, the lesser goal of achieving a network infrastructure that supports both IPv6 and IPv4—and where hosts also support both IPv6 and IPv4 but use mainly IPv6—is a more reasonable goal for which to aim. Achieving this goal is a lengthy process that involves seven main steps:

1. Upgrading your applications and services
2. Preparing your DNS infrastructure
3. Upgrading your hosts
4. Migrating from IPv4-only to ISATAP
5. Upgrading your routing infrastructure
6. Upgrading your DHCP infrastructure
7. Migrating from ISATAP to native IPv6

Step 1: Upgrading Your Applications and Services

To prepare your applications and services for migration, you will need to upgrade existing applications and services to support IPv6 in addition to IPv4. This may require upgrades from ISVs and third-party vendors or custom coding on your part. Although the ultimate goal is for all your applications and services to run native IPv6, a more appropriate target is to ensure that they work with both IPv4 and IPv6.

For further guidance, see the MSDN topic "IPv6 Guide for Windows Sockets Applications" at *http://msdn2.microsoft.com/en-us/library/ms738649.aspx*.

Step 2: Preparing Your DNS Infrastructure

You must prepare your DNS infrastructure to support the AAAA records used to resolve DNS names to IPv6 addresses. This might require upgrading your existing DNS servers. The DNS Server service of Windows Server 2008 and Windows Server 2003 supports dynamic registration of AAAA records for unicast IPv6 addresses (excluding link-local addresses).

MORE INFO **For more information on configuring Windows Server 2003 DNS servers to support IPv6 hosts, see Chapter 9, "Windows Support for DNS," in the online book *TCP/IP Fundamentals for Microsoft Windows*, which can be found at *http://technet.microsoft.com/en-us/library/bb727009.aspx*.**

Step 3: Upgrading Your Hosts

You may need to upgrade some of your hosts until all your hosts support both IPv6 and IPv4. Windows platforms from Windows XP SP2 onward support both IPv4 and IPv6, although full support for IPv6 functionality for built-in programs and services is provided only in Windows Vista and later versions.

Step 4: Migrating from IPv4-only to ISATAP

After you prepare your applications, services, hosts, and DNS/DHCP infrastructure, you can begin deploying ISATAP routers to create islands of IPv6 connectivity within your IPv4-based intranet. You will need to add A records to the appropriate DNS zones so that your ISATAP hosts can determine the IPv4 addresses of your ISATAP routers.

You may decide to deploy zero or more ISATAP routers for inter-ISATAP subnet routing within your intranet, depending on the size of your intranet and the geographical distribution of its sites. You may decide to deploy redundant ISATAP routers to provide consistent availability of IPv6 address prefixes and other configuration settings for your ISATAP hosts. You will also likely deploy one or more ISATAP routers to provide IPv6 connectivity between your IPv4-based network infrastructure and the public IPv6 Internet as this evolves.

For more information on deploying ISATAP routers using different migration scenarios, see the white paper, "Intra-site Automatic Tunnel Addressing Protocol Deployment Guide," at *http://www.microsoft.com/downloads/details.aspx?FamilyID=0f3a8868-e337-43d1-b271-b8c8702344cd &displaylang=en*.

Step 5: Upgrading Your Routing Infrastructure

After you have deployed ISATAP to enable IPv6 hosts to communicate over your IPv4 network infrastructure, you should begin upgrading your network infrastructure (including routers, gateways, and other access devices) to support IPv6. Rather than upgrading your infrastructure to support only IPv6, a more reasonable upgrade goal is dual IPv4/IPv6 support. In many cases, actual replacement of router hardware is not necessary. Because many modern hardware routers support both IPv4 and IPv6 routing, the task of upgrading your routing infrastructure to support IPv6 becomes configuration, not replacement. As you enable IPv6 routing support for a subnet, also enable the DHCPv6 relay agent for the subnet.

Typically, you will begin upgrading your routing infrastructure early in your ISATAP deployment by upgrading the core routers on your network backbone to support IPv6. This will create islands of ISATAP hosts that connect to this backbone to communicate with other IPv6 hosts anywhere in your intranet.

