Quick answers to common problems

# Windows Server 2012 R2 Administrator Cookbook

Over 80 hands-on recipes to effectively administer and manage your Windows Server 2012 R2 infrastructure in enterprise environments

Jordan Krause

[PACKT] PUBLISHING enterprise
professional expertise distilled

# Windows Server 2012 R2 Administrator Cookbook

Over 80 hands-on recipes to effectively administer and manage your Windows Server 2012 R2 infrastructure in enterprise environments

**Jordan Krause**

[PACKT] PUBLISHING enterprise
professional expertise distilled

BIRMINGHAM - MUMBAI

# Windows Server 2012 R2 Administrator Cookbook



First published: January 2015

Production reference: 1220115

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham B3 2PB, UK.

ISBN 978-1-78439-307-6

`www.packtpub.com`

# Credits

**Author**
Jordan Krause

**Reviewers**
Bede Carroll
Jochen Nickel
Matthew Stone
Kunal Udapi

**Commissioning Editor**
Kunal Parikh

**Acquisition Editor**
Vinay Argekar

**Content Development Editor**
Pooja Nair

**Technical Editor**
Prajakta Mhatre

**Copy Editor**
Ameesha Green

**Project Coordinator**
Suzanne Coutinho

**Proofreaders**
Ting Baker
Stephen Copestake
Lawrence A. Herman

**Indexer**
Hemangini Bari

**Production Coordinator**
Melwyn D'sa

**Cover Work**
Melwyn D'sa

# About the Author

**Jordan Krause** is a Microsoft MVP in the Enterprise Security group. He has the unique opportunity to work with the Microsoft networking technologies daily as a senior engineer at IVO Networks. Jordan specializes in Microsoft DirectAccess and has authored *Microsoft DirectAccess Best Practices and Troubleshooting*, *Packt Publishing*, one of the few books available worldwide on this subject. His additional technical expertise includes Microsoft Unified Access Gateway (UAG) and Microsoft Threat Management Gateway (TMG). He spends the majority of each workday planning, designing, and implementing DirectAccess for companies around the world.

Committed to continuous learning, Jordan holds Microsoft certifications as an MCP, MCTS, MCSA, and MCITP Enterprise Administrator. He regularly writes tech notes and articles reflecting his experiences with the Microsoft networking technologies; these can be found at: `http://www.ivonetworks.com/news`.

Jordan also strives to spend time helping the DirectAccess community, mostly by way of the Microsoft TechNet forums. Always open to direct contact, he encourages anyone who needs assistance to head over to the forums and find him personally. Jordan lives and works in the ever-changing climate that is Michigan.

# About the Reviewers

**Bede Carroll** is an IT everyman, working with servers, networking, and storage when not dabbling with some coding for fun. His desire to learn has led him to hold many industry certifications from Microsoft, Cisco, VMware, and EMC to name but a few. Bede has many years of industry experience obtained while working in small business environments all the way through to Fortune Global 500 organizations. This diverse experience has given Bede a unique perspective on the industry and the ability to respond to any client needs.

Bede can be found on Twitter `@bedecarroll`, on LinkedIn at `linkedin.com/in/bedecarroll`, and occasionally blogging at `www.bedecarroll.com`.


I would like to thank my wonderful family for their love and support in helping me to achieve my goals.


**Jochen Nickel** is a Cloud, Identity, and Access Management Solution Architect with a focus and deep technical knowledge on the Microsoft Enterprise Mobility Suite. He currently works for inovit GmbH in Switzerland and spends the majority of each work day planning, designing, and implementing single parts such as Azure Active Directory Premium, Azure Rights Management Services, or complete Enterprise Mobility Suite solutions.

Also, he has been part of many projects, proof of concepts, reviews, and workshops in this field of technology. Furthermore, he is a Microsoft V-TSP Security, Identity, and Access Management, Microsoft Switzerland, and he uses his experience for directly managed business accounts in Switzerland. He is also an established speaker at many technology conferences.

As an active writer and reviewer, Jochen authored *Learning Microsoft Windows Server 2012 Dynamic Access Control* and reviewed *Windows Server 2012 Unified Remote Access Planning and Deployment* by Erez Ben-Ari and Bala Natarajan, both by Packt Publishing.

Committed to continuous learning, he holds Microsoft certifications such as MCITP, MCSE/A, MCTS, MTA, and many other security titles. He enjoys spending as much time as possible with his family to get back the energy to handle such interesting technologies.

He regularly blogs at `www.inovit.ch/blog.idam.ch`.

**Matthew Stone** is an IT veteran, having worked on a wide range of technologies over the last 20 years, including Windows Server and many other Microsoft server technologies. In addition to systems administration, he has also served as a technical writer, most recently writing a training course on Office 365 administration and deployment. His current focus is on orchestration and configuration management in the enterprise, focusing on Windows Server and Puppet technologies.

**Kunal Udapi** has acquired more than 8 years of IT experience in SMB and Enterprise infrastructure design, implementation, and administration. He has provided architectural and engineering expertise in a variety of virtualization, data center, and private cloud-based engagements for company's internal IT team environment. He currently holds the title of Senior System Administrator at a company in Pune, Maharashtra, India.

Kunal holds numerous technical certifications (Windows and VMware) and was awarded VMware vExpert in 2014. He frequently answer users' questions on VMware Technology Network (VMTN) and LinkedIn groups.

Kunal is an independent blogger for `http://kunaludapi.blogspot.com`. He focuses on creating content that revolves around VMware virtualization and PowerShell/PowerCLI automation, which benefits the technology community.

# www.PacktPub.com

## Support files, eBooks, discount offers, and more

For support files and downloads related to your book, please visit www.PacktPub.com.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.PacktPub.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at service@packtpub.com for more details.

At www.PacktPub.com, you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.

PACKTLIB®

https://www2.packtpub.com/books/subscription/packtlib

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can search, access, and read Packt's entire library of books.

## Why subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print, and bookmark content
- On demand and accessible via a web browser

## Free access for Packt account holders

If you have an account with Packt at www.PacktPub.com, you can use this to access PacktLib today and view 9 entirely free books. Simply use your login credentials for immediate access.

## Instant updates on new Packt books

Get notified! Find out when new books are published by following @PacktEnterprise on Twitter or the *Packt Enterprise* Facebook page.

# Table of Contents

# Preface

I still remember the day that I was first thrown into the server world. I had been working with Desktop computers for years, even had quite a bit of experience with other items such as printers, copy machines, and phone systems. Technology was familiar to me, Microsoft Windows was familiar to me, but the company where I worked was large enough that Desktop support and Server support were separated, and there wasn't a lot of technical interaction between the sides. Other than office pranks and lots of humorous emails forwards flying around, naturally. So when I had a sudden chance to change jobs squarely into a server support role, I was a little hesitant at first. When I found out my first task would be implementing DFSR, and I thought that stood for "Digital something-or-other", I may have been a little freaked out. I start with this little story not to try to make you chuckle at my expense, but because, if we fast forward to several years down the road, I suddenly find myself with an opportunity to put together a book on the new version of this incredible operating system, Windows Server 2012 R2.

Yes, this is getting to the point. There were many different angles from which the recipes contained within these pages could have come. Do we stick to features that are brand-new in Windows Server 2012 R2 and not in previous versions of the operating system? How advanced should the topics be? What about an inexperienced admin who is reading this book? What about those who are trying to get familiar with the server world because they don't have the opportunity to have hands-on experience of it during their day job? These questions, combined with my own situation as described above have led to my writing the hundreds of pages that make up this book.

What would I have wanted to know when I started working with servers? What are some of the things that I have learned over the years that I still continue to pass down to new customers that I work with? How can I make use of this new Server operating system from Microsoft in my own company? What are some of the lesser known topics in Server 2012 R2 that I myself am interested in digging deeper into? These kinds of questions became the new baseline for putting together recipes for this book. I sincerely hope that anyone who reads this—from the fresh, brand-new server administrator who has nothing but a server to learn on, to the tenured, experienced admin who is coming up to speed on the latest iteration (and certainly the best ever Windows Server operating system)—can all take something out of this text and apply it to their jobs and lives in a way that benefits them.

The beauty of the Windows Server operating system has always been how incredibly vast it is, and Server 2012 R2 is certainly no exception. There are loads of different roles, features, and functions that you can run in order to do almost anything within your network. Need a file server? Check. Directory services to house your user and computer accounts? Look no further. Secure remote access solution? Yes sir. Need to publish certificates for a website that you want to present to the internet? Yup, we can even do that if we so desire.

We are going to start with some recipes that get us familiar with the graphical interface of Server 2012 R2. I have the unique opportunity to work with new admins almost daily, from all kinds of different companies and corners of the world. Navigating around inside the Server 2012 and Server 2012 R2 systems is still a struggle that I see constantly. Once we know how to get around comfortably, we will move on to the configuration and utilization of core infrastructure technologies that are absolutely need-to-know information for any admin interested in working within a Microsoft network. Following that, we will dig down into some of the individual facets of technology that are provided by this amazing operating system. I will let the chapters speak for themselves, but I strive with these recipes to cover a lot of different bases, and to keep things interesting as we round those bases.

We will certainly discuss new material and features that are only available in this newest version of Windows Server, but additionally there will be recipes that apply not only in R2 but also to systems across your entire network. By starting to use these functions that may even be new to an experienced admin, the hope is that your daily chores and tasks can become more streamlined and efficient along the way.

PowerShell is also discussed and scattered throughout this book. At one point, there was an intention for a separate chapter dedicated to PowerShell but, as the recipes came together, it became clear that they should not stand on their own but ought to be embedded within the chapters to which they apply. PowerShell is fully integrated into every aspect of Windows Server 2012 R2. Any task that you want to perform, *any* task, can be performed via a PowerShell command or script if you so choose. It is the core of the way that Server 2012 R2 is built. In fact, many of the things that you configure in the GUI are really just calling for pre-built PowerShell scripts to run in the background to do the actual configuration. Because PowerShell is integral to the entire operating system, we incorporated it throughout this book rather than lumping it all together.

By the time you reach the end of this book, my hope is that even someone completely unfamiliar with the Windows Server operating system will have the ability to build their own Microsoft-centric network from the ground up. Even though the recipes are grouped together in chapters and follow a fairly logical sequence, there is no need to read this front-to-back. Each recipe has the power to stand on its own, and to be useful with or without the recipes that precede it. Feel free to skip around, grab the data you are currently interested in, or use this book as a reference guide in the future. Enjoy!

## What this book covers

*Chapter 1*, *Learning the Interface*, starts us on our journey working with Windows Server 2012 R2 as we figure out how to navigate the look and feel of this new operating system, and gain some tips and tricks to make our daily chores more efficient.

*Chapter 2*, *Core Infrastructure Tasks*, will take us through configuring and working with the core Microsoft technology stack. The recipes contained in this chapter are what I consider essential knowledge for any administrator who intends to work in a Windows network.

*Chapter 3*, *Security and Networking*, teaches us some methods for locking down access on our servers. We will also cover commands that can be very useful tools as you start monitoring network traffic.

*Chapter 4*, *Working with Certificates*, will start to get us comfortable with the creation and distribution of certificates within our network. PKI is an area that is becoming more and more prevalent, but the majority of server administrators have not yet had an opportunity to work hands-on with them.

*Chapter 5*, *Internet Information Services*, brings us into the configuration of a Windows Server 2012 R2 box as a web server in our network. Strangely, in the field I find a lot of Microsoft networks with Apache web servers floating around. Let's explore Internet Information Services (IIS) as a better alternative.

*Chapter 6*, *Remote Access*, digs into using your Server 2012 R2 as the connectivity platform that brings your remote computers into the corporate network. We will discuss DirectAccess and VPN in this chapter.

*Chapter 7*, *Remote Desktop Services*, encourages you to look into using Server 2012 R2 as a virtual session host or VDI solution. RDS can be an incredibly powerful tool for anyone interested in centralized computing.

*Chapter 8*, *Monitoring and Backup*, covers some of the capabilities included with Server 2012 R2 to help keep tabs on the servers running in your infrastructure. From monitoring system performance and IP address management to backing up and restoring data using the tools baked into Windows, these recipes will walk you through some helpful tasks related to monitoring and backup.

*Chapter 9*, *Group Policy*, takes us into the incredibly powerful and far reaching management powers contained within Active Directory that are provided out of the box with Windows Server 2012 R2.

*Chapter 10*, *File Services and Data Control*, provides us with information and step-by-step recipes on some of the lesser known ways that data can be managed on a Windows server. We will cover technologies such as DFSR, iSCSI, and Server 2012 R2 Work Folders.

## What you need for this book

All of the technologies and features that are discussed in the recipes of this book are included with Windows Server 2012 R2. As long as you have access to the operating system installer disc and a piece of hardware on which you can install it, you will be able to follow along with our lessons. In fact, if you are brand new to the Microsoft world and are not sure how to acquire the Windows Server 2012 R2 installation media, you can download an evaluation copy of it here:

`http://www.microsoft.com/en-us/evalcenter/evaluate-windows-server-2012-r2`

Many of the tasks that we are going to accomplish together require a certain amount of base networking and infrastructure to be configured, in order to fully test the technologies that we are working with. If you have access to a Hyper-V server on which you can build multiple virtual machines running Windows Server 2012 R2, you will be able to build recipe upon recipe as we move through setting up the core infrastructure tasks; you can then utilize those servers that we have configured in order to build up later recipes. Building out a lab network that includes core Microsoft roles such as Active Directory, DNS, DHCP, certificates, and web and file servers will help you tremendously as you move through this book. And if you are not familiar with building out that lab, never fear. Many of the recipes contained within will help you to structure out the lab itself.

## Who this book is for

This book is intended primarily for IT administrators who are looking to learn more about the latest Windows Server operating system available from Microsoft: Server 2012 R2. Those who are coming into this book as new server administrators, or maybe some of you are working on the desktop side and are interested in moving over to the server side of the house, will have the most potential for knowledge gain as we will cover topics within the core infrastructure technologies that more experienced administrators may take for granted. However, deeper into the chapters and scattered throughout the book are recipes that dig in further, causing even the more knowledgeable and experienced server administrators to maybe look at familiar technologies in a new light, or to accomplish a normal task in a new way. I sincerely hope that anyone who reads this gets something new out of it that they start to use in their daily IT life.

## Sections

In this book, you will find several headings that appear frequently (Getting ready, How to do it, How it works, There's more, and See also).

To give clear instructions on how to complete a recipe, we use these sections as follows:

### Getting ready

This section tells you what to expect in the recipe, and describes how to set up any software or any preliminary settings required for the recipe.

### How to do it...

This section contains the steps required to follow the recipe.

### How it works...

This section usually consists of a detailed explanation of what happened in the previous section.

### There's more...

This section consists of additional information about the recipe in order to make the reader more knowledgeable about the recipe.

### See also

This section provides helpful links to other useful information for the recipe.

## Conventions

In this book, you will find a number of text styles that distinguish between different kinds of information. Here are some examples of these styles and an explanation of their meaning.

Code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles are shown as follows: "Now instead of using a simple `dir`, give this command a try: `Dir | Format-List`."

A block of code is set as follows:

```
Get-WmiObject –Class Win32_OperatingSystem –ComputerName localhost |
Select-Object –Property @{n="Last Boot Time";
e={[Management.ManagementDateTimeConverter]::ToDateTime($_.
LastBootUpTime)}}
```

Any command-line input or output is written as follows:

```
route add –p 192.168.1.0 mask 255.255.255.0 192.168.0.254 if 12
```

New terms and important words are shown in bold. Words that you see on the screen, for example, in menus or dialog boxes, appear in the text like this: "Go ahead and click on the **Add Features** button to allow this."

> Warnings or important notes appear in a box like this.

> Tips and tricks appear like this.

# Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book—what you liked or disliked. Reader feedback is important for us as it helps us develop titles that you will really get the most out of.

To send us general feedback, simply e-mail `feedback@packtpub.com`, and mention the book's title in the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide at `www.packtpub.com/authors`.

# Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

## Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books—maybe a mistake in the text or the code—we would be grateful if you could report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting `http://www.packtpub.com/submit-errata`, selecting your book, clicking on the **Errata Submission Form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded to our website or added to any list of existing errata under the Errata section of that title.

To view the previously submitted errata, go to `https://www.packtpub.com/books/content/support` and enter the name of the book in the search field. The required information will appear under the **Errata** section.

## Piracy

Piracy of copyrighted material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works in any form on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at `copyright@packtpub.com` with a link to the suspected pirated material.

We appreciate your help in protecting our authors and our ability to bring you valuable content.

## Questions

If you have a problem with any aspect of this book, you can contact us at `questions@packtpub.com`, and we will do our best to address the problem.

# 1

# Learning the Interface

In an effort to become familiar with the look and feel of Windows Server 2012 R2, you will learn how to navigate through some daily tasks using the graphical interface. On our agenda in this chapter are the following recipes:

- Shutting down or restarting the server
- Launching Administrative Tools
- Using WinKey + *X* for quick admin tasks
- Using the Search function to launch applications quickly
- Managing remote servers from a single pane with Server Manager
- Using PowerShell to accomplish any function in Windows Server
- Installing a role or feature
- Administering Server 2012 R2 from a Windows 8.x machine
- Identifying useful keyboard shortcuts in Server 2012 R2
- Setting your PowerShell Execution Policy
- Building and executing your first PowerShell script
- Searching for PowerShell cmdlets with Get-Help

## Introduction

I meet new server administrators almost daily, from all types of businesses and walks of life. Often during our projects, we introduce the very first Windows Server 2012 R2 server into an environment. This gives me the unique opportunity to observe first interactions with the updated interface, and most of the time first impressions are pretty terrible. Now don't get me wrong, after using Windows Server 2012 R2 for only a little while, I think you'll all agree that it has some of the best functionality and stability we have ever seen out of a server operating system. What I see people struggle with all the time when starting to use 2012 R2 is the user interface. This is the same challenge that has plagued Windows 8 since its release. Everything seems to look and feel different than the Windows we have always known, trying to find applications can be futile, and what in the world is that "tablet-touch-screen-wannabe-thing" that consumes my entire screen all the time? And the charms bars that pop out? Don't get me started.

These big changes in the way that we interact with the operating system can be a nuisance on a client Desktop machine, but can be downright dangerous on a server until you have a good handle on managing the interface. When you are working within three levels of **Remote Desktop Protocol** (**RDP**), bouncing from one server to another, all of these little differences are compounded. It suddenly becomes difficult to know which server it is that you are changing. Let's have a show of hands, how many of you have mistakenly rebooted the wrong server? Or even more likely, how many of you have rebooted your own computer while you were trying to reboot a remote server? I know I have! And not just once.

Hope is not lost! I promise you that once you learn to manage the interface, rather than letting it manage you, some of these changes may start to seem like good ideas. They can increase productivity and the ease of accomplishing tasks—we just need some pointers on making the best use of the new interface.

The recipes in this chapter are dedicated to doing just that. Let's work together to gain a better understanding of why the interface was built the way it is, and learn to take advantage of these new screens and settings.

## Shutting down or restarting the server

I just couldn't resist starting with this one. Yes, this seems trivial. Silly even. However, the amount of times that I have watched a simple server restart consume more mouse clicks than creating a Domain Controller has convinced me that this needed to be in the book. Perhaps the shutdown and restart options were hidden away purposefully, because once your system is up and running, there is not often a need to accomplish either of these tasks. When first configuring the box though, it is very common to have to reboot a couple of times, or to shut down a machine to move it to another location.

## Getting ready

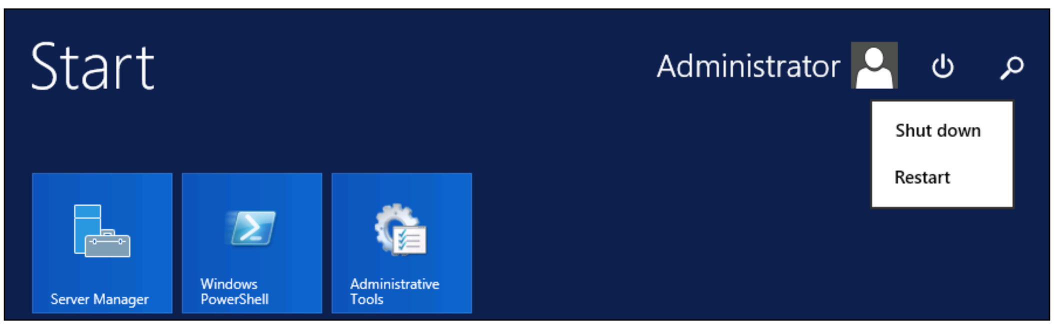
To step through this recipe, you will need a Windows Server 2012 R2 system online. There are no other prerequisites.

## How to do it...

Let's take a look at three different ways to shut down or restart your system. The first two are the most commonly employed. The third is less commonly known, but is by far my preference when tasked with restarting a remote server.

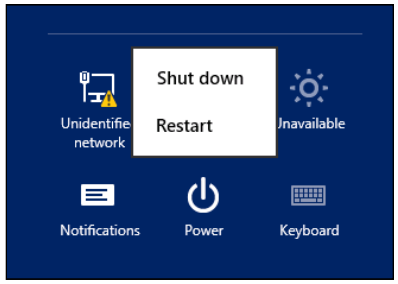
The first option would be to use the power button on the **Start** screen. So far, I don't think that many administrators have stumbled into this one, because it didn't exist in Server 2012.

1. Click on your "Start Button", or rather your button in the Taskbar that opens your **Start** screen. You will see a power icon in the top-right corner of that **Start** screen.
2. Click on this for **Shut down** and **Restart** options.




The second method to accomplish shutting down or restarting is by using the charms bar. The **charms bar** is the little menu that presents itself when you have your mouse sitting near the upper-right corner of your screen. On a touchscreen computer, you swipe your finger from the right side of the screen toward the middle to make it appear. Go ahead and hover the mouse in the upper-right corner of your screen until the charms bar appears. Then click on **Settings**, and then on **Power**. Unfortunately, this is the most common way to attempt to restart a Server 2012 R2 machine. I say unfortunately because it is so easy to hover in the corner and display the wrong charms bar, the one for your own Desktop for example.

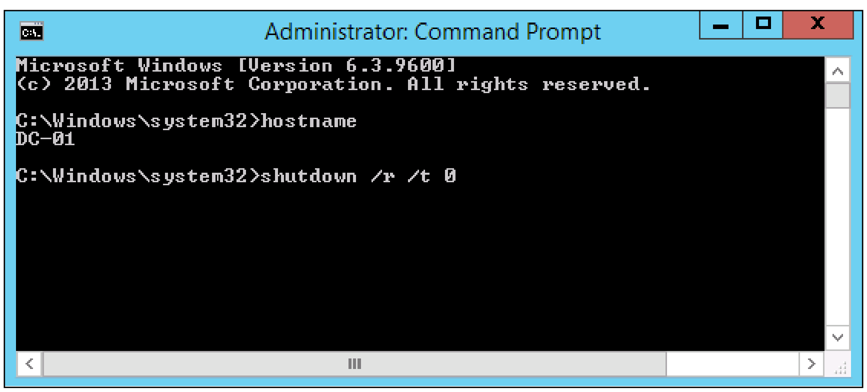
Many mistaken reboots are caused by this.




The most definitive, and dare I say most fun way of restarting your server is to utilize Command Prompt. Doing this gives you the opportunity to double check that you are manipulating the correct machine. Open up **Command Prompt** and run a quick `hostname` check to make sure you are restarting the one you really intend to. Then utilize the `shutdown` command to take care of the rest. This process can be especially helpful when logged into remote servers using RDP. Use the following commands to perform the explained operations:

```
hostname
shutdown /r /t 0
```

If you were to simply type `shutdown`, the server would shut itself down in 60 seconds. Using `/r` indicates a restart rather than a shut down, and `/t 0` is a timing flag that indicates the number of seconds the server should wait before restarting. Specifying slash zero here tells it to wait for zero seconds before initiating the restart.



## How it works...

Shutting down or restarting a server doesn't require a lot of explanation, but I hope that this small recipe gets some thought going about creative ways to do regular tasks. As you will see throughout this book, you can accomplish anything in Windows Server 2012 R2 through the use of commands or scripts. You could easily turn the `shutdown` command, the last example that we tested in this recipe, into a batch file, and place it on the Desktop of each of your servers as a quick double-click option for accomplishing this task.

However, I work with RDP windows inside RDP windows very often. When you're bouncing around between a dozen servers that all have the same background, I have decided that the only sure-fire way to make sure you are restarting the correct device is to do a quick `hostname` check before you initiate the restart. If you are interested in discovering all of the available flags that are available to use with the `shutdown` command, make sure to type in `shutdown /?` some time to take a look at all of the available options.

> Using the command prompt is also an easy way to log off a server. Let's say you are layers deep in RDP and want to log off from a single server (not all of them). Are you sure you clicked on the **Start** button of the right server? Instead, open up a prompt and simply type `Logoff`.

# Launching Administrative Tools

Earlier versions of Windows Server placed all of the Administrative Tools in a self-named folder right inside the Start menu. Server 2012 R2 no longer has a Start menu, but rather a full Start screen. It is my experience that administrators prefer to stay on the Desktop as much as possible, and do not like switching back and forth between the two. In other words, they want to accomplish everything without having to use the Start screen. Let's take a look at some ways to launch Administrative Tools right from your server's familiar Desktop interface.

## Getting ready

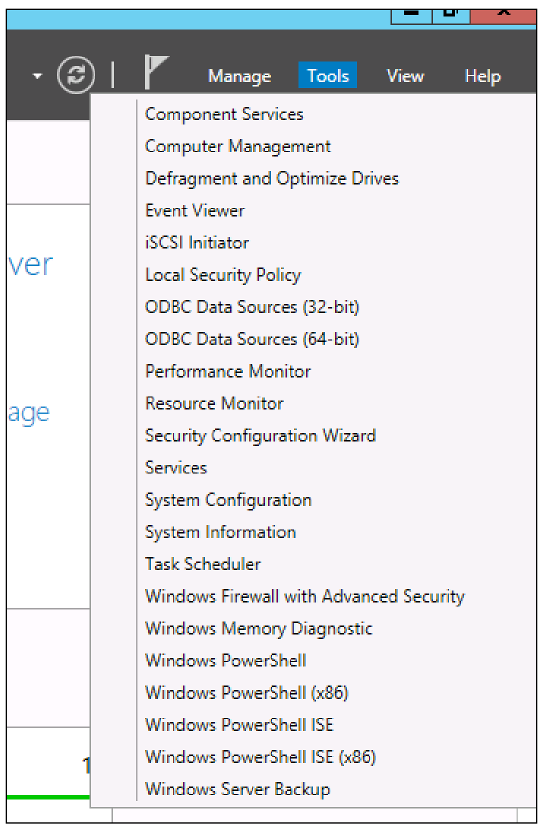
All you really need is a Windows Server 2012 R2 machine online. The more roles and services that you have running on it, the more options that you will see on your screen as we navigate these menus.

## How to do it...

To launch Administrative Tools from your Desktop, perform the following steps:

1. Open up **Server Manager**. In fact, if you just logged into the server, it's probably already open for you.
2. Click on **Tools** in the upper-right corner.

There you go. A full list of all the admin tools installed onto that server. Heading into this list is also a quick way of taking a look into what a particular server is doing, which you can take an educated guess at based on what roles and services are installed. By looking at the screenshot below, we can see that this is a pretty standard server instance that is not yet running any big roles. Once added, the management tools for the Windows roles will present themselves in this list.



## How it works...

Since Server Manager likes to open automatically when logging in, let's make quick use of it for opening the tools that we need to do our jobs. Another way to have easy access to your tools from the Desktop is to create shortcuts, or to pin each of them to your Taskbar. Sometimes this isn't as easy as it sounds. In the past, these tools were all grouped together in the `Administrative Tools` folder, so you didn't have any reason to memorize the exact names of the tools. If you were to head into the Start screen and try to recall the name of the service you are looking for to type into the Search field, it may not immediately come to you. If you're a consultant working on someone else's server, you may not want to pin anything to their Desktop anyway, and you certainly don't want to resort to using Bing in front of them to look up the name of the service. So I like to stick with launching Administrative Tools from Server Manager.

# Using WinKey + X for quick admin tasks

There are some functions in Windows that a server administrator needs to use all the time. Instead of making shortcuts or pinning them all to the Taskbar, let's get to know this hidden menu that is extremely useful for launching these commonly used admin tools.

## Getting ready

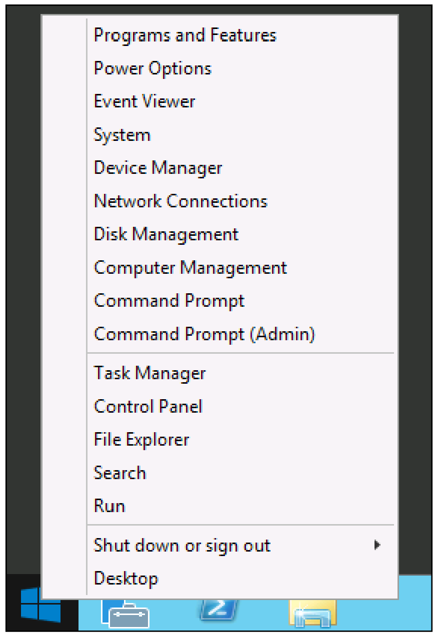
A running Windows Server 2012 R2 machine is all we need to highlight this one. In fact, this menu also exists on any Windows 8.x computer, so make use of it often!

## How to do it...

There are two ways to open this little menu. While you are in the Server 2012 R2 Desktop, you can perform either of these steps:

- Hold down your Windows key on the keyboard and press *X*

- Hover your mouse over the Windows flag in the lower-left corner of the Desktop. You know the button that opens the **Start** screen if you left click on it. Now go ahead and **right-click** on that button and you will see a menu as shown in the following screenshot:




## How it works...

This little "quick-tasks admin menu" is very easy to open, and is very convenient for launching programs and settings that are accessed often. I won't talk too much about what particulars are in the menu as it's pretty self-explanatory, but I use this menu multiple times per day to open up the System properties and the command prompt. Particularly the command prompt, as it has an option to open an administrative command prompt right from the menu.

Look at that, you can also shut down the server from here!

# Using the Search function to launch applications quickly

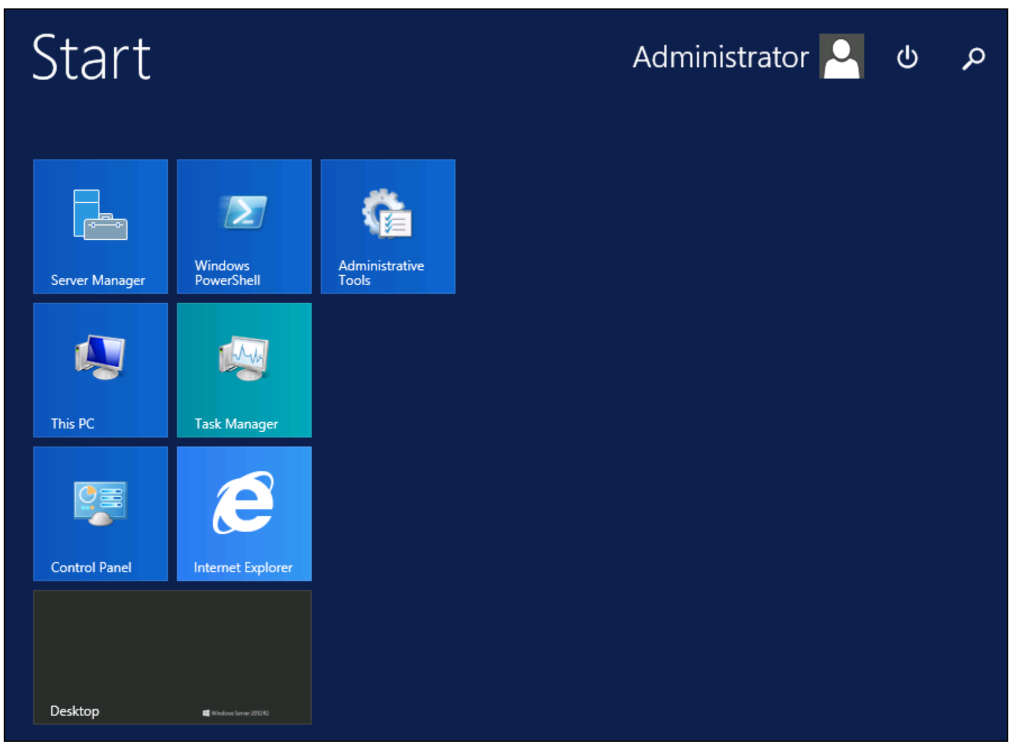
We are finally going to make some use of that **Start** screen. While many server administrators prefer to work outside of this screen 100 percent of the time, the powerful Search functions of the **Start** screen are definitely something that you should be aware of.

## Getting ready

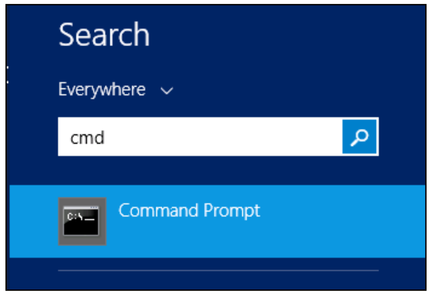
For this recipe, you will need a Windows Server 2012 R2 system online.

## How to do it…

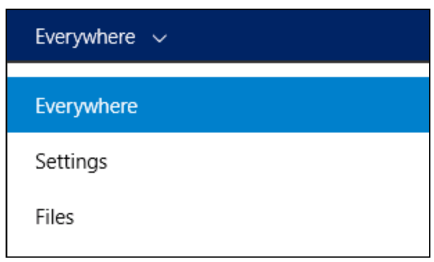
To search in Windows Server 2012 R2, we have to make our way to the **Start** screen. Fortunately, this is extremely easy to do. No matter where you are or what application you are in, if you press the Windows key or click on the **Start** button, you will be taken to the **Start** screen.



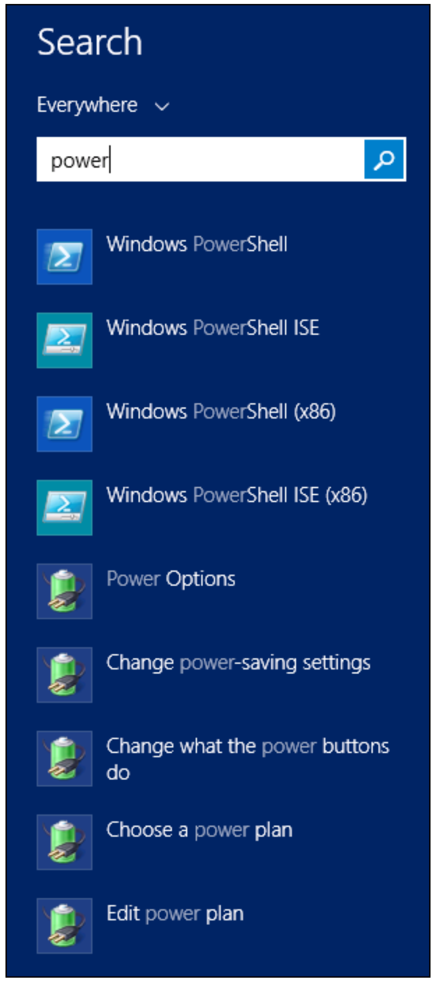
Now you simply start typing. Search results for anything found on the system that relates to the word you are inputting are automatically displayed. Once you see the item you are looking for, you can press *Enter* or click on it. Additionally, if you right-click on the item or hold down the *Ctrl* key while clicking it, you will receive additional options that you can perform with this application. For example, right-click or *Ctrl* click in order to launch command prompt as an administrator.




By default, your search will encompass the entire operating system. If you would rather narrow the scope of your search down to only **Settings** or **Files**, you can use the **Everywhere** dropdown and choose accordingly.



You can see it is very easy not only to open applications, but also its specific settings.



## How it works...

From the **Start** screen, we can search for anything on the server. This gives us the ability to quickly find and launch any program or application that we have installed. This includes Administrative Tools. Rather than moving into Server Manager in order to launch your administrative consoles from the **Tools** menu, you can also search for them on the **Search** screen, and launch from there. It also gives us the ability to find files or documents by name. Another powerful way to use the Search function in Windows Server 2012 R2 is to open any kind of setting that you might want to change. In previous versions of Windows, you had to either memorize the way to get into the settings that you wanted to change or you had to open up Control Panel where you had to poke and prod your way around until you stumbled upon the one that you were looking for. Now it is a very simple matter of pressing the Windows key, typing the first few characters of the setting or program you want to launch, and pressing *Enter*.

Another common task to perform from the **Search** screen is to right-click on the application that you are trying to launch and pin it somewhere. When you right-click on a program from the **Search** screen, you see options to pin the program to either your **Start** screen or to the Taskbar. This will create a quick-launch shortcut on either the main **Start** screen or on the Taskbar of the Desktop mode, giving you easier and faster access to launch those applications in the future.

# Managing remote servers from a single pane with Server Manager

As you have already noticed, Server Manager has changed significantly as of Server 2012, and has been made even better in Server 2012 R2. Part of these changes are a shift in mindset where emphasis is now placed on remote management of servers. Server Manager in Windows Server 2012 R2 can be used to manage and administer multiple systems at the same time, all from your single pane of glass, the monitor where you are sitting. In this recipe, you are going to learn how to manage both the local server we are logged into, as well as a remote server, from the same Server Manager window.

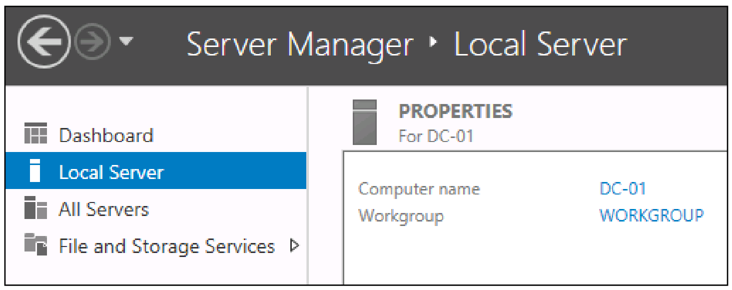
## Getting ready

For this recipe, we need two servers. One is the machine we are physically logged into. Another is a server on the same network that we can contact from our primary server, so that we can manage it from our local Server Manager.

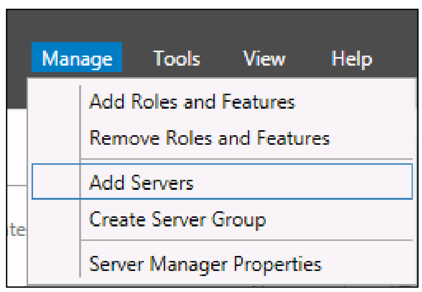
## How to do it...

To manage a local as well as a remote server from the same Server Manager window, perform the following instructions:

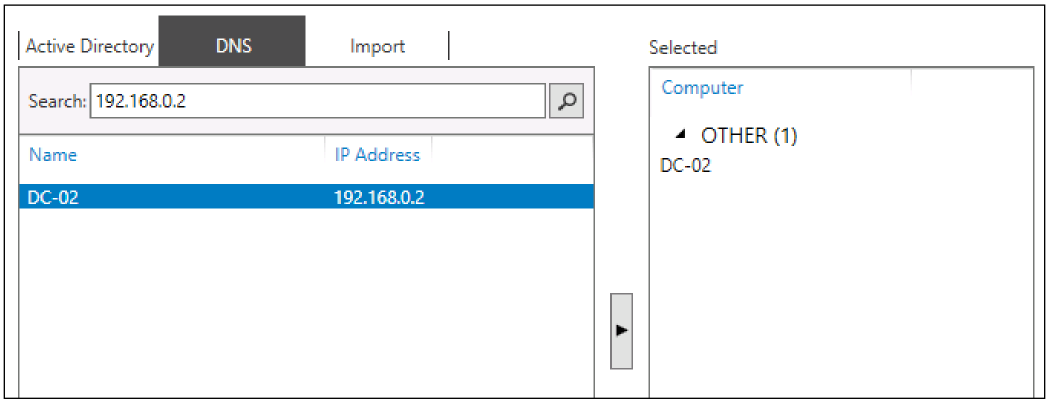
1. Log in to your primary server and launch **Server Manager**. You will see in the upper-left corner that the only server you have listed is the Local Server that we are logged into.



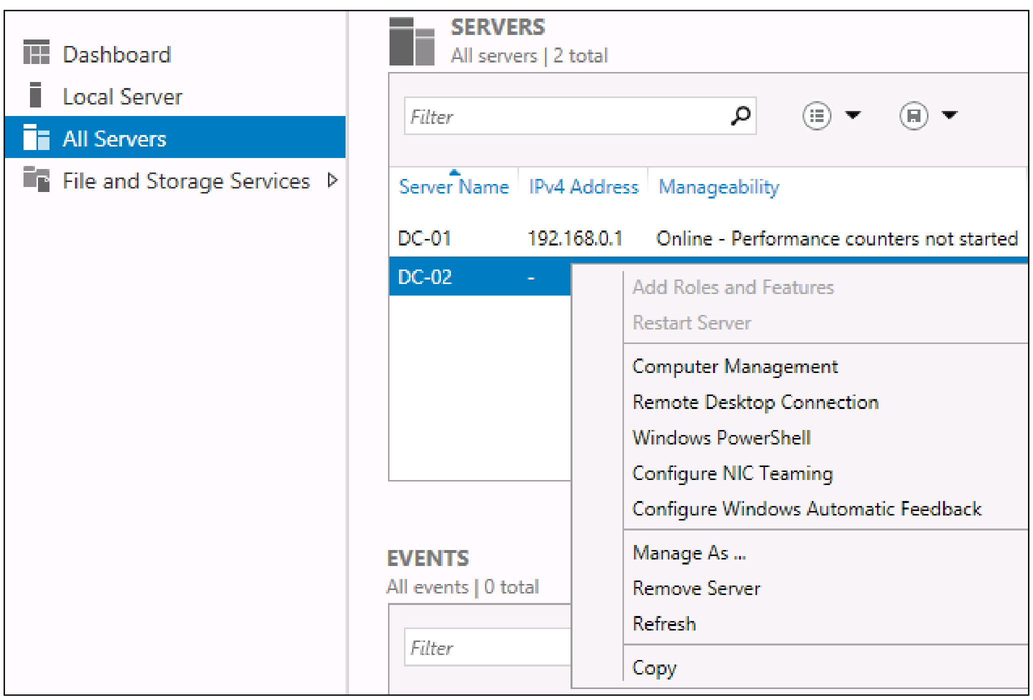

2. Now head over toward the top-right of Server Manager and click on the **Manage** button. In this menu, click on **Add Servers**.



3. If your servers are part of a domain, finding remote machines to manage is very easy by simply selecting them from the default **Active Directory** tab. Since mine are not yet part of a domain, I need to head over to the **DNS** tab and specify the name or IP address of the server that I want to connect to.




4. Now if you go ahead and click on **All Servers** in the left window pane, you will see the additional servers listed that you have selected. If you double-click or right-click on those remote server names, you have many options available to you to remotely manage those machines without having to log into them.



## How it works...

Server Manager makes use of the **Windows Remote Management** (**WinRM**) tools to remotely manipulate servers. Historically, most of us who are administering Windows Servers make extensive use of RDP, often having many windows and connections open simultaneously. This can cause confusion and can lead to tasks being accomplished on servers for which they are not intended. By using Server Manager from a single machine to manage multiple servers in your network, you will increase your administrative efficiency as well as minimize human error by having all management happen from a single pane of glass.

This recipe is written with the most common network scenario in mind, which is a domain environment where both servers have been joined to the domain. If you are working with standalone servers that are part of a workgroup, rather than being joined to a domain, you will have some additional considerations. In the workgroup scenario, WinRM will need to be enabled specifically, and the Windows Firewall will have to be adjusted in order to allow the right ports and protocols for that WinRM traffic flow to happen successfully. In general though, most of you will be working within a Microsoft domain network, in which case these items are not necessary.

## See also

- The *Administering Server 2012 R2 from a Windows 8.x machine* recipe

# Using PowerShell to accomplish any function in Windows Server

An incredibly powerful tool in Windows Server 2012 R2 is PowerShell. Think of PowerShell like a command prompt on steroids. It is a command-line interface from which you can manipulate almost anything inside Windows that you may care to. Better yet, any task that you may wish to accomplish can be scripted out in PowerShell and saved off as a `.ps1` script file, so that you can automate large tasks and schedule them for later, or at regular intervals. In this recipe, let's open up PowerShell and run some sample commands and tasks just to get a quick feel for the interface. In a later chapter of the book, we will do some more specific tasks with PowerShell to go even deeper into the technology.

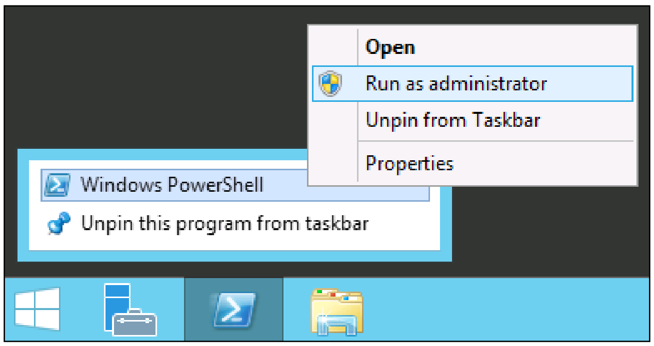
## Getting ready

To start using PowerShell, all you need is a server with Windows Server 2012 R2 installed. PowerShell is installed and enabled by default.

## How to do it...

To get a feel of using PowerShell, perform the following steps:

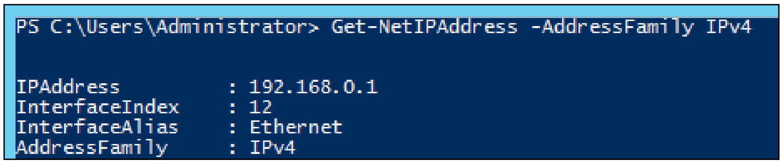
1. Right-click on your blue **PowerShell** icon and make sure to open PowerShell as administrator by clicking on **Run as administrator**. PowerShell is very powerful, but only if you have admin rights in the session.




2. Test out some commands that you are familiar with from using the command prompt, such as `dir` and `cls`. Since you are able to make use of these familiar commands, PowerShell can really be your one and only command-line interface if you choose.
3. Now let's try some of the PowerShell secret sauce, one of its **cmdlets**. These are special commands that are built into Windows and allow us to do all kinds of information gathering, as well as manipulation of server components. Let's start by pulling some data. Maybe take a look at what IP addresses are on the system with `Get-NetIPAddress`.

```
PS C:\Users\Administrator> Get-NetIPAddress

IPAddress         : fe80::c180:2ab4:d2ce:383c%12
InterfaceIndex    : 12
InterfaceAlias    : Ethernet
AddressFamily     : IPv6
```

4. The previous command probably gave you a lot more information than you needed, since most companies don't make use of `IPv6` inside their network yet. Let's whittle this information down to the `IPv4` specific info that you are most likely interested in. Enter `Get-NetIPAddress –AddressFamily IPv4` to attain it.




## How it works...

PowerShell has so many commands and cmdlets, we just wanted to get a feel for launching the program and pulling some data with this particular recipe. There are countless `Get` commands to query information from the server, and as you have seen those cmdlets have various parameters that can be appended to the cmdlets to pull more specific data to meet your needs. To make things even better, there are not only `Get` cmdlets, but also `Set` cmdlets, which will allow us to make use of the PowerShell prompt to configure many aspects of the configuration on our server, as well as remote servers. We will dive further into PowerShell in a later chapter.

# Installing a role or feature

You've installed the Windows Server 2012 R2 operating system onto a piece of hardware. Great! Now what? Without adding roles and features to your server, it makes a great paper weight. We're going to take the next steps here together. Let's install a role and a feature into Windows so that we can start making this server work for us.

## Getting ready

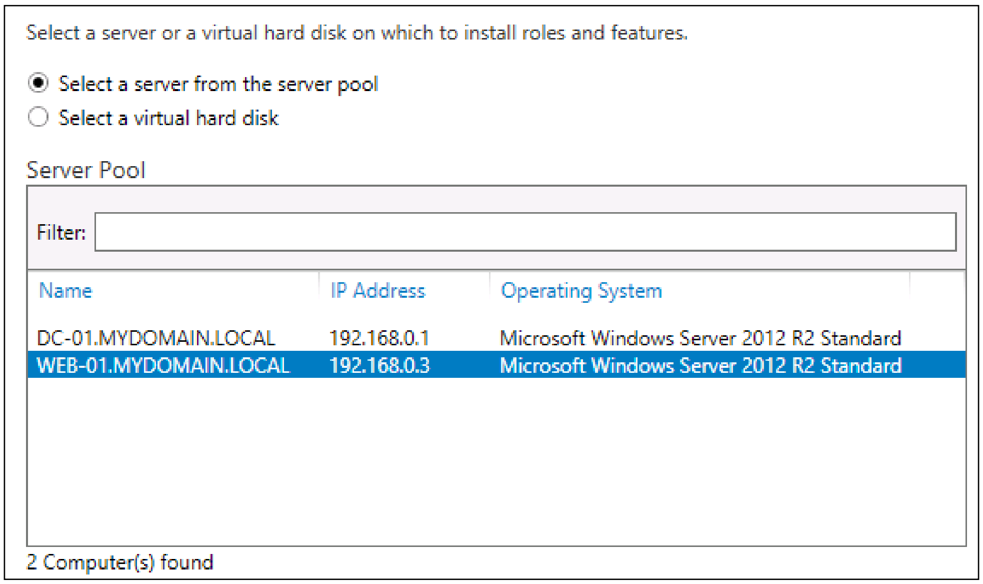
As long as you have a Windows Server 2012 R2 installed and running, you are ready to install roles and features onto that machine.

## How to do it...

To install a role and a feature into Windows, perform the following steps:

1. Open **Server Manager**. In the middle of the screen, you'll see a link that says **Add roles and features**. Click on that link.
2. Click **Next** on the first summary screen and you will come to a choice on the second page. For most roles and features, we want to leave it set at the top bullet, which is **Role-based or feature-based installation**. If we were configuring Remote Desktop Services, which we will discuss in another chapter, then we would choose the second option.

Now we choose where we want to install a new role or feature. This is a neat page, as we can choose from any server that we have added into our Server Manager, or we can even choose to install a role or feature into a virtual hard disk. I am running the **Add Roles Wizard** from DC-01, but I want to install the IIS role onto WEB-01. Rather than having to log into WEB-01 to accomplish this task, I will do it right from here. In the screenshot below, you can see WEB-01 listed as a server that I can install a role onto, even though I am opening this console on the DC-01 server.




3. Scroll down and choose the role that you want to install. For DC-01, I am choosing the **Web Server (IIS)** role. Then click **Next**.

[ You can install more than one role or feature at a time. ]

[ Some roles require additional components to be installed for them to work properly. For example, when I chose to install the IIS role and clicked **Next**, I was prompted about needing to install some management tools. Simply click on the **Add Features** button to automatically add the items that it needs to perform correctly. ]

4. Now choose any features that you would like to install. For example, in order to do some network connectivity testing later, go ahead and select **Telnet Client** from the list.
5. Read and click **Next** through the informational messages that are displayed. These messages will vary depending on which roles and features you have installed.
6. The final screen is your installation summary. If everything looks correct, go ahead and click on **Install**.

After your roles and features have finished installing, the server may or may not have to reboot. This depends on whether or not the role installation requires it. Following installation, or following the reboot, if your new role needs any additional configuration or setting up to be completed, you will be notified at the top of the Server Manager screen.

## How it works…

Adding roles and features to a Windows Server is something that every administrator will have to do sooner or later. These items are necessary to turn on the functions in the server that will perform tasks that need to be performed in your environment. Adding roles is quite straightforward. However, it is interesting to see the options that are available to add more than one role or feature at a time. Moreover, the ability to remotely install these items for servers in your network that you are not logged into is intriguing.

# Administering Server 2012 R2 from a Windows 8.x machine

In the *Managing remote servers from a single pane with Server Manager* recipe, we already discussed remotely administering another server by using Server Manager. Did you know we can accomplish the same remote management by using our day-to-day Windows 8.x computer? We will install and use the **Remote Server Administration Tools** (**RSAT**) to take even more advantage of Server 2012 R2's remote management ideology.

## Getting ready

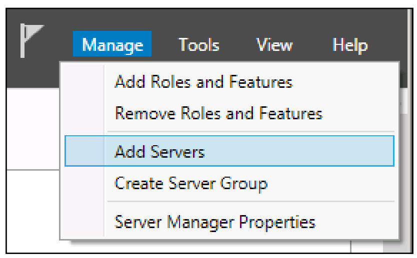
To test out the RSAT tools, we will need a Windows 8.x client machine. We will then also need a Windows Server 2012 R2 system online, and on the same network, which we can remotely control and manage.

## How to do it...

To remotely manage a server using RSAT, perform the following instructions:

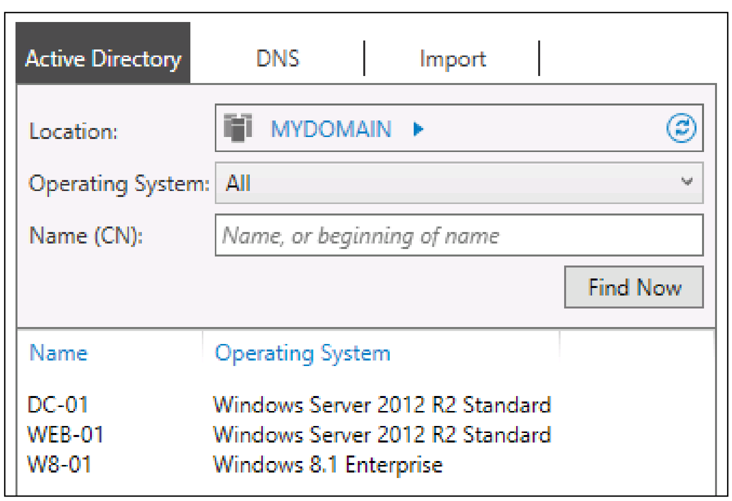
1. First we need to download the RSAT tools. You can use Bing to search for "Remote Server Administration Tools for Windows", or use this link to download RSAT for Windows 8.1: `http://www.microsoft.com/en-us/download/details.aspx?id=39296`. Here is also the link for the same RSAT tools in Windows 8.0 flavor: `http://www.microsoft.com/en-au/download/details.aspx?id=28972`.

   After you install these tools onto your Windows 8.x computer, you should now have a copy of Server Manager installed onto your computer. Go ahead and launch that from the **Start** screen. You can pin it to your Taskbar for quicker launching in the future, of course. In the same fashion, as with Server 2012 R2, you can use the **Manage** menu to add servers to Server Manager.

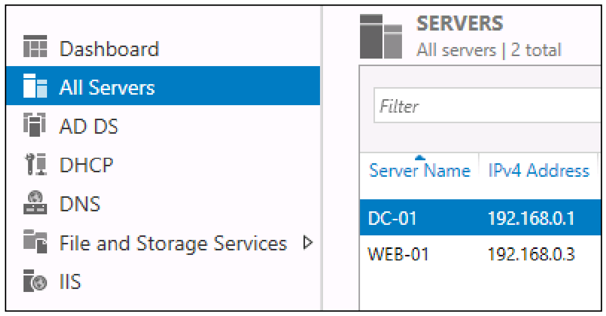
   
   

   For this recipe, I do have the machines we are working with joined to a domain, so we will take a look at adding servers that are part of the domain.

2. Click on the **Find Now** button and you will see a list of server names that are remotely manageable.

3. Click on the server names that you want to administer and click on the arrow to move them over to the right side of the screen. Upon clicking on **OK**, you will see these new servers listed and ready for management inside your Server Manager console.




## How it works...

Server Manager in Windows Server 2012 R2 is a powerful tool that can be used for the management of not only the local server, but also remote servers that you want to manage. If we take this even a step further and install the RSAT tools on a Windows 8 computer, this gives us the ability to launch and use Server Manager from our everyday Windows 8 computer. In doing so, we enable ourselves the ability to add roles, view events, and restart servers, all from our own desk. Managing servers using these tools will increase productivity and decrease errors, because your entire infrastructure of servers can be available within a single window. This is much more efficient than using the RDP client to connect to many different servers, all in different windows. If you've never tried using RSAT to manage servers, give it a try!

## See also

- The *Managing remote servers from a single pane with Server Manager* recipe

# Identifying useful keyboard shortcuts in Server 2012 R2

I prefer using a keyboard over a mouse any day, for almost any task. There are numerous keyboard shortcuts and tips and tricks that I employ on a daily basis and I want to test them out with you in this recipe. Some of these shortcuts have been around for years and will work with multiple versions of Windows Server; some are new in the Server 2012 R2 operating system. They will all be useful to you as you start working with servers in your network.

## Getting ready

We are going to run these commands and keyboard shortcuts while logged into a Windows Server 2012 R2 machine.

## How to do it...

- **Windows key**: Opens the **Start** screen where you can start typing to search for programs.
- **Windows key + X**: Opens the Quick Links menu, which we discussed in an earlier recipe.
- **Windows key + C**: Opens the main charms bar.
- **Windows key + I**: Opens the **Settings** charms bar specifically.
- **Windows key + D**: Shows or hides the Desktop. If you are in the **Start** screen, this is one way to get back over into the Desktop quickly.
- **Windows key + R**: Opens the **Run** box.
- **Windows key + M**: Minimizes all windows.
- **Windows key + E**: Opens File Explorer.
- **Windows key + L**: Locks the computer.
- **Alt + F4**: Exits the program you are currently working in. This is especially helpful in full screen apps launched from the Start screen, where it is not always obvious how to exit the program with your mouse.
- **Alt + Tab**: Displays a list of open programs so you can hop between them.
- **Shift + Delete**: Holding down *Shift* while pressing *Delete* deletes files without placing them into the Recycle Bin.
- **Using Tab inside command prompt or PowerShell**: I cannot believe that I went years without knowing about this one. When you are working inside command prompt, if you type the first letter of a file or folder that exists in the directory where you are working and then press the *Tab* key, it will auto-populate the rest of the filename. For example, you may be trying to launch a Microsoft update file with a filename that is 15 characters and comprises a mix of numbers and letters. No need to type out that filename! Let's say the file starts with `"KB"`. Simply navigate to the folder where your installer exists, type `"KB"`, and press *Tab*. The full filename is populated inside command prompt and you can press the *Enter* key to launch it.

### How it works...

Keyboard shortcuts can greatly increase productivity once you are fluent with them. This is not an extensive list by any means, there are many more key combinations that you can use to launch apps, minimize and maximize windows, and do all sorts of other functions. This is a list to get you started with the most common ones that I employ often. Start using these with your daily tasks and I bet your mouse will start to feel lonely.

If you are interested in exploring more of the Windows Server 2012 R2 key combinations available, this website is a great place to start: `http://technet.microsoft.com/en-us/library/hh831491.aspx`.

## Setting your PowerShell Execution Policy

To say that the Windows operating system can be manipulated by PowerShell is a gross understatement. They are fully intertwined, and PowerShell can be useful for so many tasks on your servers. However, the ability to run PowerShell scripts is disabled by default on many machines. The first stumbling block that many new PowerShell administrators bump into is the Execution Policy. It's quite simple, in order to allow PowerShell scripts to run on your server, the Execution Policy must be adjusted to allow that to happen. Let's introduce our first task in PowerShell by using some commands in this recipe that will set this policy for us.

This is also a good introduction to the idea of the verb-noun syntax that PowerShell utilizes. For example, we are going to make use of cmdlets called `Get-ExecutionPolicy` and `Set-ExecutionPolicy`. The `Get-(parameter name)` and `Set-(parameter name)` cmdlets are very common across all facets of cmdlets available in PowerShell. Wrap your mind around this verb-noun syntax and you will be well on your way to figuring out PowerShell on your machines.

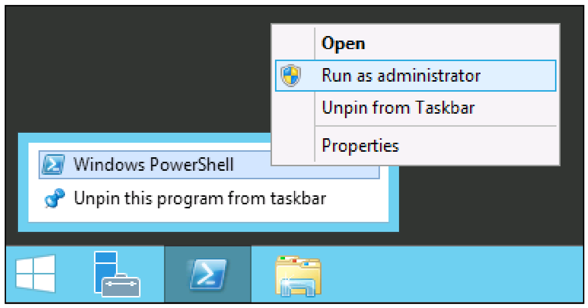
### Getting ready

We will be working within a PowerShell prompt on our Windows Server 2012 R2 box.

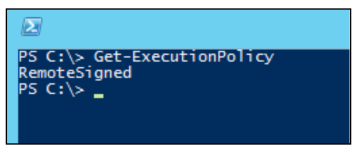
## How to do it...

Follow these steps to set the PowerShell Execution Policy:

1. Right-click on the **PowerShell** icon from the Taskbar and choose **Run as administrator**.

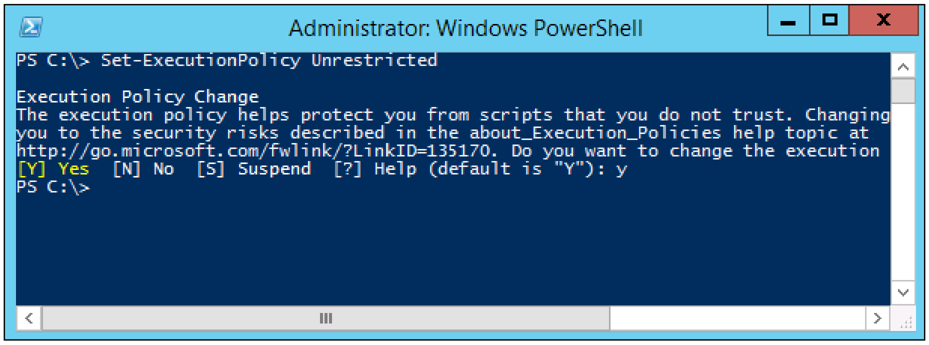



2. Type `Get-ExecutionPolicy` and press *Enter* in order to see the current setting of the PowerShell Execution Policy.




3. You can see that the current execution policy is set to **RemoteSigned**. Here is a short description of the different options for the policy:
    - **Remote Signed**: This is the default setting in Server 2012 R2, which allows PowerShell scripts that are locally created to run. If you try running remote scripts, they must be signed by a trusted publisher in order to execute successfully.
    - **All Signed**: With this setting, all scripts will only be allowed to run if they are signed by a trusted publisher.
    - **Restricted**: With this setting, PowerShell is locked down so that scripts will not run.
    - **Unrestricted**: This setting will allow PowerShell to run scripts, with or without signing.
4. For the purposes of our recipe and to make sure scripts will run for us as we progress through these recipes, let's set our Execution Policy to unrestricted. Go ahead and use this command:

```
Set-ExecutionPolicy Unrestricted
```




## How it works...

The PowerShell Execution Policy is a simple setting and easy to change, but can make a world of difference when it comes to running your first scripts. If configured to be more restrictive than you intend, you will have trouble getting your scripts to run and may think that you have mistyped something, when in fact the issue is only the policy. On the other hand, in an effort to make your servers as secure as possible, on machines where you don't need to execute PowerShell scripts, it makes sense to restrict this access. You may also want to read some additional information on the signing of scripts to see whether creating and executing signed scripts would make more sense in your own environment.

# Building and executing your first PowerShell script

Command prompt and PowerShell are both great command-line interfaces that can acquire and configure information about our servers. Most of us are familiar with creating some simple batch files that are driven by command prompt, essentially programming out small tasks within these batch files to automate a series of commands. This saves time later as we do not have to type out the commands line by line, especially for common tasks or for items that we need to run during login.

PowerShell has similar functionality, the ability to write out multiple lines of PowerShell cmdlets inside a script file. We can then launch this script file as we would a batch file, automating tasks while taking advantage of the additional features that PowerShell brings to the table over command prompt. These PowerShell scripts are put together inside `.ps1` files; let's build a simple one together to get a feel for running these scripts.

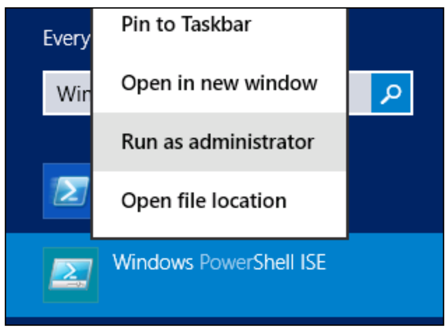
## Getting ready

Our work with PowerShell today will be accomplished from a Windows Server 2012 R2 machine. PowerShell is installed by default with Windows, and there is nothing further that we need to install.

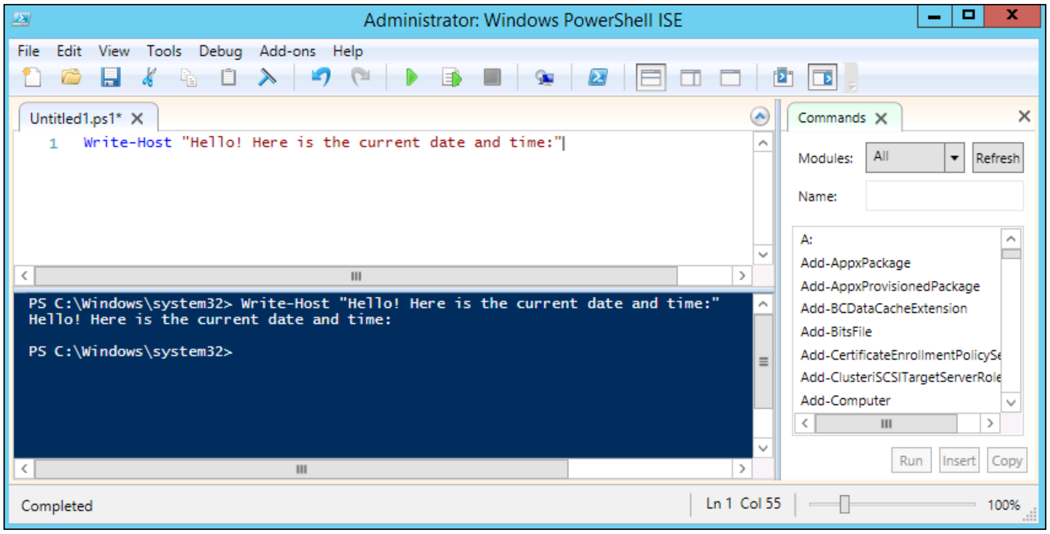
## How to do it...

Follow these steps to build and execute our first PowerShell script:

1. Open the **Start** screen and type `Windows PowerShell ISE`. Right-click to launch this tool as an administrator. PowerShell ISE is an editor for PowerShell scripts that is much more useful than opening a simple text editor such as Notepad in order to build our script.




2. Choose **File** | **New** from the menus in order to open a blank `.ps1` script file.
3. In your first line, type the following: `Write-Host "Hello! Here is the current date and time:"`.
4. From the toolbar menu, click the green arrow that says **Run Script**. Alternatively, you can simply press the *F5* button. When you run the script, the command and output are displayed in the lower portion of the ISE window.

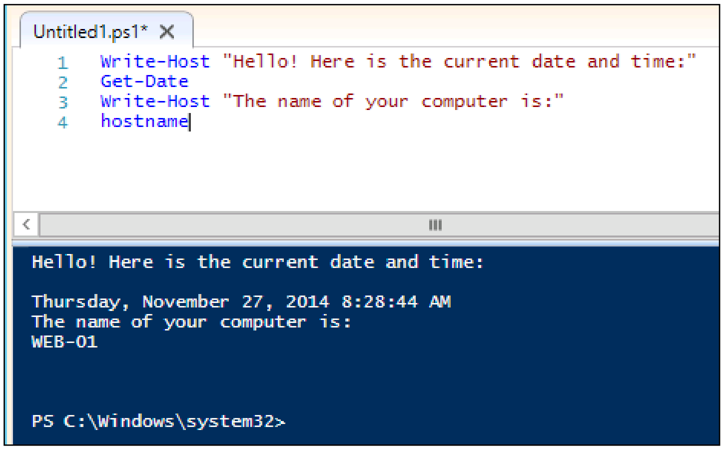



Cool! Okay, so far it's actually pretty lame. Just reflecting the text that we told it to echo, but it worked. That is the nice thing about using the ISE editing tool rather than a generic text editor, you have the ability to quickly test run scripts as you make modifications.

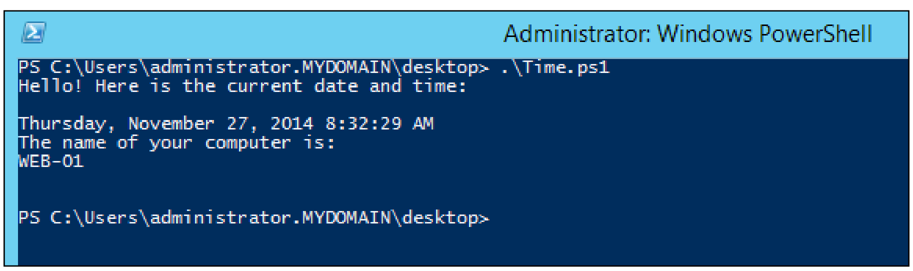
5. Now let's add some additional lines into our script to give us the information we are looking for. You can see a list of available commands on the right side of the screen if you would like to browse through what is available, but for our example simply change your script to include the following:

```
Write-Host "Hello! Here is the current date and time:"
Get-Date
Write-Host "The name of your computer is:"
hostname
```

6. Press the **Run Script** button again to see the new output.




7. Now use **File** | **Save** and save your new `.ps1` PowerShell script out to the Desktop.
8. Let's test this script by launching it from within a real PowerShell command window. Right-click on your PowerShell icon in the Taskbar and choose **Run as administrator**.
9. Browse to the location of the script file, I placed mine on the Desktop. Then launch the script by inputting `.\filename`. In my case, it looks like this: `.\time.ps1`.




## How it works...

In this recipe, we created a very simple PowerShell script and saved it on our server for execution. While in practice getting time and date information from your server may come faster by using the stand-alone `Get-Date` cmdlet, we use this recipe to give a small taste of the ISE and to get your scripting juices flowing. Expanding upon the ideas presented here will start to save you valuable time and keystrokes as you identify more and more ways to automate the tasks and information gathering that are part of your daily routines. The possibilities behind PowerShell are practically limitless, so make sure that you open it up and start becoming familiar with the interfaces and tools associated with it right away!

# Searching for PowerShell cmdlets with Get-Help

With this recipe, let's take a minute to use `Get-Help` inside PowerShell in order to, well, get some help! I see both new and experienced PowerShell administrators going to the Web a lot in order to find commands and the parameters of those commands. The Internet is great, and there is a ton of data out there about how to use PowerShell, but in many cases the information that you are looking for resides right inside PowerShell itself. By using the `Get-Help` cmdlet combined with the functions you are running or searching for, you might not have to open that web browser after all.

## Getting ready

We will be running some commands from inside PowerShell on a Windows Server 2012 R2 machine.

## How to do it...

To use the `Get-Help` function inside PowerShell, run the following steps:

1. Launch a PowerShell prompt.
2. Type `Get-Help`.
3. You're finished! No, I'm just kidding. Using `Get-Help` by itself will present you with some helpful data about the `Get-Help` command, but that's not really what we are looking for, is it? How about using `Get-Help` with a search parameter, like this:

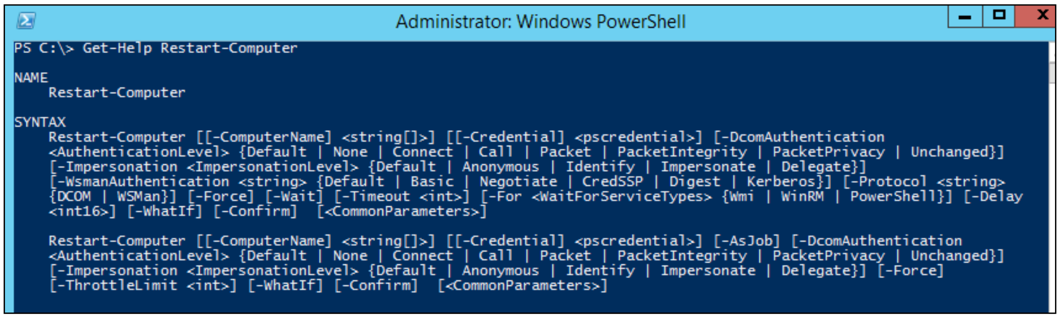
   **`Get-Help Computer`**

```
PS C:\> Get-Help Computer

Name                              Category  Module                    Synopsis
----                              --------  ------                    --------
Add-Computer                      Cmdlet    Microsoft.PowerShell.M... ...
Checkpoint-Computer               Cmdlet    Microsoft.PowerShell.M... ...
Disable-ComputerRestore           Cmdlet    Microsoft.PowerShell.M... ...
Enable-ComputerRestore            Cmdlet    Microsoft.PowerShell.M... ...
Get-ComputerRestorePoint          Cmdlet    Microsoft.PowerShell.M... ...
Remove-Computer                   Cmdlet    Microsoft.PowerShell.M... ...
Rename-Computer                   Cmdlet    Microsoft.PowerShell.M... ...
Reset-ComputerMachinePassword     Cmdlet    Microsoft.PowerShell.M... ...
Restart-Computer                  Cmdlet    Microsoft.PowerShell.M... ...
Restore-Computer                  Cmdlet    Microsoft.PowerShell.M... ...
Stop-Computer                     Cmdlet    Microsoft.PowerShell.M... ...
Test-ComputerSecureChannel        Cmdlet    Microsoft.PowerShell.M... ...
Get-SilComputer                   Function  SoftwareInventoryLogging  ...


PS C:\> _
```

Cool! That searched the available cmdlets and presented us with a list of the ones that contain the word `Computer`. Nice.

4. Now what if we wanted to find out some more particular information about one of these cmdlets? Maybe about `Restart-Computer`, that sounds like something we might use often. Use the following command:

   ```
   Get-Help Restart-Computer
   ```




Now we're really cooking! This is wonderful information. Basically, this is exactly what you would find if you were looking for information about the `Restart-Computer` cmdlet and went searching on TechNet for it.

## How it works...

The `Get-Help` cmdlet in PowerShell can be used with virtually any command in order to find out more information about that particular function. I often use it when the specific name of a cmdlet that I want to use escapes my memory. By using `Get-Help` as a search function, it will present a list of available cmdlets that include the keyword you specified. This is a brilliant addition to PowerShell, and makes it so much more powerful than command prompt.

Also included with the `Get-Help` files are all of the special syntax and parameter options for each cmdlet that you might be working with. This saves you time from having to go to the Web in order to search for these functions, and it is just way more fun doing it at the command line than in a web browser.

# 2
# Core Infrastructure Tasks

Windows Server 2012 R2 has many roles and features that can be used to accomplish all sorts of different tasks in your network. This chapter reflects on the most common infrastructure tasks needed to create a successful Windows Active Directory environment by using Server 2012 R2:

- Configuring a combination Domain Controller, DNS server, and DHCP server
- Adding a second Domain Controller
- Organizing your computers with Organizational Units (OUs)
- Creating an A or AAAA record in DNS
- Creating and using a CNAME record in DNS
- Creating a DHCP scope to assign addresses to computers
- Creating a DHCP reservation for a specific server or resource
- Pre-staging a computer account in Active Directory
- Using PowerShell to create a new Active Directory user
- Using PowerShell to view system uptime

## Introduction

There are a number of technologies in Windows Server 2012 R2 that are *need to know* if you plan to ever work in a Windows environment. These are technologies such as **Active Directory Domain Services** (**AD DS**), **Domain Name System** (**DNS**), and **Dynamic Host Configuration Protocol** (**DHCP**). If you haven't noticed already, everything in the Windows world has an acronym. In fact, you may only recognize these items by their acronyms, and that's okay.

Nobody calls DHCP the Dynamic Host Configuration Protocol anyway. But do you know how to build these services and bring a Windows Server infrastructure online from scratch, with only a piece of hardware and a Windows Server 2012 R2 installation disc to guide your way? This is why we are here today. I would like to instruct you on taking your first server and turning it into everything that you need to run a Microsoft network.

Every company and network is different and has different requirements. Some will get by with a single server to host a myriad of roles, while others have thousands of servers at their disposal and will have every role split up into clusters of servers, each of which has a single purpose in life. Whatever your situation, this will get us back to the basics on setting up the core infrastructure technologies that are needed in any Microsoft-centric network.

# Configuring a combination Domain Controller, DNS server, and DHCP server

The directory structure that Microsoft networks use to house their users and computer accounts is called **Active Directory** (**AD**), and the directory information is controlled and managed by **Domain Controller** (**DC**) servers. Two other server roles that almost always go hand in hand with Active Directory are DNS and DHCP, and in many networks these three roles are combined on each server where they reside. A lot of small businesses have always made due with a single server containing all three of these roles, but in recent years virtualization has become so easy that almost everyone runs at least two Domain Controllers, for redundancy purposes. And if you are going to have two DCs, you may as well put the DNS and DHCP roles on them both to make those services redundant as well. But I'm getting ahead of myself. For this recipe, let's get started building these services by installing the roles and configuring them for the first time: The first Domain Controller / DNS server / DHCP server in our network.

