Active Directory Administration Cookbook

Second Edition

Proven solutions to everyday identity and authentication challenges for both on-premises and the cloud

Sander Berkouwer

Active Directory Administration Cookbook

Second Edition

Proven solutions to everyday identity and authentication challenges for both on-premises and the cloud

Sander Berkouwer

Packt

BIRMINGHAM—MUMBAI

Active Directory Administration Cookbook

Second Edition

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor Packt Publishing or its dealers and distributors, will be held liable for any damages caused or alleged to have been caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

Group Product Manager: Vijin Boricha
Publishing Product Manager: Mohd Riyan Khan
Senior Editor: Arun Nadar
Content Development Editor: Sujata Tripathi
Technical Editor: Rajat Sharma
Copy Editor: Safis Editing
Project Coordinator: Shagun Saini
Proofreader: Safis Editing
Indexer: Sejal Dsilva
Production Designer: Roshan Kawale
Marketing Coordinator: Hemangi Lotlikar

First published: May 2019
Second edition: July 2022
Production reference: 1080622

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

ISBN 978-1-80324-250-7
`www.packt.com`

Contributors

About the author

Sander Berkouwer calls himself an Active Directory aficionado; he's done everything with Active Directory and Azure AD, including decommissioning. He has been MCSA-, MCSE-, and MCITP-certified for ages, as well as an MCT and a Microsoft **Most Valuable Professional** (**MVP**) on Directory Services and Enterprise Mobility for over a decade. Sander is also decorated with Veeam Vanguard and VMware vExpert awards for his international cross-platform knowledge, experience, and passion.

His background in industrial design engineering explains a lot of his creativity. His qualities extend beyond the typical four As in identity and access management. Of course, administration, authentication, authorization, auditing are necessities, but his out-of-the-box solutions get the most out of software, hardware, and the cloud.

His work as a consultant, blogger, author, and trainer are all means to achieve his goal to help people with technology so that IT is not a mere hurdle, but an infinite enabler.

About the reviewers

Carl Webster, aka **Webster**, specializes in Citrix, Active Directory, and technical documentation. He is the most active person in the Citrix zone on Experts Exchange and serves the broader Citrix community by writing articles and providing free PowerShell scripts at CarlWebster.com. Webster, a CTP since 2009, is a founding member of the Citrix Technology Professional Fellow program. He has a long history in the IT industry, starting with mainframes in 1977, moving to PCs and application development in 1986, and then shifting to Active Directory and networks in 2001. Webster has worked on hundreds of Active Directory troubleshooting, remediation, and migration projects since 2001. Admins in small to mid-sized businesses appreciate his PowerShell scripts and his detailed and thorough approach and prescriptive guidance.

James Mendez is a customer engineer at Microsoft working with customers using Active Directory, Windows Server platforms, and various Azure cloud services. He has worked in the IT industry for over 25 years, obtaining different IT certifications and also holding roles such as senior systems engineer and lead IT systems architect. He has gained experience and exposure to a variety of technologies (such as Microsoft, Cisco, and VMware) over the years, which include scripting, web development, ETL data integration, networking, virtualization, hyper-converged infrastructure. Outside of work, he has several interests, including music (composing), traveling, cycling, running, reading, continuously learning, and spending time with family.

I'd like to thank:

My brother, for the encouragement and support, being a mentor, and also sharing his invaluable experience and insight throughout the introductory years of my career.

My parents, for instilling a strong and honest work ethic in me and always encouraging me to invest 200% into anything I am passionate about.

The few but genuine friends I have (domestic and international), for truly being there and believing in me.

Table of Contents

Preface

1
Optimizing Forests, Domains, and Trusts

2
Managing Domain Controllers

3

Managing Active Directory Roles and Features

4

Managing Containers and Organizational Units

5

Managing Active Directory Sites and Troubleshooting Replication

6

Managing Active Directory Users

7

Managing Active Directory Groups

8

Managing Active Directory Computers

9

Managing DNS

10
Getting the Most Out of Group Policy

11

Securing Active Directory

12
Managing Certificates

13
Managing Federation

14

Handling Authentication in a Hybrid World (AD FS, PHS, PTA, and DSSO)

15
Handling Synchronization in a Hybrid World (Azure AD Connect)

16

Hardening Azure AD

Preface

Active Directory (AD) is an administration system for Windows administrators to automate network, security, and access management tasks in the Windows infrastructure. This second edition is updated to cover Windows Server 2022 and guides you through effective recipes for AD administration.

The book starts with a detailed focus on forests, domains, trusts, schemas, and partitions. After that, you'll learn how to manage domain controllers, organizational units (OUs), and default containers. You'll explore how to manage Active Directory sites, as well as identifying and solving replication problems. Later chapters cover different object types in Active Directory: users, groups, and computers. You'll also work through recipes that help you manage your AD domains, as well as managing user and group objects and computer accounts, expiring group memberships, and group managed service accounts (gMSAs) with PowerShell. This book discusses how to manage DNS and certificates and how to work with Group Policy. You'll then focus on security and federation before going on to explore **Azure Active Directory** (**Azure AD**) and how to integrate on-premises Active Directory with Azure AD. Finally, you'll discover how Microsoft Azure AD Connect synchronization works and how to harden Azure AD.

By the end of this book, you'll be able to make the most of Active Directory, Azure AD and Azure AD Connect.

Who this book is for

This book is for administrators of existing **Active Directory Domain Services** (**AD DS**) environments and/or Azure AD tenants looking for guidance to optimize their day-to-day tasks. Basic networking and Windows Server operating system knowledge will be useful for getting the most out of this book.

What this book covers

Chapter 1, *Optimizing Forests, Domains, and Trusts*, discusses how Active Directory for large organizations entails managing many logical aspects of Active Directory. This chapter focuses on the intangible aspects of Active Directory: forests, domains, trusts, schemas, and partitions.

Chapter 2, *Managing Domain Controllers*, shows how domain controllers represent Active Directory towards devices, applications, and users.

Chapter 3, *Managing Active Directory Roles and Features*, details how some domain controllers are created more equal than others. The differences between domain controllers and how to manage them are described in this chapter.

Chapter 4, *Managing Containers and Organizational Units*, explains how there is a standard set of containers and OUs that are created during the installation of Active Directory. These are usually confused by Active Directory administrators. This chapter will help administrators understand when and why they need to use OUs instead of containers and how to perform all common tasks.

Chapter 5, *Managing Active Directory Sites and Troubleshooting Replication*, looks at how a site is a logical means to represent the physical aspects of AD. In this chapter, you will create and manage sites, subnets, and sitelinks. The focus here will also be on identifying, managing, and solving AD replication problems.

Chapter 6, *Managing Active Directory Users*, looks at Active Directory objects, which are where you manage the organization's resources. With the effective tips and tricks given in this chapter, you will be able to create, delete, and manage users.

Chapter 7, *Managing Active Directory Groups*, looks at groups, which are the cornerstone to providing access in Active Directory. With the information in this chapter, you will be able to create, delete, and manage groups and change the scope of a group based on your requirements.

Chapter 8, *Managing Active Directory Computers*, discusses how Active Directory computer objects offer single sign-on and a secure channel between devices, domain controllers, and resources.

Chapter 9, *Managing DNS*, looks at **Domain Name System** (**DNS**), which is important to Active Directory. While not every domain controller is a DNS server, most are. You will learn how to manage DNS.

Chapter 10, *Getting the Most Out of Group Policy*, looks at Group Policy, which helps to control the settings deployed to the user objects and computers of your Active Directory infrastructure. In this chapter, we will cover recipes to work with **Group Policy objects** (**GPOs**) to help bring greater understanding to this topic.

Chapter 11, *Securing Active Directory*, discusses how Active Directory plays a critical role in the IT infrastructure and safeguards the security of different network resources in an interconnected environment. In this chapter, we will cover a set of practical techniques that will help administrators protect an enterprise Active Directory environment.

Chapter 12, *Managing Certificates*, covers certificates. To secure communications between hosts and the internet, certificates can be issued by **certification authorities** (**CAs**). In this chapter, you'll learn how to set one up, manage it, and optionally decommission it.

Chapter 13, *Managing Federation*, looks at federation, which is the way organizations collaborate using open authentication standards. You will learn how to set up, configure, and manage **Active Directory Federation Services** (**AD FS**) servers and Web Application Proxy servers in this chapter.

Chapter 14, *Handling Authentication in a Hybrid World (AD FS, PHS, PTA, and DSSO)*, shows you how to integrate Active Directory identities with your Azure AD. The information in this chapter will revolve around managing AD FS, PHS, PTA, and DSSO.

Chapter 15, *Handling Synchronization in a Hybrid World (Azure AD Connect)*, explains how synchronization works with Azure AD Connect and how to customize it. It helps you choose the right source anchor attribute and manage the Azure AD Connect service accounts.

Chapter 16, *Hardening Azure AD*, discusses how many organizations depend on the integrity of the privileged accounts that manage IT systems for the security of business assets. Cyber-attackers focus on Active Directory and Azure AD to gain access to an organization's sensitive data. This chapter will offer expert tips on hardening security with Azure AD.

To get the most out of this book

To get the most out of the book, it helps to have basic knowledge of Windows Server and Active Directory.

Many recipes are written to lift an aging Active Directory environment to new heights. It helps in these cases to know the old protocols, such as **NT LAN Manager** *(***NTLM***)*, but an open mind is a more valuable asset when engaging with the recipes.

Some recipes in this cookbook require significant hardware, so if you're staging changes in development, test, or acceptance environments, make sure you have the computational power and storage to do so.

Software/hardware covered in the book	**Operating system requirements**
Microsoft Windows Server 2022	
Microsoft Local Administrator Password Solution (LAPS)	Windows 8.1 or higher and Windows Server 2012 or higher
Microsoft SQL Server 2019	Windows Server 2016 or higher
x.509 TLS certificate	
Microsoft Pass-through Authentication agent	
DNS domain name	
Microsoft Azure Active Directory tenant	
Microsoft Azure AD Connect	Windows Server 2016 or higher
Microsoft Azure AD Connect Health agent for AD FS	Windows Server 2012 or higher
Microsoft Azure AD Connect Health agent for AD DS	Windows Server 2012 or higher

If you are using the digital version of this book, we advise you to type the code yourself or access the code from the book's GitHub repository (a link is available in the next section). Doing so will help you avoid any potential errors related to the copying and pasting of code.

Download the example code files

You can download the example code files for this book from GitHub at `https://github.com/PacktPublishing/Active-Directory-Administration-Cookbook-Second-Edition`. If there's an update to the code, it will be updated in the GitHub repository.

We also have other code bundles from our rich catalog of books and videos available at `https://github.com/PacktPublishing/`. Check them out!

Code in Action

The Code in Action videos for this book can be viewed at `http://bit.ly/2OQfDum`.

Download the color images

We also provide a PDF file that has color images of the screenshots and diagrams used in this book. You can download it here: `http://www.packtpub.com/sites/default/files/downloads/Bookname_ColorImages.pdf`.

Conventions used

There are a number of text conventions used throughout this book.

`Code in text`: Indicates code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles. Here is an example: "Alternatively, you can search for its executable (`servermanager.exe`) in the Start menu."

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

```
redirusr.exe "OU=Redirected Users OU,DC=LucernPub,DC=com"
redircmp.exe "OU=Redirected Computers OU,DC=LucernPub,DC=com"
```

Bold: Indicates a new term, an important word, or words that you see onscreen. For instance, words in menus or dialog boxes appear in **bold**. Here is an example: "In the **User settings** pane, change the **Users can register applications** setting to **No**."

> **Tips or Important Notes**
>
> Appear like this.

Get in touch

Feedback from our readers is always welcome.

General feedback: If you have questions about any aspect of this book, email us at `customercare@packtpub.com` and mention the book title in the subject of your message.

Errata: Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you have found a mistake in this book, we would be grateful if you would report this to us. Please visit `www.packtpub.com/support/errata` and fill in the form.

Piracy: If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at `copyright@packt.com` with a link to the material.

If you are interested in becoming an author: If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, please visit `authors.packtpub.com`.

Share Your Thoughts

Once you've read *Active Directory Administration Cookbook, Second Edition*, we'd love to hear your thoughts! Please click here to go straight to the Amazon review page for this book and share your feedback.

Your review is important to us and the tech community and will help us make sure we're delivering excellent quality content.

1

Optimizing Forests, Domains, and Trusts

Back in the year 2000, when Active Directory was introduced to the larger public, we lived in a different world. The internet was only just starting to deliver value to businesses. That's why, in Windows 2000 Server, Active Directory was largely disconnected from the internet. Windows 2000 Server's default **Domain Name System** (**DNS**) settings even came with a root domain; so, if you wanted to connect to the internet, you had to delete the `.` DNS zone manually.

Fast forward to today, and the internet and cloud services seem omnipresent. The default `.` DNS zone has disappeared from Windows Server, but the concepts of trees and forests in Active Directory has persisted, and they still allow for some confusion among Active Directory admins.

To explain domains, trees, and forests in Active Directory, we need to acknowledge Active Directory's past. To create anything in Active Directory, you'll need to create a domain. It starts with the name. For a hypothetical organization, Lucern Publishing, four typical domain names would be as follows:

Type	Domain name
Public DNS domain name	`lucernpub.com`
Internal part of a public DNS domain name	`ad.lucernpub.com`
Non-public DNS domain name	`lucernpub.local`
Single-label domain name	`lucernpubcom`

Table 1.1 – Typical domain names

The first two options are the preferred options, as they adhere to RFC 822 (`https://www.w3.org/Protocols/rfc822`). The third option is a common option but doesn't comply with RFC 2606 (`https://tools.ietf.org/html/rfc2606`) and should be avoided. The fourth option is a typical **single-label domain**. They are usually the result of a common error among Active Directory admins migrating from Windows NT 4 Server's model to Active Directory. Products that once supported Windows NT 4 Server's single-label domains are no longer around, or they no longer support single-label domain names, including Microsoft.

Lucern Publishing may be quite a successful organization, so they might expand their operations from Switzerland to Europe, North America, and Asia. For reasons that we'll discuss later, they might want to separate **Active Directory domains** for each of their territories, but they want them to keep working together like one organization. This is where a domain tree comes into play. Now, Lucern Publishing might choose to create three subdomains under `lucernpub.com`:

- `eu.lucernpub.com`
- `usa.lucernpub.com`
- `asia.lucernpub.com`

They've created a tree of Active Directory domains, sharing the same DNS namespace. Of course, Lucern Publishing might also choose to create multiple trees, next to the `lucernpub.com` domain or tree, to accommodate an organizational layout with different names for their global expansions, such as Austin Publishing and Wuhan Publishing. In this case, it will make sense to create separate domains such as `austinpub.com` and `wuhanpub.com`. Effectively, Lucern Publishing will create three trees this way, belonging to the same **Active Directory forest**. Yes, some Active Directory environments are large structures with many large trees, but the default Active Directory forest consists of just one tree, with one Active Directory domain.

In this chapter, we'll discuss the reasoning behind creating domains and forests. We'll also discuss **userPrincipalName** (**UPN**) suffixes and trusts. The goal of this chapter is to help you make the right choices in terms of your Active Directory structure.

The following recipes are covered in the chapter:

- Choosing between a new domain or forest
- Listing the domains in your forest
- Using `adprep.exe` to prepare for new Active Directory functionality
- Raising the domain functional level to Windows Server 2016
- Raising the forest functional level to Windows Server 2016
- Creating the right trust
- Removing a trust
- Verifying and resetting a trust
- Securing a trust
- Extending the schema
- Enabling the Active Directory Recycle Bin
- Managing UPN suffixes

Before going through these recipes, we will look at a few aspects that you will need to know for this chapter.

Let's begin!

Choosing between a new domain or forest

In organizations, sometimes, an expansion or business change requires changes in Active Directory too. In Active Directory terms, the change might require creating a new **Active Directory domain** or a new **Active Directory forest**. In this recipe, we'll look at the reasoning between these two choices, taking the entire life cycle of Active Directory into consideration.

Why would you have a new domain?

A new Active Directory domain – as either a subdomain of an existing domain or a new domain tree in an existing forest – provides a boundary.

The boundary of domains in Active Directory relates to the following:

- **DNS name**: An additional domain tree offers the possibility to add a DNS domain name to the organization to, for instance, correctly label a new business venture. An alternative might be to add an additional UPN suffix.
- **Domain DNS zones replication**: Throughout an Active Directory forest, all domain controllers replicate to exchange information on objects, schemas, and configuration. Between domains, a distinction can be made to limit the replication of information on Active Directory-integrated DNS zones. That way, this information is only replicated within the domain.
- **Password and account lock-out policies**: Fine-grained password and account lock-out policies can only be applied within an Active Directory domain. The information can be viewed by any account in the domain. If you want to shield this information or create separate policies, an additional domain is the route to go.
- **Group Policy**: **Group Policy Objects** (**GPOs**) only replicate within a domain. The only exception is the GPOs that are linked to Active Directory sites; these are copied between domains instead since Active Directory sites are created at the forest level.

However, the boundary of domains in Active Directory does not include the following:

- An Active Directory schema
- The scope of the **Enterprise Admins** group

Essentially, a new Active Directory domain is an administrative boundary, which you can create for an organization to allow for delegated management.

What are the downsides of a new domain?

Microsoft's advice is to keep Active Directory as simple as possible. When you create additional domains, the organization ends up with the following:

- At least two additional domain controllers
- Active Directory trusts between the current domain(s) and the new domain
- An increase in administrative burden

Why would you create a new forest?

A new Active Directory forest is basically a completely new Active Directory environment. When you create it, it does not have a relationship with an existing Active Directory environment, unless you choose to create Active Directory trusts afterward.

Since the new Active Directory forest is separate, a boundary is created for the following reasons:

- **Schema and configuration partitions**: The schema and configuration partitions hold information on the way that objects can be created, what attributes are required for these objects, what attributes are optional for these objects, and the domains within the forest. Since many applications require Active Directory schema extensions, introducing a legacy or cutting-edge application might result in schema conflicts. In these types of scenarios, creating an additional Active Directory forest is the best way forward. An alternative might be to add an **Active Directory Lightweight Directory Services** (**AD-LDS**) instance to the environment.
- **Global catalog replication**: Domain controllers with the additional global catalog role hold partial information on the most requested attributes for objects in Active Directory. With multiple global catalogs, the information is replicated throughout the forest. To shield this information, an additional Active Directory forest can be created.
- **Forest DNS zones replication**: To overcome the default boundary for Active Directory-integrated DNS zones, the Forest DNS zone replication scope, an additional Active Directory forest can be created.

When requirements apply in terms of schemas or replication, creating an Active Directory forest is the right choice. One thing that might be good here is to state that the forest is a security boundary as well as an administrative boundary.

Additionally, since the forest is a separate environment, by default, it can also be separated afterward. In acquisition and divestiture scenarios that can be overseen for the life cycle of Active Directory, an Active Directory forest is also the right choice.

What are the downsides of a new forest?

A separate Active Directory environment, of course, requires double the administrative effort of Active Directory admins. Additionally, since the environments are separate, creating an address list in Microsoft Exchange Server or sharing common applications, services, and/or systems can be challenging.

Now we can look at the recipes covered in this chapter.

Listing the domains in your forest

In an Active Directory environment with multiple domains and forests, it can be hard to distinguish the trees from the forest. As authentication is often per forest, an easy way to list the domains per forest is welcome.

Getting ready

Alas, the only reliable way to list the domains in a forest is to use PowerShell.

For this recipe, we'll need one of the following:

- A domain controller running Windows Server 2012 with Desktop Experience (or a newer version of Windows Server)
- A domain-joined member server running Windows Server 2012 with Desktop Experience (or a newer version of Windows Server) with the Active Directory module for Windows PowerShell installed
- A domain-joined device running Windows 8.1 (or a newer version of Windows) with the Active Directory module for Windows PowerShell installed

On domain controllers running Windows Server 2012 with Desktop Experience (and on newer versions of Windows Server), the Active Directory module for the Windows PowerShell feature is automatically installed, when promoted to a domain controller.

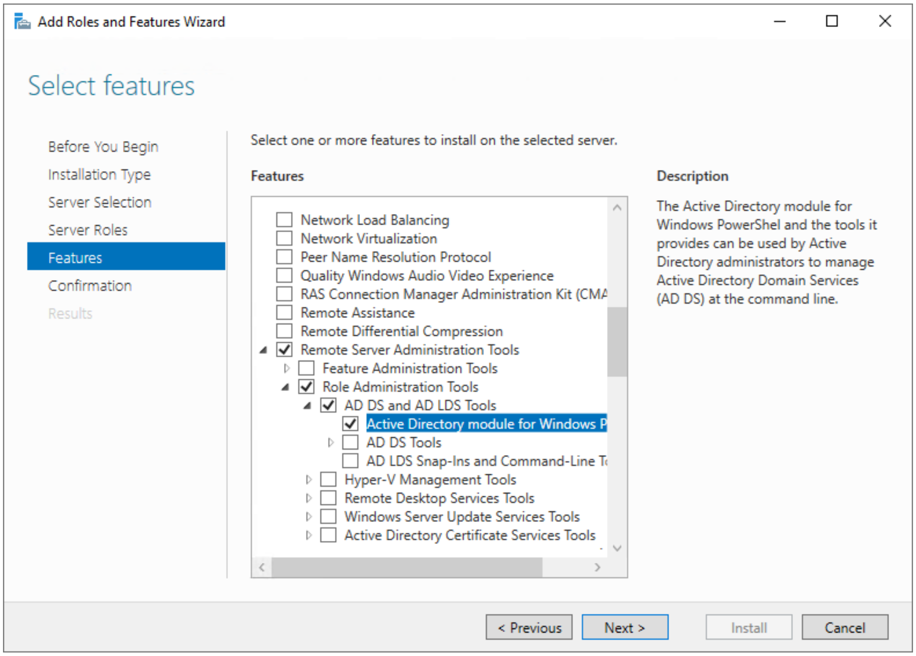
On domain controllers running Server Core installations of Windows Server 2012 (and on newer versions of Windows Server), the availability of the Active Directory module for Windows PowerShell depends on the `-IncludeManagementTools` option for the `Install-WindowsFeature` Windows PowerShell cmdlet used to install the Active Directory Domain Services role.

Installing the Active Directory module for Windows PowerShell on Windows Server

To install the Active Directory module for Windows PowerShell on a Windows Server with Desktop Experience, perform the following steps:

1. Press Start.
2. Search for **Server Manager** and select it from the search results or run `servermanager.exe`. The **Server Manager** window appears.
3. In the top gray pane, click **Manage**.
4. Select **Add Roles and Features** from the context menu.

5. In **Add Roles and Features Wizard**, click **Next** > until you reach the **Select Features** screen.
6. On the **Select Features** screen, scroll down in the list of features until you reach **Remote Server Administration Tools**.
7. Expand **Remote Server Administration Tools**.
8. Expand **Role Administration Tools**.
9. Expand **AD DS and AD LDS Tools**.
10. Select the **Active Directory module for Windows PowerShell** feature:



Select one or more features to install on the selected server.

- ☐ Network Load Balancing
- ☐ Network Virtualization
- ☐ Peer Name Resolution Protocol
- ☐ Quality Windows Audio Video Experience
- ☐ RAS Connection Manager Administration Kit (CMA
- ☐ Remote Assistance
- ☐ Remote Differential Compression
- ☑ Remote Server Administration Tools
  - ☐ Feature Administration Tools
  - ☑ Role Administration Tools
    - ☑ AD DS and AD LDS Tools
      - ☑ Active Directory module for Windows P
      - ☐ AD DS Tools
      - ☐ AD LDS Snap-Ins and Command-Line T
    - ☐ Hyper-V Management Tools
    - ☐ Remote Desktop Services Tools
    - ☐ Windows Server Update Services Tools
    - ☐ Active Directory Certificate Services Tools

The Active Directory module for Windows PowerShel and the tools it provides can be used by Active Directory administrators to manage Active Directory Domain Services (AD DS) at the command line.

Figure 1.1 – The Active Directory module for Windows PowerShell feature in the Add Roles and Features Wizard

11. Click **Next** > until you reach the **Confirm installation** selections page.
12. Click **Install**.
13. Click **Close**.

To install the Active Directory module for Windows PowerShell on the command line of a Server Core installation of Windows Server, run these two commands:

```
PowerShell.exe
Install-WindowsFeature RSAT-AD-PowerShell
```

Installing the Active Directory module for Windows PowerShell on Windows

To install the Active Directory module for Windows PowerShell on a device running Windows 8.1 and Windows 10 prior to version 1809, download the separately available **Remote Server Administration Tools** (**RSAT**) package for your version of Windows. After you install the package, all the RSAT will be available, including the Active Directory module for Windows PowerShell.

To install the Active Directory Domain Services and Lightweight Directory Services tools, including the Active Directory module for Windows PowerShell, on a device running Windows version 1809 (and newer versions of Windows), perform these steps:

1. Right-click Start.
2. Select the **Apps and Features** option from the context menu. The **Apps and Features** screen from **Settings** appears.
3. Follow the **Optional features** link.
4. On the **Optional features** screen, click the **Add a feature** button at the top. The **Add a feature** pop-up window appears.
5. Search for **RSAT: Active Directory Domain Services and Lightweight Directory Services Tools**.
6. Select the item from the search results.
7. Click **Install** to install the feature and close the **Add a feature** pop-up window.
8. After the tools have been installed, close the **Apps and Features** screen.

Alternatively, on devices running Windows 10 version 1809 (and newer versions of Windows), you can use the following line of Windows PowerShell to install the Active Directory Domain Services and Lightweight Directory Services tools, including the Active Directory module for Windows PowerShell:

```
Add-WindowsCapability -Online -Name Rsat.ActiveDirectory.
DS-LDS.Tools~~~~0.0.1.0
```

To list all the domains in a forest, use an account that is a member of the **Enterprise Admins** group in Active Directory.

How to do it...

On the system, start an elevated Windows PowerShell window or Windows PowerShell ISE window using the domain credentials for any domain account.

Then, type the following line of Windows PowerShell:

```
Get-ADForest | Select-Object domains
```

The output of this line of Windows PowerShell lists all Active Directory domains.

How it works...

We use the `Get-ADForest` cmdlet from the Active Directory module to get the information for the current Active Directory forest. Then, we pipe the output of that command to select only the domains, since that's what we're after.

You can now make the best choices for implementing new domains and/or forests, and/or decommissioning domains and/or forests.

Using adprep.exe to prepare for new Active Directory functionality

The Active Directory schema defines the way that objects can be created, and what attributes are required or are optional for these objects. With previous versions of Windows Server, the base schema has been improved and extended.

Many features require certain schema versions for Active Directory. For instance, when you want to deploy a Windows Server 2016-based **Active Directory Federation Services** (**AD FS**) farm, you'll need the Windows Server 2016 schema or a higher version of the schema.

Since Windows Server 2012, Microsoft updates the Active Directory schema automatically when you promote the first Windows Server 2012-based member server to an Active Directory domain controller.

However, consider what will happen if you want to do any of the following:

- Update the Active Directory schema only, because your organization doesn't want domain controllers running the latest version.
- Delegate the promotion of the first domain controller to a less-privileged user, instead of an admin that is a member of the **Schema Admins** group.
- Control the proper replication of the schema update to all domain controllers, before promoting the first domain controller.

- Avoid the default time-out that the Active Directory Configuration Wizard provides for proper replication.
- Perform all Active Directory preparations, including the Group Policy preparation step.

In these situations, you'll want to update the Active Directory schema manually, using `adprep.exe` from the Windows Server installation media.

Getting ready

Copy the entire contents of the `\support\adprep` folder from the Windows Server installation media to a temporary folder on your system's hard disk.

Required permissions

The Active Directory preparation process consists of four separate stages. You'll need an account with the following group memberships for each stage:

Stage	**Required group memberships**
Preparing the forest	• Enterprise Admins • Schema Admins • Domain Admins in the forest root domain
Preparing the forest for **Read-only Domain Controllers** (**RODCs**)	Domain Admins in the forest root domain
Preparing the domain	Domain Admins
Updating filesystem and Active Directory permissions on existing **Group Policy objects** (**GPOs**)	Domain Admins

Table 1.2 – Required permissions per Active Directory preparation step

How to do it...

Start Command Prompt in the **File Explorer** window of the folder that you've copied the files to.

You can simply type `cmd` in the address bar to achieve this.

The Active Directory preparation process consists of four separate stages:

- Preparing the forest
- Preparing the forest for RODCs
- Preparing the domainUpdating filesystem / Active Directory permissions on existing GPOs

After these steps, you'll want to check proper Active Directory replication.

Preparing the forest

Perform these steps to prepare the Active Directory forest:

1. To prepare the Active Directory forest, run the following command:

   ```
   adprep.exe /forestprep /forest lucernpub.com /user
   EntAdmin /userdomain lucernpub.com /password P@ssw0rd
   ```

   Replace the value for the domain and the values for the credentials with values that make sense for your Active Directory environment.

2. Next, type `c`, followed by *Enter* to continue.

   The following line at the end of the output indicates the successful preparation of the Active Directory forest:

   ```
   Adprep successfully updated the forest-wide information
   ```

When you see the preceding line appear, you can continue with the next step.

Preparing the forest for RODCs

The `/rodcprep` switch for `adprep.exe` triggers the preparation of the forest for RODCs. This action only needs to be performed when the intention is to run RODCs in the Active Directory forest:

1. To prepare the Active Directory forest for RODCs, run the following command:

   ```
   adprep.exe /rodcprep /forest lucernpub.com /user DomAdmin
   /userdomain lucernpub.com /password P@ssw0rd
   ```

   Replace the value for the domain and the values for the credentials with values that make sense for your Active Directory environment.

2. The following line at the end of the output indicates the successful preparation of the Active Directory forest for RODCs:

   ```
   Rodcprep completed without errors. All partitions are
   updated. See the ADPrep.log in directory C:\Windows\
   debug\adprep\logs\<date> for more information.
   ```

When you see the preceding line appear, you can continue with the next step.

Preparing the domain

Perform these steps to prepare the domain:

1. To prepare the Active Directory domain, run the following command:

   ```
   adprep.exe /domainprep /domain lucernpub.com /user DomAdm
   /userdomain lucernpub.com /password P@ssw0rd
   ```

   Replace the value for the domain and the values for the credentials with values that make sense for your Active Directory environment.

2. The following line at the end of the output indicates the successful preparation of the Active Directory domain:

   ```
   Adprep successfully updated the domain-wide information
   ```

When you see the preceding line appear, you can continue with the next step.

Updating filesystem and Active Directory permissions on existing GPOs

Group Policy preparation, as part of `adprep.exe`, adds two pieces of functionality to Active Directory:

- Cross-domain planning functionality for Group Policy
- **Resultant Set of Policy** (**RSoP**) planning mode

GPOs are stored in both the **System Volume** (**SYSVOL**) and Active Directory. Both locations require an update of the permissions for existing GPOs, to take advantage of the preceding functionality.

If the Active Directory domain already contains custom or delegated permissions, Group Policy preparation kicks off the replication of all Group Policy files in the SYSVOL and may deny the functionality of RSoP to delegated admins until their permissions are recreated.

> **Note**
>
> Group Policy preparation does not need to be run with every upgrade. Admins need to run Group Policy preparation only once, and they only need to run it if an Active Directory domain has run on Windows 2000 Server-based domain controllers at one point in its existence. If an environment was created with domain controllers running Windows Server 2003, or newer versions of Windows Server, the Group Policy preparation step can be skipped.

To update filesystem and Active Directory permissions for GPOs, run the following command:

```
adprep.exe /domainprep /gpprep /domain lucernpub.com /user
DomAdm /userdomain lucernpub.com /password P@ssw0rd
```

Replace the value for the domain and the values for the credentials with values that make sense for your Active Directory environment.

The following line at the end of the output indicates the successful preparation of the Active Directory domain:

```
Adprep successfully updated the Group Policy Object (GPO)
information.
```

Checking the preparation replication

When done with the preparation steps, the Active Directory schema base version should be upgraded to a higher number, corresponding to the new schema version.

The following table shows the version numbers in accordance with the Active Directory level:

Windows Server version	**Schema version**
Windows 2000 Server	13
Windows Server 2003	30
Windows Server 2003 R2	31
Windows Server 2008	44
Windows Server 2008 R2	47
Windows Server 2012	56
Windows Server 2012 R2	69
Windows Server 2016	87
• Windows Server 2019 • Windows Server 2022	88

Table 1.3 – Schema version per Windows Server version

You can manually check the schema version per domain controller with the following command from any of your domain controllers:

```
repadmin.exe /showattr *
"cn=schema,cn=configuration,dc=lucernpub,dc=com" /
atts:objectVersion
```

Replace `lucernpub` and `com` with values for your Active Directory environment.

When all domain controllers report the same schema version, the Active Directory preparation has replicated successfully to all domain controllers.

How it works...

In Windows Server 2012 (and later versions), the entire Active Directory preparation process is automated. When you promote a Windows Server 2012-based member server (or any newer version of Windows Server) to an additional domain controller for a domain or upgrade a domain controller running a previous version to Windows Server 2012 (or any newer version of Windows Server), the Active Directory Domain Services Configuration Wizard determines whether the environment needs to be prepared as part of the promotion process.

Larger organizations often separate the schema or preparation work from the actual domain controller-promotion process work to minimize risk, adhere to small change windows, and more.

However, `adprep.exe` is still available to prepare the Active Directory forest and/or Active Directory domain(s) manually.

Windows Server 2022 is the first Windows Server version that does not require a schema upgrade since Active Directory was released with Windows 2000 Server.

There's more...

Unless there are compelling reasons not to, preparing for the latest available Active Directory schema version is the recommended approach. A reason not to do this is when an organization doesn't want to enable the promotion of the latest version(s) of Windows Server to domain controllers in a delegated environment.

Raising the domain functional level to Windows Server 2016

When implementing new Active Directory domain controllers and removing domain controllers running previous versions of Windows Server, many admins forget to raise the Active Directory **domain functional level** (**DFL**) to the earliest Windows Server version still running as domain controllers. After upgrading all domain controllers from Windows Server 2008 R2 to Windows Server 2012 R2, for instance, they would not raise the DFL to Windows Server 2012 R2 but keep it at the Windows Server 2008 R2 level.

It's a shame, really, because many new Active Directory features and optional Active Directory features are only available when the functional level is raised. Furthermore, the DFL dictates the lowest version of Windows Server that admins can use to promote new domain controllers. In addition, since Windows Server 2008 R2, the DFL can also be reverted, as long as no new optional features have been enabled and the Active Directory **forest functional level** (**FFL**) is the same as the DFL that you want to revert to, or lower.

The Windows 2016 domain is the highest available DFL for Active Directory; there is no Windows 2019 or Windows Server 2022 domain level.

From an Active Directory point of view, the Windows Server 2008 DFL (or any newer version of the DFL), is required when you want to deploy Windows Server 2016-based domain controllers and domain controllers running newer versions of Windows Server. Additionally, DFS replication needs to be in use to replicate the SYSVOL.

Getting ready

Microsoft recommends raising the DFL from the Active Directory domain controller that holds the **Primary Domain Controller** (**PDC**) Emulator **Flexible Single Master Operations** (**FSMO**) role.

To locate this domain controller, run the following command on any domain-joined device, member server, or domain controller:

```
netdom.exe query fsmo
```

Alternatively, use the following line of PowerShell on a domain-joined system that has the Active Directory module for Windows PowerShell installed:

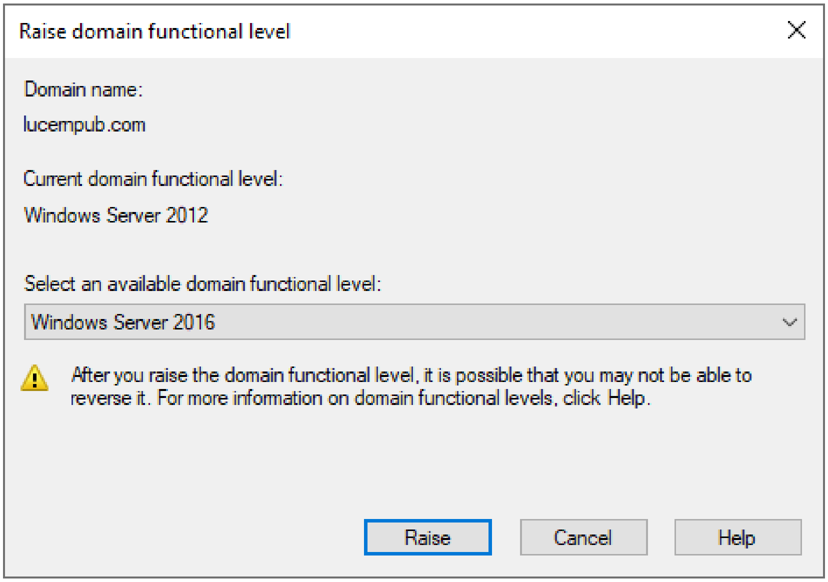
```
Get-ADDomain | Format-List PDCEmulator
```

Use an account that is a member of the **Domain Admins** group in the Active Directory domain for which you want to raise the DFL.

How to do it...

On domain controllers running Windows Server with the Desktop Experience, perform the following steps:

1. Sign in to the domain controller holding the PDC Emulator FSMO role.
2. Press Start.
3. Search for **Active Directory Domains and Trusts** and select it from the search results or run `domain.msc`. The **Active Directory Domains and Trusts** window appears.
4. In the left navigation pane, right-click the domain for which you want to raise the functional level, and then click **Raise Domain Functional Level**. The **Raise domain functional level** window appears:



Domain name:

lucernpub.com

Current domain functional level:

Windows Server 2012

Select an available domain functional level:

Windows Server 2016

After you raise the domain functional level, it is possible that you may not be able to reverse it. For more information on domain functional levels, click Help.

Figure 1.2 – The Raise domain functional level window

5. From the **Select an available forest functional level** drop-down list, select the desired DFL, and then click **Raise**.
6. The **Raise domain functional level** pop-up screen appears, stating that this change affects the entire domain and that after you raise the DFL, it is possible that you may not be able to reverse it. Click **OK**. This raises the DFL to the desired level.
7. Another **Raise domain functional level** pop-up screen appears, stating that the DFL was raised successfully. Click **OK** to acknowledge.

Alternatively, you can use the following PowerShell command:

```
Set-ADDomainMode lucernpub.com Windows2016Domain
```

Replace `lucernpub.com` with values for your Active Directory environment.

Even when under time pressure, you'll want to check for the proper replication of changes to Active Directory functional levels before making any other changes in Active Directory that might depend on them. Especially in large environments with elaborate replication technologies, replication might take a while.

To check for the proper replication of changes to Active Directory functional levels, use the following command:

```
repadmin.exe /showattr *.lucernpub.com "dc=lucernpub,dc=com" /
atts:msDS-Behavior-Version
```

Replace `lucernpub.com`, `lucernpub`, and `com` with values for your Active Directory environment.

The command checks the value for the `msDS-Behavior-Version` attribute on each of the domain controllers in the Active Directory domain and returns the value.

The following table shows the `msDS-Behavior-Version` attribute value per Active Directory DFL:

DFL	**msDS-Behavior-Version**
Windows 2000 Server	0
Windows Server 2003 Mixed	1
Windows Server 2003	2
Windows Server 2008	3
Windows Server 2008 R2	4
Windows Server 2012	5
Windows Server 2012 R2	6
• Windows Server 2016 • Windows Server 2019 • Windows Server 2022	7

Table 1.4 – msDS-Behavior-Version attribute values per Active Directory DFL

The output shows the domain controllers that are replicating a change from a lower value to a higher value. When each domain controller returns the same value, the DFL has successfully replicated throughout the Active Directory environment.

How it works...

When a domain controller operates, it references the DFL to know how it can optimally interoperate with other domain controllers in the Active Directory domain. Additionally, when you want to enable optional Active Directory features, the `msDS-Behavior-Version` attribute is referenced to see whether it's a permissible action.

If there is a domain controller running a version of Windows Server that does not meet the requirements of a certain DFL, the level is grayed out in the **Active Directory Domains and Trusts** console and the level may not be raised to the desired level. In this case, when you try to raise the DFL using Windows PowerShell or other programmatic means, it will error out.

Raising the forest functional level to Windows Server 2016

Just like the Active Directory DFL, the FFL also determines the availability of new Active Directory functionality. Where the DFL dictates the minimum version of Windows Server to run as domain controllers, the FFL dictates the minimum version of the DFL in the Active Directory forest.

The new functionality that is unlocked by raising the FFL includes the following:

- **Privileged Access Management** (**PAM**), which requires the Windows Server 2016 FFL
- Active Directory Recycle Bin, which requires the Windows Server 2008 R2 FFL
- Linked-value replication, which requires the Windows Server 2003 FFL

Getting ready

Microsoft recommends raising the FFL from the Active Directory domain controller that holds the Domain Naming Master FSMO role.

To locate this domain controller, run the following command on any domain-joined device, member server, or domain controller:

```
netdom.exe query fsmo
```

Alternatively, use the following line of Windows PowerShell on a domain-joined system that has the Active Directory module for Windows PowerShell installed:

```
Get-ADForest | Format-List DomainNamingMaster
```

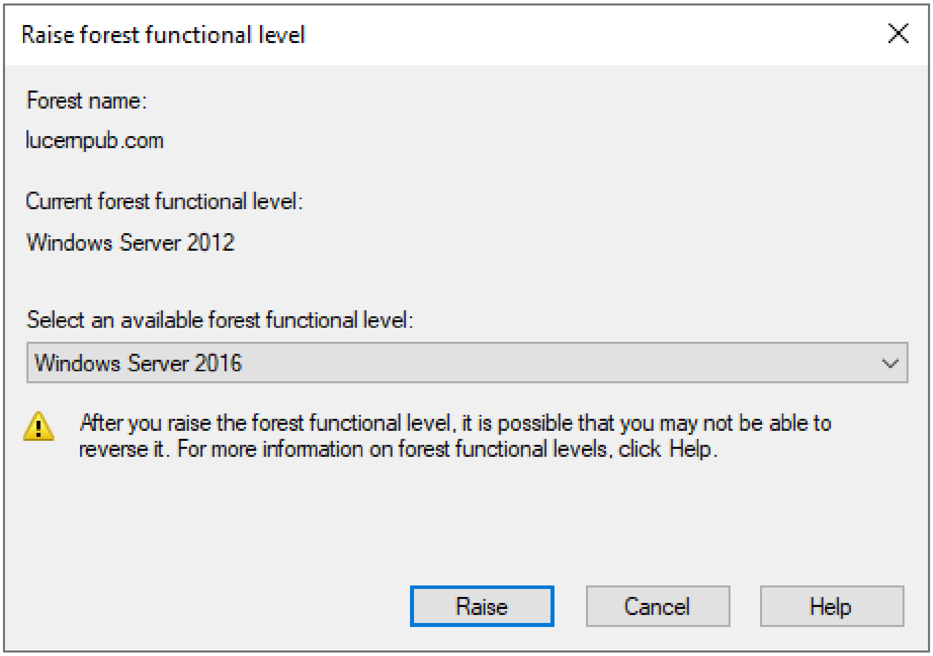
Use an account that is a member of the **Enterprise Admins** group in the Active Directory forest for which you want to raise the FFL.

How to do it...

On domain controllers running Windows Server with the Desktop Experience, perform the following steps:

1. Sign in to the domain controller holding the Domain Naming Master FSMO role.
2. Press Start.
3. Search for **Active Directory Domains and Trusts** and select it from the search results or run `domain.msc`. The **Active Directory Domains and Trusts** window appears.

4. In the left navigation pane, right-click **Active Directory Domains and Trusts**, and then click **Raise Forest Functional Level**. The **Raise forest functional level** window appears:



Raise forest functional level

Forest name:

lucernpub.com

Current forest functional level:

Windows Server 2012

Select an available forest functional level:

Windows Server 2016

⚠ After you raise the forest functional level, it is possible that you may not be able to reverse it. For more information on forest functional levels, click Help.

Figure 1.3 – The Raise forest functional level window

5. From the **Select an available forest functional level** drop-down list, select the desired FFL, and then click **Raise**.
6. The **Raise forest functional level** pop-up screen appears, stating that this change affects the entire forest and that after you raise the FFL, it is possible that you may not be able to reverse it. Click **OK**. This raises the FFL to the desired level.
7. Another **Raise forest functional level** pop-up screen appears, stating that the FFL was raised successfully. Click **OK** to acknowledge.

Alternatively, you can use the following line of Windows PowerShell:

```
Set-ADForestMode lucernpub.com Windows2016Forest
```

Replace `lucernpub.com` with values for your Active Directory environment.

How it works...

When a domain controller operates, it references the FFL to know how it can optimally interoperate with other domain controllers in the Active Directory forest. Additionally, when you want to enable optional Active Directory features, the `msDS-Behavior-Version` attribute is referenced to see whether it's a permissible action.

When a new Active Directory domain is added to an Active Directory forest, the available DFLs for the domain are shown, based on the `msDS-Behavior-Version` attribute for the forest too.

If there is a domain running a DFL that does not meet the requirements of a certain FFL, the level is grayed out in the **Active Directory Domains and Trusts** console and the level cannot be raised to this level. When you try to raise the FFL using Windows PowerShell or other programmatic means, it will error out.

Creating the right trust

In an Active Directory environment with multiple domains, you're bound to have trusts. Trusts allow people to access resources in a domain or forest other than the domain or forest where their user accounts reside.