Step 6: Upgrading Your DHCP Infrastructure

You can optionally upgrade your routing and DHCP infrastructure to support DHCPv6 for automatic assignment of global or unique local unicast IPv6 addresses or configuration settings for IPv4/IPv6 nodes on your network. By using DHCPv6, an IPv6 host can obtain subnet prefixes and other IPv6 configuration settings. A common use of DHCPv6 is to configure Windows 7–based client computers with the IPv6 addresses of DNS servers on the network. (DNS servers are not configured through IPv6 router discovery.)

The DHCP Server service in Windows Server 2003 does not support stateful address autoconfiguration or the DHCPv6 protocol. The DHCP Server role in Windows Server 2008, however, supports both stateful and stateless IPv6 address autoconfiguration using DHCPv6. The DHCP Client service in Windows 7, Windows Vista, and Windows Server 2008 supports address autoconfiguration using DHCPv6.

Just as with DHCP with IPv4, you also need to deploy and configure DHCPv6 relay agents for each subnet containing Windows 7 clients. Many hardware routers already support a DHCPv6 relay agent. You must configure relay agents with the IPv6 addresses of the DHCPv6 servers on your network. Relay agents can be configured but should not be enabled until you deploy IPv6 routing on your subnets.

When you are ready to enable DHCPv6 on subnets, configure your IPv6 routers to set the *Managed Address Configuration* and *Other Stateful Configuration* flags to the appropriate values for stateful or stateless DHCPv6 operation. For more information, see the Cable Guy article titled "The DHCPv6 Protocol" at *http://www.microsoft.com/technet/technetmag/issues/2007/03/CableGuy/default.aspx*.

Step 7: Migrating from ISATAP to Native IPv6

Finally, when all your network infrastructure devices support IPv6, you can begin to decommission your ISATAP routers because you no longer need them. Whether you will also migrate your infrastructure and hosts to support only pure-IPv6 is a decision best left for the distant future.

DIRECT FROM THE SOURCE

Tips and Tricks for Transitioning from IPv4 to IPv6

Mike Owen, Network Engineer
Data and Storage Platform Division

When transitioning a network from IPv4-only to dual stack, there are several areas that need special attention.

Addressing

This is actually one area that gets easier with IPv6 due to the huge address space that it offers. In general, you will want to add to each individual network segment a single IPv6 /64 prefix, even in cases in which you have more than one IPv4 subnet assigned to the same network (for example, by using the secondary keyword on Cisco routers). You should not need to use unique local addresses, even for lab networks. One exception might be that you do not want to use a routable /64 prefix for a segment that is not connected to your organization's globally routable space (that is, it is physically separate).

Firewalls

Deploying IPv6 can present issues for an organization's security team. Because IPsec services are available in all IPv6 stacks, it is more common to see end-to-end security implemented with IPv6-enabled desktops. When faced with end-to-end encryption, a firewall administrator has one of two choices: Either deny the traffic and drop it at the perimeter or allow it through unchecked, thus bypassing the access control lists (ACLs) and other security enabled on the firewall. Note that this problem exists even with IPv6-enabled firewalls.

Tunneling Technologies

Many transition technologies, such as ISATAP, 6to4, and manually configured IPv6-in-IPv4 tunnels, encapsulate IPv6 packets inside IPv4 to transport them across an IPv4-only part of your network. These packets are identified by the use of IP protocol 41 in the encapsulating packet. If firewalls, ACLs, or other devices in your network are not configured to forward these packets, then communications using these technologies will break. Many home routers, for example, are configured by default to only forward UDP and TCP protocols.

Here's a real-life example: After configuring a router to provide IPv6 services at an IANA meeting in Florida, IPv6 connectivity was not working. After some trouble-shooting with the service provider, I determined that their router was dropping IP protocol 41, thus preventing IPv6 connectivity across the service provider's IPv4-only network.

Network Applications

When deciding to IPv6-enable an existing workflow or application, make sure to consider all parts of the process. For example, while upgrading a Web front end to support IPv6, don't forget to enable the separate file store and back-end database servers as well, otherwise the workflow may appear to support IPv6 from the front end but actually will not be completely tested.

DNS

Many DNS products today support the AAAA records which are used to store name-to-address mappings for IPv6 end systems. However, that does not mean that they support IPv6 lookups against the database—in some cases, this functionality must be enabled through a configuration setting or an upgrade to the product itself. This is another part of an end-to-end IPv6 workflow that needs to be considered.