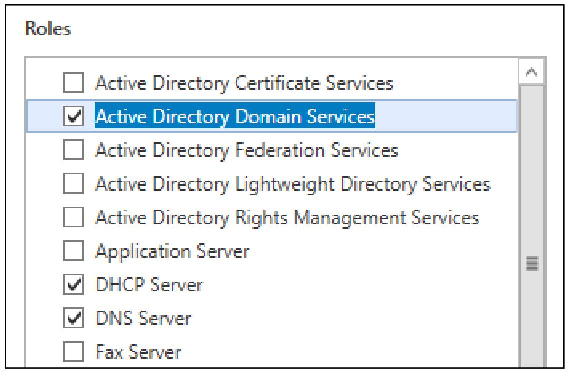
## Getting ready

The only prerequisite here is a Windows Server 2012 R2 online that we can use. We do want it to be plugged into a network and have a static IP address assigned so that as you add new computers to this network, they have a way of communicating with the domain we are about to create.

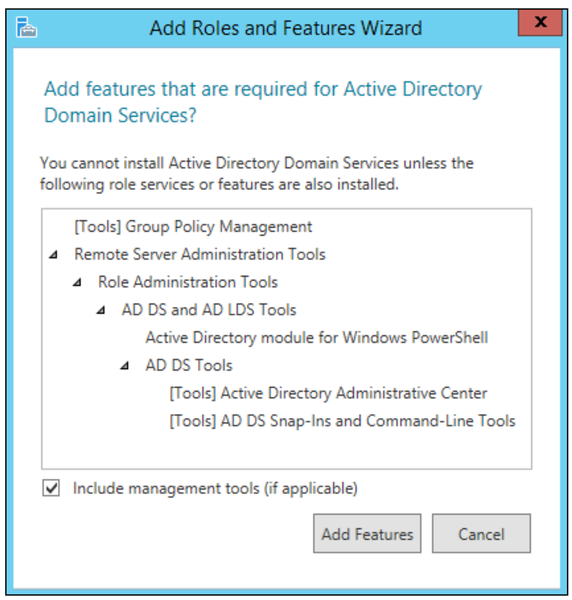
## How to do it...

Let's configure our first Domain Controller / DNS server / DHCP server by performing the following set of instructions:

1. Add the roles all at once. To do this, open up **Server Manager** and click on your link to add some new roles to this server. Now check all three: **Active Directory Domain Services**, **DHCP Server**, and **DNS Server**.

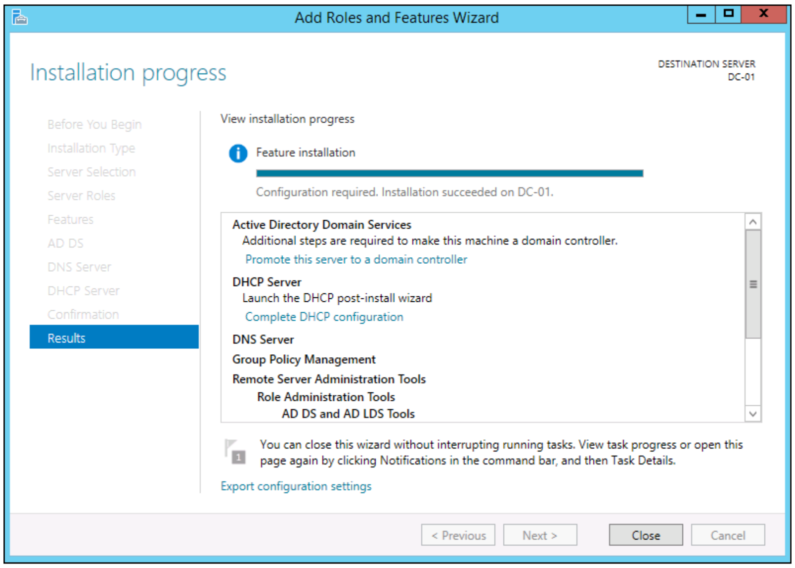



2. When you click on Active Directory, you will be prompted whether you want to install some supporting items. Go ahead and click on the **Add Features** button to allow this.

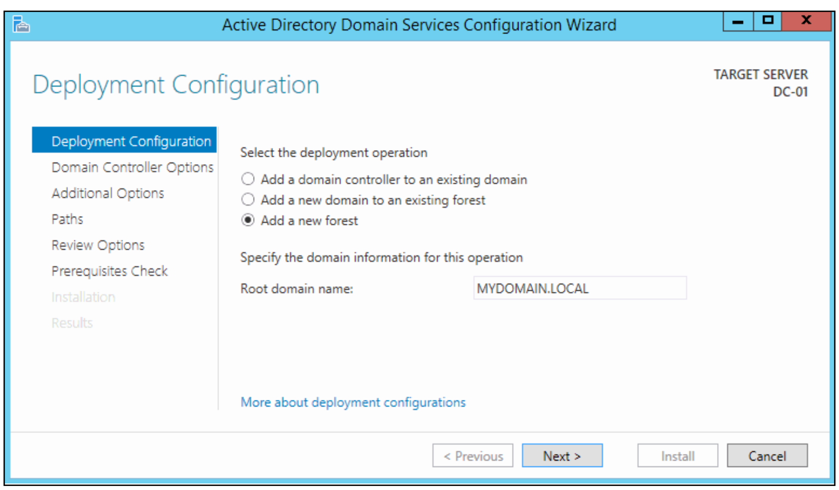



3. You are going to click **Next** through the following few screens. We don't have to add any additional features, so you can read and click through the informational screens that tell you about these new roles.

4. Once satisfied with the installation summary, press the **Install** button on the last page of the wizard.
5. Following installation, your progress summary screen shows a window with a couple of links on it. They are **Promote this server to a domain controller** and **Complete DHCP configuration**. We are going to click on the first link to promote this machine to be a DC.




6. Now we are taken into the configuration of our Domain Controller. Since this is the very first DC in our entire network, we choose the option **Add a new forest**. At this point we also have to specify a name for our root domain.



> It is very important to choose a root domain name that you like and that makes sense for your installation. Whatever you enter here will more than likely be your domain name forever and always!!

This might be a good opportunity for a little side-bar of definitions and explanations. You can think of a **forest** as the top level of your Active Directory structure. Within that forest, you are setting up a **domain**, which is the container within your forest that contains your user, computer, and other accounts that will be joined to the domain. You can contain multiple domains within a forest, and multiple forests can share information and talk to each other by using something called a trust.

You can see above that I have named my domain `MYDOMAIN.LOCAL`. The `.local` is important to discuss for a minute. It is really just a common specification that many companies use to clarify that this domain is an internal network, not a public one. However, I could have just as easily named it `CONTOSO.COM`, or `JORDAN.PRIV`, or many different things.

Another practice that I see often is for companies to use the same domain name inside their network as they do publically. So basically whatever their website ends in, that is their public domain name. You could certainly set up the internal domain name to be the same. This practice is commonly referred to as "**split-brain DNS**". It used to be something that Microsoft warned against doing, but many companies do it this way, and all of the technology has evolved around this so that the Microsoft networking parts and pieces will all work just fine with split-brain DNS these days, though it does usually take additional consideration when setting up any new piece of technology.

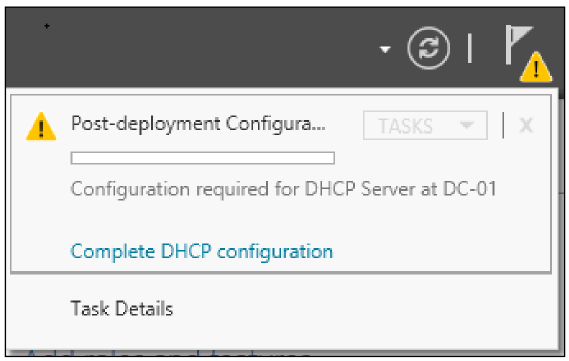
> Once last important note: it is **not** recommended to set up your domain as a single label name, for example, if I had called it just `MYDOMAIN`. While this is technically possible, it presents many problems down the road and is not recommended by Microsoft.

7. On the **Domain Controller Options** screen, you can choose to lower the functional level of your forest or domain, but this is not recommended unless you have a specific reason to do so. You must also specify a DSRM password on this screen in case it is ever needed for recovery.

   You will receive a **DNS Options** warning message on the next page. This is normal because we are turning on the first Domain Controller and DNS server in our environment.

8. The following two screens for **NetBIOS** and **Paths** can be left as the defaults, unless you have a reason to change their settings.

9. Once you have reviewed the installation plan, go for it! There may be some informational and warning messages that show themselves, but you should see a green check mark telling you **All prerequisite checks passed successfully**, which means you are ready to proceed. When the server is finished being promoted to a Domain Controller, it will have to restart.
10. Following the restart, you will have noticed that you are now forced to log in to the server as a domain account. Once a server has been promoted to a Domain Controller, it no longer contains local user accounts on the system. All logins to the server from this point forward will have to be user accounts within the domain. Go ahead and login as such.
11. Inside Server Manager you will have a notification up top to **Complete DHCP configuration**. Go ahead and click on that.




12. You don't have to specify anything in this wizard. Simply click through the steps.

## How it works...

Configuring your first Domain Controller is essential to having a successful Microsoft Windows network. Now that the roles are installed for AD, DNS, and DHCP, we have the core infrastructure in place to start joining computers to the domain, adding users to the network, and shuttling around some network traffic! Each of these technologies has enough depth to warrant their own books, so there is no way that we can cover everything here. I hope that this tutorial will get you comfortable with enabling these system critical functions in your own network. Having the ability to create a network from scratch is priceless ammunition to a server administrator.

## See also

It is also possible to install Active Directory on your Domain Controllers through the use of PowerShell. Since we are discussing the use of PowerShell throughout this book to start utilizing it for some day-to-day tasks, make sure to check out the following links and try doing it this way on the next DC that you want to create:

- `http://technet.microsoft.com/en-us/library/hh974719.aspx`
- `http://technet.microsoft.com/en-us/library/hh472162.aspx#BKMK_PS`

# Adding a second Domain Controller

Active Directory is the core of your network. It has ties in everything! As such, it makes sense that you would want this to be as redundant as possible. In Windows Server 2012 R2, creating a secondary Domain Controller is so easy that you really have no reason not to do it. Can you imagine rebuilding your directory following a single server hardware failure where you have 100 user accounts and computers that are all part of the domain that just failed? That could take weeks to clean up, and you'll probably never get it back exactly the way it was before. Additionally, while you are stuck in the middle of this downtime, you will have all kinds of trouble inside your network since your user and computer accounts are relying on Active Directory, which would then be offline. Here are the steps to take a second server in your network and join it to the existing domain that is running on the primary DC to create our redundant, secondary Domain Controller.

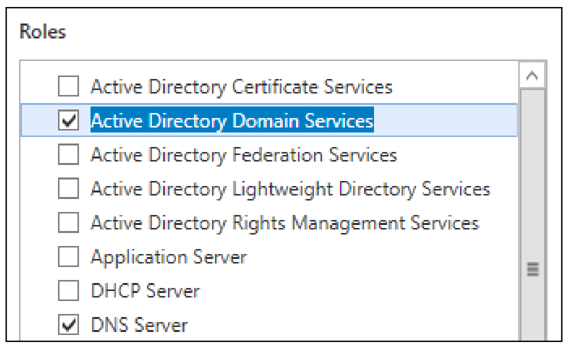
## Getting ready

Two Server 2012 R2 machines are needed for this. The first we will assume is running Active Directory and DNS already, like the one we set up in our previous recipe. The second server is online, plugged into the same network, and has been named `DC-02`.

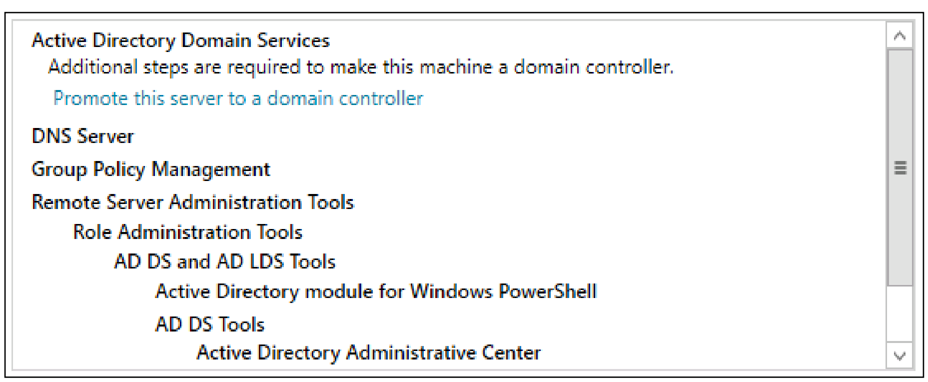
## How to do it...

To create a redundant secondary Domain Controller, perform the following steps:

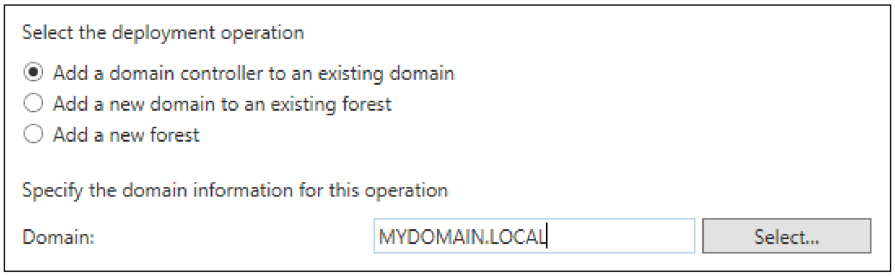
1. Open **Server Manager** on DC-02 and click the link to **Add roles and features**.
2. Click **Next** a few times until you get to the screen where we are selecting the role that we want to install. Let's choose both **Active Directory Domain Services** and **DNS Server**. It is very common for each Domain Controller to also run DNS, so that you have redundancy for both services. Both of these roles will prompt for additional features, so make sure you press the **Add Features** button when it prompts you to allow the installation of those extra components.




3. We do not require any other features so click **Next** through the remaining screens and then click on **Install** on the last page.
4. Once installation is finished, you have a link to click on that says **Promote this server to a domain controller**. Go ahead and click on that link.



5. For this second Domain Controller, we are going to choose the option **Add a domain controller to an existing domain**. Then in the **Domain** field, specify the name of the domain that is running on your primary DC. You must also specify a domain user account in the credentials field to validate against the domain.




> If you receive an error message that a Domain Controller for the domain could not be contacted, you probably haven't specified a DNS address in your TCP/IP settings. Add your primary DC's IP address in as your primary DNS server and it should work.

6. The rest of the steps reflect the same options we chose when creating our first Domain Controller in the previous recipe. Once you are finished stepping through the wizard, you will have a secondary Domain Controller and DNS server online and running.

## How it works...

Creating redundancy for Active Directory is critical to the success of your network. Hardware fails, we all know it. A good practice for any company is to run two Domain Controllers, so that everyone continues to work in the event of a server failure. An even better practice is to take this a step further and create more DCs, some of them in different sites perhaps, and maybe even make use of some **Read-Only Domain Controllers** (**RODC**) in your smaller, less secure sites. See the following link for some additional information on using an RODC in your environment, visit `http://technet.microsoft.com/en-us/library/cc754719(v=ws.10).aspx`

# Organizing your computers with Organizational Units (OUs)

Active Directory is the structure in which all of your user, computer, and server accounts reside. As you add new users and computers into your domain, they will be automatically placed into generic storage containers. You could get away with leaving all of your objects in their default locations, but there are a lot of advantages to putting a little time and effort into creating an organizational structure.

In this recipe, we will create some **Organizational Units** (**OUs**) inside Active Directory and move around our existing objects into these OUs so that we can create some structure.

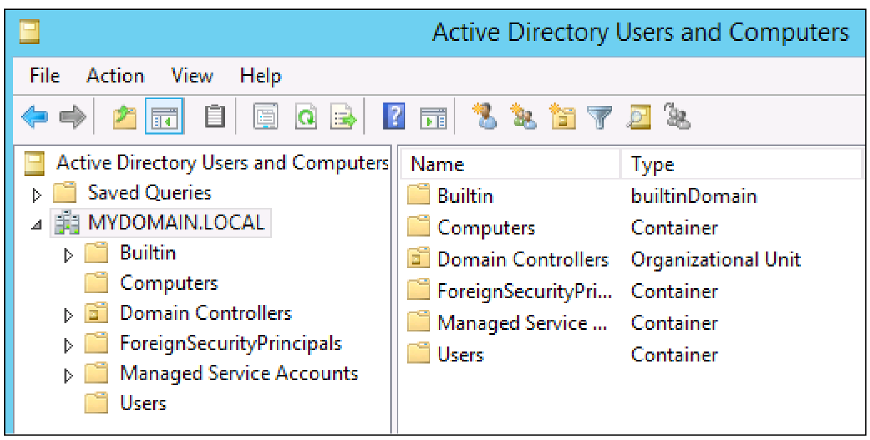
## Getting ready

We will need a Domain Controller online for this recipe, which is a Server 2012 R2 machine with the Active Directory Domain Services role installed. Specifically, I will be using the DC-01 server that we prepped in the *Configuring a combination Domain Controller, DNS server, and DHCP server* recipe.

## How to do it...

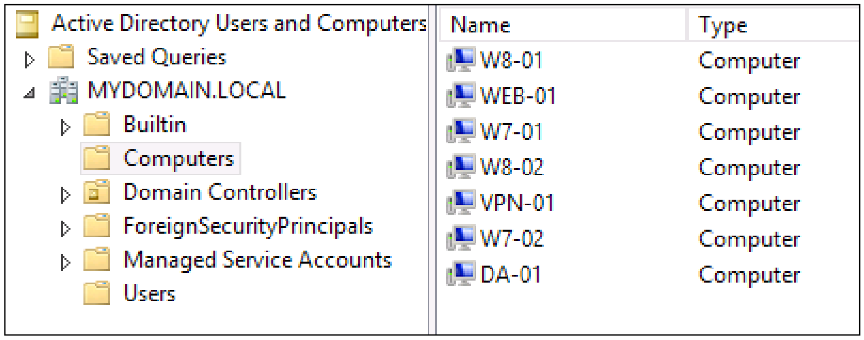
Let's get comfortable working with OUs by creating some of our own as follows:

1. Open **Active Directory Users and Computers**. This can be launched from the **Tools** menu inside Server Manager. As you can see, there are some pre-defined containers and OUs in here:

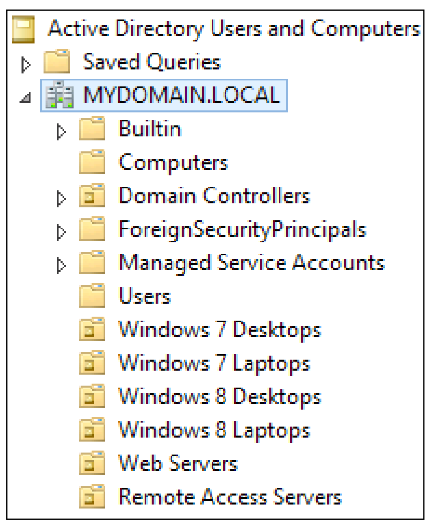



> Alternatively, you can also open **Active Directory Users and Computers** by running `dsa.msc` from a prompt or the **Start** screen.

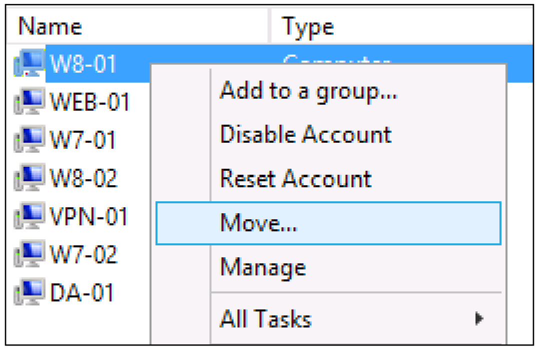
2. We can already see that the Domain Controller servers have been segmented off into their own OU. If we look in our `Computers` folder, however, we can see that currently all of the other systems we have joined to the domain have been lumped together:




3. Currently it's hard to tell what machine accomplishes what purpose. A better naming scheme might help, but what if you are working in an environment where there are hundreds of objects in here already? We want to break these machines up into appropriate groups, so that we have better management over them in the future. Right-click on the name of your domain in the left window pane, then click **New** | **Organizational Unit**.
4. Input a name for your new OU and click **OK**. I am going to create a few new OUs, called **Windows 7 Desktops**, **Windows 7 Laptops**, **Windows 8 Desktops**, **Windows 8 Laptops**, **Web Servers**, and **Remote Access Servers**.



5. Now for each object that you want to move, simply find it, right-click on it, and then click on **Move...**.




6. Choose which OU you would like this object to move into and click **OK**.

## How it works...

The actual work involved with creating OUs and moving objects around between them isn't complicated at all. What is much more important about this recipe is prompting you to think about what way works best for you to set up these OUs to make the best organizational sense for your environment. By breaking our computer accounts out into pinpointed groups, we are able in the future to easily do things such as discover how many web servers we have running, or do some quick reporting on how many user accounts we have in the sales group. We could even apply different Group Policy settings to different computer sets based on what OU they are contained within. Both reporting and applying settings can be greatly improved upon by making good use of Organizational Units inside Active Directory.

# Creating an A or AAAA record in DNS

Most folks working in IT are familiar with using the `ping` command to test network connectivity. If you are trying to test the connection between your computer and another, you can ping it from a command prompt and test whether or not it replies. This assumes that the firewalls in your computers and network allow the ping to respond correctly, which generally is true. If you are inside a domain network and ping a device by its name, that name resolves to an IP address, which is the device's address on the network. But what tells your computer which IP address corresponds to which name? This is where DNS comes in. Any time that your computer makes a request for a name, whether it is you pinging another computer or your Outlook email client requesting the name of your Exchange Server, your computer always reaches out to your network's DNS servers and asks, "How do I get to this name?".

DNS contains a list of records that tell the computers in your network what IP addresses correspond to what names. By far the most common type of DNS record is called a **Host** record. When the Host record resolves to an IPv4 address, like 192.168.0.1, it is called an **A record**. When the Host record resolves to an IPv6 address, like 2003:836b:2:8100::2, it is called a **AAAA record**. This is usually pronounced *quad A*.

Understanding how to create and troubleshoot host records in DNS is something that every Windows server administrator needs to know. Let's take a minute to create and test one of these DNS records so that we can experience firsthand how this all works together.

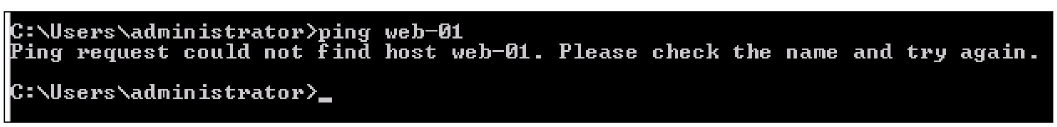
## Getting ready

We have a Domain Controller online, which also has the DNS role installed. This is all we need to create the DNS record, but we will also make use of a Windows 8 client computer and a web server to do the name resolution testing.

## How to do it...

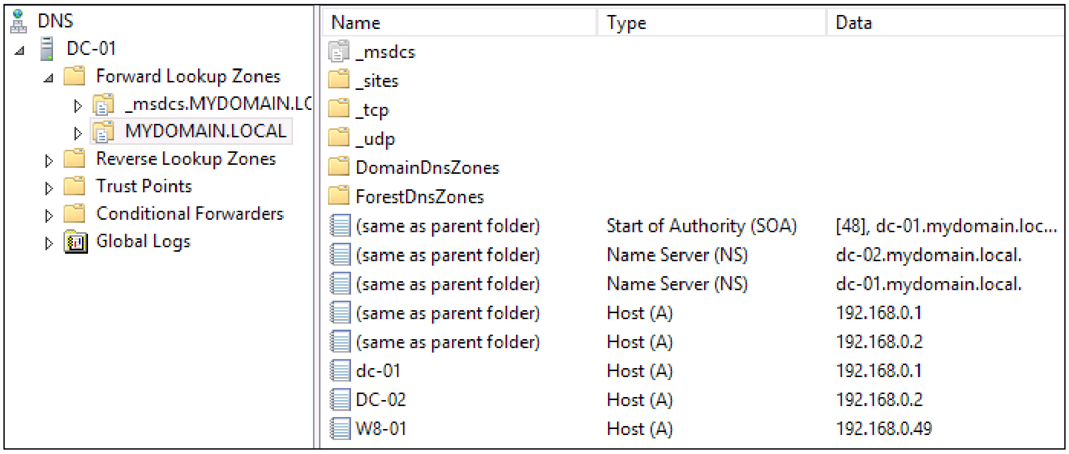
To create and test a DNS record, perform these steps:

1. There is a new web server plugged into the network, but it is not yet joined to the domain and so it has not been registered into DNS. The name of this web server is `web-01`. Open up **Command Prompt**, and type `ping web-01`. As expected, because there is no host record in DNS for this server yet, our ping request does not resolve to anything.

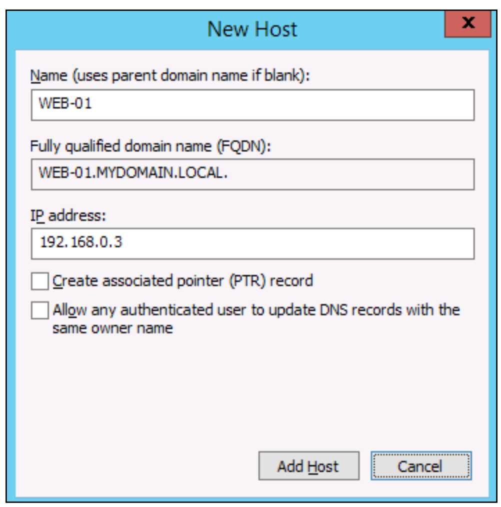



2. Now head into the DNS server and open up the **DNS** console from the **Tools** menu.

3. Inside **Forward Lookup Zones**, you should see your domain listed. Double-click on the name of your domain to see your existing DNS records.




4. Right-click on your domain, then click on **New Host (A or AAAA)...**.
5. Input the server name into the top field and the IP address where it is running into the bottom field. Then click **Add Host**.




If you are running IPv6 in your network and want to create a AAAA record instead, you use this exact same process. Simply enter the IPv6 address into the IP address field, instead of the IPv4 address.

6. Now that our new host record has been created, let's test it out! Going back to our client computer, type `ping web-01` again. You will see your output as shown in the following screenshot:

```
C:\Users\administrator>ping web-01

Pinging web-01.MYDOMAIN.LOCAL [192.168.0.3] with 32 bytes of data:
Reply from 192.168.0.3: bytes=32 time=1ms TTL=128
Reply from 192.168.0.3: bytes=32 time<1ms TTL=128
Reply from 192.168.0.3: bytes=32 time<1ms TTL=128
Reply from 192.168.0.3: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\Users\administrator>_
```

## How it works...

Any time a computer in a domain network requests to communicate with a host name, DNS is the party responsible for pointing it in the right direction. If you or your applications are having trouble contacting the servers they need, this is one of the first places you will want to look into. Understanding DNS host records is something that will be necessary when working with any networking technology. In this recipe, we only talked about the most common form of DNS record, but there are others you may want to learn and test as well. In fact, take a look at our next recipe for information on another useful type of DNS record, the CNAME.

There are a couple of other name resolution functions in the Windows operating system that may cause resolution to happen before a hostname request gets to the DNS server. For example, if someone has created a static name and IP record inside a client computer's hosts file, it will resolve to that specified IP address, no matter what is in the DNS server. This is because the hosts file has priority over DNS. Also, there is a special table called the Name Resolution Policy Table (NRPT) that is used by DirectAccess client computers, and it works in a similar way. Name resolution requests pass through the hosts file and through the NRPT before making their way to DNS. If one of the former tables has an entry for the name that is being requested, they will resolve it before the computer sends the request to the DNS server for resolution. So if you are troubleshooting a name that doesn't resolve properly, keep those additional items in mind when looking for the answer to your problem.

## See also

- The *Creating and using a CNAME record in DNS* recipe

# Creating and using a CNAME record in DNS

Now that we are familiar with moving around a little bit inside the DNS management tool, we are going to create and test another type of record. This one is called a CNAME, and it is easiest to think of this one as an alias record. Rather than taking a DNS name and pointing it at an IP address as we do with a host record, with a CNAME we are going to take a DNS name and point it at another DNS name! Why would this be necessary? If you are hosting multiple services on a single server but want those services to be contacted by using different names, CNAME records can be your best friend.

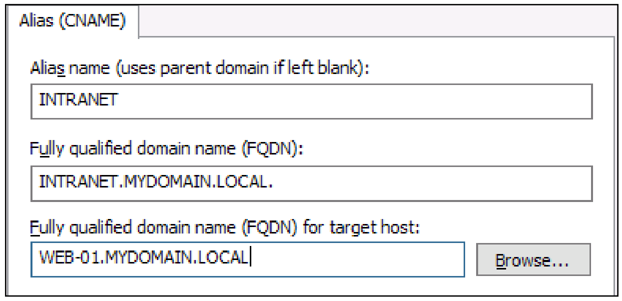
## Getting ready

We are going to make use of the same environment that we used to create our *A* records in the *Creating an A or AAAA record in DNS* recipe. There is a Domain Controller / DNS server online where we are going to create our records. Also running is WEB-01, a server where we are hosting a website as well as some file shares. We will also use a Windows 8 client to test out our CNAME records after they have been created.

## How to do it...

To create and test a CNAME record, perform the following instructions:

1. WEB-01 is hosting a website and a file share. Currently the only DNS record that exists for WEB-01 is the primary *A* record, so users have to type in the WEB-01 name to access both the website and the file shares. Our goal is to create aliases for these services by using CNAME records in DNS. First, we log into the DNS server and launch **DNS Manager**.
2. Once inside **DNS Manager**, expand **Forward Lookup Zones** and then your domain name so that we can see the list of DNS records that exist already.
3. Now right-click on your domain and select **New Alias (CNAME)...**.
4. We would like our users to be able to browse the website by typing in `http://intranet`. So in our CNAME record, we want the **Alias name** to be `INTRANET` and the **FQDN for target host** to be `WEB-01.MYDOMAIN.LOCAL`, which is the server where the website is being hosted.




5. We also want our file shares to be accessible by using `\\FILESERVER\SHARE`, so that the actual name of the server hosting this share is not visible to the users. Create another CNAME record with the **Alias name** field as `FILESERVER`, and the **FQDN for target host** field as `WEB-01.MYDOMAIN.LOCAL`.
6. Log into the test client machine and give it a try. Users are now able to open up Internet Explorer and successfully browse to `http://intranet`. They are also able to open File Explorer and access `\\fileserver\share`.

### How it works...

We have a server in our environment called WEB-01. There is a website running on this server. It is also hosting a file share called SHARE. By creating a couple of quick CNAME records inside DNS, we are able to give users the ability to use some intuitive names to access these resources. By following the above instructions, we have masked the actual server name from the users, making knowledge of that name unnecessary. Masking internal hostnames of servers is also considered a security best practice in many organizations.

### See also

- The *Creating an A or AAAA record in DNS* recipe

## Creating a DHCP scope to assign addresses to computers

In the *Configuring a combination Domain Controller, DNS server, and DHCP server* recipe, we installed the DHCP role onto a server called DC-01. Without some configuration, however, that role isn't doing anything. In most companies that I work with, all of the servers have statically assigned IP addresses, which are IPs entered by hand into the NIC properties. This way, those servers always retain the same IP address. But what about client machines that might move around, or even move in and out of the network? DHCP is a mechanism that the clients can reach out to in order to obtain IP addressing information for the network that they are currently plugged into.

This way, users or admins don't have to worry about configuring IP settings on the client machine, as they are configured automatically by the DHCP server. In order for our DHCP server to hand out IP addresses, we need to configure a scope.

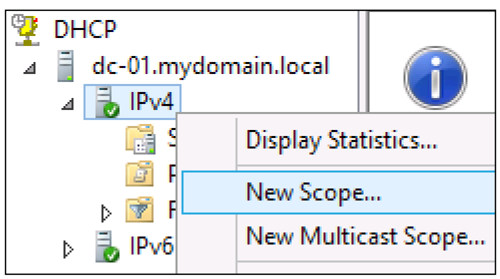
## Getting ready

We have a Server 2012 R2 machine online with the DHCP role installed. We will also be testing using a Windows 8.1 client machine to ensure that it is able to acquire IP address information properly from the server.

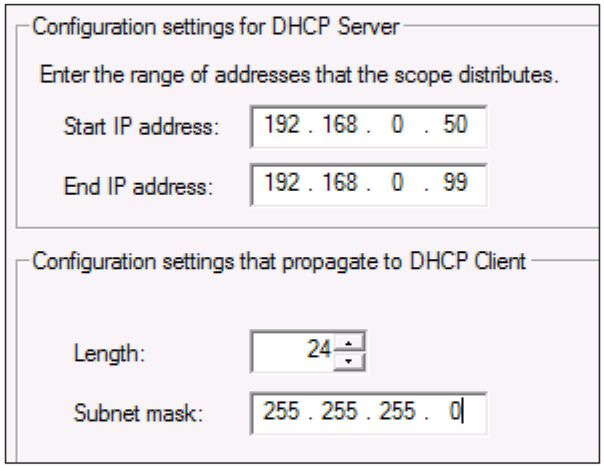
## How to do it...

Perform the following steps to create and configure a DHCP scope to assign addresses to client computers:

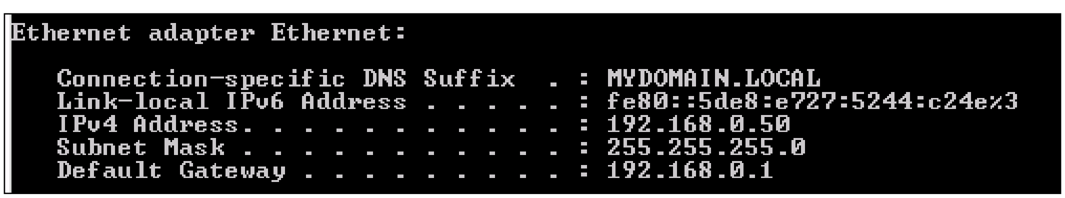
1. Dropdown the **Tools** menu inside Server Manager, then click on **DHCP**. This opens the DHCP management console.
2. Expand the left pane, where the name of your DHCP server is listed. You will see sections for **IPv4** and **IPv6**. For our network we are sticking with IPv4, so we right-click on that and choose the option for **New Scope...**.




3. Start the **New Scope Wizard** screen by creating a name for your scope. This can be anything you like.
4. Enter a range of IP addresses that you would like the DHCP server to hand out to computers. The **Subnet mask** field will likely populate automatically; just double check to make sure it is accurate.




5. On the **Add Exclusions and Delay** screen, if there are any IP addresses within the scope you just defined that you do not want handed out, specify them here. For example, if you are going to use .50 through .99, but you already have a print server running on .75, you could exclude .75 on this screen so that DHCP doesn't try to hand out the .75 address to a client computer.
6. Now set a time in your **Lease Duration** field. This is the amount of time in between DHCP *refreshes* for a client computer. If a particular computer leaves the network and comes back within its lease duration, it will be given the same IP address that it had last time. If you're not sure about this one, leave it set at the default and you can adjust it later.
7. Next we will populate the rest of the IP information that the client computers need to receive on our network. Fill out fields for **Router (Default Gateway)**, **Domain Name and DNS Servers**, and **WINS Servers** if necessary.
8. The last item to choose is **Yes, I want to activate this scope now**. We're in business!
9. As a quick test, let's boot a client computer onto this network whose NIC has not been configured with a static IP. If we take a look at its IP configuration, we can see that it has successfully received IP addressing information from our DHCP server automatically.



## How it works...

DHCP is one of the core infrastructure roles that almost everyone uses inside their networks. While we have only scratched the surface here of what DHCP is capable of, the ability to automatically hand out IP addresses to connecting client computers is DHCP's core functionality. Installing the role and creating a scope are our primary steps to make use of DHCP. Take a look at our next recipe for one of the more advanced functions that can be accomplished within your scope.

# Creating a DHCP reservation for a specific server or resource

In a simple DHCP scope, any device that connects and asks for an IP address is handed whatever IP is next available within the scope. If you have a device for which you always want to keep the same IP address, you could manually configure the NIC properties with a static IP address. Otherwise, a more centralized way to assign a particular IP to the same device on a long-term basis is to use a DHCP reservation. Using a reservation in DHCP to assign an IP to a device makes a lot of sense, because you can see that reservation right in the DHCP console and you don't have to worry about keeping track of the static IP addresses that you have configured out in the field. Let's walk through configuring a quick reservation so that you are familiar with this process.

## Getting ready

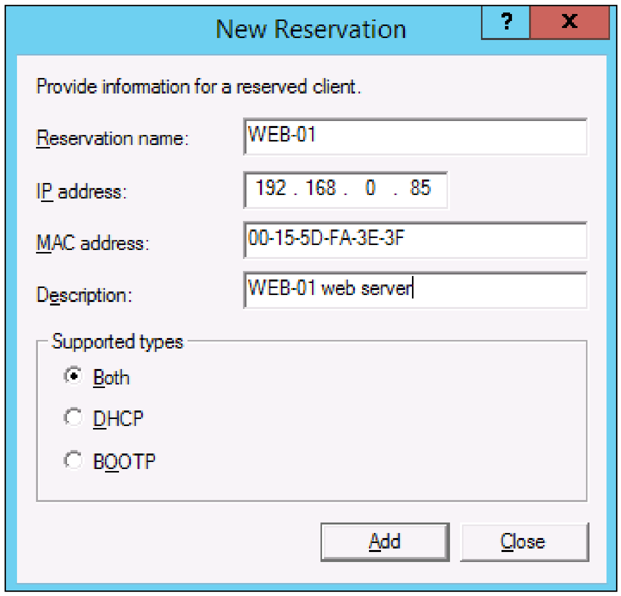
We will be using a Windows Server 2012 R2 machine as our DHCP server where we will create the DHCP reservation. Additionally, we will use our WEB-01 server to be the recipient of this reservation by assigning WEB-01 to IP address `192.168.0.85`.

## How to do it...

To create a DHCP reservation for a specific server or resource, perform these instructions:

1. Open the **DHCP** manager tool.
2. Expand the left pane down into the DHCP scope that we created earlier. Under this scope, you will see a folder called `Reservations`. Right-click on `Reservations` and click on **New Reservation...**.
3. Populate the fields. Your **Reservation name** field can contain anything descriptive. Fill out the **IP address** field with the IP address you want to reserve for this purpose. The last important piece of information is the **MAC address** field. This must be the MAC address of the device for which you want to receive this particular IP address. Since WEB-01 is a Windows Server 2012 R2 machine, we can get our MAC address by doing `ipconfig /all` on WEB-01.


4. You can see **Physical Address.........: 00-15-5D-FA-3E-3F** in Command Prompt—this is our MAC address for WEB-01. Use it to finish populating the DHCP reservation.




5. Click on **Add** and you will see your new reservation listed in the **DHCP** management console.
6. Now make sure that the NIC on WEB-01 is set to **Obtain an IP address automatically**. When WEB-01 reaches out to DHCP to grab an IP address, it will now always receive 192.168.0.85 because of the reservation, rather than getting whatever IP address is next available within the DHCP scope.

## How it works...

Typically, whenever a client computer is set to obtain an IP address automatically, it reaches out and looks for a DHCP server that hands to the client whatever IP address is free and next in the list. This causes DHCP clients to change their IP addresses on a regular basis. For desktop computers, this is usually fine. In many cases, however, it is beneficial to reserve particular IP addresses for specific devices, thereby ensuring they always receive the same IP address. Creating DHCP reservations is a good practice for servers, and also for many static devices on the network, such as print server boxes and telephony equipment.

# Pre-staging a computer account in Active Directory

Joining computers to your domain is going to be a very normal task for any IT professional, enough that all of you are probably familiar with the process of doing so. What you may not realize, though, is that when you join computers or servers to your domain, they get lumped automatically into a generic `Computers` container inside Active Directory. Sometimes this doesn't present any problem at all and all of your machines can reside inside this `Computers` container folder forever. Most of the time, however, organizations will set up policies that filter down into the `Computers` container automatically. When this is the case, these policies and settings will immediately apply to all computers that you join to your domain. For a desktop computer, this might be desired behavior. When configuring a new server though, this can present big problems.

Let's say you are interested in turning on a new remote access server that is going to be running DirectAccess. You have a domain policy in place that disables the Windows Firewall on computers that get added to the `Computers` container. In this case, if you turned on your new remote access server and simply joined it to the domain, it would immediately apply the policy to disable Windows Firewall, because it is no different than a regular client computer in your network. DirectAccess requires Windows Firewall to be enabled, and so you have effectively broken your server before you even finish configuring it! You would eventually realize this mistake and move the server into a different OU that doesn't have the firewall squash policy; however, this doesn't necessarily mean that all the changes the policy put into place will be reversed. You may still have trouble with that server on an ongoing basis.

The above example is the reason why we are going to follow this recipe. If we pre-stage the computer account for our new remote access server, we can choose where it will reside inside Active Directory even before we join it to the domain. **Pre-staging** is a way of creating the computer's object inside Active Directory before you go to the actual server and click **Join**. When you do this, as soon as the request to join the domain comes in, Active Directory already knows exactly where to place that computer account. This way, you can make sure that the account resides inside an OU that is not going to apply the firewall policy, and keep your new server running properly.

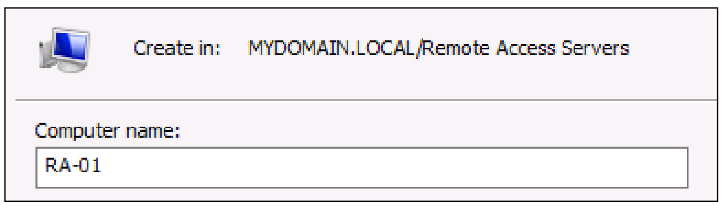
## Getting ready

We will use a Server 2012 R2 Domain Controller to pre-stage the computer account. Following the example above, we will use a second server that we are going to join to our domain, which we plan in the future to turn into a remote access server.

## How to do it...

To pre-stage a computer account so that it resides inside Active Directory, perform the following steps:

1. Open the **Active Directory Users and Computers** tool on a Domain Controller.
2. Choose a location in which you want to place this new server. I am going to use an OU that I have created called **Remote Access Servers**.
3. Right-click on your OU and choose **New** | **Computer**.
4. Enter the name of your new server. Make sure this matches the host name that you are going to assign as you build this new server, so that when it joins the domain, it matches up with this entry in Active Directory.




5. Click **OK** and that's it! Your object for this new server is entered into Active Directory, waiting for a computer account to join the domain that matches the name.
6. The last step is building the `RA-01` server and joining it to the domain just like you would with any computer or server. When you do so, it will utilize this pre-existing account in the **Remote Access Servers** OU, instead of placing a new entry into the generic `Computers` container.

## How it works...

Pre-staging computer accounts in Active Directory is an important function when building new servers. It is sometimes critical to the long-term health of these servers for them to steer clear of the default domain policies and settings that you apply to your regular computer accounts. By taking a quick 30 seconds prior to joining a new server to the domain to pre-stage its account in AD, you insure the correct placement of the system so that it fits your organizational structure. This will keep the system running properly as you continue to configure it for whatever job you are trying to accomplish.

# Using PowerShell to create a new Active Directory user

Creating new user accounts in Active Directory is pretty standard stuff, but doing it the traditional way requires a lot of mouse clicks. Since we know that PowerShell can be used to accomplish anything within Windows Server 2012 R2, but not many people actually employ it regularly, let's use this common task as a recipe to be accomplished with PowerShell rather than the GUI.

## Getting ready

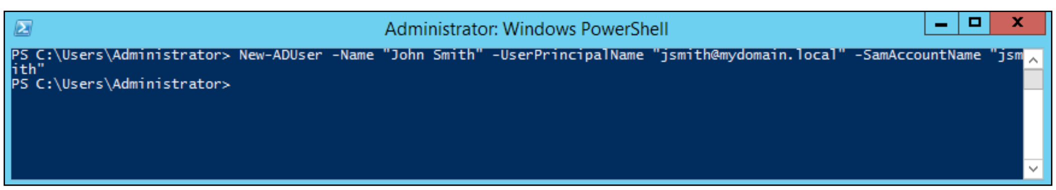
We will use PowerShell on our Windows Server 2012 R2 Domain Controller in order to create this new user account.

## How to do it...

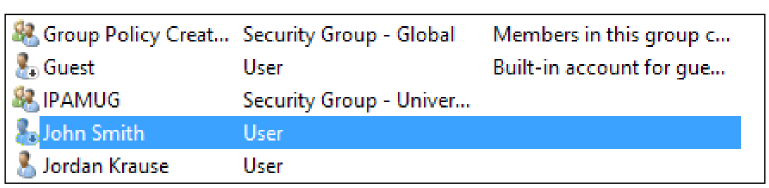
Follow along to create a new user account in Active Directory by using the PowerShell command prompt:

1. Launch a PowerShell command prompt as an Administrator.
2. Enter the following command in order to create a new user account with very simple parameters:

   ```
   New-ADUser -Name "John Smith" -UserPrincipalName
   "jsmith@mydomain.local" -SamAccountName "jsmith"
   ```

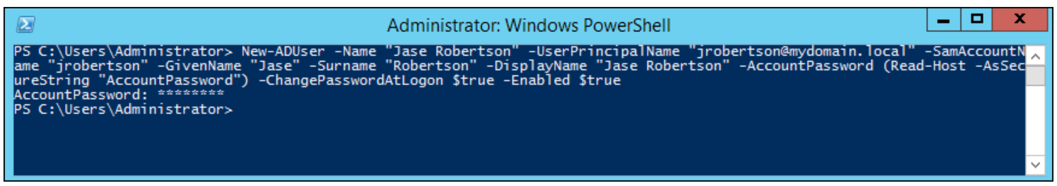



3. If you open up the GUI for **Active Directory Users and Computers**, you will see that **John Smith** has now been created as a **User** account. There aren't many properties that exist within this account, as it is pretty simple, but it will work in order to get a new user up and running.

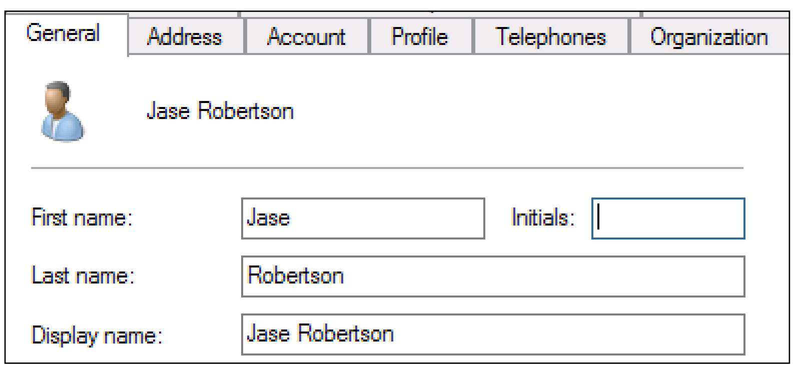


Now let's create another new user, this time adding some additional parameters to our code in order to populate more of the typical user information. You may have also noticed that our new **John Smith** user is currently disabled—this happens automatically when you create a new user account but do not populate a password. So we will add in some more information up to first name and surname. We will also specify a couple of additional parameters in order to make sure the account is enabled and to require that the user changes their password during their initial login:

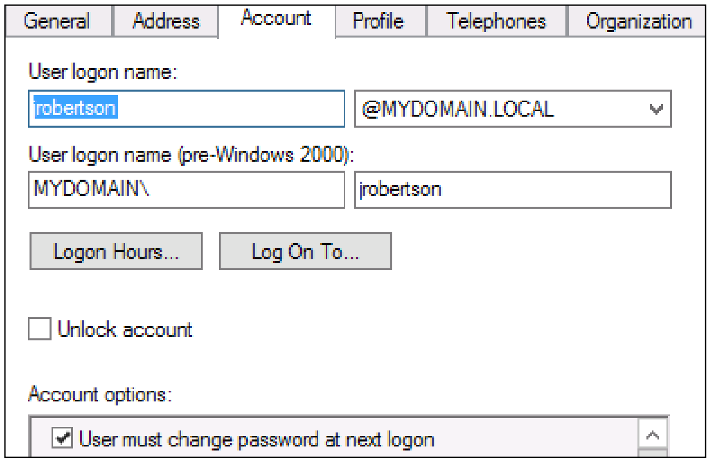
```
New-ADUser – Name "Jase Robertson" -UserPrincipalName
"jrobertson@mydomain.local" – SamAccountName "jrobertson"
-GivenName "Jase" -Surname "Robertson" -DisplayName
"Jase Robertson" -AccountPassword (Read-Host -AsSecureString
"AccountPassword") -ChangePasswordAtLogon $true -Enabled $true
```




4. Open up **Active Directory Users and Computers** again and take a look at our new **Jase Robertson** user account. You can see that the account is enabled and ready for use, and it has much more information populated inside the account.



5. Move over to the **Account** tab and you will also see the box is now checked for **User must change password at next logon**, just like we specified in our PowerShell command:




## How it works...

By using PowerShell, we are able to create new Active Directory user accounts right from a command interface, rather than logging into a server and launching the graphical interface in order to accomplish this common task. Can your `New-ADUser` commands become extremely lengthy in order to populate all of the attributes you want to include? Yes. However, can saving and running a PowerShell script that utilizes `New-ADUser` cmdlet save you time in the long run? Absolutely! It might take a few minutes of thought and testing in order to get your script to the point where it populates the information that you would like, but once you have created and saved that script, it can be modified and run quickly in the future in order to create new accounts. There is even a way to utilize the `New-ADUser` cmdlet to copy properties from an existing user account while it sets up the new one, which may also help to save you some time and energy on new user account creations.

## See also

Make sure to check out the following TechNet link. This page lists all of the possible parameters and syntax that you might want to run alongside your `New-ADUser` cmdlet script. There are a ton of options!

- `http://technet.microsoft.com/en-us/library/ee617253.aspx`

# Using PowerShell to view system uptime

I find myself constantly checking servers to figure out what time they last restarted. Usually, this is part of troubleshooting something in order to figure out whether the server rebooted as a planned action or if something went wrong and it restarted on its own during a non-standard time. For years, I had launched Event Viewer, waited for the System logs to open, hoped that they weren't corrupted in some way, and then headed over to noon on the previous day to find the amount of seconds that the system had been online. Then I'd pull out the calculator and do the math on how many days/hours that really was. Way too complicated! Thankfully, we can make calls into WMI objects with PowerShell, and there is an object in there that will tell us the last time the server started. With a few lines plugged into a `.ps1` script, we can create ourselves a nice little tool that will output the last time that a server booted. Let's give it a try.

## Getting ready

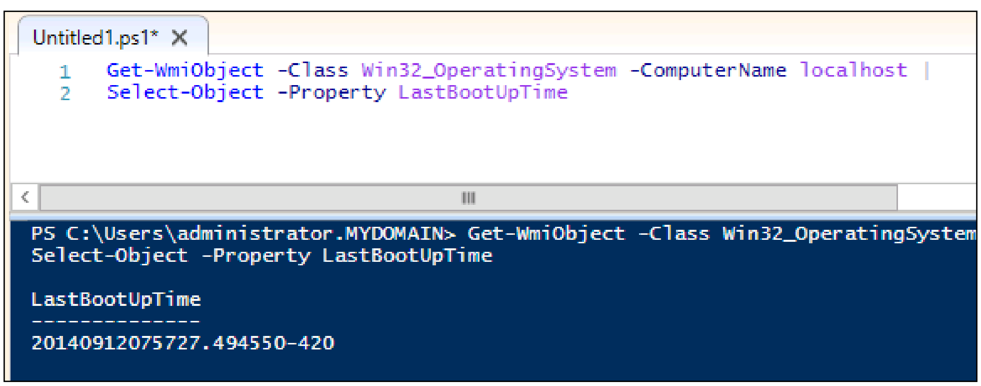
We are using a Windows Server 2012 R2 machine to build this script.

## How to do it...

To build a script that shows us the last system boot time:

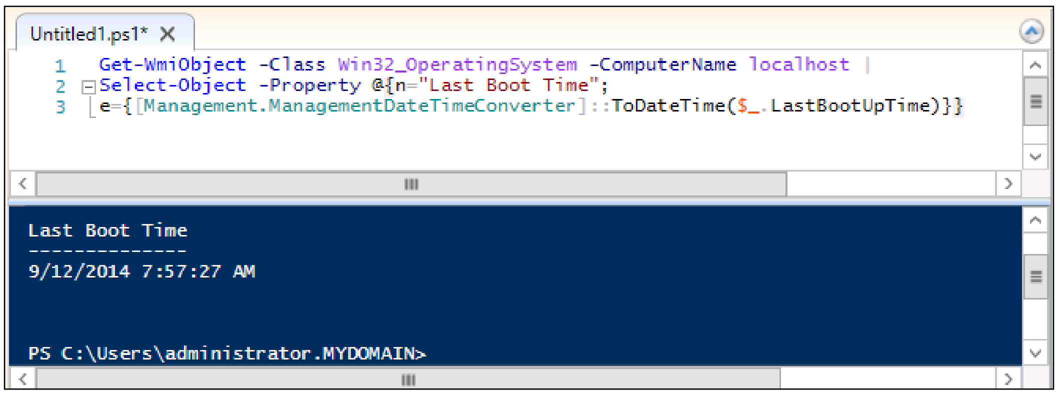
1. Launch PowerShell ISE as an Administrator.
2. Open up a new script file and input the following line:

```
Get-WmiObject -Class Win32_OperatingSystem -ComputerName
localhost | Select-Object -Property LastBootUpTime
```



3. We have some data! It's kind of messy data though. Maybe we can clean that up and make it a little more readable. With a couple of changes to our `Select-Object` code, we can change both the header for this data to something more friendly, as well as changing the output of the date and time to being way easier on the eyes:

```
Get-WmiObject -Class Win32_OperatingSystem -ComputerName
localhost | Select-Object -Property @{n="Last Boot Time";
e={[Management.ManagementDateTimeConverter]::ToDateTime
($_.LastBootUpTime)}}
```

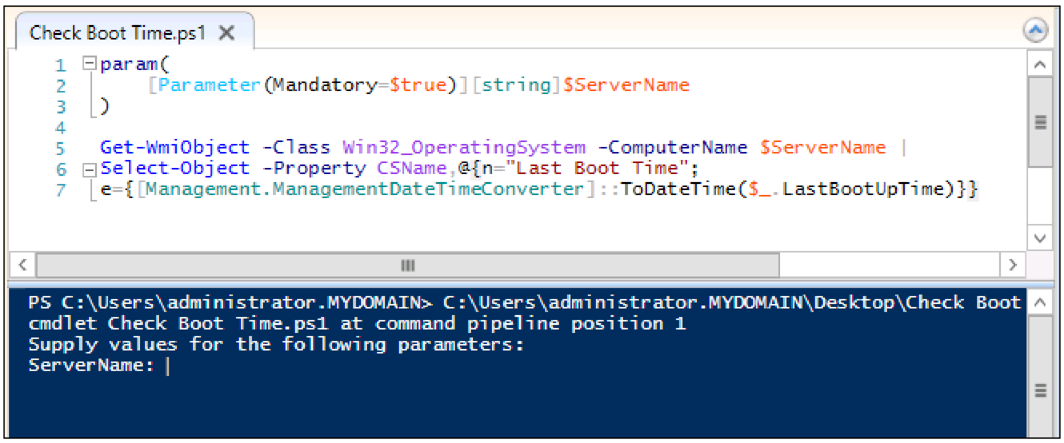



That looks much better. At this point, I would say that this script is ready to be saved and used on any individual machine, and it would quickly give you the output you are looking for on that particular server. But, as you can see in the code, we are currently hardcoding computer name to be `localhost`, the server or computer where we are currently running this script. What if we could change that so the user running this script could enter a different computer name? Maybe we could then use this script to execute a remote reach and find out when different servers last booted, without having to log into those servers? Here is an example of doing just that. With a few changes to our code, we can require that the user inputs a computer name as a flag while running the script, and outputs two properties now. We will place an additional property identifier in there for the computer name itself, so that it is clear to us in the output that the last boot time we are looking at is for the server name that we actually enter.

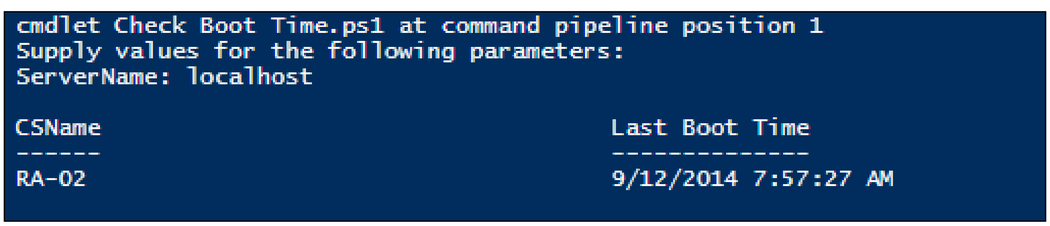
4. Use this for your script code:

```
Param(
[Parameter(Mandatory=$true)][string]$ServerName
)
Get-WmiObject -Class Win32_OperatingSystem -ComputerName
$ServerName | Select-Object -Property CSName,@{n="Last Boot Time";
e={[Management.ManagementDateTimeConverter]::
ToDateTime($_.LastBootUpTime)}}
```

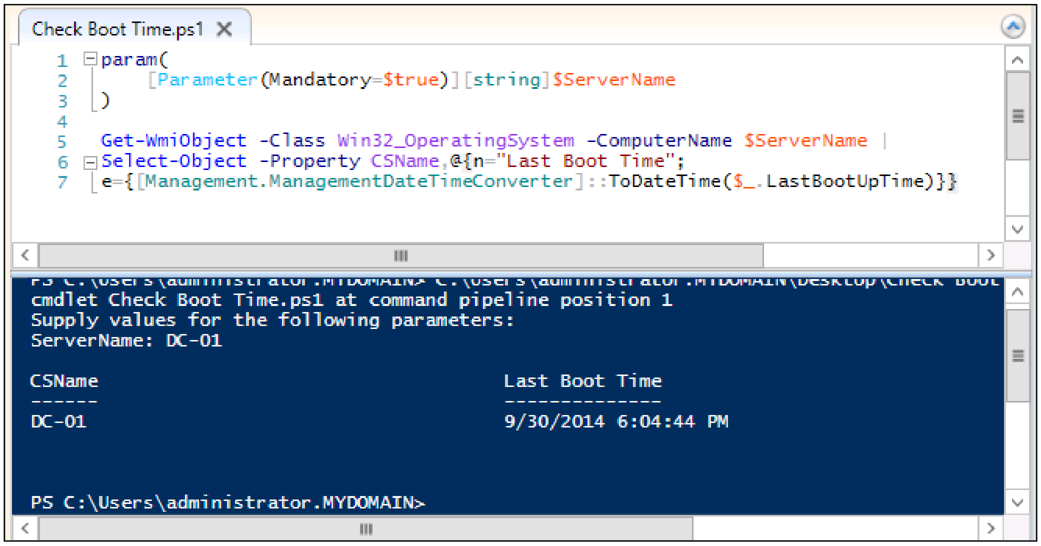
5. Now when we run this script, we are asked to input the server name that we are trying to query.




6. Go ahead and type `localhost` and you will receive the same boot time information as before; but now you see that we have a new column that shows us the server's name as well.



7. Try running the script again, but this time enter the name of a remote server when it asks for **ServerName**. I will try to query our primary Domain Controller, `DC-01`. You should now see the last boot time output for that particular server with its name in the left column.




## How it works...

In this recipe, we created a fun little script that asks for a server name and outputs the last boot information for the server entered. One of PowerShell's greatest attributes is its ability to grab information, both locally on the machine where you are running the script as well as on remote machines. This saves time since you don't have to log into those servers to accomplish tasks, and makes your work environment more efficient.

One note on this particular script that we created, you don't have to run it as a first step and then take a second step in order to enter the server name. You are able to place the `ServerName` variable into your initial command when you launch the script. For example, open PowerShell and input this to launch the script:

```
.\'Check Boot Time.ps1' -ServerName DC-01
```

This will launch the script and automatically input **DC-01** as the server that it is checking, instead of stopping to ask you for input.

# 3
# Security and Networking

Various breaches and vulnerabilities over the past few years have brought security to the forefront of all IT Administrators' minds. In a Windows Server 2012 R2 environment, there are many functions you can enable to lock down security on your own network. Let's explore some of these functions, as well as practice with some tools and tricks that can help us to better understand and navigate our own networks. We will also take a look at some common networking tasks that will help you out in your day-to-day work. In this chapter, we will look at the following recipes:

- Requiring complex passwords in your network
- Using Windows Firewall with Advanced Security to block unnecessary traffic
- Changing the RDP port on your server to hide access
- Multi-homing your Windows Server 2012 R2
- Adding a static route into the Windows routing table
- Using Telnet to test a connection and network flow
- Using the Pathping command to trace network traffic
- Setting up NIC Teaming
- Renaming and domain joining via PowerShell

# Introduction

Today we are going to tackle a number of tasks related to networking your Windows Server 2012 R2 machines and locking down that environment a little bit with the enabling of some security functions. Some of the tools we are going to use can be very useful for daily tasks, and I hope that the steps we take will prompt you to start some gears spinning in your own mind to investigate deeper into taking full advantage of what Microsoft has offered you within this operating system.

# Requiring complex passwords in your network

With the tools that attackers have available today, simple passwords should be outlawed by every company. Turning on the requirement for complex passwords in your network is pretty simple; the hard part is knowing where to find the setting. We are going to require complex passwords by making a change inside Group Policy. Later in this book, we are going to do a lot of things inside Group Policy, but the requirement for complex passwords is so common that I felt it to be a general security item rather than something to be lumped alongside other Group Policy tasks. So we will be using Group Policy in a step-by-step fashion; combining this recipe with the chapter on Group Policy will give you even more creativity in the way that you could later change the implementation of this password policy.

## Getting ready

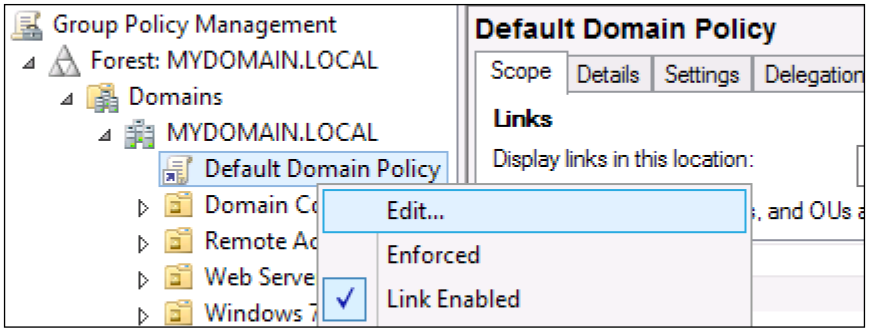
We need to be working in a domain environment as Group Policy is something that runs within Active Directory. The change that we are going to make in Group Policy is done from a Domain Controller, and we will utilize a client computer to test our policy once it has been implemented.

## How to do it...

The following steps will help you enable complex passwords for your network:

1. On your Domain Controller, launch **Group Policy Management** from inside the **Tools** menu in Server Manager.
2. Expand your forest name and find the name of your domain inside the **Domains** folder. If you expand your domain name, you will see a **Group Policy Object** (**GPO**) in there called the **Default Domain Policy**. This policy is automatically configured in a new Active Directory environment to apply to all user accounts, so for this recipe, we will modify this GPO to require complex passwords for all of our users.

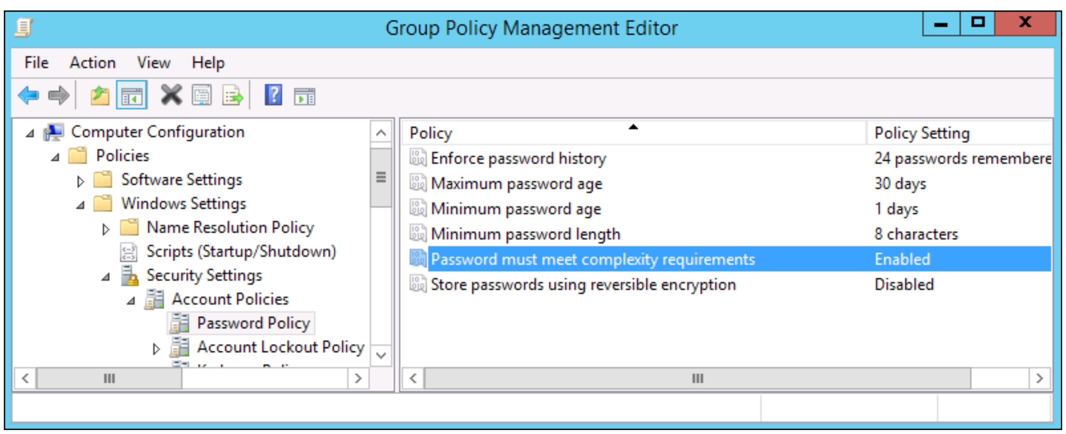
3. Right-click on **Default Domain Policy** and click **Edit...**.



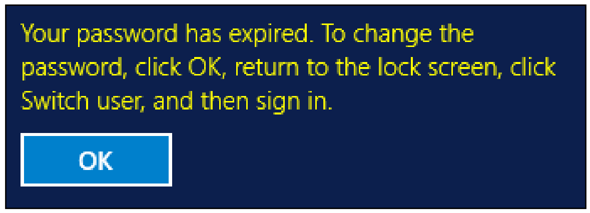

[ You can easily create a new GPO and use it instead of modifying the built-in default policy. This would give you better control over who or what gets the settings applied to them. See *Chapter 9*: *Group Policy* later in the book for more detail on managing the GPOs themselves. ]

4. Browse to the following location: **Computer Configuration** | **Policies** | **Windows Settings** | **Security Settings** | **Account Policies** | **Password Policy**.

5. Here are the configurable options that you can set for password requirements in your network. I am going to set **Maximum password age** to `30 days` so that everyone needs to change their password monthly, and I will increase **Minimum password length** to `8 characters`. I will also enable the complexity requirements setting, which sets a number of different requirements. If you double-click on that setting and browse to the **Explain** tab, you will see a list of all the items that are now required.



6. Now go ahead and try logging into a computer with a domain user account; you will discover that our password no longer meets the criteria and we have to change it accordingly.




## How it works...

Because we set requirements for password complexity in the Default Domain Policy, that requirement flows across our whole network. A solid password policy is very important in today's networks and just scratches the surface of Group Policy's abilities. These simple setting changes can make the difference in whether or not your company is compromised as a result of a brute force password attack.

# Using Windows Firewall with Advanced Security to block unnecessary traffic

I encounter far too many networks with policies in place that disable the built-in **Windows Firewall with Advanced Security** (**WFAS**) by default on all of their machines. Usually if I ask about this, the reason is either unknown or "it's always been that way." I think this is a carry-over from the Windows XP / Server 2003 days, or maybe even older, when the Windows Firewall was less than desirable. Believe me when I tell you that WFAS in today's operating systems is very advanced, stable, and beneficial. If you want to stop unnecessary or malicious traffic from getting to your server, look no further than this built-in tool.

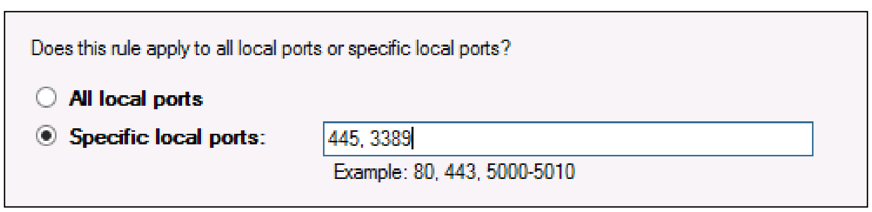
## Getting ready

We are going to use two Windows Server 2012 R2 machines for this task. We will test connectivity between the two to set our baseline and then create a rule that blocks the functions we just tested. Next we will test again to ensure that our changes did what we expected them to: blocking the traffic that we attempt to generate. It is important to set up a baseline of tests and run those same tests following each change to ensure the rules are working exactly as you want them to be.

## How to do it...

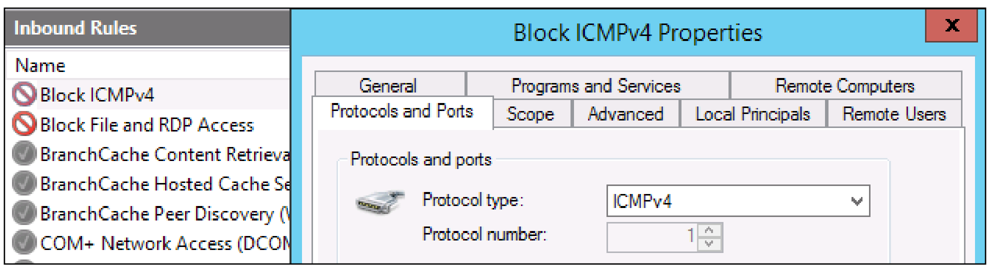
If you want to stop unnecessary traffic from getting to your server, execute the following instructions:

1. First we want to test the existing connectivity. I log in to my DC-02 server and from there I am able to successfully execute the command `ping web-01` and get a reply. I can also open up File Explorer, browse to `\\WEB-01`, and see a folder shared there. This baseline test tells me that both ICMP (ping) traffic and file access are open and allowed by WFAS on WEB-01 currently. We want to stop these functions from happening.
2. Log in to WEB-01 and open **Windows Firewall with Advanced Security**. You can open this either from the **Start** screen and typing it in, or by opening a **Run** prompt and typing `wf.msc`.
3. Inside WFAS, your two best friends when trying to control traffic are the **Inbound Rules** and **Outbound Rules** sections on the left. You need to think of Inbound and Outbound from the server's perspective. Inbound rules manipulate traffic that is flowing in toward your server, and Outbound rules handle traffic flowing out of your server toward the rest of the network. If you click on **Inbound Rules**, you will see the list of preconfigured rules that exist already.
4. Right-click on **Inbound Rules** and click on **New Rule...**.
5. First let's make a rule to block file access from happening. Choose **Port** and on the next screen enter port **TCP** `445`. Then you realize that you might as well also block RDP access since that is also currently enabled. No problem! Simply comma-separate these numbers as follows:




6. Choose **Block the connection**.
7. On the next screen where you choose which firewall profile the rule applies to, you can leave it set to all three checked as the default. This will ensure that the rule will apply to any NIC that has any firewall profile assigned. If you only have a single NIC in your server and it is joined to the domain, then you could get away with only selecting the domain profile if you wanted to deselect the other two. For our recipe, I'm going to leave them all checked.
8. Type any kind of descriptive name for your rule. Something like `Block File and RDP Access`.

9. You did it! You will see that the new rule exists, and it is immediately put into action. If you head over to your other server, you will now find that you can no longer RDP or browse the file shares at all on WEB-01.
10. We can still successfully ping WEB-01 though and we wanted to put a stop to that as well. To stop ICMP traffic, you simply need to create another rule. This one is a little bit more complicated though. First, go ahead and create a second Inbound rule, and use the exact same settings that you used for your RDP/file rule. You can enter anything into the **Port** field; it doesn't matter because we will be invalidating it in a minute. So maybe use port 445 for our example.
11. Great, now you have two rules in there that are both blocking port 445. That doesn't do us much good. Right-click on the newest rule that we just created, head into **Properties**, and let's improve this rule a little bit.
12. Inside the **Protocols and Ports** tab, drop down the **Protocol** type and choose **ICMPv4**. That's all you have to do! You have now modified this rule so that it is no longer blocking TCP port 445; rather, this rule is now blocking ICMPv4 traffic.




13. If you log back into DC-02, we no longer receive ping replies when trying to contact the WEB-01 server.

> Take some time to play around inside the **Scope** tab. This section of a WFAS rule allows you to scope the rule down so that it only applies to particular IP addresses or ranges. Maybe you only want to block file share access from a particular subnet or only for the External NIC of an edge server. Requirements like these are easy to accomplish!

## How it works...

We used the Windows Firewall with Advanced Security to create a couple of simple rules to block unwanted traffic coming into our server. These rules are put into place immediately and are very easy to generate. What is even greater is that our WFAS rules can be created centrally by making use of Group Policy so that you don't even have to touch the individual servers to apply connection rules to them. WFAS is very different than the Windows Firewall 10 years ago, and if you are not making use of it today I seriously recommend that you reconsider.

# Changing the RDP port on your server to hide access

Everybody uses RDP. Attackers and bots, curiously, also know that *everybody* uses RDP. If you are working with perimeter servers that are potentially connected to the internet, having RDP enabled can be especially dangerous because it is quite easy to leave your server in a state where it is open from outside your network. This gives anyone the ability to start guessing passwords or trying to brute force their way into your server; or they can just be a way to give you some denial-of-service headaches by throwing thousands of login attempts at that server.

Even aside from the worries of potential access from the public Internet, you may want to ensure that regular users aren't trying to poke around where they shouldn't be by opening up RDP connections to servers within your network. There are a few ways that you could restrict this access. You could come up with some creative firewall rules that only allow RDP access from certain subnets, and try to contain your IT computers to those subnets. Or maybe you could configure RDP on your destination servers to require certificates as part of the authentication process, thus only allowing users with those certificates to have access. These are both fairly complicated. I like to employ a much simpler solution to keep unwanted eyes from seeing my RDP login screens.

Change the port that RDP runs on! What? You can do that? Yes, RDP by default listens and connects on TCP port 3389. The Remote Desktop Client that is installed onto almost every client machine everywhere automatically assumes that the server they are connecting to is listening on 3389, and so you don't even have to specify a port when you try to connect. There isn't even a field for it in the client. So it's pretty rare that I talk to people (even IT people) who know that 3389 is the default. Given that, if we were able to change that 3389 to something different, something of our own choosing, I think that would do a grand job of keeping people out of our systems. Let's say we have a sensitive server and want to keep access to a minimum. Let's change the RDP port on that to something only we know, maybe port 4822. That'll keep 'em guessing for a while.

## Getting ready

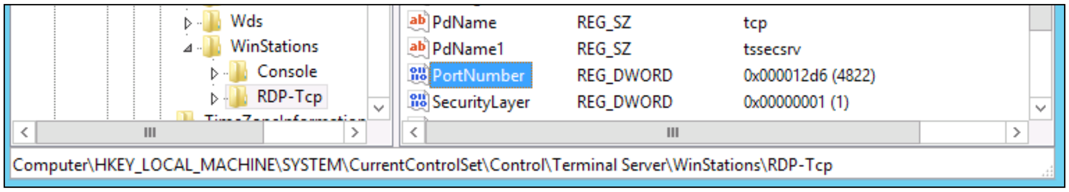
Any Windows Server 2012 R2 machine will do for this task. We are going to set our custom RDP port on WEB-01, and then we are going to test accessing it from a Windows 8 client machine.

## How to do it…

Go through the following to change the RDP port to one of your liking:

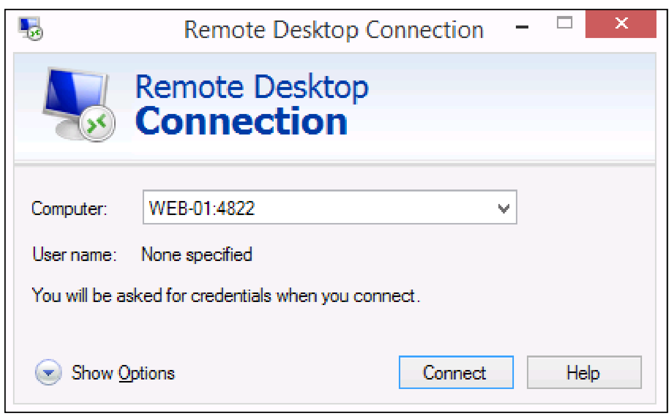
1. Open **Registry Editor**. You can do this by going to either the **Start** screen or a **Command Prompt** and typing `regedit`.

2. Browse to: [**HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp**].
3. Find the value called **PortNumber** and change it to `4822`.




4. Restart the server.
5. Now log into your client computer and open up **Remote Desktop Connection** by typing that name into your **Start** screen. You can also type `mstsc` in the command prompt to open this program.

   If you try to connect directly to `WEB-01`, your connection will fail as the server is no longer listening on the standard port 3389.
6. Enter **WEB-01:4822** and you connect successfully.




[ If at first you cannot connect, make sure to check over your Windows Firewall settings. It is possible that you may need to add a rule to WFAS on the server to allow port 4822. ]

### How it works...

With a simple registry change, we can adjust the RDP listener port on servers. This will help keep unwanted RDP connections from being made, which can be useful both inside and outside the corporate network.

After making this change, the only people who will be able to reach the RDP login screen would be those who know your new RDP port, and who know how to utilize that custom port within the Remote Desktop Connection tool.

## Multi-homing your Windows Server 2012 R2

Historically there haven't been many scenarios that require Windows servers to have more than a single network card. This is because most of the roles that they were accomplishing were done on whatever single network they were plugged into. There was no need for a server to have direct connections to multiple networks because that was the router and switch's job, right? In today's Windows Server world, there are numerous roles that can take advantage of multi-homing, which simply means having multiple NICs connected to different networks at the same time. There are some proxy roles that can use multiple NICs; Remote Access roles such as DirectAccess and VPN recommend a dual-NIC setup, and you can even use a Windows Server as a general router if you want to.