When Active Directory domains are added to an existing Active Directory domain, two-way transitive trusts are automatically created. However, in other situations, trusts have to be created manually. With many different types of trusts, two trust directions, and a choice in transitivity, which trust is the right trust for which situation?

Let's look at the six types of trust first:

- **Parent-child trust**: The parent-child trust is a trust type that is automatically created when you add a domain to a tree root. For example, a parent-child trust is automatically created between `adatum.com` and `sub.adatum.com`. You cannot manually create a parent-child trust.
- **Tree-root trust**: The tree-root trust is a trust type that is also automatically created, just like the parent-child trust. However, the tree-root trust is created when you add a new domain tree to an Active Directory forest; for example, when you add the domain to a forest that contains only the `adatum.com` domain. The difference between the tree-root trust and the parent-child trust is that with the former, you break the domain tree, whereas, with the latter, you expand on it. You cannot manually create a tree-root trust.

- **Forest trust**: A forest trust is a trust type that you will have to create manually. When accounts in two separate Active Directory forests require access to each other's resources, then this is the right trust type to create between the two forest root domains. Creating a forest trust is highly preferable over creating an external trust, because the latter only supports older authentication schemes, whereas a forest trust supports Kerberos authentication.
- **Realm trust**: The realm trust type exists to help you connect with non-Active Directory environments, such as Samba-based environments and Novell eDirectory-based environments. The requirement for the other side of the trust is that it needs to support the Kerberos version.
- **External trust**: An external trust is a trust type that you will have to create manually. This trust type is truly versatile, as you can create a trust with any other environment, including Windows NT 4.0 Server-based environments. The downside is that it leverages NTLM as its authentication protocol; this is considered an outdated and weak protocol.
- **Shortcut trust**: In large Active Directory environments, authentication to access a resource may take a long time. As a user traverses trusts within an Active Directory forest, they have to perform Kerberos authentication up and down trees until they reach the domain with the resource they want to access. The rule of thumb is to create a shortcut trust when users in one domain regularly use resources in another domain (but within the same forest), and they have to traverse five, or more, trusts to get from the domain where their user accounts reside to the domain where the resources reside.

With the preceding information on trusts, we can create the following flowchart:

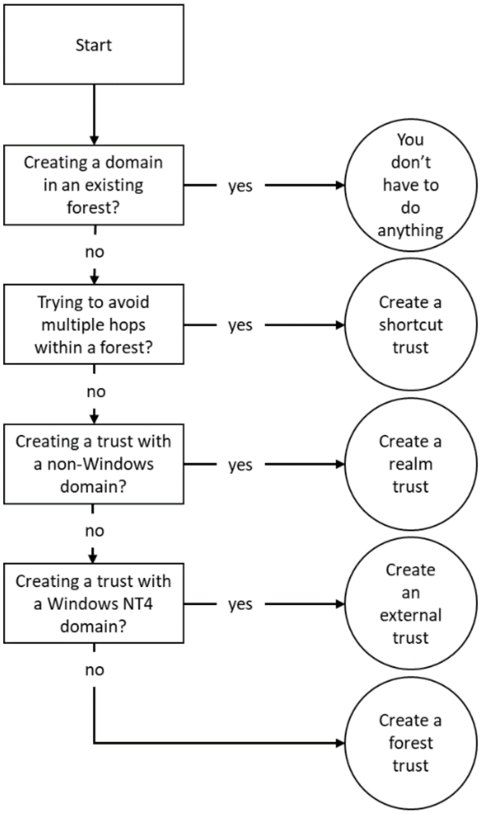

Start
Creating a domain in an existing forest?
yes
You don't have to do anything
no
Trying to avoid multiple hops within a forest?
yes
Create a shortcut trust
no
Creating a trust with a non-Windows domain?
yes
Create a realm trust
no
Creating a trust with a Windows NT4 domain?
yes
Create an external trust
no
Create a forest trust

Figure 1.4 – Flowchart for choosing the right Active Directory trust type

Trust direction

The right direction of a trust can be simply explained with an analogy. Suppose that a friend wants to borrow your car; you share your keys to the car with them and then you give them permission to use the car.

In terms of the Active Directory trust, the friend will map to the **user account**, the car to a **resource**, and you will be the **resource owner**. The trust flows from the resource to the user. The admin of the resource has to create the trust.

Trust transitivity

The same analogy also works for transitivity. Suppose that you want to lend your car to a friend, but they also want to lend the car to their friends. If you trust the friend enough, you will probably allow them to do so. If you don't have this level of trust in your friend, or you know some of their friends and you don't trust them with your car, then it's a bad idea.

In terms of Active Directory trusts, the trust type where you trust all the user accounts in all the domains to access the resource is a transitive trust; parent-child trusts, tree-root trusts, and forest trusts are transitive, by default.

The trust type where you only trust your friend and not all their friends to access the resource is a non-transitive trust. Realm trusts and external trusts are non-transitive, by default.

One-way or two-way trust

Essentially, every trust is a one-way trust. All the trusts that you create manually are one-way trusts, by default. However, you can combine two one-way trusts in opposite directions to create a two-way trust. In the scope of a two-way trust, people on both sides of the trust can access resources on both sides of the trust using their user accounts.

Microsoft recommends creating a two-way trust when your goal is to migrate accounts and resources using the **Active Directory Migration Tool** (**ADMT**). After migrating everything over, the two-way trust can be torn down.

Getting ready

To create a trust between two environments, ensure that the two environments know how to find each other. In DNS, create any necessary (conditional) forwarders or stub zones to point domain controllers from one environment to the domain controllers or Kerberos **Key Distribution Centers** (**KDCs**) of the other environment.

Additionally, take care of proper networking; the domain controller holding the PDC Emulator FSMO role and at least one global catalog for each domain on the route of the trust should be reachable from the device that someone uses to access the resource. The following networking traffic should be allowed through both host and network firewalls:

Service	Protocol	Port
Kerberos authentication	TCP and UDP	88
RPC endpoint mapper	TCP	135
NetBIOS name service	TCP and UDP	137
File Replication Service (**FRS**) between domain controllers	UDP	138
Distributed File System Replication (**DFSR**), NetBIOS session service	TCP	139
Lightweight Directory Access Protocol (**LDAP**)	TCP and UDP	389
Server Message Block (**SMB**)	TCP and UDP	445
Kerberos password change	TCP and UDP	464
Lightweight Directory Access Protocol over SSL (**LDAPS**)	TCP and UDP	636
LDAP to global catalogs	TCP	3268
LDAPs to global catalogs	TCP	3269

Table 1.5 – Network ports used by Active Directory

Preferably, you should sign in to the domain controller that is running the Domain Naming Master FSMO role or connect the **Active Directory Domains and Trusts** console to this specific domain controller.

To find this domain controller, right-click the **Active Directory Domains and Trusts** node and select **Operations Master...** from the menu. Alternatively, run the following command from any domain-joined device, member server, or domain controller:

```
netdom.exe query fsmo
```

Otherwise, you can use the following line of Windows PowerShell on a domain-joined system that has the Active Directory module for Windows PowerShell installed:

```
Get-ADForest | Format-List DomainNamingMaster
```

For shortcut trusts, sign in with the credentials of an admin account that is a member of the **Domain Admins** group. For all other trusts, sign in with the credentials of an admin account that is a member of the **Enterprise Admins** group.

How to do it…

To create a trust on Windows devices or Windows Servers with the Desktop Experience, use the **Active Directory Domains and Trusts** console for the domain you (as a resource owner) want to give access to:

1. Press Start.
2. Search for **Active Directory Domains and Trusts** and select it from the search results or run `domain.msc`. The **Active Directory Domains and Trusts** window appears.
3. In the console tree, right-click the domain that you want to allow access to, and then click **Properties...**.
4. Navigate to the **Trusts** tab, as follows:

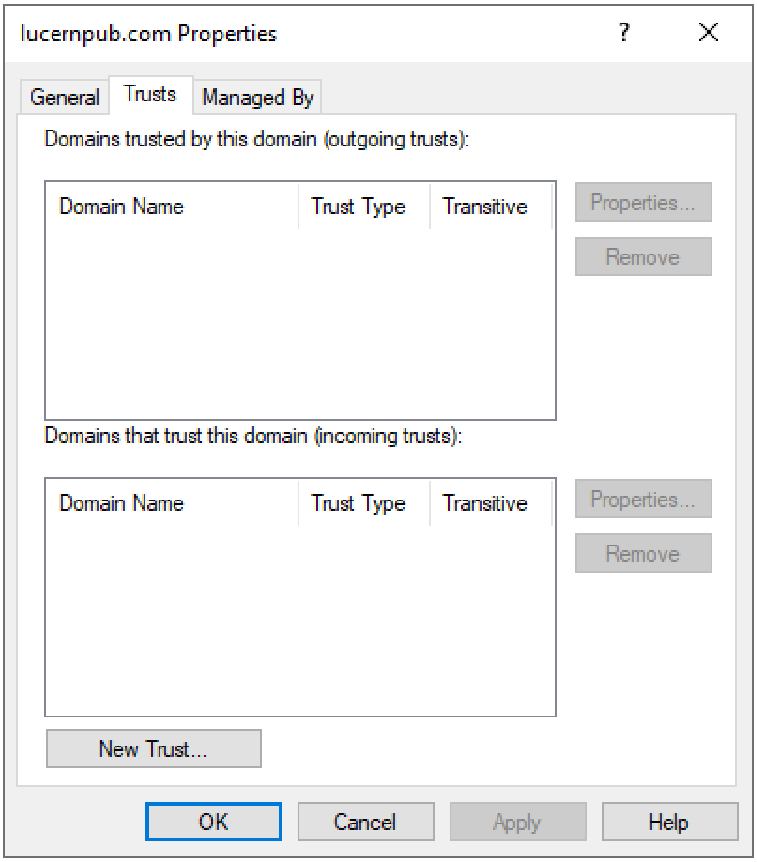
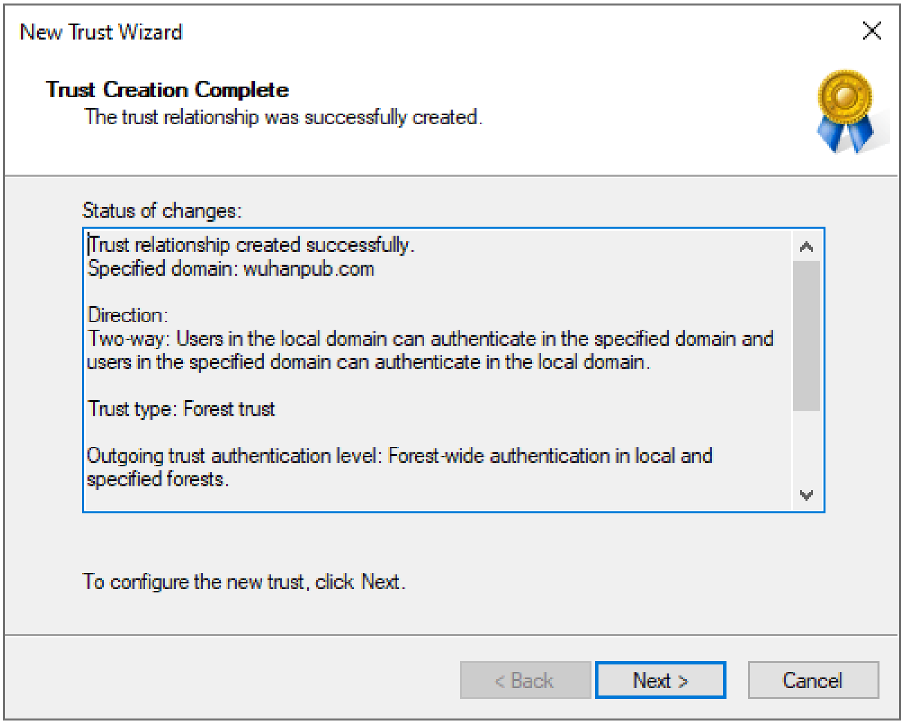


Figure 1.5 – Properties for a domain in Active Directory Domains and Trusts

5. Click the **New Trust…** button.
6. Run through the **New Trust Wizard**.
7. In the **Welcome to the New Trust Wizard** screen, click **Next** >.
8. In the **Trust Name** screen, type a name for the trust in the **Name** field. Then, click **Next** > when done.
9. In the **Trust Type** screen, choose between a **Realm trust** or a **Trust with a Windows domain**. For the latter, type the name of the domain, in case it's different to the trust name.
10. Click **Next** >.
11. In the **Trust Type** screen, choose between an **External trust** or a **Forest trust**.
12. Click **Next** >.
13. In the **Direction of Trust** screen, choose between a **Two-way**, **One-way: incoming**, or **One-way: outgoing** trust.
14. Click **Next** >.
15. In the **Sides of Trust** screen, choose between creating the trust for this domain only, or both this domain and the specified domain.
16. Click **Next** >.
17. In the **User Name and Password** screen, provide the credentials of an account that has administrative privileges in the Active Directory domain on the other side of the trust.
18. Click **Next** >.
19. In the **Outgoing Trust Authentication Level – Local Forest** and/or **Outgoing Trust Authentication Level – Specified Forest** screens, choose between **Forest-wide authentication** and **Selective authentication**.
20. Click **Next** >.
21. In the **Trust Selections Complete** screen, review the settings, and click **Next>** to create the trust.

22. In the **Trust Creation Complete** screen, click **Next** >:



Trust Creation Complete

The trust relationship was successfully created.

Status of changes:

Trust relationship created successfully.
Specified domain: wuhanpub.com

Direction:
Two-way: Users in the local domain can authenticate in the specified domain and users in the specified domain can authenticate in the local domain.

Trust type: Forest trust

Outgoing trust authentication level: Forest-wide authentication in local and specified forests.

To configure the new trust, click Next.

Figure 1.6 – The Trust Creation Complete screen of the New Trust Wizard

23. In the **Confirm Outgoing Trust** and/or **Confirm Incoming Trust** screens, choose between **No, do not confirm the outgoing trust** and **Yes, confirm the outgoing trust**.
24. Click **Next** >.
25. In the **Completing the New Trust Wizard** screen, click **Finish**.

Alternatively, you can use the following commands:

```
netdom.exe trust TrustingDomain.tld /Domain:TrustedDomain.tld /
TwoWay /Add
```

Replace `TrustingDomain.tld` with the DNS domain name of the Active Directory environment that gives access to its resources, and then replace `TrustedDomain.tld` with the DNS domain name of the Active Directory environment that gains access to the resources.

In the preceding example, a two-way trust is created where both Active Directory environments give and gain access to the other Active Directory environment.

See also

Perform the steps in the *Creating conditional forwarders* recipe in *Chapter 9*, *Managing DNS* to configure the conditional forwarders for name resolution.

Removing a trust

When a trust is no longer needed, it can be deleted.

Getting ready

Preferably, you should sign in to the domain controller that is running the Domain Naming Master FSMO role or connect the **Active Directory Domains and Trusts** console to this specific domain controller.

To find this domain controller, right-click the **Active Directory Domains and Trusts** node and select **Operations Master...** from the menu. Alternatively, run the following command from any domain-joined device, member server, or domain controller:

```
netdom.exe query fsmo
```

Otherwise, you can use the following line of Windows PowerShell on a domain-joined system that has the Active Directory module for Windows PowerShell installed:

```
Get-ADForest | Format-List DomainNamingMaster
```

To remove a shortcut trust, use an account that is a member of the **Domain Admins** group in the Active Directory domain. To remove other trusts, use an account that is a member of the **Enterprise Admins** group.

How to do it...

To remove an Active Directory trust, perform the following steps:

1. Sign in to the domain controller holding the Domain Naming Master FSMO role.
2. Press Start.
3. Search for **Active Directory Domains and Trusts** and select it from the search results or run `domain.msc`. The **Active Directory Domains and Trusts** window appears.

4. In the console tree, right-click the domain that you want to allow access to, and then click **Properties**.
5. Navigate to the **Trusts** tab.
6. From the list of **Domains trusts by this domain (outgoing trusts)**, or from the list of **Domains that trust this domain (incoming trusts)**, select the trust that you want to remove.
7. Click the **Remove** button next to the corresponding list.

How it works...

It's a recommended practice in Active Directory to remove objects and settings that have no use.

Verifying and resetting a trust

After you create a trust, you might regularly want to check whether the trust is working properly. You might be notified by people who report that they can no longer access resources in other domains or forests, or it might be an activity that you perform on a regular basis.

When a trust is broken, there is a way to reset it. Also, when you want to reset the shared secret on both sides of the trust, a reset of the trust is needed.

Getting ready

It is recommended that you sign in to the domain controller that is running the Domain Naming Master FSMO role or connect the **Active Directory Domains and Trusts** console to this specific domain controller, by right-clicking in the console on the **Active Directory Domains and Trusts** node and selecting **Change Active Directory Domain Controller…** from the context menu.

To find this domain controller, right-click the **Active Directory Domains and Trusts** node and select **Operations Master...** from the menu. Alternatively, run the following command from any domain-joined device, member server, or domain controller:

```
netdom.exe query fsmo
```

Otherwise, you can use the following line of Windows PowerShell on a domain-joined system that has the Active Directory module for Windows PowerShell installed:

```
Get-ADForest | Format-List DomainNamingMaster
```

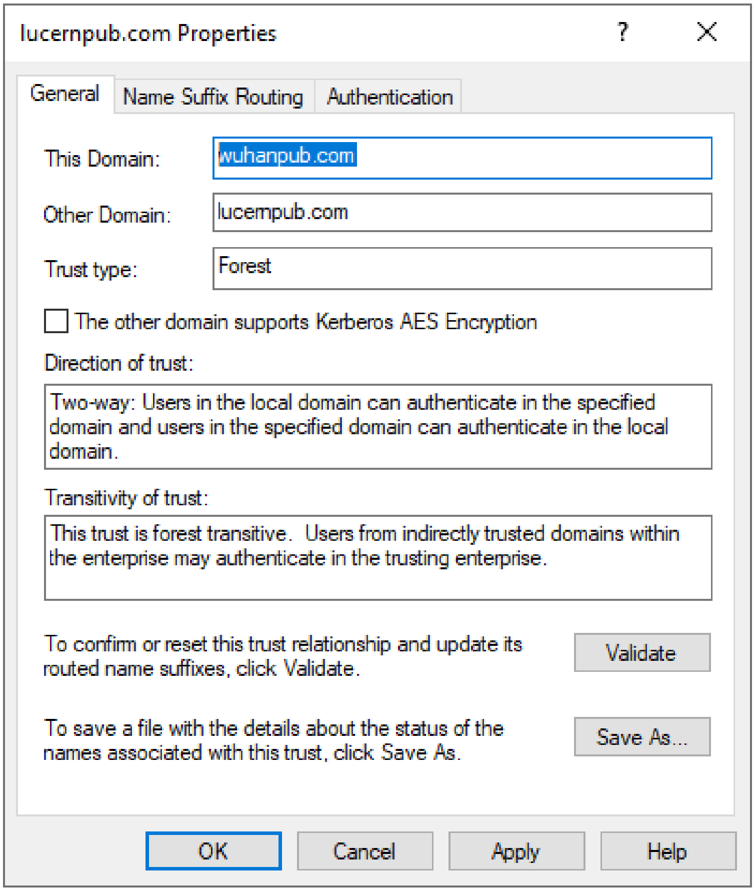
When verifying and resetting a trust, sign in with the credentials of an account that is a member of the **Domain Admins** group. For all other trust types, sign in with the credentials of an account that is a member of the **Enterprise Admins** group.

How to do it...

Perform the following steps:

1. Sign in to the domain controller holding the Domain Naming Master FSMO role.
2. Press Start.
3. Search for **Active Directory Domains and Trusts** and select it from the search results or run `domain.msc`. The **Active Directory Domains and Trusts** window appears.
4. In the console tree, right-click the domain that you want to allow access to, and then click **Properties**.
5. Navigate to the **Trusts** tab.
6. From the list of **Domains trusts by this domain (outgoing trusts)**, or from the list of **Domains that trust this domain (incoming trusts)**, select the trust you want to verify.

7. Click the **Properties** button next to the corresponding list. The **Properties** window for the selected trust appears:



This Domain: wuhanpub.com

Other Domain: lucernpub.com

Trust type: Forest

□ The other domain supports Kerberos AES Encryption

Direction of trust:

Two-way: Users in the local domain can authenticate in the specified domain and users in the specified domain can authenticate in the local domain.

Transitivity of trust:

This trust is forest transitive. Users from indirectly trusted domains within the enterprise may authenticate in the trusting enterprise.

To confirm or reset this trust relationship and update its routed name suffixes, click Validate.

To save a file with the details about the status of the names associated with this trust, click Save As.

Figure 1.7 – The Properties window for an Active Directory trust

8. Click the **Validate** button.

9. For a two-way trust, choose between **No, do not validate the incoming trust** and **Yes, validate the incoming trust**. For the latter, provide the credentials of an account that has administrative privileges in the Active Directory domain on the other side of the trust.

10. Click **OK**.

11. In the **Active Directory Domain Services** pop-up window, click **OK** to confirm that the outgoing trust has been validated. It is now in place and active.

12. In the **Active Directory Domain Services** pop-up window, notifying you of `UserPrincipalName` suffix routing, click **Yes**.

Alternatively, you can use the following command:

```
netdom.exe trust TrustingDomain.tld /Domain:TrustedDomain.tld /
TwoWay /Verify /verbose
```

Replace `TrustingDomain.tld` with the DNS domain name of the Active Directory environment that gives access to its resources, and then replace `TrustedDomain.tld` with the DNS domain name of the Active Directory environment that gains access to the resources.

In the preceding example, a two-way trust is verified.

How it works...

When a trust is verified, the following characteristics of the trust are verified:

- Networking connectivity between both sides of the trust
- Existence of the trust on the far side of the trust
- Synchronization of the shared secret on both sides of the trust

When troubleshooting, verifying a trust is a good place to start because this may be a quick way to identify potential changes made by network admins or Active Directory admins on the other side of the trust.

The option to reset the trust will be presented only if a problem has been identified during the process of verifying the trust relationship.

Securing a trust

Trusts in Active Directory can be misused for purposes not intended by the admin of the trusting domain. There are three ways to secure a trust to make it more secure:

- Enable SID filtering
- Enable quarantine
- Enable selective authentication

SID filtering is enabled on all trust relationships, by default. SID filtering operates on the same surface as trust transitivity. When enabled, SID filtering filters the user accounts over the trust to user accounts from the domain tree that is explicitly trusted, only. In a way, it allows more granular transitivity.

Quarantine is enabled on all trust relationships, by default. Quarantine for a trust allows granular access, too. Where SID filtering allows for limiting access to a trusted domain tree, quarantine limits access to a trusted domain.

Selective authentication is not enabled, by default. Where SID filtering and quarantine limit access to user accounts from trusted domains, selective authentication limits access to devices, member servers, and domain controllers in trusting domains. This means that in a default trust, all resources in the trusting domain can be accessed.

By default, Active Directory trusts are secure since the SID filtering and quarantine features are automatically enabled. You can heighten this default level of security by enabling and managing selective authentication.

Getting ready

To use the selective authentication feature, both Active Directory forests on either side of the trust need to run the Windows Server 2003 FFL, or a higher forest functional level.

It is recommended that you sign in to the domain controller that is running the Domain Naming Master FSMO role or connect the **Active Directory Domains and Trusts** console to this specific domain controller, by right-clicking in the console on the **Active Directory Domains and Trusts** node and selecting **Change Active Directory Domain Controller…** from the context menu.

To find this domain controller, right-click the **Active Directory Domains and Trusts** node and select the **Operations Master…** from the context menu. Alternatively, run the following command from any domain-joined device, member server, or domain controller:

```
netdom.exe query fsmo
```

Otherwise, you can use the following line of Windows PowerShell on a domain-joined system that has the Active Directory module for Windows PowerShell installed:

```
Get-ADForest | Format-List DomainNamingMaster
```

Sign in with the credentials of an admin account that is a member of the **Enterprise Admins** group.

How to do it...

SID filtering and quarantine on trusts can only be managed using `netdom.exe`.

Enable SID filtering

To enable SID filtering for a trust, use the following command:

```
netdom.exe trust TrustingDomain.tld /Domain:TrustedDomain.tld /
EnableSIDHistory:yes
```

Replace `TrustingDomain.tld` with the DNS domain name of the Active Directory environment that gives access to its resources, and then replace `TrustedDomain.tld` with the DNS domain name of the Active Directory environment that gains access to the resources.

Disable SID filtering

To disable SID filtering for a trust, use the following command:

```
netdom.exe trust TrustingDomain.tld /Domain:TrustedDomain.tld /
EnableSIDHistory:no
```

Replace `TrustingDomain.tld` with the DNS domain name of the Active Directory environment that gives access to its resources, and then replace `TrustedDomain.tld` with the DNS domain name of the Active Directory environment that gains access to the resources.

Enable quarantine

To enable quarantine on a trust, use the following command:

```
netdom.exe trust TrustingDomain.tld /Domain:TrustedDomain.tld /
Quarantine:yes
```

Replace `TrustingDomain.tld` with the DNS domain name of the Active Directory environment that gives access to its resources, and then replace `TrustedDomain.tld` with the DNS domain name of the Active Directory environment that gains access to the resources.

Disable quarantine

To disable quarantine on a trust, use the following command:

```
netdom.exe trust TrustingDomain.tld /Domain:TrustedDomain.tld /
Quarantine:no
```

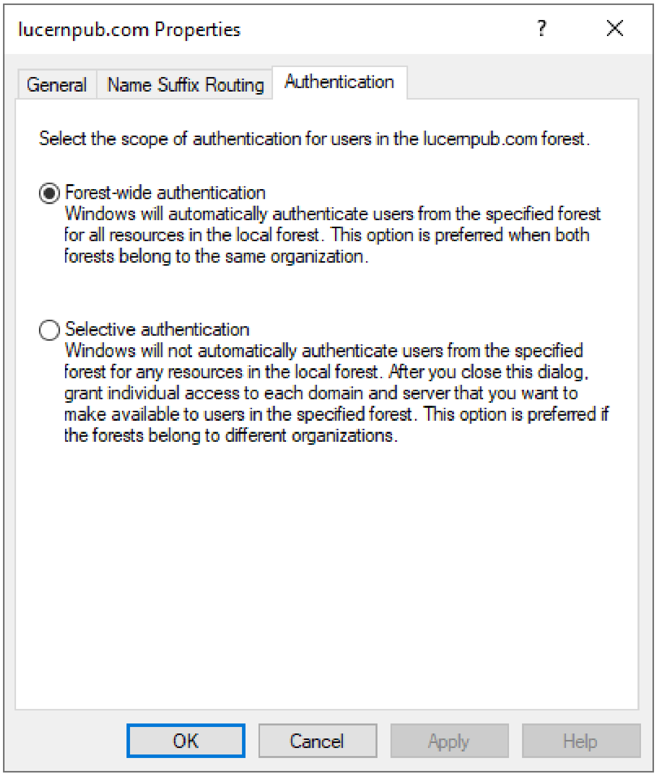
Replace `TrustingDomain.tld` with the DNS domain name of the Active Directory environment that gives access to its resources, and then replace `TrustedDomain.tld` with the DNS domain name of the Active Directory environment that gains access to the resources.

Manage selective authentication

To manage selective authentication, we can use the **graphical user interface** (**GUI**). To do so, perform these steps:

1. Press Start.
2. Search for **Active Directory Domains and Trusts** and select it from the search results or run `domain.msc`. The **Active Directory Domains and Trusts** window appears.
3. In the console tree, right-click the domain that you want to configure selective authentication for, and then click **Properties**.
4. Navigate to the **Trusts** tab.
5. From the list of **Domains trusts by this domain (outgoing trusts)**, or from the list of **Domains that trust this domain (incoming trusts)**, select the trust that you want to configure selective authentication for.
6. Click the **Properties** button next to the corresponding list.

7. Navigate to the **Authentication** tab as follows:



Select the scope of authentication for users in the lucernpub.com forest.

● Forest-wide authentication
Windows will automatically authenticate users from the specified forest for all resources in the local forest. This option is preferred when both forests belong to the same organization.

○ Selective authentication
Windows will not automatically authenticate users from the specified forest for any resources in the local forest. After you close this dialog, grant individual access to each domain and server that you want to make available to users in the specified forest. This option is preferred if the forests belong to different organizations.

Figure 1.8 – The Authentication tab of the properties of an Active Directory trust

8. On the **Authentication** tab, choose between **Forest-wide authentication** and **Selective authentication**.
9. Click **OK** to finish.

Of course, selective authentication for trusts is also available on the command line.

To enable selective authentication for a trust, use the following command:

```
netdom.exe trust TrustingDomain.tld /Domain:TrustedDomain.tld /
SelectiveAuth:yes
```

Replace `TrustingDomain.tld` with the DNS domain name of the Active Directory environment that gives access to its resources, and then replace `TrustedDomain.tld` with the DNS domain name of the Active Directory environment that gains access to the resources.

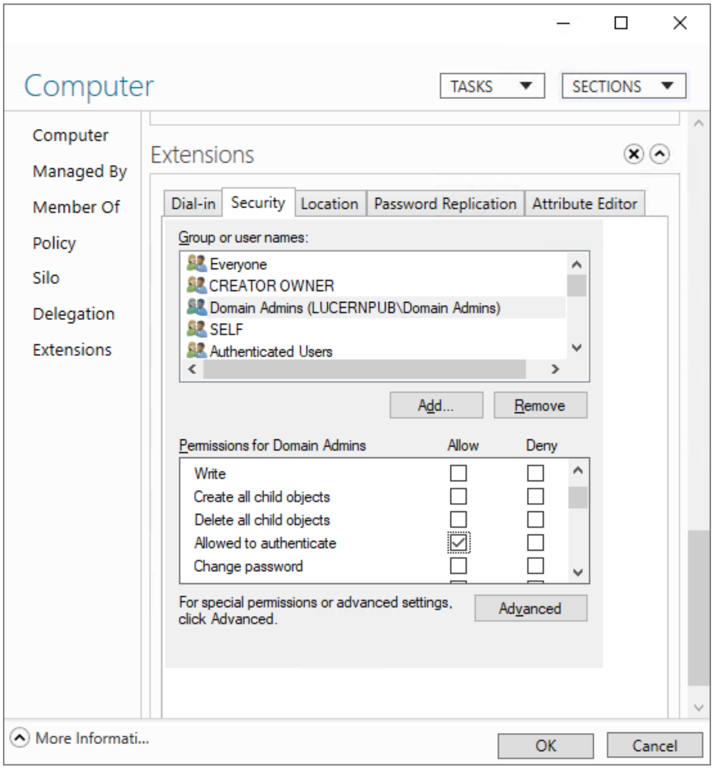
To disable selective authentication for a trust, use the following command:

```
netdom.exe trust TrustingDomain.tld /Domain:TrustedDomain.tld /
SelectiveAuth:no
```

Replace `TrustingDomain.tld` with the DNS domain name of the Active Directory environment that gives access to its resources, and then replace `TrustedDomain.tld` with the DNS domain name of the Active Directory environment that gains access to the resources.

Now, the actual domain-joined resources, which a user from another domain or forest has access to, are governed per object. Perform these steps to manage this setting:

1. Press Start.
2. Search for the **Active Directory Administrative Center** and select it from the search results or run `dsac.exe`. The **Active Directory Administrative Center** window appears.
3. Search for the domain-joined device, member server, or domain controller that you want to grant access to over the trust. Use the search box in the **Global Search** field on the **Overview** screen of **Active Directory Administrative Center** or use the left navigation pane.
4. Right-click the object and select **Properties** from the context menu.
5. In the left navigation pane of the object's properties, click **Extensions**.
6. Click the **Security** tab.
7. Select the user object(s) and/or group(s) that you want to grant access to, using the **Add…** and **Remove** buttons underneath the field for **Groups and user names**:



Group or user names:

- Everyone
- CREATOR OWNER
- Domain Admins (LUCERNPUB\Domain Admins)
- SELF
- Authenticated Users

Permissions for Domain Admins	Allow	Deny
Write	□	□
Create all child objects	□	□
Delete all child objects	□	□
Allowed to authenticate	☑	□
Change password	□	□

For special permissions or advanced settings, click Advanced.

Figure 1.9 – The Security tab of a computer object in the Active Directory Administrative Center

8. Select the **Allow** checkbox that is next to the **Allowed to Authenticate** permission.
9. Click **OK** when you're done.

How it works...

Selective authentication leverages the **Allowed to Authenticate** option to give permission to allow or disallow requests coming from user accounts over the trust because they are automatically added to the Authenticated Users group. When selective authentication is disabled (the default), every user account on the other side is allowed to authenticate. However, after selective authentication is enabled, only the user accounts with the **Allowed to Authenticate** permission explicitly set can authenticate to it over the trust, because they are not automatically added to the Authenticated Users group.

There's more…

To make managing the Active Directory trust possible for a trust that has selective authentication enabled, ensure that admins on both sides have the **Allowed to Authenticate** permission on each other's domain controllers. You can specify specific domain controllers only by modifying the DNS SRV records for domain controllers. However, ensure that you always include the domain controller holding the PDC Emulator FSMO role, and at least one global catalog.

Extending the schema

Some applications require additional object types and/or attributes to store their information in Active Directory. Some good examples of these types of applications are Microsoft Exchange Server and Microsoft's free **Local Administration Password Solution** (**LAPS**).

These applications and their schema changes are thoroughly tested, but there's also the option to create your own custom Active Directory schema extension. For instance, you can introduce your own employee or customer ID type attribute to the user object class.

Getting ready

The domain controller holding the Schema Master FSMO role is authoritative for the Active Directory schema throughout an Active Directory forest. Microsoft recommends that you perform the following actions on the domain controller that is holding the Schema Master FSMO role.

To find this domain controller, run the following command on any domain-joined device, member server, or domain controller:

```
netdom.exe  query fsmo
```

Alternatively, use the following line of Windows PowerShell on a domain-joined system that has the Active Directory module for Windows PowerShell installed:

```
Get-ADForest | Format-List SchemaMaster
```

To gain access to the Active Directory schema using the **Microsoft Management Console (MMC)**, the Schema MMC snap-in needs to be registered on the domain controller holding the Schema Master FSMO role. By default, this MMC snap-in is hidden from view, due to its sensitive nature.

Run the following command to register the Schema MMC snap-in:

```
regsvr32.exe C:\windows\system32\schmmgmt.dll
```

To extend the schema, perform the following actions using an account that is a member of the **Schema Admins** group.

To request permission to use an **official identifier** (**OID**) for your schema attribute or object, you will need to create it as part of your organization's OID branch. The following two websites allow you to view whether your organization has an official OID branch: `www.iana.org/assignments/enterprise-numbers` and `www.alvestrand.no/objectid/`.

If your organization does not have an assigned OID, go to your country's national registry to request one. Ensure that the registration is correct, but also leave room for further expansions, relocations, mergers, acquisitions, and divestitures.

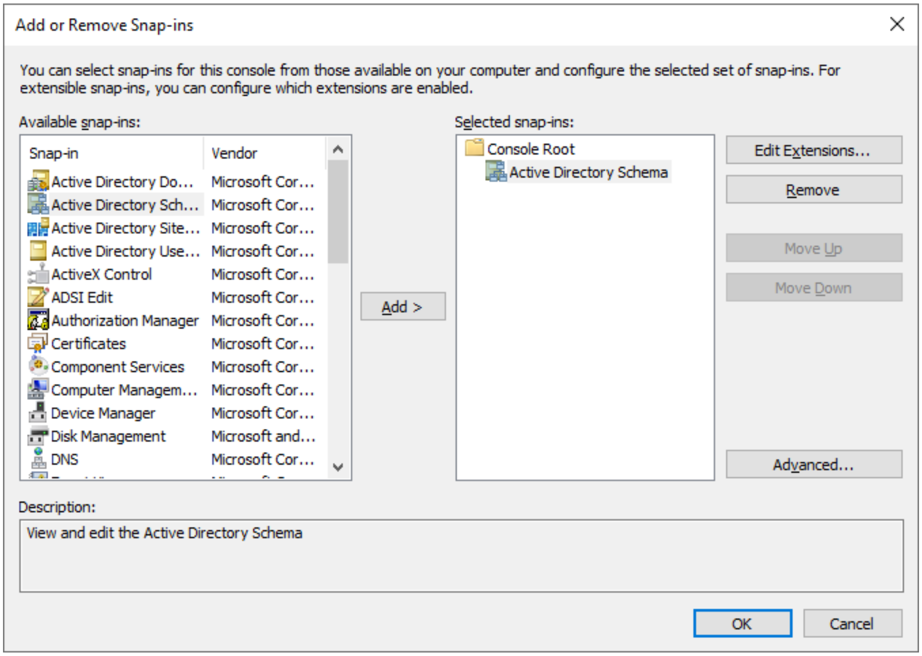
OIDs are hierarchical, so you should create them as part of your organization's branch.

How to do it...

Perform the following steps to extend the Active Directory schema with a new attribute:

1. Press Start.
2. Search for **Microsoft Management Console** and select it from the search results or run `mmc.exe`. The **Microsoft Management Console** window appears.
3. From the **File** menu, select **Add/Remove Snap-in**.

4. From the left-hand list of **Available snap-ins**, select the **Active Directory Schema** snap-in. Click the **Add >** button to add it to the right-hand list of **Selected snap-ins**:



Add or Remove Snap-ins

You can select snap-ins for this console from those available on your computer and configure the selected set of snap-ins. For extensible snap-ins, you can configure which extensions are enabled.

Available snap-ins:

Snap-in	Vendor
Active Directory Do...	Microsoft Cor...
Active Directory Sch...	Microsoft Cor...
Active Directory Site...	Microsoft Cor...
Active Directory Use...	Microsoft Cor...
ActiveX Control	Microsoft Cor...
ADSI Edit	Microsoft Cor...
Authorization Manager	Microsoft Cor...
Certificates	Microsoft Cor...
Component Services	Microsoft Cor...
Computer Managem...	Microsoft Cor...
Device Manager	Microsoft Cor...
Disk Management	Microsoft and...
DNS	Microsoft Cor...

Selected snap-ins:

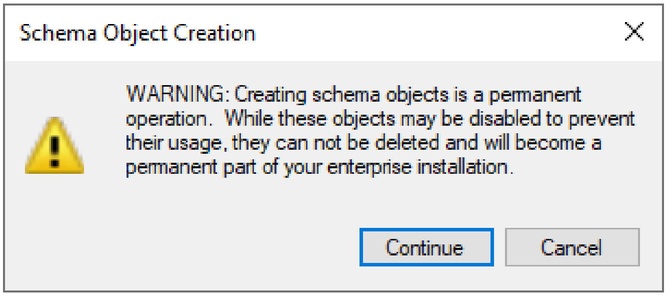
- Console Root
  - Active Directory Schema

Description:

View and edit the Active Directory Schema

Figure 1.10 – Add or remove snap-ins for the MMC

5. Click **OK**.
6. In the left navigation pane, expand **Active Directory Schema**.
7. Right-click the **Attributes** folder and select **Create Attribute** from the context menu. The **Schema Object Creation** pop-up window appears:

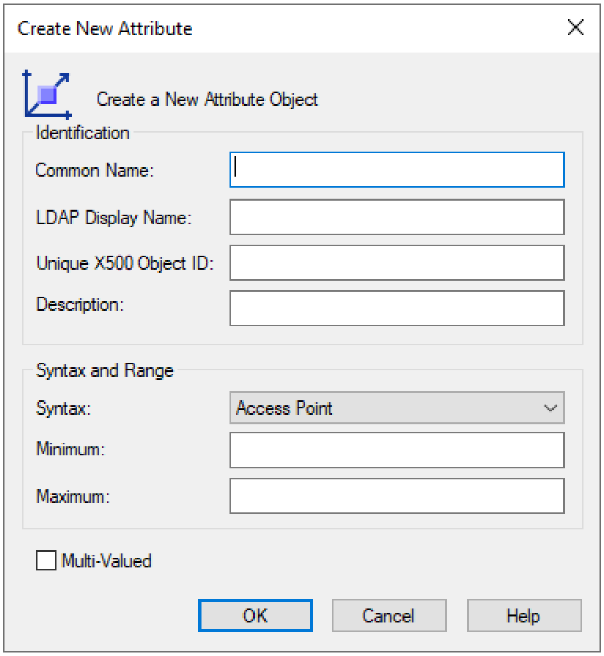


Schema Object Creation

WARNING: Creating schema objects is a permanent operation. While these objects may be disabled to prevent their usage, they can not be deleted and will become a permanent part of your enterprise installation.

Figure 1.11 – The Schema Object Creation pop-up window

8. Click the **Continue** button to confirm that you want to extend the schema. The **Create New Attribute** window appears:



Create New Attribute

Create a New Attribute Object

Identification

Common Name:

LDAP Display Name:

Unique X500 Object ID:

Description:

Syntax and Range

Syntax: Access Point

Minimum:

Maximum:

□ Multi-Valued

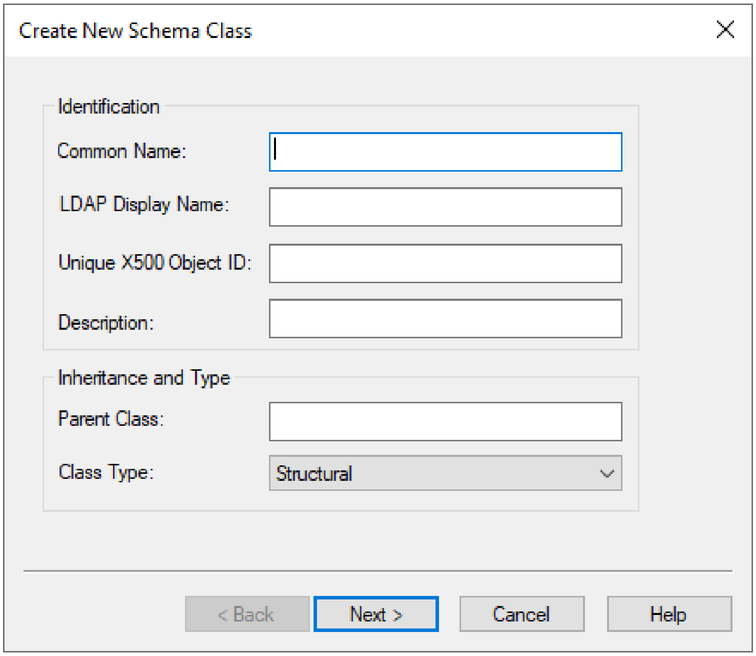
Figure 1.12 – The Create New Attribute window

9. Enter the information for the new attribute.
10. Click **OK** when done.

Perform these steps to extend the Active Directory schema with a new object class:

1. Press Start.
2. Search for **Microsoft Management Console** and select it from the search results or run `mmc.exe`. The **Microsoft Management Console** window appears.
3. From the **File** menu, select **Add/Remove Snap-in**.
4. From the left-hand list of **Available snap-ins**, select the **Active Directory Schema** snap-in. Click the **Add >** button to add it to the right-hand list of **Selected snap-ins**.
5. Click **OK**.
6. In the left navigation pane, expand **Active Directory Schema**.

7. Right-click the **Classes** folder and select **Create Class** from the context menu.
8. Click the **Continue** button to confirm that you want to extend the schema. The **Create New Schema Class** window appears:



Class Type: Structural

Figure 1.13 – The Create New Schema Class window

9. Enter the information for the new class.
10. Click **Next** > when done.
11. Enter any mandatory and optional attributes and click **Finish**.

There's more...

Any new schema additions are permanent and cannot be removed. However, you can disable an existing class or attribute by marking it as defunct instead when it is no longer needed. It's a best practice to keep the **Schema Admins** group in Active Directory as empty as possible for as long as possible. When you're done, remove any accounts that you may have added to the **Schema Admins** group.

If you want your Active Directory schema extension attributes to extend into **Azure Active Directory** (**AD**) as well, ensure it's a single-valued attribute.

Enabling the Active Directory Recycle Bin

The **Active Directory Recycle Bin** was introduced as a new Active Directory feature with Windows Server 2008 R2. It enables administrators to restore (accidentally) deleted objects.

There were features available to administrators before the advent of the Active Directory Recycle Bin, such as **Directory Services Restore Mode** (**DSRM**) and object reanimation. In contrast to booting into DSRM, the Active Directory Recycle Bin saves admins time. In contrast to reanimating objects, the Active Directory Recycle Bin prevents the typical loss of attributes and group memberships.

There are also numerous third-party solutions that are available to restore objects and their attributes. They typically expand on the functionality that is offered by the Active Directory Recycle Bin, by offering granular attribute restore and group policy versioning. These are two areas where the Active Directory Recycle Bin doesn't offer a solution.

Getting ready

The Active Directory forest needs to run the Windows Server 2008 R2 FFL (or a later version).

Microsoft recommends enabling the Active Directory Recycle Bin on the Active Directory domain controller that holds the Domain Naming Master FSMO role.

To find this domain controller, run the following command on any domain-joined device, member server, or domain controller:

```
netdom.exe query fsmo
```

Alternatively, use the following line of Windows PowerShell on a domain-joined system that has the Active Directory module for Windows PowerShell installed:

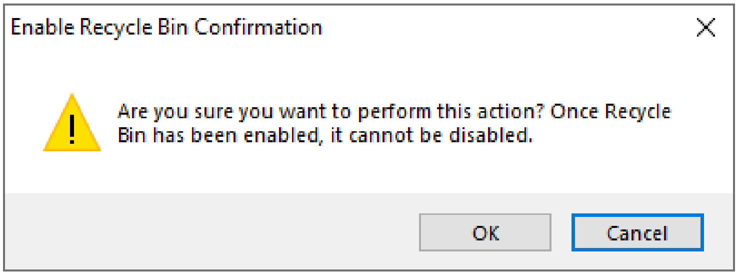
```
Get-ADForest | Format-List DomainNamingMaster
```

Sign in to the preceding domain controller using an account that is a member of the **Enterprise Admins** group in Active Directory.