Address Management

A simple way to enable IPv6 autoconfiguration on your hosts is to configure your edge routers to advertise an IPv6 prefix via Router Advertisements. This enables IPv6-enabled operating systems, including Windows Vista, Windows Server 2008, and Windows 7, to configure themselves with an IPv6 address. This method of configuration is considered stateless because the router will not track which IPv6 addresses are configured on which end system. When performing address auditing against these systems (to investigate a security incident, for example), it is impossible to determine which host was assigned a given IPv6 address at a particular time. At best, if you are lucky, the router's ARP tables will contain the necessary information, but more often than not, you will be unable to track a specific IPv6 address to the host on which it was configured.

Here's a real-life example: At a previous job, I was contacted by the local office of the U.S. Secret Service to investigate a threat made against a government official. I was able to track the IPv4 address that they provided to a high school in the school district where I worked and to a specific classroom at a certain time, based on DHCP logs and switch CAM tables. A student was subsequently identified as being in the classroom alone at the time and admitted to sending the messages, which turned out to be a hoax. Tracking down autoconfigured IPv6 addresses at this level of detail is nearly impossible.

Summary

This chapter described the features of IPv6 in Windows 7, provided an overview of how IPv6 works, and outlined best practices for migrating an existing IPv4-only network to IPv6. An IPv6 migration requires careful planning and a thorough understanding of how IPv6 works, and both Windows 7 and Windows Server 2008 R2 provide the features and tools you need to migrate your network successfully.

Additional Resources

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

Related Information

- *Understanding IPv6, Second Edition,* by Joseph Davies (Microsoft Press, 2008). See *http://www.microsoft.com/MSPress/books/11607.aspx*.
- The IPv6 home page on Microsoft TechNet at *http://www.microsoft.com/ipv6/*.
- The IPv6 blog of Sean Siler, IPv6 Program Manager, at *http://blogs.technet.com/ipv6*.
- "IPv6 for Microsoft Windows: Frequently Asked Questions" at *http://technet.microsoft.com/en-us/network/cc987595.aspx*.
- The white paper, "Introduction to IP Version 6," at *http://www.microsoft.com/downloads/details.aspx?FamilyID=CBC0B8A3-B6A4-4952-BBE6-D976624C257C&displaylang=en*.
- The white paper, "IPv6 Transition Technologies," at *http://www.microsoft.com/downloads/details.aspx?FamilyID=afe56282-2903-40f3-a5ba-a87bf92c096d&displaylang=en*.
- The white paper, "Intra-site Automatic Tunnel Addressing Protocol Deployment Guide," at *http://www.microsoft.com/downloads/details.aspx?FamilyID=0f3a8868-e337-43d1-b271-b8c8702344cd &displaylang=en*.
- The Cable Guy article, "Understanding the IPv6 Routing Table," at *http://technet.microsoft.com/en-us/library/bb878115.aspx*.
- The Cable Guy article, "Manual Configuration for IPv6," at *http://technet.microsoft.com/en-us/library/bb878102.aspx*.
- The Cable Guy article, "Troubleshooting IPv6," at *http://technet.microsoft.com/en-us/library/bb878005.aspx*.
- The Cable Guy article, "Source and Destination Address Selection for IPv6," found on Microsoft TechNet at *http://technet.microsoft.com/en-us/library/bb877985.aspx*.
- "Domain Name System Client Behavior in Windows Vista" on Microsoft TechNet at *http://technet.microsoft.com/en-us/library/bb727035.aspx*.

- Knowledge Base article 929852, "How to Disable Certain Internet Protocol Version 6 (IPv6) Components in Windows Vista, Windows 7 and Windows Server 2008," at *http://support.microsoft.com/kb/929852*.
- Knowledge Base article 929851, "The Default Dynamic Port Range for TCP/IP Has Changed in Windows Vista and in Windows Server 2008," at *http://support.microsoft.com/kb/929851*.
- Chapter 9, "Windows Support for DNS," and Chapter 12, "Troubleshooting TCP/IP," in the online book *TCP/IP Fundamentals for Microsoft Windows,* which you can download from *http://www.microsoft.com/downloads/details.aspx?FamilyID=c76296fd-61c9-4079-a0bb-582bca4a846f &displaylang=en*.

On the Companion Media

- Get-IPV6.ps1

Index

Symbols and Numbers

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z