I work a lot with DirectAccess and I find many multi-homed servers with incorrect network configurations. This recipe is a collection of points that need to be followed when configuring a Windows Server with multiple NICs to make sure it behaves and flows traffic as you expect it to.

### Getting ready

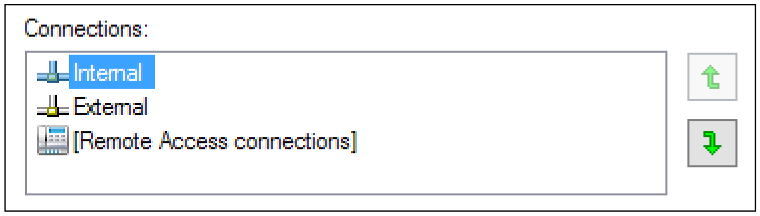
You just need a Windows Server 2012 R2 online for this one. We have two NICs installed on this server and they are plugged into different networks. I am prepping a Remote Access server that will sit on the edge, so I have one NIC plugged into the corporate internal network, while the other NIC is connected to the Internet.

## How to do it...

To configure a Windows Server with multiple NICs, perform the following steps:

- **Only one Default Gateway**: In your NIC properties, you need to make sure that you only have a Default Gateway identified on one of your NICs. This is the most common mistake that I find in the field. If you have two NICs, it would seem logical that you would simply populate their IP address settings just like you would with any server or computer, right? Nope. The purpose of a Default Gateway is to be the "fallback", or the route of last resort. Whenever your server tries to send out network traffic, it will search the local routing table for information on how to send out that traffic. If it does not find a specific route that corresponds to the IP address that you are sending to, then it will default that traffic over to the Default Gateway address. Therefore, you only ever want to have *one* Default Gateway assigned on a server, no matter how many NICs are connected. On all other NICs installed on the system, simply leave the Default Gateway field unpopulated inside the TCP/IP properties. By the way, for a DirectAccess server, the Default Gateway needs to be on the External NIC, so I will be leaving that field empty in the properties of my Internal NIC.
- **Limit your DNS servers**: Another common configuration that I have seen is to have DNS server addresses defined for every network adapter installed on the system. While this doesn't usually break anything like multiple Default Gateways can, it does cause unnecessary slowness when the system is trying to resolve DNS names. Try to have DNS server addresses configured on only one NIC. Once again using our example DirectAccess server setup, I will be configuring DNS server addresses on my Internal NIC because that is necessary for DA to work. I will not be putting my public DNS server specifications into the External NIC; instead I will leave those fields empty.
- **Use static IP addresses**: The roles and functions you may perform on a Windows Server that require multiple NICs will be best served by having static IP address information assigned to those network cards. If you let one or more of the NICs pull information from DHCP, you could easily create a situation where you have too many DNS servers defined, or where you have multiple Default Gateways on your system. As we already know, either of these scenarios is not desirable.
- **Prioritize the NIC binding**: It is a good practice to set a priority for your NICs so you can place the card that you expect to have the most network traffic as #1 in the list. For our DirectAccess server, we always want the Internal NIC to be placed on the top, so let's make sure that is set correctly using the following steps:
    1. Open up **Network and Sharing Center** and click on **Change adapter settings** so that you are in the **Network Connections** screen where you can see the network cards installed onto your system.
    2. Now press the *Alt* key on your keyboard and you will see the menus at the top of this window.

3. Head into the **Advanced** menu and click on **Advanced Settings...**. Now simply make sure that your **Internal** NIC is listed on top.




- **Add static routes**: A couple of minutes ago, you probably started thinking "Hey, if I don't have a Default Gateway on my Internal NIC, what tells the server how to get packets into the subnets of my internal network?". Great question! Because you only have one Default Gateway, when you need to send traffic out one of the other NICs, you need to make sure that a static route exists in the Windows routing table. This ensures that the server knows which interface gets traffic for each subnet. Make sure to check out our next recipe for the specific information on how to add those routes.

## How it works...

Anybody can multi-home their server by simply plugging two NICs into two different networks. The tricky part is making sure that you configure those NICs and the operating system appropriately so that network traffic flows in the right directions at the right times. Following this list of rules will give you a solid foundation so that you can build out these types of scenarios and know that you are doing so in the correct fashion. Deviating from these rules will result in unexpected behavior, which sometimes is not immediately obvious. This can make for some very frustrating troubleshooting down the road.

## See also

- The *Adding a static route into the Windows routing table* recipe

# Adding a static route into the Windows routing table

This recipe follows right on the heels of our previous topic. If you have never worked on a server that is making use of more than one NIC, then you have probably never had a reason to poke around in the Windows routing table. The minute that you are tasked with setting up a new server that needs to be connected to multiple networks, or you get thrown into a situation where you need to troubleshoot such a system, this suddenly becomes critical information to have in your back pocket.

On a server that is connected to multiple networks, you only have one Default Gateway address defined. This means any subnets that need to be reached by flowing through one of the other NICs, the ones that do not contain the Default Gateway, need to be specifically defined inside the routing table. Otherwise, Windows simply does not know how to get to those subnets and it will attempt to push all traffic through the Default Gateway. This traffic will never make it to its destination and communications will fail.

Today we are setting up a new VPN server. This server has a NIC plugged into the Internet where remote clients will come in, and another NIC plugged into the internal network so that the client traffic can make its way to the application servers that the users need to access. In this scenario, the Default Gateway must be populated on the External NIC. There will be no Default Gateway address defined on the Internal NIC, and without some assistance, Windows will have no idea how to properly route traffic toward the servers inside the network.

For our example, the Internal NIC is plugged into the 192.168.0.x network. Since it has a direct physical connection to this network, it is automatically able to properly contact other servers that reside on this subnet. So if the VPN server was 192.168.0.5 and we had a Domain Controller running on 192.168.0.2, we would be able to contact that Domain Controller without any additional configuration. But most companies have multiple subnets inside their network. So, what if our VPN users needed to contact a web server that is sitting on the 192.168.1.x network? When traffic comes into the VPN server looking for a destination of 192.168.1.8 (the web server), the VPN server will check its local routing table and find that it does not have an entry for the 192.168.1.x network. Since it doesn't know what to do with this request, it sends it to the Default Gateway, which sends the packets back out the External NIC. Those packets don't have a destination to reach out to the External NIC plugged into the Internet and so the traffic simply fails.

We need to define a static route in the routing table of our VPN server, so that, when VPN clients request resources inside the 192.168.1.x network, then that traffic makes its way to the destination network successfully. We need to bind this route to our Internal NIC so that the VPN server knows it has to send these packets through that physical network interface.

## Getting ready

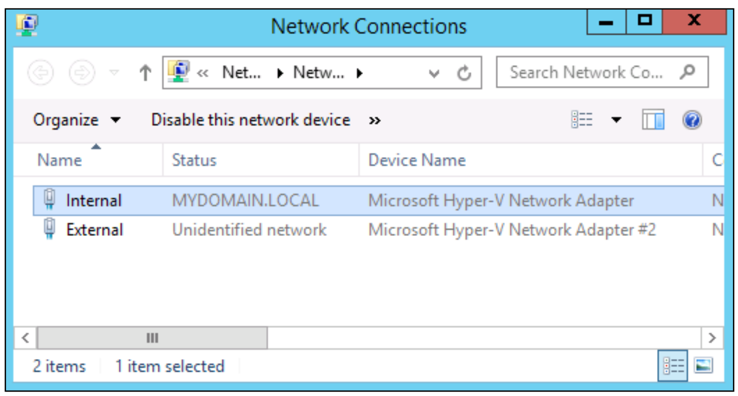
We are setting up a new Windows Server 2012 R2 VPN server. This server has two NICs installed, one plugged into the Internet and the other plugged into the internal network. Inside our corporate network, there are two subnets: 192.168.0.x (/24) that our Internal NIC is plugged into and 192.168.1.x (/24) where our web server resides. There is, of course, a router between the two internal subnets, which is how traffic physically flows between the two. The IP address of that router is 192.168.0.254. If we were able to configure a Default Gateway on the Internal NIC of our VPN server, it would be set to 192.168.0.254 and all traffic would work without any further input. However, since our VPN server is multi-homed and there can only be a Default Gateway configured on the External NIC, we need to tell the server that it has to push 192.168.1.x traffic through 192.168.0.254 by using the Internal NIC.

## How to do it...

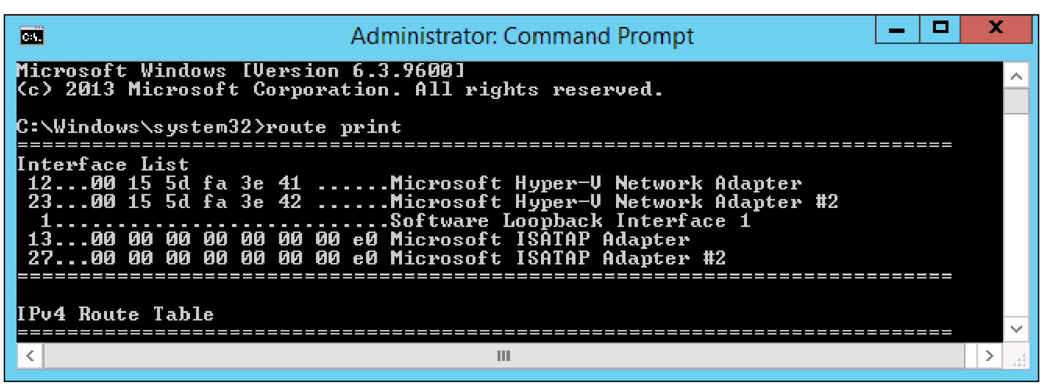
So basically, we need to do the following to create a static route in our VPN server:

- Identify the subnet that we want to contact. In our example, it is 192.168.1.0
- Identify the subnet mask, which is 255.255.255.0
- Identify the IP address of the router that will get us to that network, which is 192.168.0.254
- Identify the Interface ID of the physical NIC that needs to carry this traffic, which can be attained as follows:

  1. Discovering this NIC ID is going to take us a minute. First open up **Network Connections** and expand the fields so that you can see the device name of each NIC.




  2. Now open a command prompt and type `route print`. This is a print of your entire routing table. Scroll back up to the very top and you will see the Interface IDs of your NICs listed.

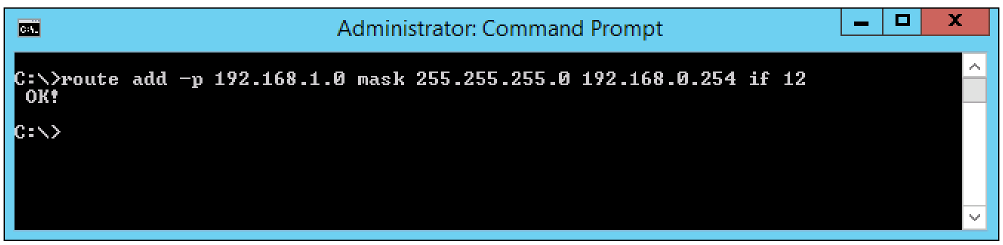


> We can see that our Internal NIC is the first `Microsoft Hyper-V Network Adapter`. Looking at that entry in the route print, there is a two-digit number over to the left of that name. This is our Internal NIC's Interface ID, which is `12` in this example.

We now have all the information needed to put together our route statement and bind it to our Internal NIC. The general format that our route add statement needs to take is `route add -p <subnet> mask <mask> <gateway> if <interfaceID>`. The `-p` part of the command is very important as it makes this route "persistent". Without `-p`, our new route would disappear following reboot.

So in order to tell our VPN server how to send traffic into the new 192.168.1.x subnet that we have been talking about, our specific command is:

```
route add -p 192.168.1.0 mask 255.255.255.0 192.168.0.254 if 12
```




This command tells the server to add a new persistent route for the 192.168.1.0/24 network, flow this network traffic through the 192.168.0.254 gateway, and bind this route to NIC ID 12.

## How it works...

With a multi-homed server, only one NIC will have a Default Gateway. Therefore, any subnets that we need to access through the other interfaces have to be specifically defined. Before we added this new route, the server was completely unable to contact the 192.168.1.x network. This is because the routing table did not have any information about this subnet, so any traffic trying to get there was being sent out the Default Gateway, which is on the External NIC plugged into the Internet. By adding in a static route to our server, we have now defined a routing path for the server to take whenever it has traffic that needs to get to 192.168.1.x.

If you have many subnets in your network, you may be able to cover them all with a blanket route statement. A **blanket route** is also known as an aggregate or supernet route. This could save you time from having to set up a new route statement for each and every one of your networks. For example, if we had many 192.168.something networks and we wanted to flow all of them through our Internal NIC, we could do that with a single route statement such as:

```
Route add -p 192.168.0.0 mask 255.255.0.0 192.168.0.254 if 12
```

This route would send any 192.168.x.x traffic through the Internal NIC. Whether you blanket your routes like this or set each one up individually doesn't make a difference to the server as long as its routing table contains information about where to send the packets that it needs to process.

# Using Telnet to test a connection and network flow

The `ping` command has always been an IT person's best friend to do quick network connection checks. How many of you are the family and neighborhood "go-to guy" to fix anything with buttons? I'm guessing most of you. And as such, if someone told you they were having trouble accessing the Internet from their laptop at home, what is the first thing you would do when you showed up? Try to ping their router, a website, or another computer in their network. You know you would! This has always been a wonderfully quick and easy way to test whether or not you have network traffic flowing between two endpoints. The same troubleshooting procedure exists in all workplaces and corporations. I have even seen many monitoring tools and scripts utilize the results of whether or not a ping replies to report on whether or not a particular service is up and running. If you get a ping reply, it's working and if it times out, it's down, right?

Not necessarily. The problem we are here to address today is that more and more networks and routers are starting to block ICMP traffic by default. Ping traffic is ICMP traffic. This means that you can no longer take your ping test results to the bank. If your network connection traverses a router or firewall that blocks ICMP, your ping test will time out, even if the service is up and running. Windows Firewall even blocks ICMP by default now. So if you bring a new server online in your network and give it an IP address, you may notice that attempting to ping that new server results in timeouts. There is nothing wrong with the server, and it is capable of sending and receiving network traffic, but the local firewall on that server is blocking the incoming ping request.

I only lay out this information to let you know that `ping` is no longer the best tool for determining a connection between machines. Today's recipe will introduce a tool that has been around for a long time, but that I don't find many administrators taking advantage of. This is the **Telnet Client**, which I use on a daily basis. I hope that you will too!

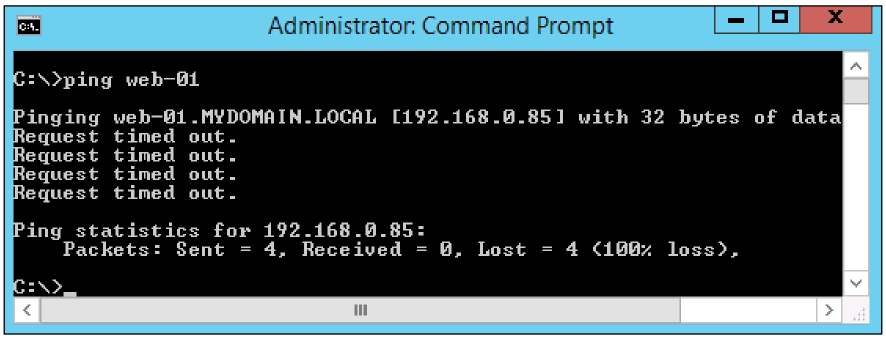
## Getting ready

We have a Server 2012 R2 web server that has a website running. It is also enabled for RDP access and file sharing, but ICMP is being blocked by the local Windows Firewall. We are going to run some tests with a client machine against this server to try to determine which services are up and running.

## How to do it...

To start working with Telnet Client, have a look at these instructions:

1. First, just to prove our point here, let's open up a command prompt on our testing client machine and try to ping web-01 using the command `ping web-01`. Because ICMP is being blocked by the firewall, all we get is a series of timeouts.




2. Now let's use the Telnet command to accomplish some more intuitive digging into whether or not WEB-01 is online and functional. Note that Telnet Client is not available inside the command prompt by default; it is a Windows feature that must be installed.

   On the client machine we are using to test, head into **Control Panel | Programs | Turn Windows features on or off** (or **Server Manager** if your testing machine is a server) and choose to add roles or features. We want to install the **feature** called **Telnet Client**.

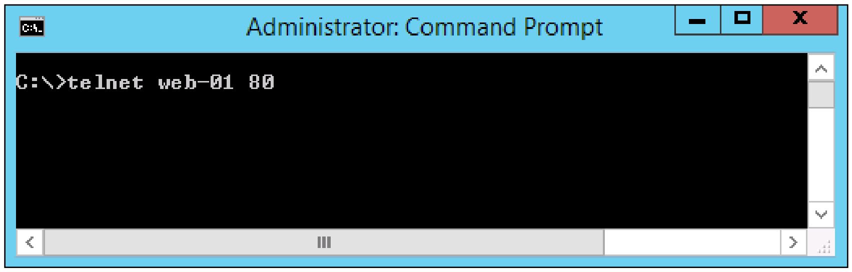
   Alternatively, you can install the **Telnet Client** feature with a simple PowerShell command:

   ```
   Install-WindowsFeature Telnet-Client
   ```

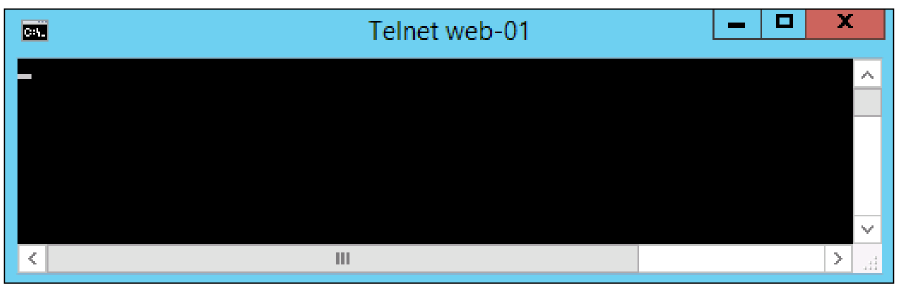
3. Now we have the `telnet` command available for use inside the command prompt. The general format of the command goes like this: `telnet <server> <port>`. When you run this command, you are effectively saying "Let's try to create a connection to this server name, on this particular port."

4. Even though we cannot ping WEB-01, let's try to use `telnet` to open a connection to port 80, which is the website that we have running. The command is:

   **`telnet web-01 80`**




5. When we press *Enter*, the command prompt window changes to a flashing cursor. This is your confirmation that telnet was able to open a successful connection to port 80 on the WEB-01 server.




6. Now try using the command `telnet 192.168.0.85 3389`. This also results in a flashing cursor, indicating that we successfully connected to port 3389 (RDP) on IP address 192.168.0.85. This is the IP address of WEB-01. I wanted to show this to point out that you can use either names or IP addresses with your `telnet` commands.
7. And finally, how about `telnet web-01 53`? This one results in a timeout, and we do not see our flashing cursor. So it appears that port 53 is not responding on the WEB-01 server, which makes sense because port 53 is commonly used by DNS, and this is a web server, not a DNS server. If we were to query one of our Domain Controllers that is also running DNS, we would be able to make a successful telnet connection to port 53 on those guys.

> Telnet queries work with TCP traffic, which covers most services that you will be polling for. Telnet does not have a connector for UDP ports.

## How it works...

Telnet is a simple but powerful command that can be run to query against particular ports and services on your servers. When trying to determine whether or not a particular service is available, or when trying to troubleshoot some form of network connectivity problem, it is a much more reliable tool than using a simple `ping` request. If you have been thinking about building some kind of script that programmatically reaches out and checks against servers to report whether they are online or offline, consider using `telnet` rather than `ping` so that you can query the individual service that the system is providing by using its particular port number.

# Using the Pathping command to trace network traffic

When building or troubleshooting a network connection, it is often very beneficial to be able to watch the path that your packets take as they make their way from source to destination. Or perhaps they never make it to the destination and you want to figure out how far they do travel before stopping so that you can focus your work efforts in that area.

One command that has been used by network admins for years is trace route (`tracert`), but the output contains some information that is often unnecessary, and the output is missing one large key ingredient. Namely, trace route shows the first hop as the first router that you traverse and does not show you what physical NIC the packets are flowing out of. Granted, many times you only have one NIC so this is obvious information, but what if you are working with a multi-homed server and you are simply checking to make sure packets for a particular destination are flowing out of the correct NIC? What if we just want to double-check that some route statements we added are working properly? Cue `Pathping`. This command has been around for a long time, but is virtually unknown. It shows the same information that `tracert` does, except it saves the information about time between hops and some other details until the end of the output. This allows you to focus on the physical hops themselves in a clear, concise manner. More importantly, it shows you our key ingredient right away—the NIC that your packets are flowing out of! Once I discovered this, I left `tracert` behind and have never looked back. `Pathping` is the way to go.

## Getting ready

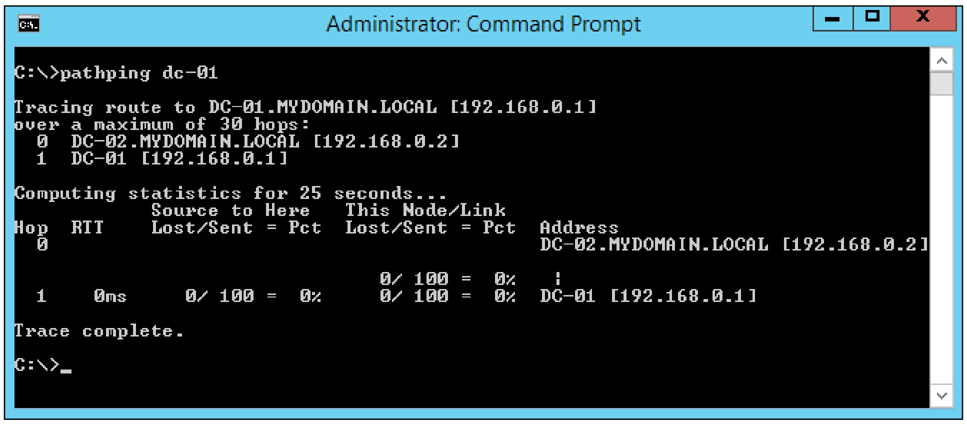
Not much to get ready for this one. All we need is a server with a network connection and a command prompt window. `Pathping` is a command that is already available to any Windows Server; we just need to start using it.

## How to do it...

The following two steps get you started with `Pathping`:

1. Open **Command Prompt** on your server.
2. Type `pathping <servername or IP>`.

Your output will be as follows:




## How it works...

`Pathping` is a networking tool that allows you to watch the path that your packets are taking as they make their way to the destination. Similar to trace route, it is much less commonly known but, in my opinion, gives a better layout of the same data. It is a command that should be added to your regular tool bag and vocabulary, right alongside `ping` and `telnet`.

# Setting up NIC Teaming

**Teaming** your network cards basically means installing two NICs onto the same server, plugging them both into the same network, and joining them together in a "team". This gives you NIC redundancy in case of a failure, and redundancy is always a great thing! Sounds simple, right? Well, with Windows Server 2012 R2 it finally is. This seemingly easy task has always been challenging to put into practice with previous versions of the operating system, but with 2012 R2 we can finally do it properly from a single interface and actually count on it to work as we expect it to.

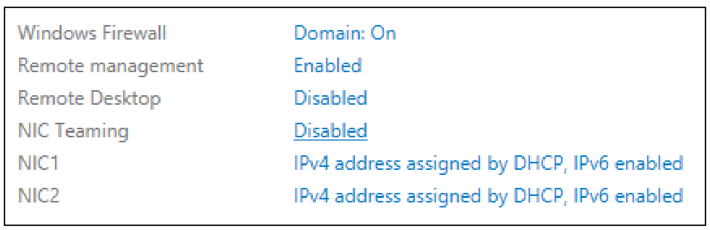
## Getting ready

We are going to set up a NIC team on a Windows Server 2012 R2 machine. There are two NICs installed onto this server, neither of which have yet been configured.

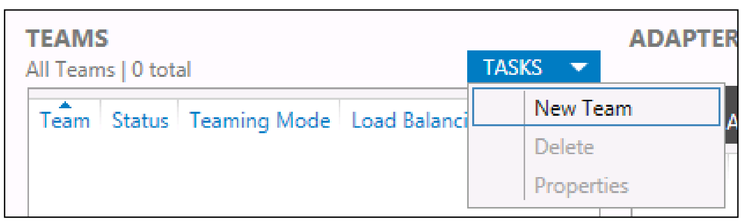
## How to do it...

With the following steps, start teaming up:

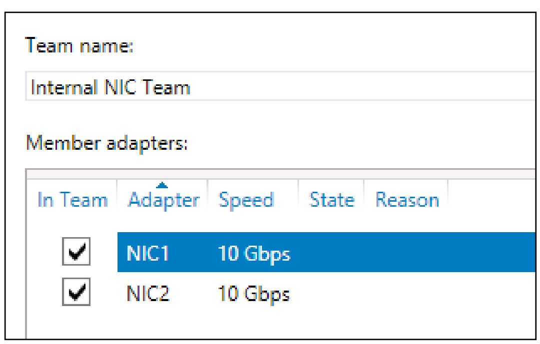
1. Open up **Server Manager** and in the left pane go ahead and click on **Local Server**.
2. Near the middle of the screen, you will see a section marked **NIC Teaming**. Go ahead and click on the word **Disabled** in order to launch the NIC Teaming screen as follows:



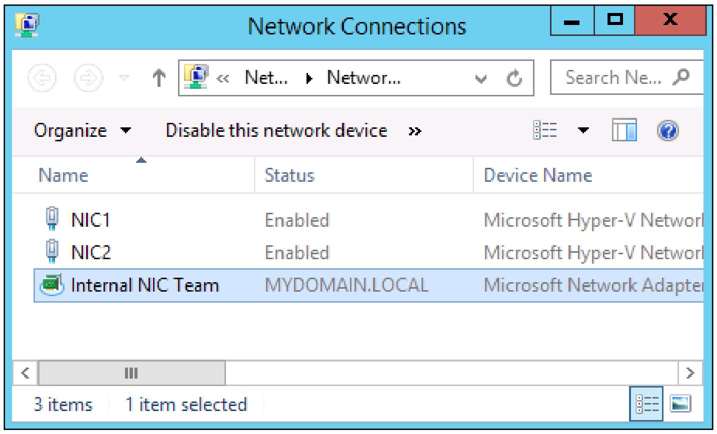

3. Down in the **TEAMS** section, drop down the **TASKS** menu and click on **New Team**.




4. Define a name for your new team and choose the two NICs that you want to be a part of it.



5. That's it! NIC1 and NIC2 are now successfully joined together in a team, and will work in tandem to make sure you are still connected in the event of a failure.
6. If you make your way to the regular **Network Connections** screen, where you define IP address information, you will see that you now have a new item listed beneath your physical network cards. This new item is the place where you will go to define IP address information that you want the server to use.




> You can create more than one team on a server! When setting up a multi-homed server with two network connections, you could easily make use of four NICs and create two teams, each containing two physical network cards.

## How it works...

Creating NIC teams is a pretty easy process that you should practice as time permits. This option for redundancy has never been very popular, I believe because it had some stability problems in earlier versions of the Server operating systems. Now that we have Windows Server 2012 R2 available to us, and the process is so straightforward to configure it, I fully expect that NIC Teaming will become a standard procedure for administrators as they build every new server.

Another benefit of and reason for setting up NIC teaming is additional bandwidth. This may be yet another reason for which you start setting up your own servers with NIC teams. Keep in mind, if you are looking to implement teaming on a large scale, there is a limit of 32 NICs that can be joined to a team, and an additional limit of 32 teams that can be created on a single server.

# Renaming and domain joining via PowerShell

Every server that you build will need a hostname set, and most likely will need to be joined to your domain. We are all familiar with doing these things with the mouse using System properties, but have you ever thought of using a command interface to do these tasks quickly? Let's work together to discover how PowerShell can once again help make these necessary tasks more efficient.

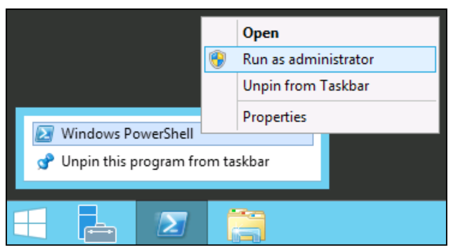
## Getting ready

We have just finished turning on a new Windows Server 2012 R2 machine. Immediately following the mini-setup wizard in order to get logged into Windows, let's now use PowerShell to set our hostname and join the system to our domain.

## How to do it...

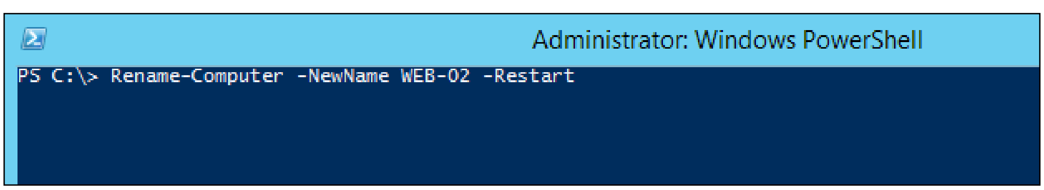
Follow these steps to rename and domain join this new server with PowerShell:

1. Right-click on your PowerShell icon from the Taskbar and choose **Run as administrator**:



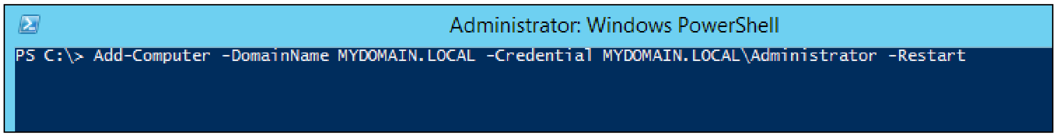

2. In order to rename our new server to `WEB-02`, input the following command. Using the `–Restart` flag will ensure that our server reboots following the name change:

```
Rename-Computer -NewName WEB-02 -Restart
```

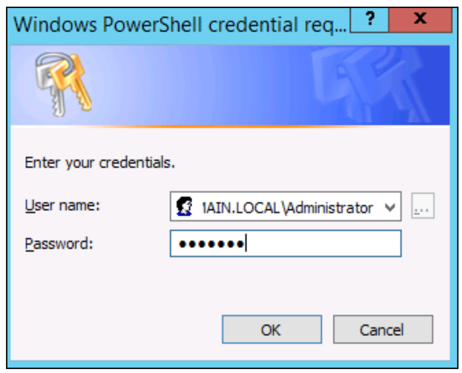


3. That's it for renaming! Now that our WEB-02 server has rebooted, open PowerShell again and use the `Add-Computer` command in order to join it to our domain:

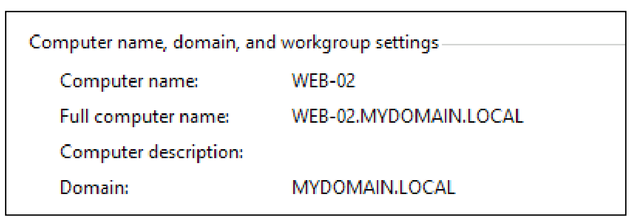
```
Add-Computer -DomainName MYDOMAIN.LOCAL -Credential MYDOMAIN.
LOCAL\Administrator -Restart
```




4. Since we specified an account to use as credentials when joining the domain, we are prompted to supply the password. As soon as you enter the password, the server will be joined to the domain and immediately restart to complete the process.




5. Following the reboot, you can see in System properties that our server is now appropriately named and domain joined.




## How it works...

Through a couple of quick PowerShell cmdlets, we can rename computers and join them to our domain. In fact, these functions are even possible without ever logging into the console of the server. There are parameters that can be added to these cmdlets that allow you to run them remotely.

For example, you could run the PowerShell commands from a local desktop computer, specifying that you want to run them against the remote server's IP address or name. By performing the functions this way, you never even have to log into the server itself in order to name and join it. See the links in the section below for additional information on these parameters.

## See also

Take a look at the following links for even more detailed information about the `Rename-Computer` and `Add-Computer` cmdlets that we used in this recipe:

- `http://technet.microsoft.com/en-us/library/hh849792.aspx`
- `http://technet.microsoft.com/en-us/library/hh849798.aspx`

# 4

# Working with Certificates

Understanding certificates is something that I avoided for many years in my technology career. For many facets of IT, you never had to deal with them. That was for the networking guys, not anybody doing development or desktop support. Times have changed, and a solid understanding of the common certificate types is quickly becoming an ability that anyone in support should possess. More and more security is being focused around certificates and, with the exponential increase in the amount of applications that are served via the Web, understanding the certificates that protect these services is more important than ever.

Almost anyone who has set up a website has dealt with SSL certificates from a public **Certification Authority** (**CA**), but did you know that you can be your own CA? That you can issue certificates to the machines in your network right from your own CA server? Follow along as we explore some of the capabilities of Windows Server 2012 R2 running as a Certification Authority server in your network. Our work today will be covering the following topics:

- Setting up the first Certification Authority (CA) server in a network
- Building a Subordinate Certification Authority server
- Creating a certificate template to prepare for issuing machine certificates to your clients
- Publishing a certificate template to allow enrollment
- Using MMC to request a new certificate
- Using the web interface to request a new certificate
- Configuring Autoenrollment to issue certificates to all domain joined systems

# Introduction

When getting to know a new customer and network as part of my day job, I generally find that one of two things are true. Either they don't have a **Certification Authority** (**CA**) server, or they do but it isn't being used for anything yet. Most folks know that certificates are upcoming and in demand; new technologies are released all the time that require a fairly large use of certificates. Technologies such as Lync, System Center, DirectAccess, or even just building a website almost always require the use of a certificate in today's world. Jumping into a project to deploy almost any new system these days will quickly bring you to the realization that a knowledge of certificates is becoming required. Even in places where they aren't required, they are usually still recommended to make the solution more secure or to adhere to best practices.

Together we are going to build a **Public Key Infrastructure** (**PKI**) environment inside our network and use it for some common certificate issuing tasks. By the end of this chapter, you should be comfortable with creating a PKI in your own environment, which will prepare you for any requirements you may encounter when working with certificate-based technologies.

# Setting up the first Certification Authority (CA) server in a network

The first hurdle to overcome when wanting to start certificate work is putting the server into place. There are many valid questions to be answered. Do I need a dedicated server for this task? Can I co-locate this role on an existing server? Do I need to install an Enterprise or Stand-alone CA? I've heard the term "offline root", what does that mean? Let's start with the basics and assume that you need to build the first Certification Authority server in your environment.

In an Active Directory domain network, the most useful CA servers are of the Enterprise variety. Enterprise CA servers integrate with Active Directory, making them visible to machines in the network and automatically trusted by computers that you join to your domain. There are differing opinions in the matter of best practices when setting up a series of CA servers. For example, there is a good test lab guide (referenced at the end of this recipe) published by Microsoft, that walks you through setting up a Stand-alone Root CA, a Subordinate Enterprise CA, and then taking the Stand-alone Root offline. An advantage to this is that certificates are issued from the subordinate, not directly from the root; thus, if certificate keys are somehow compromised in the environment, the root CA is completely offline and unavailable so that it cannot be compromised. In this situation, you could wipe out the subordinate and the certificates it has published, bring up the offline root, build out a new subordinate, and be back in business publishing certificates without having to regenerate a new root CA server.

Given the above best practice, or as defined by some anyway, it is surprising that I quite rarely see offline root CAs in the field. Almost never, in fact. And in some of the cases where I have, the existence of an offline root CA has caused problems. Just as an example, when deploying a UAG DirectAccess infrastructure with one-time-password (OTP) capabilities in a customer environment, it was discovered that in order to make the OTP work correctly, the offline root CA had to be brought back online. This wasn't in the best interests of the way the PKI had been established, and so instead we had to implement a second certificate environment to be a Stand-alone Root with two intermediaries in order to maintain an online root CA for the purpose of the OTP certificates. This caused big delays in the project as we had to build the three new servers necessary just to get the certificates published in the correct way, and created a much more complex certificate infrastructure to support afterwards.

If the description above confused you, good—because it's kind of a messy setup. If the company had instead been running on online root CA server in the first place, none of this extra work would have been necessary. I'm not advocating that an Enterprise Root CA that remains online all the time is the best way to do certificates, but it will cause you the lowest amount of problems and there are many companies that operate their production CA environments in exactly this way.

Another field observation is that most small- or medium-sized companies do not take the offline root CA approach. In fact, I find that many small businesses need to co-host servers in order to save resources and have their CA role installed onto a server that is also performing some other task. Many times, the CA role is installed onto a Domain Controller. This makes some sense actually, because the Enterprise CA services are so tightly integrated with Active Directory. So our recipes here, since I don't want them to consume a chapter's worth of pages, will reflect that scenario. We are going to utilize existing DCs in our network and place the CA roles onto them. This is done to simplify our examples, but may not be right for your environment. Make sure to do some reading from the links provided at the end of this recipe, as they should provide you with information that is helpful to make the right decisions about what certificate server setup is best suited for your network.

## Getting ready

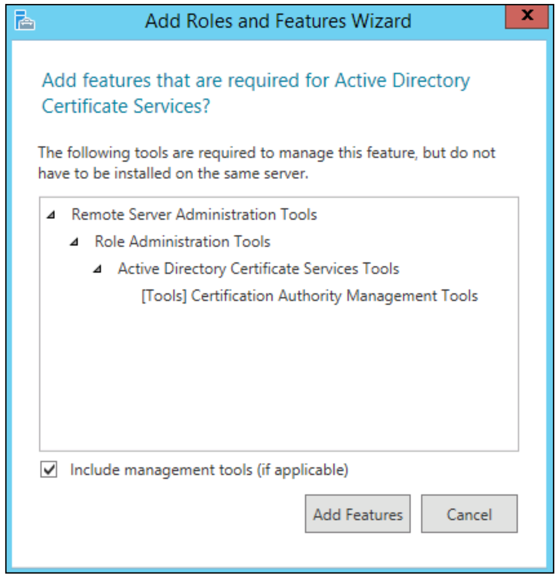
Our Certificate Services role is going to be added to an existing Domain Controller in the network. We will be using DC-01, which is a Server 2012 R2 server that is already running the Active Directory Domain Services and DNS roles.

## How to do it...

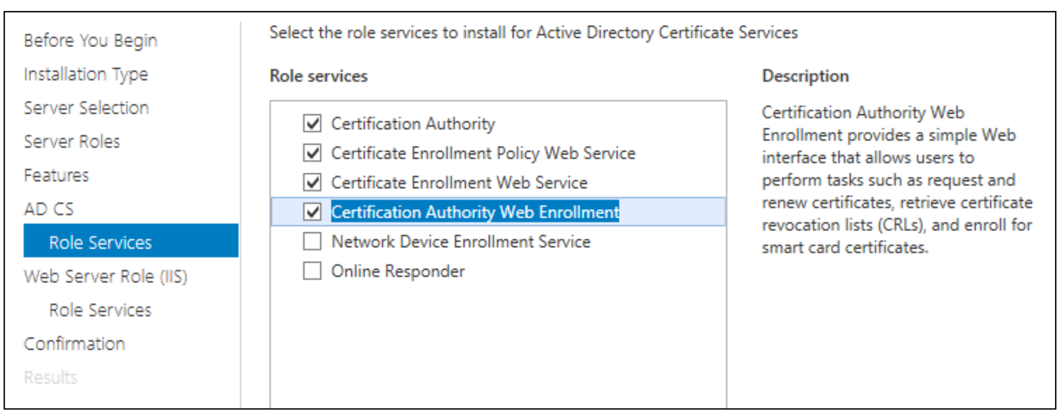
To install Active Directory Certificate Services onto your Server 2012 R2, use the following set of instructions:

1. Open **Server Manager** and click the link **Add roles and features**.
2. Walk through the steps, choosing the default settings. When you come to the **Server Roles** screen, select **Active Directory Certificate Services**.

3. Upon selecting the role, you will be prompted to confirm the installation of additional features. Go ahead and click on **Add Features**.




4. Click **Next** a couple of times until you come to the **Role Services** screen. Here you will see a few different options that can be used on your CA server. Since we would like to be able to request certificates from a web interface on the CA, I am going to check all of the boxes that relate to the web service. After selecting some of these boxes, you may receive additional pop-up boxes asking you to add features. Make sure to allow those features to be installed.



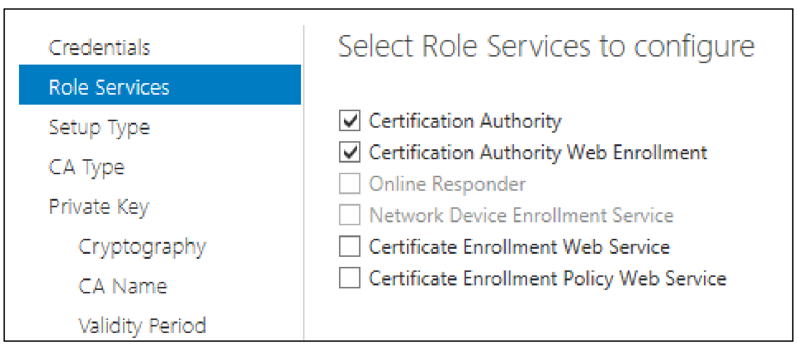
5. Click **Next** through the remaining screens until you reach the last page where you click on the **Install** button to start installation of the role.

6. Once completed, you will see a link inside your installation summary screen that says **Configure Active Directory Certificate Services on the destination server**. You can click either on this link or on the Server Manager notifications yellow exclamation mark near the top of Server Manager screen in order to continue configuring the CA role.

   In the first configuration screen, the wizard will likely auto-insert the username of the currently logged in user. As stated in the text on that screen, make sure the user you are logged in as has Enterprise Admin rights on the domain as we are planning to set this CA server up as an Enterprise Root CA.

> You can click on **More about AD CS Server Roles** at any time to read more information about the different types of CA roles and features available. For the purposes of this recipe, we will not discuss them all, but rather focus on creating our Enterprise Root CA.

7. To get certificate services rolling on our server, go ahead and check the top two options to configure **Certification Authority** and **Certification Authority Web Enrollment**.

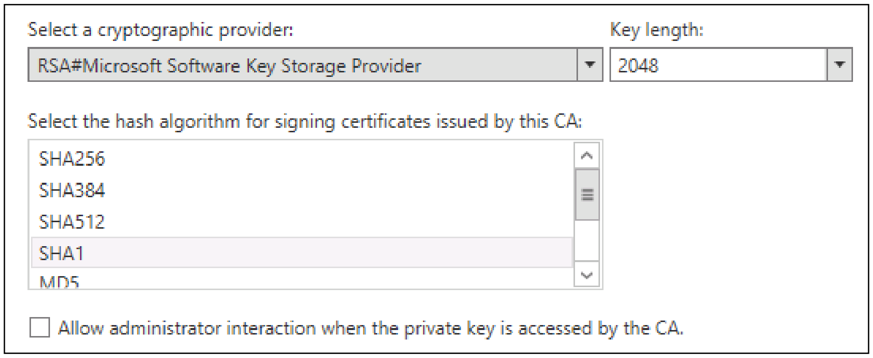



8. Choose **Enterprise CA**.

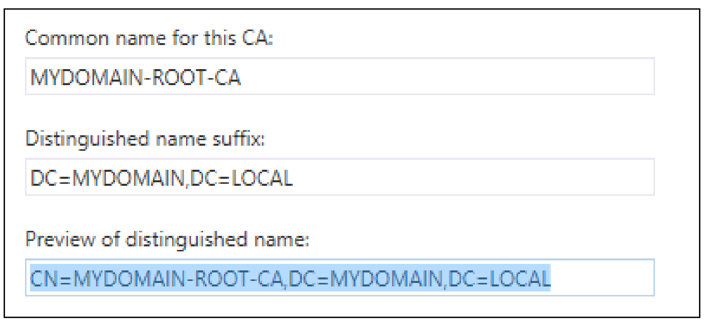
9. Choose **Root CA**. Because this is our first CA server, we need to implement a root before we can think about a subordinate.

10. Choose **Create a new private key**.

11. On the **Cryptography** screen, you have the ability to choose the kind of crypto options you can provide on your CA server. Typically, the default options will work best if you're unsure of these settings. Just make sure that the **Key length** field is set to **2048** as a minimum. This is the new industry standard for the minimum key length.

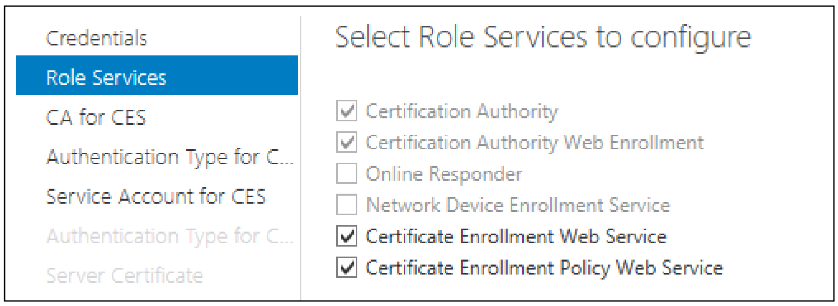



12. If desired, you may modify the **Common name for this CA**. Keep in mind, this does not have to match the hostname of the server in any way. This is the name of the Certification Authority that will show up inside Active Directory as well as inside the certificates that you issue from this CA. Typically I find that admins leave the **Distinguished name suffix** field alone.




13. Change the **Validity Period** of your root certificate if desired, but usually this is left to the default 5 years. The validity period will determine how often the root certificate has to be renewed.
14. Continue through the remaining screens, leaving the default options set in place. When this wizard finishes, your Certification Authority server is now live.
15. It is generally a good idea to schedule a reboot for this Domain Controller after such a significant role installation. Go ahead and reboot when time permits.

16. If you remember, we only selected the first two checkboxes when we started walking through the configuration wizard. This was purposefully, as you cannot accomplish all four at the same time. Following your reboot, you will notice that Server Manager is still prompting the yellow exclamation mark in the top. If you click there, you will be asked to finish the Certification Authority configuration wizards.
17. We can now choose the remaining two boxes.




18. On the **CA for CES** screen, your root CA is automatically populated. You typically want to leave this information as is and click **Next**.
19. The authentication type to access the CES is usually left at the default option **Windows integrated authentication**. If you have reason to change it, you can do so here.
20. Specify a domain service account that the CES service can use for its own communications. As stated, make sure that you add this user account into the IIS_IUSRS group on this local server. It is also a general good practice for any service account to have a non-expiring password.
21. You can simply click **Next** through the remaining screens, unless you want to utilize a different SSL certificate for this CA web interface. If so, you have the chance to specify it here. If not, it will use the root certificate that was just created as part of our CA role installation.

## How it works...

In this recipe, we installed the first Certification Authority server into our network. Our example reflected installing the role onto a Domain Controller, which is a fairly common thing to do but is not necessarily the right move for your company. As we discussed, you will want to make sure you read over some of the links listed below to help determine how many CAs you require and where they should be installed. This is one of those answers that can be different for every organization, and so I cannot make any blanket statements here that will apply for everyone. You may decide that your primary root CA should be Stand-alone rather than enterprise, and that is fine as long as it fits your needs.

We also installed the web services pieces of the role onto our primary CA because we plan to use these in upcoming recipes to issue certificates. If you are building an environment with multiple CA servers, you can determine that your root authority doesn't need the web interface; maybe only a particular subordinate CA will do that job for you. There are numerous ways that our design could play out but, through this recipe, I hope that enough information is provided so that you are comfortable with the actual process once those decisions have been made.

There are a couple of items that we did not cover in this recipe which should be pointed out. Following the steps included above will get you a CA server up and running that is ready to issue certificates, there is no doubt about that. The remainder of the recipes in this chapter reflect CAs built exactly as shown here. However, there are additional steps that can be taken in order to further customize your CA settings if you have the need. If you plan to issue SSL certificate for websites, especially if you plan to install these certificates onto web servers which are facing the Internet, then you need to familiarize yourself with the **Certificate Revocation List** (**CRL**) settings. Whenever a certificate is accessed, the client computer checks in with the CRL in order to make sure the certificate is still valid. If the certificate is not valid, or is fraudulent in some way, the CRL check will identify that compromise and disallow the connection. Particularly when publishing websites to the Internet that use certificates issued by your internal PKI, you will need to plan the publishing of your CRL so that external client computers can access it in a clean, secure fashion. Here is a great link to get you started on CRL information: `http://technet.microsoft.com/en-us/library/cc771079.aspx`

The second piece of information I would like to reference is the `CAPolicy.inf` file. This is a file that can be populated with various customization settings for your CA server. Things such as the validity period of your root certificate, information about your CRL, and whether or not you want the default certificate templates to be loaded during CA role installation. If any of these settings are of interest to you, you simply create a `CAPolicy.inf` file with the appropriate configurations and place it inside `C:\Windows` on your CA server prior to role installation. The role installation wizard will then utilize the settings inside this file during role install, and incorporate your customizations. If you do not use one of these files, it is fine and the role will be installed with some default settings in place. But if you are interested in changing some of them, check out this link for more detailed information on the `CAPolicy.inf` file: `http://technet.microsoft.com/en-us/library/jj125373.aspx`

Neither of these items, tweaking the CRL or using a `CAPolicy.inf` file, are required in order to get a certificate environment up and running. Thus, they are not included in the step-by-step configuration of the recipe itself. But I am always a fan of having all knowledge available to me on a particular subject, and so I strongly encourage you to read over these additional links provided to round out your understanding of possible functionality.

## See also

Here are some links that make for good additional reading on this subject. In order to make an informed decision about what sequence of CA servers is right for your environment, I encourage you to do as much reading on the subject as possible before proceeding in the production network.

- `http://technet.microsoft.com/en-us/library/dn786443.aspx`
- `http://technet.microsoft.com/en-us/library/dn786436.aspx`
- `http://technet.microsoft.com/en-us/library/hh831348.aspx`

# Building a Subordinate Certification Authority server

We build Subordinate CAs not really for the purpose of redundancy, like with many other kinds of servers, but because there are specific tasks that you may want to perform on a subordinate CA rather than a root CA. If you issue a lot of certificates, or different kinds of certificates, you may want to differentiate between CA servers when issuing. Perhaps you want machine certificates that are used for IPsec to be issued from IPSEC-CA, but the SSL website certificates that you issue should show as being issued from WEB-CA. Rather than building out two independent Root CAs which both have top-level rights, you should consider creating a single Root CA, maybe called ROOT-CA, and placing these two CA servers in a subordinate role, "under" the Root CA in the chain. This can also be useful for geographically dispersed networks, having Subordinate CA servers dedicated to assigning certificates for different offices or regions.

As we discussed in the previous recipe, there are certainly some best practice standards that would suggest you utilize only subordinate CAs to accomplish your certificate issuance. I don't always find that this is feasible for companies, particularly the smaller ones, but it is a good idea if you can swing it. With Subordinate CA servers online, you have the option of bringing your Root CA offline, and using the Subordinates to issue all of your certificates.

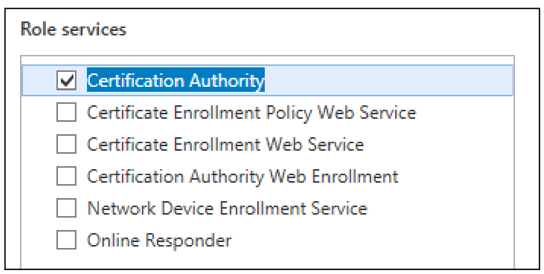
## Getting ready

We are inside a domain network, and have a single Enterprise Root CA online and running. We now require an additional server which will be joined to the CA environment as a new Subordinate CA. Since our Root CA is running on a Domain Controller, we are going to install the new Subordinate onto another Domain Controller that we have running. This is not required, but is a common practice in smaller environments.

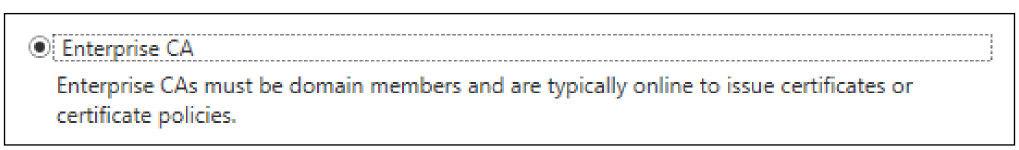
## How to do it...

To implement our new Subordinate CA server, the process will be very similar to the *Setting up the first Certification Authority (CA) server in a network* recipe. However, there are a few key differences, and that is where we will focus. Some of the specific steps may be shortened here, please refer the previous recipe for more detailed information on the specific steps and settings with regards to installation of the role.

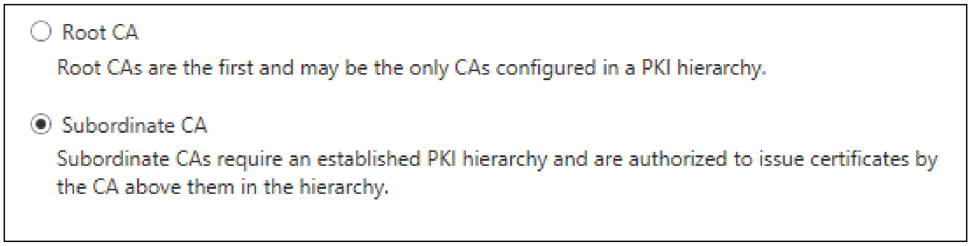
1. Login to our new server, which has already been joined to the domain. Actually, in our case it is a Domain Controller, but that is not a necessary specification.
2. Follow the steps to add the Active Directory Certificate Services role to this server.
3. When we implemented our Root CA server, we chose to install the web services as well. This will enable us to request and issue certificates from a browser inside our network. You have the option of installing these web services on the new Subordinate CA, which you would definitely do if you planned on using an offline root CA, but for our situation we are not going to do this. We will stick with only **Certification Authority** in our list of available role services.



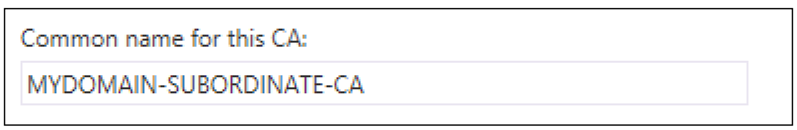

4. After the role has finished installing, go ahead and click on your link for **Configure Active Directory Certificate Services on the destination server**.
5. Input credentials as needed and choose the only option we have in the list to configure: **Certificate Authority**.
6. Here is where we start to detour off the path that we took with our Root CA creation. We are still choosing to setup an **Enterprise CA**, because we still want it to be domain-integrated.



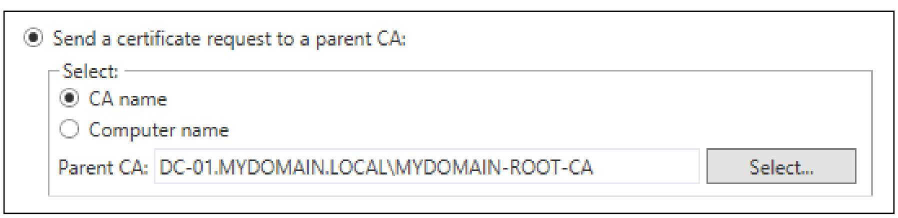
7. But instead of choosing to install a new Root CA, we are going to choose the option for **Subordinate CA**. In fact, it was already chosen for us as the default, because it recognizes that a Root CA already exists in the network. We could install another Root CA, but that is not our purpose in this recipe.




8. Choose **Create a new private key**. The only time we typically would want to use an existing private key is when rebuilding a CA server.
9. Choose your cryptography settings. These are typically going to be the same that you configured on the Root CA.
10. Name your new Subordinate CA appropriately. If you have a specific function in mind for this CA, it will be helpful to you in the future to name it accordingly.




11. Now we come to a new screen. We need to acquire a certificate from our parent CA server in order to issue certificates from this new one. Choose the option for **Send a certificate request to a parent CA**, and use the **Select...** button to choose your Root CA.




12. On the following screen, adjust the location of the certificate database files if required; otherwise, click **Next** then **Configure**.

## How it works...

Installing a Subordinate CA server in a network is very similar to implementing our first Root CA server. In our case, we simplified the installation by not having the requirement for the web services to run on the Subordinate, we will do all of those requests from the Root CA. We now have a Root CA running and a Subordinate CA running under it. For our installation, we are going to leave both online and running as we intend to issue certificates from both. We could easily run through this same process again with another new server in order to create another Subordinate CA, maybe to issue a different kind of certificate or for a different division of the company to utilize.

## See also

- The *Setting up the first Certification Authority (CA) server in a network* recipe.

# Creating a certificate template to prepare for issuing machine certificates to your clients

This recipe is the first hurdle that many new certificate admins bump into. You may have a CA server up and running, but what's next? Before you can start granting certificates to computers and users, you need to establish certificate templates that you are going to publish. You will configure these templates with particular settings, and when a certificate is requested against the template, that new certificate will be built based on the information in the template combined with the information provided by the certificate requestor.

There are some built-in certificate templates that preinstall when you add the CA role to your server. Some companies utilize these built-in templates for issuing certificates, but it is a better practice to create your own templates. There is no need to start from scratch, though. You can take one of the built-in templates, find one that comes close to meeting your needs, and tweak it to do your bidding with your particular certificate needs. This is the process we are going to be taking. We need to issue machine certificates to each of our systems in the network to authenticate some IPsec tunnels. There are a few criteria we need to meet in these certificates, and the built-in Computer template comes close to checking all the options that we need. So we will take that template, copy it, and modify it to meet our requirements.

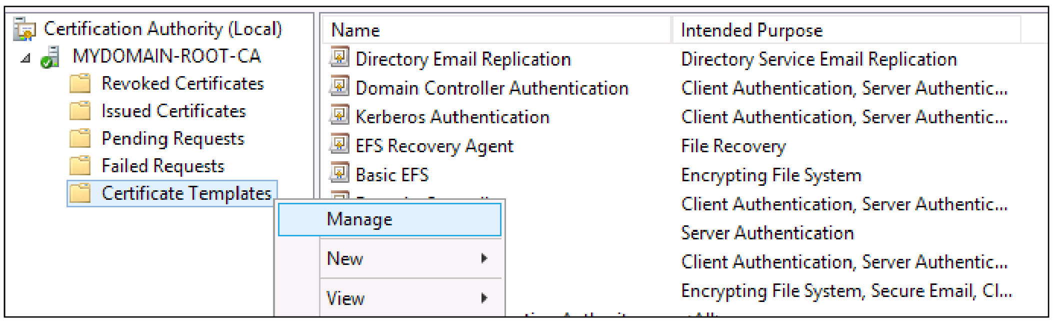
## Getting ready

This is a Server 2012 R2 domain environment with a new Certification Authority server running. We will utilize the CA console on our CA server to accomplish this work today. The new template that we create will be automatically replicated with other CA servers in the domain.

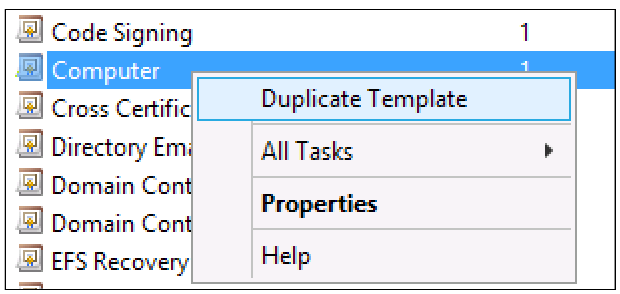
## How to do it…

The following steps will help you build a new certificate template:

1. Launch the **Certification Authority** management tool from inside Server Manager.
2. Expand the name of your CA and click on **Certificate Templates**. You will see a list of the built-in templates available to us.
3. Right-click on **Certificate Templates** and choose **Manage**.

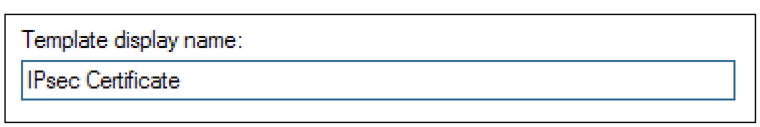



4. Right-click on the **Computer** template, and choose **Duplicate Template**.

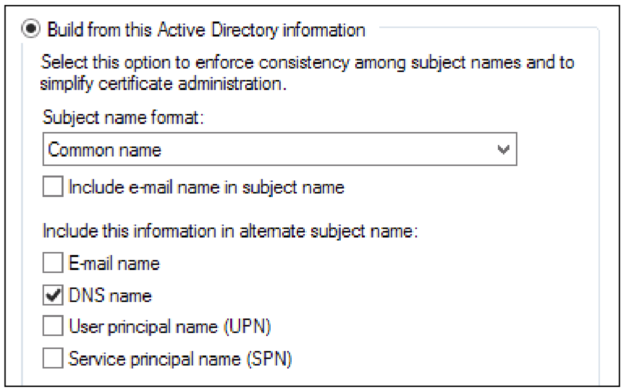



Now we adjust options within the certificate template. Any attributes that your certificates must have, you set here in the template properties. As an example, let's configure a few items that our new IPsec certificates must contain to be valid.

5. Go to the **General** tab and set the **Template display name** so that you can identify this new template we are building.



6. On the same tab, adjust the **Validity period** field to `2` years.
7. Browse to the **Subject Name** tab and set **Common name** as the **Subject name format** field. This will cause the subject name of the certificate to reflect the hostname of the computer that is requesting the certificate. Using the DNS name as the alternate subject name is another requirement that we have been given for our new certificates. You can see it checked in the screenshot below. Since we used the built-in Computer template as our starting point, this checkbox as well as other requirements that we needed covered were already taken care of for us.




8. Click **OK**. There is now a brand new certificate template in the list called `IPsec Certificate` (or whatever name you gave to yours).

### How it works...

When installing any new technology that requires certificates to be issued, your first stop should be the certificate templates on your CA server. You need to make sure that you have a template configured with the appropriate settings and switches that you need in your new certificates. By duplicating one of the built-in templates that came with our CA server, we were able to build a new template without having to configure every single option from the ground up.

## Publishing a certificate template to allow enrollment

One of the most common certificate troubleshooting tasks I encounter is figuring out why a particular certificate template is not available when the user or computer tries to request a certificate. Having created a new certificate template does not necessarily mean that you are ready to start issuing certificates based on that template. We need to "publish" our new template, so that the CA server knows that it is ready to publish out to computers and users.

There is also a security section of the template properties where you need to define who or what has access to request certificates based on that template. In this recipe, we will find those settings and configure our new certificate template so that any domain joined workstation is allowed to request a certificate from our new template.

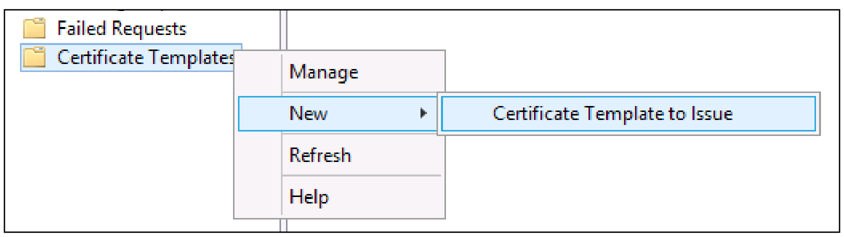
## Getting ready

We are going to use the Windows Server 2012 R2 machine that is our Enterprise Root CA.

## How to do it...

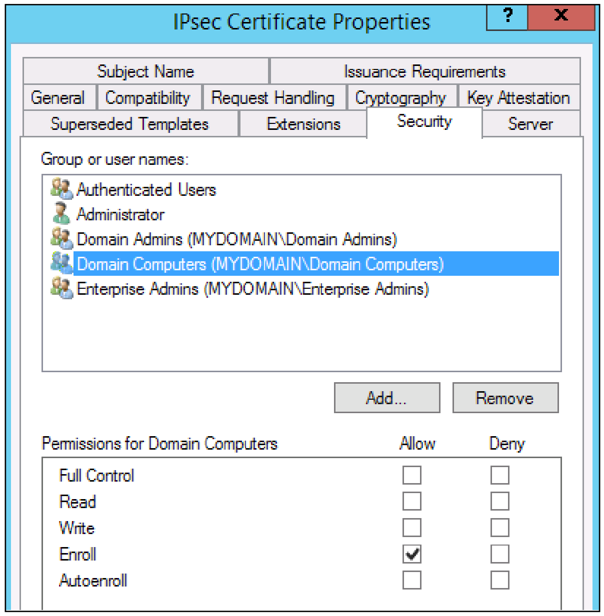
In order to issue certificates based on a particular template, we need to takes the steps to publish and adjust the security properties of that template.

1. Launch the **Certification Authority** management tool from inside Server Manager.
2. Expand the name of your CA server in the left tree.
3. Right-click on **Certificate Templates** and choose **New** | **Certificate Template to Issue**.




4. Select your new template from the list and click on **OK**.
5. Now right-click on **Certificate Templates** and choose **Manage**.
6. Find the template that you want to modify. For our recipe, we are modifying the new template called `IPsec Certificate`.
7. Right-click on the template and choose **Properties**.
8. Browse to the **Security** tab.

9. Now we need to set up permissions according to your requirements. For our particular example, we want to issue IPsec certificates to all domain joined computers so that they can be later used during IPsec negotiations inside our network. Therefore, in our permissions we add **Domain Computers** and we check the box to allow **Enroll** permissions.




## How it works...

A new certificate template doesn't do us any good without a couple of extra steps to publish that template. We need to walk through the process of specifying our new template to be issued, which is a simple option to accomplish but one that isn't immediately obvious inside the CA management console. Also, we need to make sure that the permissions we have set on our certificate template line up with the purpose for which our certificate is intended. If your user accounts are going to be requesting certificates, then you will have to add users or user groups and grant them enroll permissions. If computer accounts are going to be the ones making the requests, then make sure that the appropriate groups are entered in there with enroll rights as well.

# Using MMC to request a new certificate

The most common way that I see administrators interface with the certificates on their systems is through the MMC snap-in tool. **MMC** is short for **Microsoft Management Console**, and by using MMC, you can administer just about anything in the operating system.

Though this is perhaps a greatly underutilized tool, I only generally see it being opened for a few select tasks. Requesting certificates is one of those tasks.

We are going to use the MMC console on a new server that we have in our network. There is a new certificate template that has been created and we would like to issue one of these certificates to our new web server.

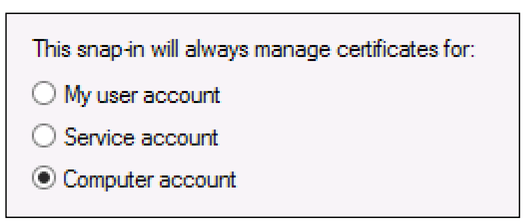
## Getting ready

A Server 2012 R2 Enterprise Root CA server is online and running in our network. On it, we have configured a new certificate template called `IPsec Certificate`. The steps have been taken to publish this template so that it may be requested from computers in our network. We are now working from a brand new web server that is also running Server 2012 R2 and joined to our domain, where we are going to accomplish the work of manually requesting a certificate from the CA server.

## How to do it...

Follow these steps to request a new certificate using the MMC console:

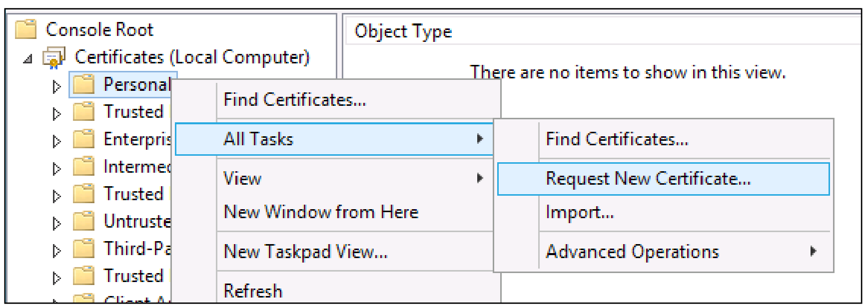
1. Open a command prompt on our new web server and type `mmc`. Then press *Enter*. Alternatively, you could open MMC from the **Start** screen.
2. Now inside the MMC console, click on the **File** menu, then on **Add/Remove Snap-in...**.
3. Choose **Certificates** from the list of available snap-ins and click on the **Add** button. This will bring up a new window with some more choices about the certificates snap-in.
4. First we need to choose whether we are opening the "User" certificate repository, or the "Computer" certificate repository. I don't generally see "Service account" used in the field. The selection here will depend on what type of certificate you are requesting. For our example, we are looking for an IPsec certificate, which needs to go in the Computer container. Choose **Computer account** and click **Finish**.



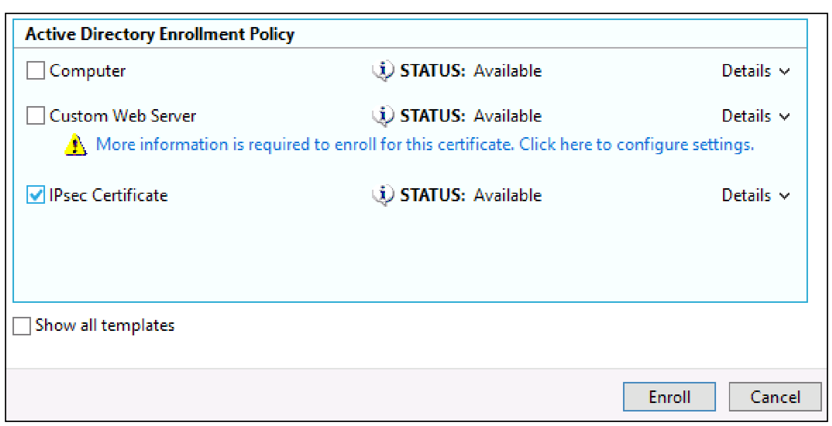

5. Leave the next option set on **Local computer** and click **Finish** again.
6. Click **OK**.

> There are also MSC launchers that can be utilized to bring you into the certificate stores even faster. Make use of these from **Start** | **Run** or Command Prompt and type the following commands:
>
> - `CERTMGR.MSC` opens User certificates
> - `CERTLM.MSC` opens Computer certificates

7. Now back inside the main MMC console, expand **Certificates (Local Computer)** and select the **Personal** folder. You can see that there are currently no certificates installed here.
8. Right-click on the **Personal** folder and choose **All Tasks** | **Request New Certificate...**.




9. Click **Next**.
10. On the **Select Certificate Enrollment Policy** screen, we are automatically selected on **Active Directory Enrollment Policy**. Simply click **Next** again to go on to the next screen.
11. Now we see a list of certificate templates that are available to us. Check the boxes for the certificates that you want to request and click **Enroll**.



> If you are expecting to see a particular template here but it doesn't show up in the list, click on **Show all templates**. This will display a list of all templates on the CA server and give an explanation for each as to why it is not currently available. This can help for troubleshooting purposes.

## How it works...

Utilizing the MMC console is a quick and easy way to request new certificates to be issued manually. In an Active Directory environment, any certificate template on the CA server that you have permissions to enroll will be visible and easy to enroll. Our example today displayed the enrollment process for a Computer certificate that we are planning to use in the future for IPsec authentication. However, there are many cases where you may want to issue user-level certificates, rather than Computer certificates. In those cases, you would want to snap-in the User account certificates, where in our example we defined Computer account certificates.

# Using the web interface to request a new certificate

Sometimes, when requesting a new certificate, you may not have access to query certificate services directly by using a tool such as the MMC snap-in. Or perhaps you want to provide a way for users to be able to request certificates even while outside the office. By enabling the web services portion of the Certification Authority role, we turn on a website that runs on our CA server. This website can be accessed from inside the corporate network and could potentially even be published out to the Internet with some kind of a reverse proxy solution.

For our recipe today, let's access the web interface that is now running on the CA server where we installed the web services part of the CA role. We will use this website to request and acquire a certificate on our client computer.

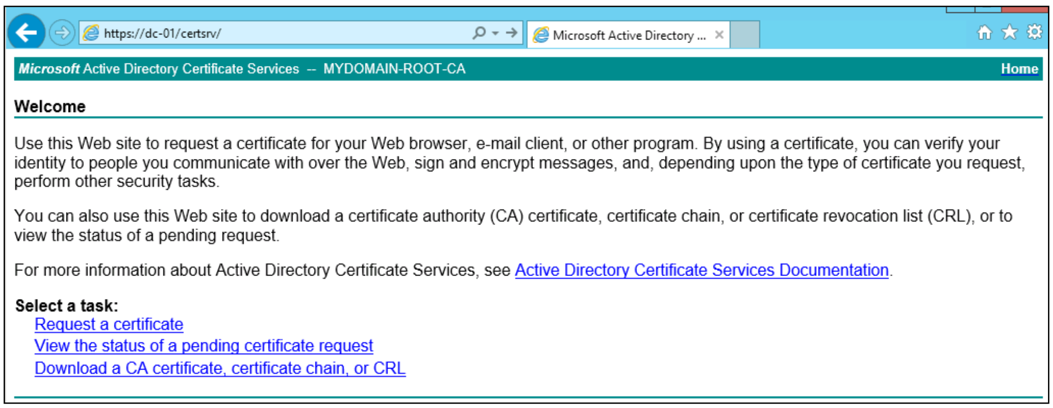
## Getting ready

Our Enterprise Root CA is a Windows Server 2012 R2 that has the Active Directory Certificate Services role installed. When we installed and configured the role, we made sure to select the options for the web service so that we could make use of it to request a new certificate.

## How to do it...

We do not have to be logged into the CA server directly to accomplish this work. Instead, we are logged into a new web server in our environment. From this web server, we take the following steps:

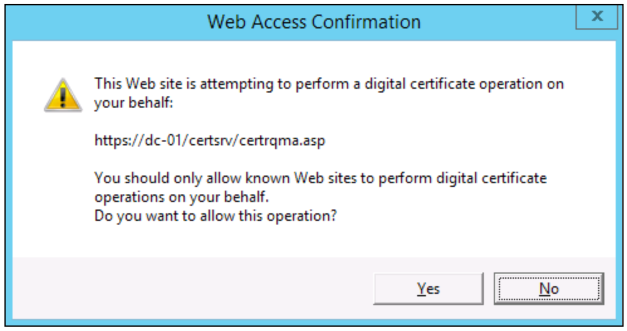
1. Open Internet Explorer and browse to https://<CAServerName>/CertSrv/. In our case, it is https://DC-01/CertSrv/.



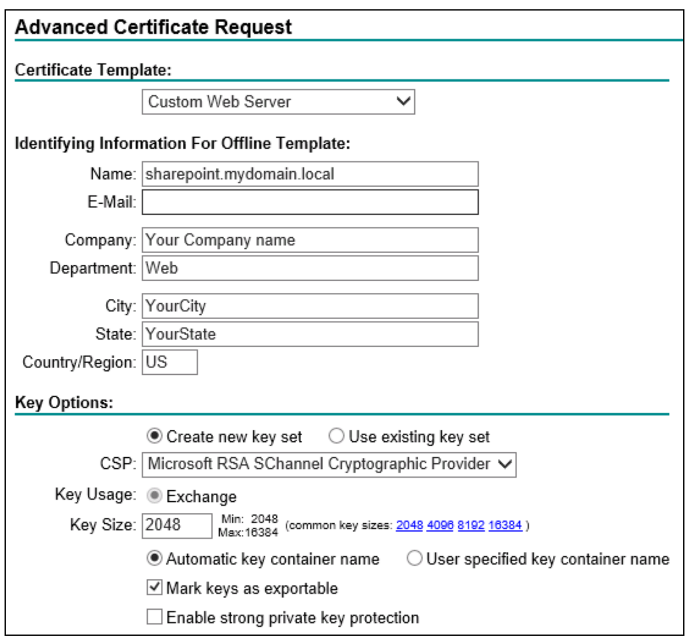

> Make sure you opt to access the site using HTTPS or you will not be allowed to finish requesting a certificate later during the wizard.

2. Click on **Request a certificate**.
3. You will see there is a pre-built request in there for acquiring a user certificate. For one of those, you simply click on that link, then click **Submit** on the next screen. However, to dig a little deeper with our recipe, we are going to request an SSL certificate, not a user certificate. To start the process, click on **advanced certificate request**.
4. Choose **Create and submit a request to this CA**.

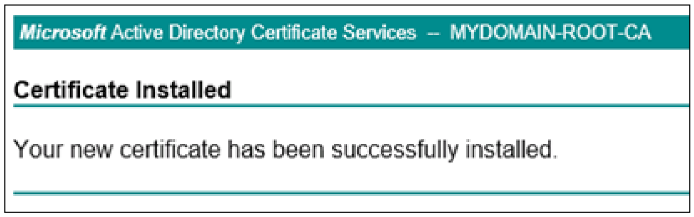
5. Click **Yes** if prompted with the following message:




6. Choose the **Certificate Template** that you would like to use in order to accomplish your certificate request. On my Root CA server where the web services are installed, I set up a new template that I duplicated from the Web Server template with my specific certificate requirements. I called this template **Custom Web Server** and have published it to be available for enrollment.
7. Because this is an SSL certificate, I need to populate the regularly requested information. My website name and company contact info is entered here.
8. The rest of the options available to change are already configured as I want them to be. This is because, when I set up my Custom Web Server template, I already specified all of these item defaults. Here is my request:



9. Click **Submit**.
10. Your browser will spin for a minute while the CA server creates the new certificate based on the information that you entered. When it is finished, you should have a link to click on called **Install this certificate**. Go ahead and click that link.