How to do it...

You can enable the Active Directory Recycle Bin from within the Active Directory Administrative Center when you're signed in with an account that is a member of the **Enterprise Admins** group on a domain controller that runs Windows Server with Desktop Experience. To do this, perform the following steps:

1. Press Start.
2. Search for **Active Directory Administrative Center** and select it from the search results or run `dsac.exe`.
3. Select the forest name in the left navigation pane.
4. In the action pane on the right, click the **Enable Recycle Bin** link. Alternatively, you can right-click the domain name in the left navigation pane and select the **Enable Recycle Bin…** option from the context menu. The **Enable Recycle Bin Confirmation** pop-up window appears:

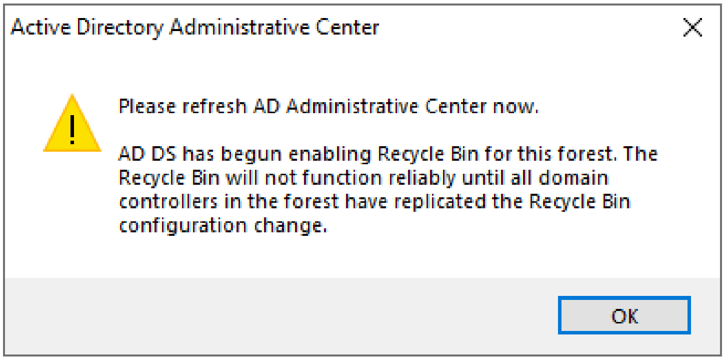


Enable Recycle Bin Confirmation

Are you sure you want to perform this action? Once Recycle Bin has been enabled, it cannot be disabled.

Figure 1.14 – The Enable Recycle Bin Confirmation pop-up window

5. In the **Enable Recycle Bin Confirmation** pop-up window, click **OK**. The pop-up message labeled **Active Directory Administrative Center** appears:



Active Directory Administrative Center

Please refresh AD Administrative Center now.

AD DS has begun enabling Recycle Bin for this forest. The Recycle Bin will not function reliably until all domain controllers in the forest have replicated the Recycle Bin configuration change.

Figure 1.15 – The Active Directory Administrative Center pop-up window

6. Click **OK**.
7. After you refresh, a new container underneath the domain root named **Deleted Objects** appears:

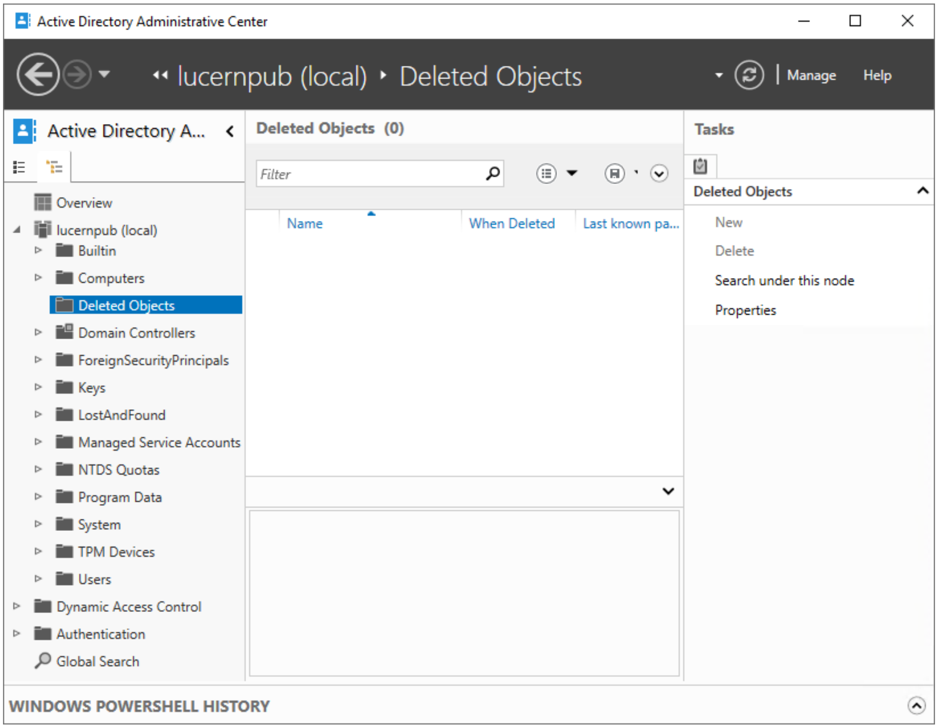


Figure 1.16 – The Deleted Objects container

On Server Core domain controllers, use the following PowerShell commands:

```
Enable-ADOptionalFeature -Identity "CN=Recycle Bin
Feature,CN=Optional Features,CN=Directory Service,CN=Windows
NT,CN=Services,CN=Configuration,DC=LucernPub,DC=com" -Scope
ForestOrConfigurationSet -Target "lucernpub.com"
```

Replace `LucernPub`, `lucernpub`, and `com` with values for your Active Directory environment.

How it works...

Since the inception of Active Directory, when an object such as a computer or a user is deleted, the `isDeleted` attribute is set to `true`. This allows the domain controller to replicate the change for the object. Each domain controller has the time configured as the tombstone lifetime period to replicate this change.

The Active Directory Recycle Bin introduces a new recycle lifetime and a new attribute: `isRecycled`. With the Active Directory Recycle Bin enabled, when an object is deleted, its `isDeleted` attribute is still set to `true`, but its `isRecycled` attribute is untouched. This is the period where the object is visible in the **Deleted Objects** container, where it can be restored by simply right-clicking on it. After the recycle lifetime has expired, the `isRecycled` attribute is also set to `true`. This is when the tombstone lifetime kicks in. Only after the tombstone lifetime period has expired is the object removed from the database by each domain controller.

Managing UPN suffixes

In Active Directory, users and services can sign in using their pre-Windows 2000 logon name (the value of the `sAMAccountName` attribute) or their Kerberos user principal name (the value of the `userPrincipalName` attribute). As Kerberos relies heavily on DNS, the user principal name features a `userPrincipalName` suffix, in the form of a DNS domain name.

These `userPrincipalName` suffixes can be added to the list of available UPN suffixes for each Active Directory forest.

By default, this list already contains the DNS domain names of the Active Directory domains in the forest.

UPN suffixes in on-premises Active Directory environments do not need to be publicly routable. Only if you intend to use them with federation and/or hybrid identity do they then need to be. In many organizations, a cloud journey begins with changing the UPN suffix on all the user objects that need to be cloud-enabled to a publicly routable UPN suffix. Some organizations have adopted `.local` as their top-level domain name, and this is the prime example of a non-publicly routable top-level domain name.

Getting ready

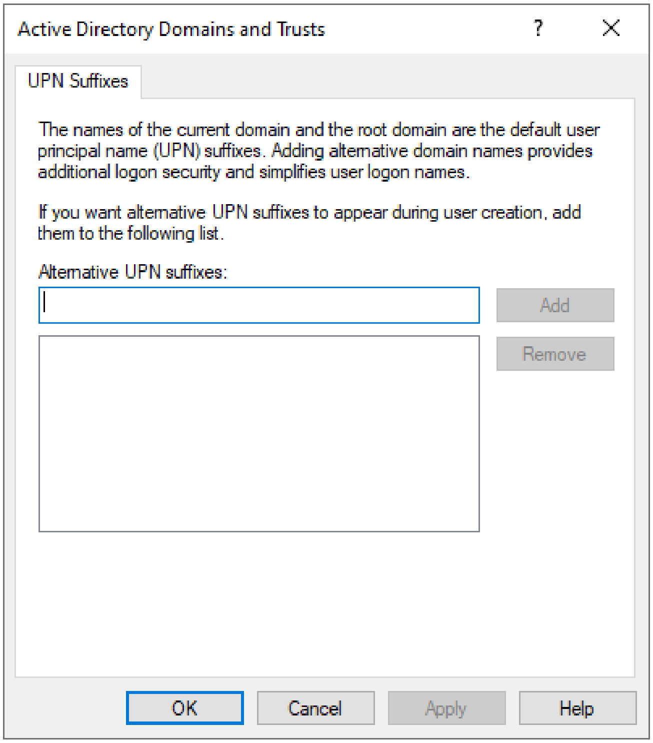
To make the most of UPN suffixes, ensure that you have an overview of the domain names and the publicly registered domain names for the organization.

How to do it…

UPN suffixes can be managed using the **Active Directory Domains and Trusts** console.

To do so, perform the following steps:

1. Press Start.
2. Search for **Active Directory Domains and Trusts** and select it from the search results or run `domain.msc`. The **Active Directory Domains and Trust** window appears.
3. Right-click **Active Directory Domains and Trusts** in the left navigation pane, and select **Properties** from the context menu:



Active Directory Domains and Trusts

UPN Suffixes

The names of the current domain and the root domain are the default user principal name (UPN) suffixes. Adding alternative domain names provides additional logon security and simplifies user logon names.

If you want alternative UPN suffixes to appear during user creation, add them to the following list.

Alternative UPN suffixes:

Figure 1.17 – Properties of Active Directory Domains and Trusts

4. Type the new UPN suffix that you would like to add to the Active Directory forest, select the UPN suffix that you would like to remove, or simply glance over the list of UPN suffixes.
5. Click **Add** or **Remove** when your goal is to add or remove a UPN suffix.
6. Click **OK** to save the changes and close the **Properties** window.

How it works...

Both `userPrincipalName` and pre-Windows 2000 logon names can be used to sign in interactively to Windows. The Windows sign-in screen can handle both. However, in cloud scenarios, the `userPrincipalName` suffix is used, by default.

There's more...

An admin cannot assign a non-existing `userPrincipalName` suffix. However, when a UPN suffix is assigned to one or more accounts, there's no alert that mentions this when you remove a UPN suffix.

2 Managing Domain Controllers

Active Directory domain controllers are your network's castles of identity. They offer services such as LDAP, Kerberos, and NTLM to people using devices, appliances, and servers. The previous chapter introduced the concepts of forests, trees, and domains. In this chapter, we are going to look at some more tangible things, such as server machines. But please don't take that literally; domain controllers these days are found to be virtual machines more often than physical machines.

I'll walk you through creating new domain controllers and show you which type of domain controller to implement. I'll also show you how to create domain controllers quickly, even when there's only a slow connection between the location with existing domain controllers and the location where you want to implement a new domain controller. We'll also look at creating hundreds of domain controllers quickly.

The following recipes are covered in this chapter:

- Preparing a Windows server to become a domain controller
- Promoting a server to a domain controller
- Using **Install From Media** (**IFM**)
- Using domain controller cloning

- Determining whether a virtual domain controller has a VM-GenerationID
- Demoting a domain controller
- Demoting a domain controller forcefully
- Inventory domain controllers
- Decommissioning a compromised read-only domain controller

Preparing a Windows server to become a domain controller

To make Active Directory a reliable service in any networking environment, the domain controllers need to be available with high integrity. Any changes an administrator needs to make to a deployed domain controller might diminish the availability. Any component or configuration that is misbehaving might diminish the integrity. Therefore, let's look at how to prepare a Windows Server installation to become a domain controller before we promote it to become one.

The following steps are my time-tested recommended practices for production domain controllers within enterprises. I highly recommend these steps to create highly reliable domain controllers.

Intending to do the right thing

The first few items on the list of preparations involve having the right ideas about promoting domain controllers throughout their life cycles:

- **Intend to create at least two domain controllers per Active Directory domain**: This way, both servers can be advertised to networking clients as LDAP servers and DNS servers. Then, when you have to reboot one of at least two servers, these clients won't be hindered. Also, restoring a domain controller while another domain controller is still available allows for scenarios such as non-authoritative restores, domain controller cloning, and domain controller re-promotion.
- **Intend to implement role separation**: By any means, do not misuse a domain controller as an exchange server or SQL server, unless it's a Windows Small Business Server. The DNS server, DHCP server, and NPS server roles are gray areas here, which should be addressed with common sense; if it means a domain controller will be harder to restore, manage, or decommission, separate the roles.

Dimensioning the servers properly

Now, let's look at how to dimension intended domain controllers:

- **Intend to create equal domain controllers in terms of hardware dimensions**: It's tempting to place one big server and one smaller server as domain controllers but consider the possibility of having to move **Flexible Single Master Operation** (**FSMO**) roles or other loads from one domain controller to another. Since domain controllers are randomly assigned to networking clients inside an Active Directory site, clients accessing a server with fewer resources may not exhibit a consistent and acceptable user experience.
- **Dimension the intended domain controllers properly in terms of hardware**: Domain controllers offer the best performance when they can cache the Active Directory database, `ntds.dit`, in RAM. Plan for ample room in RAM to cache up to 4 KB per Active Directory object, plus a 10 MB minimum for the main objects and partitions. You should start with the minimum RAM required to install Windows Server and then add on the additional memory for **Active Directory Domain Services** (**AD DS**). For physical servers, use RAID and separate spindles for storage of Active Directory-related data when possible. Use hardware that will be covered by the manufacturer's (extended) guarantee, support, and life cycle policies for the period in which you need to rely on the domain controller.
- **Dimension the intended domain controllers properly in terms of software**: Use a version of Windows Server that will be covered by Microsoft's (extended) support and life cycle policies for the period in which you need to rely on the domain controller.
- **Implement the Server Core version of Windows Server when possible**: Server Core installations of Windows Server offer higher availability and a smaller attack surface compared to Windows Server installations with the Desktop Experience feature. However, some agents and other software components in use within the organization might not properly run on Server Core installations. In the latter scenario, Windows Server installations with the Desktop Experience feature (called Full Installations in previous versions of Windows Server) should be performed, obviously.
- **Install the latest firmware for devices and/or integration components**: On physical boxes that you intend to use as a domain controller, install the latest stable firmware for the **Basic Input/Output System** (**BIOS**), the storage controller(s), the video card(s), and **Network Interface Card(s)** (**NIC(s)**). On virtual machines, implement the latest stable version of the integration components or VMware tools and follow the recommended practices from the vendor of the hypervisor platform.

- **Use a virtualization platform that offers the VM-GenerationID feature**: Place virtual domain controllers on a virtualization platform that offers the VM-GenerationID feature. This will offer the domain controller virtualization safeguards that allow administrators to take snapshots of domain controllers without compromising the integrity of the Active Directory database. Also, domain controller cloning is available on these virtualization platforms.

Preparing the Windows Server installations

Before you install Windows Server on intended domain controllers, perform these actions:

- **Run the memory diagnostics from the Windows Server installation media**: The Windows Server DVD allows administrators to check the RAM of physical and virtual machines to ensure that the memory used by the Windows Server installation is not faulty. Checking beforehand means you don't have to replace faulty memory after going live.
- **Run sysprep.exe on cloned disks**: When the Windows Server installation is the result of the cloning of a disk with a Windows Server installation on it, ensure that the Windows Server installation is sysprepped. You wouldn't want the **Security Identifier** (**SID**) on the cloned disk to become the SID for the new Active Directory forest or domain you might be creating.

Preconfiguring the Windows servers

After Windows Server is installed, configure these items on the Windows Server instance, either through Server Manager on Windows Server installations with the Desktop Experience feature or by using `sconfig.cmd` on Server Core installations:

- Change the hostname for the Windows Server installation. Leverage the server naming convention and/or policy within the organization.
- Check for proper Windows activation of the Windows Server operating system.
- Update the Windows Server installation with the latest updates.
- Configure the server with at least one static IPv4 address and/or a static IPv6 address. Leverage the networking plan and zone assignment policies within the organization. Avoid multi-homing domain controllers.

> **Tip**
>
> When the intended domain controller is to run as a virtual machine within a cloud environment, such as Amazon's AWS or Microsoft's Azure, let the cloud provider assign the IPv4 and/or IPv6 addresses, because manually setting these addresses might break the connectivity of the Windows Server installation. Instead, use IP address reservations to ensure the intended domain controllers remain reachable over the same addresses.

- **Check for at least one connected LAN connection**: Without a connected LAN connection, the promotion of a domain controller will fail. This is by design.
- **Configure a proper naming resolution**: As the DNS plays a vital role in locating Active Directory, ensure the DNS is properly configured. Plan for an Active Directory-integrated DNS. Don't forget the DNS stub zones and/or conditional forwarders when creating a new Active Directory domain and/or forest. Deploy **Windows Internet Name Service** (**WINS**) or DNS GlobalNames zones in legacy environments.
- Configure the page file correctly.
- **Implement information security measures**: Deploy agents for anti-malware, uninterruptible power supplies, backup and restore, **Security Incident and Event Management** (**SIEM**), **Technology State and Compliance Monitoring** (**TSCM**), advanced threat analytics, and other information security measures your organization's policies might require.

Documenting the passwords

In large organizations, you can't get anything done without the proper changes being filed through change management. Even if your organization doesn't require these steps, it's still a recommended practice to document at least these items:

- **Document the password for the built-in administrator account**: When deploying a new Active Directory forest or domain, deploy using a pre-configured password for the built-in administrator account. After successful promotion, change the password to one that you intend to assign to this account for a longer period. Document the latter password in a password vault.

> **Tip**
>
> As domain controllers are promoted using scripts, there is a chance the password for the built-in account will linger around unintentionally. Also, the password initially set for this account is stored with a weaker hashing algorithm than changed passwords.

- **Document the Directory Services Restore Mode (DSRM) password**: In dire situations, when the Active Directory-related services are no longer able to start, an administrator can sign in to the server using a fallback account with the DSRM password. Intend to use different DSRM passwords for each domain controller and document these properly in a password vault.

See also

See the *Creating conditional forwarders* recipe in *Chapter 9*, *Managing DNS,* to create conditional forwarders.

Promoting a server to a domain controller

Promoting a Windows Server installation to a domain controller consists of three steps:

- Installing the **Active Directory Domain Services** role
- Promoting a server to a domain controller
- Checking proper promotion

When using `dcpromo.exe`, you do not have to install the role beforehand.

You can promote the server in several ways. The following table displays the possibilities:

Method	**Install the role**	**Promote the server**
Using Server Manager	Yes	No
Using the `Install-WindowsFeature` cmdlet	Yes	No
Using Windows Admin Center	Yes	No
Using the Active Directory Domain Services Configuration Wizard	Required	Yes
Using the `Install-ADDSDomainController`, `Install-ADDSDomain`, or `Install-ADDSForest` cmdlets from the Active Directory module for Windows PowerShell	Required	Yes
Using `dcpromo.exe` with an answer file	Not necessary	Yes

Table 2.1 – Methods for installing the Active Directory Domain Services role and promoting a server to a domain controller

The methods in the table are all explained in more detail in this recipe.

Getting ready

In some organizations, changes can only be made using scripts and must be accompanied by rollback scripts. In these cases, the answer file and PowerShell cmdlets offer the best method. On Server Core installations of Windows Server, only the last two options are available to promote the server, either on the Command Prompt or through Windows Admin Center, unless you use Server Manager to remotely manage the server you intend to promote to a domain controller.

The Active Directory Domain Services Configuration Wizard no longer features the option to not reboot the Windows Server installation intended as a domain controller after successful promotion. If you need this option – for instance, to harden the domain controller before the first boot with custom scripts – then you can't use the Wizard. Use `dcpromo.exe` or the `Install-DDSDomainController`, `Install-ADDSDomain`, or `Install-ADDSForest` cmdlets in these cases.

When creating an additional domain controller in an existing Active Directory domain or forest, check for proper Active Directory replication before implementing the new domain controller.

How to do it…

Unless you are using `dcpromo.exe` to promote the Windows Server installation to a domain controller, the **Active Directory Domain Services** role needs to be installed first.

Installing the Active Directory Domain Services role

There are three ways to install the **Active Directory Domain Services** role:

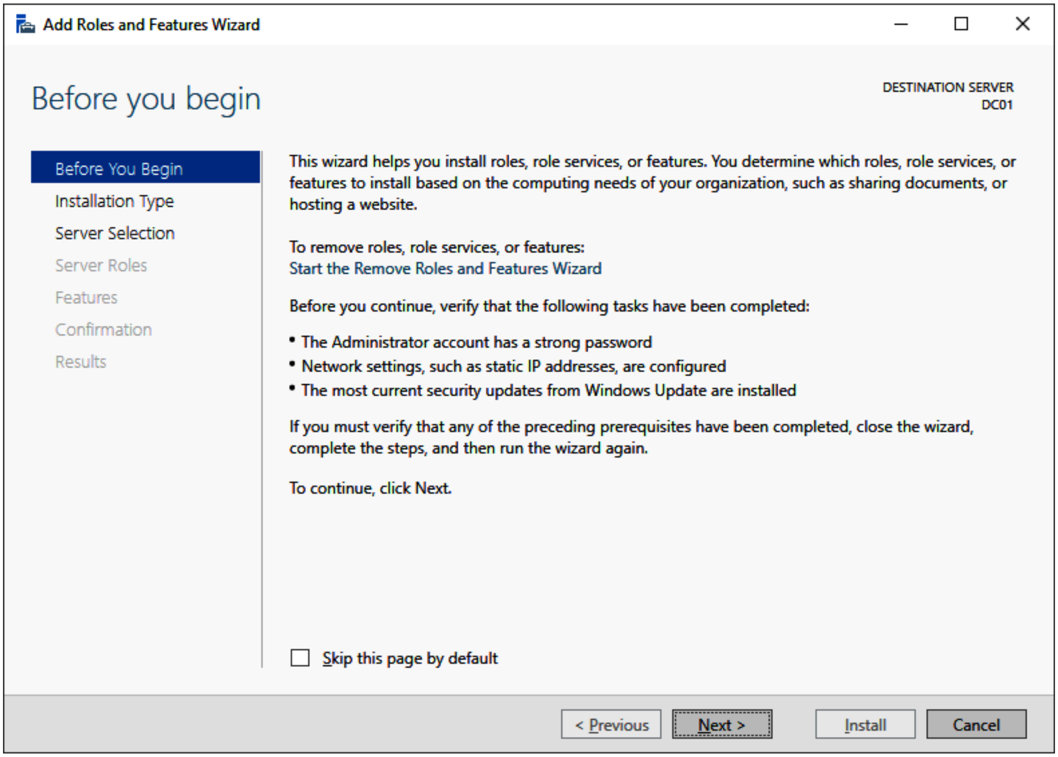
- Using **Server Manager**
- Using the `Install-WindowsFeature` cmdlet
- Using Windows Admin Center

Using Server Manager

To install the **Active Directory Domain Services** role using **Server Manager**, perform these steps:

1. Press Start.
2. Search for **Server Manager** and click its corresponding search result or run `servermanager.exe`. The **Server Manager** window appears.

3. In the gray top bar of **Server Manager**, click **Manage**.
4. Select **Add Roles and Features** from the menu. The **Before you begin** screen appears, as shown in the following screenshot:



Add Roles and Features Wizard

Before you begin

DESTINATION SERVER
DC01

This wizard helps you install roles, role services, or features. You determine which roles, role services, or features to install based on the computing needs of your organization, such as sharing documents, or hosting a website.

To remove roles, role services, or features:
Start the Remove Roles and Features Wizard

Before you continue, verify that the following tasks have been completed:

- The Administrator account has a strong password
- Network settings, such as static IP addresses, are configured
- The most current security updates from Windows Update are installed

If you must verify that any of the preceding prerequisites have been completed, close the wizard, complete the steps, and then run the wizard again.

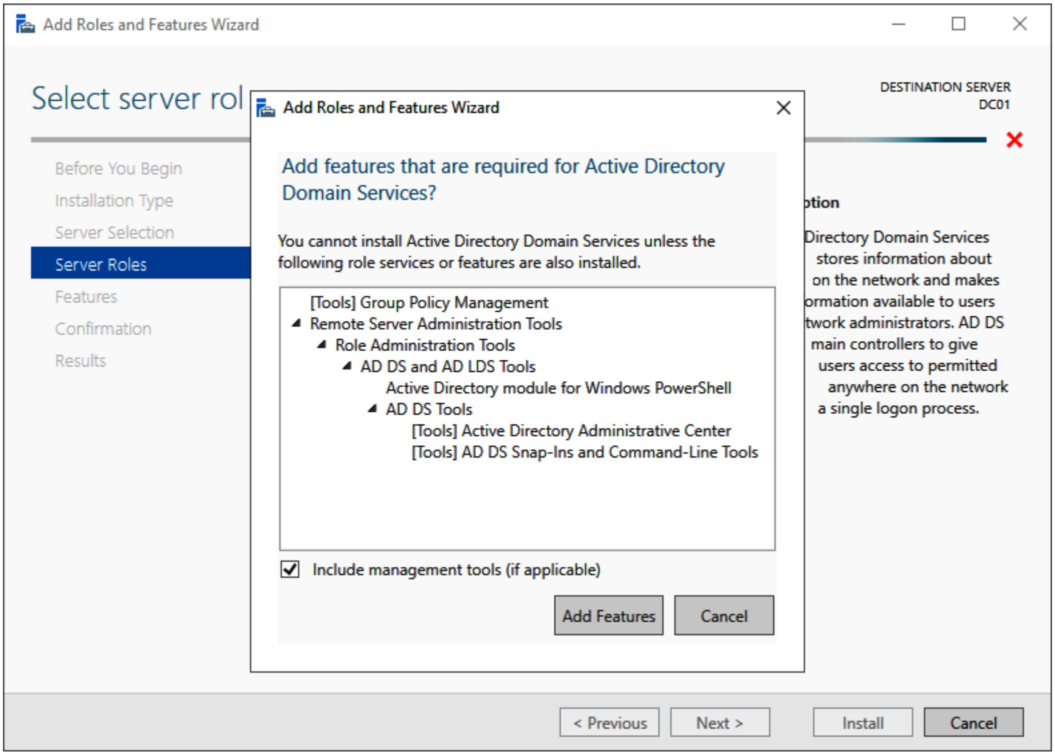
To continue, click Next.

□ Skip this page by default

Figure 2.1 – The Before you begin screen of the Add Roles and Features Wizard

5. On the **Before you begin** screen, click **Next** >.
6. On the **Select installation type** screen, select **Role-based or feature-based installation** and click **Next** >.
7. On the **Select destination server** screen, select either the local Windows Server installation from the server pool list, the remote Windows Server installation you intend to promote to the domain controller from the server pool list, or both types of resources.
8. Click **Next** >.

9. On the **Select server roles** screen, select the **Active Directory Domain Services** role from the list of available roles. The **Add Roles and Features Wizard** pop-up window appears, as shown in the following screenshot:



ption

Directory Domain Services stores information about on the network and makes ormation available to users twork administrators. AD DS main controllers to give users access to permitted anywhere on the network a single logon process.

Add features that are required for Active Directory Domain Services?

You cannot install Active Directory Domain Services unless the following role services or features are also installed.

```
[Tools] Group Policy Management
◢ Remote Server Administration Tools
    ◢ Role Administration Tools
        ◢ AD DS and AD LDS Tools
              Active Directory module for Windows PowerShell
            ◢ AD DS Tools
                  [Tools] Active Directory Administrative Center
                  [Tools] AD DS Snap-Ins and Command-Line Tools
```

☑ Include management tools (if applicable)

Figure 2.2 – The Add Roles and Features Wizard pop-up window

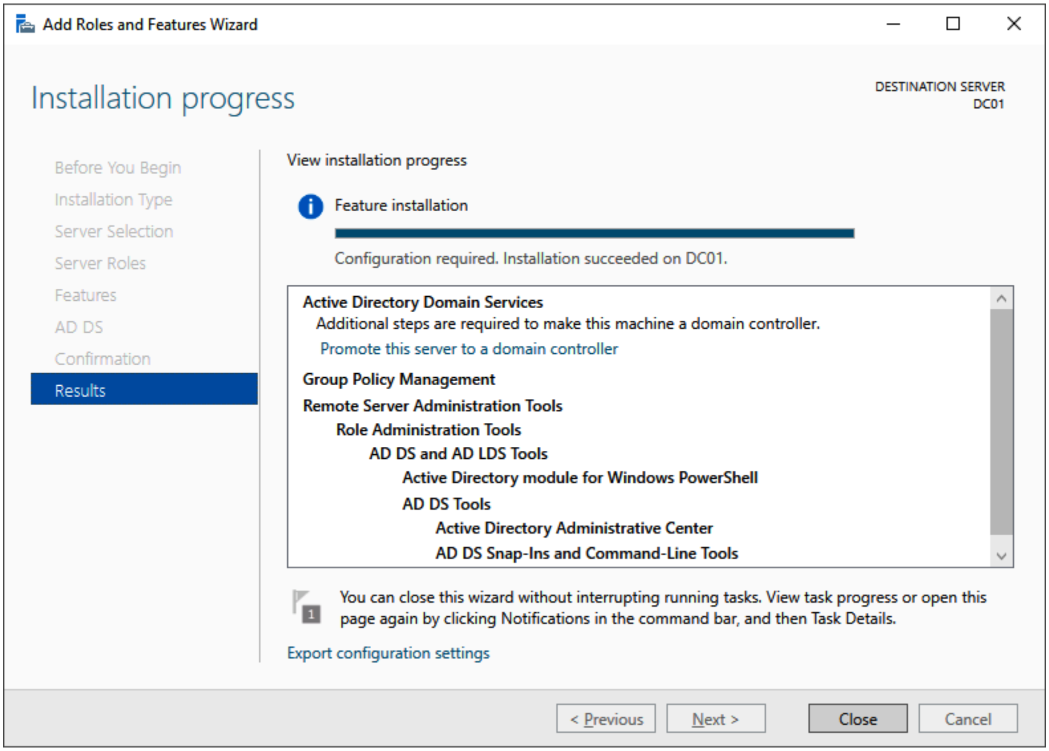
10. On the pop-up screen, click the **Add Features** button to add the features that are required for Active Directory Domain Services. These features include the **Group Policy Management** tool, **Active Directory module for Windows PowerShell**, **Active Directory Administrative Center**, and **AD DS Snap-Ins and Command-Line Tools**.

11. Back on the **Select server roles** screen, click **Next** >.

12. On the **Select server features** screen, click **Next** >.

13. On the **Active Directory Domain Services** screen, providing an overview of Active Directory and Azure AD, click **Next** >.

14. On the **Confirm installation selections** screen, click **Install**. The role and features will now be installed:



View installation progress

Feature installation

Configuration required. Installation succeeded on DC01.

Active Directory Domain Services
Additional steps are required to make this machine a domain controller.
Promote this server to a domain controller
Group Policy Management
Remote Server Administration Tools
Role Administration Tools
AD DS and AD LDS Tools
Active Directory module for Windows PowerShell
AD DS Tools
Active Directory Administrative Center
AD DS Snap-Ins and Command-Line Tools

You can close this wizard without interrupting running tasks. View task progress or open this page again by clicking Notifications in the command bar, and then Task Details.

Export configuration settings

Figure 2.3 – The Installation progress page of Add Roles and Features Wizard

15. When configuration of the **Active Directory Domain Services** server role is done, click **Close** to close the Add Roles and Features Wizard:

Using the `Install-WindowsFeature` cmdlet

As an alternative to using Server Manager, the `Install-WindowsFeature` cmdlet can be used. Perform the following line of Windows PowerShell in an elevated window to install the **Active Directory Domain Services** role:

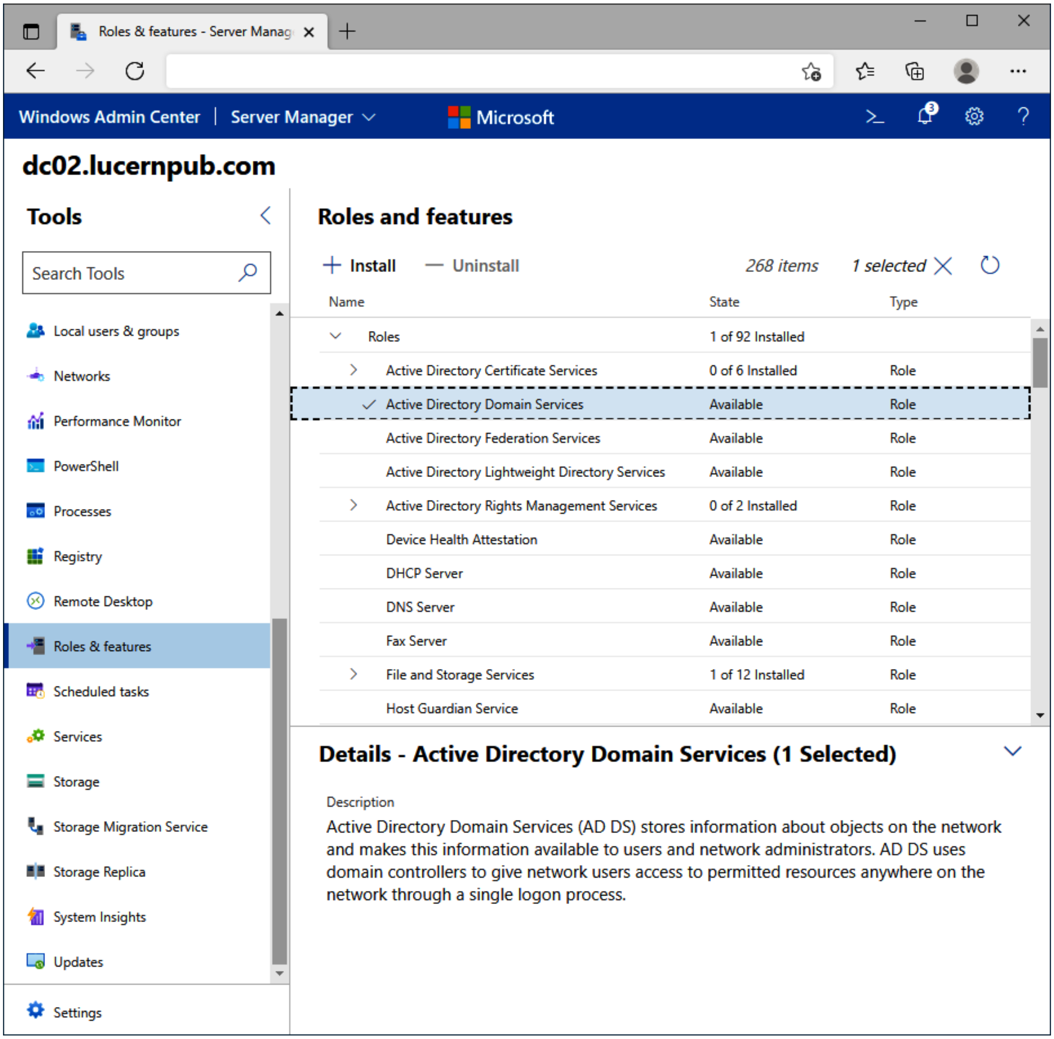
```
Install-WindowsFeature AD-Domain-Services
-IncludeManagementTools
```

The preceding line of Windows PowerShell offers the only way to install the Active Directory Domain Services role on a Server Core installation of Windows Server locally.

Using Windows Admin Center

Although a PowerShell script can be run from Windows Admin Center, it also offers a native way to install roles and features. Perform these steps:

1. In Windows Admin Center, click the Windows Server installation you want to install the **Active Directory Domain Services** role onto from the **All Connections** list.
2. In the left navigation menu, click **Roles & features**.
3. In the main pane, select the **Active Directory Domain Services** role in the **Roles and features** list by clicking the selection box to the left of it:



Name	State	Type
Roles	1 of 92 Installed	
Active Directory Certificate Services	0 of 6 Installed	Role
✓ Active Directory Domain Services	Available	Role
Active Directory Federation Services	Available	Role
Active Directory Lightweight Directory Services	Available	Role
Active Directory Rights Management Services	0 of 2 Installed	Role
Device Health Attestation	Available	Role
DHCP Server	Available	Role
DNS Server	Available	Role
Fax Server	Available	Role
File and Storage Services	1 of 12 Installed	Role
Host Guardian Service	Available	Role

Details - Active Directory Domain Services (1 Selected)

Description

Active Directory Domain Services (AD DS) stores information about objects on the network and makes this information available to users and network administrators. AD DS uses domain controllers to give network users access to permitted resources anywhere on the network through a single logon process.

Figure 2.4 – The Active Directory Domain Services role selected in Windows Admin Center

4. Click **+ Install**. The **Install Roles and Features** blade appears.
5. Click **Yes** to continue the installation with the additional **Active Directory module for Windows PowerShell**, **Group Policy Management**, **AD DS Snap-Ins and Command-Line Tools**, and **Active Directory Administrative Center** features installed.
6. A notification pops up, informing you that Windows Admin Center has **Successfully completed installation of Active Directory Domain Services**, which appears in the notification area when the roles and features have been successfully installed.

Promoting the server to a domain controller

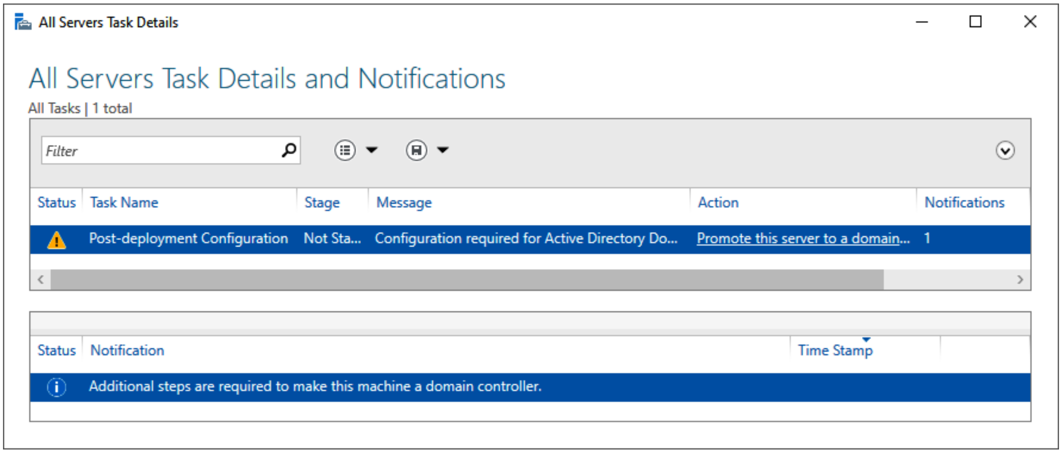
There are three ways to promote a Windows Server installation to a domain controller:

- Using the Active Directory Domain Services Configuration Wizard
- Using the `Install-DDSDomainController`, `Install-ADDSDomain`, or `Install-ADDSForest` cmdlets from the Active Directory module for Windows PowerShell
- Using `dcpromo.exe` with an answer file

Using the Active Directory Domain Services Configuration Wizard

Perform these steps to promote the server to a domain controller:

1. Press Start
2. Search for **Server Manager**, click its search result, or run `servermanager.exe` or return to **Server Manager** when you've accomplished installing the **Active Directory Domain Services** role using **Server Manager**.
3. In the left navigation pane of **Server Manager**, click **AD DS**.
4. Click the **More...** link in the blue ribbon (as shown in the following screenshot) titled **Configuration required for Active Directory Domain Services at server**:



All Servers Task Details and Notifications

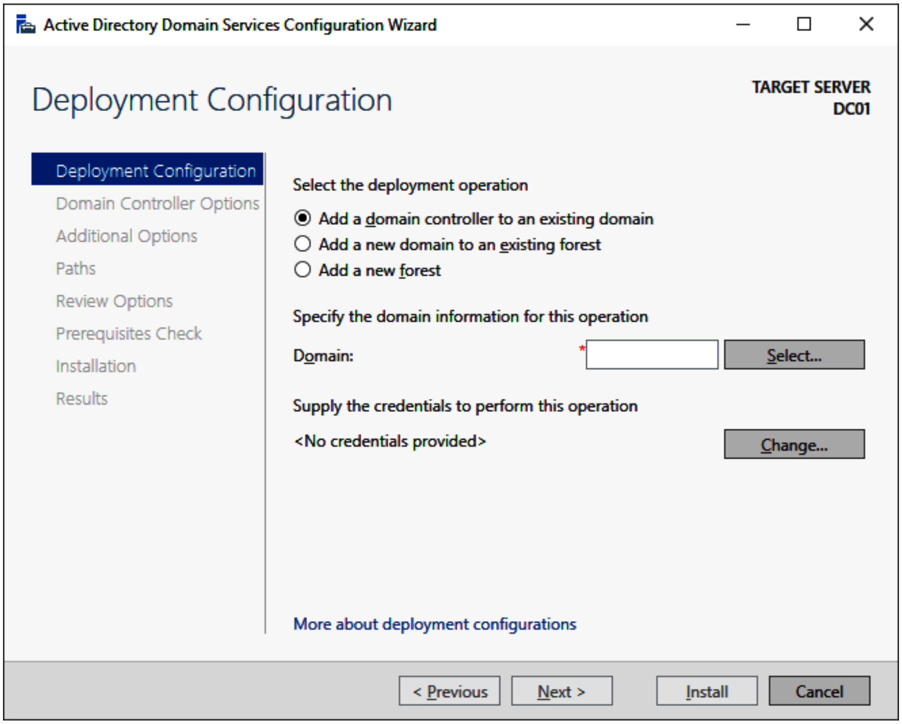
All Tasks | 1 total

Status	Task Name	Stage	Message	Action	Notifications
⚠	Post-deployment Configuration	Not Sta...	Configuration required for Active Directory Do...	Promote this server to a domain...	1

Status	Notification	Time Stamp
ⓘ	Additional steps are required to make this machine a domain controller.	

Figure 2.5 – Promote this server to a domain controller link in All Servers Task Details and Notifications

5. On the **All Servers Task Details and Notifications** screen, follow the **Promote this server to a domain controller** link. The **Active Directory Domain Services Configuration Wizard** window appears:



Deployment Configuration

TARGET SERVER
DC01

Select the deployment operation

● Add a domain controller to an existing domain
○ Add a new domain to an existing forest
○ Add a new forest

Specify the domain information for this operation

Domain: *

Supply the credentials to perform this operation

<No credentials provided>

Figure 2.6 – The Deployment Configuration screen of the Active Directory Domain Services Configuration Wizard

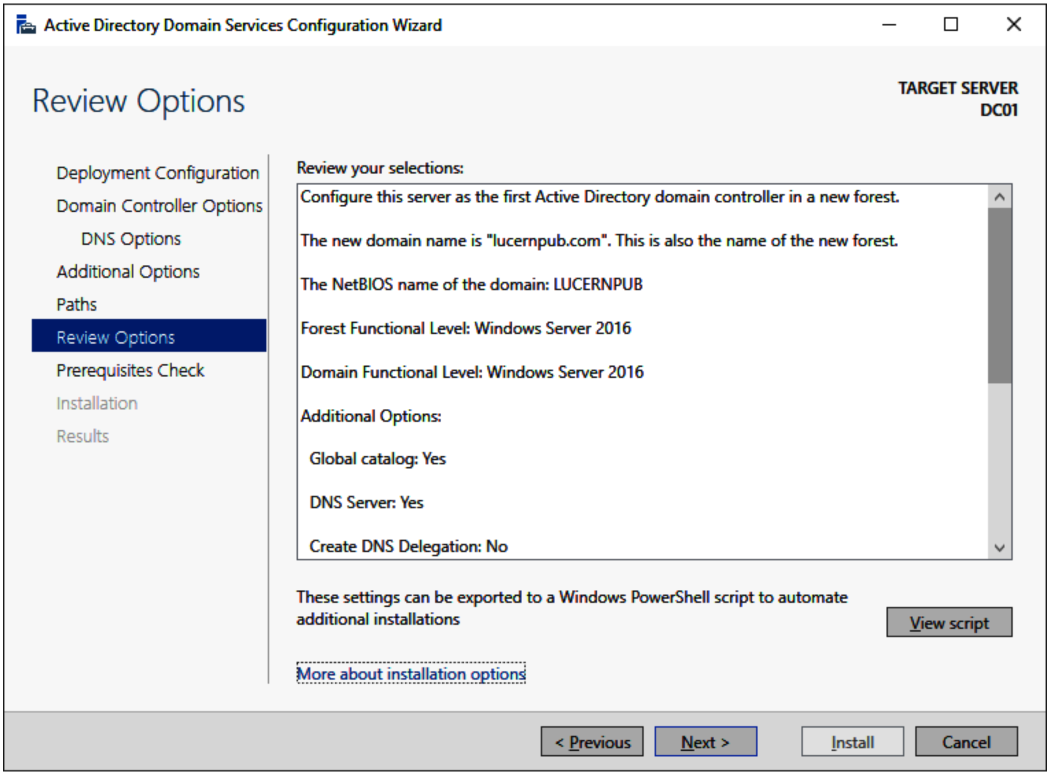
> **Tip**
>
> In the top-right corner of every **Active Directory Domain Services Configuration Wizard** screen, it shows the hostname of the Windows Server installation that you're promoting to a domain controller.