## How it works...

Running the web service on your CA server can be beneficial because it allows another method of requesting certificates. In this recipe, we were able to very quickly pull open our CA certificate requesting web page and walk through some simple steps. This enabled us to download a new certificate that we are planning to use with our new web server's SharePoint site.

Because our web server is inside the corporate network, we could have also accomplished this request right from the Certificates MMC console. However, if our web server had been in a different building separated by networking equipment and firewalls, this might not have been an option for us. Or if we were trying to acquire a certificate from another machine that didn't have the MMC access for one reason or another, this web service is a nice way to accomplish the same task.

# Configuring Autoenrollment to issue certificates to all domain joined systems

A lot of the new technologies requiring certificates to be used for authentication require those certificates to be distributed on a large scale. For example, if we want to use the Computer certificate for DirectAccess authentication, we need to issue a certificate to every DirectAccess client computer. This could be thousands of laptops in your network. If we want to start encrypting traffic inside the network with IPsec and require certificates to be distributed for that purpose, you potentially would need to issue some kind of machine certificate to every computer inside your network. While you could certainly issue each by hand using either the MMC console or the CA web interface, that doesn't sound like an awful lot of fun.

Enter **Autoenrollment**. We can turn on this feature, which is sort of like flipping a switch in Active Directory, and in doing so we can tell AD to issue certificates automatically to the computers. Even if we need to get them to every domain joined system we have. Let's work together through this recipe to turn on this option and test it out.

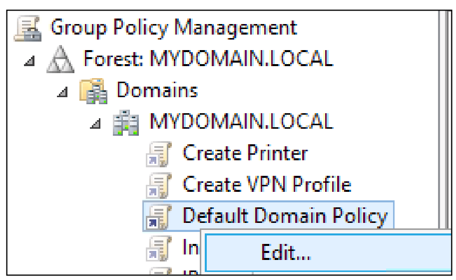
## Getting ready

We are working inside a Windows Server 2012 R2-based Active Directory domain. We also have a Server 2012 R2 Enterprise Root CA running in this network. The actual work that we will be accomplishing is going to be performed from a Domain Controller.

## How to do it...

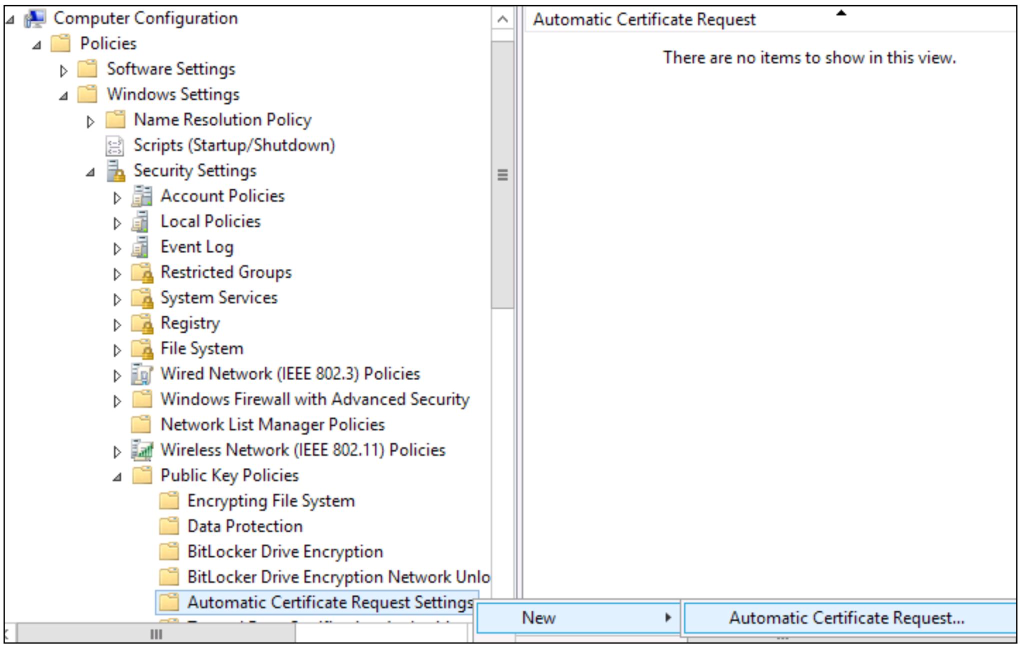
To enable Autoenrollment in your domain, take a look at these instructions:

1. Open **Group Policy Management** from the **Tools** menu of Server Manager.
2. Expand the **Forest: MYDOMAIN.LOCAL** and **Domains** folders, then expand your specific domain name.
3. Right-click on **Default Domain Policy** and choose **Edit...**.

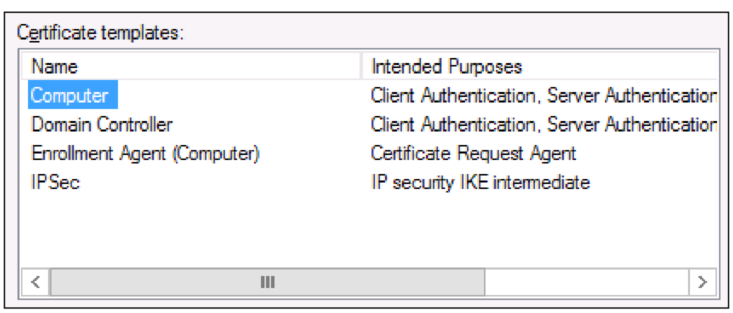



4. Navigate to **Computer Configuration** | **Policies** | **Windows Settings** | **Security Settings** | **Public Key Policies** | **Automatic Certificate Request Settings**.

5. Right-click on **Automatic Certificate Request Settings** and choose **New** | **Automatic Certificate Request...**.




6. Click **Next**.
7. Choose the type of certificate that you want to issue. For our recipe, we are going to issue **Computer** certificates. Click **Next**.




8. Click **Finish**.

## How it works...

By modifying the Default Domain Policy to issue this new automatic certificate request, we cause each of our domain computers to automatically request and grab a machine certificate. This new certificate is built from the Computer template, as we chose that option inside the policy. The GPO settings that we modified here were the Automatic Certificate Request Settings that I have found in the field to work well. However, as with many functions that can be performed inside Group Policy, there are multiple ways to issue autoenrollment of certificates in there. If you have come across or read something about a different way to set up your enrollment, by all means utilize whichever functionality works best for your own environments.

If you reboot a few domain joined machines in your network, you will notice that when they come back online, there will be a new certificate sitting in the computer's personal certificate store. Sit back and wait a few hours, and they will have rolled around to everybody automatically. If you don't like waiting for Group Policy to refresh, you can open Command Prompt on some of those computers and issue the `gpupdate /force` command to manually refresh the policies and pull down the certificate.

Keep in mind that you don't have to modify the Default Domain Policy, as that was just for our example. You could easily set up a separate GPO that is more narrowly filtered and apply a certificate issuance policy onto only a particular computer or groups of computers. This is more than likely what you would want to do as a better practice in your own environment.

# 5
# Internet Information Services

Websites and web services are used for everything these days. With the evolution of "The Cloud" we are accessing more and more via web browsers than we ever have before. The cloud can mean very different things to different people, but what I see most commonly in the enterprise is the creation of private clouds. This generally means a collection of web servers that are being used to serve up web applications for the company's user population to work from. Sometimes the private cloud is right onsite in a company datacenter, sometimes it is in a co-location, and sometimes it is a combination of a local datacenter and a true cloud web service provider such as Azure. Whatever defines a private cloud for you, one variable is the same. Your cloud consists of web servers that need to be managed and administered.

For any Microsoft-centric shop, your web servers should be running Windows Server with the **Internet Information Services** (**IIS**) role installed. IIS is the website platform in Windows Server 2012 R2, and with it we can run any kind of website or web service that we need. The intention of this collection of recipes is to give you a solid foundation to understand the way that websites work within IIS. Even if you don't normally set up new web services, you may very well have to troubleshoot one. Becoming familiar with the console and options, and just understanding the parts and pieces, can be hugely beneficial to anyone administering servers in a Windows environment. In this chapter we will cover the following recipes:

- Installing the Web Server (IIS) role with PowerShell
- Launching your first website
- Changing the port on which your website runs
- Adding encryption (HTTPS) to your website
- Using a Certificate Signing Request (CSR) to acquire your SSL certificate
- Moving an SSL certificate from one server to another

- Rebinding your renewed certificates automatically
- Using host headers to manage multiple websites on a single IP address

# Introduction

If you have been reading through this recipe book from start to finish, you probably noticed that we utilize our "new web server" a lot when testing or rounding out tasks in our infrastructure. So our web server has a whole bunch of things on it that have been pushed down as a result of regular network tasks that we did, but we aren't doing any actual web serving with it yet!

We are going to assume for most tasks in this chapter that the role for IIS is already installed onto the web server. This role is specifically called Web Server (IIS) in the list of roles, and there are numerous additional features that we can add to IIS. For all of our recipes, we only need the defaults added, the ones that are selected automatically when installing the role. That role installation is the only thing a Windows Server 2012 R2 box needs in order to serve up web pages to users, other than a little bit of knowledge on how to get the site doing what you want it to do. In order to get the role installed properly, make sure to stop by the *Installing the Web Server (IIS) role with PowerShell* recipe in order to put that component into place. Let's get familiar with some of the common tasks in IIS.

# Installing the Web Server (IIS) role with PowerShell

If you haven't started using PowerShell to accomplish some of your regular Windows Server tasks, do it now! PowerShell can be used in Windows Server 2012 R2 to accomplish any task or configuration inside the operating system. I am a huge fan of using the keyboard instead of the mouse in any situation, and saving scripts that can be used over and over to save time in the future.

In this recipe, we are going to explore the `Install-WindowsFeature` cmdlet that can be used to add a role or roles to your Server 2012 R2. Since we are discussing IIS in this chapter, let's take our newly created web server and use PowerShell to place the Web Server (IIS) role onto it.

## Getting ready

There is a new Windows Server 2012 R2 web server in our environment called WEB-02. Let's use PowerShell on this machine in order to install the IIS role.

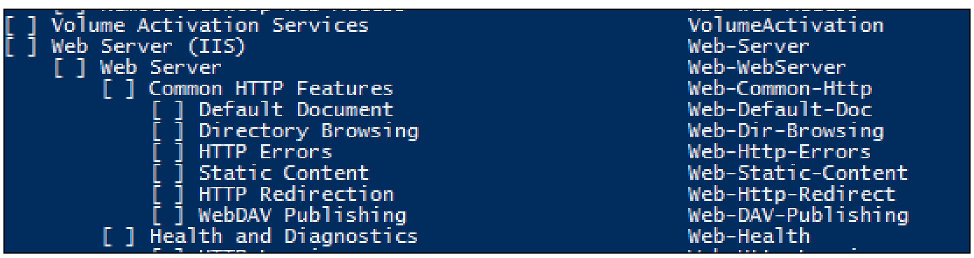
## How to do it...

To add the Web Server (IIS) role to WEB-02 via PowerShell:

1. Log in to WEB-02 and open a PowerShell prompt; make sure to run it as administrator.
2. All we have to do is run the proper cmdlet specifying the role name, but the specific name of our role evades me. Well, not really, but I thought this would be a good opportunity to explore another command that will help us see a list of the available roles to be installed. Type the following to see the list of roles:

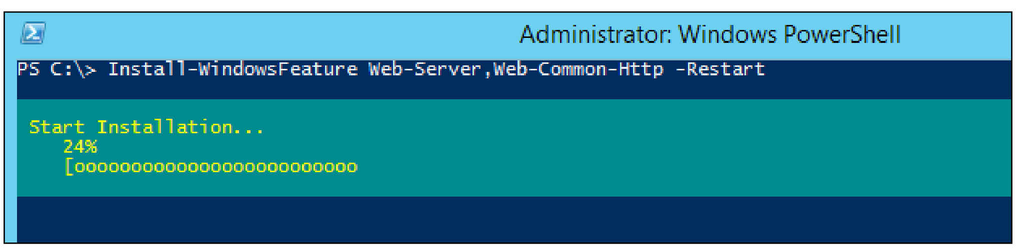
   **`Get-WindowsFeature`**

   Whoa! There's a big list of all the roles and features that can be installed onto this server. Scrolling up, I can see `Web Server (IIS)` in the list, and it looks like the role name is `Web-Server`. I am going to keep that name in mind and, since we have the ability to install multiple items at the same time, I am also going to note `Web-Common-Http` in order to install the common HTTP features when I install the role.

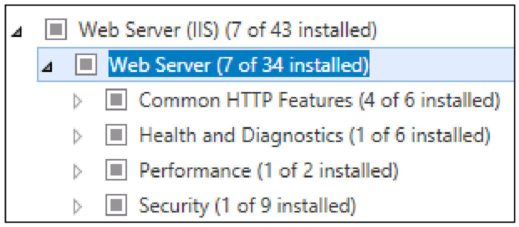



3. Now we need to build out the PowerShell command to install these two items:

   **`Install-WindowsFeature Web-Server,Web-Common-Http -Restart`**



4. Installation succeeded! Just to double-check for the sake of our recipe, if we navigate through the GUI to see the installed roles and features, we can see that the items we configured via PowerShell are fully installed.




## How it works...

We can use the `Install-WindowsFeature` cmdlet in PowerShell to easily add roles and features to our servers. This one can save a lot of time compared to running through these options in the graphical wizards. For example, if you had a group of new servers that all needed to accomplish the same task, and therefore needed the same set of roles installed, you could build out one single command to install those roles and run it on each server. No need to launch Server Manager at all.

## See also

Here are some links to additional TechNet documentation on adding roles to servers, and specifically for the `Install-WindowsFeature` cmdlet. Make sure to familiarize yourself with all of the available options. Once you start using this command, I doubt you will go back to Server Manager!

- `http://technet.microsoft.com/en-us/library/cc732263.aspx#BKMK_powershell`
- `http://technet.microsoft.com/en-us/library/jj205467.aspx`

# Launching your first website

Seems like a pretty logical first step, let's get a website started! Actually, you already have one, but it's pretty useless at the moment. As soon as you finished installing the IIS role, a standard website was started automatically so that you can verify everything is working as it should. Now we want to replace that default website with one of our own, so that we can make some real use of this new server.

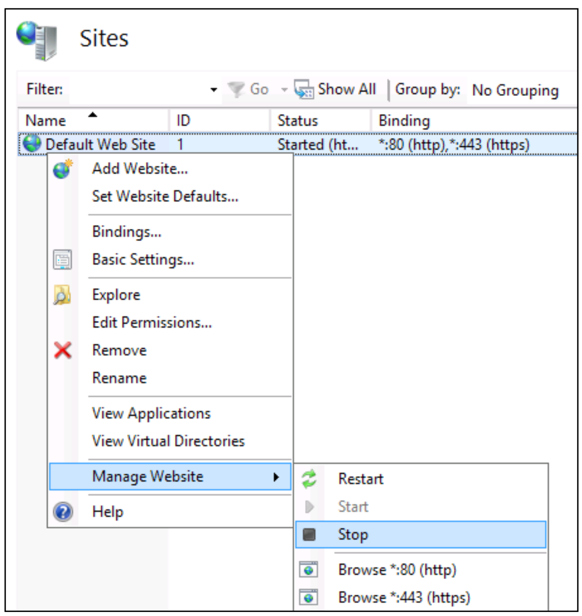
## Getting ready

We will be accomplishing all work from our new Server 2012 R2 web server. This one does happen to be domain joined, but that is not a requirement. You would be able to launch a website on a standalone, workgroup joined server just as easily.

## How to do it...

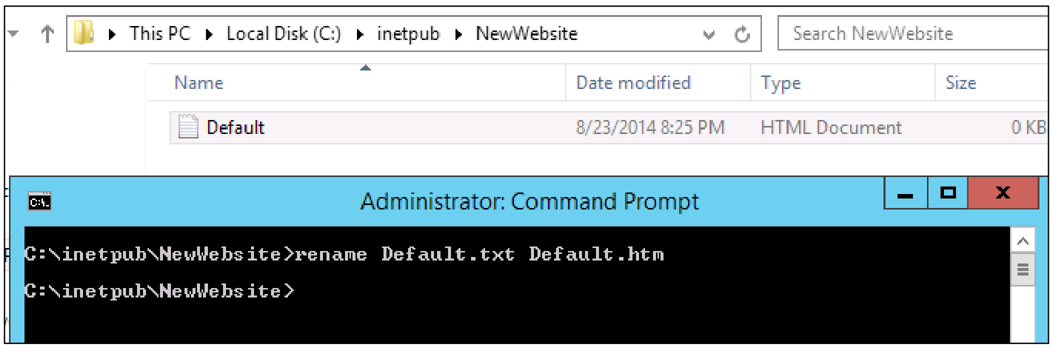
Follow these steps to start your first website on this new IIS web server:

1. Open **Server Manager** and drop-down the **Tools** menu. Then click on **Internet Information Services (IIS) Manager**.
2. In the left window pane, expand the name of your server and click on the **Sites** folder.
3. Right-click on **Default Web Site** and choose **Manage Website** | **Stop**. This will stop that automatically created website from running and getting in the way of the new website that we are about to create.

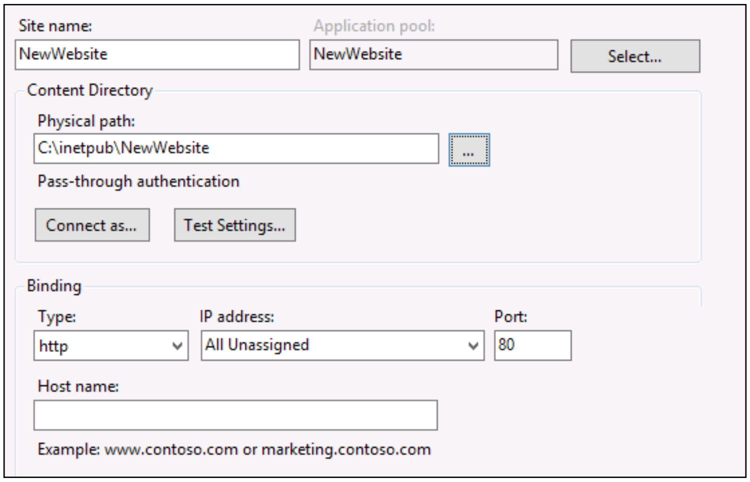



4. Before we create our new website, we will need to create an HTML webpage file that will run when users browse to the new site. Let's leave IIS Manager open for a minute and switch over to File Explorer. Browse to `C:\inetpub`. This is sort of the "home folder" that IIS creates and can be a good starting point for building your website. You do not have to create your new page within this folder; you could certainly set one up in another location, or even on a different drive altogether.

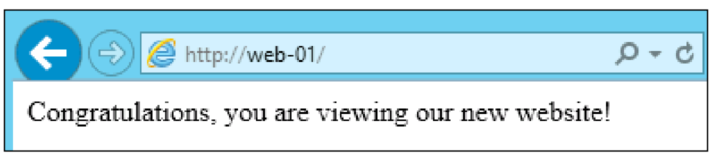
5. Create a new folder called `NewWebsite`, or whatever you want it to be called.
6. Inside this new folder, we are going to create a new file called `Default.htm`. To do this, I usually right-click and choose to create a new `.txt` file, and name this file `Default.txt`. Then I either adjust **Folder Options** so that I can see and modify file extensions, or I simply open up a command prompt window and rename the file that way. However you do it, make sure that your `Default.txt` gets changed to `Default.htm` as the final filename.




7. Now edit your new `Default.htm` file with Notepad or another text editing tool and enter some text. Thankfully, modern web browsers will properly display a page based on plain text, so that we don't have to input valid HTML code. If you know how to program in HTML, even better, though I doubt you would be reading this particular recipe. Or maybe you have a preconfigured webpage file or set of files from a software installation; you could place those into this folder as well. I am going to simply enter some text in that file that says `Congratulations, you are viewing our new website!`.
8. Head back over to IIS and let's get our site rolling. Right-click on the **Sites** folder and choose **Add Website...**.
9. Input a site name, which is just a descriptive name for your own purposes to identify the site in IIS.
10. For **Physical path**, choose our new website location, which is : `C:\inetpub\NewWebsite`.
11. If you are running multiple IP addresses on this web server and want to dedicate this new site to only run on a particular IP address, you can choose it from the **IP address** field. Otherwise, if you are running a single IP or if you want our new site to work on all IPs configured on this system, leave it set to **All Unassigned**.




12. Click **OK**.
13. From another computer on the network, open up Internet Explorer and browse to `http://<webserver>`. For our particular example, we will go to `http://web-01`.




## How it works...

Starting a new website is perhaps the simplest task that can be accomplished in IIS, but portrays the core functionality of this role. The purpose of running IIS in the first place is to publish websites. It is important to understand the location of this task and the places that you may have to reach inside the file system in order to modify or create websites of your own. Not everything is done from within the IIS Management window.

# Changing the port on which your website runs

Normally, whenever you access a website it is running on port 80 or 443. Any normal HTTP request travels over port 80, and the encrypted HTTPS uses port 443. Inside IIS, it is very easy to change the port that a website is listening on, if you have a need to do so. Probably the most common reason to institute a port change on a website is to keep it hidden.

Maybe you have an administrative site of some kind and want to make sure that nobody stumbles across it. Or perhaps your web server is limited on IP addresses, and you need to turn on another webpage but all of your IPs are already running sites. You could utilize a different port for the new site and then have the opportunity to run two (or more) sites using the same IP address, one site on each port.

Whatever your reason is for wanting to change the port that a website runs on, let's walk through the steps to accomplish this task so that it can be one more tool added to your belt.

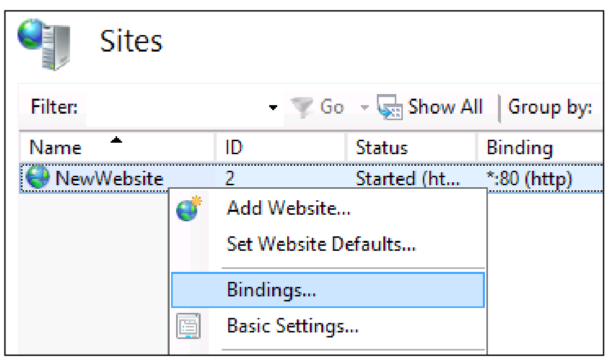
## Getting ready

We have a Windows Server 2012 R2 server online that has the IIS role installed. There is already a website running on this server. Currently it is using default port 80, but we want to change that port to 81 and test accessing it from a client computer.

## How to do it...

Here are the steps needed to change your website listener port:

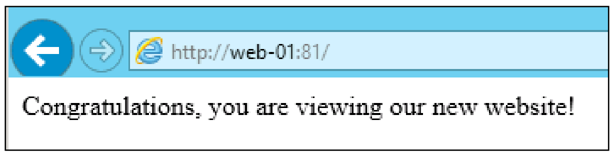
1. Open **Internet Information Services (IIS) Manager** from inside the **Tools** menu of Server Manager.
2. In the left window pane, expand the name of your web server and click on the **Sites** folder.
3. Right-click on your website, and choose **Bindings...**.




4. Choose the **http** binding that currently displays port **80** and click on the **Edit...** button.
5. Change the **Port** field to `81`. This is just for our example, of course. You could enter any valid port number in this field that wasn't otherwise in use on this server.
6. Click **OK**, then **Close**.

7. The port is immediately changed on your website. It is no longer listing on port 80. Let's test this by moving to a client computer on our network and opening Internet Explorer.
8. Try browsing to the old website address, `http://web-01`.


9. Whoops! I guess that isn't going to work any longer. Instead, we need to include our specific port in the URL from now on. Let's try `http://web-01:81`.




## How it works...

We can easily adjust the port that is used to access a website inside IIS by making one simple adjustment. After changing the port number in our website's bindings, the site immediately changes over to listening on the new port and is no longer active on the old port. Instead of changing the port, you could also add an additional binding into that same screen in order to get the website to respond from multiple ports at the same time. For example, if you wanted your website to run both HTTP for regular access and HTTPS for encrypted access to pages with sensitive information, you could create bindings for both port 80 and port 443.

One note of importance when changing your website port. Doing so means your web links for accessing the website will now have to include that specific port number at the end. Also, if you are running firewalls in your network or on the web server itself, it is possible that you will need to adjust settings on those firewalls to allow the new port to be allowed safe passage.

# Adding encryption (HTTPS) to your website

Using websites to pass data around the Internet is a staple of technology as we know it today. Installing even the most simple new tool or system will likely require you to download software or an update, or to register your information with a website. As an IT professional, I hope that you are familiar with HTTP versus HTTPS websites and the importance of distinguishing between the two. But now that we have a website running, how can we enable HTTPS on it so that we can protect this data that is traversing back and forth between our web server and the client computers?

It is typically the web developer's job to tell a website when to call for HTTPS, so you shouldn't have to worry too much about the actual content of the website. As the server administrator, however, you need to make sure that once HTTPS is called for on the website, your web server is capable of processing that traffic appropriately.

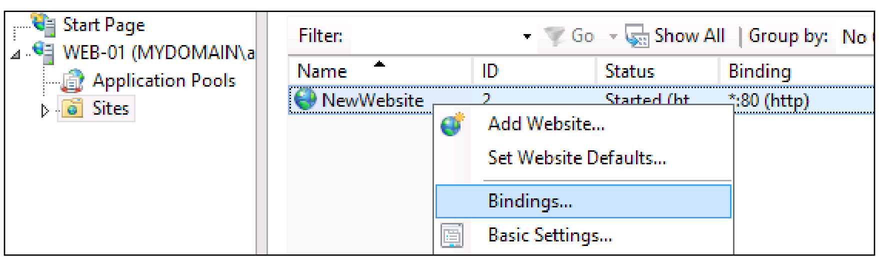
## Getting ready

We are running a Server 2012 R2 web server from which we will accomplish this task. There is a simple website currently running inside IIS on this server. Part of our recipe will be choosing an SSL certificate that we want to run on our website, so this recipe assumes that the certificate is already installed on your server. If you need assistance with the acquisition of the certificate itself, please reference the *Using a Certificate Signing Request (CSR) to acquire your SSL certificate* recipe.

## How to do it...

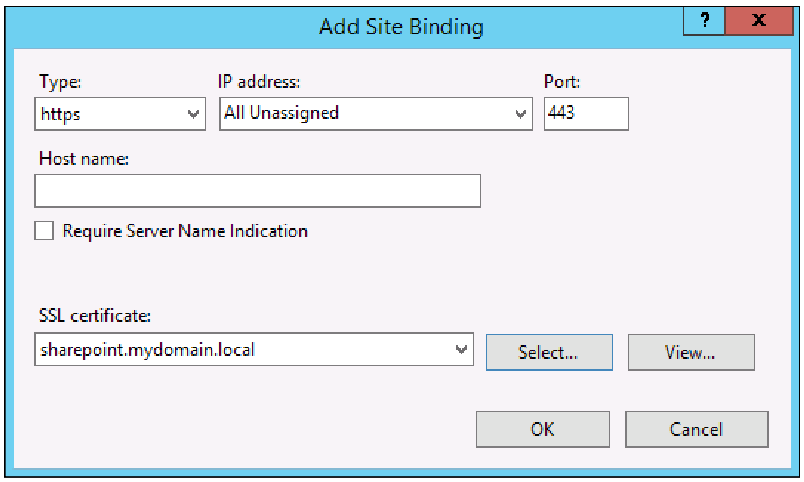
To configure your website for HTTPS traffic:

1. Launch **Internet Information Services (IIS) Manager** from the **Tools** menu inside Server Manager.
2. In the left window pane, expand your web server name and click on the **Sites** folder.
3. Right-click on your website and choose **Bindings...**.




Since it is a new website, you can see that there is only one binding listed currently. This binding is for port 80, which makes it an HTTP-only website. If you currently tried to access this site via HTTPS, it would fail. The port for HTTPS is 443, and so we need to add a new binding that uses port 443. A mistake that I have watched new admins make is to edit this existing binding and change it from 80 to 443. This will cause the website to only listen on port 443, or rather to only accept requests via HTTPS. This may be desirable in some instances, but not most. You most often want the website to respond to both HTTP and HTTPS requests.

4. Go ahead and click the **Add...** button.
5. Change the **Type** field to **https**. You will notice that the **Port** field changes to `443` automatically.
6. If you only want this new binding to work on a particular IP address, choose it now. Otherwise leave it set to **All Unassigned** to cause this new listener to be active on all IP addresses that exist on our server.
7. Select the **SSL certificate** that you want IIS to use for authenticating requests to this website. HTTPS traffic is only encrypted and guaranteed to be safe from prying eyes because the tunnel is being validated by an SSL certificate that is specific to your website name. You must have an SSL certificate installed onto the server so that you can choose it from the list here in order to create an HTTPS binding.




8. Click **OK**, then click **Close**. Your HTTPS binding is now active on this website.

## How it works...

In this recipe, we used the IIS management console to add a second binding to our new website. This new binding is for accepting HTTPS traffic. We intend to run parts of this website as HTTP, and some more sensitive pages as HTTPS. Therefore, we created a second binding, enabling both HTTP and HTTPS traffic to flow successfully to and from this site. During the course of this recipe, we needed to choose the SSL certificate that the website is going to use in order to validate the HTTPS traffic that is coming in. There was already an SSL certificate installed onto the server for our website; we simply had to choose it from the list.

# Using a Certificate Signing Request (CSR) to acquire your SSL certificate

When publishing a website out to the Internet, it is generally a best practice to use an SSL certificate on the website that you acquired from a public Certification Authority (CA). These are the big certificate issuing entities such as Entrust, Verisign, GoDaddy, and so on. It is possible to use your own internal PKI infrastructure to issue SSL certificates that can be exposed to the outside world, but it can be difficult to set up the certificate infrastructure appropriately and securely. As cheap as SSL certificates are, it is worth the investment to have the security of knowing that the certificate you are running on your website is the one and only certificate of its kind. That nobody else has a chance to get their hands on a copy of the same certificate and spoof your website. Modern browsers also have a pre-built list of the public CAs that they trust; this makes using a certificate from one of those public entities even more beneficial because your user's browsers will automatically trust those certificates without any additional work on the client side.

It is easy enough to log in to one of these CA websites and purchase a new certificate, but then comes the tricky part. Once purchased, you need to walk through some steps and enter information about your certificate. Easy enough; it asks you for some company information and the name that you plan to use for your site, of course. Then it asks for your **Certificate Signing Request** (**CSR**) and gives you either a very large empty text box to paste it into or an upload function where you can upload your CSR directly to them. This is the place where I have watched many new admins struggle to find traction on their next step.

A CSR is a file that must be created on your web server. It contains information that the CA uses when it creates your certificate. When they do this, it binds the certificate to the information in the CSR, ensuring that your certificate is built specifically for your web server. Today we are going to generate a CSR together, so that you are prepared to handle that screen when you come across it.

## Getting ready

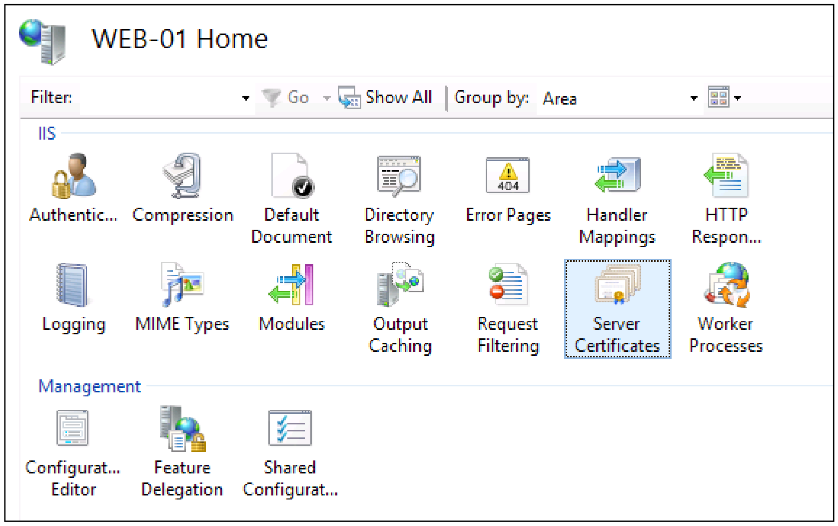
We are going to use IIS that is running on our Server 2012 R2 web server to generate a CSR. This server is the only piece of infrastructure that we need running for this task.

## How to do it…

In order to request a new certificate from a public CA, you will need to spin out a CSR on your web server. Here are the steps to do so:

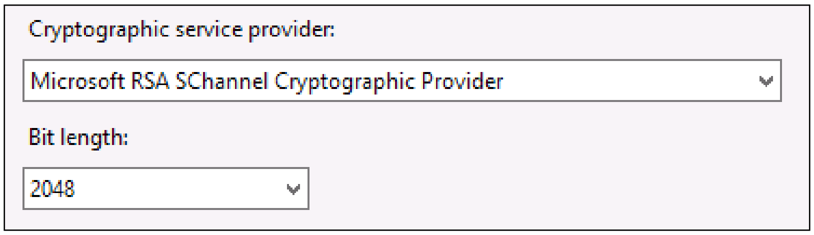
1. Open **Internet Information Services (IIS) Manager**.
2. Click on the name of your server in the left window pane.

3. Double-click on the **Server Certificates** applet. This will display currently installed certificates on your server.

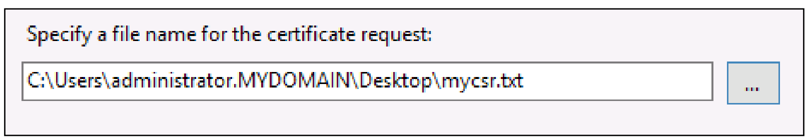



4. Click on the action near the right of your screen that says **Create Certificate Request...**.
5. Populate **Common name** with the DNS name that your website will be running on. This is the name that users will type into their browsers in order to access this site.
6. **Organization** is the name of your company or organization. Typically this information needs to match whatever is on file with the CA, so take a minute to check another certificate that you might have already and make sure to type in the same info.
7. **Organizational unit** can be anything you desire. I often just type the word `Web`.
8. Type in your **City/locality** and **State/province** to finish out this screen. Make sure to spell out the whole word of your state—for example, California. They tend to dislike abbreviations.

9. Click **Next**.
10. Increase your **Bit length** to at least **2048**. This is typically considered to be the new minimum bit length standard in the industry.




11. Click **Next**.
12. Type a location and name where you want to store your new CSR. Usually you set this into a `.txt` file. Make sure to specify the full filename including the extension. I have found that, if you do not, the file disappears into neverland.

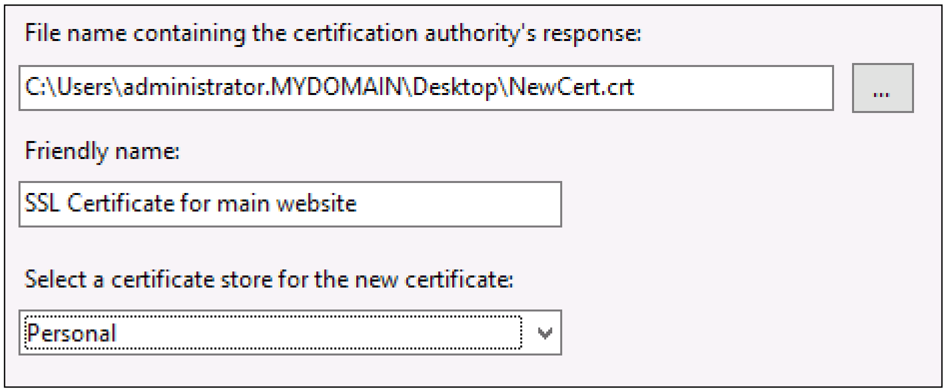



13. Click **Finish** and go take a look at that new file. It will look like a big mess of letters and numbers, which is normal!
14. Now you can proceed to your public CA's web interface and use this new CSR during the official request for a new SSL certificate. When prompted, paste the contents of the CSR file into the CA's web interface. This is the last time that you will need that CSR.

> Each authority handles this process differently, but they are all generally done through a website with a series of steps that you walk through. Many CAs will allow you to generate a 15 or 30 day trial certificate, so that you can test this without cost.

15. After the CA validates your request and your CSR, they will issue you a link where you can download your new certificate. Go ahead and download that file and copy it onto your web server.
16. Once the certificate file is on your server, you need to import it into IIS. Head back into the **Server Certificates** section and this time click on **Complete Certificate Request...**.

17. Specify the newly downloaded certificate file and fill in the **Friendly name** field if you choose. This is a descriptive name that you can give to this new certificate inside IIS, so that you can easily identify it later when assigning it to a website binding. You typically want to store these certificates in the **Personal** store, as is set by default.




18. Click **OK** and that is it! Your new certificate is installed and ready to use.

## How it works...

In this recipe, we requested a new SSL certificate from our favorite public certification authority. In order to receive a certificate from them, we had to issue a Certificate Signing Request (CSR) from our web server. Once we had our CSR generated, we simply copy and paste it into the web interface for our CA entity and they give us a new certificate based on that CSR. Once downloaded, the new certificate file can be imported back into the web server, where it is ready for use by our own website.

> One note of importance. After you install the new certificate onto your server, double-click on the certificate to open it up. You want to make sure that you have a message displaying on the main page of your certificate properties that says **You have a private key that corresponds to this certificate**. This will display near the bottom of the **General** tab of the certificate. If you do not see this message, something did not work correctly with the CSR and you will probably have to start the process over to request another new copy of the certificate. Having a private key that corresponds to your SSL certificate is critical to getting your website working properly.

# Moving an SSL certificate from one server to another

There are multiple reasons why you may need to move or copy an SSL certificate from one web server to another. If you have purchased a wildcard certificate for your network, you are probably going to use that same certificate on a lot of different servers as it can be used to validate multiple websites and DNS names. Even if you are using singularly named certificates, you may be turning on multiple web servers to host the same site, to be set up in some sort of load balanced fashion. In this case, you will also need the same SSL certificate on each of the web servers, as they could all potentially be accepting traffic from clients.

When moving or copying a certificate from one server to another, there is definitely a right way and a wrong way to go about it. Let's spend a little bit of time copying a cert from one server to another so that you can become familiar with this task.

## Getting ready

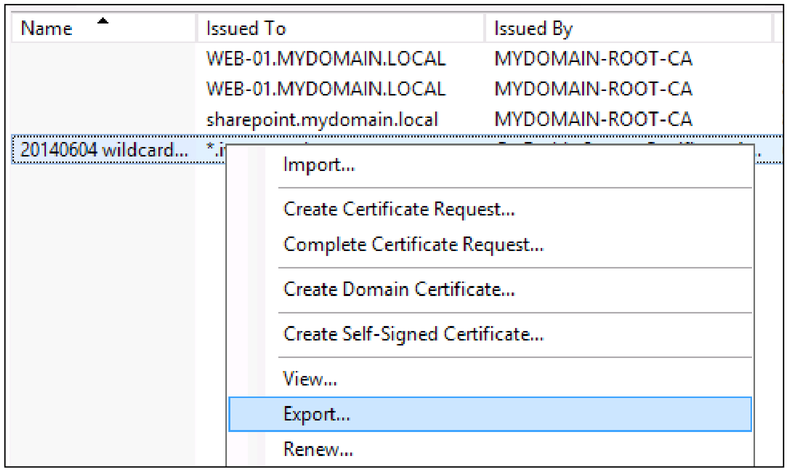
We have two Server 2012 R2 boxes online in our environment. These are both destined to be web servers hosting the same website. IIS has been installed on both. The SSL certificate that we require has been installed onto the primary server. We now need to export the certificate from there and import it successfully onto our second server.

## How to do it...

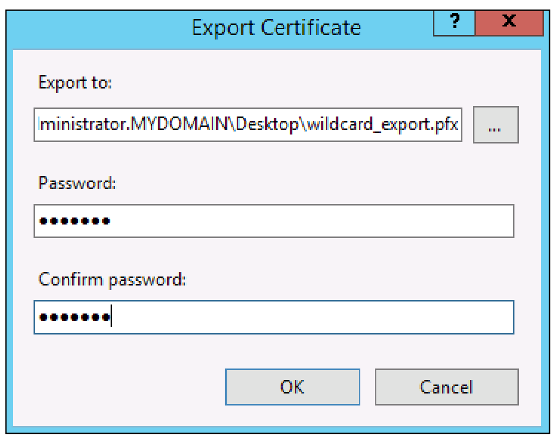
Follow these steps to copy a certificate from one server to another:

1. On your primary web server, launch **Internet Information Services (IIS) Manager** from the **Tools** menu of Server Manager.
2. Click on the name of your server in the left window pane.
3. Double-click on the **Server Certificates** applet to view the certificates currently installed on this system.

4. For our example, I am using a wildcard certificate that has been installed onto this server. Right-click on the certificate and choose **Export...**.

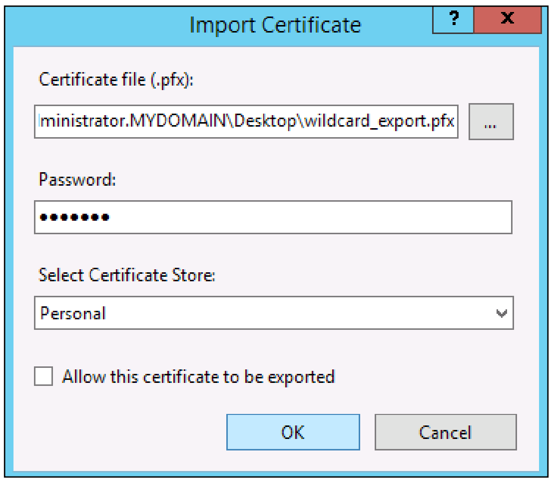



5. Choose a location to store this exported file and enter a password that will be used to protect the file.

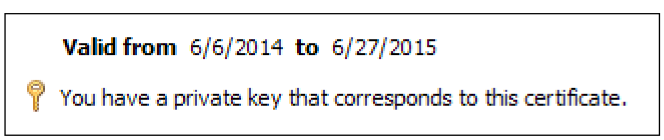



6. Clicking **OK** will create a PFX file and place it onto your Desktop (or wherever you told it to save). Now copy this PFX file over to your secondary web server.
7. Open up the IIS Management console on the second server and navigate to the same **Server Certificates** location.
8. Right-click in the center pane and choose **Import...**. Alternatively, you could choose the **Import...** action from the right window pane.

9. Browse to the location of your certificate and input the password that you used to protect the PFX file.
10. Before clicking **OK**, decide whether or not you want this certificate to be exportable from this secondary server. Sometimes this is desirable if you plan to have to export the certificate again in the future. If you do not have a reason to do that, go ahead and uncheck this box. Unchecking **Allow this certificate to be exported** helps to limit the places where you have certificates floating around the network. The more you have out there that are potentially exportable, the more chance you have of one getting out of your hands.




11. Once you click **OK**, your certificate should now be installed and visible inside the IIS window.
12. Double-click on the certificate and check over the properties to make sure everything looks correct. Make sure that you see the message across the bottom that says **You have a private key that corresponds to this certificate**. If that message is missing, something didn't work properly during your export and the private key was somehow not included in the certificate export that you did. You will have to revisit the primary server and export again to make sure that the certificate on the secondary server does contain private key information, or it will not work properly.



## How it works...

We used the IIS management console to export and import an SSL certificate, which is a pretty straightforward and simple task to do once you understand the process. The critical part is making sure that your export includes the private key information. If it does not, the certificate will not be able to validate traffic properly. Using IIS to accomplish this task is the best way to move certificates. You could also make use of the MMC snap-in for certificates, but it is a little more complicated. If you try to use that console, you will be asked whether or not you want to export the private key. The default option is set to **No, do not export the private key**. It is a common mistake to leave that setting in place and wonder later why the certificate doesn't work properly on other servers where you have installed it. You must make sure to select the option **Yes, export the private key**.

# Rebinding your renewed certificates automatically

Certificates expire; this is just a simple fact of life. Most often I find that companies purchase SSL certificates on a short-term basis, usually for only one year. This means that every year each certificate needs to be renewed. However, downloading a new copy of the certificate and installing it onto your web server is not enough to make it continue working. Simply putting the new certificate into place on the server does not mean that IIS is going to start using the new one to validate traffic on your website. Even if you delete the old certificate, there is no action that has been taken inside IIS to tell it that this new certificate that suddenly appeared is the one that it should start using as the binding for your site. So we have always had to make this additional change manually. Every time you replace a certificate, you also go into IIS and change the binding on the website. This seems particularly painful when you have the certificate renewal automated through something like Autoenrollment. You may mistakenly think that you are covered in the future and no longer have to do anything to renew your certs, because they will be renewed at the server level automatically. But alas this is not true; up until now we have still always had to go into IIS and change the binding by hand. Fear not, the future is here...

The IIS team has made a simple but powerful change to help this problem in the new version of IIS that ships with Windows Server 2012 R2. A new feature called **Certificate Rebind**, when enabled, causes IIS to automatically recognize a new certificate installation, and to automatically rebind the appropriate website to use the new copy of the certificate instead of the expiring one. Let's take a look at the interface so that you know how to turn this option on and off. We will also take a little look under the hood so that you can understand how this functionality works.

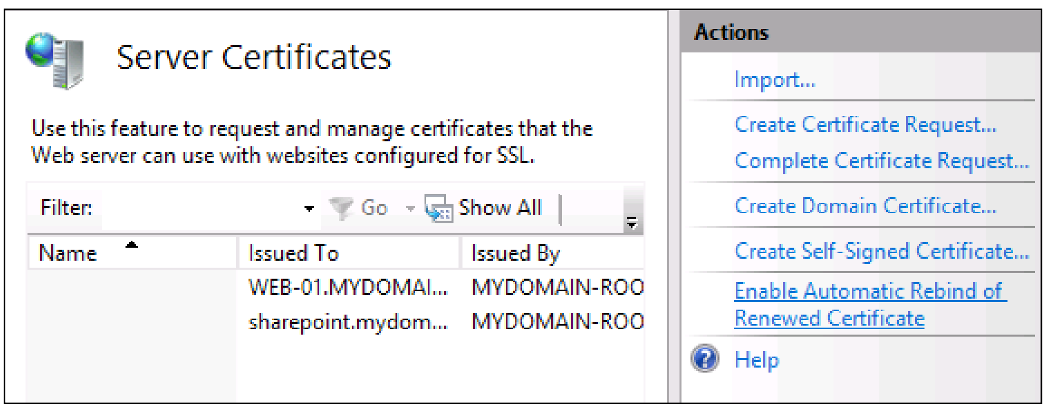
## Getting ready

This work will be accomplished on our Windows Server 2012 R2 web server. We have IIS installed and have an HTTPS website running with an SSL certificate already bound to the site.

## How to do it...

Follow these steps to enable Certificate Rebind on your IIS web server:

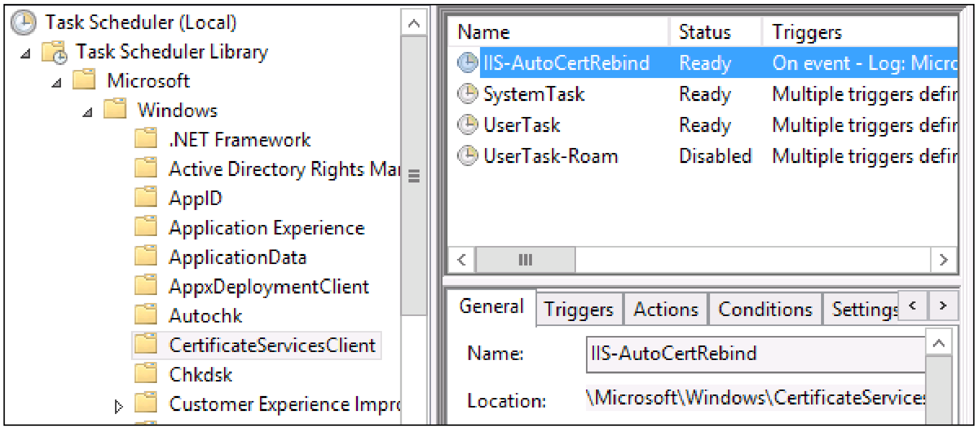
1. Open **Internet Information Services (IIS) Manager** from inside the **Tools** menu of Server Manager.
2. From the left window pane, click on the name of your web server.
3. Double-click on the **Server Certificates** applet.
4. In the right window pane, click on the action that is called **Enable Automatic Rebind of Renewed Certificate**.




That's it! IIS has now been configured so that it will recognize the installation of a renewed certificate, and will rebind your website automatically to make use of the new certificate. Now let's take a brief look at how this process actually works.

5. Use either a command prompt or the **Start** screen to launch `Taskschd.msc`. This is the **Windows Task Scheduler**.

6. In the left pane, navigate to **Task Scheduler Library** | **Microsoft** | **Windows** | **CertificateServicesClient**.




You can see a scheduled task listed here that is called **IIS-AutoCertRebind**. This is the magic of Certificate Rebind. When a certificate gets added or renewed on your Server 2012 R2 system, an event is logged. When this event is logged, this scheduled task picks it up, and uses the information that it has from IIS about the certificates to rebind the websites onto the new certificates.

7. If you head back into IIS and click on the **Action** for **Disable Automatic Rebind of Renewed Certificate**, you will notice that our scheduled task disappears from the list.

## How it works…

Certificate Rebind is a really simple action to enable inside IIS, but can make all the difference in a good or bad day at the office. When enabled, this feature builds a scheduled task inside Windows that triggers the commands to bind our IIS website to its new certificate. This task is triggered by an event that is logged in Windows when our new certificate is installed or renewed. With Certificate Rebind enabled and the configuration of your certificate distribution set to happen automatically through Autoenrollment, you can now have a truly automated certificate renewal system inside your network!

# Using host headers to manage multiple websites on a single IP address

When setting up a website inside IIS, the most straightforward approach is certainly to utilize an individual IP address for each website that you will be running. It is common to run more than one site on a single web server, and so this sometimes means that your web servers have numerous IP addresses configured on them. However, sometimes this is not possible. For example, you may be working on a web server that is internet-facing and there is a restriction on the amount of available public IP addresses that can be used. In this case, you may run across the need to host multiple websites on a single IP address.

This is where host headers come into play. Host headers can be configured on your websites so that the site responds to a particular request coming in from the client. These header requests can help the web server distinguish between traffic, directing users calling for websites to their appropriate site inside IIS. Let's work together to set up two websites inside IIS and force them to utilize the same IP address and port. We want everything to remain standard as far as the port goes, so we want them to both be able to utilize port 80, but we only have one IP address available to install onto our web server.

## Getting ready

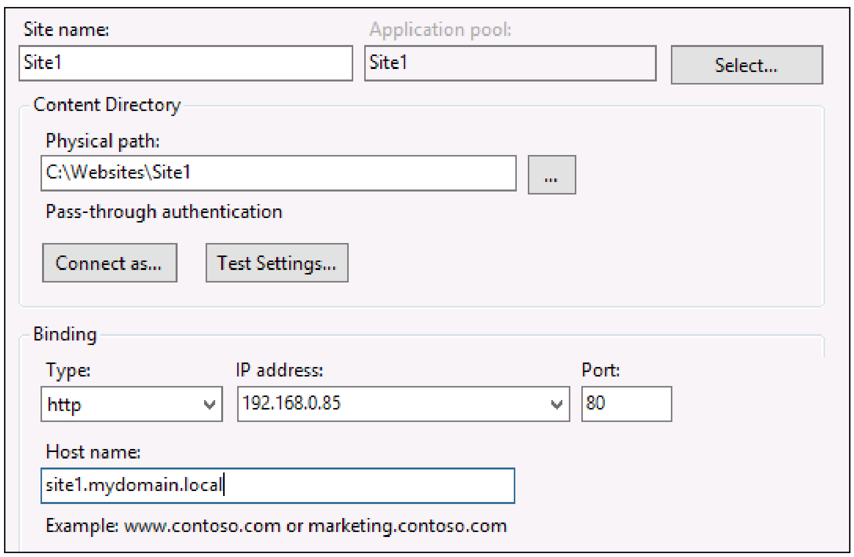
The work will be accomplished from inside IIS on our Server 2012 R2 web server. We will also utilize a client computer to test connectivity to the websites once we are finished setting them up.

## How to do it…

To create two websites that share the same IP address and split traffic by using host headers:

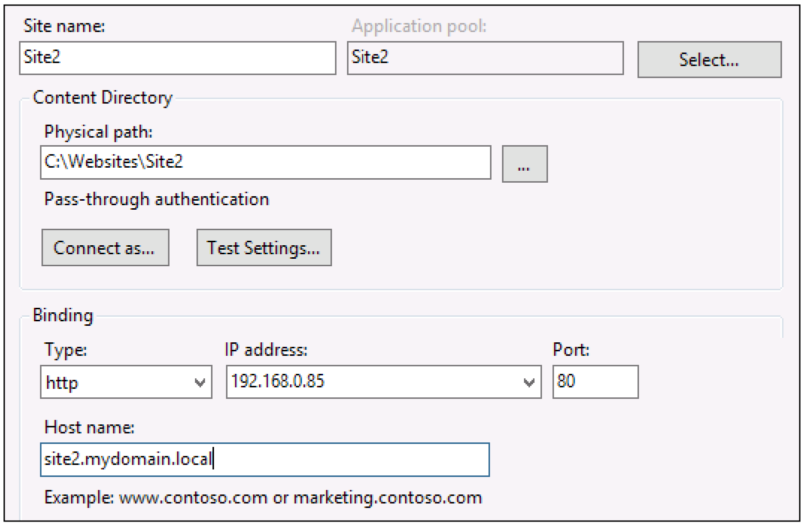
1. On your web server, open up File Explorer and create a new folder called `C:\Websites`. Inside this folder, create two new folders and call them `Site1` and `Site2`.
2. Inside each folder, create a new `Default.htm` file. You should now have two different `Default.htm` files, one sitting inside the `Site1` folder, and one sitting inside the `Site2` folder. These will be our example websites.
3. Throw some text inside each of those `Default.htm` files. Make sure that whatever text you write in them distinguishes between the websites so that we can know it is working properly when we test in a few minutes.

4. Open **Internet Information Services (IIS) Manager** from the **Tools** menu of Server Manager.
5. In the left window pane, expand the name of your web server. Then right-click on the **Sites** folder, and choose **Add Website...**.
6. We are going to name our site `Site1` and choose our `C:\Websites\Site1` folder as the location for this website. I am also going to select the **IP address** drop-down field and specify the one and only IP address on this system, so that we can prove host headers are working as they should. Remember, our intention is to get two websites running on this same IP address and port combination.
7. Here's the part that may be new territory for you: the **Host name** field. This is the DNS name that requests for this website will be coming in with. So whatever DNS name your users are going to type into their browser is the name that you need to enter here. We are going to use `site1.mydomain.local`.




8. Click **OK** and you have the first website up and running on the web server.

9. Now walk through the same process as above, but this time specify all of the information for `Site2`. We are going to choose the same **IP address** and **Port**, but we are going to specify a different name in the **Host name**:




We now have two websites, both running on the same IP address and port on the same web server. Let's test to find out whether or not IIS is smart enough to distinguish between the sites when we try to browse these websites from our client computer.

> Remember to create DNS records for these websites! You will need host records created for `site1.mydomain.local` and `site2.mydomain.local`, and they both need to be pointed at the IP address of the web server, which in our case is `192.168.0.85`.

10. On a client computer, browse to `http://site1.mydomain.local`. You should see the text from the `Default.htm` file that we put into the `Site1` folder on the web server:

11. Now browse to `http://site2.mydomain.local`. We can see that the web server recognizes our request for the second site; even though they are running on the same IP address, our request is sent over to the second website:


## How it works...

When we set up websites inside IIS to utilize different host headers, it gives us the ability to publish multiple sites on the same IP address and port numbers. This can be very useful in cases where IP addresses are limited or where you don't want to configure multiple addresses onto the web server for any reason. IIS is capable of listening on the same IP and port for web requests coming in to different host names, and forwarding those requests on to the appropriate website based on the host header name that was requested by the client computer. It is important to note that requests for these web pages must come by the name for this to work properly; you cannot type the IP address of the website into the browser and expect it to work since we are now sharing that IP address between two or more different sites.

# 6
# Remote Access

With Windows Server 2012 R2, Microsoft brings a whole new way of looking at remote access. Companies have historically relied on third-party tools to connect remote users into the network, such as traditional and SSL VPN provided by appliances from large networking vendors. I'm here to tell you those days are gone. Those of us running Microsoft-centric shops can now rely on Microsoft technologies to connect our remote workforce. Better yet is that these technologies are included with the Server 2012 R2 operating system, and have functionality that is much improved over anything that a traditional VPN can provide.

Regular VPN does still have a place in the remote access space, and the great news is that you can also provide it with Server 2012 R2. We have some recipes on setting up VPN, but our primary focus for this chapter will be **DirectAccess** (**DA**). DA is kind of like automatic VPN. There is nothing the user needs to do in order to be connected to work. Whenever they are on the Internet, they are also connected automatically to the corporate network. DirectAccess is an amazing way to have your Windows 7 and Windows 8 domain joined systems connected back to the network for data access and for management of those traveling machines. DirectAccess has actually been around since 2008, but the first version came with some steep infrastructure requirements and was not widely used. Server 2012 R2 brings a whole new set of advantages and makes implementation much easier than in the past. I still find many server and networking admins who have never heard of DirectAccess, so let's spend some time together exploring some of the common tasks associated with it.

In this chapter, we will cover the following recipes:

- Configuring DirectAccess, VPN, or a combination of the two
- Pre-staging Group Policy Objects (GPOs) to be used by DirectAccess
- Enhancing the security of DirectAccess by requiring certificate authentication
- Building your Network Location Server (NLS) on its own system
- Enabling Network Load Balancing (NLB) on your DirectAccess servers
- Adding VPN to your existing DirectAccess server

- Replacing your expiring IP-HTTPS certificate
- Reporting on DirectAccess and VPN connections

# Introduction

There are two "flavors" of remote access available in Windows Server 2012 R2. The most common way to implement the Remote Access role is to provide DirectAccess for your Windows 7 and Windows 8 domain joined client computers, and VPN for the rest. The DirectAccess machines are typically your company-owned corporate assets. One of the primary reasons that DirectAccess is usually only for company assets is that the client machines must be joined to your domain, because the DirectAccess configuration settings are brought down to the client through a GPO. I doubt you want home and personal computers joining your domain.

VPN is therefore used for down level clients such as Windows XP, and for home and personal devices that want to access the network. Since this is a traditional VPN listener with all regular protocols available such as PPTP, L2TP, SSTP, it can even work to connect devices such as smartphones.

There is a third function available within the Server 2012 R2 Remote Access role, called the **Web Application Proxy** (**WAP**). This function is not used for connecting remote computers fully into the network as DirectAccess and VPN are; rather, WAP is used for publishing internal web resources out to the internet. For example, if you are running Exchange and Lync Server inside your network and want to publish access to these web-based resources to the internet for external users to connect to, WAP would be a mechanism that could publish access to these resources. The term for publishing out to the internet like this is Reverse Proxy, and WAP can act as such. It can also behave as an ADFS Proxy.

For further information on the WAP role, please visit: `http://technet.microsoft.com/en-us/library/dn584107.aspx`

One of the most confusing parts about setting up DirectAccess is that there are many different ways to do it. Some are good ideas, while others are not. Before we get rolling with recipes, we are going to cover a series of questions and answers to help guide you toward a successful DA deployment. The first question that always presents itself when setting up DA is "How do I assign IP addresses to my DirectAccess server?". This is quite a loaded question, because the answer depends on how you plan to implement DA, which features you plan to utilize, and even upon how secure you believe your DirectAccess server to be. Let me ask you some questions, pose potential answers to those questions, and discuss the effects of making each decision.

# DirectAccess Planning Q&A

- Which client operating systems can connect using DirectAccess?

  Answer: Windows 7 Ultimate, Windows 7 Enterprise, and Windows 8.x Enterprise. You'll notice that the Professional SKU is missing from this list. That is correct, Windows 7 and Windows 8 Pro do not contain the DirectAccess connectivity components. Yes, this does mean that Surface Pro tablets cannot utilize DirectAccess out of the box. However, I have seen many companies now install Windows 8 Enterprise onto their Surface tablets, effectively turning them into "Surface Enterprises." This works fine and does indeed enable them to be DirectAccess clients. In fact, I am currently typing this text on a DirectAccess-connected Surface "Pro turned Enterprise" tablet.

- Do I need one or two NICs on my DirectAccess server?

  Answer: Technically, you could set it up either way. In practice however, it really is designed for dual-NIC implementation. Single NIC DirectAccess works okay sometimes to establish a proof-of-concept to test out the technology. But I have seen too many problems with single NIC implementation in the field to ever recommend it for production use. Stick with two network cards, one facing the internal network and one facing the Internet.

- Do my DirectAccess servers have to be joined to the domain?

  Answer: Yes.

- Does DirectAccess have site-to-site failover capabilities?

  Answer: Yes, though only Windows 8.x client computers can take advantage of it. This functionality is called Multi-Site DirectAccess. Multiple DA servers that are spread out geographically can be joined together in a multi-site array. Windows 8 client computers keep track of each individual entry point and are able to swing between them as needed or at user preference. Windows 7 clients do not have this capability and will always connect through their primary site.

- What are these things called 6to4, Teredo, and IP-HTTPS I have seen in the Microsoft documentation?

  Answer: 6to4, Teredo, and IP-HTTPS are all IPv6 transition tunneling protocols. All DirectAccess packets that are moving across the internet between DA client and DA server are IPv6 packets. If your internal network is IPv4, then when those packets reach the DirectAccess server they get turned down into IPv4 packets, by some special components called DNS64 and NAT64. While these functions handle the translation of packets from IPv6 into IPv4 when necessary inside the corporate network, the key point here is that all DirectAccess packets that are traveling over the Internet part of the connection are always IPv6. Since the majority of the Internet is still IPv4, this means that we must tunnel those IPv6 packets inside something to get them across the Internet.

That is the job of 6to4, Teredo, and IP-HTTPS. 6to4 encapsulates IPv6 packets into IPv4 headers and shuttles them around the internet using protocol 41. Teredo similarly encapsulates IPv6 packets inside IPv4 headers, but then uses UDP port 3544 to transport them. IP-HTTPS encapsulates IPv6 inside IPv4 and then inside HTTP encrypted with TLS, essentially creating an HTTPS stream across the Internet. This, like any HTTPS traffic, utilizes TCP port 443. The DirectAccess traffic traveling inside either kind of tunnel is always encrypted, since DirectAccess itself is protected by IPsec.

- Do I want to enable my clients to connect using Teredo?

  Answer: Most of the time, the answer here is yes. Probably the biggest factor that weighs on this decision is whether or not you are still running Windows 7 clients. When Teredo is enabled in an environment, this gives the client computers an opportunity to connect using Teredo, rather than all clients connecting in over the IP-HTTPS protocol. IP-HTTPS is sort of the "catchall" for connections, but Teredo will be preferred by clients if it is available. For Windows 7 clients, Teredo is quite a bit faster than IP-HTTPS. So enabling Teredo on the server side means your Windows 7 clients (the ones connecting via Teredo) will have quicker response times, and the load on your DirectAccess server will be lessened. This is because Windows 7 clients who are connecting over IP-HTTPS are encrypting all of the traffic twice. This also means that the DA server is encrypting/decrypting everything that comes and goes twice. In Windows 8, there is an enhancement that brings IP-HTTPS performance almost on par with Teredo, and so environments that are fully cut over to Windows 8 will receive less benefit from the extra work that goes into making sure Teredo works.

- Can I place my DirectAccess server behind a NAT?

  Answer: Yes, though there is a downside. Teredo cannot work if the DirectAccess server is sitting behind a NAT. For Teredo to be available, the DA server must have an External NIC that has two consecutive *public* IP addresses. True public addresses. If you place your DA server behind any kind of NAT, Teredo will not be available and all clients will connect using the IP-HTTPS protocol. Again, if you are using Windows 7 clients, this will decrease their speed and increase the load on your DirectAccess server.

- How many IP addresses do I need on a standalone DirectAccess server?

  Answer: I am going to leave single NIC implementation out of this answer since I don't recommend it anyway. For scenarios where you are sitting the External NIC behind a NAT or, for any other reason, are limiting your DA to IP-HTTPS only, then we need one external address and one internal address. The external address can be a true public address or a private NATed DMZ address. Same with the internal; it could be a true internal IP or a DMZ IP. Make sure both NICs are not plugged into the same DMZ, however. For a better installation scenario that allows Teredo connections to be possible, you would need two consecutive public IP addresses on the External NIC and a single internal IP on the Internal NIC. This internal IP could be either true internal or DMZ. But the public IPs would really have to be public for Teredo to work.

- Do I need an internal PKI?

  Answer: Maybe. If you want to connect Windows 7 clients, then the answer is yes. If you are completely Windows 8, then technically you do not need internal PKI. But you really should use it anyway. Using an internal PKI, which can be a single, simple Windows CA server, increases the security of your DirectAccess infrastructure. You'll find out during this chapter just how easy it is to require certificates as part of the tunnel building authentication process.

# Configuring DirectAccess, VPN, or a combination of the two

Now that we have some general ideas about how we want to implement our remote access technologies, where do we begin? Most services that you want to run on a Windows Server begin with a role installation, but the implementation of remote access begins before that. Let's walk through the process of taking a new server and turning it into a Microsoft Remote Access server.

## Getting ready

All of our work will be accomplished on a new Windows Server 2012 R2. We are taking the two-NIC approach to networking, and so we have two NICs installed on this server. The Internal NIC is plugged into the corporate network and the External NIC is plugged into the Internet for the sake of simplicity. The External NIC could just as well be plugged into a DMZ.

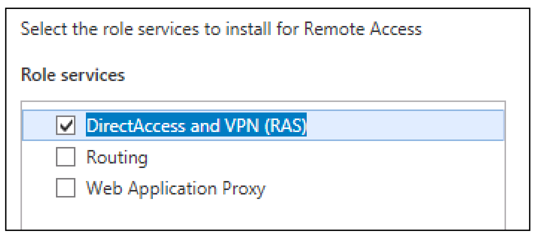
## How to do it...

Follow these steps to turn your new server into a Remote Access server:

1. Assign IP addresses to your server. Since this is a multi-homed system with both internal and external networks connected, make sure to follow the steps in the *Multi-homing your Windows Server 2012 R2* recipe in *Chapter 3*, *Security and Networking*. Remember, the most important part is making sure that the Default Gateway goes on the External NIC only.
2. Join the new server to your domain.
3. Install an SSL certificate onto your DirectAccess server that you plan to use for the IP-HTTPS listener. This is typically a certificate purchased from a public CA.
4. If you're planning to use client certificates for authentication, make sure to pull down a copy of the certificate to your DirectAccess server.

> You want to make sure certificates are in place before you start with the configuration of DirectAccess. This way the wizards will be able to automatically pull in information about those certificates in the first run. If you don't, DA will set itself up to use self-signed certificates, which are a security no-no.

5. Use Server Manager to install the Remote Access role. You should only do this after completing the steps listed earlier.
6. If you plan to load balance multiple DirectAccess servers together at a later time, make sure to also install the feature called **Network Load Balancing**.
7. After selecting your role and feature, you will be asked which Remote Access role services you want to install. For our purposes in getting the remote workforce connected back into the corporate network, we want to choose **DirectAccess and VPN (RAS)**.




8. Now that the role has been successfully installed, you will see a yellow exclamation mark notification near the top of Server Manager indicating that you have some **Post-deployment Configuration** that needs to be done.

> Do *not* click on **Open the Getting Started Wizard**!

Unfortunately, Server Manager leads you to believe that launching the **Getting Started Wizard (GSW)** is the logical next step. However, using the GSW as the mechanism for configuring your DirectAccess settings is kind of like roasting a marshmallow with a pair of tweezers. In order to ensure you have the full range of options available to you as you configure your remote access settings, and that you don't get burned later, make sure to launch the configuration this way:

1. Click on the **Tools** menu from inside Server Manager and launch the **Remote Access Management Console**.
2. In the left window pane, click on **Configuration** | **DirectAccess and VPN**.

3. Click on the second link, the one that says **Run the Remote Access Setup Wizard**. Please note that once again the top option is to run that pesky Getting Started Wizard. Don't do it! I'll explain why in the *How it works...* section of this recipe.


9. Now you have a choice that you will have to answer for yourself. Are you configuring only DirectAccess, only VPN, or a combination of the two? Simply click on the option that you want to deploy. Following your choice, you will see a series of steps (steps 1 through 4) that need to be accomplished. This series of mini-wizards will guide you through the remainder of the DirectAccess and VPN particulars. This recipe isn't large enough to cover every specific option included in those wizards, but at least you now know the correct way to bring a DirectAccess/VPN server into operation.

## How it works...

The remote access technologies included in Server 2012 R2 have great functionality, but their initial configuration can be confusing. Following the procedure listed in this recipe will set you on the right path to be successful in your deployment, and prevent you from running into issues down the road. The reason that I absolutely recommend you stay away from using the "shortcut" deployment method provided by the Getting Started Wizard is twofold:

- GSW skips a lot of options as it sets up DirectAccess, so you don't really have any understanding of how it works after finishing. You may have DA up and running, but have no idea how it's authenticating or working under the hood. This holds so much potential for problems later, should anything suddenly stop working.
- GSW employs a number of bad security practices in order to save time and effort in the setup process. For example, using the GSW usually means that your DirectAccess server will be authenticating users without client certificates, which is not a best practice. Also, it will co-host something called the NLS website on itself, which is also not a best practice. Those who utilize the GSW to configure DirectAccess will find that their GPO, which contains the client connectivity settings, will be security-filtered to the Domain Computers group. Even though it also contains a WMI filter that is supposed to limit that policy application to mobile hardware such as laptops, this is a terribly scary thing to see inside GPO filtering settings. You probably don't want all of your laptops to immediately start getting DA connectivity settings, but that is exactly what the GSW does for you. Perhaps worst, the GSW will create and make use of self-signed SSL certificates to validate its web traffic, even the traffic coming in from the Internet! This is a terrible practice and is the number one reason that should convince you that clicking on the Getting Started Wizard is not in your best interests.

# Pre-staging Group Policy Objects (GPOs) to be used by DirectAccess

One of the great things about DirectAccess is that all of the connectivity settings the client computers need in order to connect are contained within a **Group Policy Object** (**GPO**). This means that you can turn new client computers into DirectAccess-connected clients without ever touching that system. Once configured properly, all you need to do is add the new computer account to an Active Directory security group, and during the next automatic Group Policy refresh cycle (usually within 90 minutes), that new laptop will be connecting via DirectAccess whenever outside the corporate network.

You can certainly choose not to pre-stage anything with the GPOs and DirectAccess will still work. When you get to the end of the DA configuration wizards, it will inform you that two new GPOs are about to be created inside Active Directory. One GPO is used to contain the DirectAccess server settings and the other GPO is used to contain the DirectAccess client settings. If you allow the wizard to handle the generation of these GPOs, it will create them, link them, filter them, and populate them with settings automatically. About half of the time I see folks do it this way and they are forever happy with letting the wizard manage those GPOs now and in the future.

The other half of the time, it is desired that we maintain a little more personal control over the GPOs. If you are setting up a new DA environment but your credentials don't have permission to create GPOs, the wizard is not going to be able to create them either. In this case, you will need to work with someone on your Active Directory team to get them created. Another reason to manage the GPOs manually is to have better control over placement of these policies. When you let the DirectAccess wizard create the GPOs, it will link them to the top level of your domain. It also sets Security Filtering on those GPOs so they are not going to be applied to everything in your domain, but when you open up the **Group Policy Management** Console you will always see those DirectAccess policies listed right up there at the top level of the domain. Sometimes this is simply not desirable. So for this reason also, you may want to choose to create and manage the GPOs by hand, so that we can secure placement and links where we specifically want them to be located.

## Getting ready

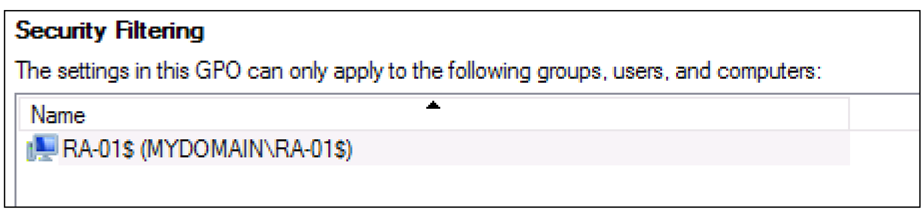
While the DirectAccess wizards themselves are run from the DirectAccess server, our work with this recipe is not. The Group Policy settings that we will be configuring are all accomplished within Active Directory, and we will be doing the work from a Domain Controller in our environment.

## How to do it...

To pre-stage Group Policy Objects (GPOs) for use with DirectAccess:

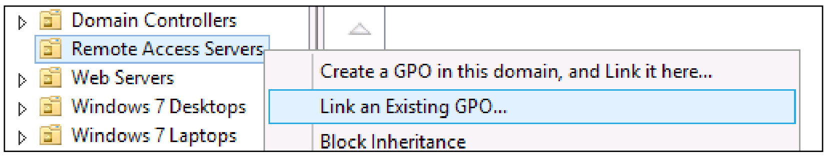
1. On your Domain Controller, launch the **Group Policy Management Console**.
2. Expand **Forest** | **Domains** | **Your Domain Name**. There should be a listing here called **Group Policy Object**. Right-click on that and choose **New**.
3. Name your new GPO something like `DirectAccess Server Settings`.
4. Click on the new **DirectAccess Server Settings** GPO and it should open up automatically to the **Scope** tab.

   We need to adjust the **Security Filtering** section so that this GPO only applies to our DirectAccess server. This is a critical step for each GPO to ensure the settings that are going to be placed here do not get applied to the wrong computers.
5. Remove **Authenticated Users** that is prepopulated in that list. The list should now be empty.
6. Click the **Add...** button and search for the computer account of your DirectAccess server. Mine is called `RA-01`. By default this window will only search user accounts, so you will need to adjust **Object Types** to include **Computers** before it will allow you to add your server into this filtering list.
7. Your **Security Filtering** list should now look like this:

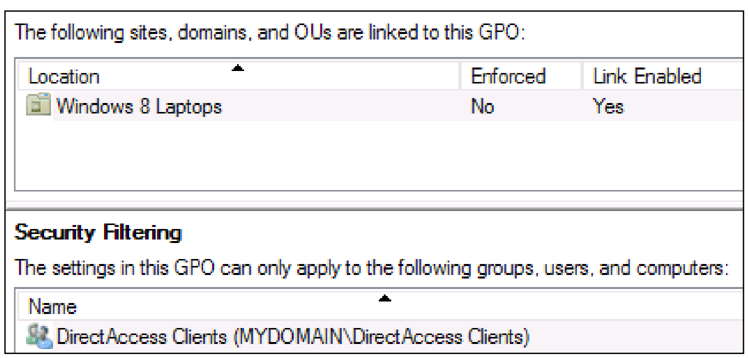



8. Now click on the **Details** tab of your GPO.
9. Change the **GPO Status** to be **User configuration settings disabled**. We do this because our GPO is only going to contain computer-level settings, nothing at the user level.

10. The last thing to do is link your GPO to an appropriate container. Since we have Security Filtering enabled, our GPO is only ever going to apply its settings to the `RA-01` server; however, without creating a link, the GPO will not even attempt to apply itself to anything. My `RA-01` server is sitting inside the OU called **Remote Access Servers**. So I will right-click on my **Remote Access Servers** OU and choose **Link an Existing GPO...**.




11. Choose the new **DirectAccess Server Settings** from the list of available GPOs and click on the **OK** button. This creates the link and puts the GPO into action. Since there are not yet any settings inside the GPO, it won't actually make any changes on the server. The DirectAccess configuration wizards take care of populating the GPO with the settings that are needed.
12. Now we simply need to rinse and repeat all of these steps to create another GPO, something like **DirectAccess Client Settings**. You want to set up the client settings GPO in the same way. Make sure that it is filtering to only the Active Directory Security Group that you created to contain your DirectAccess client computers. And make sure to link it to an appropriate container that will include those computer accounts. So maybe your clients GPO will look something like this:




## How it works...

Creating GPOs in Active Directory is a simple enough task, but it is critical that you configure the Links and Security Filtering correctly. If you do not take care to ensure that these DirectAccess connection settings are only going to apply to the machines that actually need the settings, you could create a world of trouble by internal servers getting remote access connection settings and cause them issues with connection while inside the network.

The key factors here are to make sure your DirectAccess Server Settings GPO applies to only the DirectAccess server or servers in your environment. And that the DirectAccess Client Settings GPO applies to only the DA client computers that you plan to enable in your network. The best practice here is to specify this GPO to only apply to a specific Active Directory security group so that you have full control over which computer accounts are in that group. I have seen some folks do it based only on the OU links and include whole OUs in the filtering for the clients GPO (foregoing the use of an AD group at all), but doing it this way makes it quite a bit more difficult to add or remove machines from the access list in the future.