6. On the **Deployment Configuration** screen (as shown in the preceding screenshot), select the type of deployment you intend:

    - **Add a domain controller to an existing domain**
    - **Add a new domain to an existing forest**
    - **Add a new forest**

> **Important Note**
>
> By default, the **Add a domain controller to an existing domain** option is selected. This option will create a replica domain controller in the domain. If you're not sure which selection to make, please refer to the *Choosing between a new domain or forest* recipe in *Chapter 1*, *Optimizing Forests, Domains, and Trusts*. The **More about deployment configurations** link at the bottom of the **Deployment Configuration** screen provides a Microsoft link with more information.

7. Depending on your choices on the **Deployment Configuration** screen, supply information for the **Domain** or **Credentials** fields. Click **Next** to proceed to the next screen.
8. In all the other **Active Directory Domain Services Configuration Wizard** screens, make the appropriate choices for the deployment scenario. Click **Next** > every time to proceed to the next screen, until you reach the **Review Options** screen:



Review your selections:

Configure this server as the first Active Directory domain controller in a new forest.

The new domain name is "lucernpub.com". This is also the name of the new forest.

The NetBIOS name of the domain: LUCERNPUB

Forest Functional Level: Windows Server 2016

Domain Functional Level: Windows Server 2016

Additional Options:

Global catalog: Yes

DNS Server: Yes

Create DNS Delegation: No

These settings can be exported to a Windows PowerShell script to automate additional installations

Figure 2.7 – The Review Options screen of the Active Directory Domain Services Configuration Wizard

9. On the **Review Options** screen, review the choices made. Click **Next>** to proceed to the **Prerequisites Check** screen.

> **Tip**
>
> The **Review Options** screen features a button labeled **View script**. This button displays the Windows PowerShell script used to execute the domain controller promotion. This reusable script may be a real timesaver, especially when adding several domain controllers to an existing domain.

10. After the prerequisites checks have been performed, click **Install** on the **Prerequisites Check** screen to start promotion.

After successful promotion, the Windows Server installation reboots as a domain controller.

Promoting a domain controller using Windows PowerShell

For the Active Directory module for Windows PowerShell, Microsoft has decided to take a slightly different route. Instead of using a single PowerShell cmdlet to promote a domain controller, there are three separate PowerShell cmdlets for each of the three scenarios, as presented on the **Deployment Configuration** screen of the Active Directory Domain Services Configuration Wizard:

Scenario	PowerShell cmdlet
Add a domain controller to an existing domain	`Install-ADDSDomainController`
Add a new domain to an existing forest	`Install-ADDSDomain`
Add a new forest	`Install-ADDSForest`

Table 2.2 – Windows PowerShell cmdlets per domain controller promotion scenario

To add a domain controller to an existing domain, the simplest script would look like this:

```
Install-ADDSDomainController -DomainName lucernpub.com
```

However, to add a domain controller to an existing domain, as you would in the previous example, the following script would suffice:

```
Install-ADDSDomainController -DomainName lucernpub.
com -Credential (Get-Credential) -installDNS:$true
-NoGlobalCatalog:$false -DatabasePath "E:\NTDS" -Logpath "E:\
Logs" -SysvolPath "E:\SYSVOL" -Sitename RemoteLocation
```

This adds a domain controller to the `lucernpub.com` Active Directory domain, using credentials you will be prompted for securely. The domain controller is installed with a DNS server and configured as a global catalog server. All the Active Directory-related files are stored in corresponding folders on the `E:\` drive, and when successful, the Windows Server installation you intend as the domain controller reboots automatically.

Replace the values in the preceding sample script with the values of your choice.

Promoting a domain controller using dcpromo.exe

Despite many news outlets reporting that `dcpromo` is dead, the popular option to promote a Windows Server installation to a domain controller is alive and well, even in the latest Windows Server versions. One change to the functionality of `dcpromo.exe`, when compared to previous versions of Windows Server, is that you can no longer use `dcpromo.exe` to launch the Active Directory Domain Services Configuration Wizard. You'll need to use `dcpromo.exe` with an answer file or with all the installation arguments specified.

The benefits of using dcpromo.exe include the use of many options that are not available when using the Active Directory Domain Services Configuration Wizard and also a wide array of sample answer files and scripts. As the type of answer files used when using dcpromo.exe, and the arguments for use on the command line, have been available since the early days of Windows Server, many people have used them, and many people have written them.

Using dcpromo.exe with an answer file consists of running the following command prompt line:

```
dcpromo.exe /unattend: C:\install\dcpromo.txt
```

Simply replace the text file location with the file of your choice.

You can also use network paths such as \\server\promotiontext$\dcpromo.txt to supply an answer file to dcpromo.txt. This makes for an ideal scenario where files don't remain lingering on domain controllers promoted this way.

The answer file consists of several arguments. Typical arguments found in the answer file include the ReplicaOrNewDomain, InstallDNS, and ConfirmGC arguments. A prime example of an answer file to add an additional domain controller to an existing domain would look like the following:

```
[DCINSTALL]
ReplicaorNewDomain= replica
ReplicaDomainDNSName= lucernpub.com
UserDomain= LUCERNPUB
UserName= Administrator
SiteName= RemoteLocation
Password= "P@$$w0rd"
InstallDNS= Yes
ConfirmGC= Yes
CreateDNSDelegation= No
LogPath= E:\Logs
SYSVOLPath= E:\SYSVOL
SafeModeAdminPassword= "P@$$w0rd"
RebootOnSuccess= true
```

Using this answer file adds a domain controller to the lucernpub.com Active Directory domain, using the credentials for the administrator account with the P@$$w0rd password. The domain controller is installed with a DNS server and configured as a global catalog server. All the Active Directory-related files are stored in corresponding folders on the E:\ drive, and when successful, the Windows Server installation you intend as the domain controller will be rebooted automatically.

Replace the values in the preceding sample file with the values of your choice.

When promotion is successful, the passwords specified as the values for the Password and SafeModeAdminPassword arguments are cleared from the answer file. However, when promotion is unsuccessful, these values remain and may cause harm when falling into the wrong hands.

The arguments in the answer file can also be specified as command-line arguments. The arguments can be reused one on one, so the preceding sample answer file would correspond to the following command line:

```
dcpromo.exe /unattend /replicaornewdomain:Replica /
replicadomaindnsname:lucernpub.com /userdomain:LUCERNPUB
/username:administrator /password:"P@$$w0rd" /
sitename:RemoteLocation /installdns:yes /confirmgc:yes /
databasepath:"E:\NTDS" /logpath:"E:\logs" /sysvolpath:"E:\
sysvol" /safemodeadminpassword:"P@$$w0rd"
```

Replace the values in the preceding sample file with the values of your choice.

Checking proper promotion

After promoting a Windows Server installation to the domain controller, it's recommended to check for proper promotion. Perform these steps to check the promotion:

1. **Check the logs**: The following two files contain all the actions performed when promoting the Windows Server installation to the domain controller. A good way to check for improper promotion is to search for lines containing errors and warnings:

   - C:\Windows\Debug\dcpromo.log
   - C:\Windows\Debug\dcpromoui.log

2. **Check the event viewer**: In the event viewer (eventvwr.exe), new dedicated logs are created for Active Directory. Search these logs for any Active Directory-related errors.

3. **Run Windows Update**: Even though one of the recommended steps is to update the Windows Server installation you intend to promote to the domain controller, it's also a recommended step to install Windows Updates after the Windows Server installation has been promoted, as updates apply to newly installed server roles and features too. These role-specific updates are only applied after the role is installed.

See also

For more information, refer to the following recipes:

- See the *Preparing a Windows server to become a domain controller* recipe.
- See the *Promoting a server to a read-only domain controller* recipe.
- See the *Checking replication* recipe.

Promoting a server to a read-only domain controller

Read-only domain controllers were introduced with Windows Server 2008. They have been hugely popular for providing Active Directory Domain Services to branch offices and small perimeter networks.

Read-only domain controllers are the ideal type of domain controllers for environments with the following:

- Poor physical security
- Relatively few user accounts and/or devices
- Relatively poor bandwidth to central data centers with domain controllers
- Little local IT knowledge and/or experience

These characteristics are typically true for branch offices. Before read-only domain controllers, administrators had to make the hard choice between doing nothing, placing fully (read-write) domain controllers in these locations, or upgrading the available bandwidth and/or resiliency of the networking connections between the branch offices and the head office or central data center(s).

Some organizations have opted to deploy read-only domain controllers in perimeter networks. Microsoft supports only one read-only domain controller per Active Directory site. This way, any perimeter network deployment would not have much Active Directory resiliency. Many organizations have, therefore, opted for a separate Active Directory forest for these implementation scenarios.

Getting ready

Read-only domain controllers have requirements that we need to adhere to before we can deploy and use them:

- At least one domain controller running Windows Server 2008 (or a newer version of Windows Server).
- The Windows Server 2003 **Forest Functional Level** (**FFL**), or a higher FFL.
- The Windows Server 2008 **Domain Functional Level** (**DFL**), or a higher DFL, for the Active Directory domain(s) in which you intend to implement read-only domain controllers.
- `ADPrep /rodcprep` needs to have run at least once on the domain controller holding the Domain Naming Master FSMO role, but this step may be skipped when the Active Directory environment was never set up or has never run with pre-Windows Server 2008-based domain controllers.
- When implementing read-only domain controllers for branch offices, create the corresponding Active Directory sites and site connections first.
- Check for proper Active Directory replication before implementing a read-only domain controller.

Read-only domain controllers allow for scoped replication. It's a recommended practice to determine the user accounts and computer accounts that are strictly needed in the branch office location. The read-only domain controller will be able to cache the passwords for these accounts to speed up authentication for these accounts in the branch office. The **Allowed RODC Password Replication Group** is the default group in which to add (groups of) user accounts and computer accounts for this functionality.

If you desire strict group memberships for this functionality per read-only domain controller, create the groups you need before you promote the Windows Server installation to a read-only domain controller for which you need the group scope.

Another way to think about security before promoting the first read-only domain controller is to determine the privileged accounts and otherwise sensitive accounts for which you do not want passwords replicated to the read-only domain controller you intend to create. These (groups of) accounts can be specified as the accounts that are denied from replicating passwords to the RODC.

How to do it...

Just like read/write domain controllers, promoting a Windows Server installation to a read-only domain controller consists of three steps:

1. Installing the Active Directory Domain Services role
2. Promoting a server to a domain controller
3. Checking proper promotion

When using `dcpromo.exe`, you do not have to install the role beforehand.

There are several ways to promote the server. The following table displays the possibilities:

Method	**Install the role**	**Promote the server**
Using Server Manager	Yes	No
Using the `Install-WindowsFeature` cmdlet	Yes	No
Using Windows Admin Center	Yes	No
Using the Active Directory Domain Services Configuration Wizard	Required	Yes
Using the `Install-DDSDomainController` cmdlet from the Active Directory module for Windows PowerShell	Required	Yes
Using `dcpromo.exe` with an answer file	Not necessary	Yes

Table 2.3 – Methods for installing the Active Directory Domain Services role and promoting a server to a domain controller

The methods in the table are all explained in more detail in this recipe.

Installing the Active Directory Domain Services role

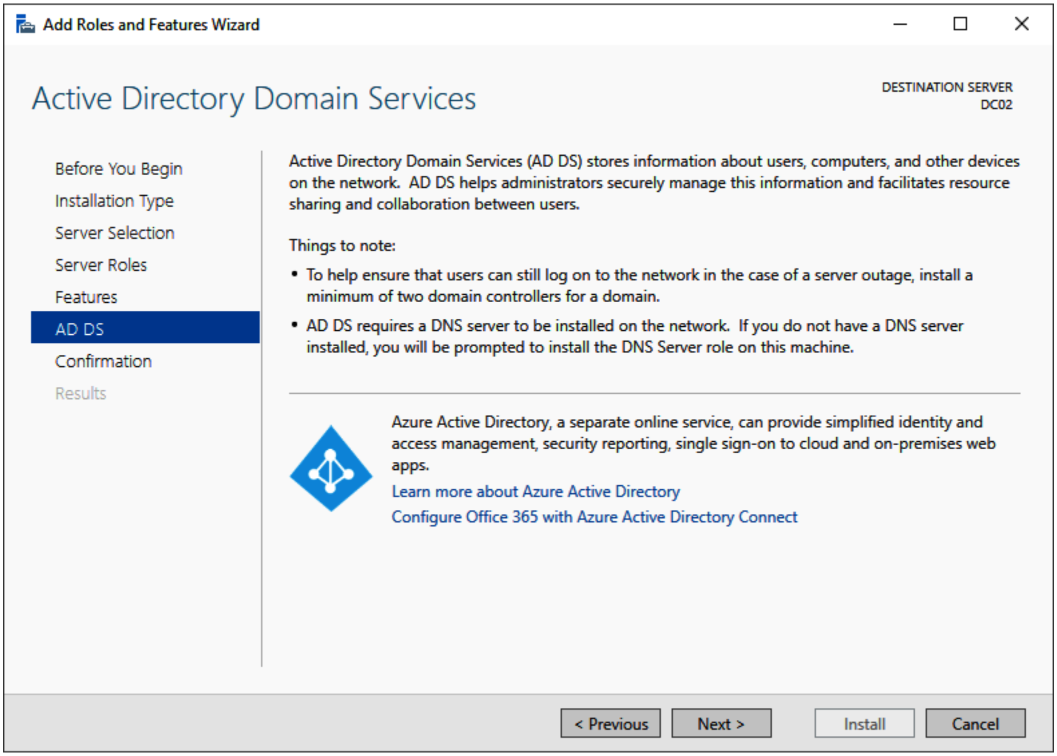
There are three ways to install the **Active Directory Domain Services** role:

- Using Server Manager
- Using the `Install-WindowsFeature` cmdlet
- Using Windows Admin Center

Using Server Manager

To install the **Active Directory Domain Services** role using **Server Manager**, perform these steps:

1. Press Start.
2. Search for **Server Manager** and click its corresponding search result or run `servermanager.exe`. The **Server Manager** window appears.
3. In the gray top bar of **Server Manager**, click **Manage**.
4. Select **Add Roles and Features** from the menu. The **Add Roles and Features Wizard** window appears, with the **Before you begin** screen.
5. On the **Before you begin** screen, click **Next** >.
6. On the **Select installation type** screen, select **Role-based or feature-based installation** and click **Next** >.
7. On the **Select destination server** screen, select either the local Windows Server installation from the server pool list, the remote Windows Server installation you intend to promote to the domain controller from the server pool list, or both types of resources.
8. Click **Next** >.
9. On the **Select server roles** screen, select the **Active Directory Domain Services** role from the list of available roles. The **Add Roles and Features Wizard** pop-up window appears.
10. On the pop-up screen, click the **Add features** button to add the features that are required for Active Directory Domain Services. These features include the **Group Policy Management** tool, **Active Directory module for Windows PowerShell**, **Active Directory Administrative Center**, and **AD DS Snap-Ins and Command-Line Tools**.
11. Back on the **Select server roles** screen, click **Next** >.
12. On the **Select server features** screen, click **Next** >.
13. On the **Active Directory Domain Services** screen, providing an overview of Active Directory and Azure AD, click **Next** >:



Active Directory Domain Services (AD DS) stores information about users, computers, and other devices on the network. AD DS helps administrators securely manage this information and facilitates resource sharing and collaboration between users.

Things to note:

- To help ensure that users can still log on to the network in the case of a server outage, install a minimum of two domain controllers for a domain.
- AD DS requires a DNS server to be installed on the network. If you do not have a DNS server installed, you will be prompted to install the DNS Server role on this machine.

Azure Active Directory, a separate online service, can provide simplified identity and access management, security reporting, single sign-on to cloud and on-premises web apps.

Learn more about Azure Active Directory

Configure Office 365 with Azure Active Directory Connect

Figure 2.8 – Methods for installing the Active Directory Domain Services role and promoting a server to a domain controller

14. On the **Confirm installation selections** screen, click **Install**.
15. When configuration of the **Active Directory Domain Services** server role is done, click **Close** to close the Add Roles and Features Wizard.

Using the Install-WindowsFeature cmdlet

As an alternative to using Server Manager, the `Install-WindowsFeature` cmdlet can be used. Perform the following line of Windows PowerShell in an elevated window to install the **Active Directory Domain Services** role:

```
Install-WindowsFeature AD-Domain-Services
-IncludeManagementTools
```

The preceding line of Windows PowerShell offers the only way to install the Active Directory Domain Services role on a Server Core installation of Windows Server locally.

Using Windows Admin Center

Although a PowerShell script can be run from the Windows Admin Center, it also offers a native way to install roles and features. Perform these steps:

1. In the Windows Admin Center, click the Windows Server installation you want to install the **Active Directory Domain Services** role onto from the list of **All Connections**.
2. In the left navigation menu, click **Roles & features**.
3. In the main pane, select the **Active Directory Domain Services** role from the **Roles and features** list by clicking the selection box to the left of it.
4. Click **+ Install**. The **Install Roles and Features** blade appears.
5. Click **Yes** to continue the installation with the additional **Active Directory module for Windows PowerShell**, **Group Policy Management**, **AD DS Snap-Ins and Command-Line Tools**, and **Active Directory Administrative Center** features installed.
6. A notification pops up informing you that Windows Admin Center has **Successfully completed installation of Active Directory Domain Services**, which appears in the notification area when the roles and features have been successfully installed.

Promoting the server to a domain controller

There are three ways to promote a Windows Server installation to a read-only domain controller:

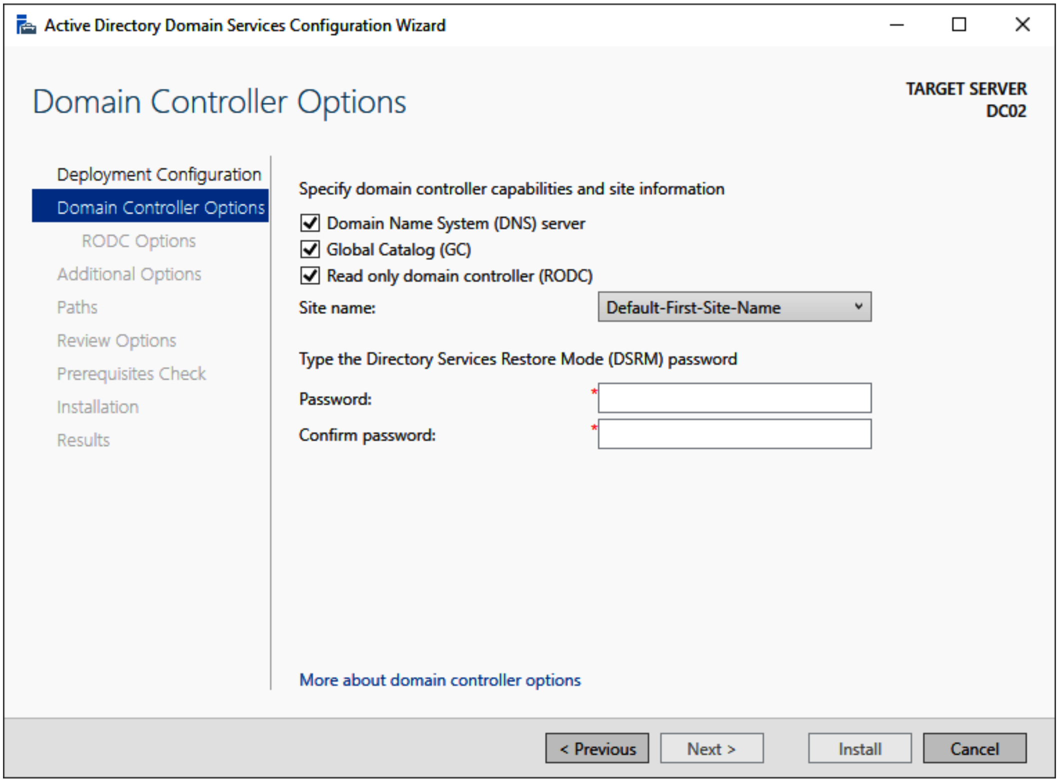
- Using the Active Directory Domain Services Configuration Wizard
- Using the `Install-ADDSDomainController` cmdlet from the Active Directory module for Windows PowerShell with the dedicated `-ReadOnlyReplica` parameter
- Using `dcpromo.exe` with an answer file

Using the Active Directory Domain Services Configuration Wizard

Perform these steps to promote the server to a read-only domain controller:

1. Press Start.
2. Search for **Server Manager**, click its search result, or run `servermanager.exe` or return to **Server Manager** when you've accomplished installing the **Active Directory Domain Services** role using **Server Manager**.

3. In the left navigation pane, click **AD DS**.
4. Click the **More...** link in the yellow ribbon titled **Configuration required for Active Directory Domain Services at server**.
5. In **All Servers Task Details and Notifications**, follow the **Promote this server to a domain controller** link. The **Active Directory Domain Services Configuration Wizard** window appears.
6. On the **Deployment Configuration** screen, select **Add a domain controller to an existing domain**. Then, input the DNS domain name and administrator credentials for the Active Directory domain for which you intend to add a read-only domain controller. Click **Next >** to proceed to the **Domain Controller Options** screen:


Active Directory Domain Services Configuration Wizard

Domain Controller Options

TARGET SERVER
DC02

Specify domain controller capabilities and site information

☑ Domain Name System (DNS) server
☑ Global Catalog (GC)
☑ Read only domain controller (RODC)

Site name: Default-First-Site-Name

Type the Directory Services Restore Mode (DSRM) password

Password: *
Confirm password: *

More about domain controller options

Figure 2.9 – The Domain Controller Options screen of the Active Directory Domain Services Configuration Wizard

7. On the **Domain Controller Options** screen, we're presented with a few options:

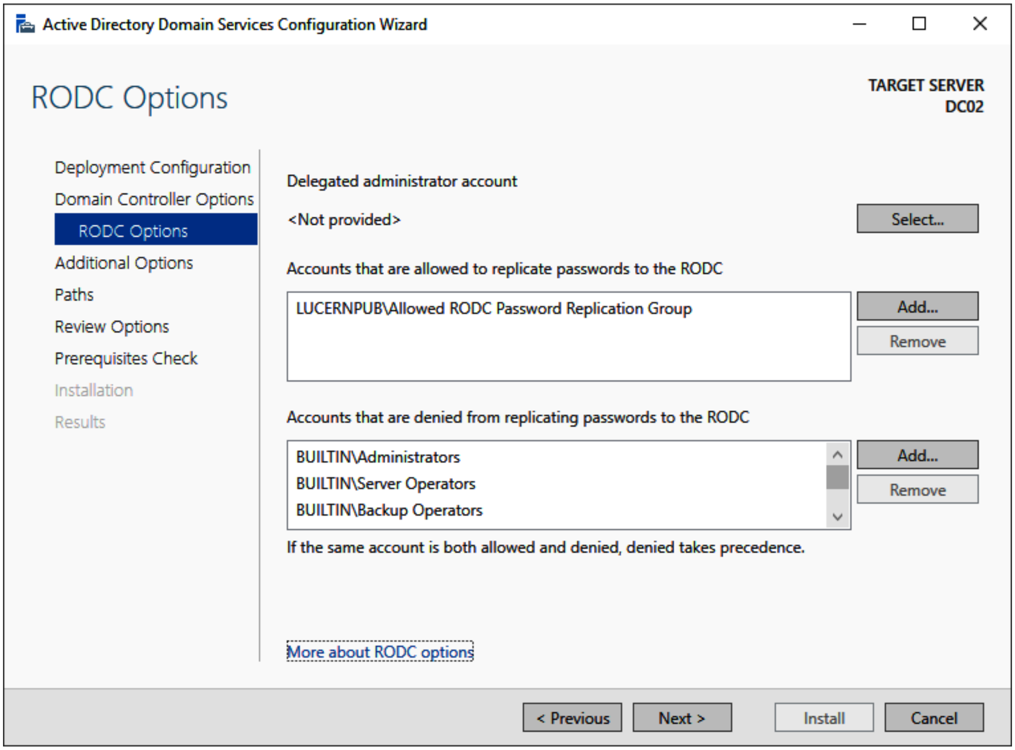
   I. Select the **Read only domain controller (RODC)** option.

   II. When preferred, select the **Domain Name System (DNS) server** and **Global Catalog (GC)** options.

   III. Select a site name from the drop-down list of available Active Directory sites.

   IV. Enter the Directory Services Restore Mode password for the intended read-only domain controller.

8. Click **Next >** to proceed to the **RODC Options** screen:



Active Directory Domain Services Configuration Wizard

RODC Options

TARGET SERVER
DC02

Delegated administrator account

<Not provided>

Accounts that are allowed to replicate passwords to the RODC

LUCERNPUB\Allowed RODC Password Replication Group

Accounts that are denied from replicating passwords to the RODC

BUILTIN\Administrators
BUILTIN\Server Operators
BUILTIN\Backup Operators

If the same account is both allowed and denied, denied takes precedence.

Figure 2.10 – The RODC Options screen of the Active Directory Domain Services Configuration Wizard

9. On the **RODC Options** page, perform the following optional actions:

   I. Select a user account for delegation.

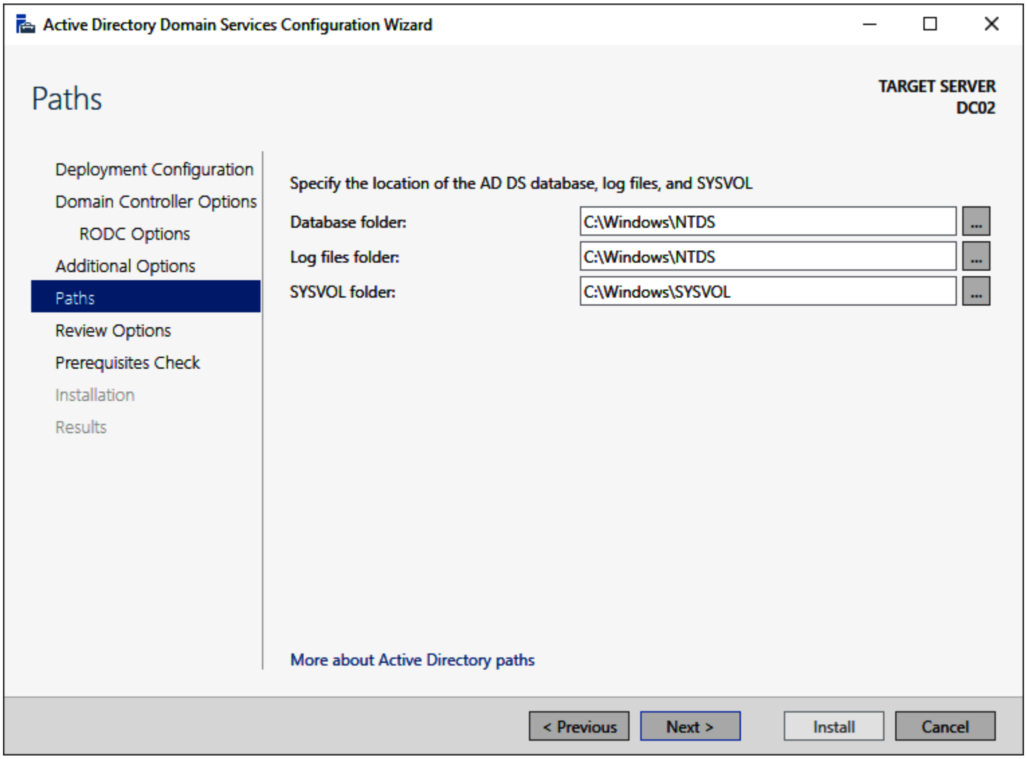
   II. Select **Accounts that are allowed to replicate passwords to the RODC**.

   III. Select **Accounts that are denied from replicating passwords to the RODC**.

Tip

If a group or an account features in both the accounts that are allowed to replicate passwords to the RODC and accounts that are denied from replicating passwords to the RODC, then the group or account is denied from replicating passwords to the RODC.

10. Click **Next >** to proceed to the next screen.
11. On the **Additional Options** screen, optionally select a fully writable domain controller from which to replicate the Active Directory database and the Active Directory `SYSVOL`. Click **Next >** to continue to the **Paths** screen:



Specify the location of the AD DS database, log files, and SYSVOL

Database folder: C:\Windows\NTDS

Log files folder: C:\Windows\NTDS

SYSVOL folder: C:\Windows\SYSVOL

Figure 2.11 – The Paths screen of the Active Directory Domain Services Configuration Wizard

12. On the **Paths** screen, verify the default locations underneath `C:\Windows` or change the values to store Active Directory-related files somewhere else.
13. Click **Next >** to proceed.
14. On the **Review Options** screen, review the choices made. Click **Next >** to proceed to the **Prerequisites Checks** screen.

> **Tip**
>
> The **Review Options** screen features a button labeled **View script**. This button displays the Windows PowerShell script used to execute the read-only domain controller promotion. This reusable script may be a real timesaver, especially when adding several read-only domain controllers to an existing domain.

15. After the prerequisites checks have been performed, click **Install** on the **Prerequisites checks** screen to start promotion.

After successful promotion, the Windows Server installation will reboot as a read-only domain controller.

Promoting a read-only domain controller using Windows PowerShell

For the Active Directory module for Windows PowerShell, Microsoft does not offer a dedicated PowerShell cmdlet to add a read-only domain controller. Instead, `Install-ADDSDomainController` is used with the dedicated `-ReadOnlyReplica` parameter. The simplest script would look like the following code:

```
Install-ADDSDomainController -DomainName lucernpub.com
-Sitename RemoteLocation -ReadOnlyReplica
```

However, to add a read-only domain controller to an existing domain as you would with the previous example, the following script would be needed:

```
Install-ADDSDomainController -DomainName lucernpub.com
-Credential (Get-Credential) -ReadOnlyReplica -installDNS:$true
-NoGlobalCatalog:$false -DatabasePath "E:\NTDS" -Logpath "E:\
Logs" -SysvolPath "E:\SYSVOL" -Sitename RemoteLocation
```

This will add a read-only domain controller to the `lucernpub.com` Active Directory domain using credentials you will be prompted for securely. The domain controller will be installed with a DNS server and configured as a global catalog server. All the Active Directory-related files are stored in corresponding folders on the `E:\` drive, and, when successful, the Windows Server installation you intend as the domain controller will be rebooted automatically.

Replace the values in the preceding sample script with the values of your choice.

Promoting a read-only domain controller using dcpromo.exe

Read-only domain controllers can be promoted using `dcpromo.exe` with an answer file or with all the installation arguments specified, just like fully writable domain controllers. An added benefit is that `dcpromo.exe` will install the **Active Directory Domain Services** server role automatically when it's not yet present.

Using `dcpromo.exe` with an answer file consists of running the following command line:

```
dcpromo.exe /unattend: C:\install\dcpromo.txt
```

A prime example of an answer file to add a read-only domain controller would look like this:

```
[DCINSTALL]
ReplicaorNewDomain= readonlyreplica
ReplicaDomainDNSName= lucernpub.com
UserDomain= LUCERNPUB
UserName= Administrator
SiteName= RemoteLocation
Password= "P@$$w0rd"
InstallDNS= Yes
ConfirmGC= Yes
CreateDNSDelegation= No
DatabasePath= E:\NTDS
LogPath= E:\Logs
SYSVOLPath= E:\SYSVOL
SafeModeAdminPassword= "P@$$w0rd"
RebootOnSuccess= true
```

The preceding answer file adds a read-only domain controller to the `lucernpub.com` Active Directory domain, using the credentials for the administrator account with the `P@$$w0rd` password. The read-only domain controller is installed with a DNS server and configured as a global catalog server. All the Active Directory-related files are stored in corresponding folders on the `E:\` drive, and when successful, the Windows Server installation you intend as the read-only domain controller is rebooted automatically.

Replace the values in the preceding sample file with the values of your choice.

The arguments in the answer file can also be specified as command-line arguments. The arguments can be reused one on one, so the preceding sample answer file would correspond to the following command line:

```
dcpromo.exe /unattend /replicaornewdomain:ReadOnlyReplica /
replicadomaindnsname:lucernpub.com /userdomain:LUCERNPUB
/username:administrator /password:"P@$$w0rd" /
sitename:RemoteLocation /installdns:yes /confirmgc:yes /
databasepath:"E:\NTDS" /logpath:"E:\logs" /sysvolpath:"E:\
sysvol" /safemodeadminpassword:"P@$$w0rd"
```

Replace the values in the preceding command line with the values corresponding to your environment.

Checking proper promotion

After promoting a Windows Server installation to a read-only domain controller, it's recommended practice to check for proper promotion. Perform these steps to check:

1. **Check the logs**: The following two files contain all the actions performed when promoting the Windows Server installation to a domain controller. A good way to check for improper promotion is to search for lines containing errors and warnings, logged in the following files:

    - `C:\Windows\Debug\dcpromo.log`
    - `C:\Windows\Debug\dcpromoui.log`

2. **Check the event viewer**: In the event viewer, `eventvwr.exe`, new dedicated logs are created for Active Directory Domain Services. Search these logs for any Active Directory-related errors.
3. **Run Windows updates**: Even though one of the recommended steps is to update the Windows Server installation you intend to promote to a read-only domain controller, it's also a recommended step to install Windows updates after the Windows Server installation has been promoted, as updates apply to the newly installed server roles and features too. These role-specific updates are only applied after the role is installed.

How it works...

Read-only domain controllers are different from normal domain controllers in the following ways:

- They allow read-only access to the Active Directory database and SYSVOL. Read-only domain controllers refer to other domain controllers for write operations such as SYSVOL.
- They allow read-only access to the DNS records. Read-only domain controllers refer to other domain controllers for DNS registration requests.
- They allow for scoped replication, so only the accounts that are needed in the Active Directory site where the read-only domain controller is placed are synchronized. This way, privileged accounts and other sensitive accounts remain in the central data center.
- They allow for a quick change of passwords for synchronized users when the read-only domain controller is stolen or otherwise compromised.
- They use their own dedicated account to encrypt their Kerberos tickets. This prevents attackers from decrypting a Kerberos **Ticket Granting Ticket** (**TGT**), issued by a read-only domain controller, to obtain the secret of the Kerberos account (`krbtgt`).

Additionally, because no Active Directory writes are expected from read-only domain controllers, normal domain controllers don't replicate from them.

See also

For more information, refer to the following recipes:

- See the *Preparing a Windows server to become a domain controller* recipe.
- See the *Promoting a server to a domain controller* recipe.
- See the *Creating an Active Directory site* recipe in *Chapter 16, Hardening Azure AD*.
- See the *Checking replication* recipe in *Chapter 16, Hardening Azure AD*.

Using Install From Media

For Active Directory environments with really low bandwidth or networking resiliency between locations with domain controllers, regardless of whether these are read-only domain controllers or fully writable domain controllers, promoting a Windows Server installation to a domain controller can take a long time or even fail.

In these types of scenarios, for adding an additional domain controller or read-only domain controller to an existing domain, Microsoft offers the **Install From Media** (**IFM**) option.

Getting ready

When creating IFM media, check for proper Active Directory replication before creating the IFM media on the domain controller. This ensures that the domain controller is up to date with all changes in Active Directory.

Create a folder on the source and destination domain controller to store the files needed for IFM.

How to do it...

IFM consists of two steps:

- Creating the `IFM` package
- Leveraging the `IFM` package

Creating the IFM package

To create the `IFM` package, perform the following actions on a domain controller in a well-connected networking location, running the same version of Windows Server on which you intend to use the `IFM` package to swiftly promote it to a domain controller in a low-bandwidth scenario:

> **Tip**
>
> `IFM` packages to create read-only domain controllers can be created on both read-only domain controllers and on fully writable domain controllers. `IFM` packages to create fully writable domain controllers can only be created on fully writable domain controllers.

1. Sign in interactively to the domain controller that you want to use as the source server for IFM.
2. Press Start.
3. Search for **Command Prompt**, right-click its search result, and choose **Run as administrator** from the context menu. Alternatively, run `cmd.exe`, but instead of running it by pressing *Enter*, press *Ctrl*, *Shift*, and *Enter*.

4. Run the following command to start the NTDS utility in interactive mode:

```
ntdsutil.exe
```

5. Type the following command in interactive mode to select the Active Directory database:

```
activate instance ntds
```

6. Type the following command in interactive mode to enter the `IFM` creation context:

```
IFM
```

7. Type the following command in interactive mode to create `IFM`, including the contents of the Active Directory `SYSVOL` for a read-only domain controller, and place it in the `C:\IFM` folder:

```
create RODC C:\IFM
```

8. Type the following command in interactive mode to exit the `IFM` context:

```
quit
```

9. Type the following command in interactive mode to exit the NTDS utility itself:

```
quit
```

10. Close the **Command Prompt** window.

Leveraging the IFM package

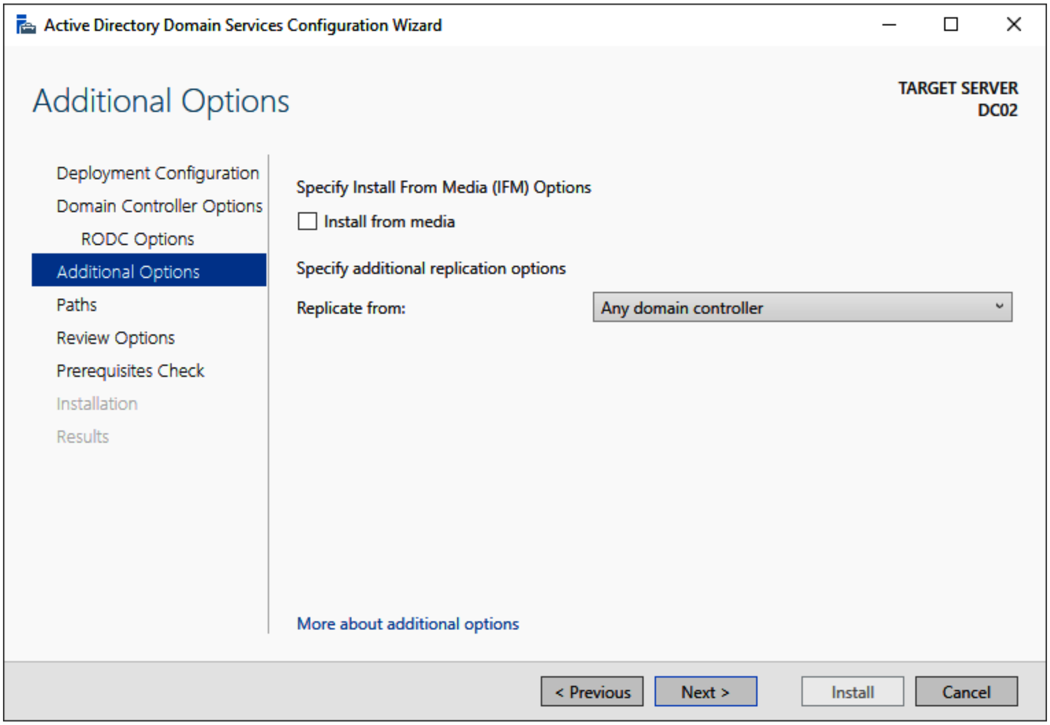
To leverage the `IFM` package on the destination domain controller in the remote location, choose one of the following methods:

- Using the Active Directory Domain Services Configuration Wizard after you've installed the **Active Directory Domain Services** role
- Using `dcpromo.exe`
- Using the `Install-ADDSDomainController` PowerShell cmdlet

Using the Active Directory Domain Services Configuration Wizard

Perform these steps to leverage the install using the Active Directory Domain Services Configuration Wizard:

1. Promote the Windows Server installation as you would normally.
2. On the **Additional Options** screen, click the **Install from media** option:



Active Directory Domain Services Configuration Wizard

Additional Options

TARGET SERVER
DC02

Specify Install From Media (IFM) Options

□ Install from media

Specify additional replication options

Replicate from: Any domain controller

Figure 2.12 – The Additional Options screen of the Active Directory Domain Services Configuration Wizard

3. On the **Install from Media** screen, specify the location on the drive of the Windows Server installation you intend to promote to a (read-only) domain controller using the **Install from Media** option.
4. Optionally, specify the fully writable domain controller you want to replicate from. Specify a domain controller that is best reachable from the intended domain controller.
5. Click **Next** > to proceed to the next screens as you normally would.

Using the Install-ADDSDomainController PowerShell cmdlet

The `Install-ADDSDomainController` PowerShell cmdlet only needs the `-InstallationMediaPath` additional parameter to leverage the `IFM` package when promoting a Windows Server installation to a domain controller.

When combining it with the sample PowerShell command for adding a domain controller to an existing domain, the following line of Windows PowerShell emerges:

```
Install-ADDSDomainController -DomainName lucernpub.com
-InstallationMediaPath "C:\IFM"
```

Replace `lucernpub.com` with the DNS domain name of your Active Directory domain.

Using dcpromo.exe

As with the `Install-ADDSDomainController` PowerShell cmdlet, `dcpromo.exe` requires an optional parameter to leverage the `IFM` package.

Perform the following steps:

1. Promote the Windows Server installation, as you would normally.
2. When using an answer file, add the following line:

   ```
   ReplicationSourcePath= "C:\IFM"
   ```

3. When using unattended mode, add the following argument:

   ```
   /ReplicationSourcePath:"C:\IFM"
   ```

How it works...

As a Windows Server installation becomes a domain controller, it replicates the contents of the Active Directory database and the Active Directory SYSVOL to its local hard drive(s). The entire package needed for this replication can also be assembled before promotion. Then, the `IFM` package can be delivered to the remote location, or even carried by the technician that will promote the (read-only) domain controller.

> **Important Note**
>
> The amount of network traffic needed when using the IFM option is heavily reduced but is certainly not zero. As the `IFM` package represents a point-in-time snapshot of the contents of the Active Directory database and the Active Directory SYSVOL, any changes between the time of the creation of the `IFM` package and using it will need to replicate before promotion of the domain controller is successfully completed.

Using domain controller cloning

The `IFM` feature for promoting domain controllers leverages the fact that the contents of the Active Directory database and the Active Directory SYSVOL are identical throughout all domain controllers within the domain. The domain controller cloning feature takes this one step further and leverages the fact that all domain controllers are largely identical – not just the Active Directory-related files but all operating system files, most agent installations, information security measures, and most configuration items.

When a domain controller is properly prepared and promoted, it can serve as a template.

Getting ready

The domain controller cloning feature requires the following:

- A hypervisor platform offering the VM-GenerationID functionality
- At least one domain controller running Windows Server 2012 or a newer version of Windows Server, promoted to a domain controller, holding the PDC Emulator FSMO role

The domain controller you intend to clone needs to adhere to the following requirements:

- Running Windows Server 2012 or a newer version of Windows Server
- Running on top of a VM-GenerationID-capable hypervisor platform
- Running the latest stable integration components or VMware tools
- Promoted as a domain controller
- Not holding the PDC Emulator FSMO role
- Not holding the RID Master FSMO role

When cloning domain controllers, check for proper Active Directory replication before cloning. This ensures that the domain controllers are up to date with all changes in Active Directory and can communicate the changes involved in adding a domain controller.

How to do it...

There are four steps to cloning a domain controller:

- Making sure all agents and software packages are cloneable
- Supplying the information for the new domain controller configuration

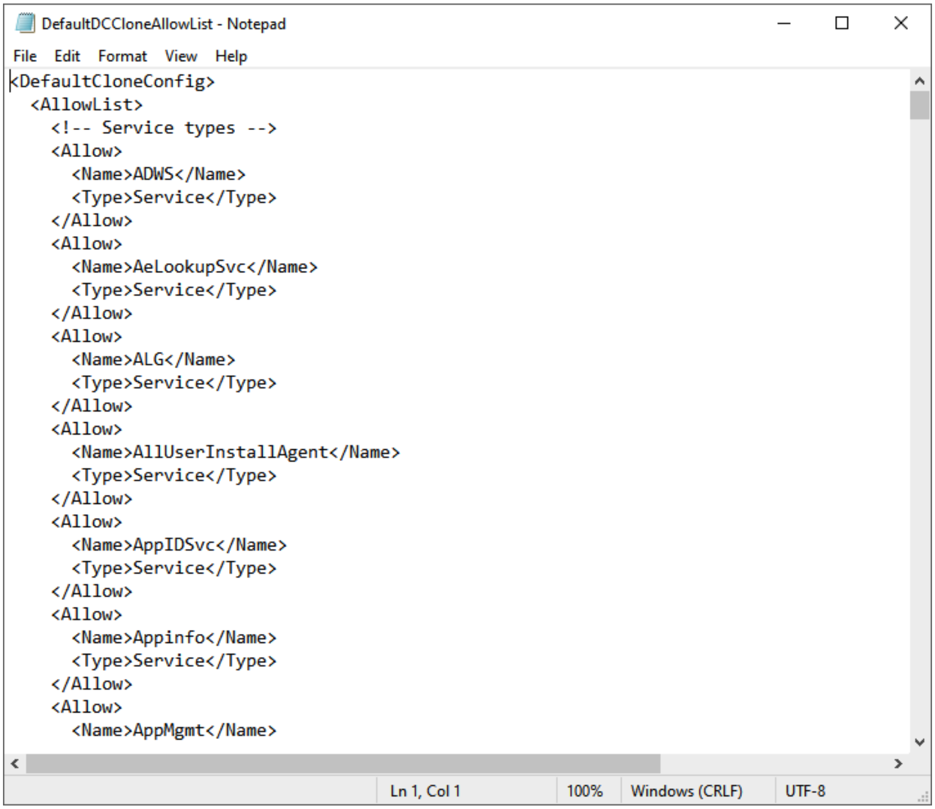
- Adding the domain controller to the Cloneable Domain Controllers group
- Cloning the domain controller from the hypervisor

Making sure all agents and software packages are cloneable

To successfully clone a domain controller, all agents and software packages that you've installed and configured on the domain controller you intend to clone need to support it.