# Enhancing the security of DirectAccess by requiring certificate authentication

When a DirectAccess client computer builds its IPsec tunnels back to the corporate network, it has the ability to require a certificate as part of that authentication process. In earlier versions of DirectAccess, the one in Server 2008 R2 and the one provided by **Unified Access Gateway** (**UAG**), these certificates were required in order to make DirectAccess work. Setting up the certificates really isn't a big deal at all; as long as there is a CA server in your network you are already prepared to issue the certs needed at no cost. Unfortunately, though, there must have been enough complaints back to Microsoft in order for them to make these certificates "recommended" instead of "required" and they created a new mechanism in Windows 8 and Server 2012 called KerberosProxy that can be used to authenticate the tunnels instead. This allows the DirectAccess tunnels to build without the computer certificate, making that authentication process less secure.

I'm here to strongly recommend that you still utilize certificates in your installs! They are not difficult to set up, and using them makes your tunnel authentication stronger. Further, many of you may not have a choice and will still be required to install these certificates. Only simple DirectAccess scenarios that are all Windows 8 on the client side can get away with the shortcut method of foregoing certs. Anybody who still wants to connect Windows 7 via DirectAccess will need to use certificates on all of their client computers, both Windows 7 and Windows 8. In addition to Windows 7 access, anyone who intends to use the advanced features of DirectAccess such as load balancing, multi-site, or two-factor authentication will also need to utilize these certificates. With any of these scenarios, certificates become a requirement again, not a recommendation.

In my experience, almost everyone still has Windows 7 clients that would benefit from being DirectAccess connected, and it's always a good idea to make your DA environment redundant by having load balanced servers. This further emphasizes the point that you should just set up certificate authentication right out of the gate, whether or not you need it initially. You might decide to make a change later that would require certificates and it would be easier to have them installed from the get-go rather than trying to incorporate them later into a running DA environment.

## Getting ready

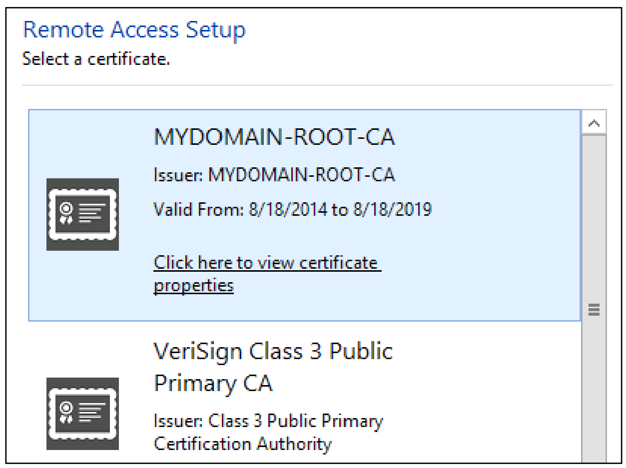
In order to distribute certificates, you will need a CA server running in your network. Once certificates are distributed to the appropriate places, the rest of our work will be accomplished from our Server 2012 R2 DirectAccess server.

## How to do it...

Follow these steps to make use of certificates as part of the DirectAccess tunnel authentication process:

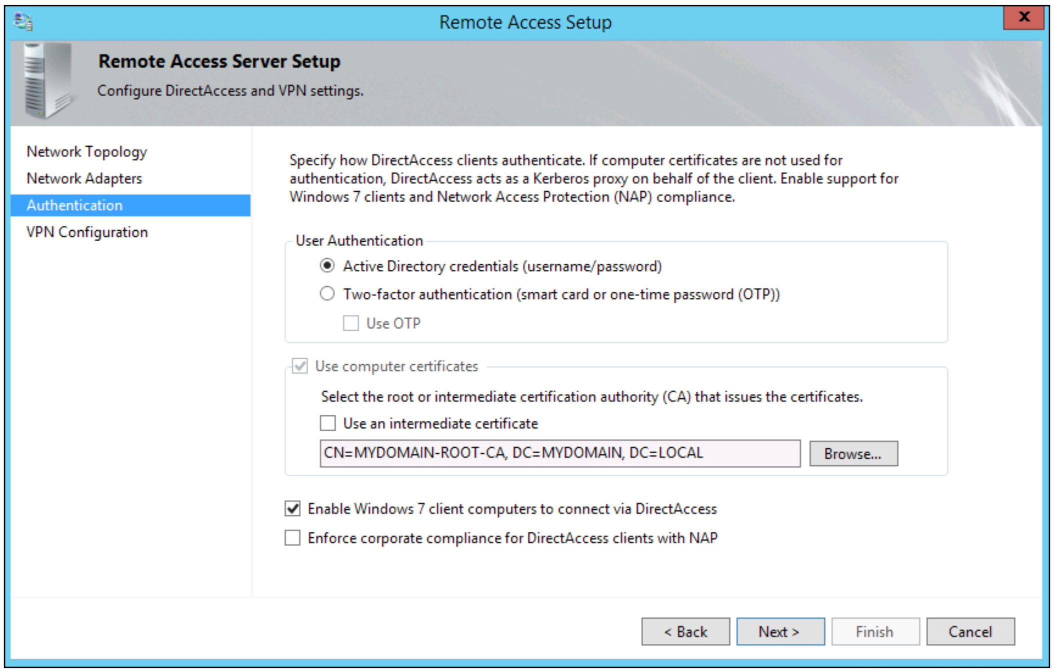
1. The first thing that you need to do is distribute certificates to your DirectAccess servers and all DirectAccess client computers. The easiest way to do this is by using the built-in Computer template provided by default in a Windows CA server. If you desire to build a custom certificate template for this purpose, you can certainly do so. I recommend that you duplicate the Computer template and build it from there. Whenever I create a custom template for use with DirectAccess, I try to make sure that it meets the following criterias:
    - The **Subject Name** of the certificate should match the **Common Name** of the computer (which is also the FQDN of the computer).
    - The **Subject Alternative Name (SAN)** of the certificate should match the **DNS Name** of the computer (which is also the FQDN of the computer).
    - The certificate should serve the **Intended Purposes** of both **Client Authentication** and **Server Authentication**.
2. For the actual distribution of these certificates, I'm going to direct you to review a couple of other recipes in this book. You can issue these certificates manually using **Microsoft Management Console** (**MMC**), as described in the *Using MMC to request a new certificate* recipe in *Chapter 4*, *Working with Certificates*. Otherwise, you can lessen your hands-on administrative duties by enabling Autoenrollment, which is talked about in the *Configuring Autoenrollment to issue certificates to all domain joined systems* recipe in *Chapter 4*, *Working with Certificates*.
3. Now that we have certificates distributed to our DirectAccess clients and servers, log in to your primary DirectAccess server and open up the **Remote Access Management Console**.

4. Click on **Configuration** in the top-left corner. You should now see steps 1 through 4 listed.
5. Click **Edit...** listed under **Step 2**.
6. Now you can either click **Next** twice or click on the word **Authentication** to jump directly to the authentication screen.
7. Check the box that says **Use computer certificates**.
8. Now we have to specify the Certification Authority server that issued our client certificates. If you used an intermediary CA to issue your certs, make sure to check the appropriate checkbox. Otherwise, most of the time, certificates are issued from a root CA and in this case you would simply click on the **Browse...** button and look for your CA in the list.




> This screen is sometimes confusing because people expect to have to choose the certificate itself from the list. This is not the case. What you are actually choosing from this list is the Certificate Authority server that issued the certificates.

9. Make any other appropriate selections on the **Authentication** screen. For example, many times when we require client certificates for authentication, it is because we have Windows 7 computers that we want to connect via DirectAccess. If that is the case for you, select the checkbox for **Enable Windows 7 client computers to connect via DirectAccess**.




## How it works...

Requiring certificates as part of your DirectAccess tunnel authentication process is a good idea in any environment. It makes the solution more secure, and enables advanced functionality. The primary driver for most companies to require these certificates is the enablement of Windows 7 clients to connect via DirectAccess, but I suggest that anyone using DirectAccess in any capacity make use of these certs. They are simple to deploy, easy to configure, and give you some extra peace of mind that only computers who have a certificate issued directly to them from your own internal CA server are going to be able to connect through your DirectAccess entry point.

# Building your Network Location Server (NLS) on its own system

If you zipped through the default settings when configuring DirectAccess, or worse used the Getting Started Wizard, chances are that your **Network Location Server** (**NLS**) is running right on the DirectAccess server itself. This is not the recommended method for using NLS, it really should be running on a separate web server. In fact, if you later want to do something more advanced such as setting up load balanced DirectAccess servers, you're going to have to move NLS off onto a different server anyway. So you might as well do it right the first time.

NLS is a very simple requirement, yet a critical one. It is just a website, it doesn't matter what content the site has, and it only has to run inside your network. Nothing has to be externally available. In fact, nothing should be externally available, because you only want this site being accessed internally. This NLS website is a large part of the mechanism by which DirectAccess client computers figure out when they are inside the office and when they are outside. If they can see the NLS website, they know they are inside the network and will disable DirectAccess name resolution, effectively turning off DA. If they do not see the NLS website, they will assume they are outside the corporate network and enable DirectAccess name resolution.

There are two *gotchas* with setting up an NLS website:

- The first is that it must be HTTPS, so it does need a valid SSL certificate. Since this website is only running inside the network and being accessed from domain-joined computers, this SSL certificate can easily be one that has been issued from your internal CA server. So no cost associated there.
- The second catch that I have encountered a number of times is that for some reason the default IIS splash screen page doesn't make for a very good NLS website. If you set up a standard IIS web server and use the default site as NLS, sometimes it works to validate the connections and sometimes it doesn't. Given that, I always set up a specific site that I create myself, just to be on the safe side.

So let's work together to follow the exact process I always take when setting up NLS websites in a new DirectAccess environment.

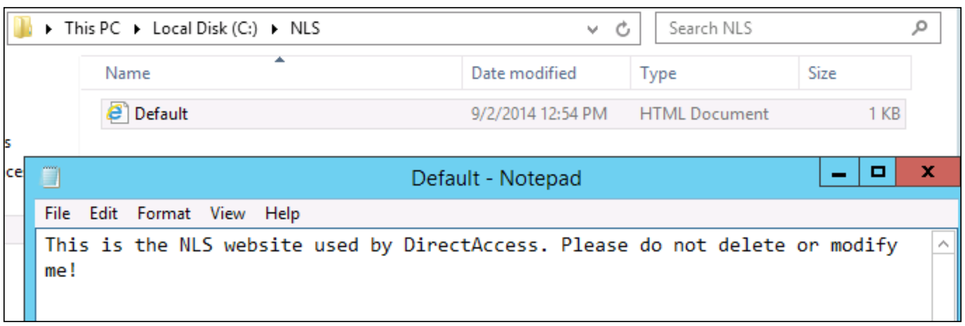
## Getting ready

Our NLS website will be hosted on an IIS server we have that runs Server 2012 R2. Most of the work will be accomplished from this web server, but we will also be creating a DNS record and will utilize a Domain Controller for that task.

## How to do it...

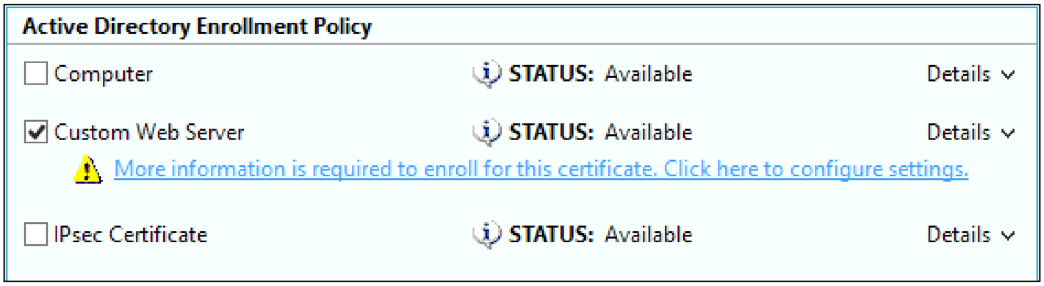
Let's work together to set up our new Network Location Server website:

1. First decide on an internal DNS name to use for this website and set it up in DNS of your domain. I am going to use `nls.mydomain.local` and am creating a regular Host (A) record that points `nls.mydomain.local` at the IP address of my web server.
2. Now log in to that web server and let's create some simple content for this new website. Create a new folder called `C:\NLS`.
3. Inside your new folder, create a new `Default.htm` file.
4. Edit this file and throw some simple text in there. I usually say something like `This is the NLS website used by DirectAccess. Please do not delete or modify me!`.

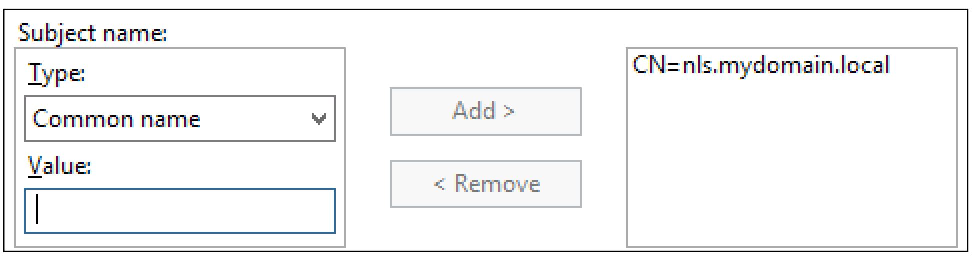



5. Remember, this needs to be an HTTPS website, so before we try setting up the actual website, we should acquire the SSL certificate that we need to use with this site. Since this certificate is coming from my internal CA server, I'm going to open up MMC on my web server to accomplish this task.
6. Once MMC is opened, snap-in the **Certificates** module. Make sure to choose **Computer account** and then **Local computer** when it prompts you for which certificate store you want to open.
7. Expand **Certificates (Local Computer)** | **Personal** | **Certificates**.
8. Right-click on this **Certificates** folder and choose **All Tasks** | **Request New Certificate...**.
9. Click **Next** twice and you should see your list of certificate templates that are available on your internal CA server. If you do not see one that looks appropriate for requesting a website certificate, you may need to check over the settings on your CA server to make sure the correct templates are configured for issuing.

10. My template is called **Custom Web Server**. Since this is a web server certificate, there is some additional information that I need to provide in my request in order to successfully issue a certificate. So I go ahead and click on that link that says **More information is required to enroll for this certificate. Click here to configure settings.**.

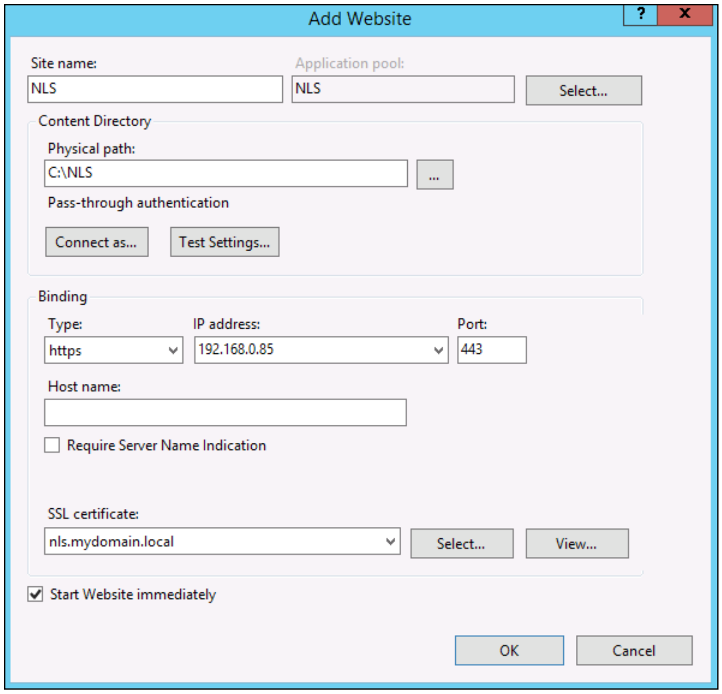



11. Drop-down the **Subject name** | **Type** menu and choose the option **Common name**.
12. Enter a common name for our website into the **Value** field, which in my case is `nls.mydomain.local`.
13. Click the **Add** button and your CN should move over to the right side of the screen like this:




14. Click on **OK** then click on the **Enroll** button. You should now have an SSL certificate sitting in your certificates store that can be used to authenticate traffic moving to our `nls.mydomain.local name`.
15. Open up **Internet Information Services (IIS) Manager**, and browse to the **Sites** folder. Go ahead and remove the default website that IIS automatically set up, so that we can create our own NLS website without any fear of conflict.
16. Click on the **Add Website...** action.

17. Populate the information as shown in the following screenshot. Make sure to choose your own IP address and SSL certificate from the lists, of course:




18. Click the **OK** button and you now have an NLS website running successfully in your network. You should be able to open up a browser on a client computer sitting inside the network and successfully browse to `https://nls.mydomain.local`.

## How it works...

In this recipe, we configured a basic Network Location Server website for use with our DirectAccess environment. This site will do exactly what we need it to when our DA client computers try to validate whether they are inside or outside the corporate network. While this recipe meets our requirements for NLS, and in fact puts us into a good practice of installing DirectAccess with NLS being hosted on its own web server, there is yet another step you could take to make it even better. Currently this web server is a single point of failure for NLS. If this web server goes down or has a problem, we would have DirectAccess client computers inside the office who would think they are outside, and they would have some major name resolution problems until we sorted out the NLS problem. Given that, it is a great idea to make NLS redundant. You could cluster servers together, use **Microsoft Network Load Balancing** (**NLB**), or even use some kind of hardware load balancer if you have one available in your network. This way you could run the same NLS website on multiple web servers and know that your clients will still work properly in the event of a web server failure.

# Enabling Network Load Balancing (NLB) on your DirectAccess servers

DirectAccess is designed so that you always get a single server environment up and running first, before you start tinkering with arrays or load balancing. This way you can validate that all of the environmental factors are in place and working, and that you can successfully build DirectAccess tunnels from your client computers before introducing any further complexity into the design. Once established, however, it is a common next step to take a look at turning up another new server and creating some redundancy for your new remote access solution.

While joining two similar servers together to share load is commonly called clustering and sometimes I hear admins refer to it as such in the DirectAccess world, load balancing DirectAccess servers together actually has nothing to do with Windows Clustering. When you install both the Remote Access role and the Network Load Balancing feature onto your remote access servers, you have already equipped them with all the parts and pieces they need in order to communicate with each other and run an Active/active sharing configuration. The operating system will make use of Windows Network Load Balancing to shuttle the traffic to the appropriate destinations, but everything inside NLB gets configured from the Remote Access Management Console. This gives you a nice visual console that can be used to administer and manage those NLB settings right alongside your other remote access settings.

Once DirectAccess is established and running on a single server, there really are just a couple of quick wizards to run through to configure this Network Load Balancing. However, the verbiage in these options can be quite confusing, especially if you're not overly familiar with the way that DirectAccess transmits packets. So let's take some time to walk through creating an array from our existing DA server and adding a second node to that array.

## Getting ready

We are going to use our existing RA-01 server that is already running DirectAccess. This and our new server, RA-02, are both running Windows Server 2012 R2. They both have the Remote Access role and the Network Load Balancing feature installed. Both are joined to our domain, and have their required certificates (SSL and IPsec) installed for use with DirectAccess. The same SSL certificate has been installed to both servers since they are going to be sharing the load, and all requests to both systems will be coming in from the same public DNS name so they are able to share that certificate.

If your DirectAccess servers are virtual machines, there is one very important prerequisite. You must go into your VM's NIC settings and choose the option for **Enable spoofing of MAC addresses**. Without this box checked for each of the NICs, your network traffic will stop working altogether when you create a load balanced array.

## How to do it...

For the purposes of this recipe, we are going to assume that RA-01 has been configured for use with Teredo, meaning that it has two public IP addresses assigned on the External NIC. We are using this as an example because it is the most complex configuration to walk through when setting up NLB. The same procedure applies for a single IP on the External NIC; it would simply mean that you are only configuring one **Virtual IP** (**VIP**) instead of two.

1. First we need to have a clear understanding of which IP addresses are going to be used where. This is critical information to possess and understand before trying to start any kind of configuration.The current RA-01 IP addresses are as follows:
    - External IPs: 1.1.1.10 and 1.1.1.11
    - Internal IP: 192.168.0.25
    - These three IP addresses that are currently running on RA-01 are going to turn into our Virtual IPs (VIPs). These are the IP addresses that are going to be shared between both DirectAccess servers. Since we are changing the roles of these IPs, this means that we need to dedicate new **Dedicated IPs** (**DIPs**), both internally and externally, to both RA-01 and RA-02.

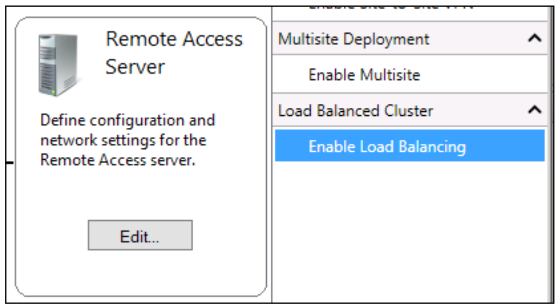
    New IP address assignments are shown as follows:
    - External VIPs (shared): 1.1.1.10 and 1.1.1.11
    - Internal VIP (shared): 192.168.0.25
    - RA-01 External DIP: 1.1.1.12
    - RA-01 Internal DIP: 192.168.0.26
    - RA-02 External DIP: 1.1.1.13
    - RA-02 Internal DIP: 192.168.0.27

    So to summarize, when using Teredo (dual public IPs) and creating a two-node DirectAccess server load balanced array, you will need a total of four public IP addresses and three internal IP addresses.

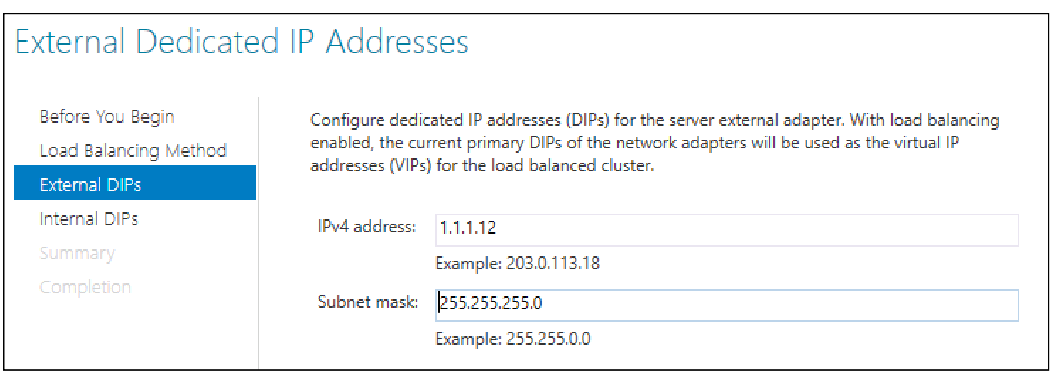
2. On RA-01, we are going to leave the VIPs in place for now. The DirectAccess wizards will change them for us later.
3. On the new RA-02 server, set its final DIP addresses on the NICs. So in our example, the External NIC gets 1.1.1.13 and the Internal NIC gets 192.168.0.27.
4. There are only four steps to take on a DirectAccess array node server such as RA-02:
    - Assign IP addresses.
    - Join it to the domain.
    - Install the certificates.
    - Add the Remote Access role and Network Load Balancing feature.

The remainder of its configuration is accomplished from RA-01.

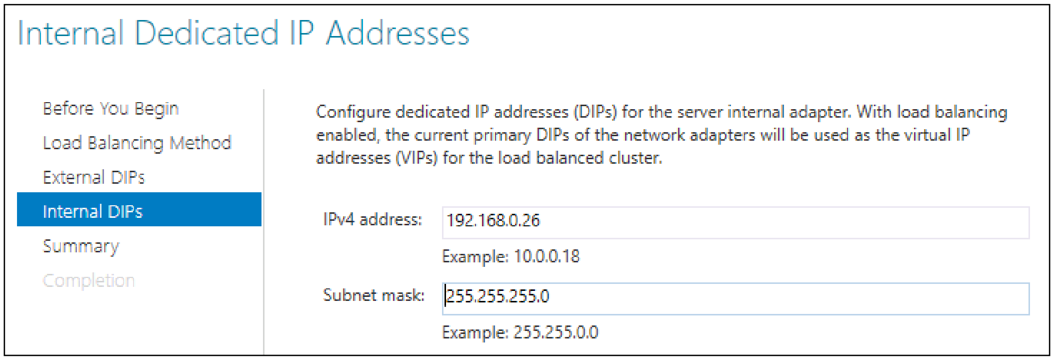
5. On RA-01, your primary DirectAccess server, open **Remote Access Management Console**.
6. In the left window pane, click on **Configuration** | **DirectAccess and VPN**.
7. Now over in the right **Tasks** pane, down at the bottom choose **Enable Load Balancing**.




8. Click **Next**.
9. Choose **Use Windows Network Load Balancing (NLB)**. You can see there is also an option for using an external load balancer, if you have one available to you. I find that the majority of customers utilize the built-in NLB, even when hardware load balancers are available.
10. The next screen is **External Dedicated IP Addresses**. This is where things start to get confusing and mistakes are often made. If you read the text on this screen, it is telling you that the current IP addresses assigned to the NICs are now going to be used as Virtual IPs (VIPs). You do not need to specify anything about the VIPs on this screen. Instead, what we are doing on this screen and the next is specifying what *new* Dedicated IPs (DIPs) are now going to be assigned to the physical NICs on this server. First, since this is the external screen, we specify our new public IP that will be used by RA-01:

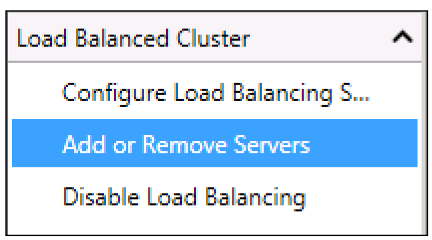


11. On the following screen, do the same thing but this time for the Internal NIC. The current IP address of `192.168.0.25` is going to be converted over into a shared VIP, and so we need to specify the new Internal DIP that is going to be assigned to RA-01's Internal NIC:



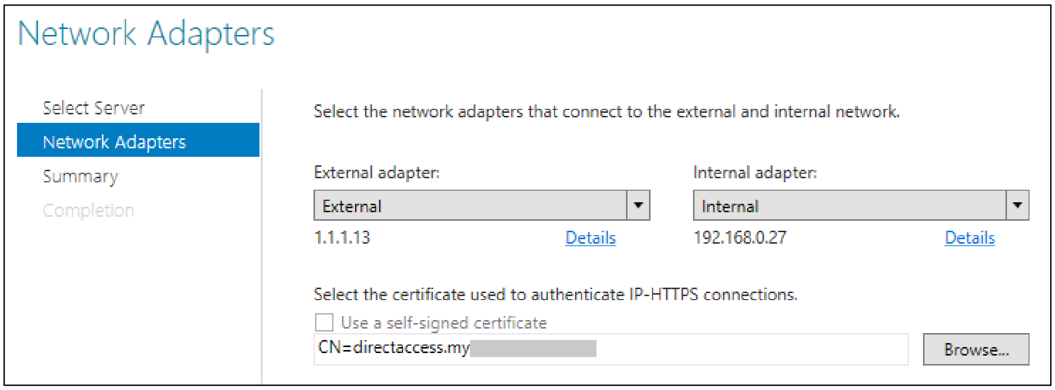

[ Now you can see why having a definitive list of IP addresses before starting this wizard is important! ]

12. Click **Next** then, if everything looks correct in the **Summary** screen, go ahead and click on the **Commit** button. This will roll the changes into the GPO settings and apply the changes to our RA-01 server. Remember, nothing has been done to RA-02 yet as we haven't specified anything about it in these screens. We now have an active "array", but so far there is only one member, RA-01.
13. Now that you are back inside the main **Configuration** screen, go ahead and click on **Load Balanced Cluster** | **Add or Remove Servers**.

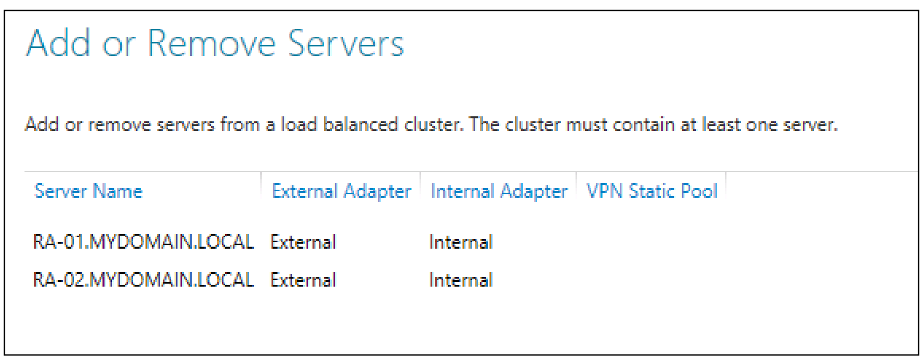



14. Click on the **Add Server...** button.
15. Input the FQDN of your second server. Mine is `RA-02.MYDOMAIN.LOCAL`. Then click **Next**.

16. If you have appropriately configured your second remote access server with correct IP address information and the certificates that it needs, the **Network Adapters** screen should self-populate all of the necessary information. Double-check this info to make sure it looks correct and click **Next**.




17. If the **Summary** page all looks correct, click on the **Add** button.
18. Click **Close**. Then back at the **Add or Remove Servers** screen, you should now see both of your remote access servers in the list. Go ahead and click on the **Commit** button to finalize the addition of this second node.




Following the addition of the second node, I always go back into the NIC properties of both NICs on both servers and make sure that all of the expected IP addresses got added correctly. Sometimes I find that the wizard is not able to successfully populate all of the VIPs and DIPs, and that I have to add them manually afterwards. Each NIC now has a specific DIP, as listed in the beginning of this recipe. In addition to those DIPs, the External NIC on each server should also list both External VIPs, and the Internal NIC on each server should list the Internal VIP. The TCP/IPv4 properties of the NICs sure look to be overly-populated with IP addresses, but this is all normal and well for a successfully load balanced DirectAccess array.

## How it works...

The ability to load balance DirectAccess servers together right out of the box with Windows Server 2012 R2 is an incredibly nice feature. Redundancy is key for any good solution, and configuring this array for an Active/active failover situation is a no-brainer. While the wizards for enabling NLB are centralized right alongside all the other DirectAccess settings, they can certainly be confusing when running through them for the first time. As with any system whose job is to shuttle network traffic around, planning correctly for IP addressing and routing is key to the success of your DirectAccess NLB deployment. Hopefully this recipe helps to clear up questions surrounding this commonly requested task on our remote access servers.

Following the creation of your array, you will notice that navigation through some of the screens inside the Remote Access Management Console has changed slightly. When you access screens such as **Configuration** to make changes, **Operations Status** to check on the status of your servers, or **Remote Client Status** to see what clients are connected, you will now notice that the nodes are listed separately. You can now click on the individual node name to see information on those screens that is specific to one particular server in the array, or you can click on the words **Load Balanced Cluster** in order to see information that is shared among all of the array members.

One other important note. Now that we have a load balanced array up and running, it is easy to add a third node to this array as well! Your DirectAccess array can grow as your company grows, up to eight node servers if so required. Simply join additional servers into this array by using the **Load Balanced Cluster | Add or Remove Servers** task.

# Adding VPN to your existing DirectAccess server

It is fairly common when starting work with the new Remote Access role for administrators to choose the option for "Deploy DirectAccess only". Maybe you initially thought this box was only going to be used for DA, or that all of your client connections would be handled by only the DA role. While this is true for some organizations, it is pretty common to find some benefit out of having both DirectAccess and VPN configured on your remote access entry point. Maybe you have some mobile phones or personal tablets that you want connected to the corporate network. Or perhaps you want to give the ability for home computers, or even Macs, to connect in remotely. These would be scenarios that are outside the scope of DirectAccess and would require some other form of VPN connectivity.

Making significant changes on a production server can be intimidating, and you want to make sure that you select the right options. Also, IP addressing remote access servers isn't always a cakewalk, and so it would be natural to assume that turning a DirectAccess server into a DirectAccess plus VPN server would involve some additional IP addressing. You would actually be wrong about that last one.

VPN can share the public IP address already configured and running for your DirectAccess clients, so thankfully when you decide to add VPN to your server, you don't have to reconfigure the NICs in any way. Since we don't have to make networking changes first, let's jump right into taking our production DA server and adding the VPN role to it.

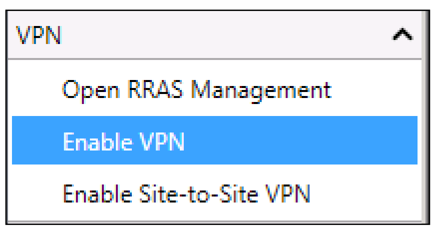
## Getting ready

We are working today from our new DirectAccess server, which is a Windows Server 2012 R2 that has the Remote Access role installed.

## How to do it...

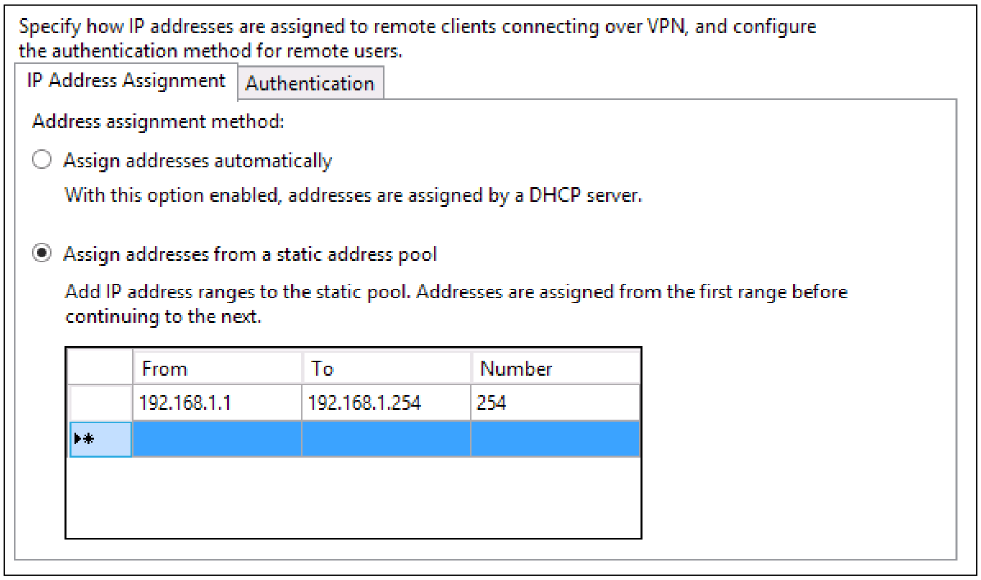
To add VPN functionality to your existing 2012 R2 DirectAccess server:

1. Open **Remote Access Management Console** from the **Tools** menu inside Server Manager.
2. In the left window pane, browse to **Configuration** | **DirectAccess**. This is the screen where you see the four setup steps listed in the middle.
3. Now over on the right, you see a section of buttons related to **VPN**. Go ahead and click on **Enable VPN**.




4. You will receive a pop-up message asking you to make sure you intend to configure VPN settings on this server. Go ahead and click **OK**. This will cause your remote access server to spin through some processes, reaching out to the GPOs and reconfiguring the necessary settings so that they include VPN connectivity.
5. VPN is now enabled on our server, but we have yet to configure IP addressing that will be handed out to the client computers. Once you are back at the main **Configuration** screen, click the **Edit...** button listed under **Step 2**.
6. There is now a fourth screen available inside this mini-wizard called **VPN Configuration**. Go ahead and click on that.

7. If you want VPN clients to pull IP addresses from an internal DHCP server, leave the radio button set to the top option. If you would rather specify a particular range of IP addresses that should be handed out to client computers, choose **Assign addresses from a static address pool** and specify the range of addresses in the given fields.




> When you specify a static range like this, your remote access server will start handing out these addresses to the client computers that connect using VPN. However, these client computers will most likely not be able to connect to any internal resources without a little additional networking consideration. When you create a static address pool for assigning IP addresses to VPN clients, there are two rules you need to keep in mind:
>
> - The pool of addresses to be handed out to clients should come from a subnet that does not exist in the remote access server's internal routing table. For example, my network is 192.168.0.x and I am going to assign VPN client licenses from 192.168.1.x.
> - You need to set the default route for this other subnet so that it points back to the internal IP address of the remote access server. Without doing this, traffic from the VPN clients might make its way into the 192.168.1.x subnet, but responses from that subnet aren't going to know how to get back to the VPN client computers. By setting a default route on the 192.168.1.x subnet to point back to the Internal NIC of the remote access server, you fix this.

## How it works...

The act of enabling VPN on a DirectAccess server is a single action, but without a couple of extra configuration steps, that VPN enablement isn't going to do much for you. With this recipe, you should now have the information you need to enable and configure VPN on your remote access server and get those machines connected that do not meet the requirements to be DirectAccess -connected. In the field, I find that most companies try to get all the computers they can connected via DirectAccess, because it is a much easier technology to deal with on the client side and is better for managing domain joined systems. When faced with the need to connect computers that aren't Windows 7 or Windows 8 or are not domain joined, it is nice to know that traditional VPN connectivity options exist right in our Server 2012 R2 operating system.

# Replacing your expiring IP-HTTPS certificate

DirectAccess has the ability to utilize certificates in a couple of different ways. Depending on how you configure DA, there are different places that certs may or may not be used, but one common variable to all DirectAccess implementations is **IP-HTTPS**. This is a transition technology that is always enabled on a DirectAccess server, and it requires an SSL certificate to work properly. IP-HTTPS traffic is coming in from the Internet, and so I always recommend that the SSL certificate used for the IP-HTTPS listener to be one purchased from a public CA entity.

As with any SSL certificate, they are only valid for a certain time period. Typically these certificates are purchased on a one, two, or three year basis. This means that eventually, you will have to renew that certificate and figure out how to make DirectAccess recognize and utilize the new one. IP-HTTPS makes use of a web listener inside IIS, and so it is a natural assumption that when you need to change your certificate, you do so inside IIS. This is an incorrect assumption. What's worse is that you can actually dig into the site inside IIS and change the certificate binding, and cause it to work for a while. This is not the correct place to change the certificate! If you simply change the binding inside IIS, your change will eventually be reversed and it will go back to using the old certificate. Unfortunately, I get calls quite regularly from customers who do this exact thing and then have all sorts of users unable to connect remotely because the DirectAccess server has resorted back to using the old, now expired, certificate.

Let's work through this recipe together to configure our DirectAccess to utilize a new certificate that was recently purchased and installed onto our server.

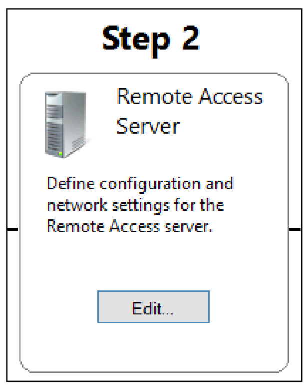
## Getting ready

We have DirectAccess up and running on our Windows Server 2012 R2 Remote Access server. Our SSL certificate that we use for IP-HTTPS is about to expire and we have renewed it with our CA. The new copy of the certificate has already been downloaded and installed onto the server itself, so now we just need to figure out where it needs to be adjusted for DirectAccess to start using it.

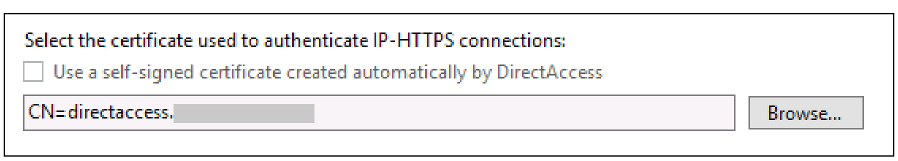
## How to do it...

To adjust the DirectAccess configuration to start using a new certificate for the IP-HTTPS listener:

1. Open **Remote Access Management Console** on your DirectAccess server.
2. In the left window pane, browse to **Configuration** | **DirectAccess and VPN**.
3. Under **Step 2** of the configuration, click on **Edit...**.




4. Click **Next**.
5. You will now see the currently assigned certificate for IP-HTTPS. This is the certificate that is about to expire. Go ahead and click on the **Browse...** button.

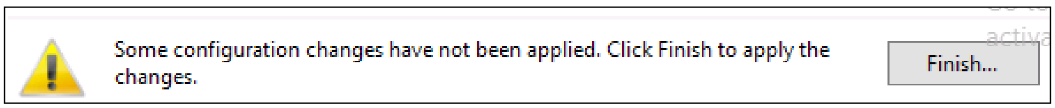



6. Now simply choose the new certificate with the new expiration date from the list of available certificates that opens.

7. Click **Next** a couple more times to finish up the **Step 2** wizard.

> Keep in mind that the IP-HTTPS certificate is a per-node setting. If you have an array of multiple DirectAccess servers, you make all changes from the primary server's console, but you must install the certificate on each server and then make the certificate change on each node separately within the configuration.

8. At this point, nothing has actually been changed with the live configuration. To make this change active, you need to press the **Finish...** button that is available near the bottom of **Remote Access Management Console**.




9. If everything in the review looks good, click on **Apply** and this will push your changes into action. The new certificate is now in place and working to validate those IP-HTTPS connections.

## How it works...

Replacing the SSL certificate that is used by IP-HTTPS is a regular and necessary task for any DirectAccess server administrator, but is a task that only comes maybe once per year. This generally means that, by the time your certificate expiration date rolls around, you have likely forgotten where this setting is in the configuration. I hope this recipe can be a quick reference to alleviate that worry.

I always check the certificate from outside the network after making the change to ensure the new certificate is really the one that is now live on the system. If you take a computer outside of your network on the Internet, try browsing to a dummy site off your public DNS record on your DirectAccess server. For example, if the public DNS record that you are using on your server is `directaccess.contoso.com`, try browsing to `https://directaccess.contoso.com/test`. You can expect to get a 404 error because the page we are requesting doesn't actually exist, but when you get the 404 you have the ability (depending on what browser you are using—I tend to prefer Chrome for this task) to view which certificate is being used to validate your web traffic. Click to view the certificate details and make sure that it is your new certificate with the newest validity dates. Further, if you encounter any kind of certificate warning message when you are trying to browse to this test website, this likely indicates that there is some kind of problem with the certificate and you may need to investigate that further.

# Reporting on DirectAccess and VPN connections

One of the big benefits that Microsoft brought to the table in these newer versions of the remote access role is reporting. In the past, it was difficult to tell who was connected and even harder to find out what they were doing or when they had been connected previously. Historical reporting on remote sessions was kind of absent. All of that changes in the newer editions, as we now have a nice interface to show us who is connecting, how often they are connecting, and even some information on what things they are doing while they are connected. Today we'll take a look into those interfaces and explore some of the information that is available to consume. We will also make sure you know how to turn on the historical reporting, as it is not enabled by default.

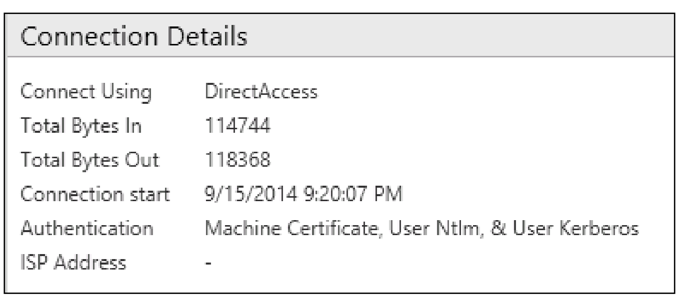
## Getting ready

All work with this recipe will be accomplished from our Windows Server 2012 R2 Remote Access server that is servicing both DirectAccess and VPN clients.

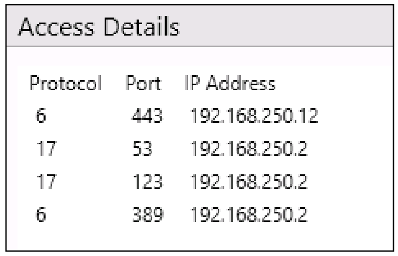
## How to do it...

Follow these steps to get familiar with the remote access reporting options available in Server 2012 R2:

1. Open **Remote Access Management Console** from the **Tools** menu inside Server Manager.
2. In the left window pane, browse to **Remote Client Status**. Here you will see a list of all currently connected devices and users. This shows both DirectAccess connections as well as VPN connections.
3. If you click on a particular connection, you will see some additional data displayed below. You can easily find out whether the user is connected using DirectAccess or VPN, and some more specific information about their connection:

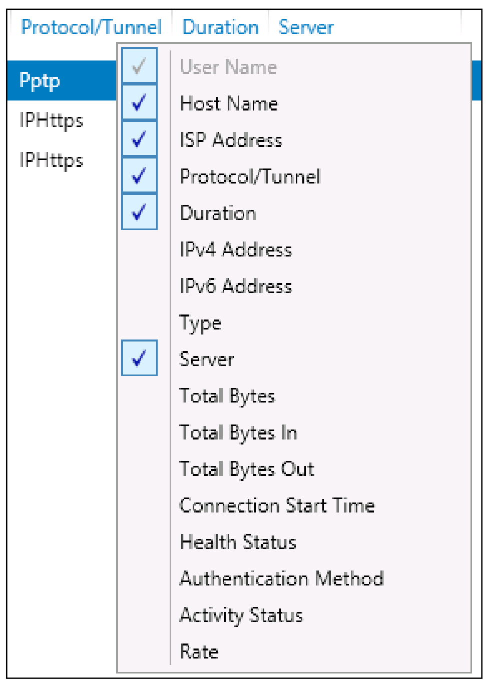


4. Look over toward the left a little where is says **Access Details** and you can even see what internal resources have been accessed by the user and computer.

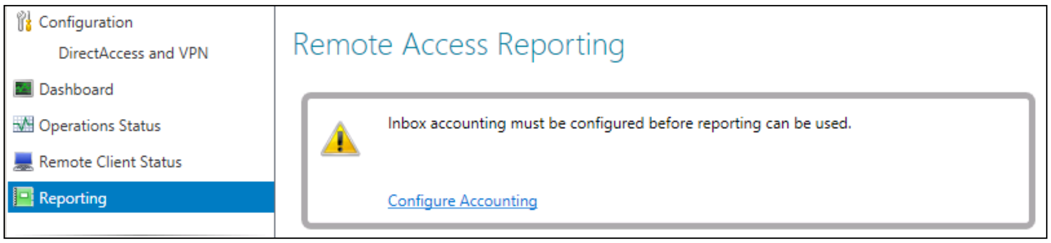

Access Details

| Protocol | Port | IP Address |
|---|---|---|
| 6 | 443 | 192.168.250.12 |
| 17 | 53 | 192.168.250.2 |
| 17 | 123 | 192.168.250.2 |
| 6 | 389 | 192.168.250.2 |

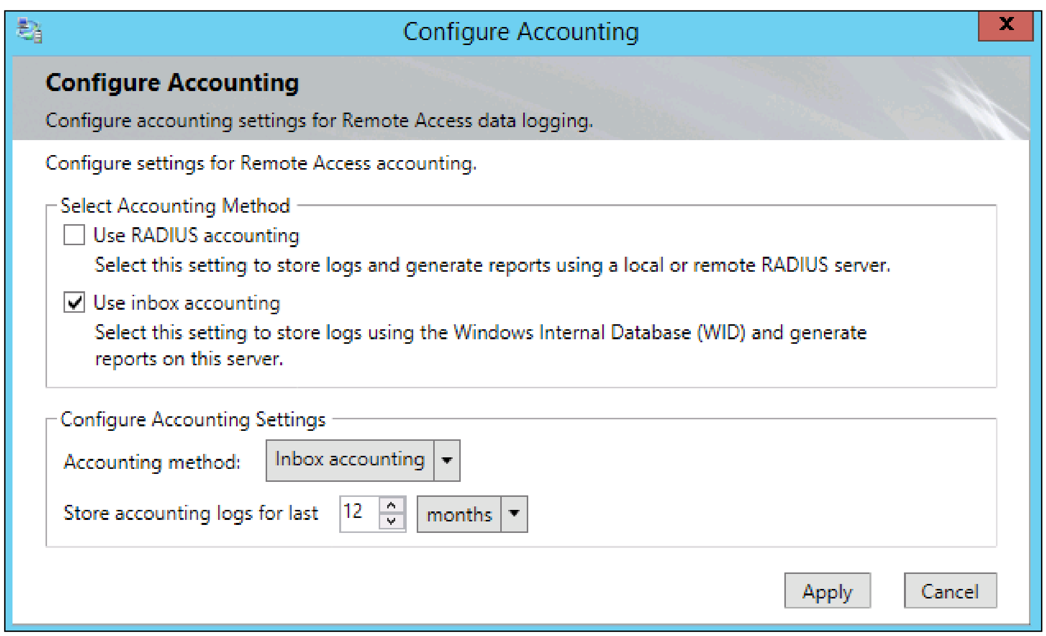
5. Once your environment is large enough that this screen becomes filled with connections, the **Search** box at the top comes in very handy. You simply type in any information you want to search for, and the results in the window will filter down to your search criteria.
6. If you would like to display more data on the screen, you can right-click on any of the existing column names and select additional columns to show or hide.



7. All of this information is great! But what if we want to look back and view this data historically? Maybe you want to view connections from the past day, or week. Maybe you need to come up with some kind of report on how many connections happened over the past month. In the left window pane, click on **Reporting** to get started with that.
8. Since reporting is not enabled by default, we don't have any data here yet. Instead, you will see a message indicating that you need to configure accounting. Go ahead and click on this link.




9. Now you have options for **Use RADIUS accounting**, **Use inbox accounting**, or both. RADIUS accounting implies that you have a RADIUS server set up and ready to accept this kind of data. I don't see many customers use this option. Instead, most select **Use inbox accounting**, which writes all of the data right to the Windows Internal Database (WID) on the DirectAccess server itself.



10. Once you have made your selection, click **Apply**. You will see that the **Reporting** screen now looks a lot more like the **Remote Client Status** screen, except that inside **Reporting** you have additional options to select date ranges and pull historical information.

## How it works...

The reporting of user connection data is critical to most remote access systems. The inclusion of this data, particularly for historical connections, is a great feature add that I am sure every remote access administrator is going to make use of. With a simple configuration change, we set up our Windows Remote Access server to keep track of these DirectAccess and VPN connections so that we can run and save reports on that data in the future.

# 7
# Remote Desktop Services

**Remote Desktop Services** (**RDS**) is an outstanding way to provide users with access to applications and data, without those applications and data needing to reside on their local workstations. Formerly known as Terminal Services, this technology enables companies to retain control of all data and apps on centralized Remote Desktop servers, which users connect to from their workstations in order to access these items. There are two primary means of providing this information to the users. First is through a remote session, where users log into a **Remote Desktop Session Host** (**RDSH**) server and end up landing inside a session hosted on the server. This session looks and feels like a regular desktop computer to the user, as they have a full desktop and Start button and are able to launch any application available to them within that session. They are also able to save documents inside their session, keeping everything centralized. This is the most common flavor of RDS that I see used in the field, and is where we will focus the majority of our administrative tasks that we discuss today.

A second way to provide data to users via RDS is **RemoteApp**. This is a neat function that is able to provide only the application itself remotely to the user's computer, rather than a full desktop session. This is a nice way to further restrict the access that is being provided to the user, and simplifies the steps the user must take in order to access those resources.

An RDS environment has the potential to contain many servers, enough to fill its own book. Given that, let's work together to get a simple RDS environment online that you can start testing with, and provide you with knowledge of some common administrative tasks that will be useful in an environment like this.

In this chapter, you'll be taking a look at the following recipes:

- Building a single server Remote Desktop Services (RDS) environment
- Adding an additional RDSH server to your RDS environment
- Installing applications on an RD Session Host server
- Disabling the redirection of local resources
- Shadowing another session in RDS
- Installing a printer driver to use with redirection
- Removing an RD Session Host server from use for maintenance
- Publishing WordPad with RemoteApp

# Introduction

I would like to take a minute and describe the different parts and pieces that could potentially make up your RDS environment. We won't be covering the installation or use of all components that might be involved with a full RDS deployment, but you should at least be aware of the components and their intended functions.

- **Remote Desktop Session Host**: This is the most common type of RDS server, as it is the one hosting the programs and sessions that the users are connecting to. Depending on the size of your environment, there may be many of these servers running concurrently.
- **Remote Desktop Connection Broker**: This guy is like the load balancer for RDS servers. It distributes users evenly across Remote Desktop Session Host (RDSH) servers, and helps users to reconnect to existing sessions rather than creating fresh ones.
- **Remote Desktop Licensing**: This one is responsible for managing the licenses that are required for RDS use in a network.
- **Remote Desktop Gateway**: This is a gateway device that can bring remote users out on the Internet into an RDS environment. For example, a user at home could utilize the connection provided by an RD Gateway in order to access work information.
- **Remote Desktop Web Access**: This enables users to access desktops and applications by using the local Start menu on their Windows 7 or 8 computer. Users can also utilize this to access applications via a web browser.
- **Remote Desktop Virtualization Host**: This is a role that integrates with Hyper-V in order to provide virtual desktop sessions to users that are spun up from Hyper-V, rather than shared resources like an RDSH.

Many of these roles can be placed together on a single server, which is what we will be doing in our recipe to bring a simple RDS environment online. As your deployment grows and you continue to add users and servers, it is generally a good idea to make these roles decentralized and redundant when possible.

# Building a single server Remote Desktop Services (RDS) environment

If you aren't coming into an environment where RDS is already up and running, it will be helpful to understand where the roles come from and how they are put into place. In this recipe, we are going to combine a number of the Remote Desktop roles onto a single server so that we can take a look at that installation process. When we are finished, we should have an RDS server that will allow users to connect and utilize a Remote Desktop session.

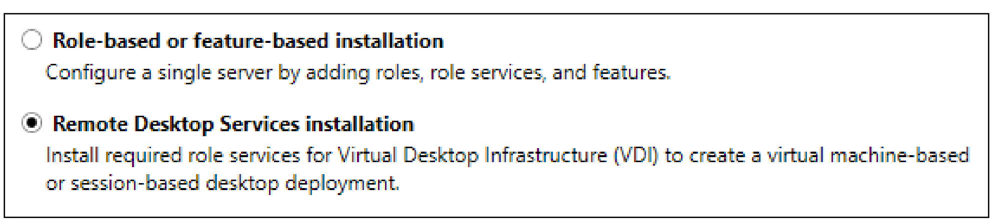
## Getting ready

We will be using a Windows Server 2012 R2 machine to install the RDS roles. This server is already joined to our domain.

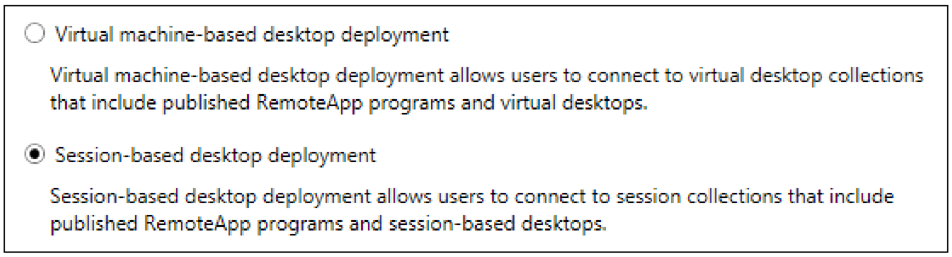
## How to do it...

The following steps will direct you on installing the roles necessary for starting your first simple RDS server:

1. Open up **Server Manager** and click on the link **Add roles and features**.
2. Click **Next**, which will bring you to the **Installation Type** screen. This is where we deter from normal as far as role installations go. For the majority of roles, we tend to blow right through this screen without a second thought. For Remote Desktop Services, though, we need to make a change on this screen.
3. Choose the option for **Remote Desktop Services installation**. Then click **Next**.

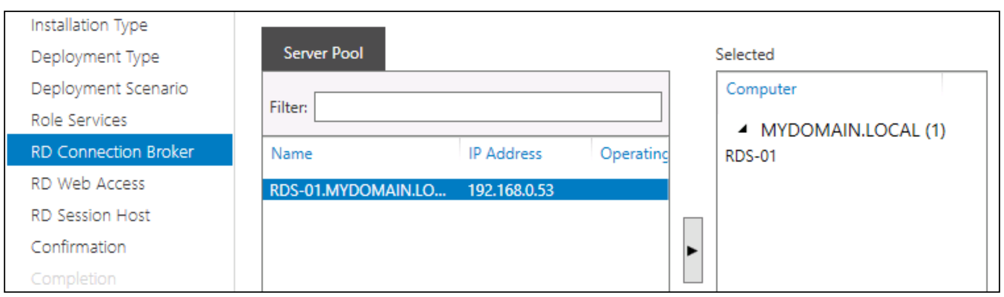


4. Leave the default setting as **Standard deployment** and click **Next**. On this screen, we could choose the other Quick Start option since we are intending to only configure a single server at this time. I am choosing not to go this shortcut route because we want a good look at the different services that are going to be installed, and want to leave our installation open to having multiple RDS servers down the road.
5. With this RDS server, we are planning to provide access to traditional desktop sessions, not the integration with Hyper-V. So on the **Deployment Scenario** screen, choose **Session-based desktop deployment**.

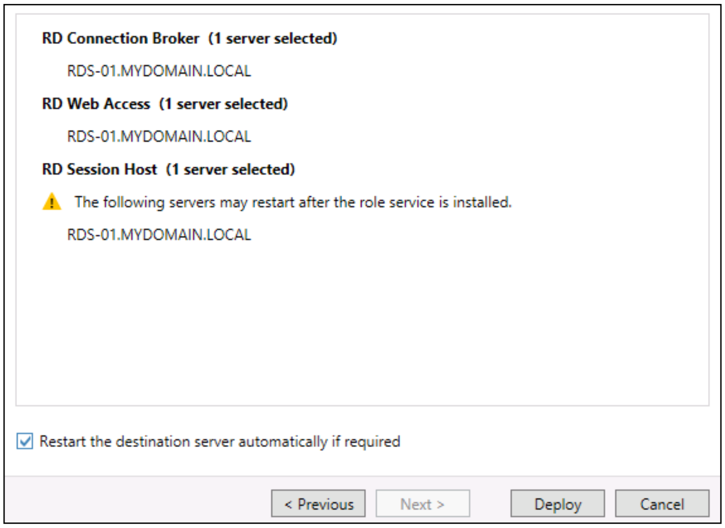



6. We now see a summary of the role services required for our installation. Based on the options we have chosen, you should see **RD Connection Broker**, **RD Web Access**, and **RD Session Host** in this list. The next few screens will be used to define which servers are going to be used for these roles.

   Since we are installing everything onto a single server for now, we only have one option in the **Server Pool** list and we simply move it over to the right column. Go ahead and click the arrow to do this on the **RD Connection Broker** page.




7. Now do the same thing on the next two screens. In our example, we are using the server named RDS-01, so I am going to use it as both the RD Web Access server as well as the RD Session Host server.
8. Now you should be up to the **Confirmation** screen, which gives you a summary of the actions about to be performed. For us, all three RDS services are being installed onto the RDS-01 server. We must now check the box that says **Restart the destination server automatically if required** and then press the **Deploy** button.




## How it works...

We can follow this recipe to get our first simple RDS environment up and running. Our server will now allow users to connect and access virtual sessions that are hosted right on this RDS-01 server. To log in, users may either launch the Remote Desktop Connection tool on their client computer and type in the `RDS-01` name of our server, or open up a web browser and head over to `http://rds-01/rdweb`. Either way, they will land inside a desktop session that looks and feels pretty similar to a Windows 8 desktop. Inside this desktop provided by the RDS server, they are able to launch applications and save documents, having everything run and stored right on the server itself rather than their local desktop computers. From this simple, single server RDS implementation, we can build and grow out to provide additional RDS roles on more servers, or for the purposes of handling additional user loads.

# Adding an additional RDSH server to your RDS environment

Most RDS implementations start out with a single server, or at least a single Remote Desktop Session Host (RDSH). Once you have the roles established for successful connectivity here, it is a natural next step to add additional RDSH servers to accommodate more users. Or perhaps you want to segregate different types of users (and their applications) onto different RDSH servers. Whatever your reasoning, chances are that at some point you will want to add additional servers into your RDS environment. Let's add a second server to ours so that you can see how this process works.

## Getting ready

We have a single RDS server online, running Windows Server 2012 R2. It is named RDS-01 and is already performing the roles of RD Connection Broker, RD Session Host, and RD Web Access. We will now use the management interface on RDS-01 in order to add a second RDSH server to our infrastructure. The name of our new server is RDS-02, and it is already joined to our domain.

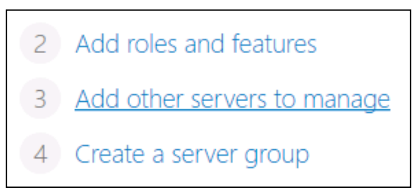
## How to do it...

Follow these steps to add a new RDSH server into our existing RDS environment:

1. On the existing RDS server, RDS-01, open up **Server Manager**.

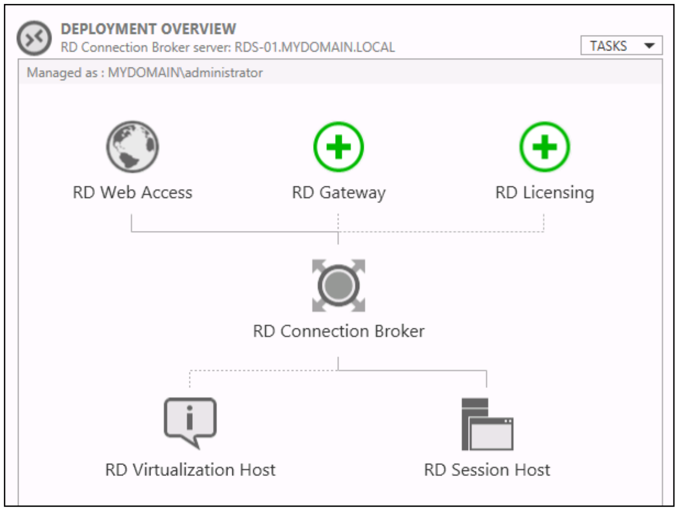
   > We have to add our new RDS-02 server into this instance of Server Manager that is running on RDS-01. Until we perform this step, we will be unable to make modifications to RDS-02 from here.

2. Click on the link **Add other servers to manage**.

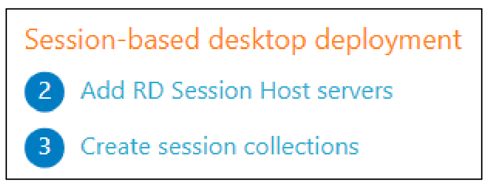
   
   

3. Type in the name of the new server that you intend to turn into an RDSH. For our example, the server name is RDS-02. Then click the arrow to add this server into the **Selected** list and click on **OK**.
4. Now back in the main page of Server Manager, in the left window pane go ahead and click on the listing for **Remote Desktop Services**. This will bring us into the management interface for RDS.

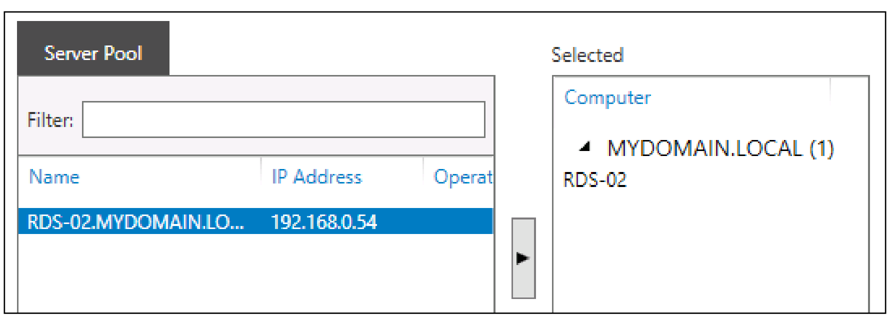
   Take a look at the **DEPLOYMENT OVERVIEW**, which is a self-generated diagram of what your current RDS deployment looks like. Since we are only testing with our current servers and not accessing them from outside the network, we see plus symbols next to **RD Gateway** and **RD Licensing**. This simply means we have not yet configured these roles, and we could click on those pluses and follow the prompts if we intended to do so. We have no requirement for these services at the present time, so we will ignore this for now.




5. To add a new RDSH server, head over to the top right of this window and click on the link that says **Add RD Session Host servers**.




6. Since we added it into Server Manager earlier in the recipe, we should now be able to see in this list the new server available to select. Select the new RDS-02 server and click the arrow to move it into the **Selected** column. Then click **Add**.




7. Click the **Next** button, and on the **Confirmation** screen you will need to check the box that says **Restart remote computers as needed**. Then click on **Add**.

## How it works...

In this recipe, we used the Remote Desktop Service management console on our primary RDS server to take a new server that we had running and turn it into a Remote Desktop Session Host (RDSH) server. This RDSH is now part of our RDS infrastructure, and can be managed right from this centralized management platform. In an RDS environment, this is typically the way that new roles are added onto servers that are being brought into the environment. Using this management console to perform many tasks in RDS makes a lot of sense, because it is easy to see the big picture of your RDS infrastructure as you make changes or updates.

# Installing applications on an RD Session Host server

As soon as you take a Windows Server and turn it into a Remote Desktop Session Host server to be used within an RDS environment, the way that applications work on that server changes significantly. Whenever programs and apps are installed onto that RDSH, it first needs to be put into a special **Install Mode**. Placing the server into Install Mode prior to launching the program installer is important to make sure that the applications are going to be installed in a way that will allow multiple users to run them simultaneously. Remember, our RDSH servers will be hosting multiple user sessions, probably dozens of them.

Using Install Mode is so important to applications working properly on an RDSH, that you really should not install any programs onto the server before you turn it into an RDSH. Once that role has been established, then apps can be safely installed, as long as you are using Install Mode. Programs installed prior to converting that server into an RDSH may not work properly, and you might have to uninstall and reinstall them. There are a couple of different ways that Install Mode can be invoked during a program installation; let's take a look at both of them.

## Getting ready

We have the requirement to install a program onto our RDSH server. This box is running Windows Server 2012 R2 and is already part of our RDS environment. We will also need, of course, the application installer files that we intend to launch.

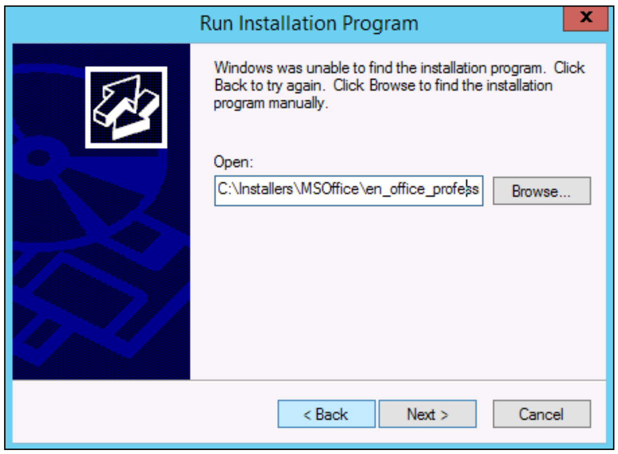
## How to do it...

One way to properly install programs onto an RDSH is by using Control Panel to install the application:

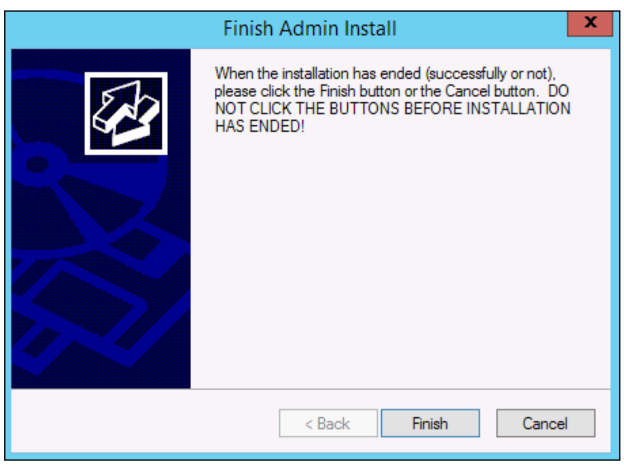
1. Right-click on your **Start** flag and choose to open Control Panel.
2. Click on **Programs**.

3. Choose the button that says **Install Application on Remote Desktop...**.


4. Click **Next** and you will be able to specify the location of your installer file for the application.

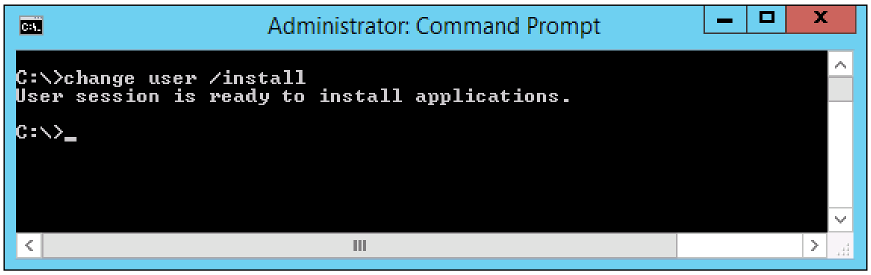



5. Click **Next** and your program will install. When finished, make sure to click the **Finish** button on that Install Mode mini-wizard screen, so that the RDSH is placed back into Execute Mode and is ready for normal operation.

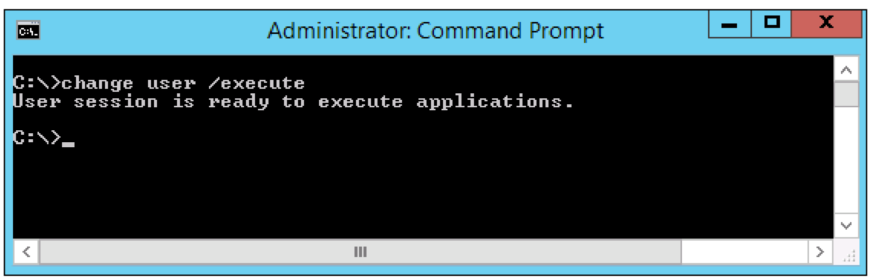


The second way to place an RDSH into Install Mode is by using the command prompt:

1. Right-click on the **Start** flag and choose to open command prompt (admin).
2. Type `change user /install` and press *Enter*.




3. Now find your program installer file and launch it. Walk through the installation steps in the same way you would on any regular server or computer.
4. Once the program has finished installing, head back to the command prompt window and now type `change user /execute`. Then press *Enter*. This takes the RDSH out of the special Install Mode and places it back into normal Execute Mode.




> Restarting the server also automatically places it back into Execute Mode. So if your application installer asks you to restart as part of the installation process, your RDSH will be placed back into Execute Mode when it boots, and in that case you do not have to enter the command manually.

## How it works...

When installing applications onto an RDSH, it must first be placed into a special Install Mode. Doing this re-maps certain parts of the program being installed, so that it can be run and utilized by many users at the same time. Installing your applications by using one of the methods discussed in this recipe will be critical to the success of your RDS environment being able to provide applications to the users.

Also keep in mind that it is recommended you have no users logged into an RDSH during the time of installation. When you are building fresh servers, this is easy, as you don't typically allow anyone to connect until everything is installed and configured. But if you need to install new programs or updates to existing programs onto a production RDSH, you will want to take steps to ensure that users are not logged in to the server before you place it into Install Mode and launch those executables. If you are running a farm of RDS servers and want to remove just one or some of them for maintenance or the installation of an application, make sure to check out the *Removing an RD Session Host server from use for maintenance* recipe.

I mentioned placing the RDSH into Install Mode even when just installing updates to existing applications. This is important. However, you do not need to place a server into Install Mode in order to install regular Windows operating system updates. These are able to install correctly even when the server is in normal Execute Mode.

# Disabling the redirection of local resources

One of the neat things about users connecting to virtual sessions within an RDS environment, especially when connecting in remotely, is local resource redirection. This feature enables the users to have access to things that are local to where they are sitting, from inside their virtual session, things such as the clipboard so that copy and paste functions will work between local computer and RDS session. Also things like drive redirection, so that you can save documents back and forth between the local hard drive and the RDS session. One of the most common uses of resource redirection is printers, so that users can print from inside their RDS session, which is sitting on a server in the corporate network, directly to a printer on the local network where they are connected. Short example: someone needing to print a document on a home printer.

This redirection technology can be very helpful, but is often not desirable from a security and policies standpoint. Many organizations have written security policy that dictates corporate data must remain within the corporate network and cannot move outside. Most often I see this in medical environments, where strict standards are in place to make sure data stays private and secure. This means that data cannot be copied and pasted to the local computer, documents cannot be saved outside of the RDS session, and printing of documents is also often not allowed.

While it may be saddening that you cannot use these functions if your security policy dictates it, thankfully disabling redirection is an easy thing to accomplish. Follow along to learn where these settings reside.

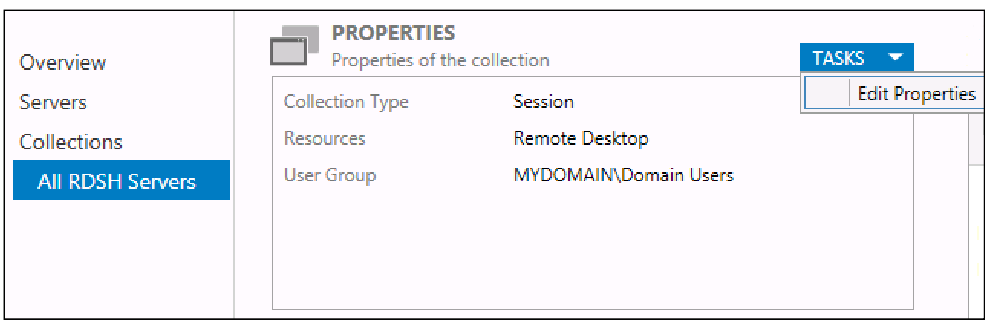
## Getting ready

We are logged into our Server 2012 R2 Remote Desktop Session Host (RDSH) server. This server is hosting some sensitive information and we want to make sure that users cannot save documents to their local computers, cannot print documents to local printers, and cannot copy/paste within the clipboard in order to move data from the RDS session to their local computers.

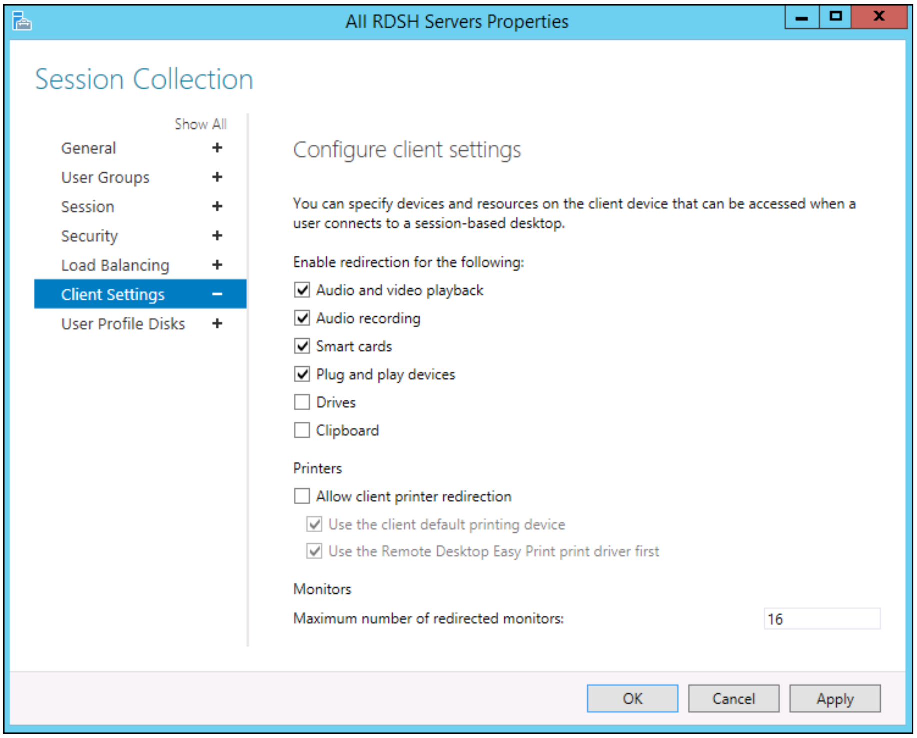
## How to do it...

Follow along to disable these redirection features on our RDSH collection:

1. Open up **Server Manager** and click on **Remote Desktop Services** to open up the management of your RDS environment.
2. We currently only have one RDSH collection listed, which contains both of our RDSH servers. This is the collection that all of our users connect to when they have to access this sensitive information. Click on the name of that collection. For our example, this one is called **All RDSH Servers**.
3. Near the top of the screen, look for the section called **Properties**. Click on the **TASKS** drop-down box and go to **Edit Properties**.




4. Click on **Client Settings**.
5. Here is your list of the items that are currently capable of being redirected. Go ahead and deselect each of the redirections that you want to disable. For our example, we are unchecking **Drives**, **Clipboard**, and **Allow client printer redirection**.