When you install the Active Directory Domain Services role on a Windows Server 2012 installation, or on any newer version of Windows Server, there is the `Get-ADDCCloningExcludedApplicationList` PowerShell cmdlet that you can use. When you run this PowerShell cmdlet, it will return the applications and services that Microsoft does not know whether you can successfully clone.

All Microsoft services and add-on packages that ship with Windows Server are tested, so these are already part of the `DefaultDCCloneAllowList.xml` file. The contents of `C:\Windows\System32\DefaultDCCloneAllowList.xml` are shown as follows:



```
<DefaultCloneConfig>
  <AllowList>
    <!-- Service types -->
    <Allow>
      <Name>ADWS</Name>
      <Type>Service</Type>
    </Allow>
    <Allow>
      <Name>AeLookupSvc</Name>
      <Type>Service</Type>
    </Allow>
    <Allow>
      <Name>ALG</Name>
      <Type>Service</Type>
    </Allow>
    <Allow>
      <Name>AllUserInstallAgent</Name>
      <Type>Service</Type>
    </Allow>
    <Allow>
      <Name>AppIDSvc</Name>
      <Type>Service</Type>
    </Allow>
    <Allow>
      <Name>Appinfo</Name>
      <Type>Service</Type>
    </Allow>
    <Allow>
      <Name>AppMgmt</Name>
```

Figure 2.13 – Contents of the DefaultDCCloneAllowList.xml file

For any other service and/or application, the recommended practice is to ask the vendor whether domain controller cloning is supported. When all services and applications check out, you can run the following line of PowerShell to add them to your organization's `CustomDCCloneAllowList.xml` file:

```
Get-ADDCCloningExcludedApplicationList -GenerateXml -Path C:\
Windows\NTDS -Force
```

In the preceding line of Windows PowerShell, the default path for the Active Directory database is supplied. Change it accordingly before running it.

After cloning, the domain controller picks up this file when you store it on removable media or in the same path as the Active Directory database.

Supplying the information for the new domain controller configuration

The new domain controller that is created when an existing domain controller is cloned will need to be different from the existing one. It will need a different hostname, IPv4 address(es), IPv6 address(es), possibly different DNS Server allocations, or a different Active Directory site.

Microsoft provides a way to supply this information through the `DCCloneConfig.xml` file. Again, after cloning, the domain controller picks up this file when you store it on removable media or in the same path as the Active Directory database.

If no `DCCloneConfig.xml` file is supplied, the new domain controller will boot into Directory Services Restore Mode.

If an empty `DCCloneConfig.xml` file is supplied, the new domain controller will be assigned the following:

- IP addresses through DHCP
- An automatically assigned hostname
- The same Active Directory site as the source domain controller

If a specific hostname, Active Directory site, or IP address is needed, look at the parameters you can specify for `New-ADDCCloningConfig`, such as the `-SiteName`, `-CloneComputerName`, and `-Static -IPv4Address` parameters.

A sample PowerShell one-liner to create a new domain controller with the name DC04 in the Active Directory site named RemoteLocation with the correct IPv4 information would look like the following:

```
New-ADDCCloneConfigFile -CloneComputerName "DC04" -SiteName
RemoteLocation -Static -IPv4Address "10.0.1.3" -IPv4SubnetMask
"255.255.255.0" -IPv4DefaultGateway "10.0.1.1" -IPv4DNSResolver
"10.0.0.2"
```

Change the values for the -SiteName, -CloneComputerName, -Static, -IPv4Address, -IPv4SubnetMask, -IPv4DefaultGateway, and -IPv4DNSResolver parameters for parameters that make sense for your environment.

Adding the domain controller to the Cloneable Domain Controllers group

In large organizations, the team responsible for managing Active Directory is usually a different team from the one managing the hypervisor platform. Through the integration components and/or VMware tools, the latter team might configure domain controllers for cloning and clone them, adding to the management burden of the Active Directory management team.

Therefore, the Active Directory team must explicitly allow a domain controller to be cloned in Active Directory. The mechanism to do so is to add source domain controllers to the **Cloneable Domain Controllers** group.

The following line of PowerShell accomplishes this for a source domain controller named DC03 in the lucernpub.com Active Directory domain:

```
Add-ADGroupMember "Cloneable Domain Controllers"
"CN=DC03,OU=Domain Controllers,DC=LucernPub,DC=com"
```

Replace the distinguishedName value of DC03 with the distinguishedName value of the domain controller you want to add to the **Cloneable Domain Controllers** group.

Cloning the domain controller from the hypervisor

Now, the hypervisor platform team can clone the source domain controller.

As an Active Directory administrator, shut down the domain controller you intend to clone. After cloning has been successful, remove the source domain controller from the **Cloneable Domain Controllers** group and start it again as one of the domain controllers for the domain, or leave it off and allow it to be cloned repeatedly for a maximum period of 60 to 180 days, depending on the current tombstone lifetime period settings.

How it works...

Domain controller cloning leverages the VM-GenerationID feature found in most modern hypervisor platforms. Through the specifications that Microsoft wrote for this feature, this ID is stored in every virtual machine's RAM and only changes under certain circumstances. These circumstances are the following:

- When a virtual machine's hard disk is attached to a different virtual machine
- When a previous snapshot for a virtual machine is applied

Active Directory Domain Services is the first server role to take advantage of the VM-GenerationID feature to do the following:

- Increase the integrity of the contents of the Active Directory database and the Active Directory SYSVOL by employing virtualization safeguards
- Clone a perfectly prepared domain controller using domain controller cloning

By storing the 128-bit value for the VM-GenerationID in RAM in the Active Directory database, and the domain controller checking the value stored in the database with the value in RAM before each major action, the domain controller can sense when a snapshot is applied or when the hard disk is reused.

> **Important Note**
>
> As the VM-GenerationID feature is a hypervisor platform feature, a domain controller cannot sense when a snapshot is applied or when the hard disk is reused when these actions originate from the storage fabric or otherwise outside of the hypervisor platform.

When a hard disk is reused and the domain controller is properly prepared to be cloned, domain controller cloning creates a perfect clone of the source domain controller.

Domain controller cloning only allows cloning of fully writable domain controllers. It does not apply to read-only domain controllers.

See also

Use the information in the *Determining whether a virtual domain controller has a VM-GenerationID* recipe to see whether the hypervisor platform supports domain controller cloning.

Refer to the *Modifying the tombstone lifetime period* recipe in *Chapter 16*, *Hardening Azure AD*, to find out whether domain controllers can be cloned for 60 or 180 days.

Determining whether a virtual domain controller has a VM-GenerationID

One of the requirements for Active Directory Virtualization Safeguards and domain controller cloning is the ability of the hypervisor platform to provide the **VM-GenerationID** to the virtual domain controller.

How to do it…

To determine whether a virtual domain controller has the VM-GenerationID, perform these steps:

1. Sign in to the virtual domain controller.
2. Press Start.
3. Search for **Device Manager** and click its corresponding search result, or run `devmgmt.msc`. The **Device Manager** window appears.
4. In the taskbar of **Device Manager**, open the **View** menu and select **Show hidden devices**.
5. In the main pane of **Device Manager**, expand **System devices**.
6. Search for the `Microsoft Hyper-V Generation Counter` system device. The existence of such a device means the virtual domain controller has a VM-GenerationID.
7. Close **Device Manager**.

How it works...

When the hypervisor platform supports the VM-GenerationID feature, it will create a device to place the value of the VM-GenerationID in the virtual memory of the virtual domain controller.

To determine whether a virtual domain controller has a VM-GenerationID, look for this system device.

Demoting a domain controller

Every domain controller has a life cycle. After a certain period, it should make room for newer, better, more agile, or even more cost-efficient domain controllers, or other solutions, such as Azure Active Directory Domain Services.

Getting ready

Before you demote a domain controller, you should ensure of the following:

- It no longer hosts any FSMO roles.
- It no longer offers networking services, such as DNS, LDAP, RADIUS, or WINS. These protocols are manually configured on networking devices and other servers. Demoting a domain controller that offers these services might negatively impact the networking infrastructure. Reconfigure networking devices and servers to use alternative domain controllers or services first.
- It is not an Enterprise Root **Certification Authority** (**CA**). When a domain controller is configured as an Enterprise Root CA using **Active Directory Certificate Services** (**AD CS**), it cannot be demoted. First, the CA needs to be migrated.
- There are other global catalog servers available when you remove a domain controller that is also configured to be a global catalog server.

For successful demotions, the domain controller you intend to demote needs to have at least one network interface card attached to the network. Other domain controllers should be reachable and Active Directory replication should be working properly.

How to do it...

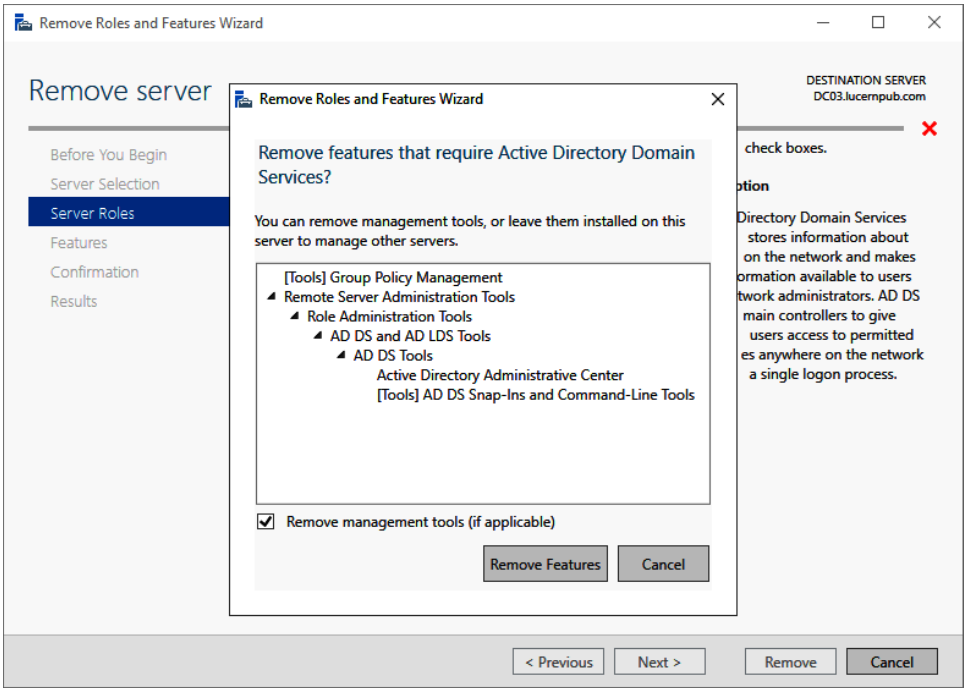
This recipe describes two supported ways to demote a domain controller graciously:

- Using the **Remove Server Roles and Features Wizard**
- Using the **Active Directory module for Windows PowerShell**

Using the Remove Server Roles and Features Wizard

To demote a domain controller graciously using Server Manager, perform these steps:

1. Press Start.
2. Search for **Server Manager** and click its corresponding search result, or run `servermanager.exe`. The **Server Manager** window appears.
3. In the gray top bar of **Server Manager**, click **Manage**.
4. Select **Remove Server Roles and Features** from the context menu. The **Remove Roles and Features Wizard** window appears.
5. On the **Before you begin** screen, click **Next**.
6. On the **Select destination server** screen, select the local Windows Server installation from the server pool list, and then click **Next**.
7. On the **Select server roles** screen, deselect the **Active Directory Domain Services** role from the list of installed roles. The **Remove Roles and Features Wizard** pop-up window appears.
8. In the pop-up window, click the **Remove Features** button to remove features that are required for Active Directory Domain Services:



Remove features that require Active Directory Domain Services?

You can remove management tools, or leave them installed on this server to manage other servers.

[Tools] Group Policy Management
Remote Server Administration Tools
Role Administration Tools
AD DS and AD LDS Tools
AD DS Tools
Active Directory Administrative Center
[Tools] AD DS Snap-Ins and Command-Line Tools

☑ Remove management tools (if applicable)

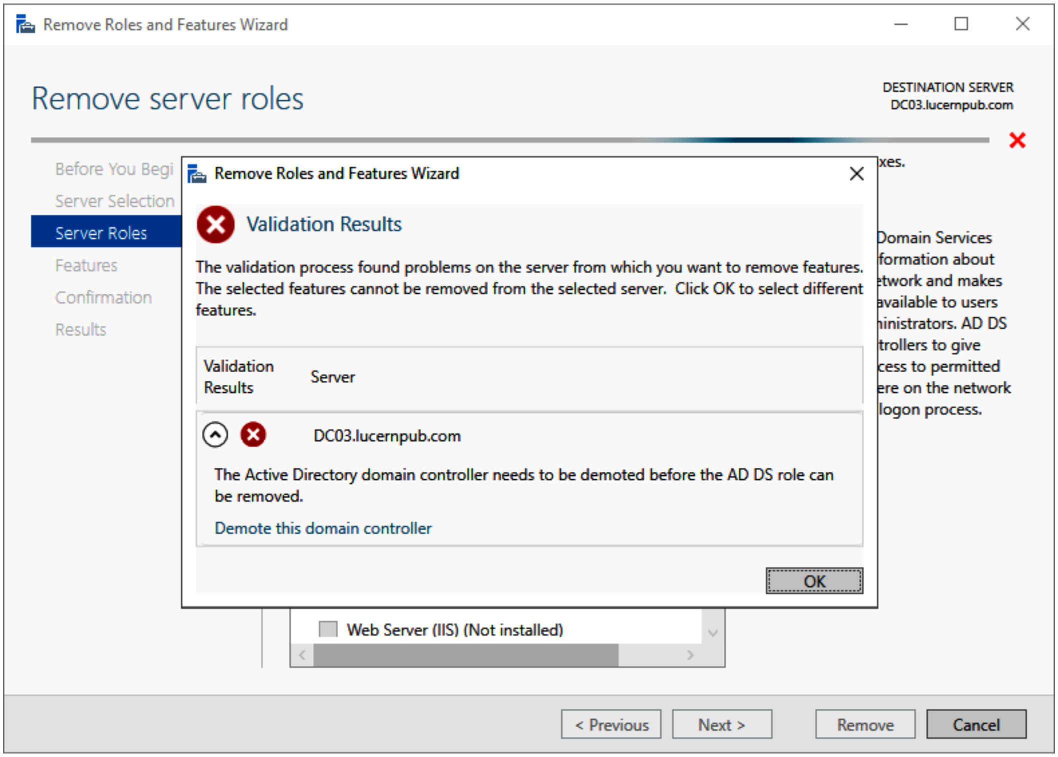
check boxes.

ption

Directory Domain Services stores information about on the network and makes ormation available to users twork administrators. AD DS main controllers to give users access to permitted es anywhere on the network a single logon process.

Figure 2.14 – The Remove Roles and Features Wizard pop-up screen

9. On the **Validation Results** screen, follow the **Demote this domain controller** link to acknowledge that the domain controller needs to be demoted before the Active Directory Domain Services role can be removed:



The validation process found problems on the server from which you want to remove features. The selected features cannot be removed from the selected server. Click OK to select different features.

DC03.lucernpub.com

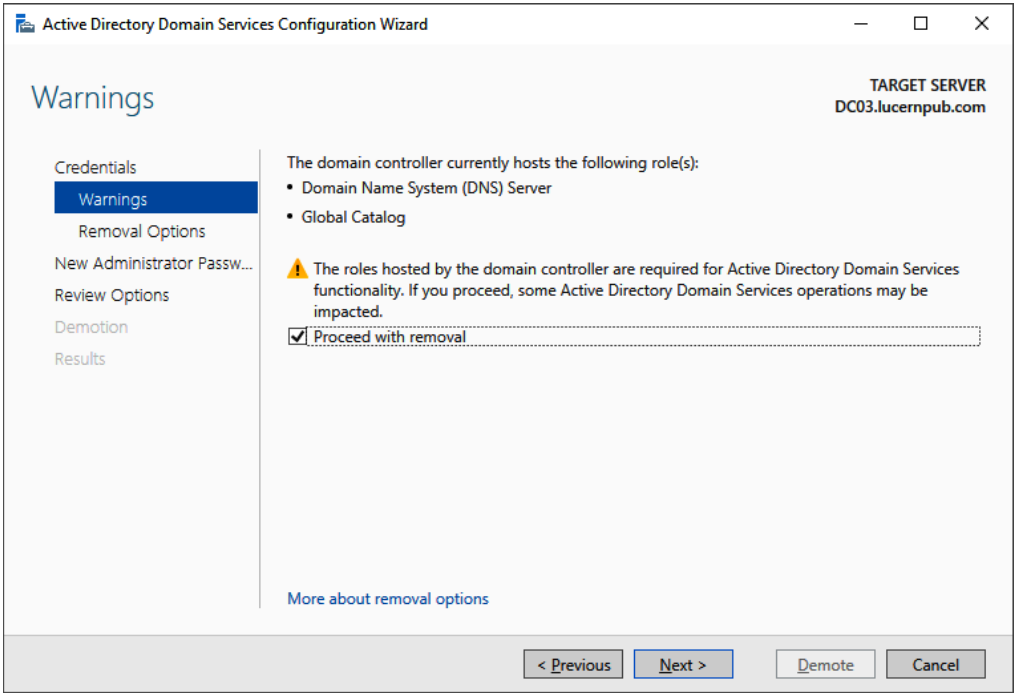
The Active Directory domain controller needs to be demoted before the AD DS role can be removed.

Demote this domain controller

Figure 2.15 – Validation Results for the Remove Roles and Features Wizard

10. The **Active Directory Domain Services Configuration Wizard window** appears. On the **Credentials** screen, optionally enter the credentials to perform the demotion, or click **Next >** to perform the operation with the credentials of the account you signed in with.

11. On the **Warnings** screen, select the **Proceed with removal** option and click **Next >**:



The domain controller currently hosts the following role(s):

- Domain Name System (DNS) Server
- Global Catalog

⚠ The roles hosted by the domain controller are required for Active Directory Domain Services functionality. If you proceed, some Active Directory Domain Services operations may be impacted.

☑ Proceed with removal

Figure 2.16 – The Warnings screen of the Active Directory Domain Services Configuration Wizard

12. On the **Removal Options** screen, select the **Remove DNS delegation** option and click **Next** >.
13. On the **New administrator password** screen, enter the new password for the built-in administrator account. Click **Next** > to proceed to the next screen.
14. On the **Review Options** screen, click **Demote**.
15. When configuration of the **Active Directory Domain Services** server role is done, click **Close** to close the Remove Roles and Features Wizard.

Using the Active Directory module for Windows PowerShell

To demote a domain controller graciously, you can use the `Uninstall-ADDSDomainController` PowerShell cmdlet like this:

```
Uninstall-ADDSDomainController
```

This removes the domain controller from the Active Directory domain and prompts you for the new password for the built-in administrator account after demotion. Replace the values in the previous sample file with the values of your choice.

To remove the **Active Directory Domain Services** role after demotion, use the following line of Windows PowerShell:

```
Uninstall-WindowsFeature AD-Domain-Services
-IncludeManagementTools
```

The domain controller is then demoted, and the **Active Directory Domain Services** role is removed.

How it works...

Every domain controller has its information stored in numerous places throughout the Active Directory database.

To remove this information and stop other domain controllers from replicating to non-existing domain controllers, the domain controllers should be demoted.

There's more...

Proper demotion of a domain controller will remove all the references to the domain controller from Active Directory.

However, it is a recommended practice to check the following tools manually after demotion:

- **DNS**: (`dnsmgmt.msc`)
- **Active Directory Sites and Services** (`dssite.msc`)

Demoting a domain controller forcefully

It's also an option to forcefully remove a domain controller from Active Directory. While graciously demoting should be the preferred option, you might have to resort to this option.

The process of demoting a domain controller forcefully consists of these steps:

- Performing a metadata cleanup
- Deleting the domain controller from DNS
- Deleting the computer object for the domain controller
- Deleting the SYSVOL replication membership

- Deleting the domain controller from Active Directory Sites and Services
- Seizing any FSMO roles that were hosted by the domain controller (you can do this first to ensure there are no impacts on domain members)
- Taking care of the existence of global catalog servers

If the domain controller was the last domain controller for a domain in an existing forest, the domain will need to be removed, as it is now an orphaned domain.

Getting ready

Although you would demote a domain controller forcefully when it no longer replicates, you should ensure that the remaining domain controllers are replicating properly.

How to do it...

This recipe describes two ways to do it:

- Using the Active Directory Domain Services Configuration Wizard
- Using manual steps

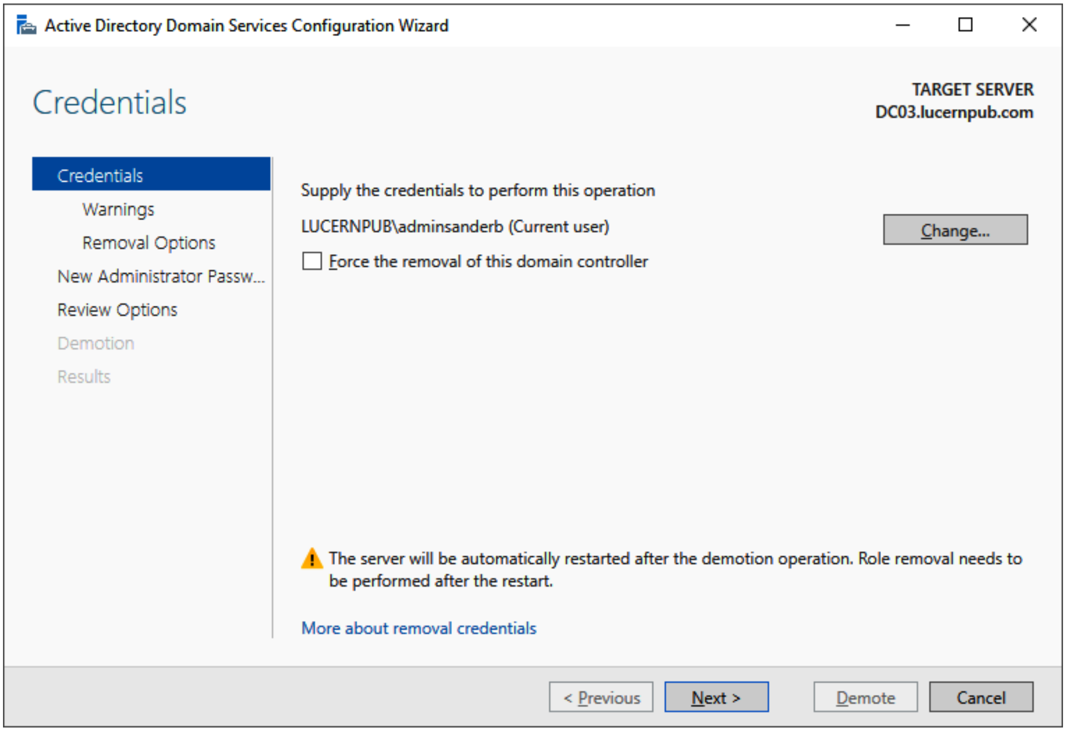
Using the Active Directory Domain Services Configuration Wizard

The Active Directory Domain Services Configuration Wizard can be used to forcefully demote a domain controller when the Windows Server installation is still bootable and you are able to sign in to it with administrative credentials.

Perform these steps:

1. Press Start.
2. Search for **Server Manager** and click its corresponding search result, or run `servermanager.exe`. The **Server Manager** window appears.
3. In the gray top bar of **Server Manager**, click **Manage**.
4. Select **Remove Server Roles and Features** from the menu.
5. On the **Before you begin** screen, click **Next** >.
6. On the **Select destination server** screen, select the local Windows Server installation from the server pool list.
7. Click **Next** >.

8. On the **Select server roles** screen, deselect the **Active Directory Domain Services** role from the list of installed roles.
9. In the pop-up window, click the **Remove Features** button to remove features that are required for Active Directory Domain Services.
10. On the **Validation Results** screen, follow the **Demote this domain controller** link to acknowledge that the domain controller needs to be demoted before the **Active Directory Domain Services** role can be removed.
11. On the **Credentials** screen, select the **Force the removal of this domain controller** option:



Active Directory Domain Services Configuration Wizard

Credentials

TARGET SERVER
DC03.lucernpub.com

Supply the credentials to perform this operation

LUCERNPUB\adminsanderb (Current user)

☐ Force the removal of this domain controller

The server will be automatically restarted after the demotion operation. Role removal needs to be performed after the restart.

More about removal credentials

Figure 2.17 – The Credentials screen of the Active Directory Domain Services Configuration Wizard

12. Click **Next** >.
13. On the **Removal options** screen, select the **Proceed with removal** option and click **Next** >.
14. On the **New administrator password** screen, enter the new password for the built-in administrator account.

15. Click **Next** > to proceed to the next screen.
16. On the **Review Options** screen, click **Demote**.
17. When configuration of the **Active Directory Domain Services** server role is done, click **Close** to close the Remove Roles and Features Wizard.

Using manual steps

Sometimes, the Active Directory Domain Services Configuration Wizard cannot be used, such as in the following situations:

- You can no longer sign in interactively or remotely to the domain controller.
- The physical hardware of the domain controller has been damaged beyond repair.
- The domain controller is no longer reachable for other domain controllers.
- The domain controller, for some reason, can no longer be trusted to provide Active Directory Domain Services in a meaningful way.

In these scenarios, the following manual steps can be performed to remove the domain controller from Active Directory.

Performing metadata cleanup

Perform these steps to perform metadata cleanup:

1. Sign in interactively to the domain controller that is known to be good. The domain controller holding the Domain Naming Master is preferred.
2. Press Start.
3. Search for **Command Prompt**, right-click its search result, and choose **Run as administrator** from the context menu. Alternatively, run `cmd.exe`, but instead of running it by pressing *Enter*, press *Ctrl*, *Shift*, and *Enter*.
4. Run the following command to start the NTDS utility in interactive mode:

```
ntdsutil.exe
```

5. Type the following command in interactive mode to start the metadata cleanup:

```
metadata cleanup
```

6. Type the following command to remove the `DC04.lucernpub.com` server:

   ```
   remove selected server
   "CN=DC04,CN=Servers,CN=RemoteLocation,CN=Sites,
   CN=Configuration,DC=LucernPub,DC=com"
   ```

7. Type the following command in interactive mode to exit the metadata cleanup context:

   ```
   quit
   ```

8. Type the following command in interactive mode to exit the NTDS utility itself:

   ```
   quit
   ```

9. Close the **Command Prompt** window.

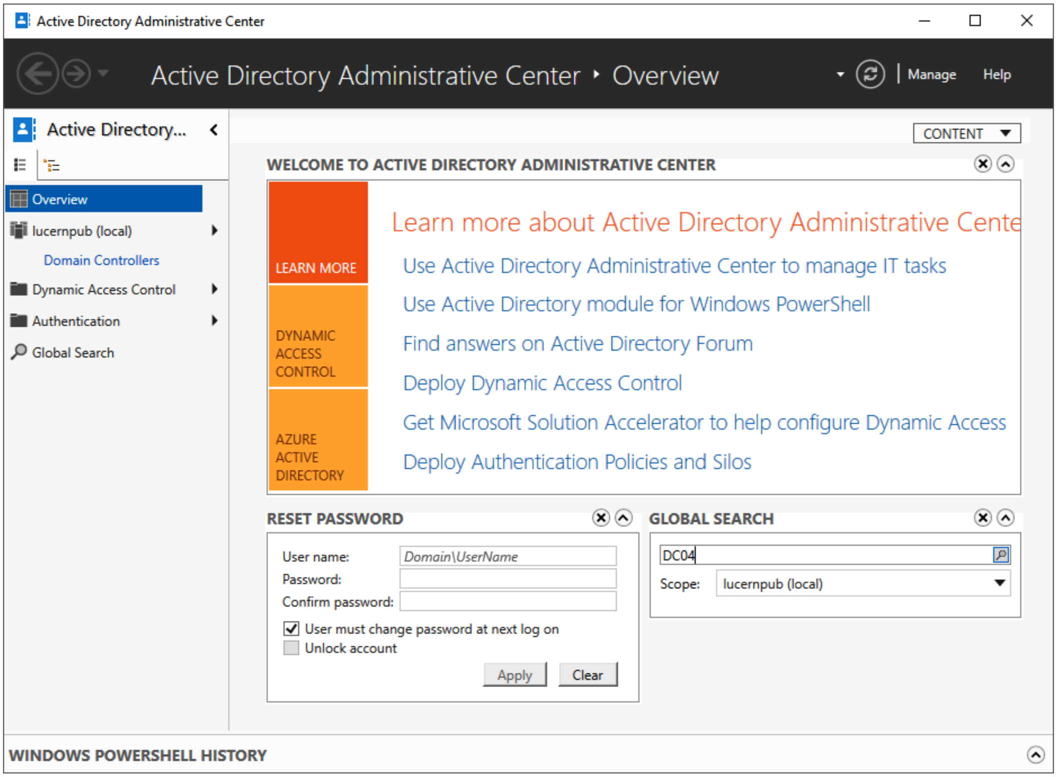
Deleting the domain controller from DNS

After the metadata cleanup, the DNS records for the domain controller may still be present. Use the DNS MMC Snap-in to remove the DNS A, AAAA, PTR, and SRV records for the domain controller.

Deleting the computer object for the domain controller

To delete the computer object for the domain controller, use the Active Directory Administrative Center:

1. Press Start.
2. Search for **Active Directory Administrative Center** and click its corresponding search result, or run `dsac.exe`. The **Active Directory Administrative Center** window appears.
3. In the main **Welcome to Active Directory Administrative Center** pane, in the **GLOBAL SEARCH** field, enter the search criteria of the desired object and then click the **Search** button:



Learn more about Active Directory Administrative Cente

Use Active Directory Administrative Center to manage IT tasks

Use Active Directory module for Windows PowerShell

Find answers on Active Directory Forum

Deploy Dynamic Access Control

Get Microsoft Solution Accelerator to help configure Dynamic Access

Deploy Authentication Policies and Silos

User name: Domain\UserName
Password:
Confirm password:
☑ User must change password at next log on
□ Unlock account

DC04
Scope: lucernpub (local)

Figure 2.18 – Searching for an object using Global Search in the Active Directory Administrative Center

4. From the search results, locate the domain controller object.
5. Right-click it and then click **Delete**.
6. Confirm you want to delete the domain controller.
7. Close the **Active Directory Administrative Center** window.

To delete the computer object for the domain controller, alternatively, run the following line of Windows PowerShell:

```
Remove-ADComputer -Identity DC04
```

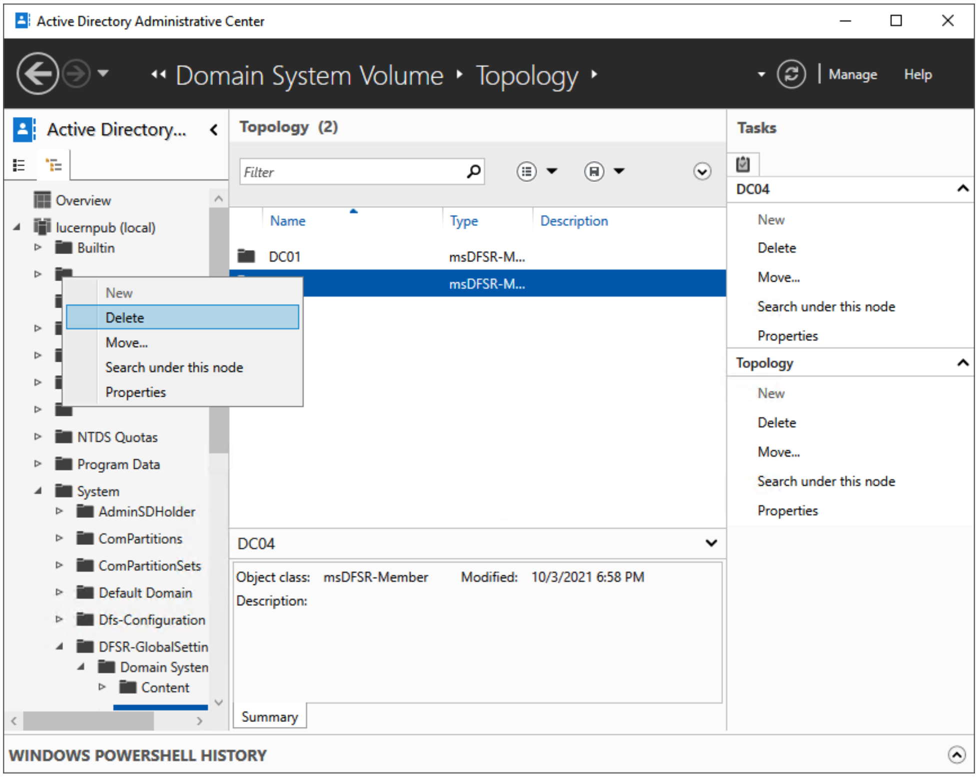
Replace `DC04` with the name of the domain controller you want to remove.

Deleting the SYSVOL replication membership

The domain controller was also a member of the replication group for the Active Directory SYSVOL.

Perform these steps to remove the domain controller:

1. Press Start.
2. Search for **Active Directory Administrative Center** and click its corresponding search result, or run `dsac.exe`. The **Active Directory Administrative Center** window appears.
3. At the top of the left navigation pane, switch to **Tree view** from **List view**.
4. Expand the Active Directory domain.
5. Expand the **System** container.
6. Expand the **DFSR-GlobalSettings** container.
7. Expand the **Domain System Volume** container.
8. Expand the **Topology** container.
9. In the main window, right-click the object for the domain controller you want to delete from the **Topology** container:



Object class: msDFSR-Member Modified: 10/3/2021 6:58 PM
Description:

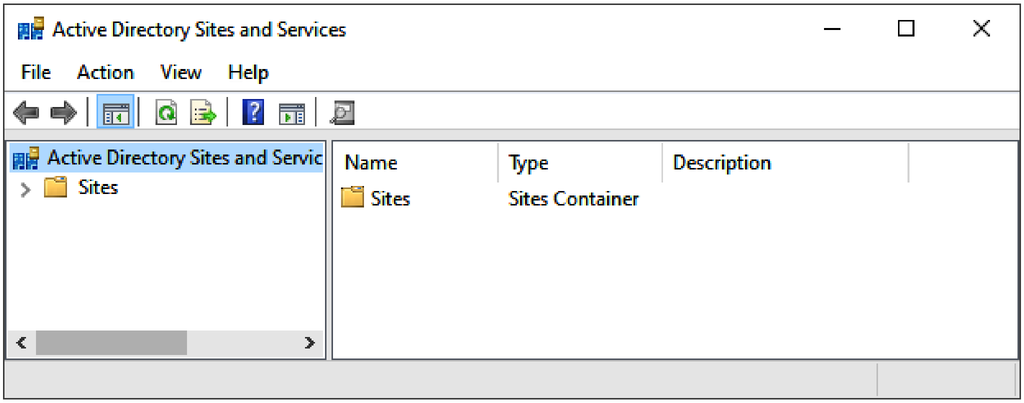
Figure 2.19 – Deleting the SYSVOL replication membership in the Active Directory Administrative Center

10. Select **Delete**.
11. In the **Delete Confirmation** pop-up window, click **Yes** to acknowledge that you are sure you want to delete **msDFSR-Member**.
12. Close the **Active Directory Administrative Center** window.

Deleting the domain controller from Active Directory Sites and Services

To delete the domain controller from Active Directory Sites and Services, perform these steps:

1. Press Start.
2. Search for **Active Directory Sites and Services** and click its corresponding search result, or run `dssite.msc`. The **Active Directory Sites and Services** window appears:


Active Directory Sites and Services
File
Action
View
Help
Active Directory Sites and Servic
Sites

Name	Type	Description
Sites	Sites Container	

Figure 2.20 – The Active Directory Sites and Services window

3. In the left navigation pane, expand **Sites**.
4. Expand the Active Directory site where the domain controller resides.
5. Expand the **Servers** node.
6. Right-click the object for the domain controller you want to delete from the **Servers** node and select **Delete**.
7. In the **Active Directory Domain Services** pop-up window, click **Yes** to acknowledge that you are sure you want to delete the server.
8. Close **Active Directory Sites and Services**.

Deleting an orphaned domain

When you've removed the last domain controller for a domain, it becomes an orphaned domain. Perform these steps to perform a metadata cleanup for the orphaned domain:

1. Sign in interactively to the domain controller that is known to be good with an account that is a member of the **Enterprise Admins** group. A domain controller that hasn't experienced any replication challenges throughout its lifetime might be your best choice. Also, note that performing the following actions on the domain controller holding the Domain Naming Master FSMO role is preferred.
2. Press Start.
3. Search for **Command Prompt**, right-click its search result, and choose **Run as administrator** from the context menu. Alternatively, run `cmd.exe`, but instead of running it by pressing *Enter*, press *Ctrl*, *Shift*, and *Enter*.
4. Run the following command to start the NTDS utility in interactive mode:

   ```
   ntdsutil.exe
   ```

5. Type the following command in interactive mode to start the metadata cleanup:

   ```
   metadata cleanup
   ```

6. Type the following commands to specify the current domain controller as the server on which to make the changes, as it is the Domain Naming Master:

   ```
   Connections
   connect to server localhost
   quit
   ```

7. Type the following commands to list the Active Directory domains in the Active Directory forest so that you can select the domain you wish to remove:

   ```
   select operation target
   list domains
   ```

8. This outputs a list of domains in the forest, denoted by an identifier. Note down the identifier for the orphaned domain you want to remove.
9. Type the following command in interactive mode to select the domain to remove:

   ```
   select domain <ID>
   ```

10. Type the following command to exit the operation target selection context:

    ```
    quit
    ```

11. Type the following command to remove the domain:

    ```
    remove selected domain
    ```

12. Type the following command in interactive mode to exit the metadata cleanup context and then the NTDS utility itself:

    ```
    quit
    quit
    ```

13. Close the **Command Prompt** window.

See also

To seize the FSMO roles, see the *Managing FSMO roles* recipe in *Chapter 3*, *Managing Active Directory Roles and Features*.

To configure domain controllers as global catalog servers, see the *Managing global catalogs* recipe in *Chapter 3*, *Managing Active Directory Roles and Features*.

Inventory domain controllers

It's a good thing to know all the domain controllers throughout an Active Directory domain. This activity doesn't just show the management burden for Active Directory administrators; it also allows them to make smart choices, especially when the environment is breached.

Although it's not recommended practice, administrators may place domain controllers outside the Domain Controllers **Organizational Unit** (**OU**). In that case, simply checking the computer accounts in that OU will not provide a 100% view of the domain controllers in use.

How to do it...

This recipe shows two ways to get a good overview of the domain controllers in an Active Directory domain:

- Using **Active Directory Users and Computers**
- Using the Active Directory module for Windows PowerShell

Using Active Directory Users and Computers to inventory domain controllers

Active Directory Users and Computers allows for querying the entire Active Directory domain for either **writable domain controllers** or **read-only domain controllers** in the following way:

1. Press Start.
2. Search for **Active Directory Users and Computers** and click its corresponding search result, or run `dsa.msc`. The **Active Directory Users and Computers** window appears.
3. In the left navigation pane, right-click the target domain name with which you want to inventory the domain controllers.
4. Select **Find**.
5. From the **Find** drop-down box, select **Computers**.
6. From the **Role** drop-down box, select **Writable Domain Controllers** or **Read-Only Domain Controllers**.
7. Click **Find Now**.

The list of domain controllers for the domain is now shown in the search results pane.

Using the Active Directory module for Windows PowerShell to inventory domain controllers

Using the Active Directory module for Windows PowerShell to inventory domain controllers is even easier.

Simply use the following line of Windows PowerShell:

```
Get-ADDomainController | Select-Object Name
```

If you want more information on the domain controllers within the current domain, simply add the characteristics you would like to see after the `Select-Object` cmdlet. For instance, you can add `IPv4Address`, `IsGlobalCatalog`, `isReadOnly`, `OperatingSystem`, and `Site` for good measure. If you're looking for a smart layout, simply append `| Format-Table`. If you want to get the information straight to your clipboard so that you can paste it into a report or anywhere else, append | `clip`.

Decommissioning a compromised read-only domain controller

One of the benefits of deploying read-only domain controllers is their ability to recover quickly from an information security breach.

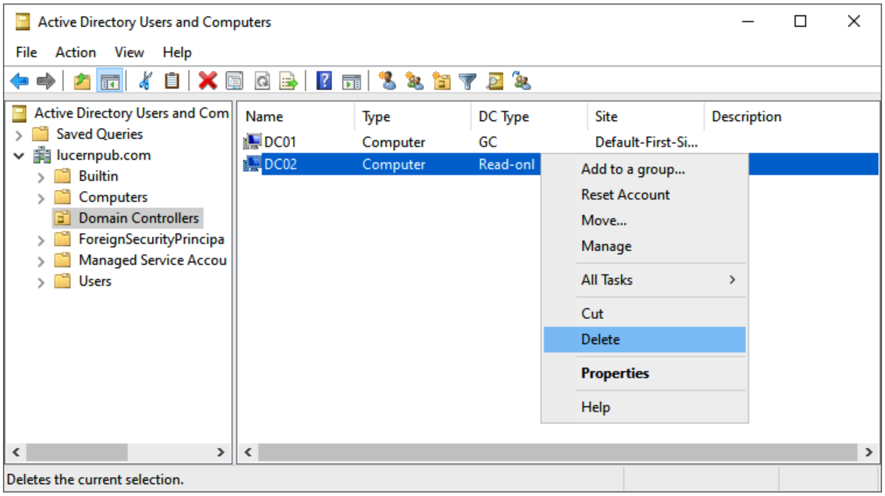
Since only the passwords for a subset of users are cached on the read-only domain controller when these users signed on through the read-only domain controller and the passwords for really sensitive accounts weren't allowed to be cached on the read-only domain controller, the impact of a stolen read-only domain controller is small, compared to a fully writable domain controller.

How to do it...

To render the read-only domain controller useless to an attacker or thief, perform these steps:

1. Press Start.
2. Search for **Active Directory Users and Computers** and click its corresponding search result, or run `dsa.msc`. The **Active Directory Users and Computers** window appears.
3. In the left navigation pane, expand the domain name.
4. In the left navigation pane, expand the **Domain Controllers** OU.

5. Right-click the compromised read-only domain controller and select **Delete**:



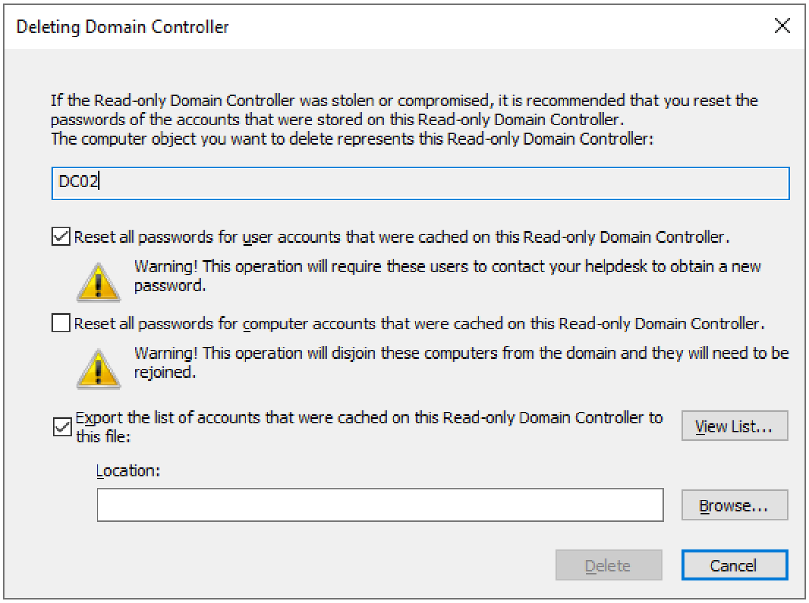
Name	Type	DC Type	Site	Description
DC01	Computer	GC	Default-First-Si...	
DC02	Computer	Read-onl		

Deletes the current selection.

Figure 2.21 – Deleting a read-only domain controller in Active Directory Users and Computers

6. In the **Active Directory Domain Services** pop-up window, click **Yes** to answer the **Are you sure you want to delete the Computer?** question.

7. On the **Deleting Domain Controller** screen, select the **Reset all passwords for user accounts that were cached on this Read-only Domain Controller** option:



If the Read-only Domain Controller was stolen or compromised, it is recommended that you reset the passwords of the accounts that were stored on this Read-only Domain Controller.
The computer object you want to delete represents this Read-only Domain Controller:

DC02

☑ Reset all passwords for user accounts that were cached on this Read-only Domain Controller.

Warning! This operation will require these users to contact your helpdesk to obtain a new password.

☐ Reset all passwords for computer accounts that were cached on this Read-only Domain Controller.

Warning! This operation will disjoin these computers from the domain and they will need to be rejoined.

☑ Export the list of accounts that were cached on this Read-only Domain Controller to this file:

Location:

Figure 2.22 – The Deleting Domain Controller window

8. Since the persons associated with the user accounts will be forced to have their passwords reset by service desk personnel, it's a recommended practice to also check the **Export the list of accounts that were cached on this Read-only Domain Controller to this file** option and to specify a file. This way, the service desk may proactively approach affected colleagues.
9. Click **Delete**.
10. In the **Delete Domain Controller** pop-up window, click **OK**.
11. In the **Delete Domain Controller** pop-up window, click **Yes** to continue with this deletion.

How it works...

Each read-only domain controller caches the hashes of the passwords for users signing in through the read-only domain controller. For this functionality, the read-only domain controller contacts a writable domain controller.

When a user account is denied having its password cached, the password is not cached. For accounts on which the passwords have been cached, the best remedy is to reset these passwords.

Every Kerberos ticket that is given to devices or user accounts is encrypted using the separate `krbtgt` account for the read-only domain controller. These tickets are bound to the read-only domain controller. When the read-only domain controller is removed from the Active Directory domain, these Kerberos tickets become useless.

3
Managing Active Directory Roles and Features

Creating domain controllers is fun but leveraging all of them is key to implementing a responsive and highly available Active Directory. Therefore, we'll focus on the differences between domain controllers in terms of **Flexible Single Master Operations** (**FSMO**) roles and global catalog configuration. There's a reason why the following quote, inspired by George Orwell's *Animal Farm*, is popular among Active Directory admins:

"All domain controllers are equal, but some domain controllers are more equal than others."

The following recipes are covered in this chapter:

- Querying FSMO role placement
- Transferring FSMO roles
- Seizing FSMO roles
- Configuring the Primary Domain Controller emulator to synchronize time with a reliable source

- Managing time synchronization for virtual domain controllers
- Managing global catalogs

Before we begin with the recipes, we will look at FSMO roles and practices.

About FSMO roles

Active Directory uses the multi-master model, where domain controllers are all able to respond to client requests. When comparing to the old Windows NT 4 server-style **Primary Domain Controller** (**PDC**) and **Backup Domain Controller** (**BDC**) models, where BDCs were read-only until switched to become the only PDC, this model offers many benefits.

However, some tasks within Active Directory don't work well with the multi-master model. When designing Active Directory, Microsoft created five roles outside of the model, labeled FSMO roles, to prevent conflicting updates.

These roles can be flexibly assigned to domain controllers and seized when a domain controller holding the role fails. These roles offer single master operations within the scope of the roles, hence the name "FSMO role."

In every Active Directory environment, there are five roles with two different scopes:

Role	**Scope**
Domain Naming Master	Active Directory forest
Schema Master	Active Directory forest
PDC Emulator	Active Directory domain
RID Master	Active Directory domain
Infrastructure Master	Active Directory domain

Table 3.1 – FSMO roles and their scopes

In an environment with multiple domains in a forest, the FSMO roles with the domain scope will be present for each domain, where the two FSMO roles for the forest would only be represented once.

The roles for each of these FSMO roles can best be described as follows:

- **Domain Naming Master**: The domain controller holding the Domain Naming Master FSMO role is responsible for changes to the forest-wide domain namespaces. The domain controller with this FSMO role is the only domain controller that can add and remove domains to and from the forest.

- **Schema Master**: The domain controller holding the Schema Master FSMO role is responsible for changes to the schema partition. The domain controller with this FSMO role is the only domain controller that can make changes to the Active Directory schema. It is the source of authority for replicating schema changes.
- **PDC Emulator**: The domain controller acting as the PDC Emulator FSMO role has many responsibilities in the Active Directory domain. It is the authority for time synchronization, replicating password changes, and resolving authentication errors due to recently changed passwords through PDC chaining.
- **RID Master**: The domain controller holding the RID Master FSMO role is responsible for handing out pools of RIDs to all the other domain controllers in the domain (and itself). These identifiers are used to create SIDs by appending to the domain SID namespace. Because SIDs are used to authorize access to resources, it is important to not hand out the same RID to multiple accounts. Before a domain controller reaches the end of its RID Pool, it asks the domain controller holding the RID Master FSMO role for a new RID Pool. By default, RID Pools contain 500 RIDs.
- **Infrastructure Master**: The domain controller holding the Infrastructure Master FSMO role is responsible for updating objects in cross-domain object references.

When the Active Directory Recycle Bin optional feature is enabled, every domain controller becomes responsible for updating cross-domain object references. In this case, the Infrastructure Master FSMO role can be neglected, including the placement considerations for global catalog servers.

Recommended practices for FSMO roles

Microsoft has published the following recommendations as part of Knowledge Base article 223346 (`https://support.microsoft.com/en-us/help/223346`):

- Place the schema master on the PDCE of the forest root domain.
- Place the domain naming master on the forest root PDCE.
- Place the PDCE on your best hardware in a reliable hub site that contains replica domain controllers in the same Active Directory site and domain.
- Place the RID master on the domain PDCE in the same domain.

For a single domain in a single forest with the Active Directory Recycle Bin optional feature enabled, this means that the PDCE and RID Master FSMO roles can best be placed on one domain controller and the Schema Master and Domain Naming Master FSMO roles on another domain controller.

For an Active Directory forest with multiple domains, it means the forest-wide Schema Master and Domain Naming Master FSMO roles can best be placed on one domain controller and the PDCE and RID Master roles for the forest-root domain on another domain controller. For every other domain, the PDCE and RID Master roles are to be placed on a domain controller per domain. Whether or not the Infrastructure Master FSMO role plays any part depends on the Active Directory Recycle Bin.

The recommended practices may not be followed in every Active Directory environment, indicating the need to manage the FSMO roles.

Querying FSMO role placement

Follow this recipe to find out which domain controllers run which FSMO roles.

Getting ready

To query FSMO roles, sign in with a domain account.

How to do it...

To locate the domain controllers running the FSMO roles, run the following command on any domain-joined device, member server, or domain controller:

```
netdom.exe query fsmo
```

Or use the following lines of Windows PowerShell on a domain-joined system that has the Active Directory Module for Windows PowerShell installed:

```
Get-ADForest | Format-List DomainNamingMaster,SchemaMaster
Get-ADDomain | Format-List
InfrastructureMaster,PDCEmulator,RIDMaster
```

You can insert the last line in a **foreach** loop when you have multiple domains in the forest.

How it works...

Domain controllers hold FSMO roles. Each FSMO role is automatically assigned to one domain controller. This information is stored in the Active Directory database. The information for the domain controller holding the Schema Master and the information for the domain controller holding the Domain Naming Master is stored at the forest level. The information for the domain controllers holding the other roles is stored at the domain level.

Transferring FSMO roles

Use this recipe to transfer FSMO roles between domain controllers.

Getting ready

When both the domain controller holding a FSMO role and the domain controller you intend to transfer the FSMO role to are both online, functioning and replicating properly, you can transfer FSMO roles without problems. If you can't transfer successfully, you can seize the FSMO roles.

To transfer the Schema Master and/or Domain Naming Master FSMO roles, you'll need to be signed in with an account that is a member of the **Enterprise Admins** group. To transfer the other FSMO roles, you'll need to be signed in with an account that is a member of the **Domain Admins** group.

How to do it...

You can transfer FSMO roles using one of the following snap-ins:

Tool	**FSMO role**
Active Directory Users and Computers MMC snap-in (`dsa.msc`)	RID Master PDC Emulator Infrastructure Master
Active Directory Domains and Trusts MMC snap-in (`domain.msc`)	Domain Naming Master
Active Directory schema MMC snap-in	Schema Master

Table 3.2 – MMC Snap-ins to transfer FSMO roles

Alternatively, you can use the `ntdsutil.exe` command-line tool or the `Move-ADDirectoryServerOperationMasterRole` PowerShell cmdlet from the Active Directory Module for Windows PowerShell.

Transferring FSMO roles using the MMC snap-ins

To transfer the Domain Naming Master FSMO role, perform these steps while on a domain controller in the root forest domain, signed in with an account that is a member of the **Enterprise Admins** group:

1. Press Start.
2. Search for **Active Directory Domains and Trusts** and click its search result or run `domain.msc`. The **Active Directory Domains and Trusts** window appears.
3. In the left navigation pane, right-click **Active Directory Domains and Trusts** and select **Change Active Directory Domain Controller**.
4. Select the domain controller you want to transfer the Domain Naming Master FSMO role to from the list of domain controllers.
5. Click **OK** when you're done.
6. In the left navigation pane, right-click **Active Directory Domains and Trusts** and select **Operation Master…**:

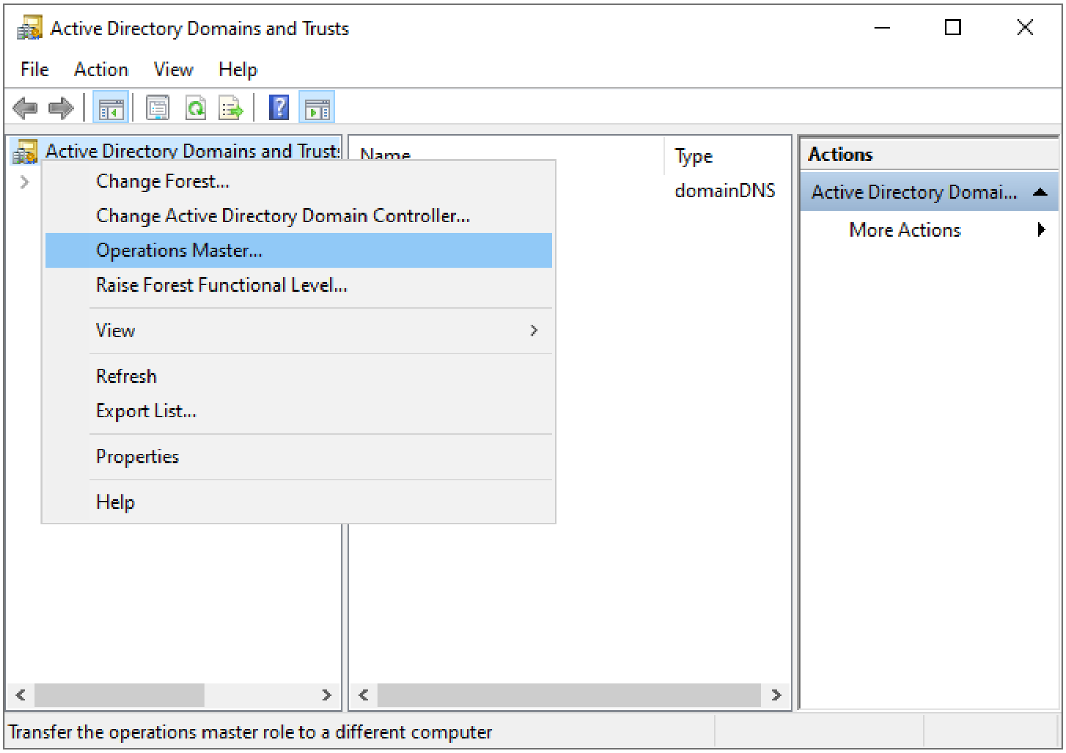

Active Directory Domains and Trusts
File
Action
View
Help
Active Directory Domains and Trusts
Name
Type
domainDNS
Actions
Active Directory Domai...
More Actions
Change Forest...
Change Active Directory Domain Controller...
Operations Master...
Raise Forest Functional Level...
View
Refresh
Export List...
Properties
Help
Transfer the operations master role to a different computer

Figure 3.1 – Operations Master in the context menu for Active Directory Domains and Trusts

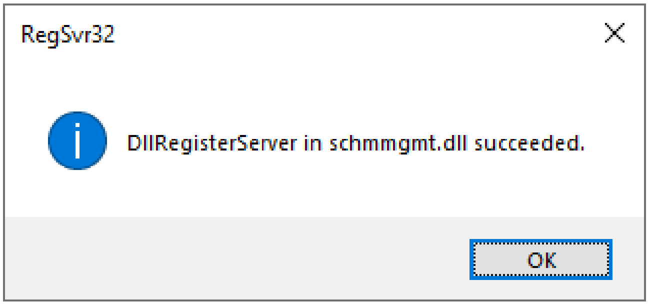
7. Click **Change**.
8. Click **Yes** to answer the question **Are you sure you want to transfer the operations master role to a different computer?** in the **Active Directory Domains and Trusts** pop-up window.
9. Click **OK** to acknowledge that **The operations master was successfully transferred** in the second **Active Directory Domains and Trusts** pop-up window.
10. Click **Close** to close the **Operations Master** window.
11. Close **Active Directory Domains and Trusts**.

To transfer the Domain Naming Master FSMO role, perform these steps while on a domain controller in the root forest domain, signed in with an account that is a member of the **Enterprise Admins** group:

1. Press Start.
2. Search for **Command Prompt**, right-click its search result and choose **Run as administrator** from the context menu. Alternatively, run `cmd.exe`, but instead of running it by pressing *Enter*, press *Ctrl* + *Shift* + *Enter*.
3. Register the `schmmgmt.dll` library on Command Prompt (`cmd.exe`) with the following command:

```
regsvr32.exe schmmgmt.dll
```

4. Click **OK** in the **RegSvr32** pop-up screen stating **DllRegisterServer in schmmgmt.dll succeeded**:



DllRegisterServer in schmmgmt.dll succeeded.

Figure 3.2 – The RegSvr32 pop-up screen stating DllRegisterServer in schmmgmt.dll succeeded

5. Run the following command to start the **Microsoft Management Console** (**MMC**):

```
mmc.exe
```

6. In the MMC screen, open the **File** menu from the taskbar.
7. Click **Add/Remove Snap-in** in the **File** menu.
8. In the **Add or Remove Snap-ins** screen, in the right list of **Available Snap-ins**, select **Active Directory Schema**.
9. Click **Add** > to add the snap-in to the list of **Selected snap-ins**.
10. Click **OK** to close the **Add or Remove Snap-ins** screen.
11. In the left navigation pane, right-click **Active Directory Schema** and select **Change Active Directory Domain Controller**.
12. Select the domain controller you want to transfer the Schema Master FSMO role to from the list of domain controllers and click **OK** when done.
13. Click **OK** in the **Active Directory Schema** pop-up window to acknowledge that the **Active Directory Schema snap-in is not connected to the schema operations master. You will not be able to perform any changes. Schema modifications can only be made on the schema FSMO holder**.
14. Right-click **Active Directory Schema** again and select **Operations Master** from the menu.
15. Click **Change**.
16. Click **Yes** to answer the question **Are you sure you want to change the Operations Master?** in the **Active Directory Schema** pop-up window.
17. Click **OK** to acknowledge that the operations master transferred successfully.
18. Click **Close** to close the **Operations Master** window.
19. Close the **Microsoft Management Console**. If the Active Directory schema is useful as an MMC snap-in, you can optionally save the console settings.

To transfer the RID Master, PDC Emulator, and Infrastructure Master FSMO roles, perform these steps on a domain controller, logged in with an account that is a member of the **Domain Admins** group:

1. Press Start.
2. Search for **Active Directory Users and Computers** and click its search result or run `dsa.msc`. The **Active Directory Users and Computers** window appears.
3. In the left navigation pane, right-click **Active Directory Users and Computers** and select **Change Active Directory Domain Controller...**.

4. Select the domain controller you want to transfer the RID Master, PDC emulator, and/or Infrastructure Master FSMO roles to from the list of domain controllers.
5. Click **OK** when done.
6. In the left navigation pane, right-click the domain name and select **Operations Masters…**:

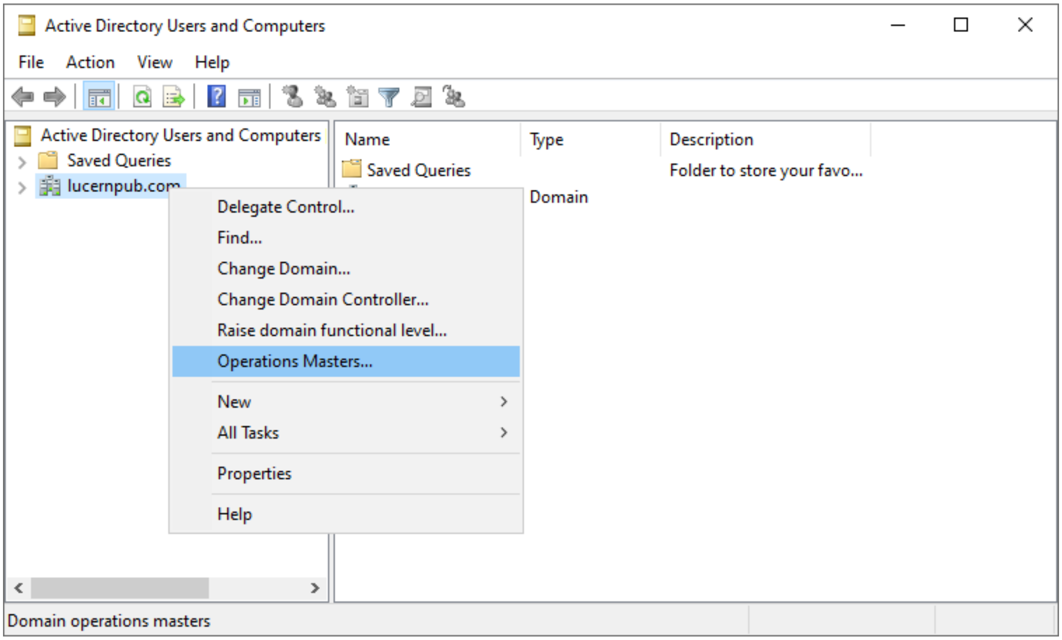

Active Directory Users and Computers
File
Action
View
Help
Active Directory Users and Computers
Saved Queries
lucernpub.com
Name
Type
Description
Saved Queries
Folder to store your favo...
Domain
Delegate Control...
Find...
Change Domain...
Change Domain Controller...
Raise domain functional level...
Operations Masters...
New
All Tasks
Properties
Help
Domain operations masters

Figure 3.3 – Operations Masters in Active Directory Users and Computers

7. Select the appropriate tab for the FSMO role you intend to transfer and click **Change**.
8. Click **Yes** to answer the question **Are you sure you want to transfer the operations master role?** in the **Active Directory Domain Services** pop-up window.
9. Click **OK** to acknowledge that **The operations master was successfully transferred** in the second **Active Directory Domain Services** pop-up window.
10. Repeat steps 7 through 9 to transfer the other FSMO roles, if applicable.
11. Click **Close** to close the **Operations Master** window.
12. Close **Active Directory Users and Computers**.

Transferring FSMO roles using the ntdsutil command-line tool

To transfer FSMO roles using the `ntdsutil.exe` command-line tool, perform these steps:

1. Sign in with an account that is a member of the **Enterprise Admins** group on the domain controller that you intend to transfer the FSMO roles to.
2. Press Start.
3. Search for **Command Prompt** and click its search result or run `cmd.exe`. The **Command Prompt** window appears.
4. Run the following command to start the NTDS utility in interactive mode:

```
ntdsutil.exe
```

5. Type the following command in interactive mode to enter the FSMO maintenance context:

```
roles
```

6. Type the following commands to specify the current domain controller as the server to transfer the FSMO roles to:

```
Connections
connect to server localhost
quit
```

7. Type the following command to transfer the Schema Master FSMO role:

```
transfer schema master
```

8. Type the following command to transfer the Domain Naming FSMO role:

```
transfer naming master
```

9. Type the following command to transfer the PDC Emulator FSMO role:

```
transfer PDC
```

10. Type the following command to transfer the RID Master FSMO role:

```
transfer RID master
```

11. Type the following command to transfer the Infrastructure Master FSMO role:

```
transfer infrastructure master
```

12. Type the following command to exit the FSMO maintenance context:

    ```
    quit
    ```

13. Type the following command in interactive mode to exit the NTDS utility:

    ```
    quit
    ```

14. Close the **Command Prompt** window.

Transferring FSMO roles using Windows PowerShell

To transfer FSMO roles using the `Move-ADDirectoryServerOperationMasterRole` PowerShell cmdlet from the Active Directory Module for Windows PowerShell, use the following lines:

```
Move-ADDirectoryServerOperationMasterRole -Identity "DC01"
-OperationMasterRole SchemaMaster
Move-ADDirectoryServerOperationMasterRole -Identity "DC01"
-OperationMasterRole DomainNamingMaster
Move-ADDirectoryServerOperationMasterRole -Identity "DC01"
-OperationMasterRole PDCEmulator
Move-ADDirectoryServerOperationMasterRole -Identity "DC01"
-OperationMasterRole RIDMaster
Move-ADDirectoryServerOperationMasterRole -Identity "DC01"
-OperationMasterRole InfrastructureMaster
```

How it works...

FSMO roles can be transferred between domain controllers to allow for scheduled maintenance or redistribution of FSMO roles between domain controllers.

To accommodate domain controller life cycles, FSMO roles need to be transferable between domain controllers, so domain controllers that are no longer needed can be decommissioned. FSMO roles are not automatically transferred when a domain controller is decommissioned.

Seizing FSMO roles

When the domain controller that holds an FSMO role is no longer available and there is no prospect that the domain controller will ever be restored, then transferring the FSMO role is no longer an option. In this case, seizing the FSMO role is the best way to go.

Ensure the remaining domain controllers are replicating properly.

Getting ready

To seize the Schema Master and/or Domain Naming Master FSMO roles, you'll need to be signed in with an account that is a member of the **Enterprise Admins** group. To seize the other FSMO roles, you'll need to be signed in with an account that is a member of the **Domain Admins** group.

How to do it...

Seizing FSMO roles is not available in the GUI but can be accomplished using the `ntdsutil.exe` command-line tool and the `Move-ADDirectoryServerOperationMasterRole` PowerShell cmdlet from the Active Directory Module for Windows PowerShell.

Seizing FSMO roles using the ntdsutil command-line tool

To transfer FSMO roles using the `ntdsutil.exe` command-line tool, perform these steps:

1. Sign in with an account that is a member of the **Enterprise Admins** group on the domain controller that you intend to transfer the FSMO roles to.
2. Press Start.
3. Search for **Command Prompt** and click its search result or run `cmd.exe`. The **Command Prompt** window appears.
4. Run the following command to start the NTDS utility in interactive mode:

   ```
   ntdsutil.exe
   ```

5. Type the following command in interactive mode to enter the FSMO maintenance context:

   ```
   roles
   ```

6. Type the following commands to specify the current domain controller as the server on which to seize the FSMO roles:

   ```
   Connections
   connect to server localhost
   quit
   ```

7. Type the following command to seize the Schema Master FSMO role:

   ```
   seize schema master
   ```

8. Type the following command to seize the Domain Naming Master FSMO role:

   ```
   seize naming master
   ```

9. Type the following command to seize the PDC Emulator FSMO role:

   ```
   seize PDC
   ```

10. Type the following command to seize the RID Master FSMO role:

    ```
    seize RID master
    ```

11. Type the following command to seize the Infrastructure Master FSMO role:

    ```
    seize infrastructure master
    ```

12. Type the following command to exit the FSMO maintenance context:

    ```
    quit
    ```

13. Type the following command in interactive mode to exit the NTDS utility:

    ```
    quit
    ```

14. Close the **Command Prompt** window.

Seizing FSMO roles using Windows PowerShell

To seize FSMO roles using the `Move-ADDirectoryServerOperationMasterRole` PowerShell cmdlet from the Active Directory Module for Windows PowerShell, use the following lines:

```
Move-ADDirectoryServerOperationMasterRole -Identity "DC01"
-OperationMasterRole SchemaMaster -Force
Move-ADDirectoryServerOperationMasterRole -Identity "DC01"
```

```
 -OperationMasterRole DomainNamingMaster -Force
Move-ADDirectoryServerOperationMasterRole -Identity "DC01"
 -OperationMasterRole PDCEmulator -Force
Move-ADDirectoryServerOperationMasterRole -Identity "DC01"
 -OperationMasterRole RIDMaster -Force
Move-ADDirectoryServerOperationMasterRole -Identity "DC01"
 -OperationMasterRole InfrastructureMaster -Force
```

Notice the `-Force` parameter that has been added. This parameter indicates seizing the FSMO role as opposed to not including it.

How it works...

A situation may occur where, for whatever reason, a domain controller is no longer usable. In this case, the FSMO role needs to be seized on another domain controller to ensure the multi-master model is retained.

Configuring the PDC Emulator to synchronize time with a reliable source

The domain controller holding the PDC Emulator FSMO role in the forest root domain is the authoritative source for time in an Active Directory domain in the default time synchronization hierarchy.

Getting ready

Before a Windows Server installation can synchronize time, the **Network Time Protocol** (**NTP**) should be available. By default, NTP is allowed inbound to domain controllers through their Windows Firewalls. However, NTP traffic toward the internet might not be available.

When an organization has deployed a reliable time source within the network, with, for instance, a GPS-enabled network time appliance, then the IP address or the hostname for this appliance can be used to configure the domain controller holding the PDC Emulator FSMO role to synchronize time with a reliable source.

In other scenarios, synchronizing time with a reliable source depends on the availability of a reliable internet-based time source. In this case, my recommendation is to use a list of sources, some denoted as DNS names and others denoted as IPv4 or IPv6 addresses. This way, the domain controller holding the PDC Emulator FSMO role can synchronize time, even in the case of missing DNS resolution.

For a list of available servers, refer to `http://support.ntp.org/bin/view/Servers/WebHome`.

UDP port `123` should be allowed from the domain controller to NTP servers on the internet, except for networking infrastructures, where dedicated NTP appliances are deployed and the domain controller holding the PDC Emulator role synchronizes with these hosts.

How to do it...

Perform these steps to configure the domain controller holding the PDC Emulator FSMO role to synchronize time with a reliable source:

1. Sign in to the domain controller holding the PDC Emulator FSMO role.
2. Press Start.
3. Search for **Command Prompt**, right-click its search result, and choose **Run as administrator** from the context menu. Alternatively, run `cmd.exe`, but instead of running it by pressing *Enter*, press *Ctrl* + *Shift* + *Enter*. The **Command Prompt** window appears.
4. Run the following commands:

```
w32tm.exe /config /manualpeerlist:"europe.pool.ntp.
org time.nist.gov 192.43.244.18 193.67.79.202" /
syncfromflags:manual /reliable:yes /update
net.exe stop w32time && net.exe start w32time
```

5. Close the **Command Prompt** window.

How it works…

The first of the preceding commands instructs the Windows Time Service to synchronize time with the following sources:

- `europe.pool.ntp.org`
- `time.nist.gov`
- `192.43.244.18`
- `193.67.79.202`

A mix of both DNS names and IP addresses is recommended for the `/manualpeerlist` parameter to avoid time synchronization problems when DNS problems occur within the environment. Also, it hardens time synchronization from DNS poison attacks. However, the IP addresses should be checked regularly to ensure they're still time sources.

The second command stops and then starts the Windows Time Service on the domain controller to make the settings take effect.

Microsoft's recommendation is to have the domain controller holding the PDC Emulator FSMO role in the forest root domain synchronize its internal clock with an external time source, so other domain controllers can synchronize their clocks with its clock. The domain controller holding the PDC Emulator FSMO role in additional domains throughout the forest can synchronize their clocks to the domain controllers in the forest root domain, so that eventually, all servers, networking appliances, and client devices in those domains can synchronize their clocks.

Under normal circumstances, time synchronization is not terribly important for Active Directory. For Kerberos authentications, a time difference of up to 5 minutes is acceptable. However, when domain controllers handle conflicts for multiple changes to the same object, the timestamp of the last change determines the state of the object.

Managing time synchronization for virtual domain controllers

Virtual domain controllers consist of Windows Server installations running as virtual machines on hypervisor platforms. The following two hypervisor platforms dominate the server virtualization scene today:

- VMware vSphere
- Microsoft Hyper-V

Getting ready

This recipe describes two methods to manage time synchronization for virtual domain controllers:

- For virtual domain controllers running on Hyper-V, make the hardware clock of the virtualization host work together with the clock of the virtual domain controller.
- For virtual domain controllers running on vSphere, make the clock of the virtual domain controller work independently from the hardware clock of the virtualization host.

How to do it...

Choose between the two methods:

- Managing time synchronization for virtual domain controllers running on VMware vSphere
- Managing time synchronization for virtual domain controllers running on Microsoft Hyper-V

Managing time synchronization for virtual domain controllers running on VMware vSphere

For virtual domain controllers running on VMware vSphere, change the following 8 lines for the virtual machines in the **Advanced Configuration Options**:

```
tools.syncTime = "0"
 time.synchronize.continue = "0"
 time.synchronize.restore = "0"
 time.synchronize.resume.disk = "0"
 time.synchronize.shrink = "0"
 time.synchronize.tools.startup = "0"
 time.synchronize.tools.enable = "0"
 time.synchronize.resume.host = "0"
```

To add these settings across multiple virtual machines at once, use VMware vRealize Orchestrator.

Managing time synchronization for virtual domain controllers running on Microsoft Hyper-V

For virtual domain controllers running on Microsoft Hyper-V, take care of the following:

- Ensure that the virtual domain controllers synchronize their clocks with the internal clocks of the Hyper-V hosts.
- Ensure that hardware clocks on all Hyper-V hosts are correct.
- Configure reliable NTP sources on all Hyper-V hosts or join the Hyper-V hosts to an Active Directory forest, where the domain controller holding the PDC Emulator FSMO role correctly synchronizes time with a reliable source.

How it works…

For both hypervisor platforms, it is important to have the integration components or VMware Tools installed on the virtual domain controllers.

It's tempting to think that disabling the **Synchronize guest time with host** option in the integration components or VMware Tools keeps virtual domain controllers from synchronizing time with the virtualization host on which they run. Even if the option is disabled, a virtual domain controller will synchronize its time when you perform the following actions:

- Suspend it, the next time you resume it.
- Migrate the virtual domain controller using vMotion or live migration.
- Take a snapshot.
- Restore to a snapshot.
- Shrink the virtual disk.
- Restart the VMware Tools service.
- Reboot the virtual domain controller.

When the virtual domain controller happens to run on a virtualization host with incorrect time settings, the domain controller picks up the wrong time and will share this wrong time with other servers, networking appliances, and client devices. When users rely on these servers or devices to access resources that have the correct time, the authentication to these resources might fail when the time difference exceeds 5 minutes.

Managing global catalogs

Domain controllers with the additional global catalog role hold partial information on the most requested attributes for objects in Active Directory. With multiple global catalogs, the information is replicated between the global catalogs throughout the forest.

There are two tasks concerning global catalog servers:

- Enable a domain controller as a global catalog.
- Disable a domain controller as a global catalog.

Getting ready

Before enabling or disabling domain controllers as global catalog servers, the placement rules should be considered.

Global catalogs are of importance in environments with multiple domains. Except for Active Directory environments in which the Active Directory Recycle Bin is enabled, the placement of global catalogs should be considered in close relation to the domain controller holding the Infrastructure Master FSMO role.

In environments where we consider the placement of global catalogs, either of the following should be true:

- All domain controllers should be configured to be global catalogs.
- The Infrastructure Master FSMO role should be hosted on a domain controller that is not configured as a global catalog.

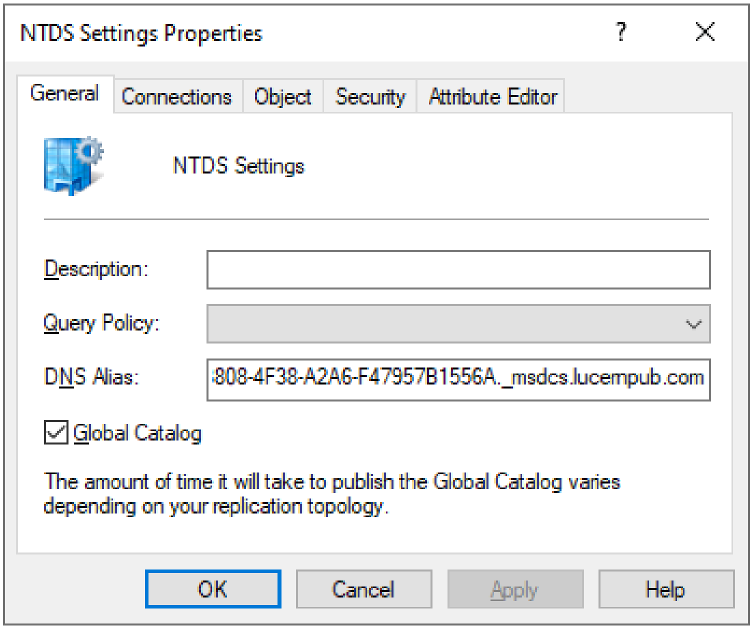
In environments with Microsoft Exchange Server, sufficient global catalog servers should be configured to handle Exchange Server address book lookups, both in the domain where the Exchange Organization lives, and in the domain where user accounts reside that access the services of the Exchange Server implementation.

To manage global catalogs, you'll need to be signed in with an account that is a member of the **Domain Admins** group.

How to do it...

Perform these steps to enable a domain controller to be configured as a global catalog server:

1. Press Start.
2. Search for **Active Directory Sites and Services** and click its search result or run `dssite.msc`. The **Active Directory Sites and Services** window appears.
3. In the left navigation pane, expand **Sites**.
4. Expand the Active Directory site where the domain controller resides.
5. Expand the **Servers** node.
6. Expand the domain controller you want to enable as a global catalog.
7. Right-click the **NTDS Settings** node underneath the domain controller.
8. Select **Properties**.
9. On the **General** tab, check the option next to **Global Catalog**:



Description:

Query Policy:

DNS Alias: 808-4F38-A2A6-F47957B1556A._msdcs.lucernpub.com

☑ Global Catalog

The amount of time it will take to publish the Global Catalog varies depending on your replication topology.

Figure 3.4 – The Global Catalog option on the General tab of NTDS Settings Properties

10. Click **OK**.
11. Close **Active Directory Sites and Services**.

In environments with Microsoft Exchange Server, reboot the domain controller after enabling it as a global catalog. Otherwise, the global catalog will not be functioning from the point of view of the Microsoft Exchange Server.

Perform these steps to disable a domain controller to be configured as a global catalog server:

1. Press Start.
2. Search for **Active Directory Sites and Services** and click its search result or run `dssite.msc`. The **Active Directory Sites and Services** window appears.
3. In the left navigation pane, expand **Sites.**
4. Expand the Active Directory site where the domain controller resides.
5. Expand the **Servers** node.
6. Expand the domain controller you want to disable as a global catalog.
7. Right-click the **NTDS Settings** node underneath the domain controller.
8. Select **Properties**.
9. On the **General** tab, uncheck the option next to **Global Catalog**.
10. Click **OK**.
11. Close **Active Directory Sites and Services**.

How it works

Global catalog servers are used when you do the following:

- Perform forest-wide searches.
- Perform Exchange Server address book lookups.
- Sign in to Active Directory forests that have multiple domains.

In these cases, the global catalog server responds to requests using TCP port `3268` (insecure) and TCP port `3269` (secure).

The preceding steps enable or disable a domain controller to be a global catalog server and, thus, support the preceding services.

4
Managing Containers and Organizational Units

A default Active Directory domain comes with default **containers** and a default **Organizational Unit** (**OU**). The default containers serve different purposes. The **Computers** and **Users** containers act as default locations when you create these objects. Other containers contain important objects, for example, the **Builtin** container. This container contains all default administrative accounts and groups. The default **Domain Controllers** OU serves as the location for domain controller objects.

The following containers are available by default:

- **Builtin**
- **Computers**
- **ForeignSecurityPrincipals**
- **Keys**
- **LostAndFound**
- **Managed Service Accounts**

- **NTDS Quotas**
- **Program Data**
- **System**
- **TPM devices**
- **Users**

> Tip
>
> In **Active Directory Users and Computers** (`dsa.msc`), some of these default containers are hidden. You can display them by selecting the **Advanced Features** option in the **View** menu.

We can review the default structure by using either the **Active Directory Administrative Center** (`dsac.exe`), **Active Directory Users and Computers** (`dsa.msc`), or Windows PowerShell. Type the following line of PowerShell on a domain-joined device with the Active Directory module for Windows PowerShell installed or on a domain controller to get a list of the OUs in the Active Directory domain that you've signed into:

```
Get-ADOrganizationalUnit -filter * | ft name
```

In Active Directory, containers and OUs are different from user and computer objects – containers and OUs don't have **Security Identifiers** (**SIDs**), so they can't be used as security principals to directly grant access to resources.

The configuration partition of the Active Directory database can get messy over the life cycle of the domain. As complexity is the archenemy of every admin, there needs to be a way of organizing or even segregating objects.

This chapter serves up the following recipes:

- Creating an OU
- Deleting an OU
- Modifying an OU
- Delegating control of an OU
- Modifying the default location for new user and computer objects

Differences between OUs and containers

Before we get into the recipes, it's important for us to understand the differences between OUs and containers. OUs and containers play different roles and act differently in Active Directory.

Containers

Containers are created by default. Creating or deleting containers using the built-in tools is not supported. Containers don't support delegation or **Group Policy** either. Creating an OU is not possible in a container.

OUs

OUs can be created and deleted by Active Directory admins. They support delegation of control, using the **Delegation of Control Wizard** and the built-in tools. Group Policy objects and managers can be applied to OUs. OUs can be nested.

OUs versus Active Directory domains

One of the most heated discussions when setting up and/or extending Active Directory environments is to create OUs for departments, locations, and/or organizations. Please refer to the *Choosing between a new domain or forest* recipe in *Chapter 1*, *Optimizing Forests, Domains, and Trusts*, to make the best-informed decision.

Creating an OU

In this recipe, we create an OU.

Getting ready

To create an OU, sign in with an account that is a member of the **Domain Admins** group or an account that has delegated privileges to create OUs.

See the *Delegating control of an OU* recipe in this chapter for more information on delegating control of an OU.

How to do it…

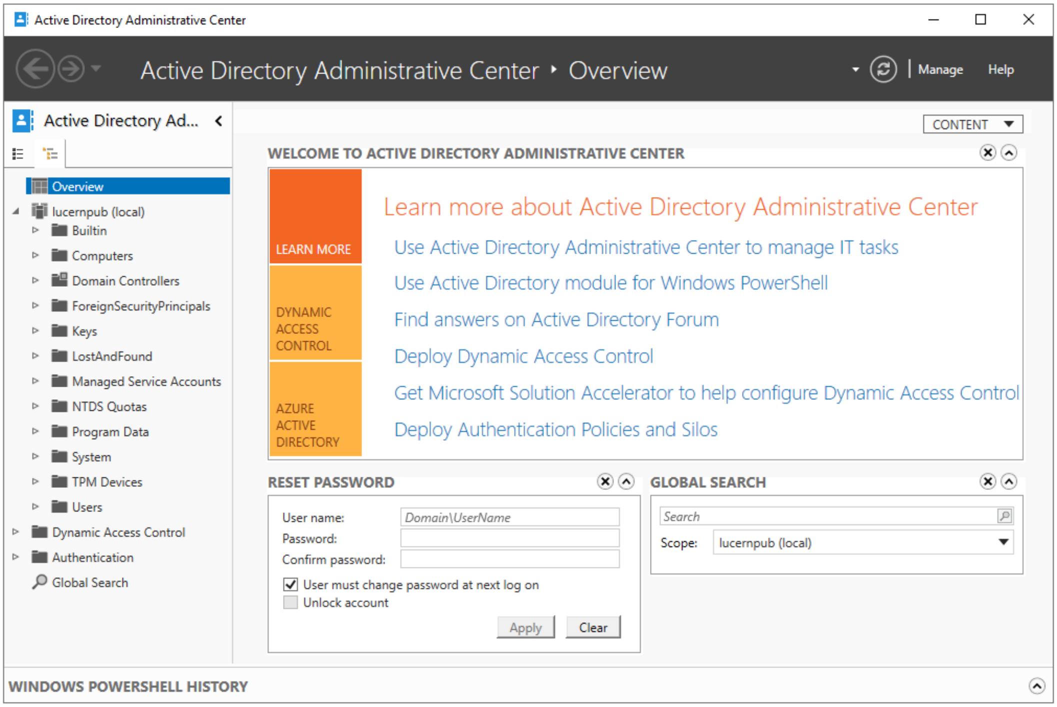
This recipe shows three ways to create an OU:

- Using the **Active Directory Administrative Center**
- Using the command line
- Using Windows PowerShell

Using the Active Directory Administrative Center

To create an OU using the **Active Directory Administrative Center**, perform the following steps:

1. Press Start.
2. Search for **Active Directory Administrative Center** and click its search result or run `dsac.exe`. The **Active Directory Administrative Center** window appears.
3. In the left navigation pane, switch to the tree view.



Learn more about Active Directory Administrative Center

Use Active Directory Administrative Center to manage IT tasks

Use Active Directory module for Windows PowerShell

Find answers on Active Directory Forum

Deploy Dynamic Access Control

Get Microsoft Solution Accelerator to help configure Dynamic Access Control

Deploy Authentication Policies and Silos

User name: Domain\UserName

Password:

Confirm password:

☑ User must change password at next log on

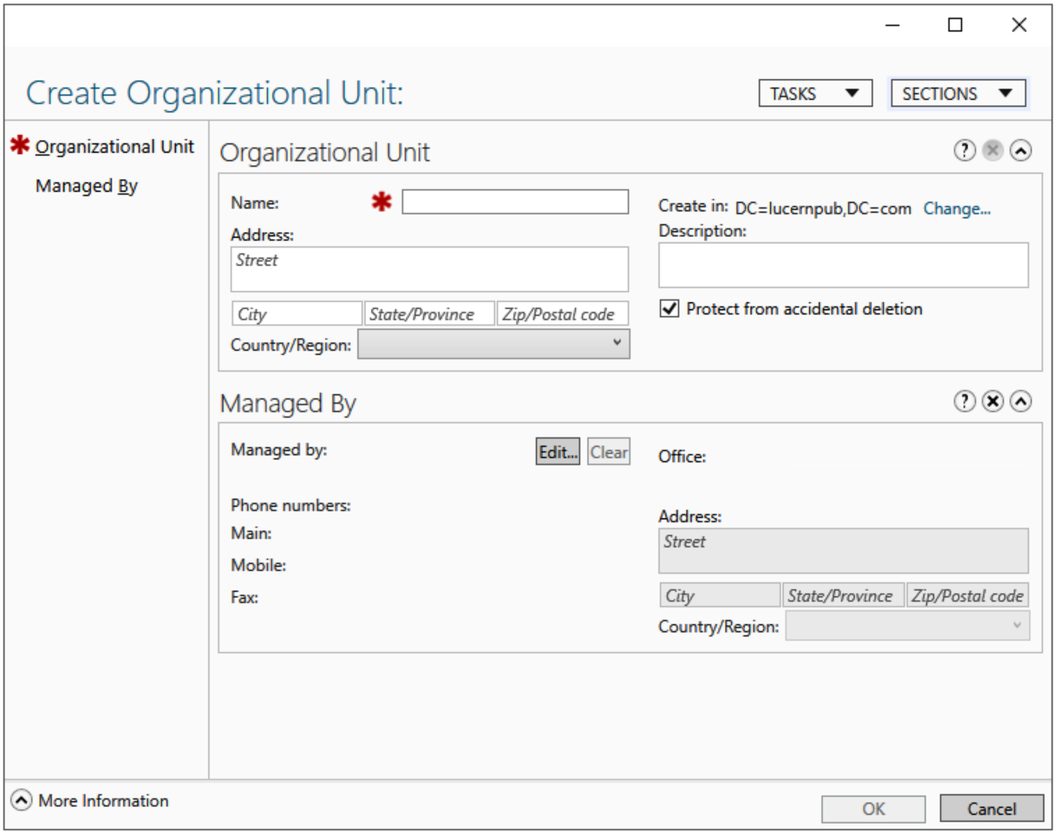
□ Unlock account

Search

Scope: lucernpub (local)

Figure 4.1 – Tree view in the Active Directory Administrative Center

4. If necessary, expand the tree to locate the domain or the OU you want to be the parent object.
5. Select the parent object for the new OU.
6. Right-click the parent object and select **New…**. From the menu, select **Organizational Unit**. The **Create Organizational Unit:** window appears:



Create in: DC=lucernpub,DC=com

☑ Protect from accidental deletion

Figure 4.2 – The Create Organizational Unit: window

7. Enter a name for the OU. You can also enter a description or provide information for any of the other optional fields.
8. Click **OK** to create the OU.

Using the command line

Since Windows 2000 Server, `dsadd.exe`, a command-line tool, has been available to create objects, including OUs.

Run the following command line on a domain controller or any domain-joined Windows or Windows Server installation with the **Remote Server Administration Tools** (**RSAT**) for Active Directory Domain Services installed:

```
dsadd.exe ou "OU=Finance,DC=LucernPub,DC=com" -desc "Finance
OU"
```

Replace `"OU=Finance,DC=LucernPub,DC=com"` with the distinguished name of the OU you want to add and `"Finance OU"` with the optional description that you want to use.

Using Windows PowerShell

Use the following line of PowerShell to create an OU on a system with the Active Directory module for Windows PowerShell installed:

```
New-ADOrganizationalUnit "Finance" -Description "Finance OU"
```

The `New-ADOrganizationalUnit` PowerShell cmdlet is the most elaborate option to create OUs on Windows Server installations without the desktop experience and through the automation of administrative tasks.

How it works...

This recipe showed how to use the **Active Directory Administrative Center**, command line, and/or Windows PowerShell to create OUs. When you create them using the aforementioned ways, OUs are automatically protected from accidental deletion.

Creating OUs using the **Active Directory Administrative Center** allows for a greater overview of the creation process, but it is not very scalable. You wouldn't use it to create 1,000 OUs, for instance. The command line and Windows PowerShell are more suited for these scenarios.

The PowerShell History Viewer feature in the **Active Directory Administrative Center** might help you discover the Windows PowerShell cmdlets to get the latter process going.