6. Click **OK** and those redirected resources are no longer available to client computers connecting to this RDSH collection.

## How it works...

Providing users with the capability of moving data back and forth between their local computers and RDS sessions sounds like a great feature, but is often less than desirable. With some simple checkboxes, we can wholesale disable these capabilities so that you can adhere to security policies and make sure sensitive data remains protected. Once you are familiar with the location of these settings, the enablement or disablement of them is intuitive and easy to accomplish. What is even better is that these settings can be changed at any time; it doesn't have to be a decision made while the RDS environment is being built. If you make the decision down the road to turn on or off some of these options, you can make these changes at any time to a production RDS.

# Shadowing another session in RDS

Let's say you receive a phone call from a remote user in your company; they are currently sitting in a hotel and are having trouble figuring out how to open an application. This application isn't installed on their local computer, they are an RDS user, and they connect into a virtual session on an RDSH server in your network whenever they need to access this app. You think about asking for their password, as that way you could just log into the RDSH as them and take care of the problem. But alas, asking for a password is a serious breach of company security policy. Instead, perhaps you can use some kind of online meeting software to share the screen of their laptop and try to walk them through fixing the problem. But that would mean walking them through the installation of that meeting software and hoping you could explain over the phone how to use it.

Looking for a better solution? Use the *Shadowing* feature of RDS. If you log in to the RDSH server that the user is already logged into, you can simply shadow their session in order to see what they are seeing. You can then work together to resolve the issue. You'll be able to take control and fix the problem, and maybe they can even take some notes and learn how to do it themselves next time to save the phone call.

This recipe is included here particularly because RDS Shadowing was always available in older versions of Terminal Server, but was then removed from Server 2012 RDS. Well, good news! It is back for Server 2012 R2. And I want to make sure that everyone knows how to take advantage of it.

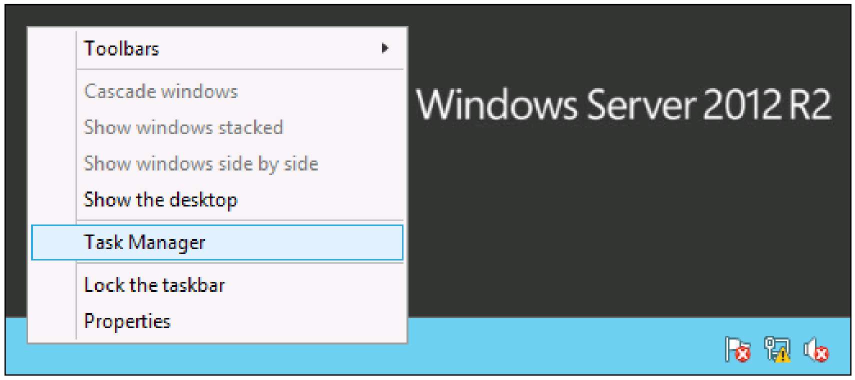
## Getting ready

Our remote user is logged into a virtual session on our RDSH server, which is called RDS-01. This is a Server 2012 R2 machine that is part of our RDS infrastructure.

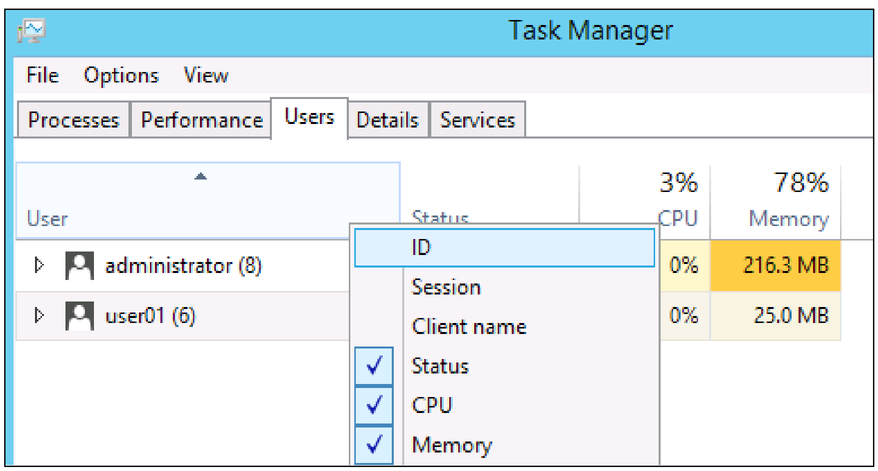
## How to do it...

Let's help out this remote user by shadowing their RD session:

1. First we need to log into the same RDSH server that the user is logged into. On your computer, open **Remote Desktop Connection** and input the server name in order to connect.
2. Now that you are logged into the RDSH, right-click on the Taskbar and open **Task Manager**.

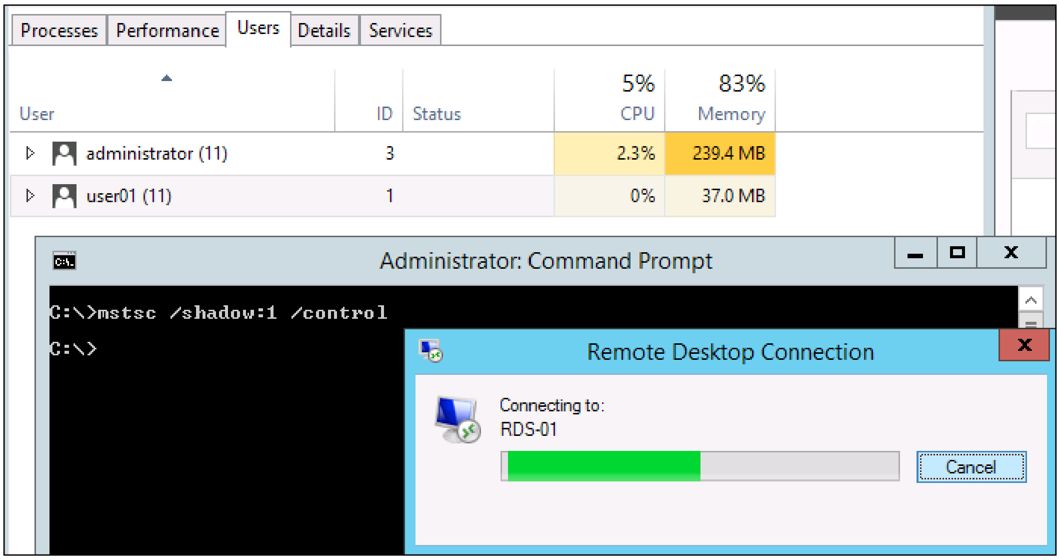



3. Click on **More details** in order to see more information about the server.
4. Navigate to the **Users** tab.
5. Right-click on one of the column headings and choose to show the **ID** column.




6. Leave Task Manager open so that you can see the username that you want to connect to and their ID number.

7. Now open a command prompt, and type the following: `mstsc /shadow:<id> /control`. So for our particular user01 login who is currently running on ID 1, we use this command: `mstsc /shadow:1 /control`.




8. This command will launch a shadowing session to the RD session of the ID number that you used, so make sure to use the correct ID for the user you want to shadow. Since we used the `/control` switch, you should also have the capability of using your own mouse and keyboard inside the user's session.

## How it works...

While shadowing in Server 2012 R2 isn't quite as easy as it used to be in earlier versions of Terminal Server, it's great to know that this capability has returned in Server 2012 after a noticeable absence. RDS Shadowing is a great tool to use for troubleshooting or collaboration, as it enables you to share the screen of other personnel and assist with your own keyboard and mouse control when necessary. Having two sets of eyes on the same RD session can be invaluable in many situations. Go try it out today!

# Installing a printer driver to use with redirection

When a user connects to an RD session, if the client and server are configured properly, that connection will attempt to set up printer redirection between the RD session and the local computer. Specifically what happens is that every printer that is installed onto the local computer will be configured as a separate printer inside the user's RD session. This is the feature that enables users to be able to print to their local printers, even if the information that they are accessing and printing is halfway around the world.

When the RD connection builds these virtual printers, it attempts to use real printer drivers for them. For example, if the printer is an HP LaserJet 4100 and the RDSH server has the HP LaserJet 4100 driver installed, then when that printer gets set up inside the user session, it will utilize that existing, official driver. If the user logs in to an RDSH with a printer whose driver does not exist on the RDSH server however, by default that printer will not be installed. There is a setting in the same configuration page where we enable or disable printer redirection on the RDSH server collection that can partially help with this. If you select the option on that screen for **Use the Remote Desktop Easy Print print driver first**, when the real driver doesn't exist for a particular printer, it will use a generic driver that may or may not actually work with the printer. This can certainly help bridge the gap when it comes to missing printer drivers, but doesn't always solve the problem.

The best way to make sure your users are going to be able to print properly is to install the real driver onto the RDSH. So what's the point of this recipe, who doesn't know how to install a printer driver, right? I write this because most printer driver software packages are now full blown applications, and we don't need a quarter of what comes with them. Driver install packages consume much more space than necessary for use with RDS, and we would have to take into consideration that we are installing actual applications that could potentially show up inside user sessions and cause confusion. So what is the answer? Extract the simple driver files from those driver packages and use just the files themselves in order to install the driver into Windows. Let's do one together so you can see what I'm talking about.

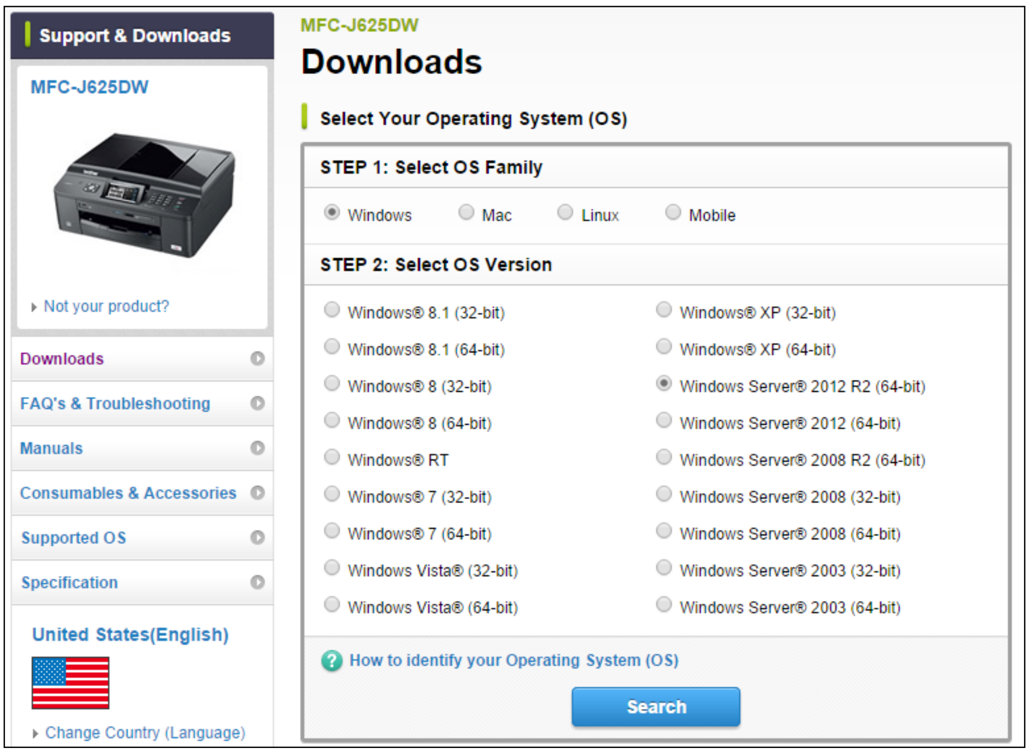
## Getting ready

We will be installing this printer driver onto our RDSH server that is running Windows Server 2012 R2. For our example, we will be using a Brother MFC-J625DW printer, since that is one I installed for a customer just recently. Brother is usually good about providing a simple, small driver download that contains only the files we need for the driver itself.

# How to do it...

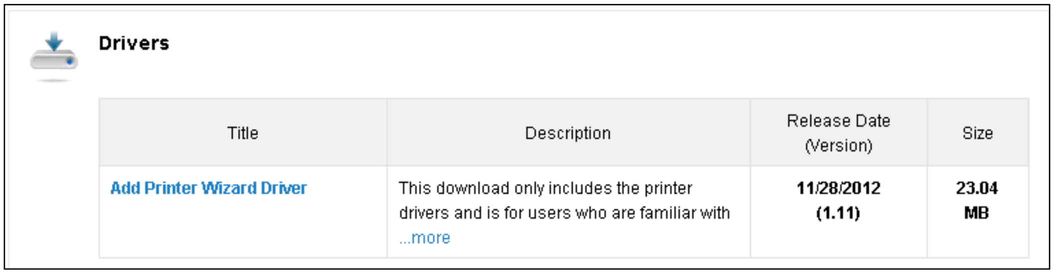
Let's work together to download and install this printer driver onto our RDSH, so that it can be used for printer redirection:

1. First, download the driver files onto your RDSH server. Make sure to choose the driver for the server's operating system, not the client. So I am choosing Windows Server 2012 R2.



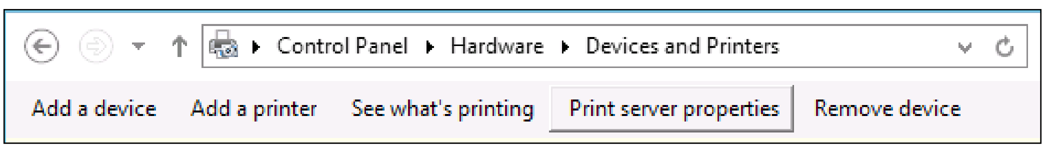

[ If you are unable to locate the driver for Server 2012 R2, try using the one built for Windows 8 or 8.1. Often these drivers will work. ]

2. We can see that this driver is nice and small, only 23 MB. Often the small packages will range between 10-30 MB, and the large installation packages that contain the applications we do not need are more like 100-200 MB. Go ahead and click on **Add Printer Wizard Driver**.



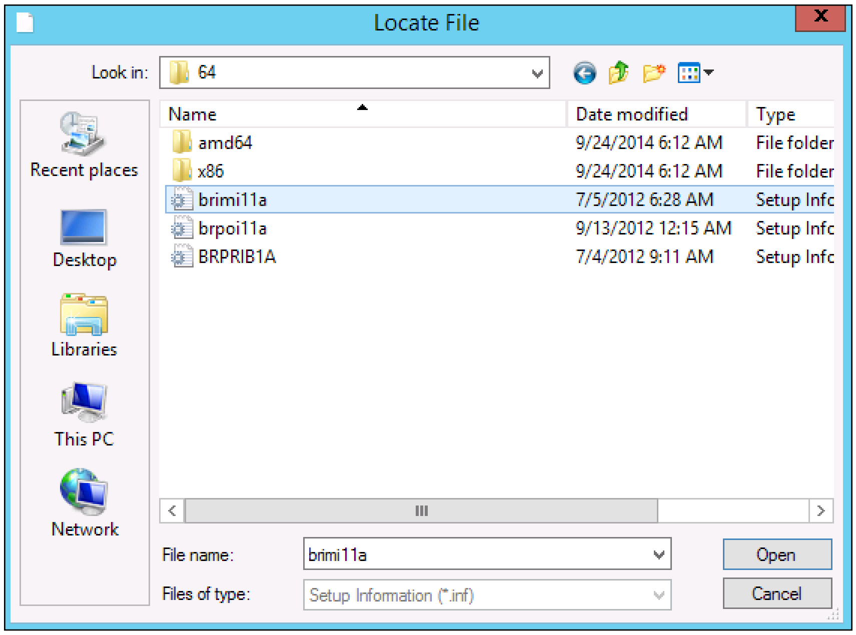

[ With most driver downloads, you will also have to double-click on it once downloaded in order to extract the files. ]

3. Right-click on your **Start** flag and choose **Control Panel**.
4. Click on **Hardware** | **View devices and printers**.
5. Click on any existing printer in the list and then click on the button in the top Taskbar that says **Print server properties**.

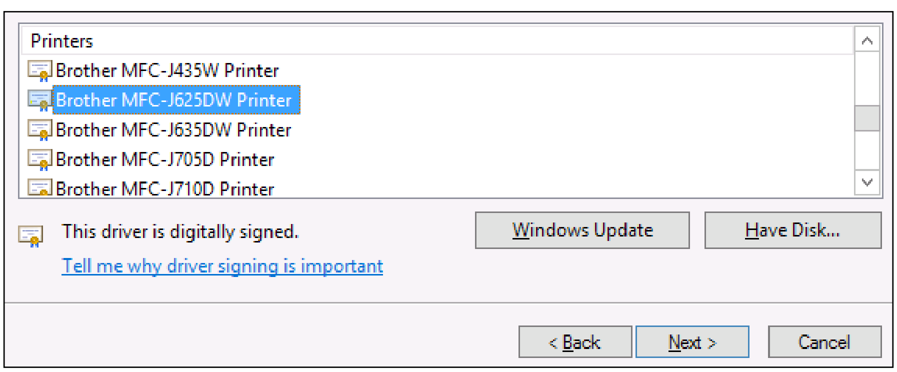



6. Browse to the **Drivers** tab. This displays a list of currently installed printer drivers on this server. Then click the **Add...** button.
7. Click **Next** twice. We can leave the **Processor Selection** screen marked as only **x64**, since Windows Server 2012 R2 only comes in 64-bit.

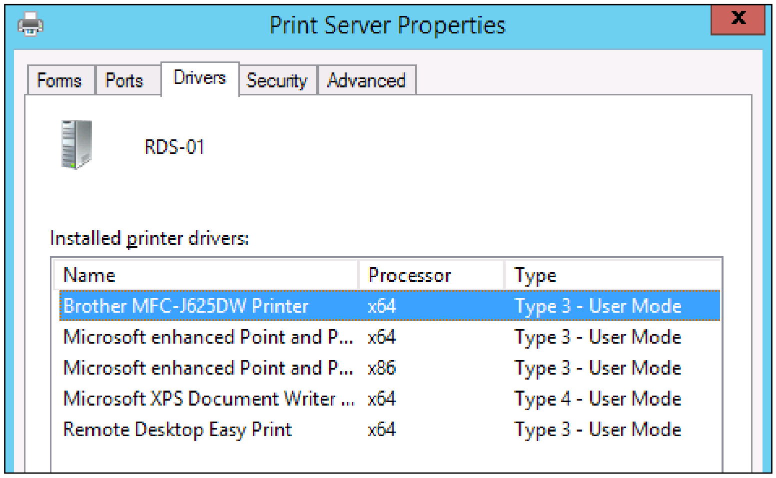
8. Now click on **Have Disk...** and browse to the location of the driver files that you downloaded. You are looking for an `INF` file that typically sits in the root of that driver folder. Sometimes you will have to poke around a little until you find it, but the file is always an `INF` file.




9. Once you have selected the `INF` file, the **Add Printer Driver Wizard** will now display a list of the drivers that are contained within that `INF` file. Choose the specific printer driver that you want to install and click **Next**.



10. Click **Finish** and the driver will install. You should now see it in the list of printer drivers that are installed on this server.




## How it works...

Installing printer drivers onto RDSH servers is a pretty common administrative task in environments where printer redirection is allowed. We walked through one of the nice, simple installers that was easy to extract and contained only the actual driver files that we needed. These kinds of driver downloads are perfect for our purposes here.

As you experience more and more of these driver installations, you will start to learn which manufacturers provide simple driver packages for this purpose and which ones do not. Ultimately though, the software always contains the simple driver files; sometimes it's just a matter of launching the huge installer program so that it places the files somewhere in a temporary location on the hard drive. What I normally do in these situations is launch the installer and walk it through whatever steps are necessary in order to see that it is unpacking/extracting files. Once it has done that, you don't have to run any more of the wizard to install the software applications, because you know that the driver files you need are sitting on the hard drive of the server somewhere. We just need to find them. Using a utility such as FileMon can help identify file locations that have been recently modified, and is a pretty quick way to track down those driver files that are usually hidden away in a `temp` folder. Once you find the files, you can copy and paste them into a more permanent folder for driver installation purposes, cancel out of the install wizard, and walk through the steps in this recipe to install that driver manually instead.

# Removing an RD Session Host server from use for maintenance

Occasionally, you will have to perform some maintenance on your RDSH servers. Whether it is for installing updates, installing new applications, or taking them down for some physical maintenance, it will happen sooner or later. If you have multiple RDSH servers in a collection and simply take one offline, the user loads will eventually sort themselves out as the RD broker will send new connections to the RDSH servers that are still online, but you will have caused frustration and headaches for any users who were logged in when you shut it down. It is much more user friendly to flag an RDSH to make it unable to accept new user connections and let the existing ones dissolve naturally over a period of time. This is kind of like a "drain stop" in the NLB world.

Let's take a look at the setting included in RDS that allows us to flag an RDSH as unusable and force the broker to keep new connections from coming through to it. We'll also reverse that change to make sure it starts accepting user connections again after our maintenance is complete.

## Getting ready

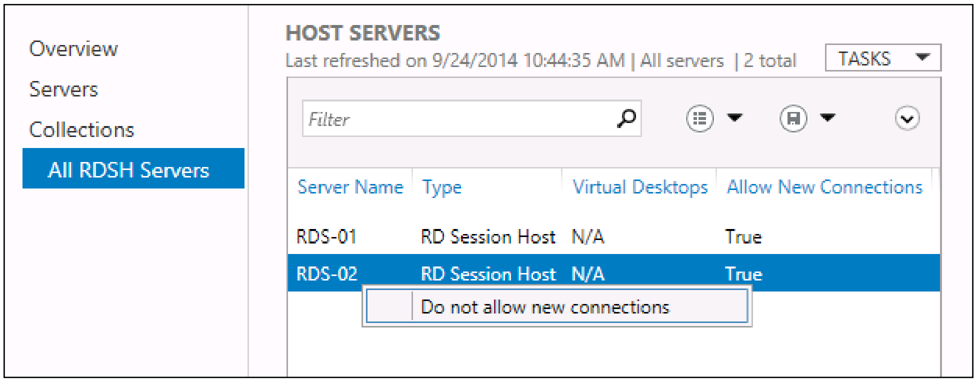
We have an RDS environment configured with two RDSH servers. These are called RDS-01 and RDS-02, and we are required to do some maintenance on RDS-02. All of our work will be accomplished from inside the Remote Desktop management console on RDS-01.

## How to do it...

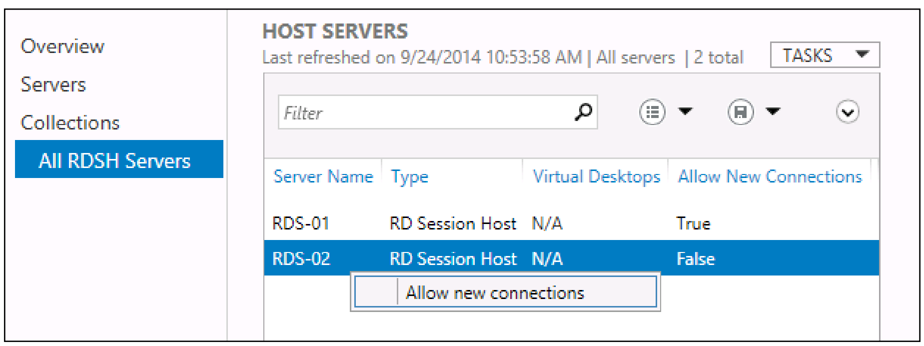
To stop new user connections from flowing to RDS-02:

1. Open **Server Manager** and click on **Remote Desktop Services** from the left window pane.
2. Navigate to **Collections** | **All RDSH Servers**. This is the name of the collection in my environment; you will need to click on whatever the name of your collection is.
3. Scroll down to the bottom, where you can see the **Host Servers** section. This is a list of the RDSH servers that are part of your collection.
4. Right-click on the RDSH server that we need to perform some maintenance on. In our case, it is **RDS-02**.

5. Click on **Do not allow new connections**.




6. This will cause any new connections to be sent over to RDS-01 or whatever other RDSH servers you have in your collection. Then once your maintenance is complete and you are ready to reintroduce RDS-02 back into the collection, simply right-click on its name here again and this time choose **Allow new connections**.




## How it works...

This simple option can be a very helpful utility when considering maintenance within your RDS infrastructure. Remember, disallowing new connections to a particular RDSH does not mean that it is immediately available for maintenance, because existing users will still be logged in to it. We have only set it so that no new connections will flow there. You can give it some planned time to naturally drop the remaining connections that do exist on the server before performing your maintenance.

# Publishing WordPad with RemoteApp

Most of the recipes in this chapter are focused around full desktop sessions provided by RDSH servers, because this is the most common scenario that I find RDS used for in the field. One additional piece of RDS, I would like to take a quick look into is RemoteApp publishing. This is the ability to publish individual applications out to the remote users from an RDSH server, rather than a full desktop session. It provides a seamless window for the application, allowing the RemoteApp to look and feel like any other program on the user's computer. Let's set up a sample application and test using it from a client computer. For the sake of simplicity in demonstrating this capability, we will use WordPad as our application to publish and launch.

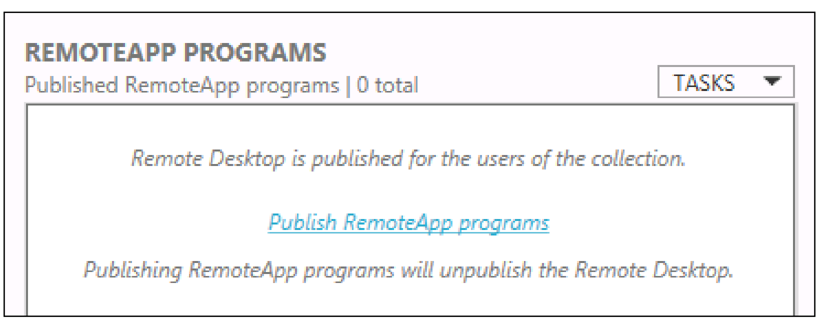
## Getting ready

Our work to publish WordPad as a RemoteApp will be performed from our Server 2012 R2 RDSH called RDS-01. We will also use a client computer in order to test accessing this application once we are finished publishing it.

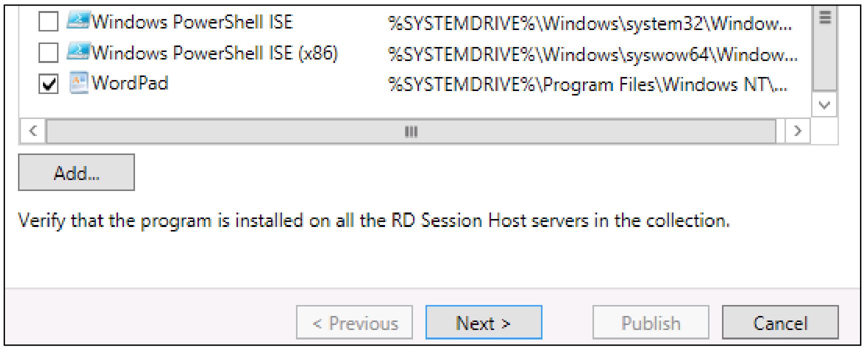
## How to do it...

To publish WordPad as a RemoteApp:

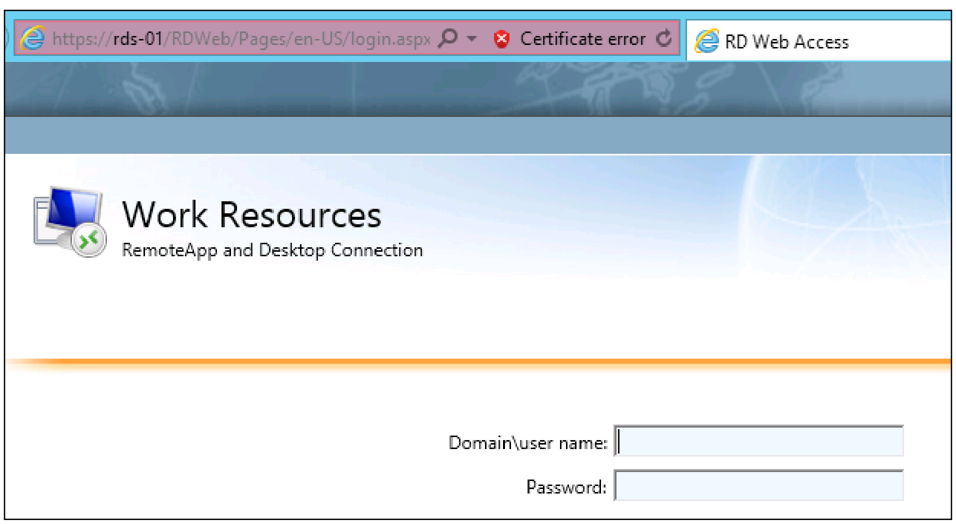
1. On RDS-01, launch **Server Manager** and click on **Remote Desktop Services** from the left window pane.
2. Browse to the collection of RDSH servers where you want to publish this new application. For our example, I am browsing to **Collections** | **All RDSH Servers**.
3. Near the middle of this window, you will see a section called **REMOTEAPP PROGRAMS**. Click on the link in the middle of this window that says **Publish RemoteApp programs**.



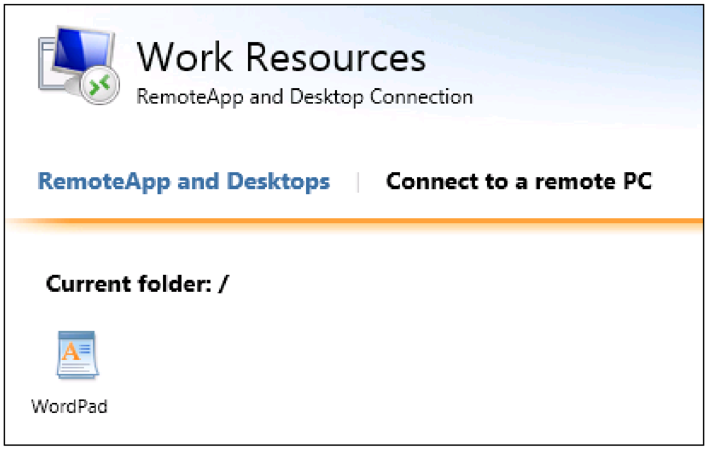
4. The wizard will now poll the server for a list of available applications. Look through the list until you see **WordPad**; it is most likely on the bottom. Choose it and click **Next**.



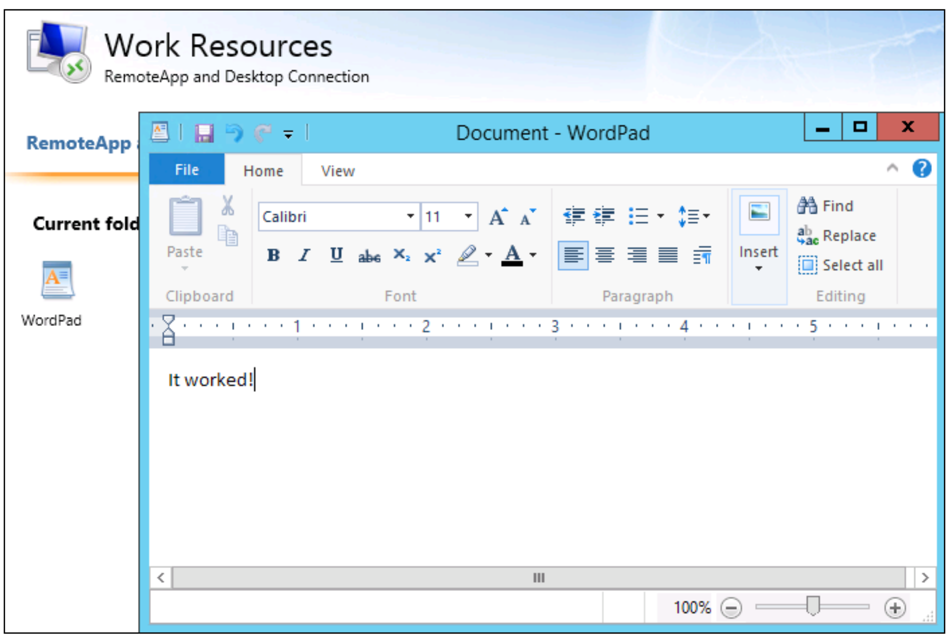

5. On the **Confirmation** screen, click **Publish**.
6. Now that we have the WordPad application published, log in to a client computer so that we can test accessing it.
7. On the client computer, open up a web browser and navigate to `http://RDS-01/RDweb`.



8. Input your credentials and you should see our published resources that are available in the RDS environment. As expected, WordPad is now visible here.




9. Click on the **WordPad** icon and it opens on your computer.



## How it works…

If you do not have the need for users to receive access to a full desktop when they log in to the RDS environment, you have the option of publishing individual applications instead. This can be useful for restricting the resources that employees have access to, or perhaps someone like a vendor or a temporary assignment that only needs access to certain programs and data. While this was a very simple demonstration using the WordPad program baked into Windows, you can take this same process with other applications that you have installed onto your RDSH servers yourself.

> Make sure you install the applications onto all of the RDSH servers in your collection.

# 8

# Monitoring and Backup

Monitoring and backing up servers are usually mundane tasks that are easy to be overlooked or forgotten. When everything is running smoothly, you may not even think about whether or not your servers have backed up properly, maybe for weeks at a time. Except in the largest of companies, there usually aren't dedicated "Backup Admins" or "Performance Monitoring Gurus." In IT, we all wear many different hats, and they don't always fit on top of each other.

The key phrase above is "when everything is running smoothly." Unfortunately, this state of bliss cannot continue indefinitely. Hardware fails, malware happens, files are accidentally deleted. Suddenly those dull chores of due diligence, such as monitoring the health of your servers and making sure you have solid backups, jumps from backburner to mission critical on the importance scale.

The good news is that monitoring and backups have never been easier than they are in Windows Server 2012 R2. Let's explore together some of the tools that exist to make these areas of your infrastructure efficient and automatic:

- Using Server Manager as a quick monitoring tool
- Using the new Task Manager to its full potential
- Evaluating system performance with Windows Performance Monitor
- Using Format-List to modify PowerShell data output
- Configuring a full system backup using Windows Server Backup
- Recovering data from a Windows backup file
- Using IP Address Management (IPAM) to keep track of your used IP addresses

# Introduction

There are many third-party tools available for performing functions like data backups and performance monitoring, and because these tools exist it is easy to automatically assume that they will do a better job than anything that comes in the operating system. Given that, we often categorize backups and monitoring into areas where we will have to spend extra money. I'm not trying to argue that every add-on tool for these functions is unnecessary, because they do certainly give benefit to the right kinds of companies. But for anyone willing to dig into Server 2012 R2 and discover what it can accomplish on its own accord, without extra add-ons, I think you will find that it meets the needs of many businesses.

# Using Server Manager as a quick monitoring tool

Sometimes change is difficult for us old school IT guys. You know, the ones who prefer keyboards over mice and command lines over graphical interfaces. In Server 2012 and 2012 R2, Server Manager has changed, a lot. I find that many admins automatically dislike it, even before they have started using it. It looks "cloudy," full of links to click on rather than applications. Certainly more of a web app interface than the Server Manager we are used to.

Let's use this recipe to point out some of the important data that exists in Server Manager, and discover for ourselves that Microsoft may actually have a valid point in causing it to open automatically every time that you log in to a server. No, it's not just there to annoy you.

## Getting ready

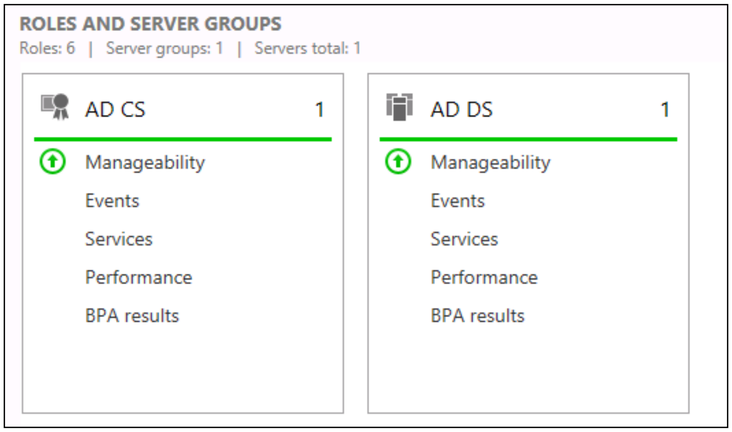
All we need is a Windows Server 2012 R2 in order to poke around in Server Manager. The server we are using is domain joined with a few roles installed so that we can get a better feel for the layout of data on a production system.

## How to do it...

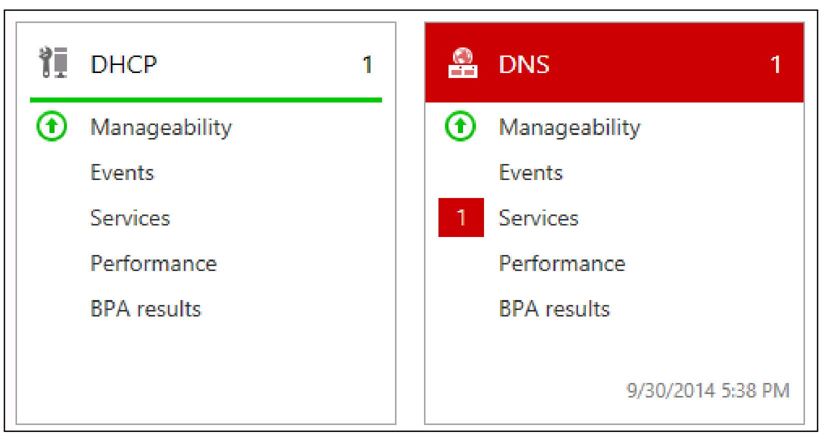
Follow these steps to discover some of the functions that Server Manager can perform:

1. Open up **Server Manager**. If you just logged into your server, it is probably opening automatically. Otherwise, click on the **Server Manager** button from your Taskbar, right next to the **Start** flag.
2. Normally at the top of Server Manager is the section entitled **Welcome to Server Manager**. In the lower right corner of that section there is a button that says **Hide**. Go ahead and click on that button to hide this section of the screen.

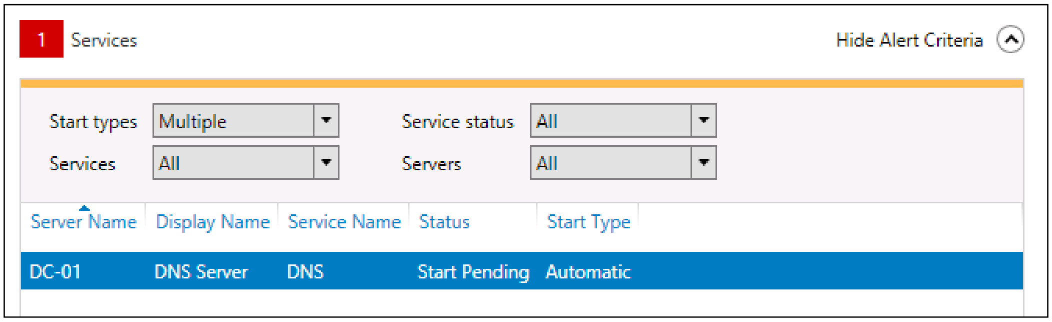
3. Now take a look at the information on your screen. These normally green bars listed under each service that you have installed are your first indication as to whether or not everything is running smoothly. Everything is green on mine, which indicates that everything is working properly.



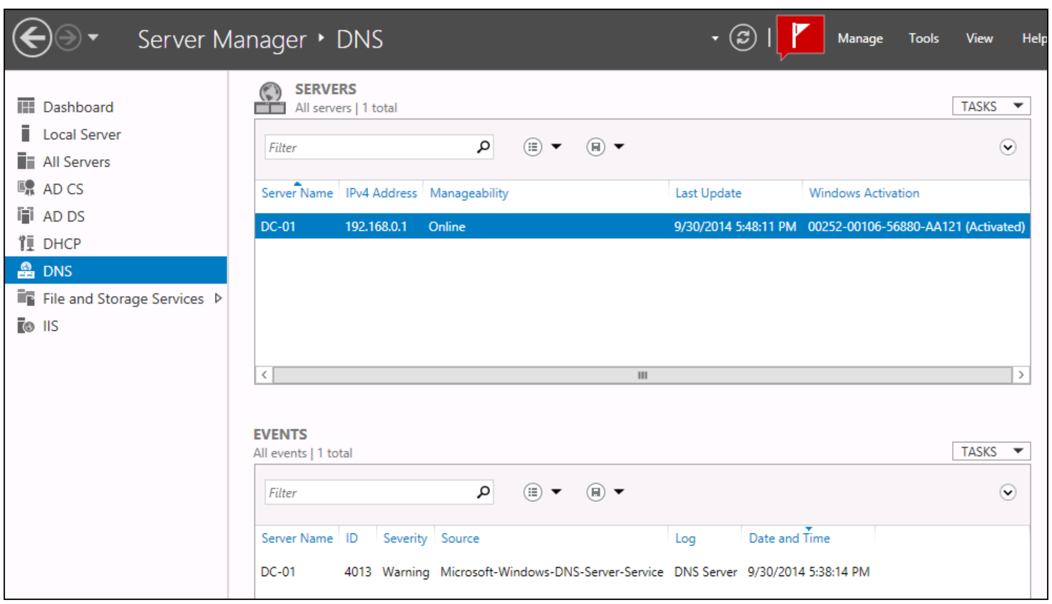

4. Now I'm going to break my DNS service purposefully to demonstrate what it looks like when things aren't running smoothly. You may or may not want to do this depending on whether or not you are looking at this on a production server. I have stopped my main service for DNS and now see the following in Server Manager.



5. If I click on the **Services** button, where it is indicating that I have one notification, I can see the details of what is going on.

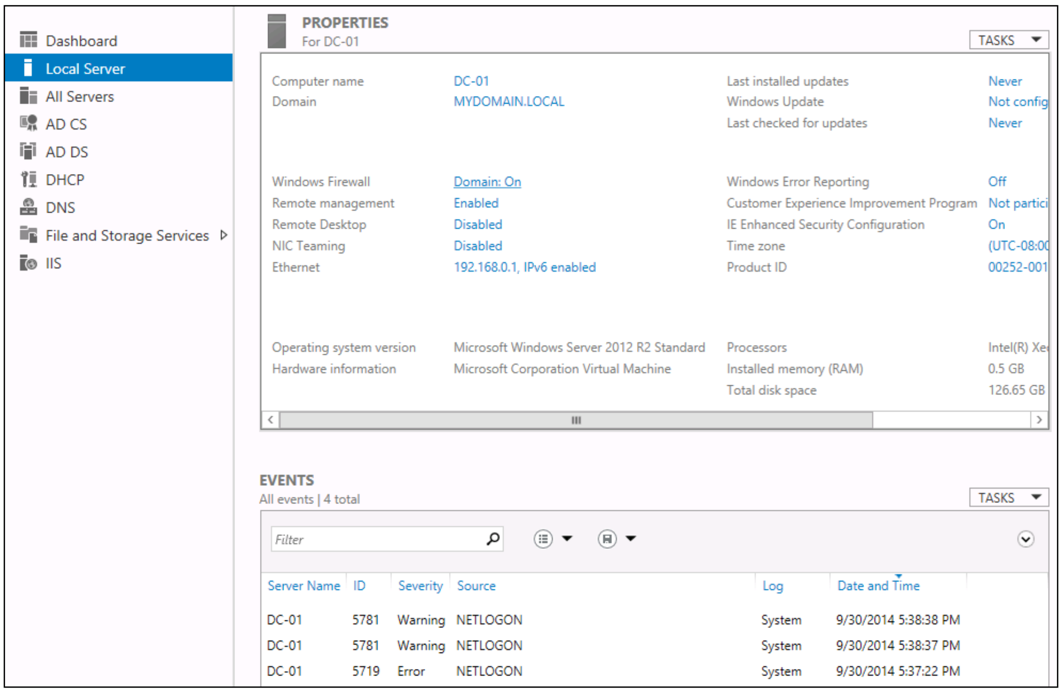



6. There is a button near the bottom of this screen that says **Go To DNS**. Go ahead and click on that button and you will see that it brings us to the same screen as if we had clicked on **DNS** from the left window pane in Server Manager. On this screen, we can see even more information about the trouble that our DNS role is having.

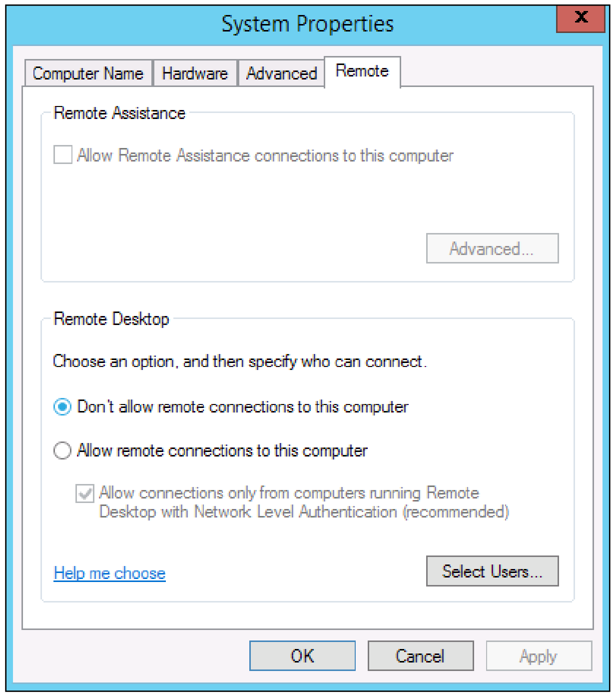


For any role that you have installed on your server, there is a quick link to that role's section of Server Manager in the left window pane. Click on each role to view events and information that is specific to that role.

7. Now click on **Local Server** from the left window pane. Here we see a number of items listed that are helpful for troubleshooting any facet of the operating system, and for reviewing general status and health of the server. Scroll down near the bottom of this page for a list of events that are happening on this server, without having to open a separate Event Viewer window.



8. Many of the items listed inside this **Local Server** screen are links to open additional configuration windows. For example, where it tells us that **Remote Desktop** is currently disabled, if we click on that word **Disabled**, we get the properties page for configuring Remote Desktop settings on this server.




## How it works...

Server Manager is full of opportunities to quickly find information that will help monitor your servers. This recipe is just a sampling of the data that you can pull into Server Manager, so I suggest you continue navigating around in there to make it look and feel the best that it can for your environment. Another extremely helpful option here is to add multiple servers into your Server Manager for monitoring purposes. If you use the **Manage** menu near the top and the **Add Servers** function in that menu, you can add additional systems into your Server Manager window pane. Doing this causes Server Manager to pull information not only about the local server that you are logged in to, but also about these remote servers, all into one pane of glass. This way you can use Server Manager on one server in order to monitor and maintain your entire server infrastructure, if you choose to do so.

# Using the new Task Manager to its full potential

We have all used *Ctrl* + *Alt* + *Delete* to open Task Manager and attempt to close problematic applications. With Task Manager provided by Windows Server 2012 R2, we can do much more right from that same interface. Let's work through this recipe to explore some of the new things that can be done to take full advantage of this tool.

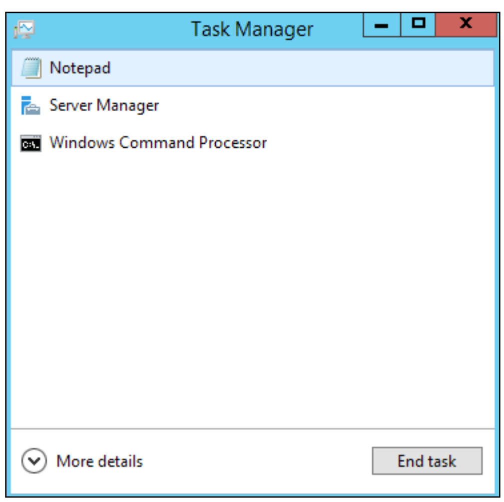
## Getting ready

We are logged into a Windows Server 2012 R2 server. This is the only system required for our recipe.

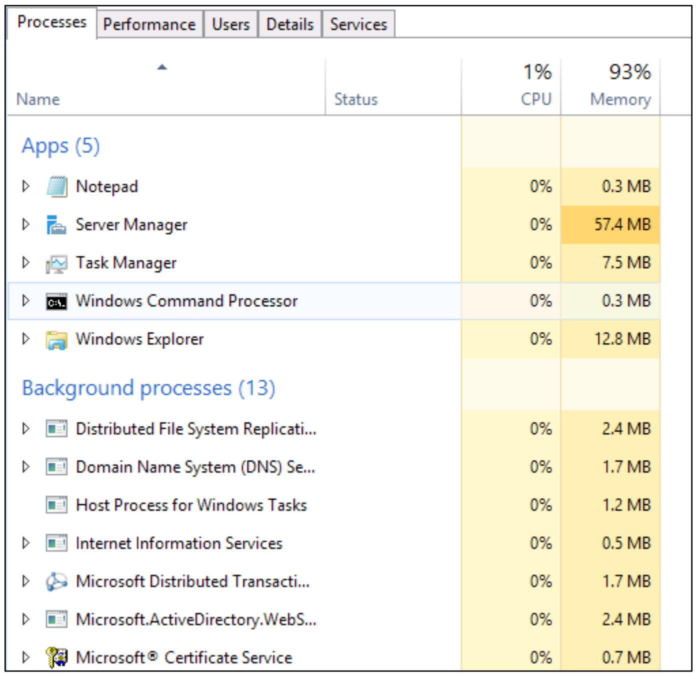
## How to do it...

Follow these steps to learn a little more about Task Manager:

1. Right-click on the Taskbar, and choose to open **Task Manager**. This is an alternate way to get into the utility, other than using the *Ctrl* + *Alt* + *Delete* key combination. I prefer using the Taskbar right-click in fact, because when using the keyboard it is easy to open the wrong Task Manager when you are using a virtualization console or RDP to administer remote servers.
2. You are now looking at the simple version of Task Manager, where you can choose an application and click **End task** in order to forcibly close that application. To dig a little deeper, click on the **More details** link near the bottom.

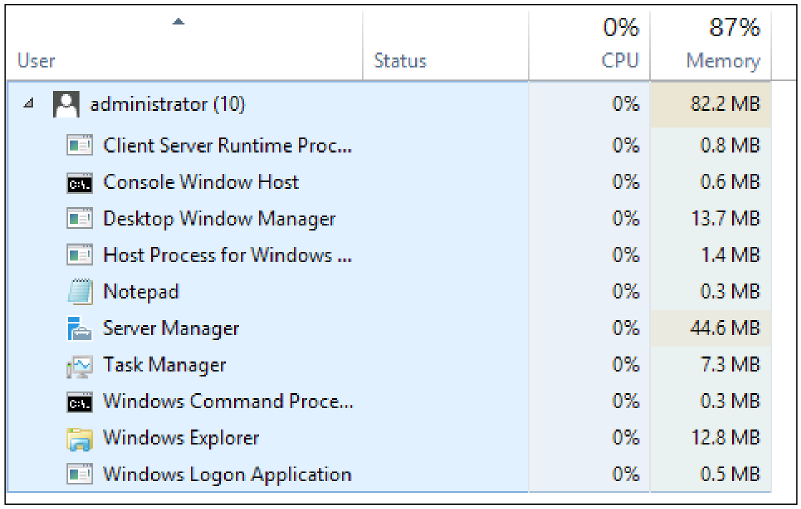


Now this is more like it! We can see all of the open applications at a quick glance, including how many resources each one is consuming. This makes it pretty easy to identify applications that might be stuck and consuming large amounts of CPU or memory. It also lists out **Background processes** separately, which can be hugely helpful for finding malware or rogue processes.

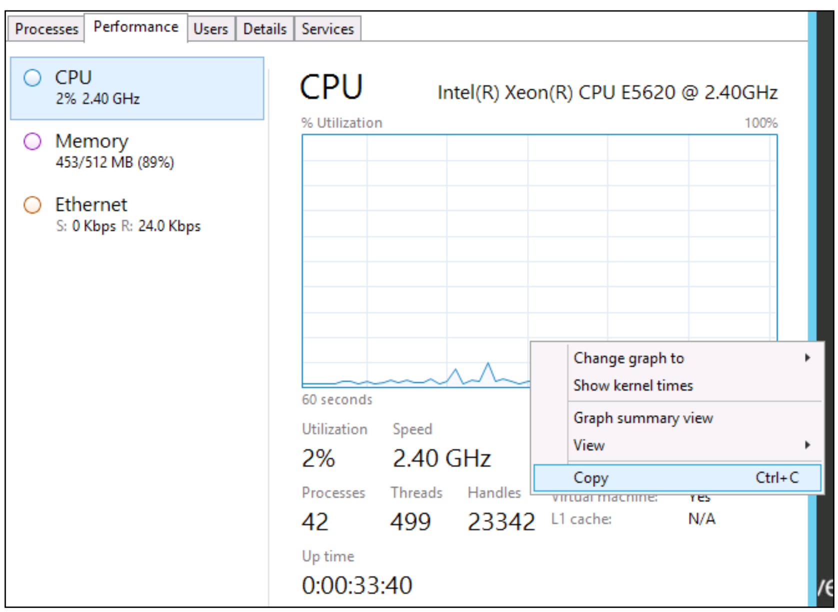



The **Details** and **Services** tabs are pretty self-explanatory. **Details** will show even more information about the individual processes that are running and consuming resources on your server. The **Services** tab shows the list of services installed on your server and their current statuses.

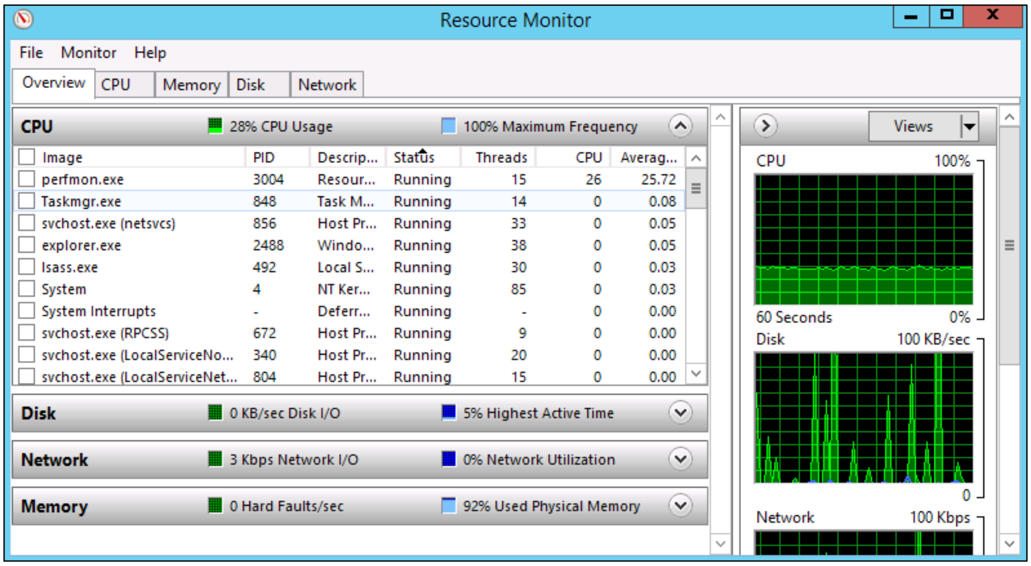
3. Click on the **Users** tab and then click the arrow listed under your username to see the expanded view. Listed under each username are the applications that they have open. This sorting of running programs is especially nice when logged into a server hosting many user connections at once, like a Remote Desktop Session Host.




4. Now browse over to the **Performance** tab. You will find that this screen looks much nicer visually than previous versions. You can click between the different performance counters on the left to see the different details. If you right-click on the graph itself, you will notice there are some additional options. You can click on **Graph summary view** in order to change the Task Manager window into a smaller, graph-only mode that you can leave running in the corner of the screen. You can also choose to copy the screen, which can be helpful for grabbing a quick copy of this data and sending it on for troubleshooting or monitoring purposes.



5. At the bottom of your Task Manager screen, click on **Open Resource Monitor**. This runs the new Resource Monitor, which is an even more extensive tool for monitoring hardware resources and utilization. This is very helpful for monitoring hardware in real time.




## How it works...

The new Task Manager provided with Server 2012 R2 contains many additional pieces of information that are helpful for monitoring system performance in real time. As you start to administer your new Server 2012 R2 machines, make sure to spend some time in this interface so that you are familiar with the new layout of information when the time comes that you will need to access it quickly.

# Evaluating system performance with Windows Performance Monitor

While good old Task Manager and the new Resource Monitor are great utilities for monitoring system performance in real time, for any more extensive monitoring needs I tend to prefer Performance Monitor. Perfmon, as it is often nicknamed, is an excellent tool that can be used for collecting specific data over a predefined period of time.

We have all had cases where a report comes across our desk that a certain server is misbehaving or running slowly. By the time we get logged in, everything looks normal. Other than Event Viewer, we don't have a whole lot of options for investigation into what was happening during the time of the problem. But it might happen again, and if we plan ahead with the Performance Monitor tool, we might be able to catch the server in the act, even if we don't see the data until after the event has finished.

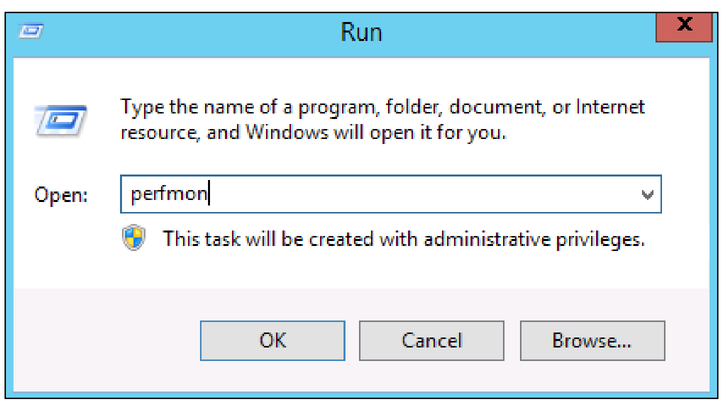
## Getting ready

We will be monitoring a Windows Server 2012 R2 server in our environment for this recipe. Nothing needs to be installed, as Performance Monitor is part of Windows by default.

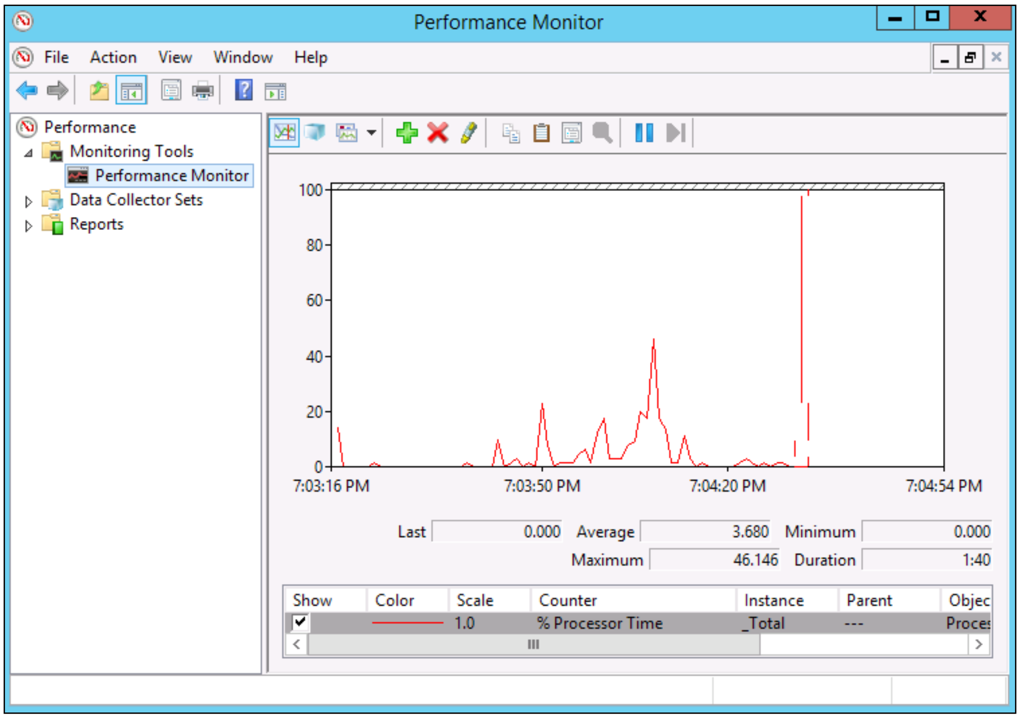
## How to do it...

In order to collect server performance data using Performance Monitor:

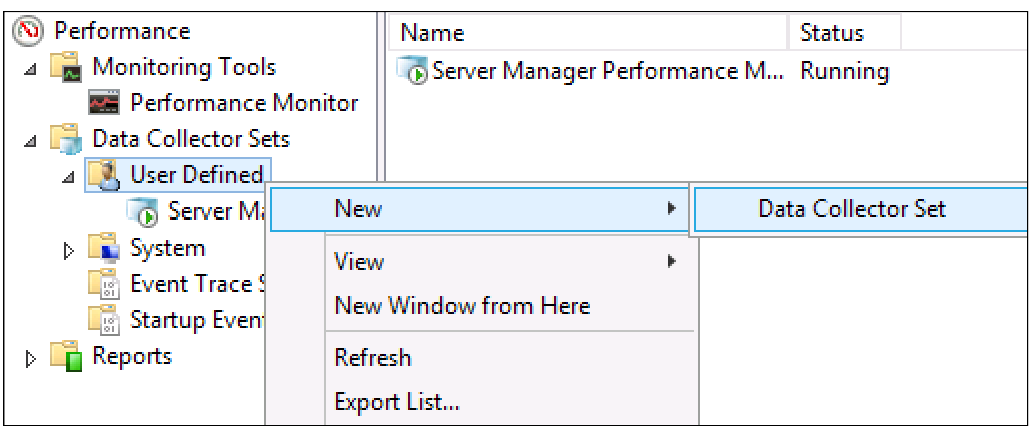
1. Open up a command prompt or your **Run** box and type `perfmon`. This will launch the Performance Monitor tool.



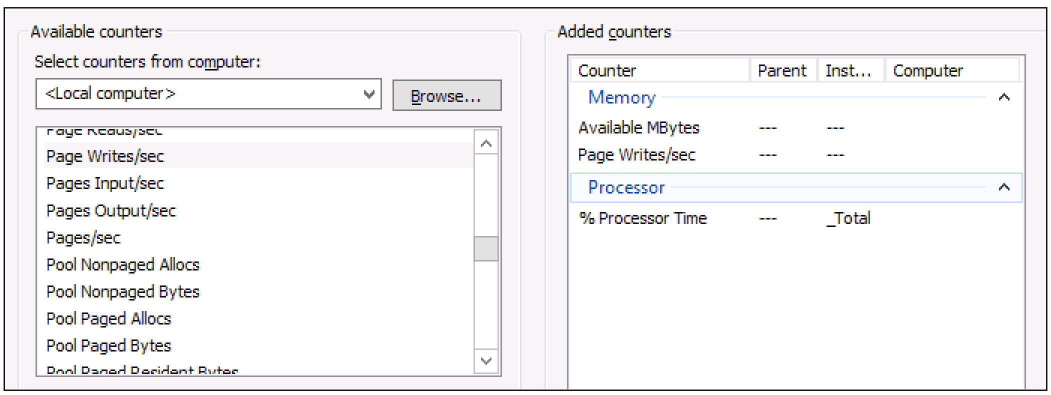
2. From the left window pane, choose **Monitoring Tools** | **Performance Monitor**. You can see that it is showing some real-time data about the processor by default.




3. Browse to **Data Collector Sets** | **User Defined**. Right-click on this folder and choose **New** | **Data Collector Set**.

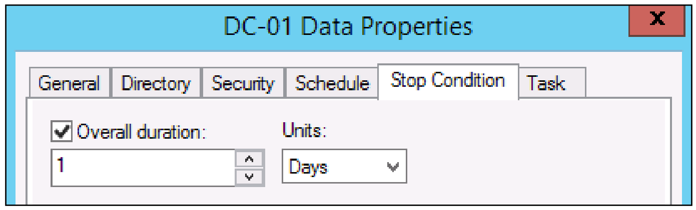


4. Name your new Data Collector Set and choose the bottom radio button that says **Create manually (Advanced)**. Then click **Next**.
5. Check the box for **Performance counter** and click **Next**.
6. Click the **Add...** button in order to add in some performance counters that we want to keep track of on this server.
7. For my evaluation on this server, I am going to add the following counters:
   - **Processor** | **% Processor Time**: This will tell us how busy the CPU is.
   - **Memory | Available MBytes**: This will tell us how much RAM is available.
   - **Memory** | **Page Writes/sec**: This will tell us how often Windows is looking to the paging file in order to create virtual memory, which helps to indicate whether or not the system is running out of physical memory.
8. As you can see, there are so many different counters that you can add. Today we are only interested in these three, and so we can click on the **OK** button.

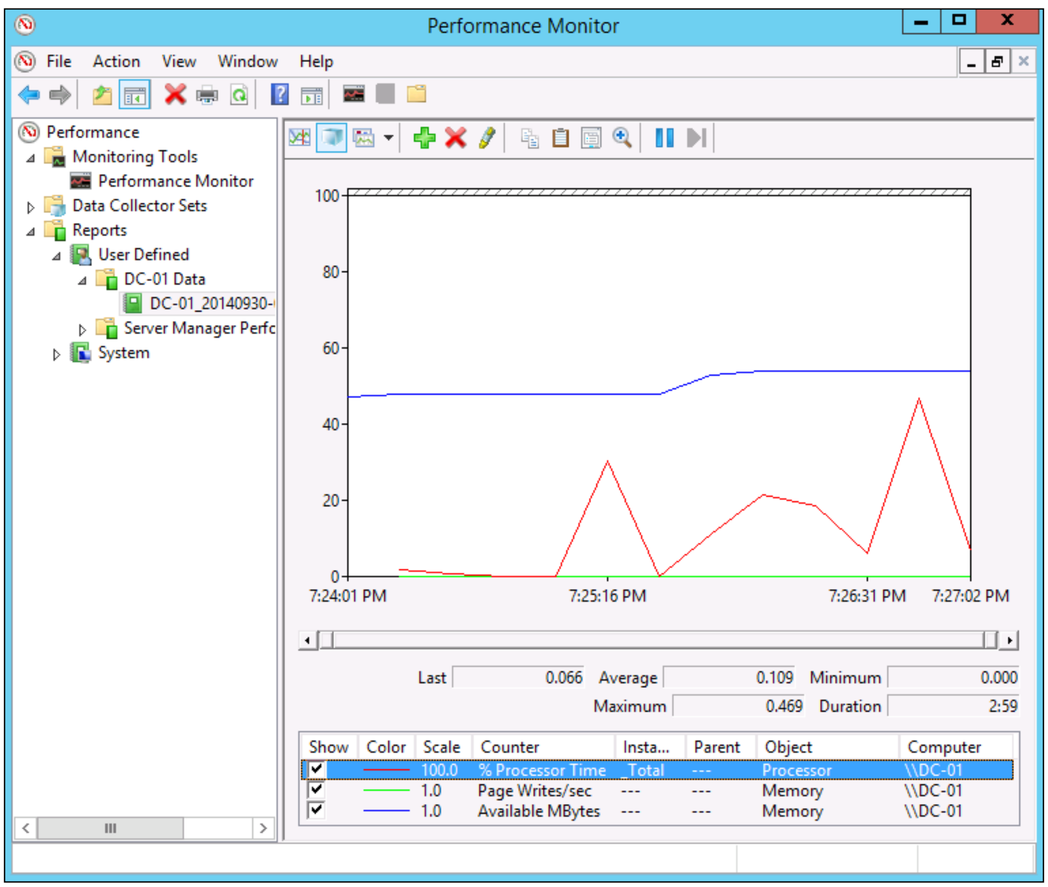



9. Back in our wizard for setting up the new Data Collector set, you should see our three counters now listed. Go ahead and click **Next**.
10. Change where you would like the data saved, if necessary. Then click **Next**.
11. On the last screen of the wizard, choose the radio button for **Open properties for this data collector set**. Then click the **Finish** button.
12. Navigate over to the **Schedule** tab and click the **Add** button to set your preferred time in **Start time** field for these performance counters to be collected.

13. Once you have set a start time, you can either plan to stop the data collection manually, or you can use the **Stop Condition** tab in order to stop the collection after a predetermined amount of time. Using a combination of the **Schedule** and **Stop Condition** tabs is a great way to collect data for a specific time range, like one day.




14. Now we have some data that has been collected, head down to **Reports** | **User Defined** in order to see the data that was stored during the time period that we specified.



## How it works...

Performance Monitor is a great tool for collecting hardware and server performance data. The ability to be very granular in identifying which resources you want to monitor is extremely helpful. Combine that with the scheduling capabilities for the collection times and you have a recipe for successful server monitoring. It can also be useful to run a Perfmon data set as a baseline after installing a new server. This way you can hold onto that report and compare it against later similar reports once user load increases, to look back and find out what kind of an impact certain services or users have on a system.

# Using Format-List to modify PowerShell data output

There is a special parameter that can be used with just about any PowerShell command or cmdlet in order to display different, and usually more, data from that particular command. This parameter is called **Format-List**, and if you are a fan of finding as much information as possible about the tools you are working with, this is something you will certainly want to become familiar with. PowerShell is often used for monitoring many different facets of Windows Server, and getting to know the intricacies of Format-List will certain help you to sculpt the output information that you are looking for when performing monitoring functions from the PowerShell command line.

We all know that a `dir` command will display the list of files and folders that are within our current directory; this works in either Command Prompt or in PowerShell. Let's start learning how to make use of Format-List by using it to modify the output of our `dir` information.

## Getting ready

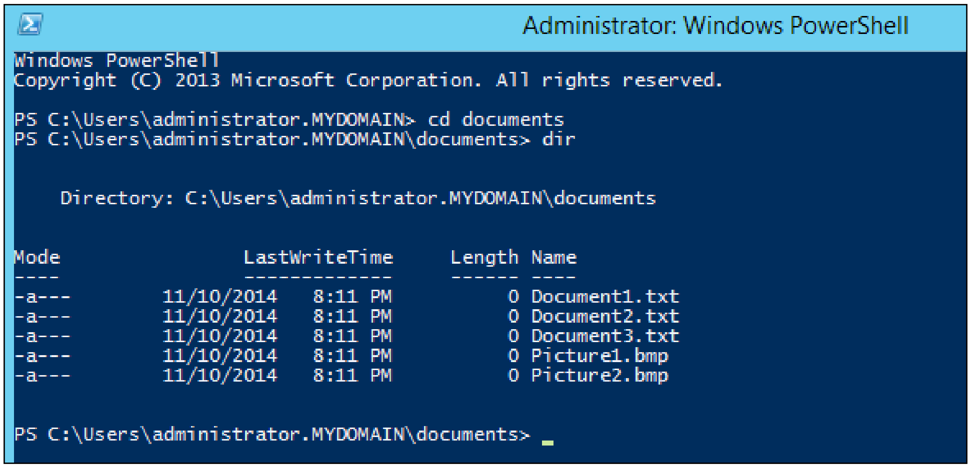
We will be running these commands from a PowerShell prompt on one of our Windows Server 2012 R2 machines.

## How to do it...

Let's use Format-List to modify our information output on a couple different PowerShell cmdlets:

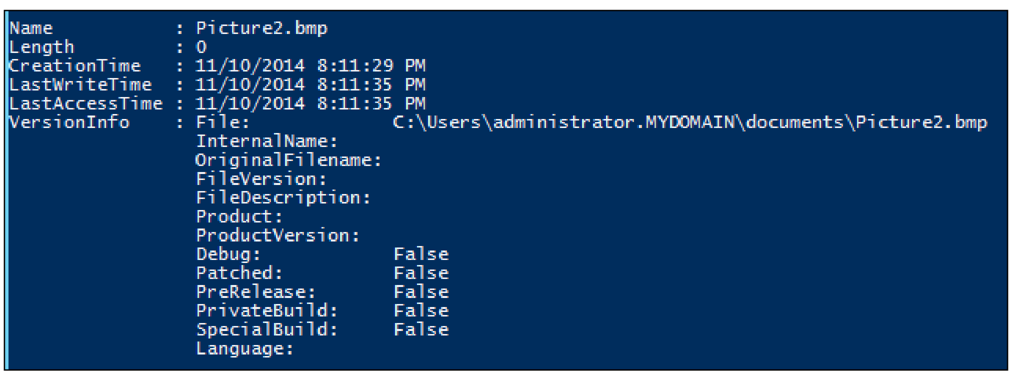
1. Right-click on your PowerShell icon from the Taskbar and choose **Run as Administrator**.
2. Browse to a location that contains some files. I have a few saved in my `Documents` folder, so I will input `cd documents` in order to navigate into my `Documents` folder.

3. Type `dir`. Then press *Enter*. You see the normal output of the `dir` command, a simple list of the files, and a little bit of information about each of them.




4. Now instead of using a simple `dir`, give this command a try: `Dir | Format-List`.

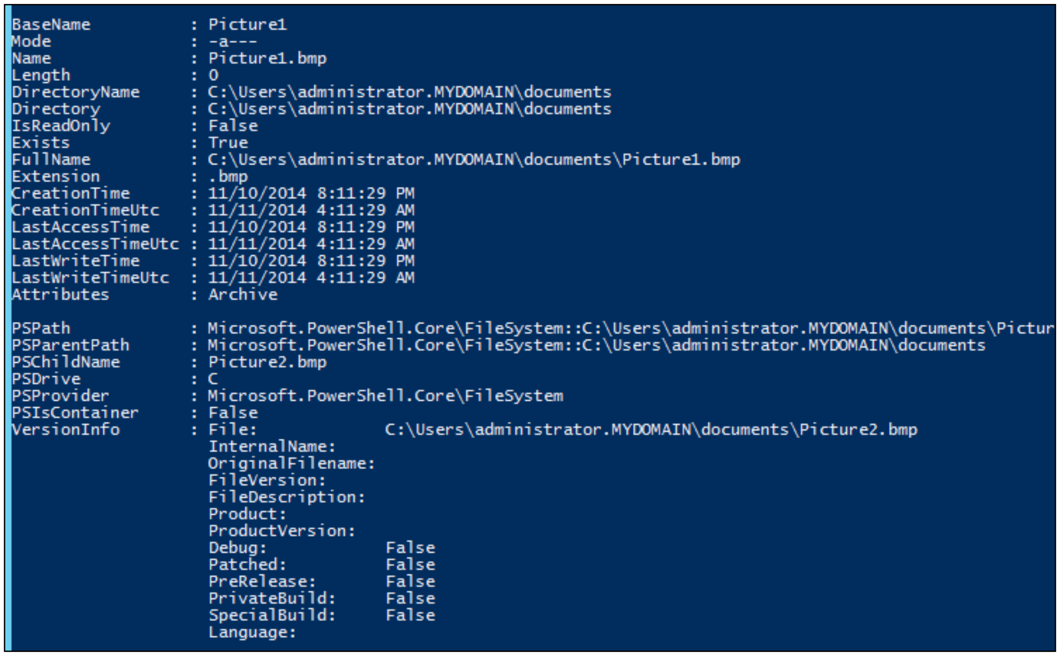
5. That is a lot more data! As you can see, by simply adding a pipe with Format-List following it, we have enhanced the `dir` command to give us more information about these files.



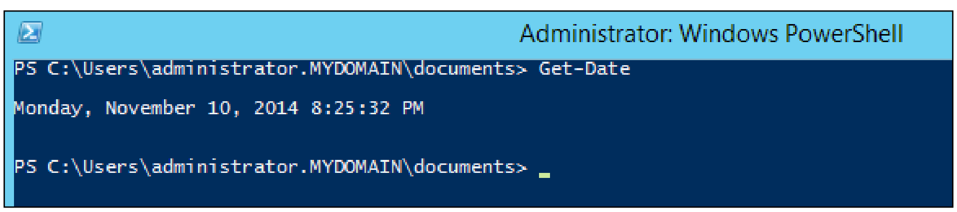

Using Format-List by itself will adjust the data output to another default format, but one that contains more information. In order to see everything there is to see in the output of a particular command, you can also add a `*` to the end of the command. Let's give that a try.