When you create an OU, the **Name** attribute is the only attribute you're required to specify.

There's more...

If you use the PowerShell example from this recipe, you can optionally use the `-ProtectedFromAccidentalDeletion $false` parameter, if you do not want the OU to be protected from accidental deletion. This is useful for test scenarios.

Deleting an OU

Use this recipe to delete an OU.

Getting ready

To delete an OU, sign in with an account that is a member of the **Domain Admins** group or has delegated privileges to delete OUs.

OUs are protected from accidental deletion by default. When you try to delete an OU that is protected from accidental deletion, you'll get an **Access denied** error. In this case, uncheck the **Protected from Accidental Deletion** checkbox from the OU's properties.

> **Note**
>
> Unless you create OUs using `csvde.exe` or `ldifde.exe`, they are protected from accidental deletion by default.

How to do it...

This recipe shows three ways to delete an OU:

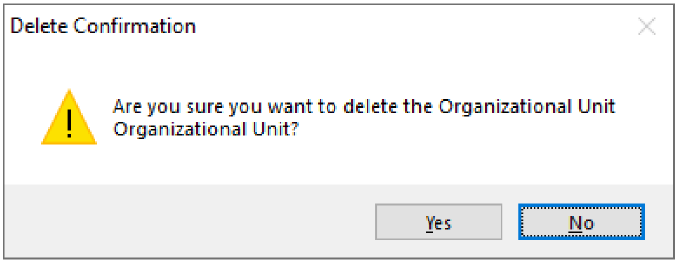
- Using the **Active Directory Administrative Center**
- Using the command line
- Using Windows PowerShell

Using the Active Directory Administrative Center

To delete an OU using the **Active Directory Administrative Center**, perform the following steps:

1. Press Start.
2. Search for **Active Directory Administrative Center** and click its search result or run `dsac.exe`. The **Active Directory Administrative Center** window appears.
3. In the left navigation pane, switch to the tree view.

4. If necessary, expand the tree to locate the OU you want to delete.
5. Select the OU.
6. Right-click the OU and click **Properties**.
7. Uncheck the **Protected from Accidental Deletion** checkbox.
8. Click **OK**.
9. Right-click the OU again. This time, click **Delete**. The **Delete Confirmation** pop-up window appears:



Delete Confirmation

Are you sure you want to delete the Organizational Unit Organizational Unit?

Figure 4.3 – The Delete Confirmation pop-up window

10. Click **Yes** to confirm.
11. If the OU contains child objects, click **Yes** again.

Using the command line

Since Windows 2000 Server, a command-line tool has been available to delete objects, including OUs: `dsrm.exe`.

Run the following command line on a domain controller or any domain-joined Windows or Windows Server installation with the RSAT for Active Directory Domain Services installed:

```
dsrm.exe "OU=Finance,DC=LucernPub,DC=com" -subtree
```

Replace `"OU=Finance,DC=LucernPub,DC=com"` with the distinguished name of the OU you want to delete.

Using Windows PowerShell

Use the following line of PowerShell on a system with the Active Directory module for Windows PowerShell installed:

```
Remove-ADObject -Identity "OU=Finance,DC=LucernPub,DC=com"-
Recursive -Confirm:$False
```

The `Remove-ADObject` PowerShell cmdlet is the most elaborate option to delete objects on Windows Server installations without the desktop experience and through automation of administrative tasks.

How it works...

As organizations change, so do the requirements on Active Directory. When an OU is no longer of use for many reasons, such as it no longer holds any objects, the linked GPOs no longer apply, or the level of delegation and/or management is no longer needed, you can delete it.

Deleting containers using the built-in tools is not supported.

There's more...

When the Active Directory Recycle Bin is enabled in Active Directory, you can safely restore OUs and their child objects, if you need to. See the *Enabling the Active Directory Recycle Bin* recipe in *Chapter 1*, *Optimizing Forests, Domains, and Trusts*, for more information.

Modifying an OU

After you've created an OU, you might find you need to change its properties.

Getting ready

To modify an OU, sign in with an account that is a member of the **Domain Admins** group or has delegated privileges to modify OUs.

How to do it...

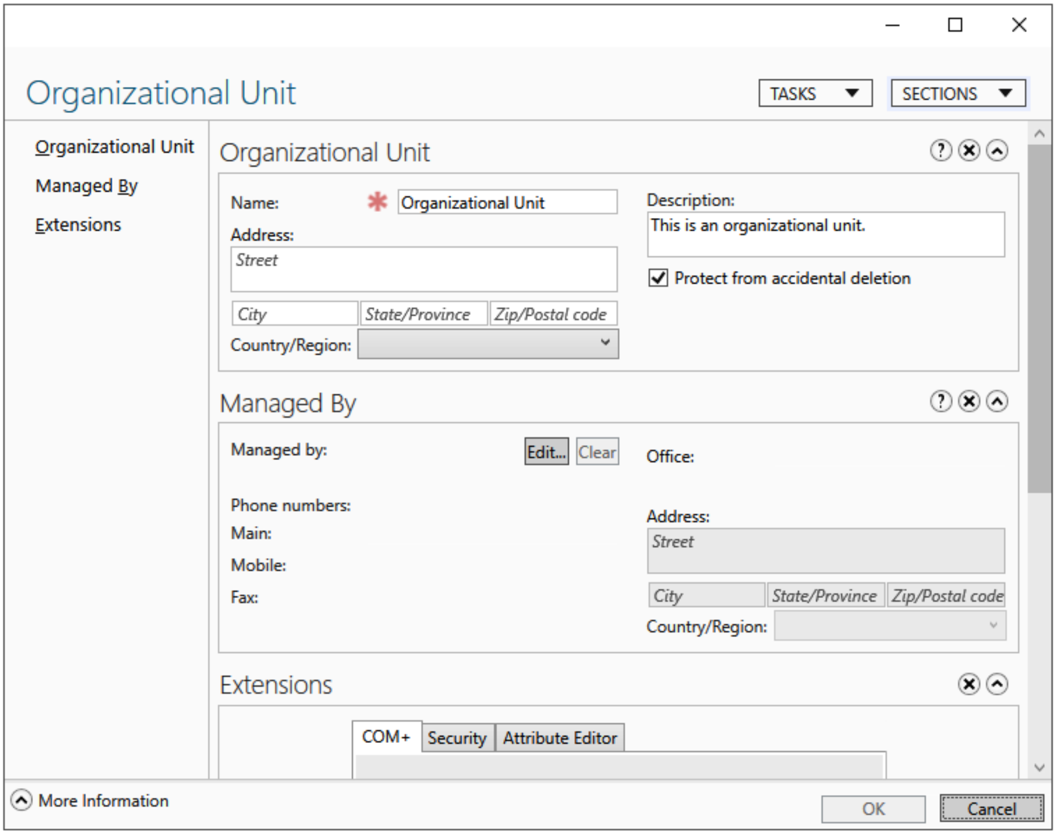
This recipe shows three ways to modify an OU:

- Using the **Active Directory Administrative Center**
- Using the command line
- Using Windows PowerShell

Using the Active Directory Administrative Center

To modify an OU using the **Active Directory Administrative Center**, perform the following steps:

1. Press Start.
2. Search for **Active Directory Administrative Center** and click its search result or run `dsac.exe`. The **Active Directory Administrative Center** window appears.
3. In the left navigation pane, switch to the tree view.
4. If necessary, expand the tree to locate the OU you want to modify.
5. Select the OU.
6. Right-click the OU and click **Properties**.



Organizational Unit

Name: * Organizational Unit

Description: This is an organizational unit.

☑ Protect from accidental deletion

Figure 4.4 – The properties of an OU

7. Change the properties you want to modify:

   I. Modify the **Description** field.

   II. Modify the **Managed by** field.

   III. Uncheck **Protect from accidental deletion**.

8. Click **OK**.

Using the command line

Since Windows 2000 Server, a command-line tool has been available to modify objects, including OUs: `dsmod.exe`. Unfortunately, for OUs, it can only be used to change the description.

Run the following command line on a domain controller or any domain-joined Windows or Windows Server installation with the RSAT for Active Directory Domain Services installed to change the description of an OU:

```
dsmod.exe ou "OU=Finance,DC=LucernPub,DC=com" -desc "New
description"
```

Replace `"OU=Finance,DC=LucernPub,DC=com"` with the distinguished name of the OU you want to modify and `"New description"` with the description that you want to use.

Using Windows PowerShell

You can modify the following properties with the `Set-ADOrganizationalUnit` PowerShell cmdlet on a system with the Active Directory module for Windows PowerShell installed:

- `City`
- `Country`
- `Description`
- `LinkedGroupPolicyObjects`
- `ManagedBy`
- `Name`
- `PostalCode`
- `State`
- `StreetAddress`

Simply use the following line of PowerShell on a system with the Active Directory module for Windows PowerShell installed and change the properties of the OU:

```
Set-ADOrganizationalUnit -Identity
"OU=Finance,DC=LucernPub,DC=com" -ManagedBy
"CN=User,CN=Users,DC=LucernPub,DC=com"
```

The `Set-ADOrganizationalUnit` PowerShell cmdlet is the most elaborate option to modify OUs on Windows Server installations without the desktop experience and through the automation of administrative tasks.

How it works...

Modifying the properties of an OU means modifying its attributes in Active Directory.

A typical OU has a number of attributes:

- `description`
- `gPLink`
- `gPOptions`
- `msDS-Approx-Immed-Subordinates`
- `managedBy`
- `ou`

You can change these attributes to modify the OU.

There's more...

In PowerShell, it's not possible to change the distinguished name for an OU, as this would effectively move the OU. The process of moving OUs is different and requires the `Move-ADObject` PowerShell cmdlet.

Protection from accidental deletion is not an attribute of an OU. Instead, the checkbox you see in the **Active Directory Administrative Center** manifests itself as security permissions on the OU and the parent object.

See also

Modifying the `gPLink` and `gPOptions` attributes is covered in the *Linking Group Policy objects to an organizational unit* recipe in *Chapter 10*, *Getting Most Out of Group Policy*.

Delegating control of an OU

In large Active Directory environments, administration may be challenging. Therefore, in environments with several teams of administrators and service desk personnel, delegation can be quite helpful. This way, just to name a few possibilities, service desk personnel may reset passwords, application administrators may change group memberships, and only true Active Directory admins may manage OUs.

Getting ready

To perform delegation of control, sign in with an account that is a member of the **Domain Admins** group or has **full control** privileges of the OU you want to delegate control over.

How to do it...

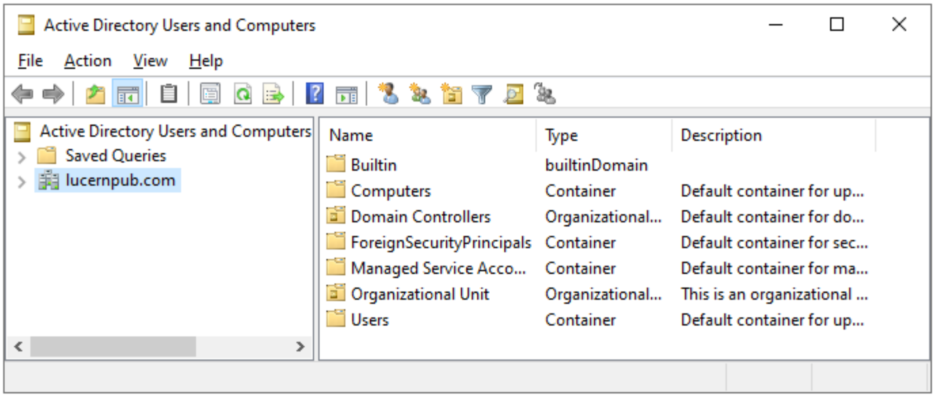
This recipe describes two ways to delegate control over an OU:

- Using **Active Directory Users and Computers**
- Using the command line

Using Active Directory Users and Computers

By far the easiest way to perform delegation of control is using the **Delegation of Control Wizard** from **Active Directory Users and Computers** (`dsa.msc`). Perform the following steps:

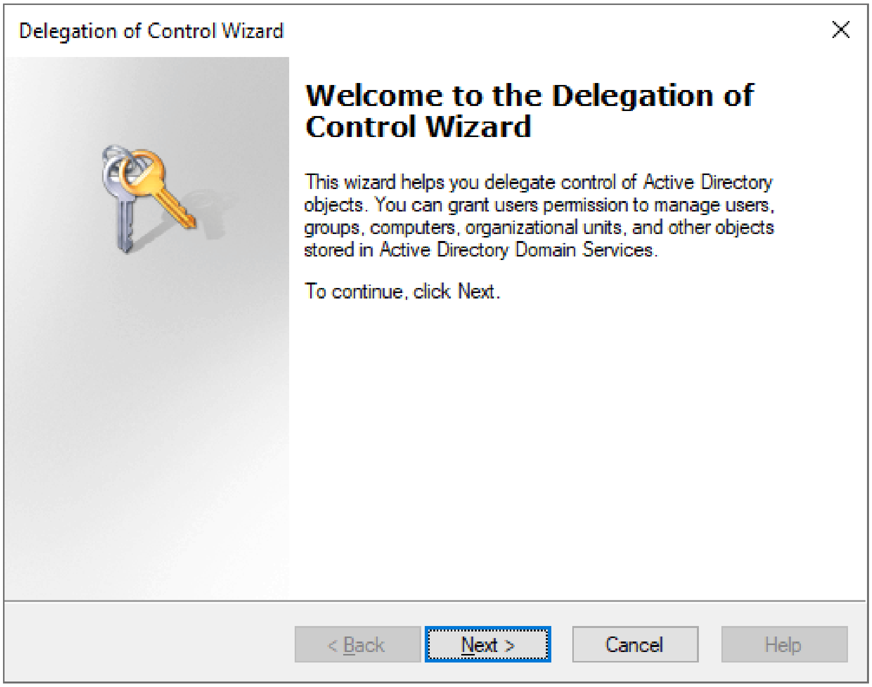
1. Press Start.
2. Search for **Active Directory Users and Computers** and click its search result or run `dsa.msc`. The **Active Directory Users and Computers** window appears.


Active Directory Users and Computers
File
Action
View
Help
Active Directory Users and Computers
Saved Queries
lucernpub.com

Name	Type	Description
Builtin	builtinDomain	
Computers	Container	Default container for up...
Domain Controllers	Organizational...	Default container for do...
ForeignSecurityPrincipals	Container	Default container for sec...
Managed Service Acco...	Container	Default container for ma...
Organizational Unit	Organizational...	This is an organizational ...
Users	Container	Default container for up...

Figure 4.5 – Active Directory Users and Computers

3. Right-click the OU you want to delegate control over to and select **Delegate control…** from the menu. The Delegation of Control Wizard appears:



Delegation of Control Wizard

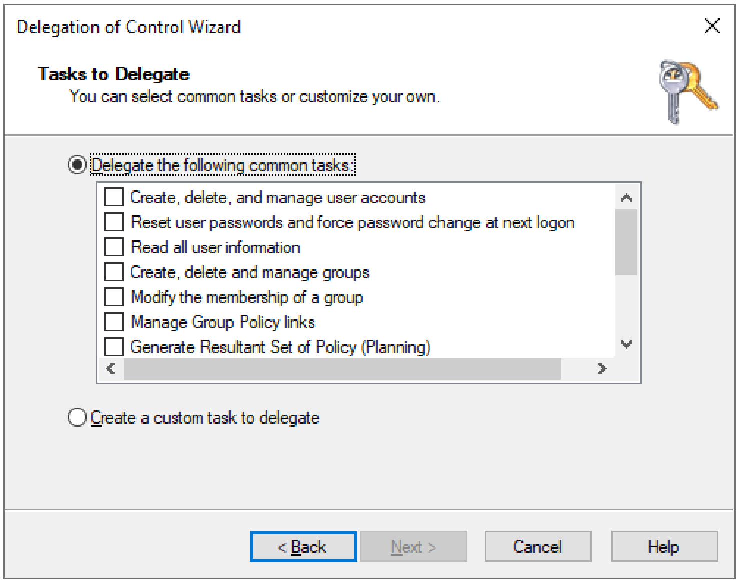
Welcome to the Delegation of Control Wizard

This wizard helps you delegate control of Active Directory objects. You can grant users permission to manage users, groups, computers, organizational units, and other objects stored in Active Directory Domain Services.

To continue, click Next.

Figure 4.6 – The Delegation of Control Wizard

4. On the **Welcome to the Delegation of Control Wizard** screen, click **Next** >.
5. On the **Users or Group** screen, click the **Add…** button to select one or more users or groups to whom you want to delegate control.
6. In the **Select Users, Computers, or Groups** pop-up, select the groups you want to delegate to.
7. Click **OK** when done.
8. On the **Users or Group** screen, click **Next** > to continue to the next screen of the Delegation of Control Wizard.



Delegation of Control Wizard

Tasks to Delegate

You can select common tasks or customize your own.

● Delegate the following common tasks:

- □ Create, delete, and manage user accounts
- □ Reset user passwords and force password change at next logon
- □ Read all user information
- □ Create, delete and manage groups
- □ Modify the membership of a group
- □ Manage Group Policy links
- □ Generate Resultant Set of Policy (Planning)

○ Create a custom task to delegate

Figure 4.7 – The Tasks to Delegate screen of the Delegation of Control Wizard

9. On the **Tasks to Delegate** screen, select the common task(s) you want to delegate from the list below **Delegate the following common tasks** or select the **Create a custom task to delegate** radio button to create a custom delegation task.

> **Note**
>
> If you select **Create a custom task to delegate**, select the object(s) and action(s) on the **Active Directory Object Type** screen and the permissions on the **Permissions** screen.

10. Click **Next** >.
11. On the **Completing the Delegation of Control Wizard** screen, click **Finish**.

The resulting delegation permissions (**Access Control Entries**, or **ACEs**) can now be viewed on the **Security** tab of the properties of the OU.

Using the command line

Using `dsacls.exe`, you can delegate permissions from the command line. This is ideal for scripted deployments of predetermined permissions and for applying permissions on domain controllers running as Server Core installations.

In its most basic usage, `dsacls.exe` can be used to display permissions (using `/a`), deny permissions (using the `/d` parameter), and grant permissions (using `/g`). These permissions are set to an object, and denoted as a distinguished name for a group (or user, if you must). The permissions themselves take the form of generic permissions or specific permissions, which are denoted by two letters, the so-called permission bits. The generic permissions are denoted in the table that follows:

Permission bit	**Permission**
`GR`	Generic read
`GE`	Generic execute
`GW`	Generic write
`GA`	Generic all

Table 4.1 – Permission bits and their permissions

The most popular specific permissions are as follows:

Permission bit	**Permission**
`SD`	Delete an object
`DT`	Delete an object and all child objects
`RC`	Read security information
`WD`	Change security information
`WO`	Change owner information
`CC`	Create a child object
`DC`	Delete a child object
`RP`	Read a property
`WP`	Write to a property

Table 4.2 – Popular permissions

When you use the last four specific permissions (`CC`, `DC`, `RP`, and `WP`), it's a recommended practice to also include the object type or attribute for which you want the permission to apply.

For example, use the following command line to delegate write permissions to a group for the `mS-DS-ConsistencyGUID` attribute for user objects:

```
dsacls.exe "OU=Organizational Unit,DC=LucernPub,DC=com" /I:S /G
"LucernPub\Group:RPWP;"mS-DS-ConsistencyGUID";user"
```

Replace `LucernPub\Group` with the name of the group you want to delegate control to and `"OU=Organizational Unit,DC=LucernPub,DC=com"` with the distinguished name of the OU you want to delegate control over.

In environments with Azure AD Connect, the `mS-DS-ConsistencyGUID` attribute is written to by Azure AD Connect, by default, to use as a source anchor attribute.

How it works...

This recipe describes two ways to delegate:

- Using built-in groups
- Using delegation of control

Using built-in groups

Using built-in groups, such as **Account Operators** and **Server Operators**, is an easy and fast way to delegate administrative tasks in Active Directory. However, there are a number of things you need to be aware of:

- The built-in **Account Operators** group provides more permissions than are actually required in many organizations. While you might expect the members of this group to merely have permissions to reset passwords of non-admins, they can create, modify, and delete all objects in all OUs except **Domain Controllers**. Another exception is that they cannot modify group memberships for the **Domain Admins** group. However, they have permissions to interactively sign in to domain controllers and have permissions to shut them down, by default.
- The built-in **Server Operators** group grants permissions to interactively sign in to domain controllers. This might pose an unexpected security risk.

Using delegation of control

Built-in groups offer broadly delegated permissions. As an alternative to using built-in groups, you can granularly delegate permissions per OU. There are a couple of recommended practices to keep you and your colleagues from insanity:

- Build a delegation of control model and/or authorization matrix before performing delegation of control. This way, delegation settings can be continually documented, agreed upon, and transferred to other admins without adding unnecessary complexity.
- Always use groups when delegating permissions, not individual user or computer accounts. This way, giving permissions is a matter of (temporarily) adding a user account to a group, instead of using the Delegation of Control Wizard each time. It also makes auditing *that* much easier.
- Try to avoid denying permissions to avoid complexity. Denied permissions take precedence over allowed and/or granted permissions.
- Use a hacker mindset. Always test the delegation settings for any unwanted effects.
- Use delegation of control of groups in combination with NTDS Quotas to prevent group administrators from creating over 1,000 groups, adding members to these groups, and performing a denial-of-service attack, because user accounts can't be used to sign in when they have over 1,010 group memberships.

See also

If you need help creating OUs, take a look at the *Creating an OU* recipe.

Modifying the default location for new user and computer objects

When you join a device to the Active Directory domain or create a user object without context, these objects will be placed in default containers. Devices end up in the **Computers** container and user objects end up in the **Users** container. You can change these locations to accommodate for processes, delegation, and Group Policy structure – when a computer object is placed in an OU other than the **Computers** container, it might get picked up by an endpoint management solution automatically, have proper settings deployed by Group Policy automatically, and be manageable by delegated service desk personnel automatically.

Getting ready

To modify the default location for new user objects and computer objects, the Active Directory environment needs to run the Windows Server 2003 **Domain Functional Level** (**DFL**), or higher. If you try to modify the location in an Active Directory environment running the Windows 2000 Server DFL, you will receive the following error:

```
Error, unable to modify the wellKnownObjects attribute. Verify
that the domain functional level of the domain is at least
Windows Server 2003:
Unwilling To Perform
Redirection was NOT successful.
```

Additionally, sign in interactively with a user object that is a member of the **Domain Admins** group on a domain controller. This action cannot be performed remotely from a management workstation or management server.

How to do it...

To modify the default location for new user and computer objects, we first need to create the OUs to which we want to redirect. I recommend creating two separate OUs to separate user objects from computer objects. `Redirected Users OU` and `Redirected Computers OU` would do nicely.

Then, in an elevated **Command Prompt** window, run the following commands:

```
redirusr.exe "OU=Redirected Users OU,DC=LucernPub,DC=com"
redircmp.exe "OU=Redirected Computers OU,DC=LucernPub,DC=com"
```

Replace the values for the OUs to redirect to match your environment.

If you ever need to revert the changes, run the following two commands:

```
redirusr.exe "CN=Users,DC=LucernPub,DC=com"
redircmp.exe "CN=Computers,DC=LucernPub,DC=com"
```

Replace `DC=LucernPub,DC=com` with the values that match your environment.

How it works...

Microsoft deliberately chose to put devices in the **Computers** container and user objects in the **Users** container because containers can't have policies applied to them. Even if an Active Directory administrator misconfigured the directory, you should at least be able to join a device to the domain and sign in to it without having Group Policy objects applied to it.

Active Directory admins can change the default locations for computer and user objects. Because this action has the potential to impact the directory, Microsoft has decided to make this change available only on the command line, through two specific executables, as follows:

Executable	**Purpose**
`Redircmp.exe`	Redirect computer objects to a different OU or container
`Redirusr.exe`	Redirect user objects to a different OU or container

Table 4.3 – Command lines to redirect objects

See also

If you need help creating OUs, take a look at the *Creating an OU* recipe.

5
Managing Active Directory Sites and Troubleshooting Replication

When I first learned about Active Directory sites, the concept was explained to me as being locations of readily available connectivity.

There's an easy analogy for it: islands. In island states, people live on islands, but not everything they need might be available on their island. Additionally, something on their island might break, and there are only a few trade routes for goods and services.

In this analogy, the trade routes between geographical locations are the networking connections between Active Directory sites, the islands of readily available connectivity. The island's roads are that readily available connectivity: you can use them all you want, without additional cost.

Not many organizations place the domain controllers that hold **Flexible Single Master Operation** (**FSMO**) roles in poorly connected Active Directory sites. Many networking topologies for organizations feature a hub-and-spoke layout with a central location and several outlying locations with, optionally, further outlying locations.

This chapter serves up the following recipes:

- Creating a site
- Managing a site
- Managing subnets
- Creating a site link
- Managing a site link
- Modifying replication schedules for an Active Directory site link
- Creating a site link bridge
- Managing bridgehead servers
- Managing the **Inter-site Topology Generator** (**ISTG**) and **Knowledge Consistency Checker** (**KCC**)
- Managing **Universal Group Membership Caching** (**UGMC**)
- Working with `repadmin.exe`
- Forcing replication
- Managing inbound and outbound replication
- Modifying the tombstone lifetime period
- Managing strict replication consistency
- Upgrading **System Volume** (**SYSVOL**) replication from **File Replication Service** (**FRS**) to **Distributed File System Replication** (**DFSR**)
- Checking for and remediating lingering objects

Before studying the recipes, we will look at a few points on Active Directory sites and recommendations.

What do Active Directory sites do?

Active Directory sites govern access and replication.

Active Directory's `DClocator` process allows for devices to find the nearest domain controllers. By default, these would be the domain controllers in the current Active Directory site where the device resides. The way the device knows in which site it resides is derived from its **Internet Protocol** (**IP**) address, which matches a subnet, as defined for the Active Directory site.

When there are no domain controllers in an Active Directory site, the site link costs define the nearest domain controllers to `DClocator`. The domain controllers in the site connected with the lowest cost will be returned to the device.

Domain controllers in different Active Directory sites replicate partitions over the same Active Directory site links through bridgehead servers; these are domain controllers that take on the additional role of replicating over the site link to the bridgehead server on the other side, on top of replicating with the domain controllers within its site.

Replication over site links can be managed in terms of schedule and replication type.

The **Domain Name System** (**DNS**) and the **Distributed File System** (**DFS**) are both Active Directory site-aware. They can provide access to services closest to the end user. For instance, in a DFS setup with a file server on each location, the end user in a specific site would be redirected to the file server in their respective Active Directory site and have readily available connectivity, while the file servers take care of any replication needed.

Recommendations

It is a recommended practice to design Active Directory sites following these rules of thumb:

- Create one Active Directory site per location. If the bandwidth between locations is above 10 megabits and reliable, and you don't want to segment services or subnets, create one Active Directory site for these locations.
- Configure one Active Directory site link between two Active Directory sites.
- Configure a catch-all subnet (for instance, a `10.0.0.0/8` subnet) in your main location and create subnets with smaller ranges (for instance, `10.1.0.0/16` and `10.3.1.0/24` subnets) for other locations.
- Do not disable the **Bridge all site links** option for all IP-based site links and all **Simple Mail Transfer Protocol** (**SMTP**)-based site links.
- Do not enable the **Ignore schedules** option for all IP-based site links.

- Keep the ISTG enabled.
- Keep the KCC enabled.
- Keep **Strict Replication Consistency** enabled.
- Define a process where networking administrators communicate changes in their environment to Active Directory admins so that they can optimize Active Directory to take advantage of these changes.
- Do not link **Group Policy Objects** (**GPOs**) to Active Directory sites, if you can avoid it.

Let's now begin the recipes for this chapter.

Creating a site

An Active Directory site definition tells Active Directory to treat the subnets, domain controllers, member servers, and domain-joined devices as well connected. Its Active Directory site link defines the connectivity to other Active Directory sites.

To define connectivity, two sites are needed. The `Default-First-Site-Name` site is present in Active Directory, by default. As Active Directory admins add Active Directory sites, the relationships between the sites evolve to align the logical Active Directory topology with the physical topology of even the most complex environments.

Getting ready

To create an Active Directory site, you'll need to be signed in with an account that is a member of the **Enterprise Admins** group. That's because Active Directory sites are defined for the Active Directory forest in the `CN=Configuration` partition.

How to do it...

This recipe describes two ways to create an Active Directory site:

- Using **Active Directory Sites and Services**
- Using Windows PowerShell

Using Active Directory Sites and Services

To create an Active Directory site using **Active Directory Sites and Services**, perform these steps:

1. Press Start.
2. Search for **Active Directory Sites and Services** and select it from the search results or run `dssite.msc`. The **Active Directory Sites and Services** window appears:

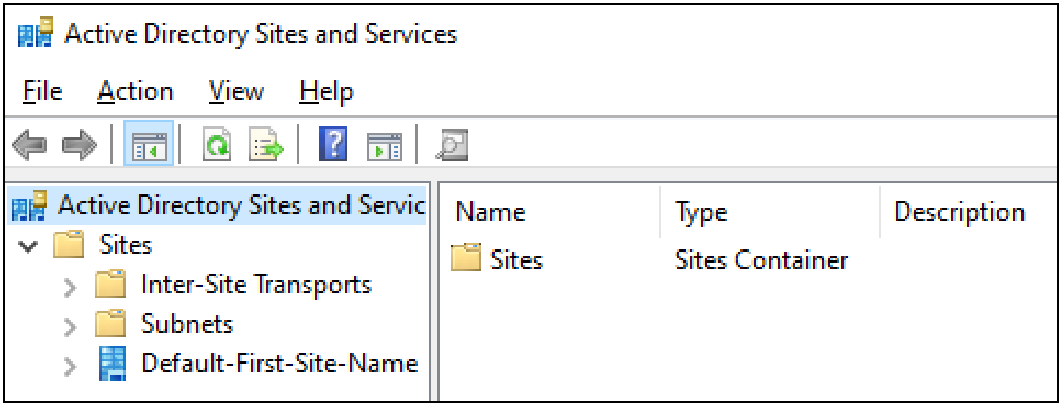
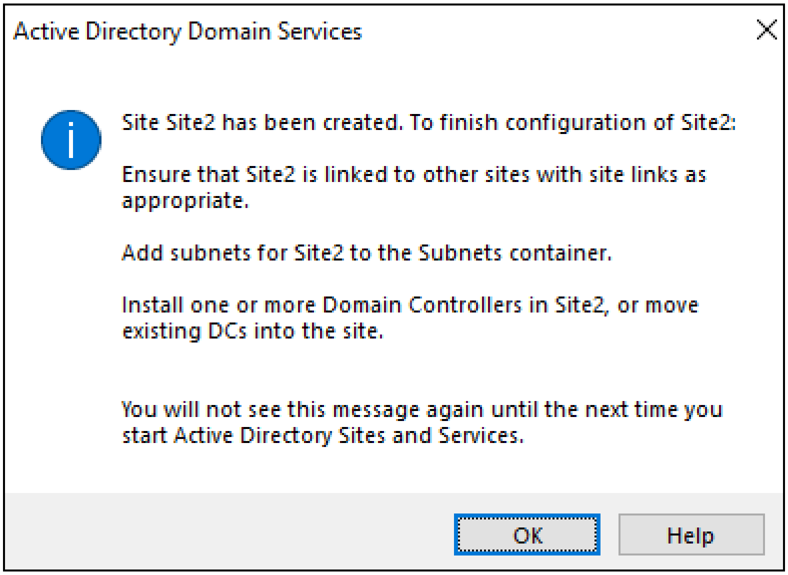


Figure 5.1 – Active Directory Sites and Services

3. In the left navigation pane, select the `Sites` node.
4. Right-click the **Sites** node and click **New Site…**.
5. The **New Object - Site** window appears.
6. Type a logical name for the Active Directory site in the **Name:** field.
7. If this is the first site you're creating, next to `Default-First-Site-Name`, select the `DEFAULTIPSITELINK` link.

8. Click the **OK** button to create an Active Directory site. The **Active Directory Domain Services** pop-up window appears:



Active Directory Domain Services

Site Site2 has been created. To finish configuration of Site2:

Ensure that Site2 is linked to other sites with site links as appropriate.

Add subnets for Site2 to the Subnets container.

Install one or more Domain Controllers in Site2, or move existing DCs into the site.

You will not see this message again until the next time you start Active Directory Sites and Services.

Figure 5.2 – Active Directory Domain Services pop-up window

9. Click **OK** in the **Active Directory Domain Services** pop-up window, which notifies you of additional actions to take to finish configuring the Active Directory site to create an Active Directory site, and close the pop-up window.

After you've performed the preceding steps, you can see the new Active Directory site in the left navigation pane, under the **Sites** node.

Using Windows PowerShell

To create an Active Directory site using the `Active Directory module` for Windows PowerShell, use the `New-ADReplicationSite` PowerShell cmdlet. Here's an example of the simplest line of PowerShell code to achieve this goal:

```
New-ADReplicationSite -Name "Site2"
```

You can verify your change by listing the Active Directory sites:

```
Get-ADReplicationSite -Filter * | Select-Object Name
```

Omitting the pipe in the last line of Windows PowerShell code provides more details on Active Directory sites, including their `Description`, `ManagedBy`, and `InterSiteTopologyGenerator` attributes.

See also

To make the right choice between creating an Active Directory site or an Active Directory domain, refer to the *Choosing between a new domain or forest* recipe in *Chapter 1*, *Optimizing Forests, Domains, and Trusts*.

To finish the configuration of the Active Directory site, refer to the following recipes:

- *Managing subnets*
- *Creating a site link*
- *Managing a site link*
- *Promoting a server to a domain controller* in *Chapter 2*, *Managing Domain Controllers*
- *Promoting a server to a read-only domain controller* in *Chapter 2*, *Managing Domain Controllers*

Managing a site

After creating an Active Directory site, you might need to manage it. You might want to do the following:

- Rename it.
- Change its description.
- Change its location.
- Delete it.
- Delegate control over it.

Getting ready

To manage an Active Directory site, you'll need to be signed in with an account that is a member of the **Enterprise Admins** group or with an account that has been delegated permissions to manage the Active Directory site.

If your aim is to delete an Active Directory site, remove any domain controller for the site first by right-clicking the domain controller object(s) under the site node and selecting **Move**. Alternatively, demote the domain controller.

How to do it...

The manageable properties of an Active Directory site are stored in its attributes. The best ways to change them are through **Active Directory Sites and Services** and the Active Directory module for Windows PowerShell.

Using Active Directory Sites and Services

To change the name of an Active Directory site, perform these steps:

1. Press Start.
2. Search for **Active Directory Sites and Services** and select it from the search results or run `dssite.msc`. The **Active Directory Sites and Services** window appears.
3. In the left navigation pane, expand the **Sites** node.
4. Select an Active Directory site you want to manage.
5. Right-click the site name and select **Rename** from the menu.
6. Type the new name for the Active Directory site and press *Enter* when done.

To change the description of an Active Directory site, perform these steps:

1. Press Start.
2. Search for **Active Directory Sites and Services** and select it from the search results or run `dssite.msc`. The **Active Directory Sites and Services** window appears.
3. In the left navigation pane, expand the **Sites** node.
4. Select an Active Directory site you want to manage.
5. Right-click the site name and select **Properties** from the menu.
6. On the **General** tab, type a description for the Active Directory site.
7. Press **OK** when done.

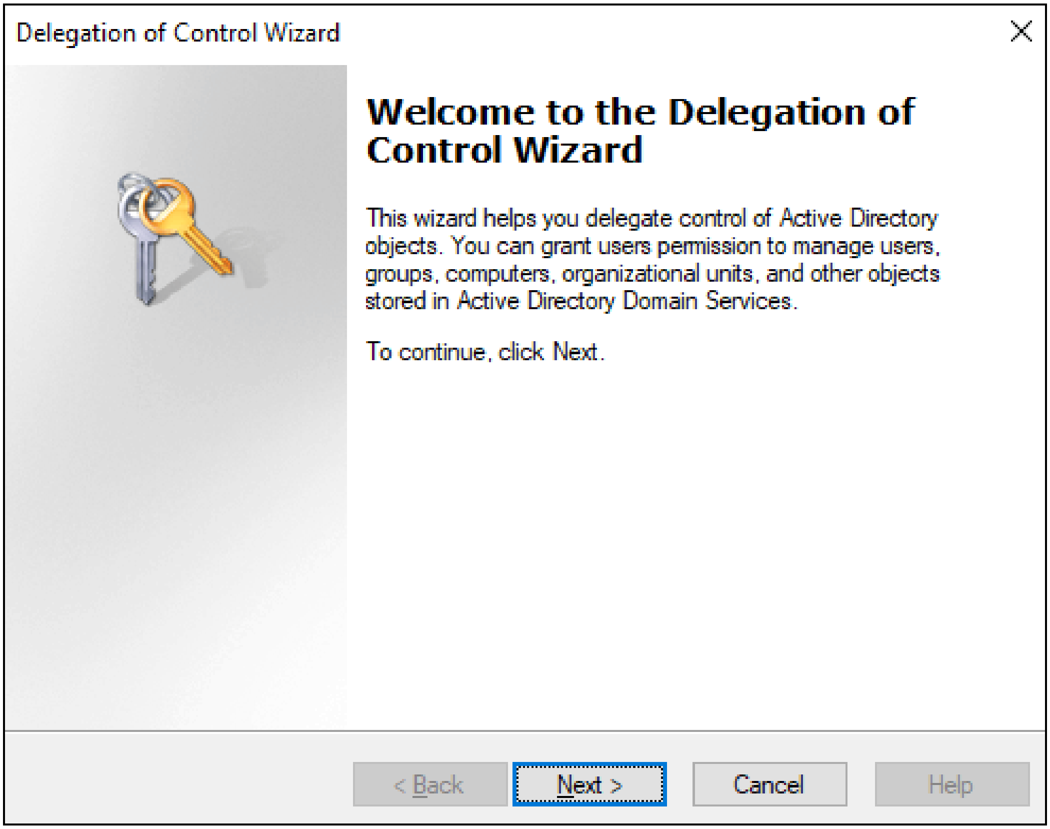
To change the location of an Active Directory site, perform these steps:

1. Press Start.
2. Search for **Active Directory Sites and Services** and select it from the search results or run `dssite.msc`. The **Active Directory Sites and Services** window appears.
3. In the left navigation pane, expand the **Sites** node.
4. Select an Active Directory site you want to manage.
5. Right-click the site name and select **Properties** from the menu.
6. Navigate to the **Location** tab.

7. On the **Location** tab, type a location for the Active Directory site.
8. Press **OK** when done.

To delegate the **Manage Group Policy links** permission of an Active Directory site, perform these steps:

1. Press Start.
2. Search for **Active Directory Sites and Services** and select it from the search results or run `dssite.msc`. The **Active Directory Sites and Services** window appears.
3. In the left navigation pane, expand the **Sites** node.
4. Select an Active Directory site you want to manage.
5. Right-click the site name and select **Delegate Control…** from the menu. The **Delegation of Control Wizard** window appears:



Delegation of Control Wizard

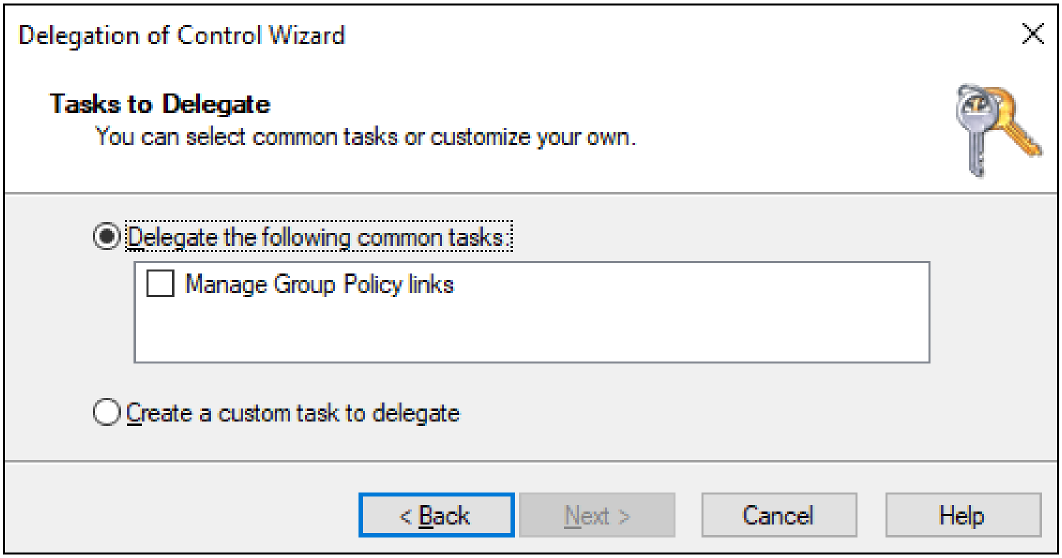
Welcome to the Delegation of Control Wizard

This wizard helps you delegate control of Active Directory objects. You can grant users permission to manage users, groups, computers, organizational units, and other objects stored in Active Directory Domain Services.

To continue, click Next.

Figure 5.3 – Delegation of Control Wizard

6. On the **Welcome to the Delegation of Control Wizard** screen, click **Next** >.
7. On the **Users or Groups** screen, click the **Add…** button and select the group(s) you want to delegate control for in the **Select Users, Computers or Groups** pop-up. Click **OK** when done. Back on the **Users or Groups** screen, click **Next** > to go to the **Tasks to Delegate** screen:



Delegation of Control Wizard

Tasks to Delegate

You can select common tasks or customize your own.

● Delegate the following common tasks:

□ Manage Group Policy links

○ Create a custom task to delegate

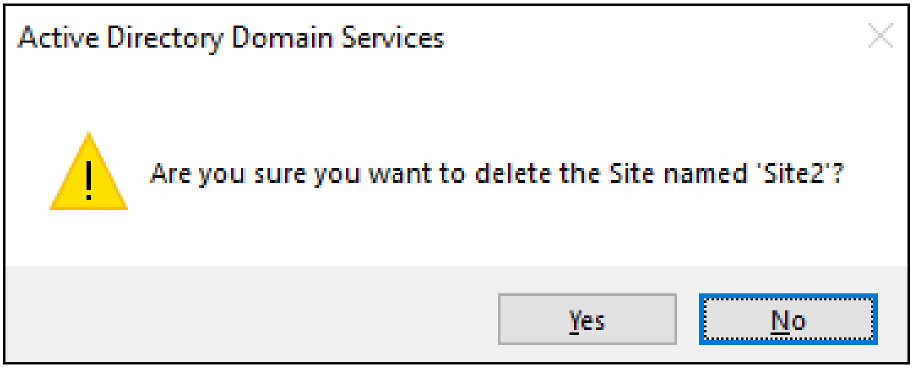
Figure 5.4 – Tasks to Delegate

8. On the **Tasks to Delegate** screen, select the **Manage Group Policy links** task underneath **Delegate the following common tasks:** and click **Next** >.
9. On the **Completing the Delegation of Control Wizard** screen, click **Finish**.

To delete an Active Directory site, perform these steps:

1. Press Start.
2. Search for **Active Directory Sites and Services** and select it from the search results or run `dssite.msc`. An **Active Directory Sites and Services** window appears.
3. In the left navigation pane, expand the **Sites** node.
4. Select an Active Directory site you want to delete.

5. Right-click the site name and select **Delete** from the menu. The **Active Directory Domain Services** pop-up window appears:



Active Directory Domain Services

Are you sure you want to delete the Site named 'Site2'?

Figure 5.5 – Active Directory Domain Services pop-up window

6. In the **Active Directory Domain Services** pop-up window, click **Yes** to answer the question **Are you sure you want to delete the Site named 'Site name'?**.

> **Note**
>
> Do not delete the `Default-First-Site-Name` Active Directory site. Instead, rename it.

Using Windows PowerShell

Use the following line of PowerShell code to rename an Active Directory site:

```
Rename-ADObject -Identity "CN=Default-First-Site-Name,CN=Sites,
CN=Configuration,DC=LucernPub,DC=com" -NewName "ADSite2"
```

Use the following line of PowerShell code to change the description of an Active Directory site:

```
Set-ADReplicationSite -Identity "CN=Default-First-Site-Name,CN
=Sites,CN=Configuration,DC=LucernPub,DC=com" -Description "New
description here"
```

Use the following line of PowerShell code to change the location of an Active Directory site:

```
Set-ADReplicationSite "CN=Default-First-Site-Name,CN=Sites,CN=C
onfiguration,DC=LucernPub,DC=com" -Location "New location here"
```

Windows PowerShell makes it easy to make bulk changes to Active Directory sites.

How it works...

When you rename an Active Directory site, you change the **Common Name** (**CN**) of the object of the Active Directory site. While this is not a problem with most software, such as **System Center Configuration Manager** (**SCCM**), your organization might get in trouble with software that references Active Directory sites by their names and not their **globally unique identifiers** (**GUIDs**).

An Active Directory site's location might be a field you'd want to use to define the physical location and/or network type of an Active Directory site. This way, when there is a problem with the Active Directory site, admins might make their way to the root cause—the core problem—and/or physical location faster. In multi-domain environments, it might be useful to denote the domain for which the Active Directory site is in use. A perfect example for such an environment using airfield **International Air Transport Association** (**IATA**) code would be `P ZRH LucernPub` to denote a production Active Directory site for *Lucern Publishing* near Zurich Airport.