6. Type this command: `Dir | Format-List *`.

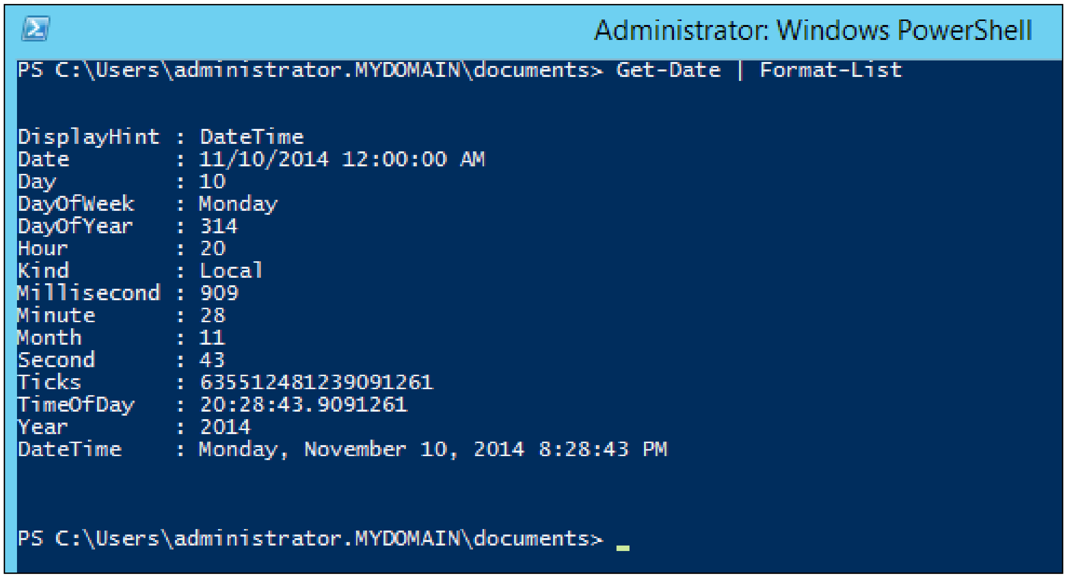
7. Now we have almost a full page worth of data about each individual file.




8. Before we lay this one to rest, let's test Format-List with another cmdlet, just to make sure this isn't something that only works with file information.
9. Use the `Get-Date` cmdlet to see the current date and time. Pretty simple, right?



10. Now try this: `Get-Date | Format-List`.




## How it works...

As we have shown here in this recipe, using the Format-List parameter on the end of any command or cmdlet is a good practice to get into, because it can help display much more information than would normally be available with the original command. From system timestamps and file information up to very particular information about NIC settings and system components, making Format-List part of your regular arsenal will help to get you a greater quantity of information that you can use to do your job.

# Configuring a full system backup using Windows Server Backup

Maintaining a good backup solution is so critical to administering a corporate server environment in today's IT world. There are limitless potential options for designing your particular backup plan, all the way from file copy backups to redundant servers sitting in hot standby mode.

While many of the third-party tools and technologies provide the capability to back up all of your servers simultaneously while retaining multiple previous versions of each, those tools are not always on the table because of cost and implementation complexity. Let's take a few minutes and familiarize ourselves with the built-in backup solution that Microsoft provides free of charge, right in the Server 2012 R2 operating system.

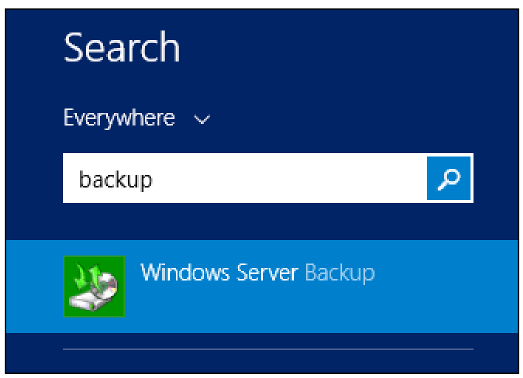
## Getting ready

We are logged into our Server 2012 R2 web server. We will be using the built-in Windows Server Backup tool in order to create a full image of this server.

## How to do it...

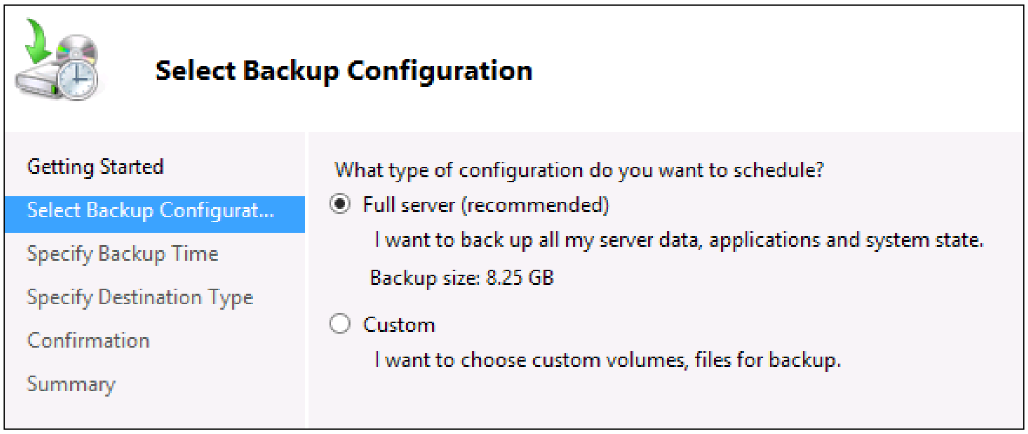
Follow along to back up your Server 2012 R2 using the built-in Windows Server Backup:

1. Open **Server Manager** and click the link **Add roles and features**. Walk through this wizard, following the steps in order to install the feature called Windows Server Backup.
2. Launch **Windows Server Backup** from your **Start** screen.

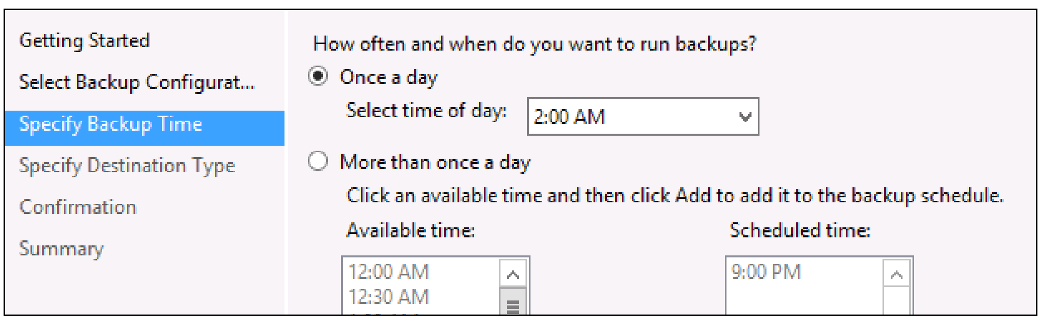



3. In the left window pane, choose **Local Backup**.
4. Then toward the right of your screen, click on the **Backup Schedule...** action and click on **Next**.

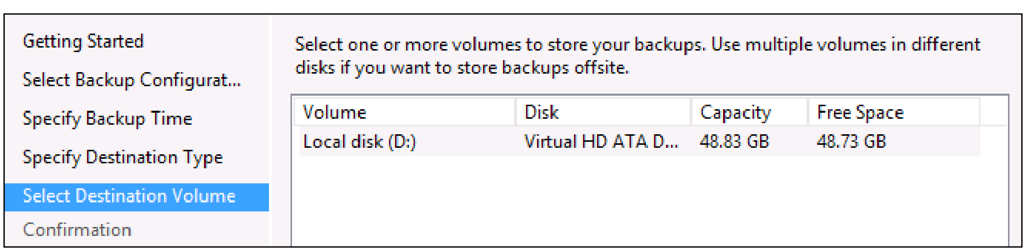
5. On the **Select Backup Configuration** screen, I am going to choose **Full server**. If you have only specific items you would like to backup, you can use the **Custom** option for that purpose.




6. Specify the schedule for how often you would like these backups to run. I'm going to have mine run every morning at **2:00 AM**.




7. As you can see in the text, the best way to store backups is to have a dedicated hard disk plugged into your server. However, I don't have an extra drive installed here, so I am going to choose **Back up to a volume** and specify my D drive, a separate partition that has no data on it currently, as my storage container for backups.



## How it works...

In this recipe, we installed the Windows Server Backup feature into our server and walked through the wizard in order to schedule daily full backups. This is a straightforward process, but the storage location of your backup files can take a little bit of consideration. A dedicated hard disk is the best solution for storing backups; that way if your drive goes down you will have all of the backup files on another physical disk. And then of course if you configure an option for replicating that data off to another physical site, or rotating drives on a schedule, that will protect your data even better in the event of a site failure or catastrophe. Storing onto a separate volume on the same disk is also an option, but then you are in a situation where that physical disk is a single point of failure for both your live operating system and the backup files. The third option is storing backup files on the network. This is something that I expect a lot of admins will choose, but you have to keep in mind that, when making this configuration, you will only be able to have one backup file stored in that network location at a time from your server, as they will be overwritten with each new backup process.

There is a second action available from inside the backup console that we didn't touch on. In order to accomplish ad hoc backups, or backups that you intend to create manually on an as-needed basis, you could launch the action called **Backup Once...**. Use this to create a manual backup copy at any time.

# Recovering data from a Windows backup file

Creating a backup or even a backup schedule is easy enough, but what is the process for restoring information from one of those backup files that we have sitting around? This is where the rubber really meets the road, as they say. Let's run through the process of restoring some data from a backup file that was taken yesterday. Perhaps some data was corrupted or accidentally deleted. Whatever the reason for our recovery needs, we will work together to restore some data from a backup file and get comfortable with that interface.

## Getting ready

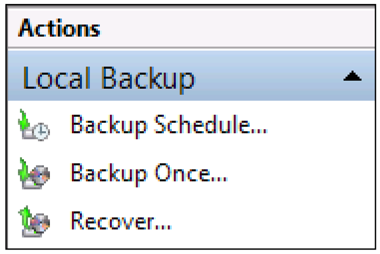
We are still working on our Server 2012 R2 web server. This server was previously configured for Windows Server Backup, so it already has that feature installed. Yesterday we created a full backup of our server, and today we need to recover some of the data from that backup file.

## How to do it...

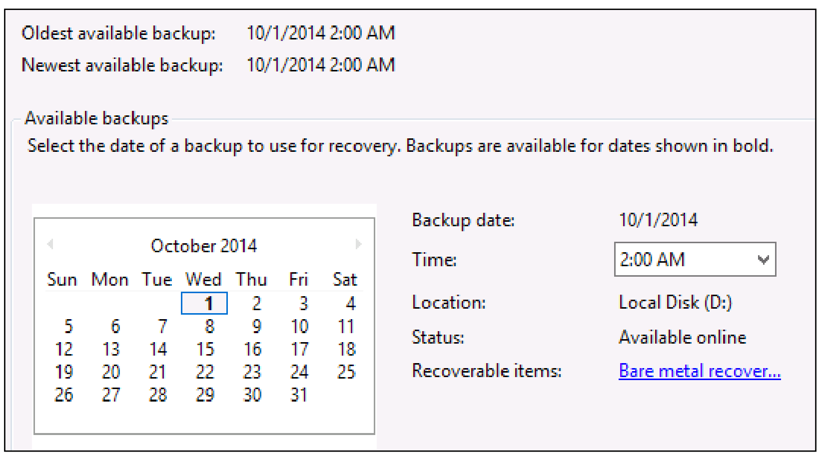
Follow these steps in order to restore the server using the Windows Server Backup utility:

1. Open up the Windows Server Backup management interface. You can launch this from either the Start screen or from the Tools menu of Server Manager.
2. Choose **Local Backup** from the left window pane.

3. Near the right side of your screen, click on the **Recover...** action.



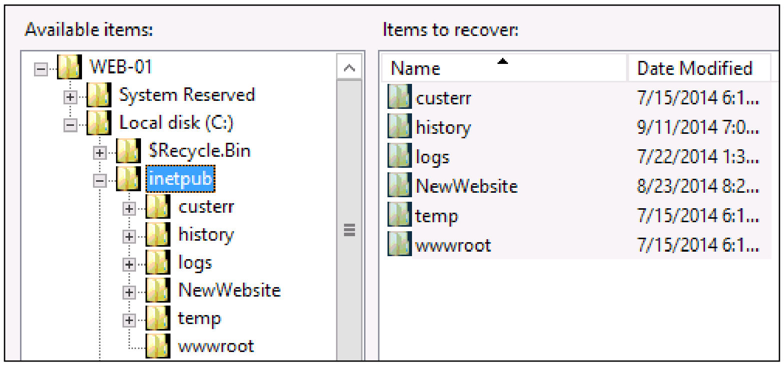

4. Since our backup file is stored right here on one of the server's volumes, we choose **This server** and click **Next**.
5. Now you will see a calendar with bold dates, indicating which days have valid backup files that you can restore back to. We are selecting the backup that ran yesterday and clicking **Next**.




6. Now on the **Select Recovery Type** screen, we are going to choose **Files and folders** and click **Next**.

> You will notice a grayed out option here for Hyper-V. If you use Windows Server Backup on a Hyper-V server, you have options for backing up and restoring individual virtual machines on that host. This is a great feature enhancement and a good reason to start using Windows Server Backup on your Hyper-V servers.

7. We are now looking at the **Select Items to Recover** screen. Simply choose the files and folders that you want to restore from yesterday's backup. For our web server, it was the website itself that was compromised and we want to roll back to the website files that were running yesterday. So I am going to choose to restore the `C:\inetpub` folder.




8. Choose the options to recover files to **Original location** and click **Overwrite the existing versions with the recovered versions**. This will ensure that the files from yesterday's backup get placed on top of the files that still exist on our server today.
9. On the **Confirmation** screen, you will see a summary of the items that are going to be recovered. If everything looks good, click on the **Recover** button.

## How it works...

This recovery recipe is a good baseline for getting familiar with the options that are available to you for restoring from Windows backup files. Here we restored some simple files that had been compromised on our web server. In the event of a more serious system failure, where you might need to take a full disk backup and recover the whole thing onto a new server, that process is slightly more complicated. To accomplish a full system recovery of that magnitude, you would boot the server into your Windows setup disk, and choose to run Windows Recovery Environment. Through this tool you could make use of your Windows backup file and restore the server.

# Using IP Address Management (IPAM) to keep track of your used IP addresses

The **IP Address Management** (**IPAM**) tool is a little-known utility built into Windows Server 2012 R2. IPAM is a way that you can centrally monitor and manage some of the common infrastructure roles spread out around your network. Most specifically for our recipe today, we will be taking a look at IP addressing by using IPAM.

Particularly in environments where there may be many different DHCP servers that are hosting different scopes spread out around your network, IPAM can be extremely useful for pulling all of that information into one management interface. This saves a lot of time and effort as opposed to launching the DHCP Management console on each of your DHCP environments separately and trying to monitor them individually.

## Getting ready

We have a domain network running that consists of all Server 2012 R2 servers. Included in our network is a Domain Controller that is also serving as a DHCP server. We are adding a new server to this mix called IPAM-01. This new server will be our IPAM management server, as the IPAM feature should not co-exist with either the AD DS Role or with the DHCP Role.

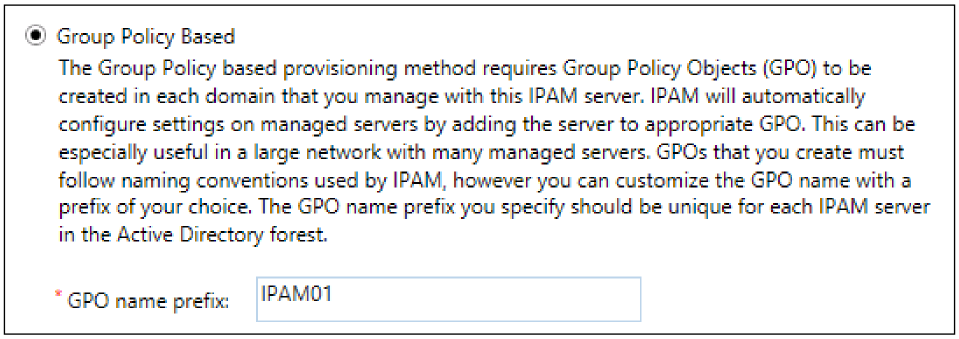
## How to do it...

Let's take a look at our IP address utilization with the IPAM feature:

1. While logged into the new server that you intend to use for IPAM, click the **Add roles and features** link from inside Server Manager.
2. Walk through this wizard, choosing the option to add the feature called **IP Address Management (IPAM) Server**.
3. Once the feature has been installed, you should see a new listing for IPAM in the left window pane of Server Manager. Go ahead and click there.
4. You will see that step 1 is already accomplished, the IPAM console is successfully connected to the local server. Go ahead and click on step 2 in order to provision the IPAM server.


5. Click **Next**, taking care to read over the information listed on that screen. As you can see, the best way to set up interaction between the IPAM server and the infrastructure servers is to utilize Group Policy. We will define the settings for that on an upcoming screen in this wizard.
6. You should now be on the **Configure database** screen and we will leave the default option selected to utilize **Windows Internal Database (WID)**.

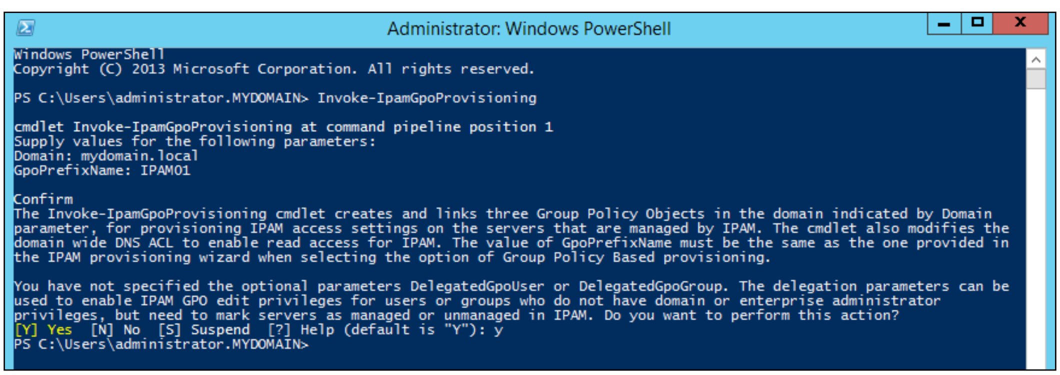
7. Now we get to select our provisioning method, which is where we are going to tell IPAM to use Group Policy in order to distribute the settings that it needs in order to manage and grab data off our infrastructure servers. Define a GPO prefix that is specific to this IPAM server.




8. Before we complete this wizard, we need to take a special action in order to provision these GPOs, so that the wizard can make use of them. To do this, we are going to use a PowerShell cmdlet. Right-click on your PowerShell icon in the Taskbar, and open a PowerShell prompt as Administrator. Make sure you are logged into the server as a domain admin before running this cmdlet.

9. Type the following command into PowerShell: `Invoke-IpamGpoProvisioning`.

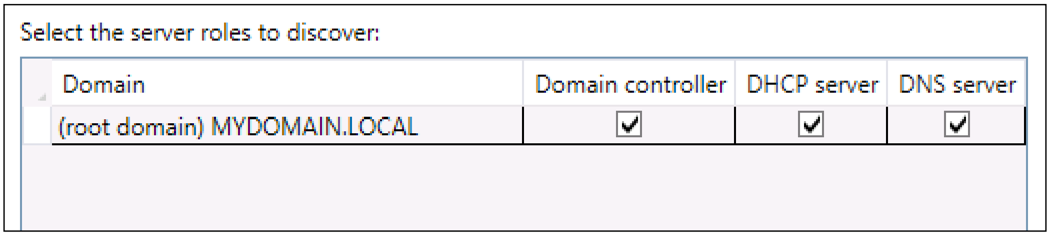
10. It will ask you to key in the name of your domain, as well as the GpoPrefixName. This is the same prefix that you just typed into the IPAM Wizard, so make sure you enter it exactly the same.



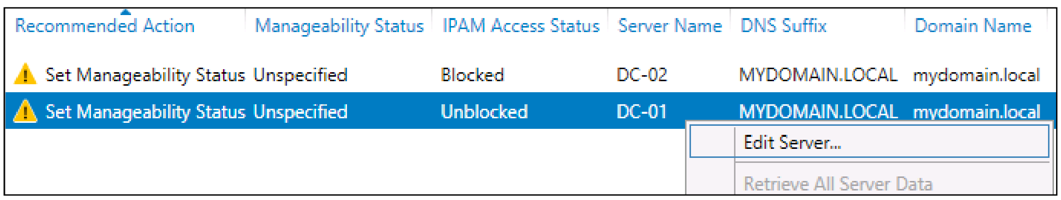

11. Now that our GPOs have been created, head back over to the IPAM Wizard and click the **Apply** button to finish it.

12. Now back at the IPAM section of Server Manager, click on step 3—**Configure server discovery**.

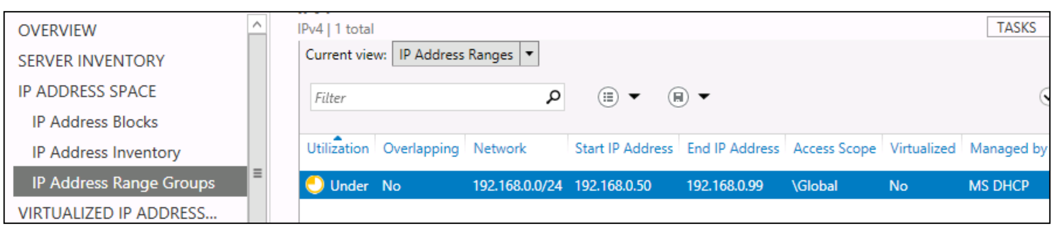
13. Use the **Add** button in order to query your domain for infrastructure services that can be monitored by IPAM. Select the roles you would like to pull data from (I am going to leave all three checked) and click the **OK** button.




14. Click on step 4—**Start server discovery**. Wait for discovery to complete.
15. Click on step 5—**Select or add servers to manage and verify IPAM access**.
16. Right-click on the server that you want to collect data from and choose **Edit Server...**.




17. Change the server's **Manageability status** field to **Managed**.
18. Now head back to the main IPAM window in Server Manager and click on step 6—**Retrieve data from managed servers**.
19. Once data collection completes, you now have the ability to browse around inside the IPAM management console and view data about your DNS and DHCP infrastructure. For example, click on **IP Address Range Groups** to see a list of the DHCP scopes that are present on the DHCP servers that you are currently managing.



## How it works...

The IP Address Management (IPAM) tool takes a little bit of work to configure initially, but can be very beneficial later. Once configured to pull in data from your Domain Controllers, DNS servers, and DHCP servers, IPAM can be your one-stop-shop for monitoring and managing data related to these infrastructure roles. This is particularly helpful in cases where you have many servers providing these roles, such as the case of multiple DHCP servers that each contain different scope definitions. In the past, you would have had to log in to each DHCP server or at least do remote management of them via Server Manager or some other tool, but ultimately you would still be viewing and managing the DHCP scopes individually. With IPAM, it brings all of this information into one place so that you can make decisions and configuration changes within your network while looking at the overall bigger picture.

# 9
# Group Policy

In this book, we have already discussed a few recipes that call for the modification of **Group Policy Objects** (**GPOs**), but we have not taken the time to discuss why Group Policy is important in the first place. To those who have worked within Active Directory for a while, Group Policy may be familiar territory. I still find, though, that many IT folks working in the Server Administrator role are not overly familiar with Group Policy and how it can benefit them. Particularly in smaller companies, this incredibly powerful feature of Windows tends to be overlooked. It is easy to think of Active Directory as the storage container for your user and computer accounts, because those are the core necessary tasks that it accomplishes. But as soon as you install the Domain Services role to configure your first Domain Controller, you have automatically included Group Policy capabilities into that domain.

Let's walk through some recipes together to make sure you are able to interact with Group Policy comfortably and scratch the surface on what capabilities lie beneath these covers:

- Creating and assigning a new Group Policy Object (GPO)
- Mapping network drives with Group Policy
- Redirecting the My Documents folder to a network share
- Creating a VPN connection with Group Policy
- Creating a printer connection with Group Policy
- Using Group Policy to enforce an Internet proxy server
- Viewing the settings currently enabled inside a GPO
- Viewing the GPOs currently assigned to a computer
- Backing up and restoring GPOs

# Introduction

**Group Policy** is a centralized administration tool for your domain joined systems. To summarize its capabilities, you can create policies in Active Directory, assign those policies to particular users or computers, and within those policies change any number of settings or configurations that are within the Windows operating system. The item inside Active Directory that contains these settings is called a Group Policy Object (GPO), so we will be focusing on the creation and manipulation of these in order to make some centralized management decisions that will affect large numbers of computers in our environment. GPOs can be utilized for user accounts, client computer settings, or for putting configurations onto your servers. Any domain joined system can be manipulated by a GPO; typically those settings put into place by GPOs cannot be overridden by the users, making them a very integral part of security for the companies that are familiar with making use of Group Policy regularly.

We will place a number of different configuration settings inside the GPOs that we create throughout this chapter, but we will not come close to covering even a fraction of the available settings that could be manipulated. For a full reference of Group Policy settings that are available, please check out the following link: `http://www.microsoft.com/en-us/download/details.aspx?id=25250`.

# Creating and assigning a new Group Policy Object (GPO)

In order to start using Group Policy, we first need to create a Group Policy Object. Most commonly referred to as a GPO, this object contains the settings that we want to deploy. It also contains the information necessary for domain joined systems to know which machines and users get these settings and which ones do not. It is critical that you plan GPO assignment carefully. It is easy to create a policy that applies to every domain joined system in your entire network but, depending on what settings you configure in that policy, this can be detrimental to some of your servers. Often I find that admins who are only somewhat familiar with Group Policy are making use of a built-in GPO called Default Domain Policy. This, by default, applies to everything in your network. Sometimes this is actually what you want to accomplish. Most of the time, it is not!

We are going to use this section to detail the process of creating a new GPO, and use some assignment sections called **Links** and **Security Filters** that will give us complete control over which systems receive these systems, and more importantly, which ones do not.

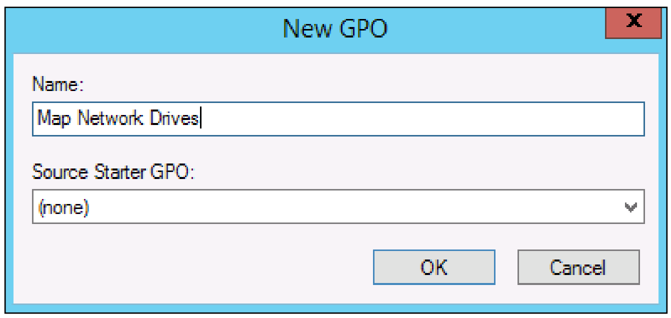
## Getting ready

Our work today will be accomplished from a Server 2012 R2 Domain Controller server. If you are running the Domain Services role, you already have the items installed that are necessary to manage Group Policy.

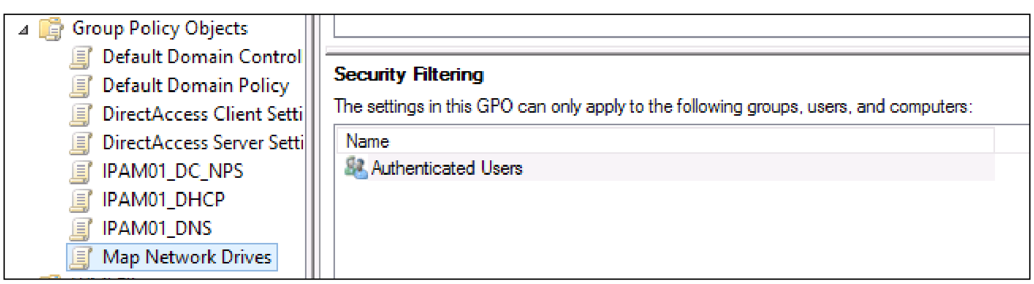
## How to do it...

Follow these steps to create and assign a new Group Policy Object:

1. Open **Server Manager**, click on the **Tools** menu, and choose to open the **Group Policy Management** Console.
2. Expand your domain name and click on the folder called **Group Policy Objects**. This shows you a list of your current GPOs.
3. Right-click on the **Group Policy Objects** folder and click on **New**.
4. Insert a name for your new GPO. I am going to call mine `Map Network Drives`. We will end up using this GPO in a later recipe.

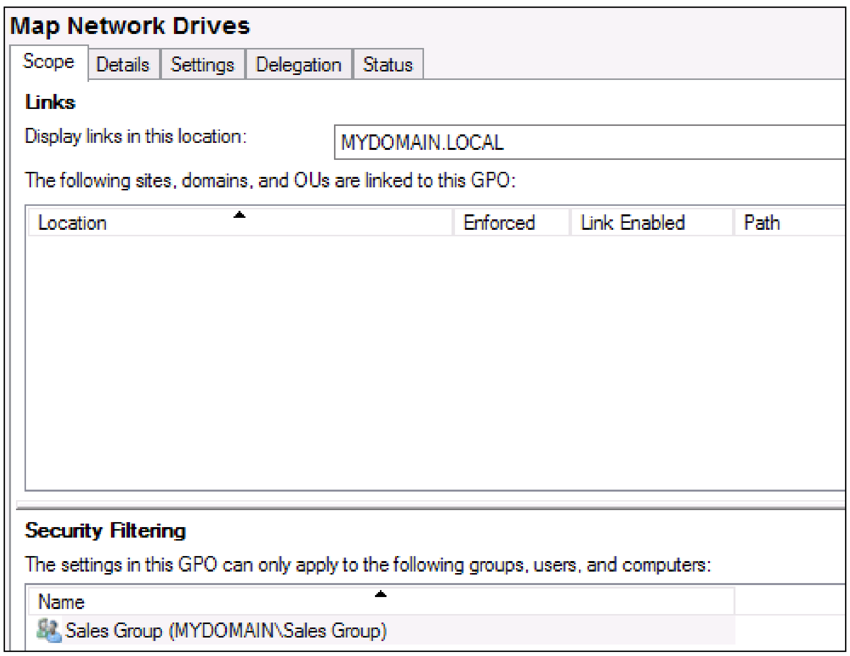



5. Click **OK**, and then expand your **Group Policy Objects** folder if it isn't already. You should see the new GPO in this list. Go ahead and click on the new GPO in order to see its settings.
6. We want this new GPO to apply only to a specific group of users that we have established. This assignment of the GPO is handled at the lowest level by the **Security Filtering** section that you see on this screen. You can see that, by default, **Authenticated Users** is in the list. This means that, if we created a link between this GPO and an OU in the domain, the policy settings would immediately start applying to any user account.



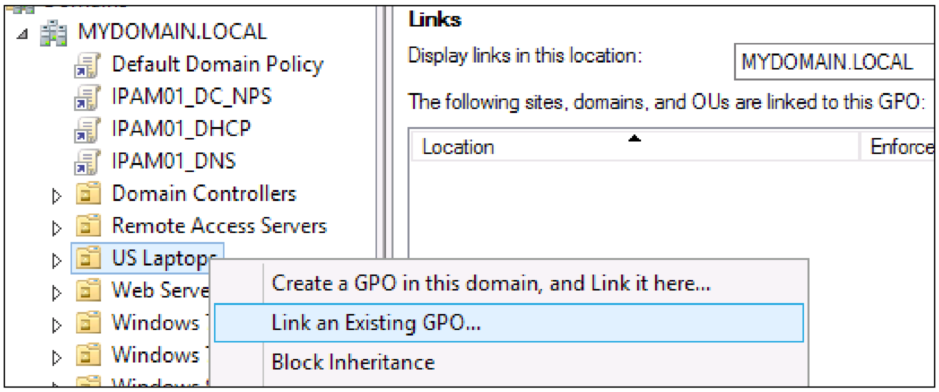
Since we want to make absolutely sure that only specific user accounts get these drive mappings, we are going to modify the **Security Filtering** section and list only the user group that we have created to house these user accounts. Under the **Security Filtering** section, click on the **Remove** button in order to remove **Authenticated Users** from this list. It should now be empty.

7. Now click on the **Add...** button, also listed under the **Security Filtering** section.
8. Type the name of your group for which you want to filter this GPO to. My group is called **Sales Group**. Click **OK**.
9. Now this GPO will only apply to users we place into the group called **Sales Group**, but at this point in time the GPO isn't going to apply anywhere because we have not yet established any links. This is the top section of your **Scope** tab, which is currently blank.

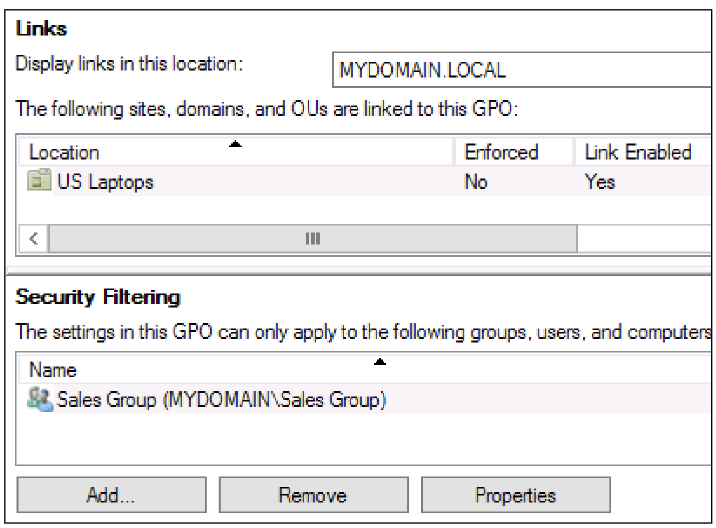



We need to link this GPO to some place in our domain structure. This is essentially telling it "apply this policy from here down" in our OU structure. By creating a link with no security filtering, the GPO would apply to everything under that link. However, since we do have Security Filtering enabled and specified down to a particular group, the security filtering will be the final authority in saying that these GPO settings will only apply to members of our Sales group. For this Map Network Drives policy, we want it to apply to the OU that is called **US Laptops**.

10. Right-click on the OU called **US Laptops** and then click on the option for **Link an Existing GPO...**.




11. Choose the name of our new GPO, **Map Network Drives**, and click **OK**.




Our new GPO is now linked to the **US Laptops** OU; so at this level any system placed inside that OU would get the settings. Except that we paired it down a step further with the Security Filtering section. Since we populated this with only the name of our specific Sales Group, this means that this new drive mapping policy will only apply to those added into this group.

## How it works...

In our example recipe, we created a new Group Policy Object, and took the steps necessary in order to apply this GPO only to the computers and users that we deemed necessary inside our domain. Each network is different, and you may find yourself relying only on the Links to keep GPOs sorted according to your needs, or you may need to enforce some combination of both Links and Security Filtering. In any case, whichever works best for you, make sure that you are confident in the configuration of these fields so that you can know beyond a shadow of a doubt where your GPO is being applied. You may have noticed that, in our recipe here, we didn't actually configure any settings inside the GPO, so at this point it still isn't doing anything to those in the Sales Group. Continue reading to navigate the actual settings portion of Group Policy.

# Mapping network drives with Group Policy

Almost everyone uses mapped drives of some flavor in their environments. Creating drive mappings manually as part of a new user start-up process is cumbersome and unnecessary. It is also work that will likely need to be duplicated as users move from one computer to another in the future. If we utilize Group Policy to centralize the creation of these drive mappings, we can ensure that the same users get the same drive mappings wherever they log into the network. Planned correctly, you can enable these mappings to appear on any domain joined system across the network by the user simply logging in to the computer as they always do. This is a good, simple first task to accomplish within Group Policy to get our feet wet and to learn something that could turn out to be useful in your organization.

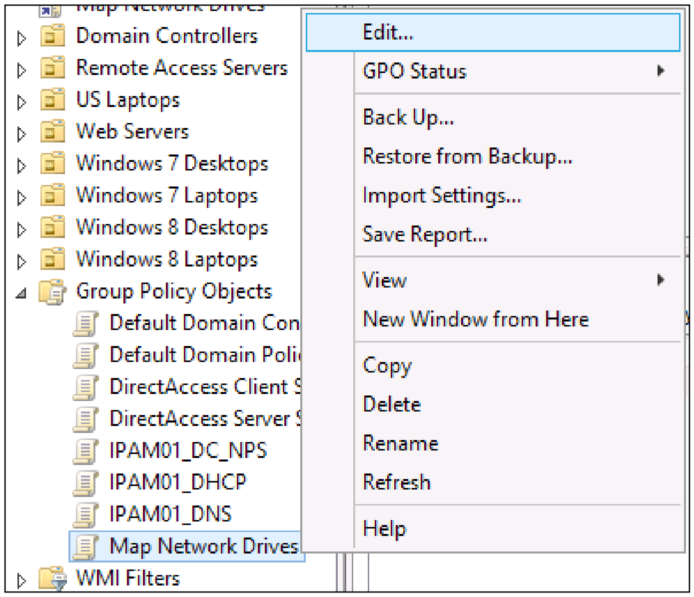
## Getting ready

We are using a Server 2012 R2 Domain Controller in our environment in order to create and configure this Group Policy Object. We will assume that you have already created a new GPO for this task that has been configured for Links and Security Filtering.

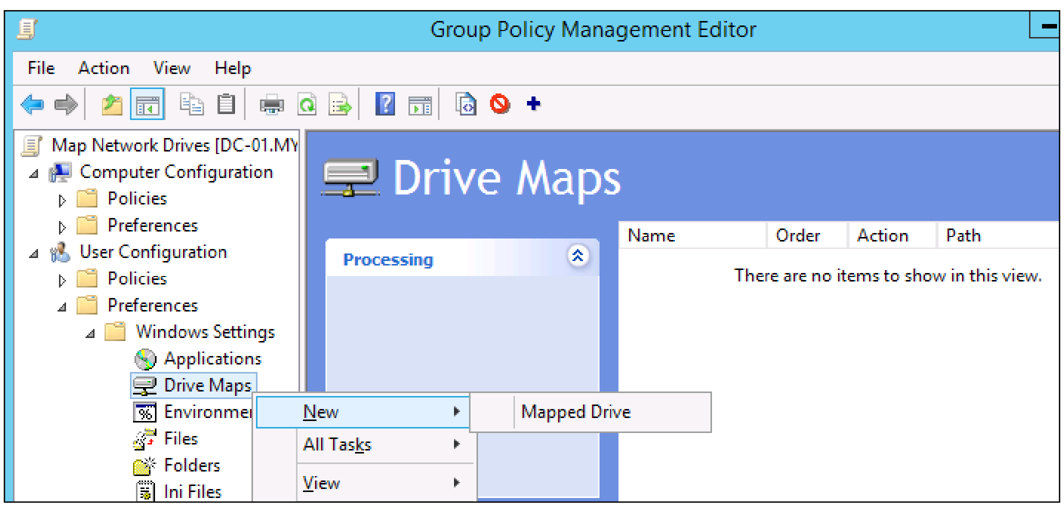
## How to do it...

To create a drive mapping in Group Policy use the following steps:

1. Open the **Group Policy Management** Console from the **Tools** menu of Server Manager.
2. Expand the name of your domain and then expand the **Group Policy Objects** folder. There we see our new GPO called **Map Network Drives**.
3. Right-click on the **Map Network Drives** GPO and click on **Edit...**.

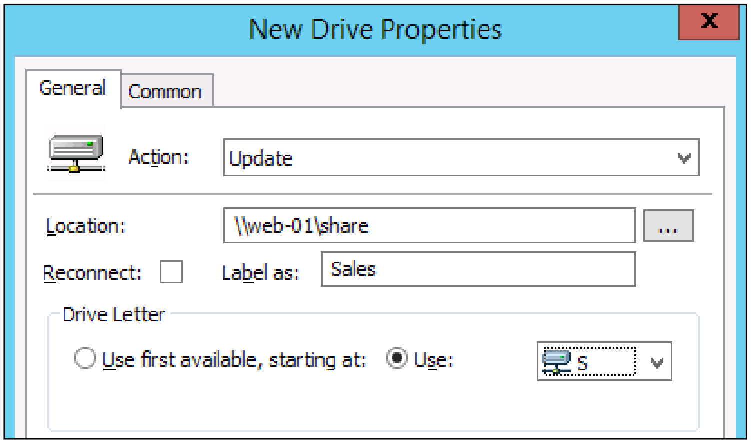



4. Browse to **User Configuration** | **Preferences** | **Windows Settings** | **Drive Maps**.
5. Right-click on **Drive Maps** and choose **New** | **Mapped Drive**.

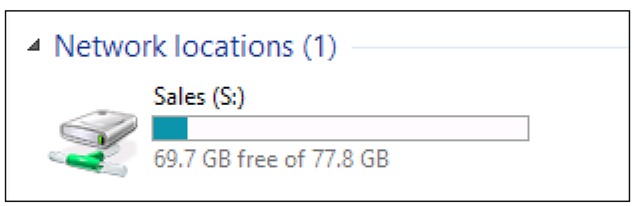



6. Set **Location** as the destination URL of the drive mapping, and use the **Label as** field if you want a more descriptive name to be visible to the users.

7. Choose a **Drive Letter** to be used for this new mapping from the down menu listed on this screen:




8. Click **OK**.
9. We are assuming you have already created the Links and Security Filtering appropriate to where you want this GPO to apply. If so, you may now log into a computer on your domain as a user account to which this policy will apply. Once logged into the computer, open up File Explorer and you should see the new network drive mapped automatically during the login process.




## How it works...

There are a few different ways that drive mappings can be automated within a Windows environment, and our recipe today outlines one of the quickest ways to accomplish this task. By using Group Policy to automate the creation of our network drive mappings, we can centralize the administration of this task and take drive mapping creation off our helpdesk processes.

# Redirecting the My Documents folder to a network share

Users are accustomed to saving documents, pictures, and more into their `My Documents` folder, because that is what they do at home. When working on an office computer at their job, the natural tendency is to save right into the local `Documents` folder as well. This is generally not a desired behavior, because backing up everyone's documents folders individually would be an administration nightmare. So the common resolution to this problem is to provide everyone with mapped network drives and train users to save documents into these mapped drives. This is good in theory, but difficult to execute in practice. As long as users still have the capability to save documents into their local `My Documents` folder, there is a good chance that they will save at least some things in there, probably without realizing it.

This recipe is a quick Group Policy change that can be made so that the `My Documents` folders on your domain joined computers get redirected onto a network share. This way, if users do save a document into `My Documents`, that document gets written over to the file server where you have directed them.

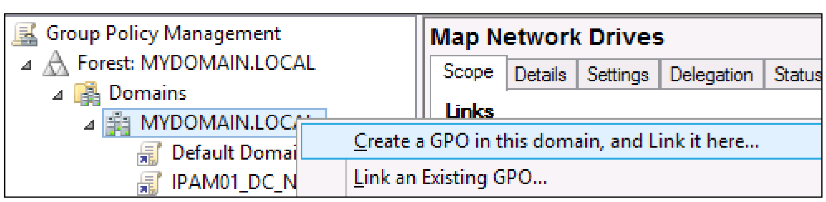
## Getting ready

We will set up our new GPO on a Server 2012 R2 Domain Controller.

## How to do it...

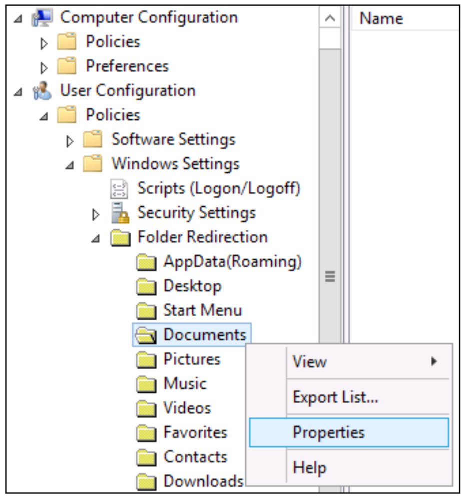
To redirect the `My Documents` folders via Group Policy use the following steps:

1. Launch the **Group Policy Management** Console from the **Tools** menu of Server Manager.
2. Right-click on the name of your domain and choose **Create a GPO in this domain, and Link it here...**.

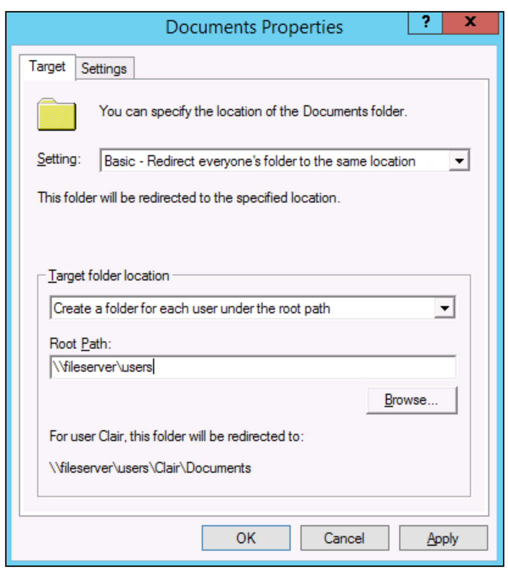



3. Input something in the **Name** field for your new GPO. I am going to call mine **Redirect My Documents**. Then click **OK**.
4. Browse to the **Group Policy Objects** folder that is listed under your domain name.
5. Right-click on the name of our new redirection GPO and click **Edit...**.

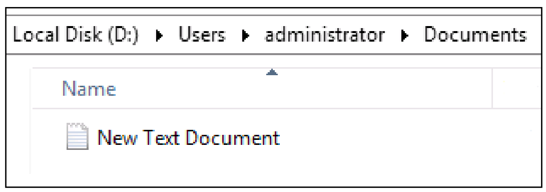
6. Browse to **User Configuration** | **Policies** | **Windows Settings** | **Folder Redirection** | **Documents**.
7. Right-click on **Documents** and go into **Properties**.




8. From the **Setting** drop-down select **Basic - Redirect everyone's folder to the same location**.
9. Type in the **Root Path** field where you want everyone's `Documents` folder to be directed to. I am going to use a share that I have created on our file server. Mine will look like this: `\\fileserver\users\`.



10. Click **OK**.
11. Your setting should be put immediately into place within the GPO. Now go ahead and log in to a test client machine and open up the `My Documents` folder.
12. Create a new text document inside the local `Documents` folder. We are just creating something here in local `Documents` so that we can see where it is actually being stored.
13. Now log in to your file server and check inside the `Users` directory that we specified. We now have a folder in there with my username, and inside that folder is a `Documents` folder that contains the new text document that I just created and stored inside the local `My Documents` on my client computer!




## How it works...

Redirecting everyone's My Documents to be automatically stored on a centralized file server is an easy change with Group Policy. You could even combine this configuration with another that maps a network drive, then simply specify the drive letter under your Document Redirection setting rather than typing out a UNC that could potentially change in the future. However you decide to configure it in your environment, I guarantee that using this setting will result in more centralized administration of your data and fewer lost files for your users.

# Creating a VPN connection with Group Policy

If you have administered or helped support a VPN connectivity solution in the past, you are probably more than familiar with setting up VPN connection profiles on client computers. In an environment where VPN is utilized as the remote access solution, what I commonly observe is that the VPN profile creation process is usually a manual step that needs to be taken by human hands, following the user's first login to the computer. This is inefficient and easily forgotten. With tools existing in your Windows Server 2012 R2, you can automate the creation of these VPN connections on the client computers. Let's use Group Policy to create these profiles for us during user login.

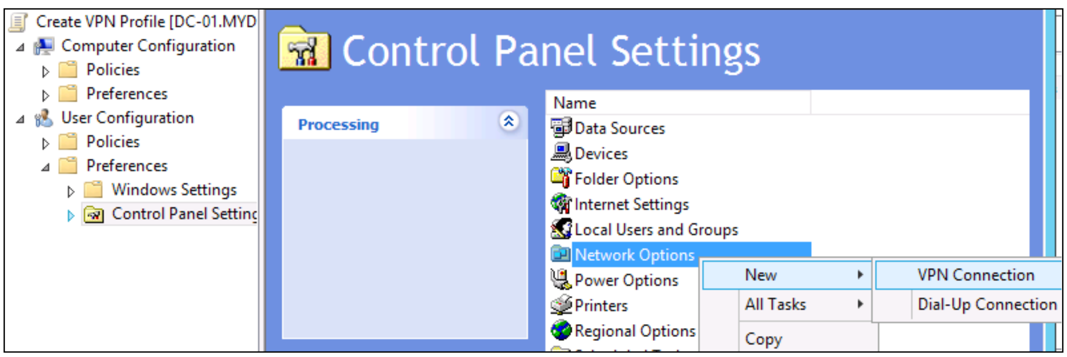
## Getting ready

We will use a Server 2012 R2 Domain Controller in order to configure our new Group Policy Object. Once finished, we will also use a Windows 8 client computer to log in and make sure that our VPN profile was successfully created. For this recipe, we are going to assume that you create the GPO and setup links, and filter it according to your needs before we get started with the actual configuration of this GPO.

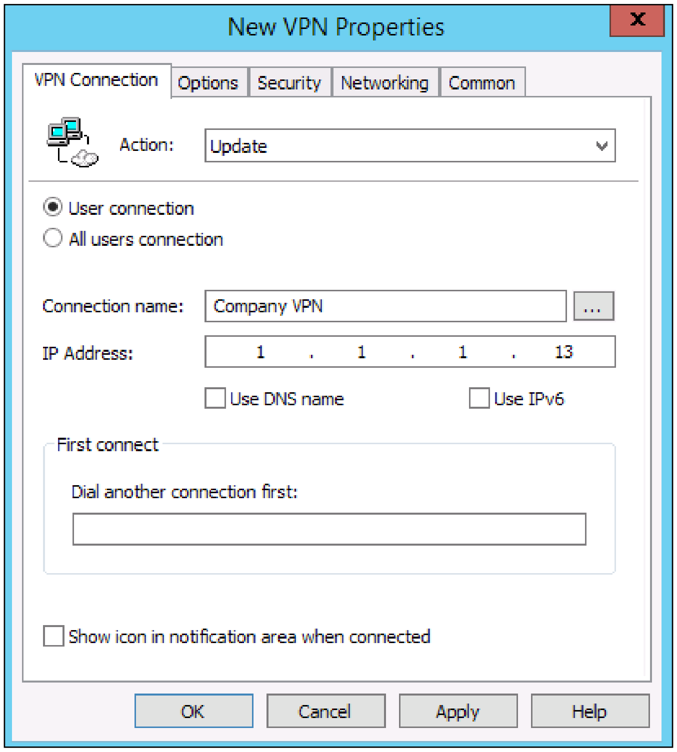
## How to do it...

Follow along to configure a GPO that will automatically create a VPN connection profile on your remote client computers:

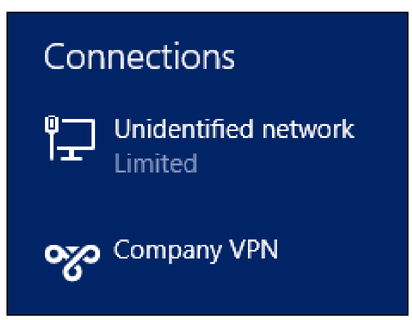
1. Inside the **Group Policy Management** Console, right-click on your new GPO that will be used for this task and click on **Edit...**.
2. Browse to **User Configuration** | **Preferences** | **Control Panel Settings**.
3. Right-click on **Network Options** and choose **New** | **VPN Connection**.




4. Input something in the **Connection name** field for this new VPN connection; this name will be displayed on the client computers, and the public **IP address** field that client computers will need to connect to while working remotely. Depending on the needs for your particular VPN connection, you may also have to visit the additional tabs available on this screen to finish your specific configurations. Then click **OK**.




5. Now log in to your client computer and click on the Network icon in the systray, the same place where you would click in order to connect to a wireless network. You can see that, during our login to this computer, a new VPN connection called **Company VPN** has been added and is now available to click on.



## How it works...

In this recipe, we used Group Policy to automate the creation of a new VPN connection for our remote laptops. Using a GPO for something like this saves time and effort, since you are no longer setting up these connections by hand during a new PC build. You can also use this function to update settings on an existing VPN connection in the future, if you need to change IP addresses or something like that. As you are starting to see through these recipes, there are all kinds of different things that Group Policy can be used to accomplish.

# Creating a printer connection with Group Policy

Let's say you just installed a new network printer in the office. You have installed it on a few computers to make sure it works properly, but now you are staring down the rows and rows of computers that would like to print to this printer occasionally. The prospect of logging in to every computer in order to launch and walk through the printer creation wizard isn't sounding like the way you would like to spend your Friday night. Let's see if we can once again make use of Group Policy to save the day. We will utilize a new GPO that will be configured to automatically install this new printer on the client desktops.

## Getting ready

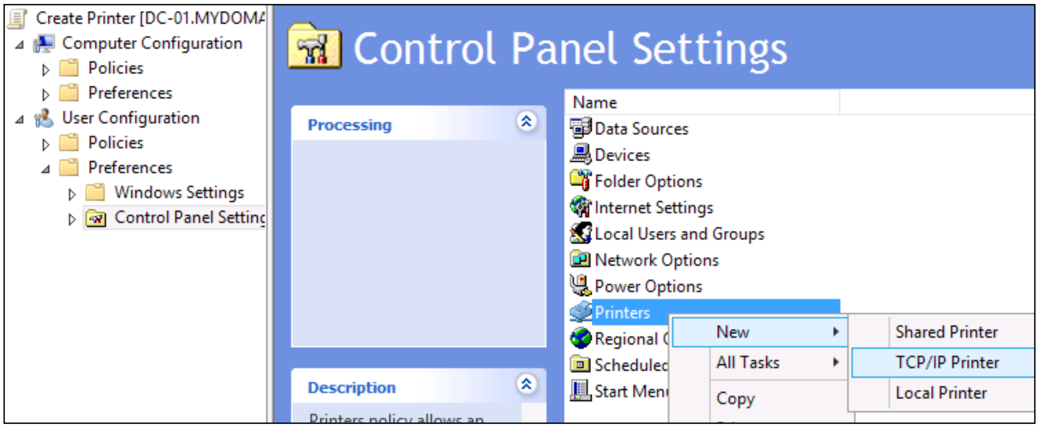
We are assuming you have already created the new GPO and have linked it accordingly so that only the computers that need this new printer are going to receive these GPO settings. Now we are going to use Group Policy Management Console on our primary Domain Controller that is running Windows Server 2012 R2.

## How to do it...

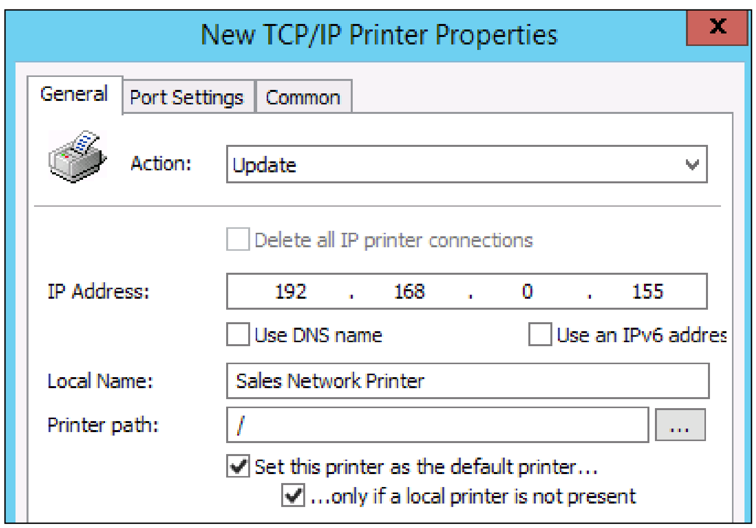
To configure your new GPO for a new printer creation:

1. Open **Group Policy Management** Console from the **Tools** menu of Server Manager.
2. Right-click on your new GPO that is going to be used for this printer creation and click **Edit...**.
3. Navigate to **User Configuration** | **Preferences** | **Control Panel Settings**.

4. Right-click on **Printers** and choose **New** | **TCP/IP Printer**.




5. Input the information that is necessary for the printer connection. Since we chose to set up a new TCP/IP Printer, we need to input something in the **IP Address** field and **Local Name** for the users to be able to see this new printer in their list. I am also going to choose **Set this printer as the default printer...only if a local printer is not present**.




6. Click **OK** and this printer will start building itself for the users that you filtered the GPO to.

### How it works...

Using Group Policy to automate regular IT tasks makes a lot of sense for all kinds of technologies. In this recipe, we built a simple printer connection so that we didn't have to do it by hand on our dozens of computers that wanted the ability to be able to print here.

## Using Group Policy to enforce an Internet proxy server

Most networks of significant size use a forward proxy server to filter their Internet traffic. This is essentially a box that sits out near the edge of the corporate network; whenever client computers in the network try to access the Internet, their requests are sent out through this server. Doing this enables companies to monitor Internet use, restrict browsing permissions, and keep many forms of malware at bay. When implementing a proxy server, one of the big questions is always "How do we enforce the use of this proxy?". Some solutions do a default route through the proxy server so that all traffic flows outbound that way at a network level. More often, though, it is desirable for the proxy server settings to be configured at the browser level because it is likely unnecessary that all traffic needs to flow through this proxy, only the browser's web traffic. In these cases, you could certainly open up Internet Explorer options on everyone's computers and enter the proxy server information, but that is a huge task to undertake and it gives the users the ability to remove those settings if they choose to.

By using Group Policy to set the Internet Explorer proxy configuration, this task will be automated and hands-off. This also ensures that users are not able to manipulate these fields in the future, and you can be assured that your web traffic is flowing through the proxy server as you have defined it.

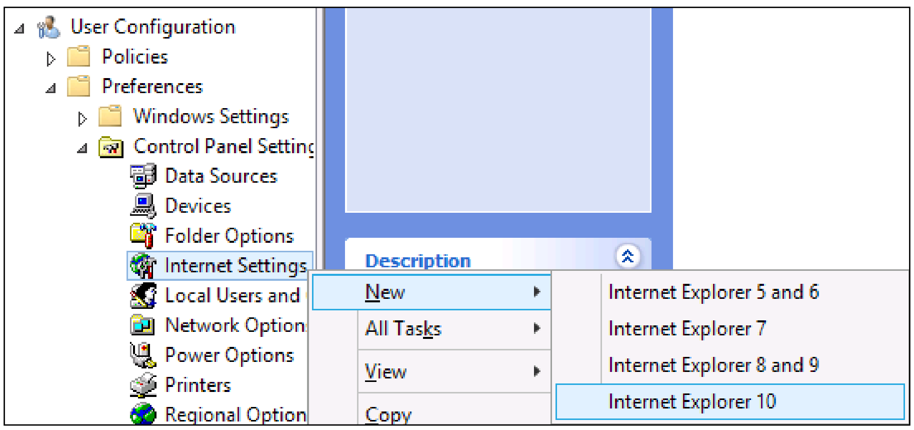
### Getting ready

Our GPO has already been created, now we are using the Group Policy Management Console on our Server 2012 R2 Domain Controller to configure settings within the GPO. A Windows 8 client computer is also sitting waiting for use as we will want to test this GPO after we finish the configuration.

## How to do it...

Follow these steps to set everyone's Internet proxy settings via Group Policy:

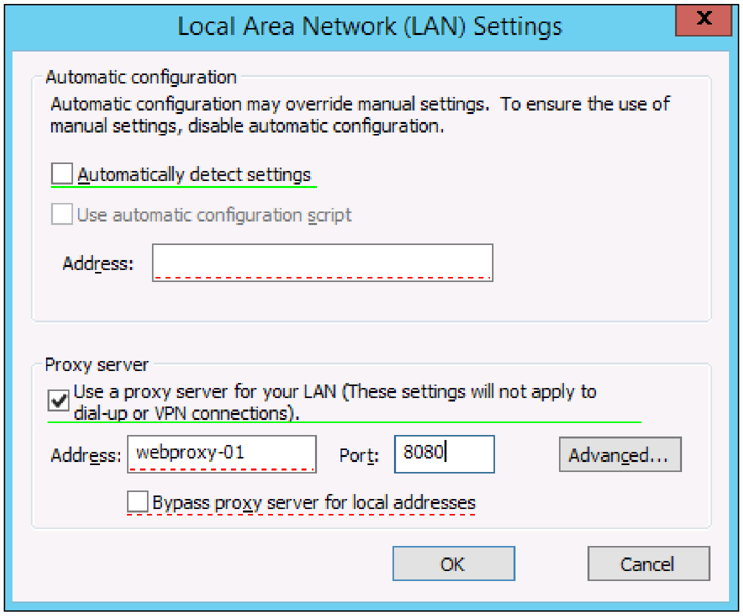
1. Open the **Group Policy Management** Console from the **Tools** menu of Server Manager.
2. Find the new GPO that you have created for this task, right-click on it, and choose **Edit...**.
3. Browse to **User Configuration** | **Preferences** | **Control Panel Settings** | **Internet Settings**.
4. Right-click on **Internet Settings** and choose **New** | **Internet Explorer 10**.




> You may have to create multiple policies here if you are using multiple versions of Internet Explorer on your workstations.

5. You will see a dialog box that looks just like the regular Internet options available in IE. You have the ability to change many things here but, for our purposes today, we are heading over to the **Connections** tab.
6. Click on the **LAN settings** button.

7. Check the box for **Proxy server**. Then input the **Address** and **Port** fields for your particular proxy server:




8. Click **OK** and your setting will be put into place.
9. Now log in to the client computer, and let's see whether this proxy server information was successfully implemented. Launch Internet Explorer and open **Internet options**.
10. Browse to the **Connections** tab and click on the **LAN settings** button to ensure your proxy server settings have been properly plugged in. Also notice that they are now grayed out, showing you that they have been configured by Group Policy and are not able to be manipulated manually.

## How it works...

Using Group Policy to assign Internet proxy server settings to all of your client computers with one simple GPO creation is another example of the power behind Group Policy. The possibilities for centralized administration of your domain joined machines are almost endless; you just need to do a little digging and find the right place inside the GPOs for changing your settings. Maybe you don't have a proxy server in your network and don't have a need for this recipe. But I still encourage you to take the steps listed here and apply them to some piece of technology that you do utilize. I guarantee there are settings inside Group Policy that can be of interest to anyone! Go out and find some that will help save you time and money.

# Viewing the settings currently enabled inside a GPO

So far we have been creating GPOs and putting settings into them, so we are well aware of what is happening with each of our policies. Many times, though, you enter into a new environment with lots of existing policies, and you may have the need to figure out what is happening in those policies. I have had many cases where I install a new server, join it to the domain, and it breaks. It doesn't necessarily nose-dive, but some component won't work properly or I can't flow network traffic to it for some reason. Something like that can be hard to track down. Since the issue seemed to happen during the domain join process, I can suspect that some kind of policy from an existing GPO has been applied to my new server and is having a negative effect on it. Let's take a look inside Group Policy at the easiest way to display the settings that are contained within each GPO.

## Getting ready

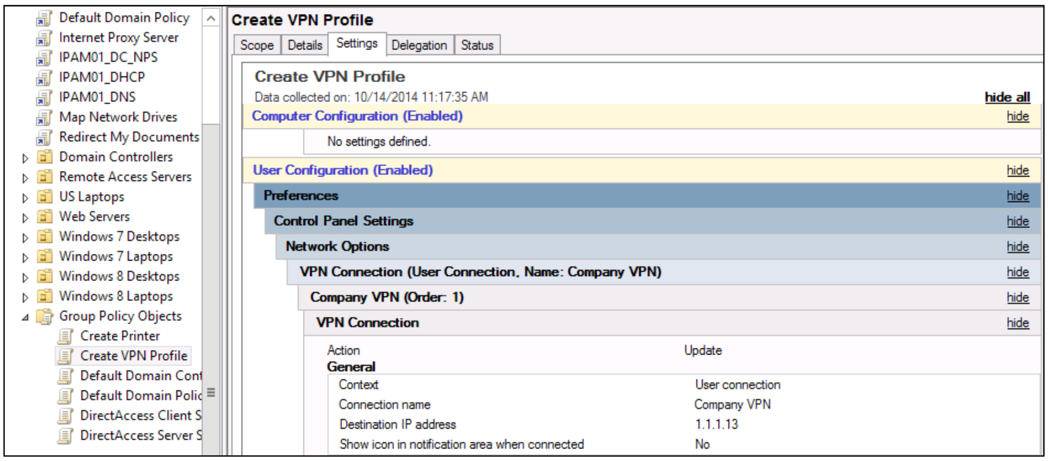
For this recipe, we only need access to the Group Policy Management Console, which I am going to run from my Server 2012 R2 Domain Controller server.

## How to do it...

To quickly view the settings contained within a GPO:

1. On the Domain Controller, open up **Server Manager** and launch the **Group Policy Management** Console from inside the **Tools** menu.
2. Expand the name of your domain, then expand the **Group Policy Objects** folder. This displays all of the GPOs currently configured in your domain.
3. Click on one of the GPOs, so that you see the **Links** and **Security Filtering** sections in the right window pane.
4. Now click on the **Settings** tab near the top.

5. Once you have **Settings** tab open, click on the **show all** link near the top right. This will display all of the settings that are currently configured inside that GPO:




## How it works...

In this very simple recipe, we use the Group Policy Management Console in order to view the currently configured settings inside our Group Policy Objects. This can be very useful for checking over existing settings and for comparing them against what is actually being configured on the client computers. Taking a look through this information can also help you to spot potential problems such as duplicate settings spread across multiple GPOs.

## See also

Viewing the settings included in a GPO can be helpful during troubleshooting, but there are many other tools that can be additionally used in order to troubleshoot Group Policy. Here are a couple of links to help you understand the recommended procedures for troubleshooting Group Policy:

- `http://technet.microsoft.com/en-us/library/jj134223.aspx`
- `http://technet.microsoft.com/en-us/library/cc749336%28v=ws.10%29.aspx`

# Viewing the GPOs currently assigned to a computer

Once you start using Group Policy to distribute settings around to many client computers, it will quickly become important to be able to view the settings and policies that have or have not been applied to specific computers. Thankfully there is a command built right into the Windows operating system to display this information. There are a number of different switches that can be used with this command, so let's explore some of the most common ones that I see used by server administrators.

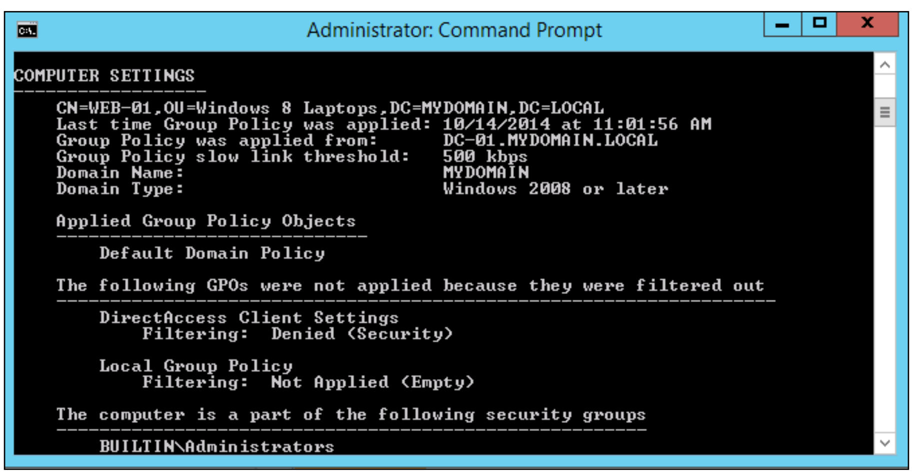
## Getting ready

We have a number of GPOs in our domain now; some are applied at the top level of the domain and some are applied to only specific OUs. We are going to run some commands on our Server 2012 R2 web server in order to find out which GPOs have been applied to it and which ones have not.

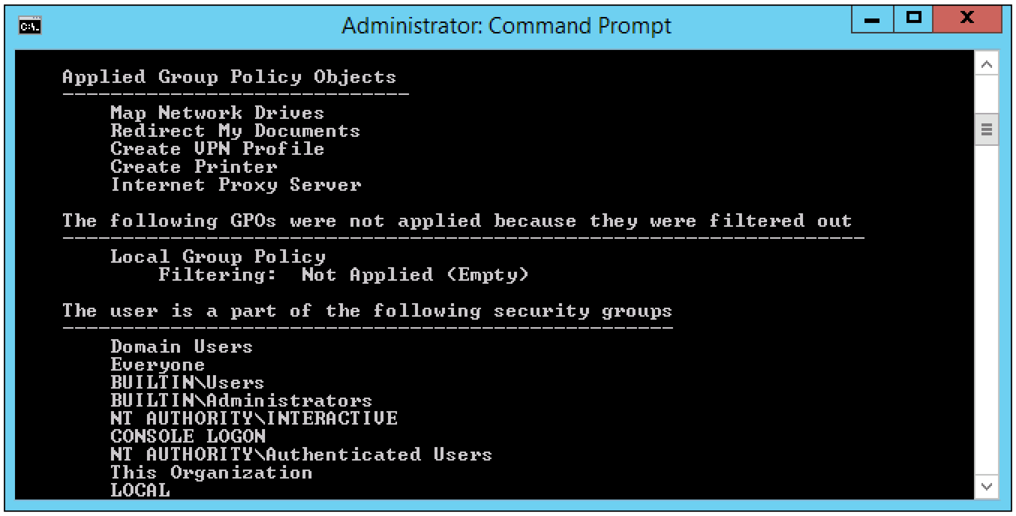
## How to do it...

Let's use the `gpresult` command to gather some information on policies applied to our server:

1. Log in to the web server, or any client computer on which you want to see these results, and open up an administrative command prompt.
2. Type `gpresult /r` and press *Enter*. This displays all of the resultant data on which policies are applied, and are not applied, to our system. You can scroll through this information to get the data that you need.



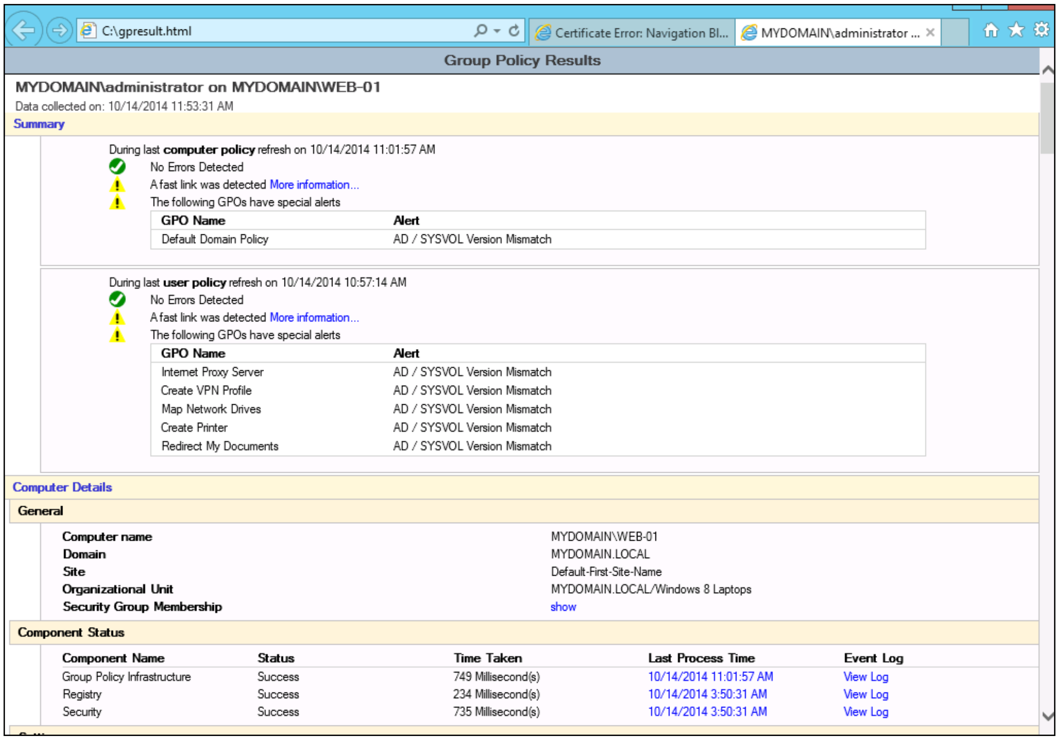
3. Now let's clean that data up a little bit. For instance, the general output we just received had information about both the Computer policies as well as the User policies. Now we want to display only policies that have applied at the User level. Go ahead and use this command: `gpresult /r /scope:user`.




[ You can use either the `/SCOPE:USER` switch or the `/SCOPE:COMPUTER` switch in order to view specifically the User or Computer policies applied to the system. ]

4. And if you aren't a huge fan of looking at this data via a command prompt, never fear! There is another switch that can be used to export this data out to HTML format. Try the following command: `gpresult /h c:\gpresult.html`.

5. After running that command, browse to your `C:` drive and you should have a file sitting there called `gpresult.html`. Go ahead and open that file to see your gpresult data in a web browser, with a slightly nicer look and feel.




## How it works...

The `gpresult` command can be used in a variety of ways to display information about which Group Policy Objects and settings have been applied to your client computer or server. This can be especially useful when trying to determine what policies are being applied, and maybe even more helpful when trying to figure out why a particular policy hasn't been applied. If a policy is denied because of rights or permissions, you will see it in this output. This likely indicates that you have something to adjust in your Links or Security Filtering in order to get the policy applied successfully to your machine. However you decide to make use of the data for yourself, make sure to play around with the `gpresult` command and get familiar with its results if you intend to administer your environment using Group Policy.

One additional note about another command that is very commonly used in the field. Windows domain joined machines only process Group Policy settings every once in a while; by default they will refresh their settings and look for new policy changes every 90 minutes. If you are creating or changing policies and notice that they have not yet applied to your endpoint computers, you could hang out for a couple of hours and wait for those changes to be applied. If you want to speed up that process a little, you can log in to the endpoint client computer, server, or whatever it is that should receive the settings, and use the command `gpupdate /force`. This will force that computer to revisit Group Policy and apply any settings that have been configured for it. When we make changes in the field and don't want a lot of time waiting around for replication to happen naturally, we often use `gpupdate /force` numerous times as we make changes and progress through testing.

## See also

I tend to prefer `gpresult` to view the policies that are currently applied to a computer that I am working on, but it's not the only way. You may also want to check out `RSOP.MSC`. This is a tool that can be launched in order to see a more visually stimulating result of the policies and settings that are currently applied to your computer. Check out the details here:

- `http://technet.microsoft.com/en-us/library/cc772175.aspx`

# Backing up and restoring GPOs

As with any piece of data in your organization, it is a good idea to keep backups of your GPOs. Keeping these backups separately from a full Domain Controller or full Active Directory backup can be advantageous, as it enables a quicker restore of individual GPOs in the event of an accidental deletion. Or perhaps you updated a GPO, but the change you made is now causing problems and you want to roll that policy back to make sure it is configured the way that it was yesterday. Whatever your reason for backing up and restoring GPOs, let's take a look at a couple of ways to accomplish each task. We will use the Group Policy Management Console to perform these functions, as well as figuring out how to do the same backup and restores via PowerShell.

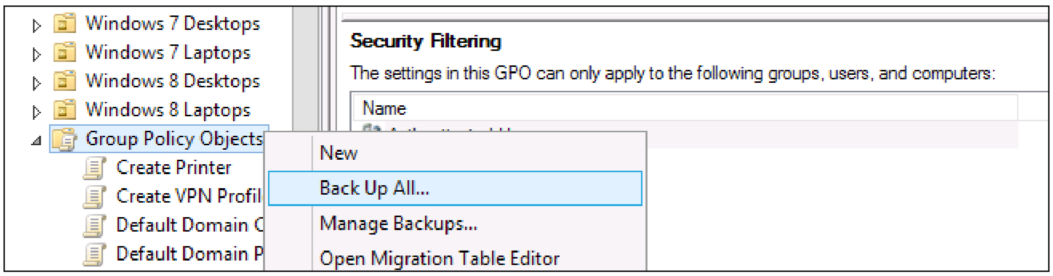
## Getting ready

We are going to perform these tasks from a Windows Server 2012 R2 Domain Controller in our environment. We will utilize both the Group Policy Management Console as well as the PowerShell command line.

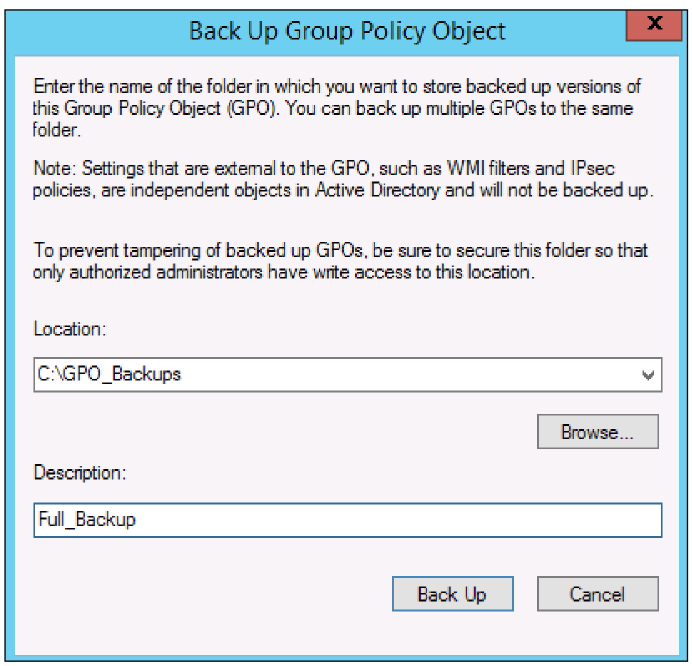
## How to do it...

There is a GPO in our domain called **Map Network Drives**. First we will use Group Policy Management Console to back up and restore this GPO:

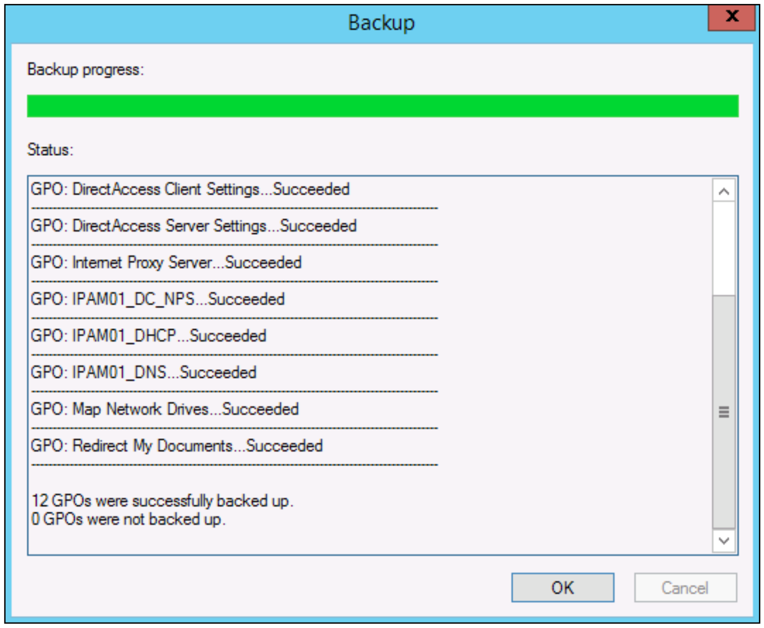
1. From the **Tools** menu of Server Manager, open up the **Group Policy Management** Console.
2. Expand **Forest** | **Domains** | **Your Domain Name** | **Group Policy Objects**.
3. If you want to back up a single GPO, you simply right-click on the specific GPO and choose **Back Up...**. Otherwise, it is probably more useful for us to back up the whole set of GPOs. To accomplish that, right-click on the **Group Policy Objects** folder and then choose **Back Up All...**.




4. Specify a location where you want the backups to be saved and a description for the backup set. Then click **Back Up**:



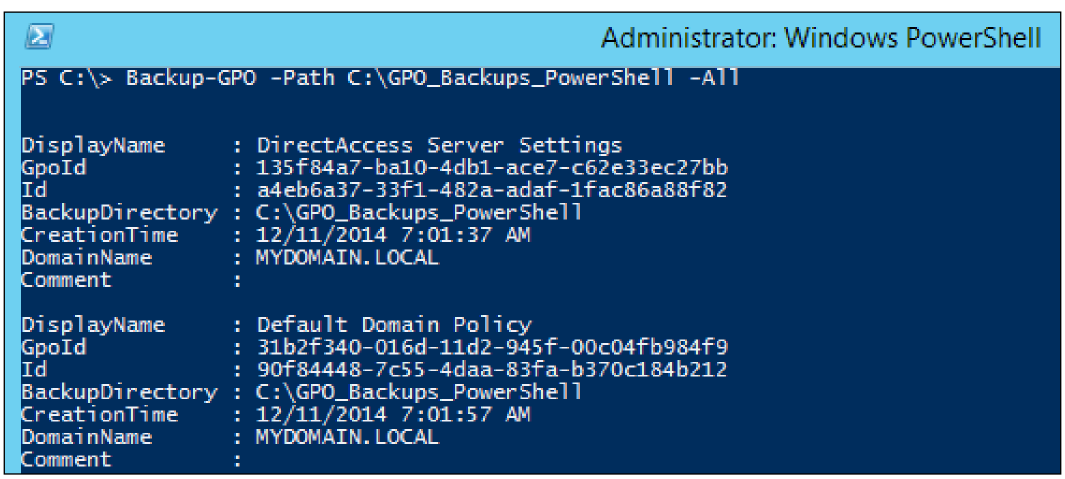
5. Once the backup process is complete, you should see a status of how many GPOs were successfully backed up:




Now let's try accomplishing the same full GPO backup, but this time using PowerShell:

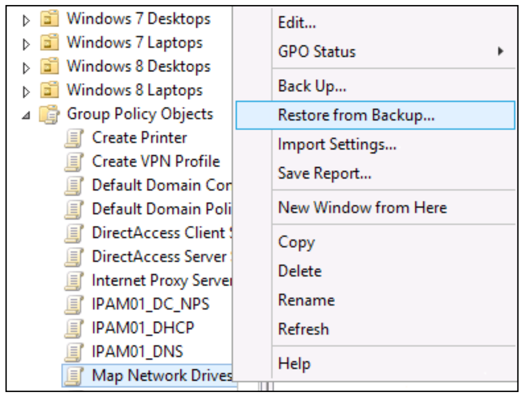
1. Open an administrative PowerShell prompt.
2. Use the following command:

```
Backup-GPO -Path C:\GPO_Backups_PowerShell -All
```

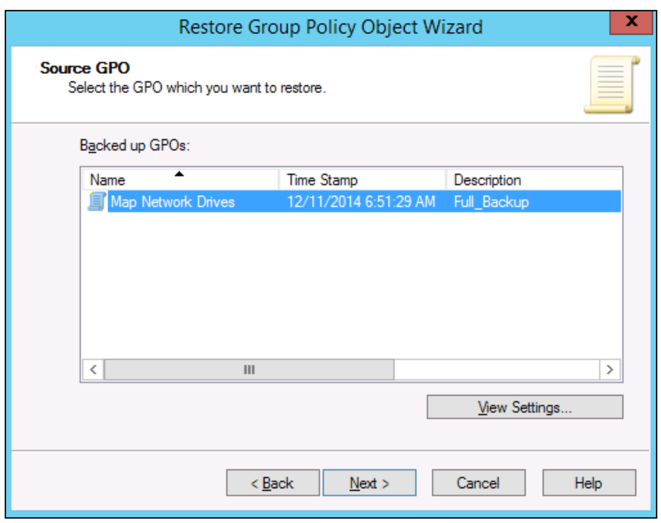


Now that we have two full backup sets of the GPOs, let's try to restore the GPO called **Map Network Drives**.

1. Navigate back inside the **Group Policy Management** Console and find the **Group Policy Objects** folder. The same location that we used to back up a minute ago.
2. Right-click on the **Map Network Drives** GPO and choose **Restore from Backup...**.




3. Click **Next** and specify the folder where your backup files are stored. Then click **Next** again.
4. As long as a backup copy of the Map Network Drives GPO exists in that folder, you will see it in the wizard. Select that GPO and click **Next**.

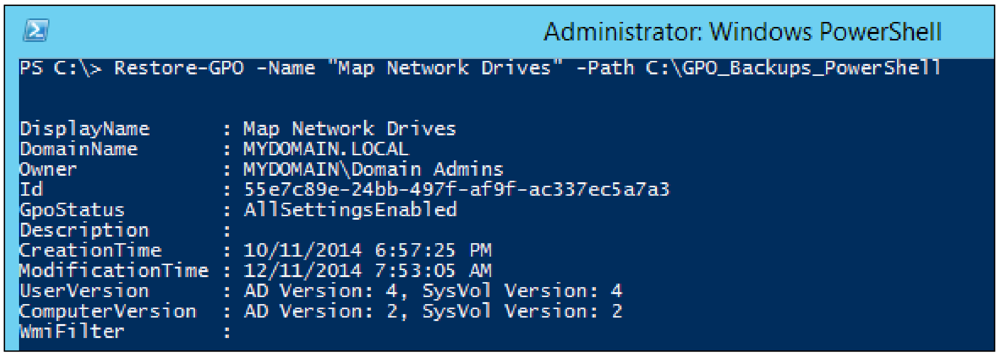



5. Click **Finish** and the GPO will be restored to its previous state.

Now we will restore the same Map Network Drives GPO, but using PowerShell as follows:

1. Head back to your administrative PowerShell prompt.
2. Use the following command to restore the previous version of this GPO, from the backup we created earlier:

```
Restore-GPO -Name "Map Network Drives" -Path C:\GPO_Backups_
PowerShell
```




Rather than typing out the name of the GPO in this command, you could instead specify the GUID of the policy. This number is generally a lot longer than the name, however, and so I tend to see admins preferring to utilize the name of the policy.

## How it works...

Backing up and restoring GPOs is going to be a regular task for anybody administering Active Directory and Group Policy. In this recipe, we walked through each process, using a couple of different tools for each procedure. Group Policy Management Console is nice because it is graphically interfaced, and easy to look at the options available to you. PowerShell is often preferred, however, because it can be automated (think scheduled backups). It also makes remote execution of these commands easy, doing so from another machine inside the network.

## See also

Here are some links for more extensive information about the PowerShell cmdlets we used today:

- `http://technet.microsoft.com/en-us/library/hh967480.aspx`
- `http://technet.microsoft.com/en-us/library/ee461030.aspx`

# 10
# File Services and Data Control

File storage needs exist for any organization of any size. Whether we are talking about simple document storage for your team of users to utilize, or something like block storage accessed over a network by a high-performance computing environment, you are going to have servers in your network that are responsible for storing data safely and securely. The File Services role has grown significantly through the years as our storage needs have changed and evolved. Many of us can no longer satisfy the data needs for our environments with simple file shares and physical disks. Let's use this chapter to explore some of the more interesting ways that data can be managed in a Windows Server 2012 R2 environment:

- Enabling Distributed File System (DFS) and creating a Namespace
- Configuring Distributed File System Replication (DFSR)
- Creating an iSCSI target on your server
- Configuring an iSCSI initiator connection
- Configuring Storage Spaces
- Turning on data deduplication
- Setting up Windows Server 2012 R2 Work Folders

## Introduction

I read last year that the Internet carries more than 1,800 Petabytes of information every day. That is an incredible number! As all of that data comes and goes, it is easy in this day and age to think of this information as being stored in *the cloud*, a magic box in the sky. All of that data is sitting somewhere, though.

On hard drives, installed inside servers, sitting inside datacenters. All this talk about the cloud has really morphed for companies into a debate about private clouds; when you break it down, though, what they are really talking about is different ways to provide a centralized group of information to users that may be accessing it from various places. There are numerous technologies baked into Server 2012 R2 that can assist with the centralization and securing of your data, so let's explore a few of them together.

# Enabling Distributed File System (DFS) and creating a Namespace

**Distributed File System** (**DFS**) is a technology included with Windows Server 2012 R2 that enables multiple file servers to share a single Namespace, enabling end users to access files and folders from a single network name. Those accessing the files don't have to worry about which physical server they are currently in contact with, they simply utilize the namespace of the DFS environment and let the servers do all the grunt work in making sure that all files and folders are available to the users, no matter where those files happen to be physically sitting. Another way to think of it is a collection of network shares, all stuck together under the same umbrella that is the **DFS Namespace**. Users access folders and files via the Namespace, and have access to everything in one place. It helps to think of DFS Namespaces sort of like CNAME records in DNS. They essentially allow us to virtualize the file resources.

Let's work together to get a basic DFS environment up and running, with a single Namespace created so that we can test browsing to it without having to specify the name of our file server. We will also be taking steps during this recipe to prep our DFS server for replication, synchronizing the files between two file servers. The actual configuration of replication (DFSR) will be accomplished in our next recipe, but when we build out FILE-01, we are prepping it to be prepared for that role as well.

## Getting ready

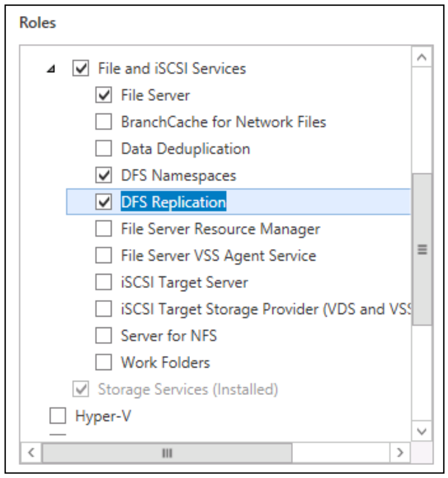
We are working inside a domain environment, and the actual work today will be accomplished from our new file server. This is called FILE-01, and is running Windows Server 2012 R2. It has been joined to the domain, but nothing else has yet been configured.

## How to do it...

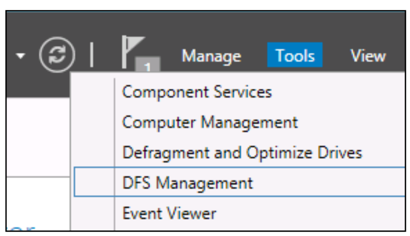
Follow these steps to configure this new file server for DFS:

1. Log in to FILE-01 and launch **Server Manager**. Click on the link **Add roles and features**.
2. Click **Next** a few times until you reach the **Select server roles** screen.
3. Expand **File and Storage Services** | **File and iSCSI Services**.

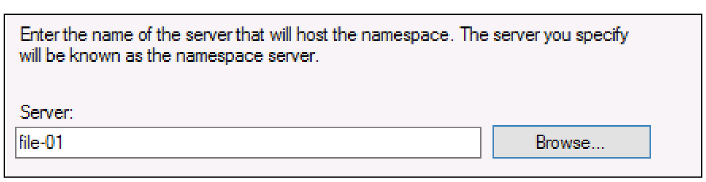
4. Check the box for **DFS Namespaces**. When prompted to install additional required features, click **Add Features**.
5. Also check the box for **DFS Replication**. You should now have **File Server**, **DFS Namespaces**, and **DFS Replication** checked on your select roles screen.



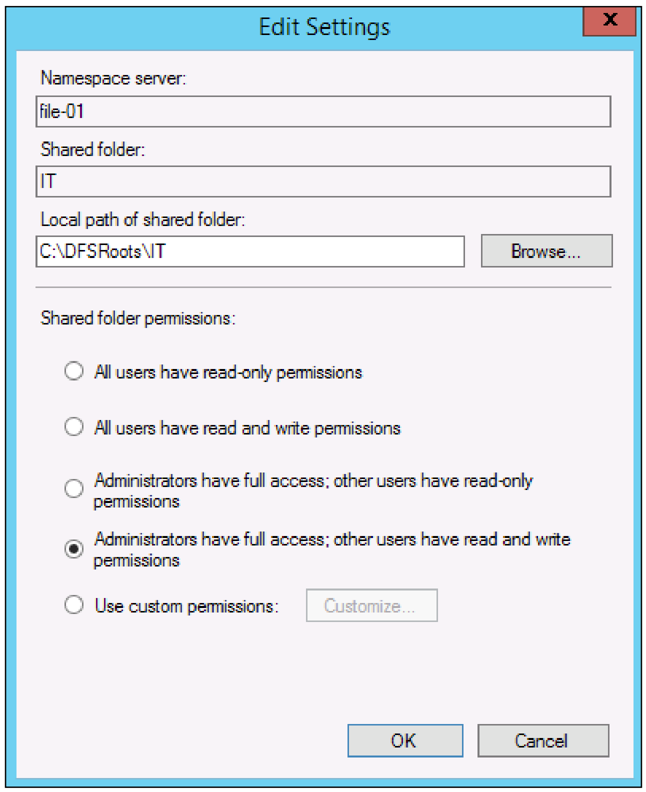

6. Click **Next**, **Next**, then **Install**. This will place the necessary roles onto FILE-01.
7. Now open Server Manager's **Tools** menu and launch **DFS Management**.



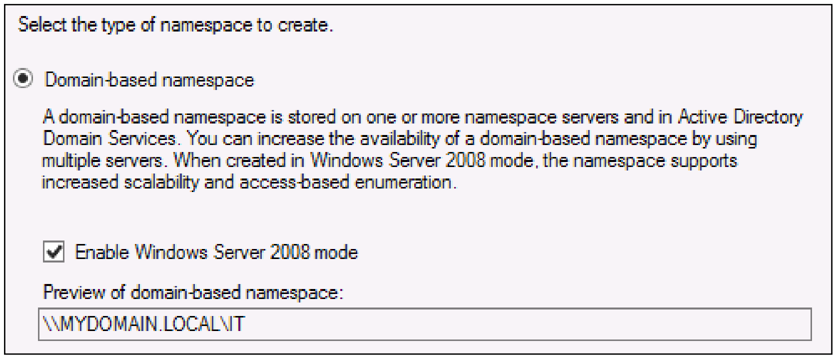

8. First we are going to create a Namespace that will be published in our domain. Right-click on **Namespaces** and choose **New Namespace...**.
9. Enter the name of the server that is going to be your **Namespace server**. We are going to use the primary file server, `file-01`.



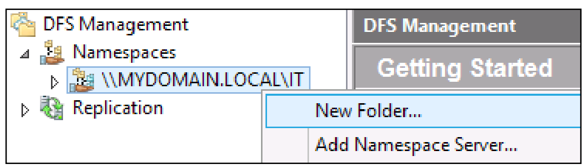
10. On the next screen, **Namespace Name and Settings**, input a name for this new Namespace. My first share is going to be for IT purposes, so I'm calling mine `IT`. Then click on the **Edit Settings...** button.
11. The wizard is going to create a new share to use for my Namespace storage location. If you would like this share to be created in a particular place on the hard drive, specify it here. I am also going to choose the option for **Administrators have full access; other users have read and write permissions** so that users without administrative rights can still save into this Namespace.



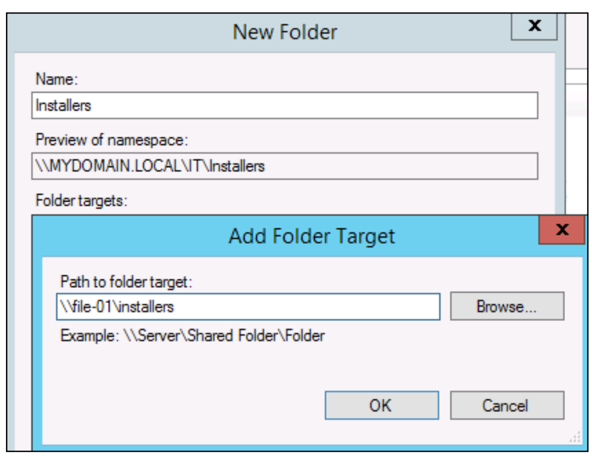

12. Click **OK**, then click **Next**.
13. Typically you want to choose the default options on the **Namespace Type** screen. It should be selected as **Domain-based namespace**, which is great, and have the checkbox enabled for **Enable Windows Server 2008 mode**. You can also see here a preview of the final Namespace name listed for your review. Just go ahead and click **Next** on this screen.



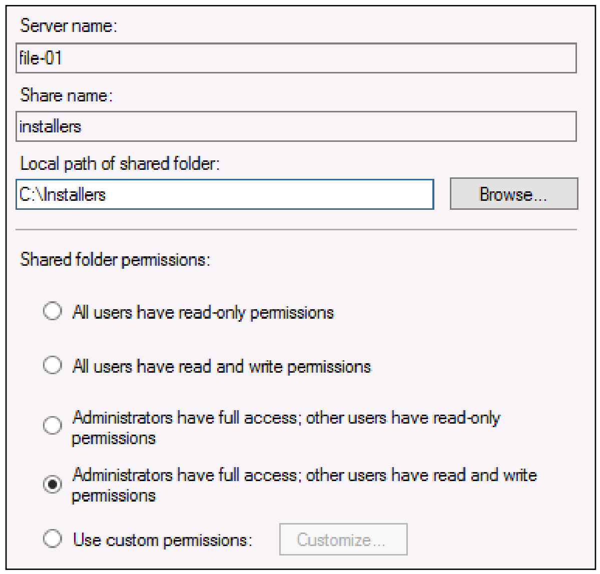

14. Review the settings you have chosen on the final screen and then click the **Create** button in order to create your Namespace.
15. Your new Namespace is now visible in the left window pane of **DFS Management**. Let's go ahead and create a folder inside this Namespace. Right-click on the new Namespace and choose **New Folder...**.



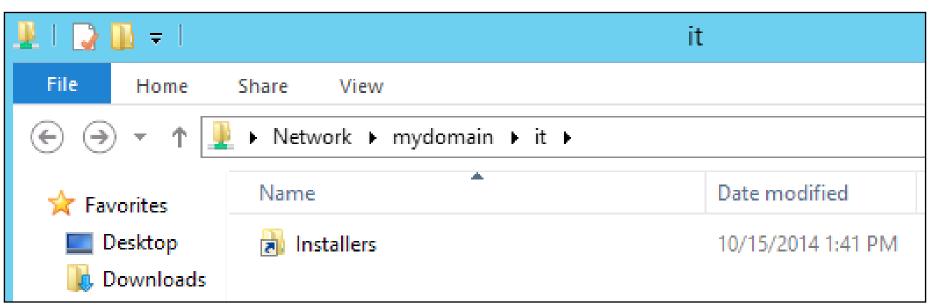

16. Input a name for your new folder in the **Name** field; this is the name that will be displayed inside the DFS Namespace when users access it. Then click the **Add...** button to specify a share that this new folder is going to link to. I am specifying a folder that happens to be sitting here on FILE-01, but you could even specify a network share that is on another server.



17. When you click **OK**, if the share that you entered doesn't already exist, you will be prompted with a question on whether or not you want the wizard to create it for you. I chose **Yes**, so that it could create this new share for me.
18. After choosing **Yes** in order to create the new shared folder, I have another screen that allows me to specify permissions on this new folder. Go ahead and choose the permission setting that is appropriate for the kind of information you are planning to place in this folder. You must also now specify the physical location of this share on the hard drive.




19. Now that we have a DFS Namespace created and a folder within that Namespace, let's test this out! Log in to a client computer and try browsing to `\\mydomain\it`.



## How it works...

In this recipe, we took a new file server and turned it into our first DFS box. The new Namespace that we created now contains a folder where users are able to store documents, and is published on the domain so that these files and folders can be accessed via the DFS Namespace name, with no need to know the name of our specific file server. DFS is a great tool for centralizing data and creating ease-of-use for employees in your company when they need to access their data. It is also a great tool for redundancy via replication, and we'll discuss that in greater depth in just a minute...

# Configuring Distributed File System Replication (DFSR)

**Distributed File System Replication** (**DFSR**) is a piece of DFS that enables automatic file replication between multiple servers. In our first recipe in this chapter, we added the roles and created a DFS Namespace, so we have access into files and folders that are sitting within our DFS environment. So far, though, it is all sitting on a single file server. Follow along to enable the "R" part of DFSR, Replication. We will set up DFSR between the two file servers in our environment, FILE-01 and FILE-02, and test it to make sure that data is being synchronized between the two.

## Getting ready

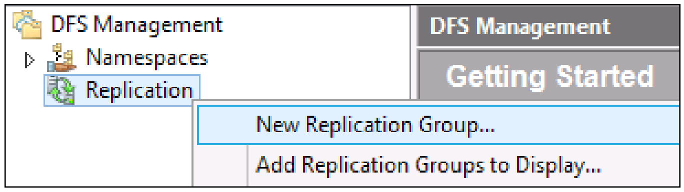
We already have a DFS server online, FILE-01. It is hosting a DFS Namespace with a folder inside. A new file server, FILE-02, is online and joined to the domain. This recipe expects that you have already installed the necessary roles for using this server with DFS. The procedure for installing these roles is outlined in our previous recipe, *Enabling Distributed File System (DFS) and creating a Namespace*. Add the roles to the new FILE-02 in exactly the same way that you did for FILE-01, and then continue on with this recipe to configure the replication.

## How to do it...

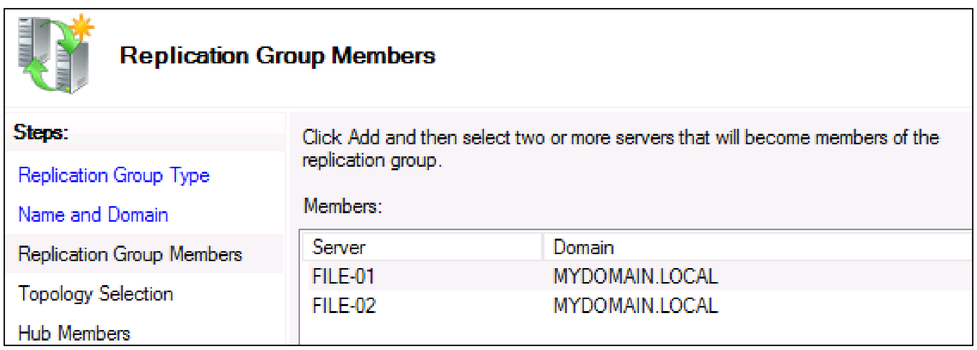
To set up DFSR between the two file servers in our environment, follow these steps:

1. On FILE-01, our primary file server, launch **Server Manager** and then open the **DFS Management** Console from inside the **Tools** menu.

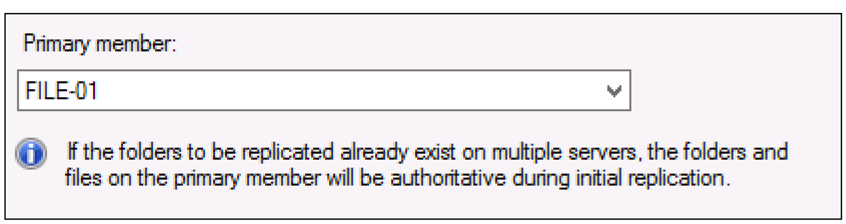
2. In the left window pane, right-click on **Replication** and choose **New Replication Group...**.



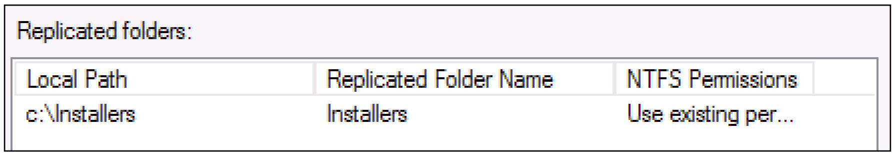

3. Choose **Multipurpose replication group** and click **Next**.
4. Enter a name for your new replication group. Then click **Next**.
5. On the **Replication Group Members** screen, click the **Add...** button and choose both of your file servers that you want to be part of this group.



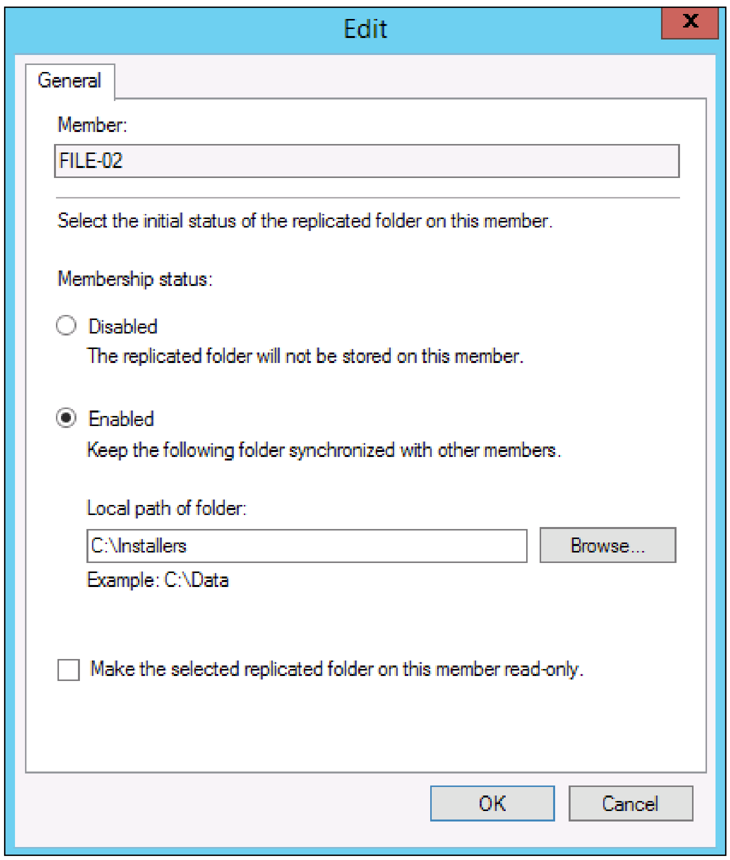

6. Leave the topology set to **Full mesh** and click **Next**.
7. Use the **Bandwidth** screen to throttle the connection if you need to; otherwise, just click **Next** again.
8. Choose the **Primary member** from the list. For our example, it is **FILE-01**.



9. Now use the **Add...** button in order to add all folders that you want to replicate. For our example, I'm going to replicate the new `Installers` folder that we configured inside our DFS Namespace.

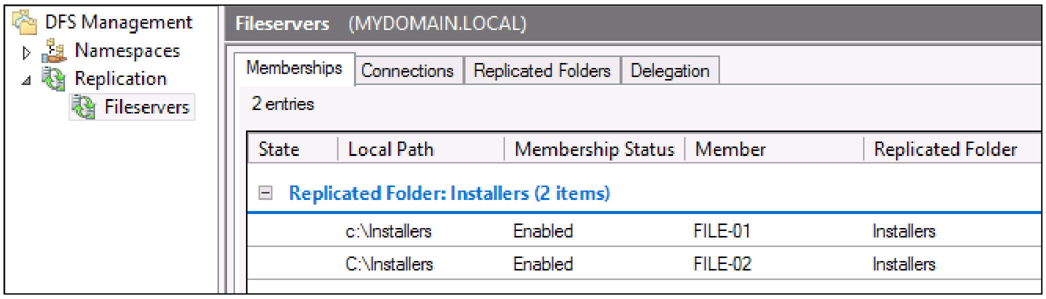



10. You now have to specify the local path for the `Installers` folder to exist on the other member server, **FILE-02**. Click on the **Edit...** button and configure it as follows:

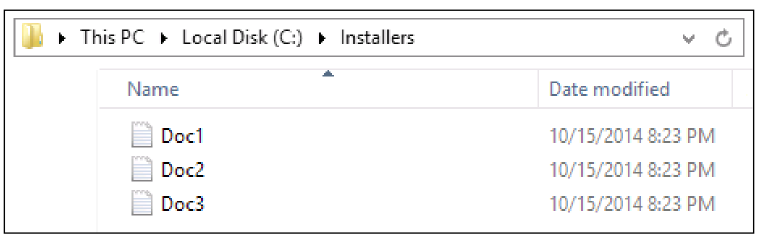



11. Take a look over the summary of settings that are about to be put into place; once satisfied, go ahead and click on the **Create** button.

12. Now back at the main **DFS Management** screen, click on the name of your new replication group that is listed in the left tree. You should see both member servers listed here, indicating that replication is configured!




13. Now let's test this thing out. From a client computer, open up File Explorer and navigate to `\\mydomain\it\installers`.
14. Create a few test files in this folder.
15. Give it a little bit of time for replication to happen, then check inside the `C:\Installers` folder on each file server. You should see that there are copies of your new files now located on both servers' hard drives!




## How it works...

In this recipe, we took our DFS environment and expanded its capabilities a little by adding in replication. DFSR is a great tool to use for distributing files around to your branch offices, while keeping the user experience and drive mappings similar no matter where everyone happens to be accessing the files from. Historically speaking, Microsoft's Distributed File System has a bad reputation, because of some issues there were in older Windows Server operating systems. Those days are gone, and if you haven't tried out this technology in your own environment yet, you've got no reason to wait!

# Creating an iSCSI target on your server

iSCSI is another way to share storage across a network. Well, the term iSCSI itself has more to do with the actual protocol level and the way that the data is transported across the LAN or WAN, but what it looks like as a consumer of iSCSI is that a machine has a drive letter for a disk, although that "disk" is not physically connected to the server. For example, you might log in to a server and see an `M` drive. This drive looks just like a local volume, but it is actually a network connection to storage that might be sitting on the other side of the datacenter. Sounds like a mapped network drive, right? Yes, but it works on a lower level. iSCSI virtual disks, as they are called, work with the server as if they are local disks. This gives servers the ability to interface with this data at a system level, and does not require a user context in order to work, like mapped network drives do. This is commonly referred to as **block storage**.

One good example that I worked with was a database application that a customer was installing onto a new server. The requirements for installing this software insisted that a drive was to be dedicated for storage; it had to be a full drive letter on the system and could not be a mapped drive or a UNC mapping. We were not able to add another hard drive to the physical server, and that wasn't really desirable anyway. We utilized iSCSI to create an iSCSI target on their main storage server, and then connected to that block of storage with an iSCSI initiator on the application server where we were installing the software.

I haven't seen a lot of places utilize iSCSI, which is exactly why I thought we should test the waters with it here. We now have the option in Server 2012 R2 to create our own iSCSI targets right on the server, so let's work on creating one of these targets together.

## Getting ready

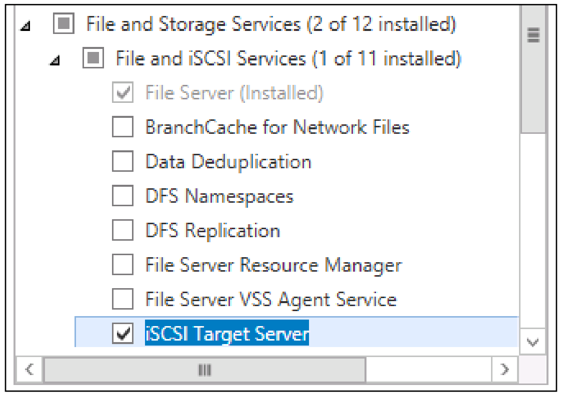
We have a Windows Server 2012 R2 running that we are going to prep to be our iSCSI target server.

## How to do it...

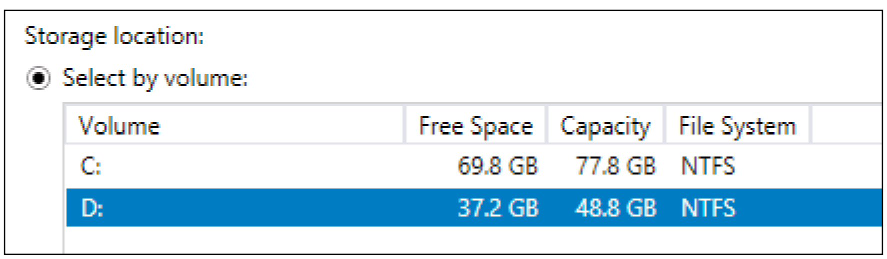
To create an iSCSI target on your server go through the following steps:

1. Open **Server Manager** and click on the link **Add roles and features**.
2. Click **Next** until you come to the **Select server roles** screen.

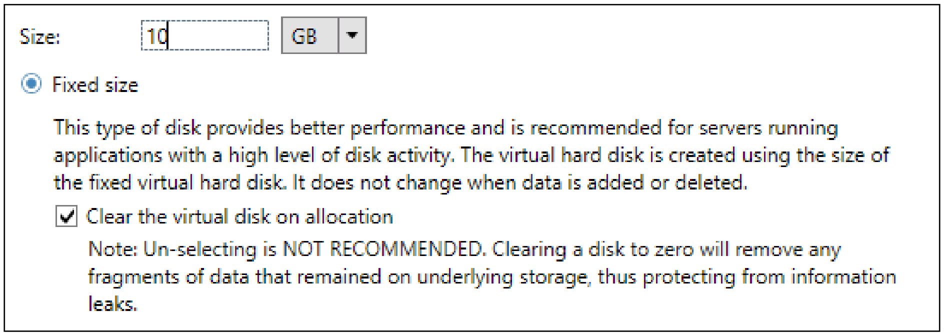
3. Expand **File and Storage Services** | **File and iSCSI Services**, and check the box next to **iSCSI Target Server**.



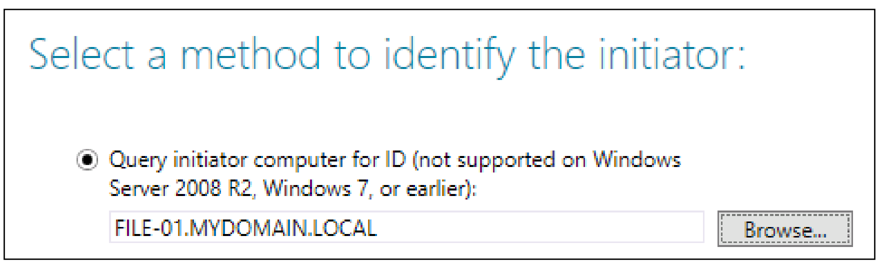

4. Click **Next**, **Next**, and **Install** to finish putting this new role into place.
5. In the left pane of Server Manager, click on **File and Storage Services**. Then click on **iSCSI**.
6. From the **Tasks** menu located in the far right corner, click on **New iSCSI Virtual Disk...**.
7. Choose a location for this iSCSI target to reside. I am going to utilize my `D` volume for this storage.




8. On the next screen, specify a name for your iSCSI virtual disk. You can see that it is going to create and utilize a `VHDX` file for this storage. Nice!
9. Now we specify the **iSCSI Virtual Disk Size**. Read over the text on this screen so that you understand the different types of disks and sizing available to you. For our recipe, we are setting up a **Fixed size** disk with a size of `10` GB.




10. Since this is our first iSCSI target on this server, the **New iSCSI target** option should be selected for you. Click **Next**.
11. Create a name for the iSCSI Target. This is the name that you will use on the iSCSI initiator server later in order to connect to the storage. I am calling mine `Database1`.
12. Click the **Add...** button in order to specify which initiators will later connect to this target. We are going to connect to this storage from a server called FILE-01, so I am adding that server into the list here.




13. When you add a server to this list and click **OK**, you will notice that it is specified on the **Access Servers** screen with an **iSCSI Qualified Name (IQN)** value that you did not specify. This is a unique identifier for the server to the iSCSI environment and is normal behavior. Go ahead and click **Next**.
14. If you would like to utilize CHAP or Reverse CHAP in order to authenticate connections between initiator and target, you can use the next screen to specify user names and passwords for those authentications. For the purpose of testing this out quickly and in a simple manner, we are not setting anything on this page and are only clicking **Next**.
15. Review the settings and click the **Create** button to finish setting up your iSCSI target. This server is now running as a target and is waiting for a connection to it from our iSCSI initiator server.

## How it works...

In this recipe, we are starting to figure out how iSCSI might benefit our environment. In addition to the scenario I discussed where our database server requires a constant drive letter connection to remote storage, there are some other common use cases for iSCSI connections. You could use iSCSI connections in order to consolidate storage. For example, take multiple servers that have locally attached storage and map them to iSCSI storage blocks. You could then move the physical storage to your iSCSI target for all application servers involved. They would still access the same data, and the applications running on those servers wouldn't know any different, but the physical storage would now be consolidated into a centralized area for safekeeping and better data management.

iSCSI is also an interesting use case for diskless booting. You could equip diskless computers with NICs that are iSCSI-ready, and those computers could boot over the network, over iSCSI, to virtual disks that are sitting on the iSCSI target.

## See also

Check out the following links for some more great information on iSCSI:

- `http://technet.microsoft.com/en-us/library/hh848272.aspx`
- `http://technet.microsoft.com/en-us/library/dn305893.aspx`

# Configuring an iSCSI initiator connection

Turning on your first iSCSI target is great, but so far you aren't using that storage for anything. Let's take it a step further and connect a server to that storage so that it can start to be used. The device connecting to an iSCSI target is called an iSCSI initiator. We are going to take a file server in our environment and configure it to connect over the network using iSCSI to our target server. When finished, we will have a new hard disk "attached" to our server, even though it is really just block storage from the iSCSI target that is being accessed via the network.

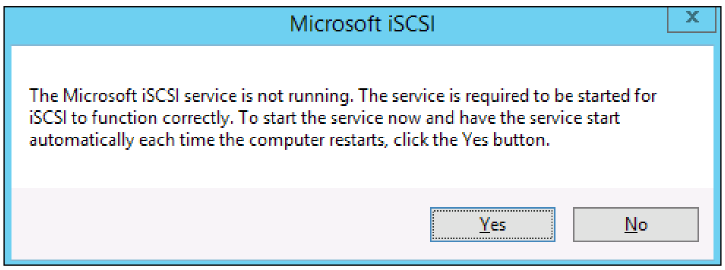
## Getting ready

We have already configured one Windows Server 2012 R2 to be an iSCSI target, and are now configuring a second 2012 R2 box as our iSCSI initiator that will be connecting to the target.

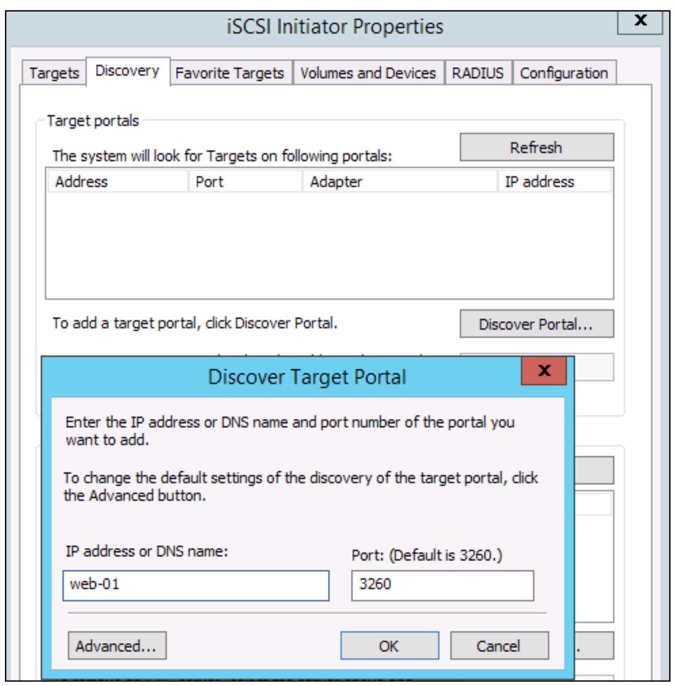
## How to do it...

Follow along to create the iSCSI initiator connection on our FILE-01 server:

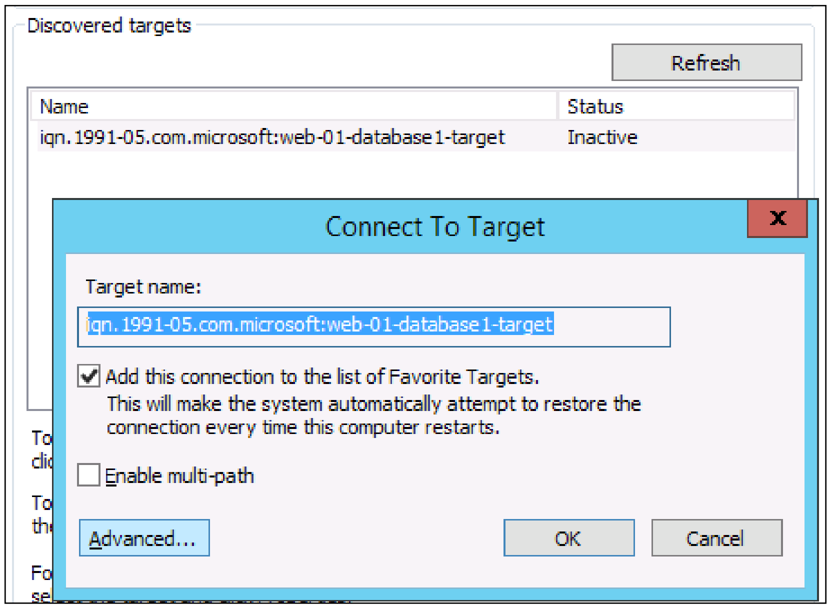
1. Launch **Server Manager**. Open up the **Tools** menu and choose **iSCSI Initiator**.
2. If you have never tried using iSCSI on this machine before, you will receive a message that the Microsoft iSCSI service is not running. To start the service and make sure it continues to start on subsequent boots, click **Yes**.




3. Currently there is nothing listed in the **Targets** tab, which opens by default. Move over to the **Discovery** tab and click on the **Discover Portal...** button.
4. Type in the name of the server where you have an iSCSI target running and click **OK**. Now move back over to the **Targets** tab of the **iSCSI Initiator Properties** screen.



5. The iSCSI connection is now shown on the **Targets** tab by its IQN number. Currently, the status is set to **Inactive**. Select this connection and click on the **Connect** button.




6. Click **OK** to finish connecting to this iSCSI target. Make sure to leave the checkbox enabled for **Add this connection to the list of Favorite Targets** so that the connection is persistent and reconnects following server reboots.

## How it works...

We have now connected our iSCSI initiator to our iSCSI target; if you open any of the normal hard disk management tools such as Disk Management on your initiator server, you will see the new "disk" listed and available! You can then manipulate this storage as you would with any other physical storage, including turning it into a permanent drive letter available to the operating system.

It is important to note that an iSCSI initiator is often used without a Windows Server 2012 R2 iSCSI target server being the other end of the connection. One of the great things about iSCSI is that it doesn't care about what kind of storage you are connecting to, as long as that storage supports being accessed via iSCSI. There are many SAN technologies that you can acquire, or may already have running in your environment, that you can tap into by using the iSCSI initiator on your Windows Server. This gives you the ability to consume storage from the non-Windows SAN device on your Windows application and file servers.

# Configuring Storage Spaces

**Storage Spaces** is an incredibly cool technology that isn't flaunted or marketed on its own accord; it just does its job and does it well. How many times have you caught yourself stuck between a rock and a hard place because you are running out of room on a single hard drive that is in one of your servers? I have plenty of times, especially working with technologies such as RDS that may contain a lot of user data all stored on the system drive. In most current server hardware, it is easy to add multiple hard drives, but not always easy to decide how to partition and volume those drives so you don't run out of space on `C:` while having 200 GB of free space on `D:`.

These kinds of situations are where Storage Spaces can save a lot of time and headache. What if you didn't have to worry about what size hard disks you were running as your primary drive, secondary drive, and so on? What if you could lump them all together and utilize the storage out of one big bucket, or pool, as you will. This is exactly what we can accomplish with Storage Spaces in Windows Server 2012 R2. You combine multiple physical hard disks into a storage pool, and then within that pool you can create one or many volumes to consume that storage space. The multiple disks combine storage together to behave as one large drive, with options for RAID-style redundancy built into the storage pool configuration. Let's work together to combine a few hard drives and create a new single volume to be used by the operating system.

## Getting ready

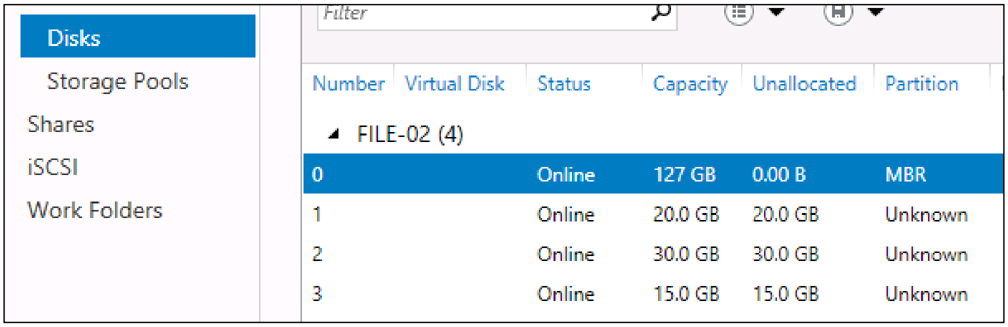
We are going to configure Storage Spaces on our FILE-02 server, which is running Windows Server 2012 R2.

## How to do it...

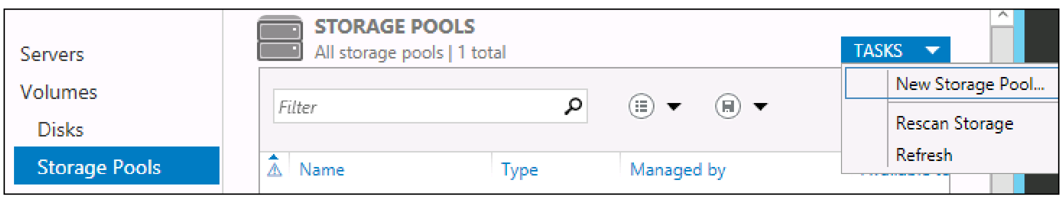
To enable Storage Spaces on your server use the following steps:

1. Make sure you have hard drives connected that you intend to utilize for your storage pool. On our FILE-02 server, I have added three new drives, all of various sizes.
2. Launch **Server Manager** and click **File and Storage Services** from the left window pane.

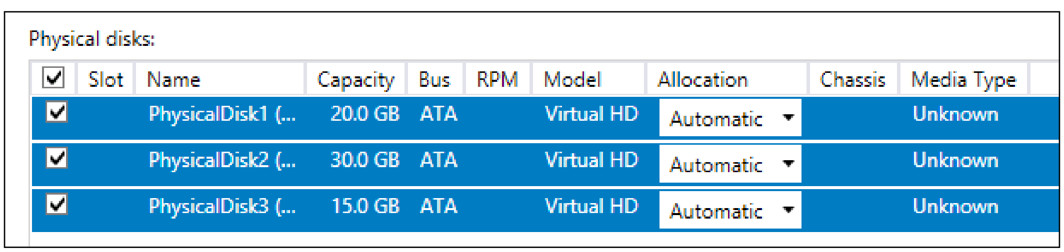
3. First click on **Disks** and make sure that we can see the new drives that we are going to combine together to make a storage pool.



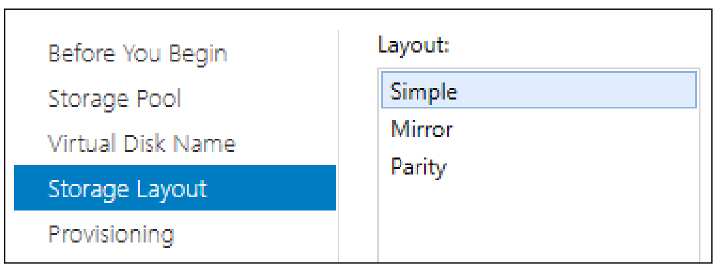

4. Now that we have confirmed our disks are visible within Windows, go ahead and click on **Storage Pools**.
5. Open the **Tasks** menu and choose **New Storage Pool...**.



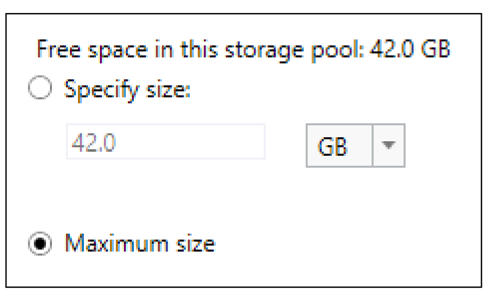

6. Enter something in **Storage Pool Name** and click **Next**.
7. Select the physical disks that you want to be included in this pool. I am going to add all three of the unused drives on my system.



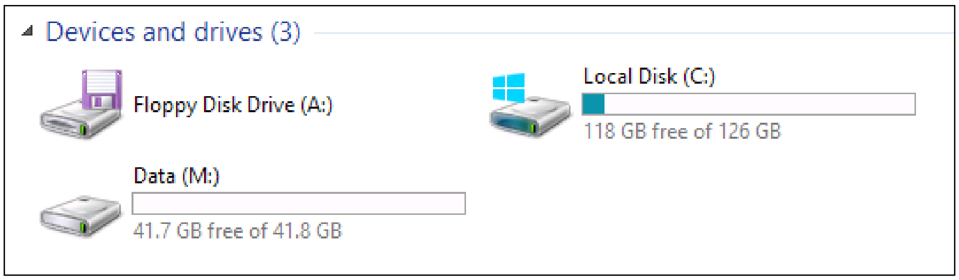
8. Click **Next**, then click **Create**. Once finished building, you will be taken back into the **Storage Pools** section of Server Manager where you can now see the new pool listed. The disks are now grouped together in a pool, but are not yet usable by the operating system.
9. Click on the name of your new storage pool in order to select it.
10. Now down in the **Virtual Disks** section, drop-down the **Tasks** menu and choose **New Virtual Disk...**.
11. Select the storage pool from which you want to create a volume and give it a virtual disk name.
12. On the **Storage Layout** screen, you need to choose the method that will be used for storing data on this new virtual disk. Depending on how many physical disks you have in the pool, you may have different options here. For our example, we need as much data storage space as possible and are not worried about redundancy across disks. So I will choose **Simple**.




13. I am going to dedicate the full amount of this storage space to the virtual disk right away, so I will leave the **Provisioning** screen selected for fixed provisioning.
14. On the **Size** screen, it will indicate how much free space exists in the pool. Simply size your new virtual disk to a number that is equal to or below that number. I am going to consume the full space of the pool, so I can either enter 42 GB as indicated or choose the radio button for maximum size.



15. Finish up the wizard and your new virtual disk will be created! When you finish this virtual disk creation wizard, you will be automatically placed inside **New Volume Wizard**. Walk through these steps or utilize a regular tool such as Disk Management in order to create your new volume, format it, and assign it a new driver letter. Then you can start using the new volume as you would with any regular volume inside Windows.




## How it works...

By combining these three small hard drives into one storage pool, we can build a single volume that is larger than any one of the hard drives on their own. Storage Spaces can be used like this, or in a myriad of other ways, creating multiple pools and volumes at will, simply by bundling together groups of physical drives on the system.

Hard disk space utilization is something that we have traditionally planned very hard for. What size drives to get? Do they need to match? How large do each of my volumes need to be? Should I use RAID? Should I use dedicated hardware for that RAID? Storage Spaces is a way to bring many of those questions together, package them up, and throw them in the trash can.

## See also

If you're interested in Storage Spaces, make sure to check into Storage tiers as well. On a Server 2012 R2, if you combine SSD drives with regular mechanical drives, you can create a storage tier, which is sort of like those hybrid hard drives that come in laptops occasionally. The drives are combined into a single storage unit, but Windows will keep the most commonly accessed items on the SSD, making them faster to access. Here's a link to get you started with looking further into using storage tiers with storage spaces:

- `http://technet.microsoft.com/en-us/library/dn387076.aspx#bkmk_tiers`

# Turning on data deduplication

Deduplication is something that we as people do naturally. Every once in a while you clean out the refrigerator, right? And if there are seven half-empty bottles of ketchup, you probably deduplicate that and throw some away. Or your closet. If you dig around and find thirty blue shirts, chances are that you can part with a few to save some space. These things make common sense, and so does deduplication when talking about the data that is stored on our servers.

Starting with Windows Server 2012, data deduplication became possible at the file system level. When enabled, Windows runs optimization jobs that search out for duplicate files and data, and consolidates them. If you have two of the same file, stored in two different locations, all that is doing is consuming extra hard disk space. Data deduplication removes the secondary copy and utilizes the primary whenever that file is called for from either location on the disk.

New in Server 2012 R2 is the ability to extend this deduplication into Hyper-V, specifically for VDI type deployments. This is huge! Think about all of the different VDI systems that are going to be spun up by that system. With so many similar systems running under the same drive context, there is potential to have thousands of duplicated files, and all duplicated numerous times. In this recipe we are going to walk through the steps to enable data deduplication on a server so that you can start trying this out in your own environments.

## Getting ready

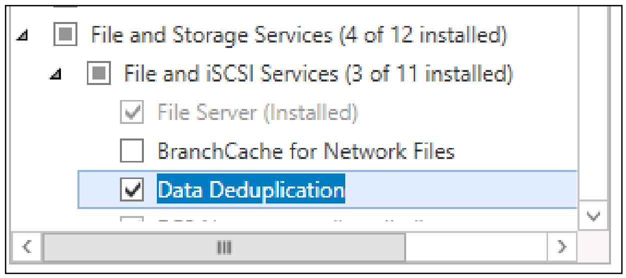
We will be enabling data deduplication on a single server for this recipe, running Windows Server 2012 R2, of course.

## How to do it...

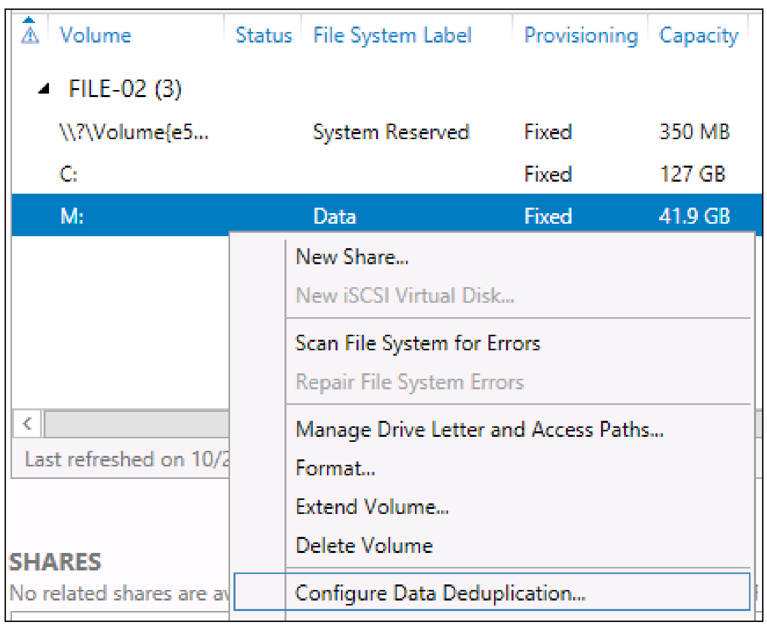
To enable data deduplication on our server follow these steps:

1. Open up **Server Manager** and click on your link **Add roles and features**.
2. Click **Next** until you get to the **Select server roles** screen.

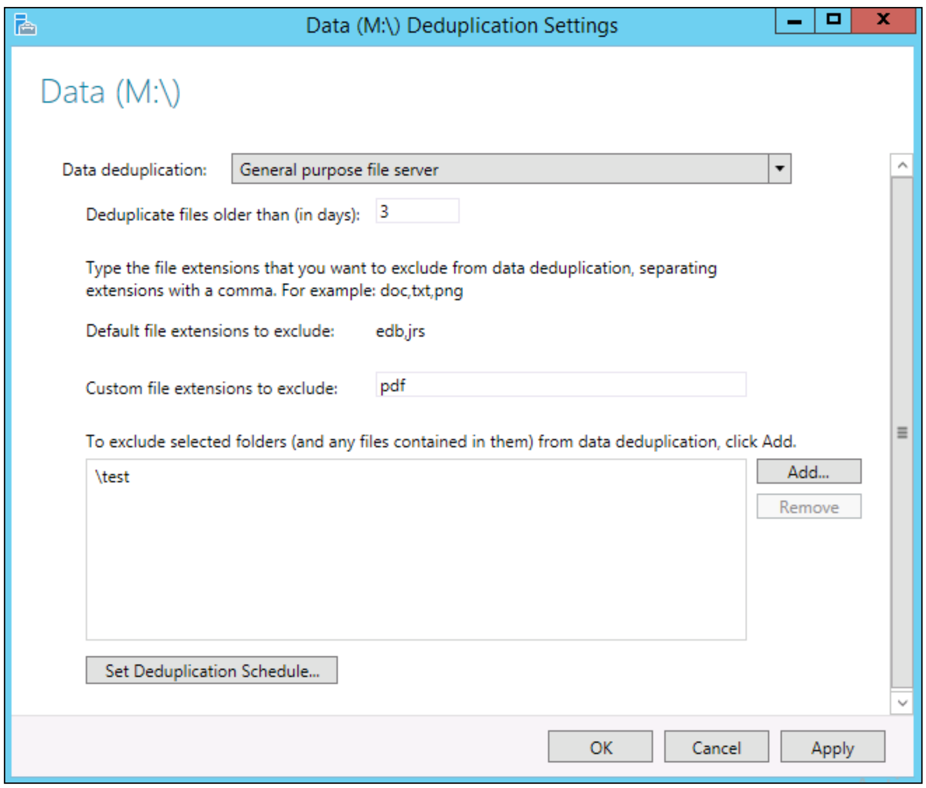
3. Expand **File and Storage Services** | **File and iSCSI Services** and check the box next to **Data Deduplication**.




4. Finish out the wizard in order to complete the installation of the deduplication role.
5. Now, in the left pane of Server Manager, click on **File and Storage Services**.
6. Click on **Volumes**.
7. Right-click on a data volume and choose **Configure Data Deduplication...**.



8. Click on the **Data deduplication** drop-down box and specify whether you are intending to run deduplication on a general purpose file server or Virtual Desktop Infrastructure (VDI) server. If you test out selecting one or the other, you will notice that the default list of file extensions to exclude from deduplication changes automatically. These are the file types that Microsoft has determined need to be excluded from deduplication in order for it to run effectively.
9. If there are any specific files or folders that you want the deduplication process to leave alone, you can specify them here as exclusions. There is also a button named **Set Deduplication Schedule...** where you can specify the times of day that the optimization jobs run to consolidate the data.



## How it works...

Data deduplication is very easy to enable, but can be a powerful tool for saving disk space on your file servers. There is a graph available on one of the links listed below that portrays some numbers on percentages of space savings for different kinds of data. These numbers are quite a bit larger than I expected to see, around 50 percent for general file shares and over 80 percent for VHD libraries! Try data deduplication on some of your own systems and watch your available disk space start to increase.

## See also

Check out the following links for additional information on data deduplication in Windows Server 2012 R2:

- `http://technet.microsoft.com/en-us/library/hh831434.aspx`
- `http://blogs.technet.com/b/filecab/archive/2013/07/31/extending-data-deduplication-to-new-workloads-in-windows-server-2012-r2.aspx`

# Setting up Windows Server 2012 R2 Work Folders

Accessing data from wherever you happen to be is becoming more and more important with today's mobile workforce. Given that, it makes sense that more and more technologies are being designed that allow for the accessing of this data from more locations, and more device types. This is what Work Folders in Windows Server 2012 R2 is all about. It is a way to publish access to files and folders to multiple device types that the users may be logging in to. These files are accessed via a web listener that is configured on the Work Folders file server, which enables this data to be accessed from inside or outside the corporate network, from both domain joined and non-domain joined systems.

Configuring a full-fledged Work Folders environment with all of the moving parts and pieces is far too much data to be contained in a single recipe. Today we will focus on the steps that need to be taken on the file server itself in order to make it ready for hosting Work Folders. Make sure to check out the link provided at the end of this section in order to continue gaining knowledge on this subject. Once you get started with Work Folders and realize the benefits that you can provide, I have no doubt that you will also be tapping into Group Policy in order to roll some of these settings around, and working with a reverse proxy solution such as the **Web Application Proxy** (**WAP**) in order to further enhance the capabilities that Work Folders can bring to the table.

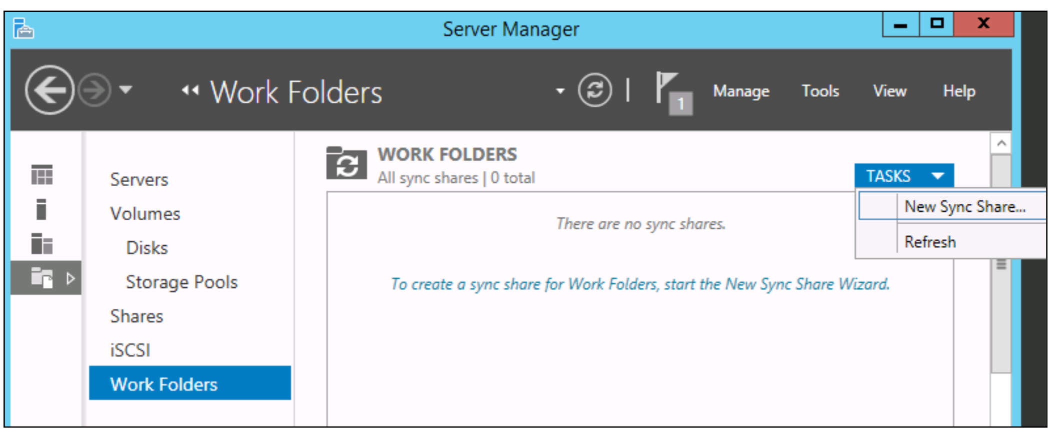
## Getting ready

Our work today is happening on a Windows Server 2012 R2 that we use as a file server. Specifically, I am using the FILE-01 server in the lab that we have been working with throughout this book. To fully configure Work Folders, you will also need the ability to acquire a valid SSL certificate and access to your public DNS environment in order to create a record.

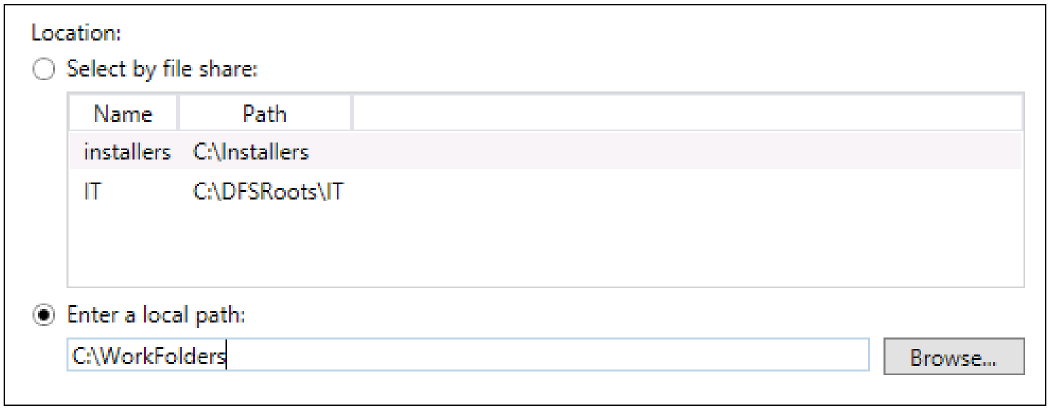
## How to do it...

Follow these steps to enable Work Folders in your environment:

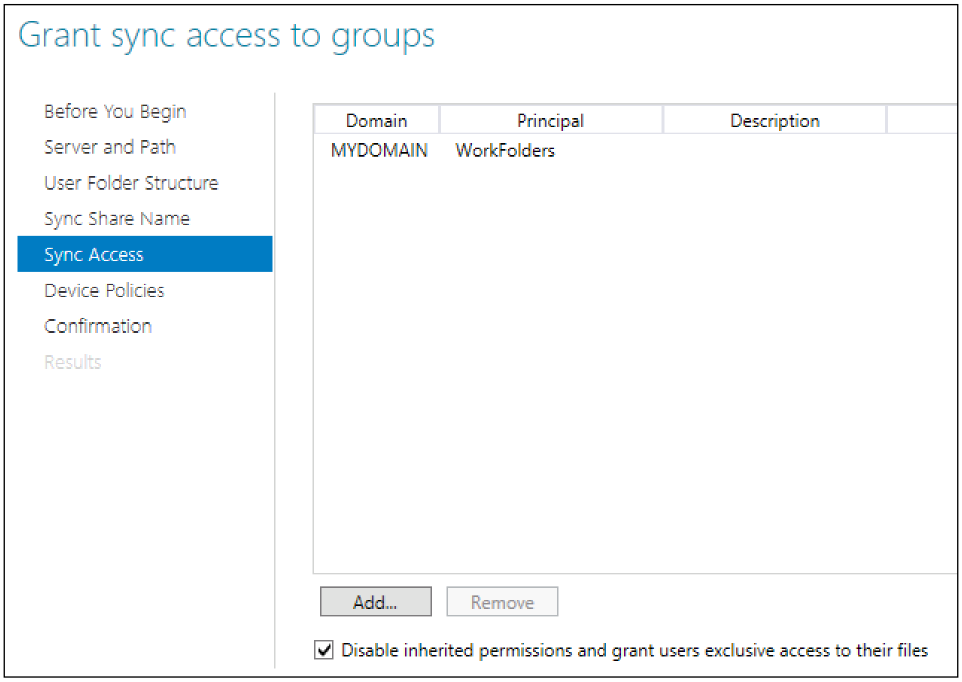
1. Log in to your file server and launch **Server Manager**.
2. Choose the link for **Add roles and features**. Walk through the role installation wizard until you get to the **Select server roles** screen.
3. Expand **File and Storage Services** | **File and iSCSI Services**. Then check the box next to **Work Folders**. When you receive the pop-up message about adding the additional IIS feature that is required, make sure to click on the **Add Features** button.
4. Finish out the wizard in order to install the Work Folders role on this server.
5. Once the role has finished installing, head back to Server Manager and navigate to **File and Storage Services** | **Work Folders**.
6. Dropdown the **TASKS** menu, and choose **New Sync Share...**.



7. Choose or enter a path where you want the new Work Folders to be stored. This is the location on our file server that will be populated by folders that are named after our users. If you have already set up a folder and shared it, you will see it in the list to choose from. I have not yet set up any such folder, and so I am going to type in the location where I want the wizard to create a new folder for me.




8. Click **Next** and, if you entered the location of a folder that did not yet exist, you will be prompted with a confirmation box asking whether you want the new folder to be created. Go ahead and click **OK** on that message.
9. On the **User Folder Structure** screen, you choose how the user's folders will be named within our WorkFolders sync share. Each user that utilizes Work Folders will get their own folder setup inside our share. These individual username folders will be named via either their username alone, or by their username@domain. In a lot of environments, you can get away easily enough with only the username alias. If you have users that will be accessing Work Folders from multiple domains, then you have the potential for conflict among usernames and should choose alias@domain. Additionally on this screen, you can specify to sync only a particular subfolder for the users. For example, if you wanted their `Documents` folder to be synced across all of their devices but didn't care about the other folders such as `Pictures` and `Music`, you could specify only `Documents` on the line here.
10. Specify a name for **Sync Share** and click **Next** again.
11. For the **Sync Access** screen of the wizard, we need to define which users and groups have access to use this sync. I created an Active Directory Security Group called **WorkFolders** and placed my users inside that group. So on this screen, I will simply specify my WorkFolders group.




> Note the checkbox near the bottom of this screen. If you leave the box enabled for **Disable inherited permissions...** then users will be granted exclusive rights to each of their folders. This means that even administrators will not have access to these folders. If you would like to change that behavior and let the normal file system inherited rights persist, simply uncheck this box.

12. Click **Next** and then **Create**, and your new WorkFolder Sync Share will be created and ready for use.

    Client devices will connect to Work Folders on this file server via HTTPS. In order to make that happen successfully, we need to configure a DNS record that points at this file server, and an SSL certificate to be bound to the web listener on the server.

13. On your public DNS, set up the name workfolders.<yourdomain> and point it at the IP address that will flow to this file server. For example, the best way to do this is to publish the web listener with a reverse proxy server of some kind; let's say that proxy server is running on the internet IP address 1.1.1.1. You would configure a DNS record for workfolders.contoso.com and point it at 1.1.1.1, then let the reverse proxy server bring that traffic inside the network and submit it to the file server where we have Work Folders running.

14. Install an SSL certificate that contains the appropriate `workfolders.contoso.com` name and bind it to the default web site on the Work Folders server. Since the full IIS Management Console is not installed with the Work Folders role, you can utilize the IIS Management tools from another server in your network in order to bind the certificate onto the default web site. Alternately, you can use the following netsh command in order to bind the certificate to the site:

```
 netsh http add sslcert ipport=<IP address>:443 certhash=<Cert
thumbprint> appid={CE66697B-3AA0-49D1-BDBD-A25C8359FD5D}
certstorename=MY.
```

> Please note that the command above should not be run exactly as shown here. There are variables in this netsh command that you need to adjust to meet your own environment. The IP address of the web server, `certhash` and `appid` need to be adjusted to match your particulars.

Now, WorkFolders is configured and listening on our file server. The next step is to configure our client computers to tap into this WorkFolders sync share. The process for accomplishing this is different depending on what client devices you are connecting, but the starting point for Windows 8.1 machines is **Control Panel** | **System and Security** | **Work Folders**.

## How it works...

It is pretty easy to overlook Work Folders at first glance, thinking it is just another way to access the same data in a similar way to the folder sharing options that we have had around for years. However, looking a little bit deeper shows us that the ability to publish access to files and folders to both domain joined systems as well as non-domain joined systems, working from both the corporate network or from home, can be of enormous advantage. You could utilize Work Folders as a way to grant access to corporate data without needing to issue a company laptop. You could also grant access to file level detail without the need to incorporate some form of VPN, which may give more access to a home computer than you are comfortable with handing out. There are numerous situations where a technology like Work Folders could increase productivity for your users, and the security of information within your IT infrastructure. Make sure to check it out!

## See also

Take a look at the following link for even more detailed information on setting up Work Folders:

- `http://technet.microsoft.com/en-us/library/dn528861.aspx`

# Index

## V



## W

# About Packt Publishing

Packt, pronounced 'packed', published its first book, *Mastering phpMyAdmin for Effective MySQL Management*, in April 2004, and subsequently continued to specialize in publishing highly focused books on specific technologies and solutions.

Our books and publications share the experiences of your fellow IT professionals in adapting and customizing today's systems, applications, and frameworks. Our solution-based books give you the knowledge and power to customize the software and technologies you're using to get the job done. Packt books are more specific and less general than the IT books you have seen in the past. Our unique business model allows us to bring you more focused information, giving you more of what you need to know, and less of what you don't.

Packt is a modern yet unique publishing company that focuses on producing quality, cutting-edge books for communities of developers, administrators, and newbies alike. For more information, please visit our website at `www.PacktPub.com`.

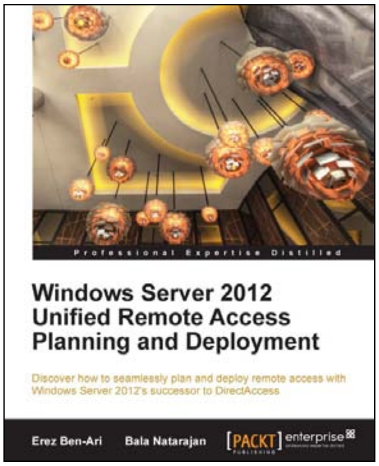
# About Packt Enterprise

In 2010, Packt launched two new brands, Packt Enterprise and Packt Open Source, in order to continue its focus on specialization. This book is part of the Packt Enterprise brand, home to books published on enterprise software – software created by major vendors, including (but not limited to) IBM, Microsoft, and Oracle, often for use in other corporations. Its titles will offer information relevant to a range of users of this software, including administrators, developers, architects, and end users.

# Writing for Packt

We welcome all inquiries from people who are interested in authoring. Book proposals should be sent to `author@packtpub.com`. If your book idea is still at an early stage and you would like to discuss it first before writing a formal book proposal, then please contact us; one of our commissioning editors will get in touch with you.

We're not just looking for published authors; if you have strong technical skills but no writing experience, our experienced editors can help you develop a writing career, or simply get some additional reward for your expertise.



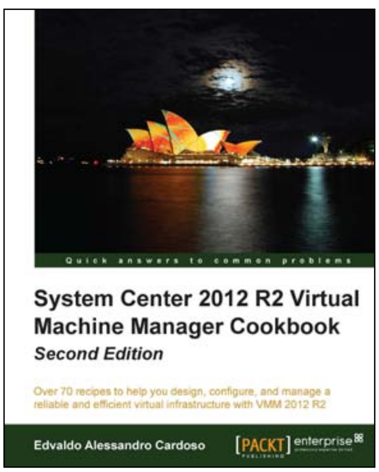

## Windows Server 2012 Unified Remote Access Planning and Deployment

ISBN: 978-1-84968-828-4 Paperback: 328 pages

Discover how to seamlessly plan and deploy remote access with Windows Server 2012's successor to DirectAccess

1. The essential administrator's companion for the successor to DirectAccess.
2. Get to grips with configuring, enabling and deploying Unified Remote Access.
3. A quick start guide to have you up and running with Windows Server 2012 URA in no time.

---




## System Center 2012 R2 Virtual Machine Manager Cookbook *Second Edition*

ISBN: 978-1-78217-684-8 Paperback: 428 pages

Over 70 recipes to help you design, configure, and manage a reliable and efficient virtual infrastructure with VMM 2012 R2

1. Create, deploy, and manage datacenters and private and hybrid clouds with hybrid hypervisors using VMM 2012 R2.
2. Integrate and manage fabric (compute, storages, gateways, and networking), services and resources, and deploy clusters from bare metal servers.
3. Explore VMM 2012 R2 features such as Windows 2012 R2 and SQL 2012 support, converged networks, network virtualization, live migration, Linux VMs, and resource throttling and availability.

Please check **www.PacktPub.com** for information on our titles

[PACKT] enterprise
professional expertise distilled
PUBLISHING

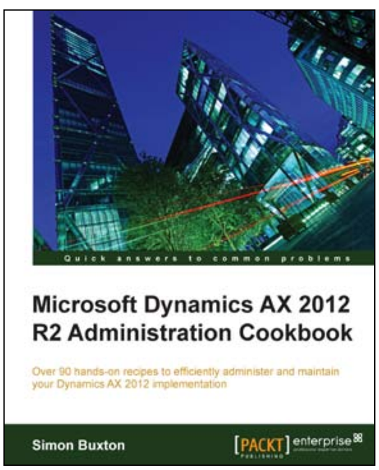

## Windows Server 2012 Automation with PowerShell Cookbook

ISBN: 978-1-84968-946-5 Paperback: 372 pages

Over 110 recipes to automate Windows Server administrative tasks using PowerShell

1. Extend the capabilities of your Windows environment.
2. Improve the process reliability by using well defined PowerShell scripts.
3. Full of examples, scripts, and real-world best practices.

---




## Microsoft Dynamics AX 2012 R2 Administration Cookbook

ISBN: 978-1-84968-806-2 Paperback: 378 pages

Over 90 hands-on recipes to efficiently administer and maintain your Dynamics AX 2012 implementation

1. Task-based examples for application and third-party interactions through the AIF.
2. Step-by-step instructions for performing user and security management operations.
3. Detailed instructions for performance and troubleshooting AX 2012.

Please check **www.PacktPub.com** for information on our titles