Before deleting an Active Directory site, ensure to move all domain controllers from that site to a different site. Don't worry about site links or subnets; these are not deleted when you delete an Active Directory site.

When you delegate control of an Active Directory site, you allow a group to manage Active Directory site Group Policy links through the `gPLink` attribute of an Active Directory site, out of the box, or any more granular task on any of the attributes you want to delegate control over.

See also

To finish the configuration of the Active Directory site, refer to the following recipes:

- *Managing subnets*
- *Creating a site link*
- *Managing a site link*

Managing subnets

Subnets define logical network segments to Active Directory. Based on the subnet, a device contacts the domain controller(s) in the right site. There might be domain controllers in the same site as the device, but depending on the costs of the site links, devices might also look for domain controllers in other sites.

Getting ready

To manage subnets, you'll need to be signed in with an account that is a member of the **Enterprise Admins** group.

How to do it...

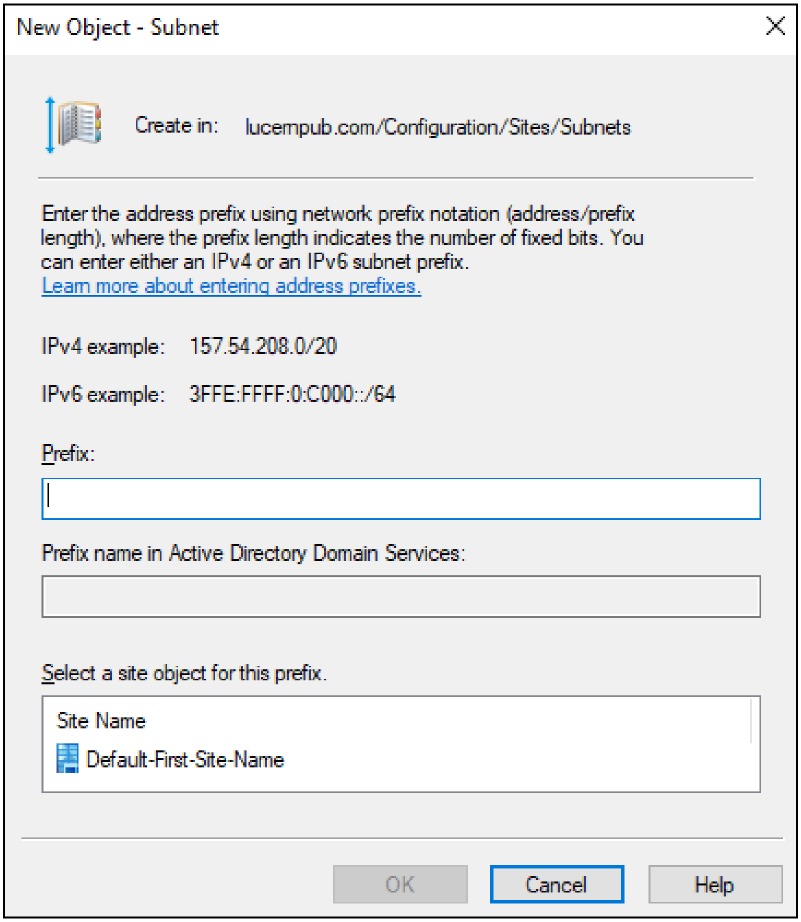
Subnets can be created and deleted using **Active Directory Sites and Services** and the Active Directory module for Windows PowerShell.

Using Active Directory Sites and Services

To create a subnet, perform these steps:

1. Press Start.
2. Search for **Active Directory Sites and Services** and select it from the search results or run `dssite.msc`. The **Active Directory Sites and Services** window appears.
3. In the left navigation pane, expand the **Sites** node.
4. Select the **Subnets** node.

5. Right-click the node's name and select **New Subnet…** from the menu. The **New Object - Subnet** window appears:



New Object - Subnet

Create in: lucernpub.com/Configuration/Sites/Subnets

Enter the address prefix using network prefix notation (address/prefix length), where the prefix length indicates the number of fixed bits. You can enter either an IPv4 or an IPv6 subnet prefix.
Learn more about entering address prefixes.

IPv4 example: 157.54.208.0/20

IPv6 example: 3FFE:FFFF:0:C000::/64

Prefix:

Prefix name in Active Directory Domain Services:

Select a site object for this prefix.

Site Name
Default-First-Site-Name

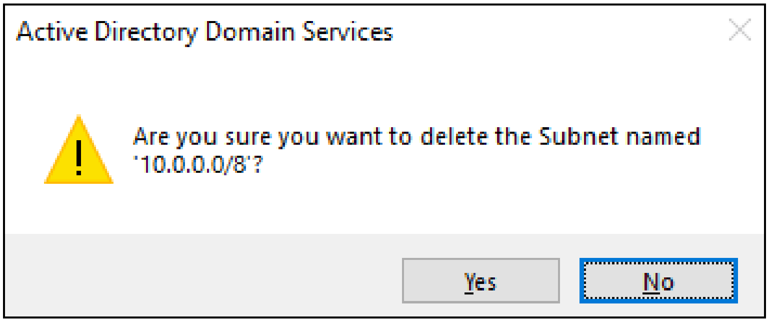
Figure 5.6 – New Object - Subnet window

6. Type a prefix for the subnet in the **Prefix:** field.
7. Select an Active Directory site for the subnet.
8. Click **OK** to create a subnet and close the **New Object - Subnet** window.

To delete a subnet, perform these steps:

1. Press Start.
2. Search for **Active Directory Sites and Services** and select it from the search results or run `dssite.msc`. The **Active Directory Sites and Services** window appears.

3. In the left navigation pane, expand the **Sites** node and then the **Subnets** node.
4. Right-click the name of the subnet you want to delete and select **Delete** from the menu. The **Active Directory Domain Services** pop-up window appears:



Active Directory Domain Services

Are you sure you want to delete the Subnet named '10.0.0.0/8'?

Figure 5.7 – Active Directory Domain Services pop-up window

5. In the **Active Directory Domain Services** pop-up window, click **Yes** to answer the question **Are you sure you want to delete the Subnet named 'prefix'?**.

To change the description of a subnet, perform these steps:

1. Press Start.
2. Search for **Active Directory Sites and Services** and select it from the search results or run `dssite.msc`. The **Active Directory Sites and Services** window appears.
3. In the left navigation pane, expand the **Sites** node and then the **Subnets** node.
4. Select a subnet you want to manage.
5. Right-click the subnet and select **Properties** from the menu.
6. On the **General** tab, type a description for the Active Directory subnet.
7. Press **OK** when done.

To change the location of a subnet, perform these steps:

1. Press Start.
2. Search for **Active Directory Sites and Services** and select it from the search results or run `dssite.msc`. The **Active Directory Sites and Services** window appears.
3. In the left navigation pane, expand the **Sites** node and then the **Subnets** node.
4. Select a subnet you want to manage.
5. Right-click the subnet and select **Properties** from the menu.
6. Navigate to the **Location** tab.

7. On the **Location** tab, type a location for the Active Directory subnet.
8. Press **OK** when done.

To change the networking prefix of a subnet, perform these steps:

1. Press Start.
2. Search for **Active Directory Sites and Services** and select it from the search results or run `dssite.msc`. The **Active Directory Sites and Services** window appears.
3. In the left navigation pane, expand the **Sites** node and then the **Subnets** node.
4. Select a subnet you want to manage.
5. Right-click the subnet and select **Properties** from the menu.
6. On the **General** tab, type the new network prefix for the Active Directory subnet.
7. Press **OK** when done.

To change the site of a subnet, perform these steps:

1. Press Start.
2. Search for **Active Directory Sites and Services** and select it from the search results or run `dssite.msc`. The **Active Directory Sites and Services** window appears.
3. In the left navigation pane, expand the **Sites** node and then the **Subnets** node.
4. Select a subnet you want to manage.
5. Right-click the subnet and select **Properties** from the menu.
6. On the **General** tab, from the drop-down list of Active Directory sites, select an existing Active Directory site, or select an empty entry from the list to remove the subnet from any Active Directory site.
7. Press **OK** when done.

Using Windows PowerShell

Use the following line of PowerShell code to create a subnet:

```
New-ADReplicationSubnet -Name 10.0.0.0/8 -Site Default-First-
Site-Name
```

Use the following line of PowerShell code to delete a subnet:

```
Remove-ADReplicationSubnet -Identity 10.0.0.0/8
```

Use the following line of PowerShell code to change the description of a subnet:

```
Set-ADReplicationSubnet -Identity 10.0.0.0/8 -Description "New
description here"
```

Use the following line of PowerShell code to change the location of a subnet:

```
Set-ADReplicationSubnet -Identity 10.0.0.0/8 -Location "New
location here"
```

Use the following line of PowerShell code to change the site of a subnet:

```
Set-ADReplicationSubnet -Identity 10.0.0.0/8 -Site NewSiteName
```

Windows PowerShell makes it easy to make bulk changes to Active Directory subnets.

How it works...

Based on the IP address of the device, member server, or domain controller, the subnet determines the Active Directory site it needs to become a member of.

Subnets use the network prefix notation for network segments. This means that the network IP address is noted, followed by the number of bits in the subnet mask. This way, `10.0.0.0/8` denotes a network with **IP version 4** (**IPv4**) addresses starting with `10.0.0.1` up to `10.255.255.254`. In the same way, `10.31.1.1/32` denotes a host with IPv4 address `10.31.1.1`.

Subnets with less available IP addresses have precedence over larger subnets. This provides the ability to create a catch-all subnet at the main location and branch off smaller networks to outlying Active Directory sites.

Both **IP version 6** (**IPv6**) and IPv4 address ranges can be used with Active Directory subnets.

See also

To finish the configuration of the Active Directory site, refer to the following recipes:

- *Creating a site*
- *Managing a site*
- *Creating a site link*
- *Managing a site link*

Creating a site link

Active Directory site links help domain controllers find replication partners in other Active Directory sites.

Getting ready

To create an Active Directory site link, you'll need to be signed in with an account that is a member of the **Enterprise Admins** group.

To adhere to the recommended practices, be sure you have created two Active Directory sites already. Then, create a site link between exactly two Active Directory sites, for each combination. Only when the network topology is not fully routed might it make sense to create an Active Directory site link with more than two sites.

While the naming of Active Directory objects should ideally adhere to the naming convention for the environment, a common practice is to name Active Directory site links for the two sites they connect, separated by a hyphen.

How to do it...

This recipe describes two ways to do this:

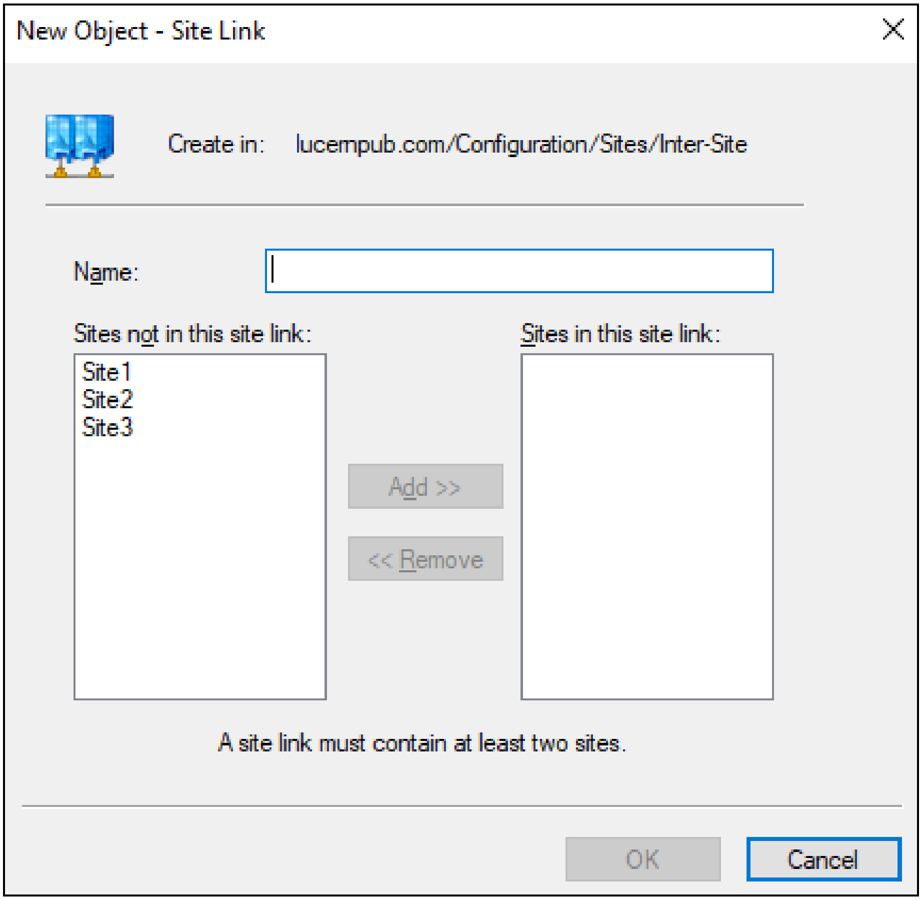
- Using **Active Directory Sites and Services**
- Using Windows PowerShell

Using Active Directory Sites and Services

To create an Active Directory site link with default settings using **Active Directory Sites and Services**, perform these steps:

1. Press Start.
2. Search for **Active Directory Sites and Services** and select it from the search results or run `dssite.msc`. The **Active Directory Sites and Services** window appears.
3. In the left navigation pane, expand the **Sites** node.
4. In the left navigation pane, expand the **Inter-Site Transports** node.

5. Right-click the **IP** node and click **New Site link…**. The **New Object - Site Link** window appears:



Create in: lucempub.com/Configuration/Sites/Inter-Site

Site1
Site2
Site3

A site link must contain at least two sites.

Figure 5.8 – New Object - Site Link window

6. Type a logical name for the Active Directory site link in the **Name:** field.
7. In the **Sites not in this site link:** field, select two Active Directory sites for this Active Directory site link. Click the **Add** >> button to move them to the **Sites in this site link:** field.
8. Click the **OK** button to create an Active Directory site link and close the **New Object - Site Link** window.

Using Windows PowerShell

To create an Active Directory site link using the Active Directory module for Windows PowerShell, use the `New-ADReplicationSiteLink` PowerShell cmdlet. Here's an example of the simplest line of PowerShell code to achieve this goal:

```
New-ADReplicationSiteLink -Name "SiteLinkName" -SitesIncluded
Site1,Site2
```

You can verify your change by listing the Active Directory sites:

```
Get-ADReplicationSiteLink -Filter * | Select-Object Name
```

Omitting the pipe in the last line of Windows PowerShell code provides more details on Active Directory sites, including their `Cost` and `SitesIncluded` attributes.

How it works…

Active Directory site links help domain controllers find replication partners in other Active Directory sites. Additionally, when all domain controllers are unavailable in an Active Directory site or no domain controllers were ever available, site links help devices to find the nearest domain controller.

There are two types of Active Directory site links:

- IP-based site links
- SMTP-based site links

IP-based site links are preferred, but when a networking connection is low on bandwidth and is unreliable, SMTP-based site links may offer better performance.

> **Note**
>
> SMTP-based site links do not require Microsoft Exchange Server or any other mail server functionality. Active Directory merely uses the same mail protocol.

See also

To finish the configuration of the Active Directory site, refer to the following recipes:

- *Creating a site*
- *Managing a site*
- *Managing a site link*

Managing a site link

After creating an Active Directory site link, you might need to manage it. You might want to do the following:

- Rename it.
- Change its description.
- Change its location.
- Modify sites that are part of the site link.
- Delete it.

Let's discuss each of these in the upcoming sections.

Getting ready

To manage an Active Directory site link, you'll need to be signed in with an account that is a member of the **Enterprise Admins** group.

How to do it...

Site links can be managed and deleted using **Active Directory Sites and Services** and the Active Directory module for Windows PowerShell.

Using Active Directory Sites and Services

To rename an Active Directory site link using **Active Directory Sites and Services**, perform these steps:

1. Press Start.
2. Search for **Active Directory Sites and Services** and select it from the search results or run `dssite.msc`. The **Active Directory Sites and Services** window appears.
3. In the left navigation pane, expand the **Sites** node and then the **Inter-Site Transports** node.
4. Select either **IP** or **SMTP**.
5. In the main pane, select a site link that you want to rename.
6. Right-click the site link and click **Rename** in the menu.
7. Type the new name for the Active Directory site link and press *Enter* when done.

To change the description for an Active Directory site link using **Active Directory Sites and Services**, perform these steps:

1. Press Start.
2. Search for **Active Directory Sites and Services** and select it from the search results or run `dssite.msc`. The **Active Directory Sites and Services** window appears.
3. In the left navigation pane, expand the **Sites** node and then the **Inter-Site Transports** node.
4. Select either **IP** or **SMTP**.
5. In the main pane, select a site link that you want to manage.
6. Right-click the site link and click **Properties** in the menu.
7. On the **General** tab, type a new description for the Active Directory site link.
8. Click **OK**.

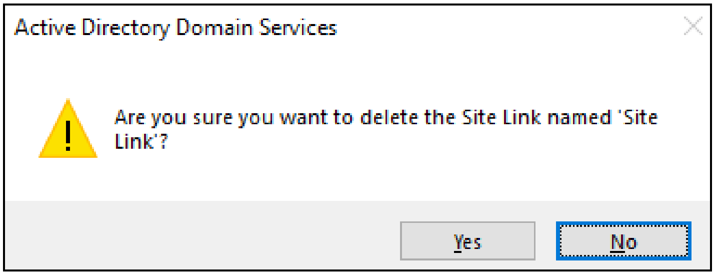
To modify sites that are part of the site link using **Active Directory Sites and Services**, perform these steps:

1. Press Start.
2. Search for **Active Directory Sites and Services** and select it from the search results or run `dssite.msc`. The **Active Directory Sites and Services** window appears.
3. In the left navigation pane, expand the **Sites** node and then the **Inter-Site Transports** node.
4. Select either **IP** or **SMTP**.
5. In the main pane, select a site link that you want to manage.
6. Right-click the site link and click **Properties** in the menu.
7. On the **General** tab, select Active Directory sites to include in the Active Directory site link and/or exclude from the Active Directory site link.
8. Click **OK**.

To delete an Active Directory site link using **Active Directory Sites and Services**, perform these steps:

1. Press Start.
2. Search for **Active Directory Sites and Services** and select it from the search results or run `dssite.msc`. The **Active Directory Sites and Services** window appears.
3. In the left navigation pane, expand the **Sites** node and then the **Inter-Site Transports** node.

4. Select either **IP** or **SMTP**.
5. In the main pane, select a site link that you want to delete.
6. Right-click the site link and click **Delete** in the menu. The **Active Directory Domain Services** pop-up window appears:



Active Directory Domain Services

Are you sure you want to delete the Site Link named 'Site Link'?

Figure 5.9 – Active Directory Domain Services pop-up window

7. In the **Active Directory Domain Services** pop-up window, click **Yes** to answer the question **Are you sure you want to delete the Site Link named 'Site Link'?**.

Using Windows PowerShell

Use the following line of PowerShell code to rename an Active Directory site link:

```
Rename-ADObject -Identity "CN=SiteLink1,CN=IP,CN=Inter-Site
Transports,CN=Sites,CN=Configuration,DC=LucernPub,DC=com"
-NewName "NewADSiteLinkName"
```

Replace `SiteLink1` and `DC=LucernPub,DC=com` with values that represent your environment.

Use the following line of PowerShell code to change the description of an Active Directory site link:

```
Set-ADReplicationSiteLink -Identity "SiteLink1" -Description
"New description here"
```

Replace `SiteLink1` with a value that represents site links in your environment.

Use the following line of PowerShell code to modify sites that are part of the site link:

```
Set-ADReplicationSiteLink -Identity "SiteLink1" -SitesIncluded
Site1,Site2
```

Replace `SiteLink1` with a value that represents site links in your environment and `Site1` and `Site2` with values that represent sites in your environment.

Use the following line of PowerShell code to delete a site link:

```
Remove-ADReplicationSiteLink -Identity "SiteLink1"
```

Replace `SiteLink1` with a value that represents a site link in your environment.

See also

Take a look at the following recipes to further manage Active Directory site links:

- *Modifying replication settings for an Active Directory site link*
- *Creating a site link bridge*

Modifying replication settings for an Active Directory site link

After creating Active Directory site links, you might need to change the following replication settings:

- Change the cost for an Active Directory site link.
- Change the replication interval for an Active Directory site link.
- Change the replication schedule for an Active Directory site link.

Getting ready

To manage an Active Directory site link, you'll need to be signed in with an account that is a member of the **Enterprise Admins** group.

How to do it...

Replication settings for Active Directory site links can be managed using **Active Directory Sites and Services** and the Active Directory module for Windows PowerShell.

Using Active Directory Sites and Services

To modify the cost for an Active Directory site link using **Active Directory Sites and Services**, perform these steps:

1. Press Start.
2. Search for **Active Directory Sites and Services** and select it from the search results or run `dssite.msc`. The **Active Directory Sites and Services** window appears.
3. In the left navigation pane, expand the **Sites** node and then the **Inter-Site Transports** node.
4. Select either **IP** or **SMTP**.
5. In the main pane, select a site link that you want to manage.
6. Right-click the site link and click **Properties** in the menu.
7. On the **General** tab, in the **Cost:** field, enter a new cost for the site link.
8. Click **OK**.

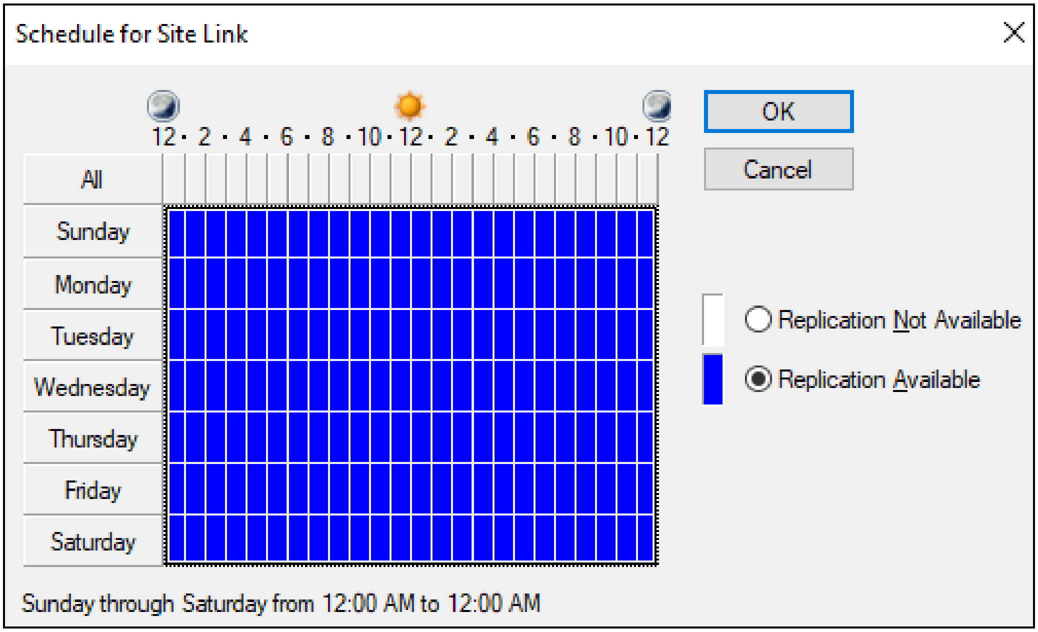
To modify the replication interval using **Active Directory Sites and Services**, perform these steps:

1. Press Start.
2. Search for **Active Directory Sites and Services** and select it from the search results or run `dssite.msc`. The **Active Directory Sites and Services** window appears.
3. In the left navigation pane, expand the **Sites** node and then the **Inter-Site Transports** node.
4. Select either **IP** or **SMTP**.
5. In the main pane, select a site link that you want to manage.
6. Right-click the site link and click **Properties** in the menu.
7. On the **General** tab, in the **Replicate every** field, enter a new replication interval, specified in minutes. The lowest value that can be added here is `15` minutes.
8. Click **OK**.

To modify the replication schedule using **Active Directory Sites and Services**, perform these steps:

1. Press Start.
2. Search for **Active Directory Sites and Services** and select it from the search results or run `dssite.msc`. The **Active Directory Sites and Services** window appears.

3. In the left navigation pane, expand the **Sites** node and then the **Inter-Site Transports** node.
4. Select either **IP** or **SMTP**.
5. In the main pane, select a site link that you want to manage.
6. Right-click the site link and click **Properties** in the menu.
7. On the **General** tab, click the **Change Schedule…** button. The **Schedule for 'site link name'** window appears:



Schedule for Site Link

○ Replication Not Available

● Replication Available

Sunday through Saturday from 12:00 AM to 12:00 AM

Figure 5.10 – Schedule for Site Link window

8. Exclude 60-minute time periods from the replication schedule.
9. Click **OK**.
10. Back in the **Properties** window of the Active Directory site link, click **OK** to save the new replication schedule for the Active Directory site link.

Using Windows PowerShell

Use the following line of PowerShell code to modify the cost of an Active Directory site link:

```
Set-ADReplicationSiteLink -Identity SiteLink1 -Cost 50
```

Replace `SiteLink1` with a value that represents a site link in your environment.

Use the following line of PowerShell code to modify the replication interval for an Active Directory site link to `30` minutes:

```
Set-ADReplicationSiteLink -Identity SiteLink1
-ReplicationFrequencyInMinutes 30
```

Replace `SiteLink1` with a value that represents site links in your environment.

The `-replicationschedule` parameter for `Set-ADReplicationSiteLink` allows for the most granular management of the replication schedule for an Active Directory site link. Just as with the graphical schedule tool, it can be used to specify the days of the week for the replication schedule. But in contrast to the graphical tool, it can be used to define available or non-available replication times per minute.

Use the following lines of PowerShell code to modify the replication schedule for an Active Directory site link by only allowing replication between 8:00 A.M. and 5:00 P.M:

```
$replicationSchedule = New-Object -TypeName System.
DirectoryServices.ActiveDirectory.ActiveDirectorySchedule
$replicationSchedule.
SetDailySchedule("Eight","Zero","Seventeen","Zero")
Set-ADReplicationSiteLink SiteLink1 -ReplicationSchedule
$replicationSchedule
```

Replace `SiteLink1` with a value that represents a site link in your environment.

The default replication schedule is to allow replication Sunday through Saturday from 12:00 A.M. to 12:00 A.M. and is defined as `$null` in the `Get-ADReplicationSiteLink` and `Set-ADReplicationSiteLink` Windows PowerShell cmdlets.

How it works...

This recipe details site link costs and site link replication schedules.

Site link costs

The cost for an Active Directory site link defines the weight of the link. An Active Directory site link with a high cost tells domain controllers, member servers, and devices that the link is less preferable to use compared to an Active Directory site link with a lower cost.

The following factors would cause a higher cost when designing an Active Directory site link layout:

- The available bandwidth for Active Directory traffic on the networking connection (the networking connection might be flooded with other traffic already)
- The reliability of the networking connection, in terms of unavailability, such as traffic loss or service level

In complex networking environments, where ring, hub-and-spoke, and mesh topologies are used, **end-to-end** (**E2E**) costs for traversing site links should also be taken into consideration.

The default cost of an Active Directory site link is 100.

Site link replication schedules

As opposed to intra-site replication, inter-site replication—by default—uses a replication schedule. Instead of processing change notifications, domain controllers wait for their scheduled replication time, based on an interval.

By default, domain controllers on each side of the site link replicate every 180 minutes, and the site link's replication schedule is to allow replication Sunday through Saturday from 12:00 A.M. to 12:00 A.M.

See also

To change other properties of an Active Directory site link, look at the *Managing a site link* recipe.

Creating a site link bridge

For IP-based and SMTP-based Active Directory site links, an option exists to disable the **Bridge all site links** option. This option is enabled, by default, and defines the site links as transitive. This transitivity means that domain controllers, member servers, and devices may traverse site links to get from *site A* to *site C* through *site B*, even though *site A* and *site C* are only connected with site links to *site B* and not with each other.

In complex networking environments, more granular control over replication and service discovery might be needed. In these scenarios, disabling site link transitivity and manually defining site link bridges might be the preferred method.

Getting ready

To create a site link bridge and disable the **Bridge all site links** option, you'll need to be signed in with an account that is a member of the **Enterprise Admins** group.

The **Bridge all site links** option needs to be disabled for all site links. Perform these actions while signed in with an account that is a member of the **Enterprise Admins** group:

1. Press Start.
2. Search for **Active Directory Sites and Services** and select it from the search results or run `dssite.msc`. The **Active Directory Sites and Services** window appears.
3. In the left navigation pane, expand the **Sites** node and then the **Inter-Site Transports** node.
4. Right-click the **IP** node and select **Properties** from the menu.
5. On the **General** tab, deselect the **Bridge all site links** option.
6. Click **OK**.

Repeat the last three steps for SMTP-based site links, if needed.

The **Bridge all site links** option can also be disabled using the Active Directory module for Windows PowerShell, as follows:

```
Set-ADObject "CN=IP,CN=Inter-Site
Transports,CN=Sites,CN=configuration,DC=LucernPub,DC=com"
-Replace @{Options=2}
```

Replace `DC=LucernPub,DC=com` with values that represent your Active Directory environment.

How to do it...

You can take advantage of manually created site link bridges after you disable the **Bridge all site links** option.

To create a site link bridge using **Active Directory Sites and Services**, perform these steps:

1. Press Start.
2. Search for **Active Directory Sites and Services** and select it from the search results or run `dssite.msc`. The **Active Directory Sites and Services** window appears.
3. In the left navigation pane, expand the **Sites** node and then the **Inter-Site Transports** node.
4. Right-click the **IP** node and select **New Site Link Bridge…** from the menu. The **New Object - Site Link Bridge** window appears.
5. Select site links from the **Site links not in this site link bridge:** list.
6. Click **Add** >> to add site links to the list underneath **Site links in this site link bridge**.
7. Click **OK**.

To create a site link bridge using the Active Directory module for Windows PowerShell, use the following line of PowerShell code:

```
New-ADReplicationSiteLinkBridge "SiteLinkBridgeName"
-SiteLinksIncluded SiteLink1,SiteLink2
```

Replace `SiteLink1` and `SiteLink2` with values that represent site links in your environment. The site links can be denoted as **distinguished names** (**DNs**) or as names.

See also

To change properties of an Active Directory site link, look at the *Managing a site link* and *Modifying replication settings for an Active Directory site link* recipes.

Managing bridgehead servers

Bridgehead servers are the domain controllers on each side of an Active Directory site link that take care of inter-site replication on behalf of all the domain controllers in their respective sites.

Getting ready

To manage bridgehead servers, you'll need to be signed in with an account that is a member of the **Enterprise Admins** group.

How to do it...

Bridgehead servers can be set using **Active Directory Sites and Services** and the Active Directory module for Windows PowerShell.

Using Active Directory Sites and Services

To set a domain controller as the bridgehead server for the Active Directory site, using **Active Directory Sites and Services**, perform these steps:

1. Press Start.
2. Search for **Active Directory Sites and Services** and select it from the search results or run `dssite.msc`. The **Active Directory Sites and Services** window appears.
3. In the left navigation pane, expand the **Sites** node.
4. In the left navigation pane, expand the **Servers** node.
5. In the left navigation pane, right-click the domain controller that you want to set as the bridgehead server and select **Properties** from the menu.
6. On the **General** tab, select **IP** and/or **SMTP** from the list under **Transports available for inter-site data transfer:**. Click **Add** >> to set the DC as the bridgehead server for the selected transport(s) in the **This server is a preferred bridgehead server for the following transports:** list underneath.
7. Click **OK**.

Using Windows PowerShell

Run the following line of PowerShell code to set a domain controller as the bridgehead server for the IP transport on its Active Directory site:

```
Set-ADObject -Identity "CN=DC01,CN=Servers,CN=Default-First-
Site-Name,CN=Sites,CN=Configuration,DC=LucernPub,DC=com" -Add
@{bridgeHeadTransportList="CN=IP,CN=Inter-Site Transports,
CN=Sites,CN=Configuration,DC=LucernPub,DC=com"}
```

Replace `DC01` and `DC=LucernPub,DC=com` with values that represent your environment.

Run the following line of PowerShell code to set a domain controller as the bridgehead server for the SMTP transport on its Active Directory site:

```
Set-ADObject -Identity "CN=DC01,CN=Servers,CN=Default-First-
Site-Name,CN=Sites,CN=Configuration,DC=LucernPub,DC=com" -Add
@{bridgeHeadTransportList="CN=SMTP,CN=Inter-Site Transports,
CN=Sites,CN=Configuration,DC=LucernPub,DC=com"}
```

Replace `DC01` and `DC=LucernPub,DC=com` with values that represent your environment.

How it works...

By default, bridgehead servers are dynamically set to domain controllers by the KCC. However, bridgehead servers can also be set manually to accommodate even the most complex routing challenges in a networking environment.

When a domain controller is set as the bridgehead server for an Active Directory partition but becomes unavailable, replication stops for that partition to the Active Directory site where the domain controller resides. Hence, you should configure at least two bridgehead servers per site where possible.

See also

To further manage Active Directory sites, refer to the following recipes:

- *Creating a site*
- *Managing subnets*
- *Creating a site link*
- *Managing a site link*

Managing the ISTG and KCC

The ISTG and KCC play an important role in Active Directory replication.

Getting ready

To manage the ISTG and the KCC, you'll need to be signed in with an account that is a member of the **Enterprise Admins** group.

How to do it...

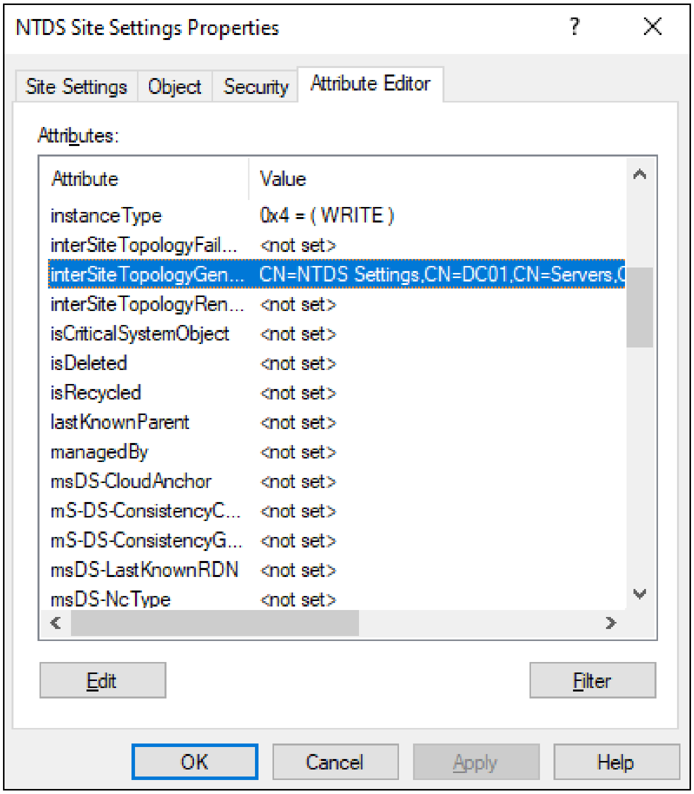
Moving the ISTG to another domain controller and disabling the KCC can be performed using **Active Directory Sites and Services** and the Active Directory module for Windows PowerShell.

Using Active Directory Sites and Services

To move the ISTG to a different domain controller using **Active Directory Sites and Services**, perform these steps:

1. Press Start.
2. Search for **Active Directory Sites and Services** and select it from the search results or run `dssite.msc`. The **Active Directory Sites and Services** window appears.
3. In the left navigation pane, expand the **Sites** node.
4. In the left navigation pane, expand the Active Directory site for which you want to manage the ISTG.
5. Expand the **Servers** node.
6. Expand the domain controller object.
7. In the main pane, right-click the **NTDS Settings** node and select **Properties** from the menu.
8. Navigate to the **Attribute Editor** tab.

9. In the **Attributes:** list, find and select the `interSiteTopologyGenerator` attribute:



Attribute	Value
instanceType	0x4 = (WRITE)
interSiteTopologyFail...	<not set>
interSiteTopologyGen...	CN=NTDS Settings,CN=DC01,CN=Servers,C
interSiteTopologyRen...	<not set>
isCriticalSystemObject	<not set>
isDeleted	<not set>
isRecycled	<not set>
lastKnownParent	<not set>
managedBy	<not set>
msDS-CloudAnchor	<not set>
mS-DS-ConsistencyC...	<not set>
mS-DS-ConsistencyG...	<not set>
msDS-LastKnownRDN	<not set>
msDS-NcType	<not set>

Figure 5.11 – interSiteTopologyGenerator attribute

10. Click **Edit**.
11. In the **Value** field, change the DN of the domain controller to the DN of the domain controller you want to set as the ISTG.
12. Click **OK** to set the new domain controller.
13. Click **OK** to close the **NTDS Settings Properties** window.

To disable the KCC and/or ISTG on an Active Directory site, perform these steps:

1. Press Start.
2. Search for **Active Directory Sites and Services** and select it from the search results or run `dssite.msc`. The **Active Directory Sites and Services** window appears.

   In the left navigation pane, select an Active Directory site for which you want to manage the KCC and/or ISTG.

3. Expand the **Servers** node.
4. Expand the domain controller object.
5. In the main pane, right-click the **NTDS Settings** node and select **Properties** from the menu.
6. Navigate to the **Attribute Editor** tab.
7. In the **Attributes:** list, find and select the `options` attribute. The `options` attribute is made up of bits, ranging from 0 to 4. When you enable a bit, you add the corresponding `2^x` for the bit.

   To disable the KCC, enable bit 4. Enter `16` as the value. To disable the ISTG, enable bit 0. Enter `1` as the value. To disable the KCC and ISTG, enable bits 0 and 4. Enter `17` as the value.
8. Click **OK** to close the **NTDS Settings Properties** window.

Using Windows PowerShell

Run the following line of PowerShell code to move the ISTG to a different domain controller using the Active Directory module for Windows PowerShell:

```
Set-ADObject -Identity "CN=NTDS Site
Settings,CN=Site1,CN=Sites,CN=Configuration,DC=LucernPub,
DC=com" -Replace @{interSiteTopologyGenerator="CN=NTDS
Settings,CN=DC01,CN=Servers,CN=Site1,CN=Sites,CN=Configuration,
DC=LucernPub,DC=com"}
```

Replace `Site1`, `DC=LucernPub,DC=com`, and `DC01` with values that represent your environment.

Run the following line of PowerShell code to disable the KCC and/or the ISTG for an Active Directory site using the Active Directory module for Windows PowerShell:

```
Set-ADObject "CN=NTDS Site
Settings,CN=Site1,CN=Sites,CN=Configuration,DC=LucernPub,
DC=com" -Replace @{Options="<Value>"}
```

Replace `Site1` and `DC=LucernPub,DC=com` with values that represent your environment.

Define one of the following values, depending on the outcome you desire:

Outcome	Value
Disable the KCC	`16`
Disable the ISTG	`1`
Disable the KCC and ISTG	`17`

Table 5.1 – Site options values and their outcomes

After issuing the preceding line of PowerShell code, the changes take effect immediately.

How it works...

In every Active Directory site, one domain controller is dynamically assigned to the ISTG. Its role is to automatically create inter-site connection objects between the bridgehead server in its respective Active Directory site and bridgehead servers in other Active Directory sites.

While you can disable the ISTG and create inter-site connection objects manually, it is highly recommended to let the ISTG create connection objects based on Active Directory sites, Active Directory site links, and bridgehead servers.

The KCC is a process that runs on all domain controllers within an Active Directory site to automatically create intra-site connection objects between domain controllers in the Active Directory site.

While you can disable the KCC and create intra-site connection objects manually, it is highly recommended to let the KCC create connection objects.

See also

To further manage Active Directory sites, refer to the following recipes:

- *Creating a site*
- *Managing subnets*
- *Creating a site link*
- *Managing a site link*

Managing UGMC

To avoid placing a global catalog in every Active Directory site and not requiring a global catalog server for every sign-in, you can use UGMC.

Getting ready

To enable UGMC for an Active Directory site, all domain controllers in the site will need to run Windows Server 2003, or a newer version of Windows Server.

To enable or disable UGMC, you'll need to be signed in with an account that is a member of the **Enterprise Admins** group.

How to do it...

Managing UGMC can be performed using **Active Directory Sites and Services** and the Active Directory module for Windows PowerShell.

Using Active Directory Sites and Services

To manage UGMC for an Active Directory site using **Active Directory Sites and Services**, perform these steps:

1. Press Start.
2. Search for **Active Directory Sites and Services** and select it from the search results or run `dssite.msc`. The **Active Directory Sites and Services** window appears.
3. In the left navigation pane, expand the **Sites** node.
4. Select an Active Directory site you want to manage UGMC for.
5. In the main pane, right-click the **NTDS Settings** node and select **Properties** from the menu.
6. On the **General** tab, under **Universal Group Membership Caching**, select the **Enable Universal Group Membership Caching** option.
7. Optionally, select a specific Active Directory site from the **Refresh cache from:** drop-down list, or leave it at **<Default>** to refresh the universal group membership cache from the global catalog server(s) in the Active Directory site nearest to the Active Directory site you're managing UGMC for.
8. Click **OK**.

To disable UGMC, deselect the **Enable Universal Group Membership Caching option**.

Using Windows PowerShell

Run the following line of PowerShell code to enable UGMC for an Active Directory site:

```
Set-ADReplicationSite -Identity Site1
-UniversalGroupCachingEnabled $True
```

Replace `Site1` with a value that represents a site link in your environment.

Run the following lines of PowerShell code to disable UGMC for an Active Directory site:

```
Set-ADReplicationSite -Identity Site1
-UniversalGroupCachingEnabled $False
```

Replace `Site1` with a value that represents a site link in your environment.

How it works...

In multi-domain and multi-forest environments, it is a common practice to use universal groups to traverse Active Directory trusts. This is also why, in Microsoft Exchange Server, it is recommended you create these types of groups for distribution lists, instead of domain local groups or global groups.

However, for every sign-in, a global catalog server is required to enumerate the universal groups the account is a member of. In multi-domain and multi-forest environments, global catalog servers require more replication and, therefore, replication bandwidth. Often, it is far from ideal to place global catalogs in poorly connected Active Directory sites.

To accommodate the scenario of a non-global catalog in an Active Directory site, but not require a global catalog for every sign-in, you can use UGMC. Now, only the first sign-in in an Active Directory site involves contacting a global catalog. In the background, domain controllers will exchange the information needed with global catalog servers to keep group memberships up to date throughout the Active Directory environment, every 8 hours.

See also

Please refer to the *Managing global catalog servers* and *Managing domain controllers* recipes from *Chapter 2*, *Managing Domain Controllers,* for more information on global catalogs and global catalog placement.

Working with repadmin.exe

In this recipe, we'll concentrate on `repadmin.exe` for troubleshooting Active Directory replication.

Getting ready

To work with `repadmin.exe`, sign in to a domain controller.

How to do it...

`repadmin.exe` has the following high-level commands:

`repadmin.exe /kcc`	Forces the KCC on the targeted domain controller to immediately recalculate its inbound replication topology
`repadmin.exe /prp`	Allows an admin to view or modify the **Password Replication Policy** (**PRP**) for read-only domain controllers
`repadmin.exe /queue`	Displays inbound replication requests that the domain controller needs to issue to become consistent with its source replication partners
`repadmin.exe /replicate`	Triggers the immediate replication of the specified directory partition to the destination domain controller from the source domain controller
`repadmin.exe /replsingleobj`	Replicates a single object between any two domain controllers that have common directory partitions
`repadmin.exe /replsummary`	Quickly and concisely summarizes the replication state and relative health of an Active Directory forest
`repadmin.exe /rodcpwdrepl`	Triggers replication of passwords for the specified user(s) from the source domain controller to one or more read-only domain controllers
`repadmin.exe /showattr`	Displays the attributes of an object domain controller
`repadmin.exe /showbackup`	Displays the last time the domain controller was backed up
`repadmin.exe /showobjmeta`	Displays the replication metadata for a specified object stored in Active Directory, such as attribute ID, version number, originating and local **update sequence number** (**USN**), and originating server's GUID and date- and timestamp
`repadmin.exe /showrepl`	Displays the replication status and when the specified domain controller last attempted to inbound replicate Active Directory partitions
`repadmin.exe /showutdvec`	Displays the highest committed USN that the targeted domain controllers copy of Active Directory shows as committed for itself and its transitive partners
`repadmin.exe /syncall`	Synchronizes a specified domain controller with all replication partners

Table 5.2 – repadmin.exe high-level commands

Optionally, the following parameters can be used with `repadmin.exe` to specify the pre-Windows 2000 logon name for a user account that has permissions to perform operations in Active Directory:

`/u:`	DOMAIN/User name
`/pw:`	Password for the user account

Table 5.3 – repadmin.exe parameters

`userPrincipalName` sign-ins are not supported with `repadmin.exe`.

How it works...

Active Directory sites, site links, site link bridges, subnets, replication schedules, and the ISTG, KCC, and UGMC can make every Active Directory admin feel overwhelmed at first.

When troubleshooting replication in an environment with a lot of moving replication parts, the complexity of any environment hinders you from getting to the root cause.

Microsoft has offered three tools in the past to check replication:

- `replmon.exe`
- `repadmin.exe`
- The **Active Directory Replication Status Tool** (**ADREPLSTATUS**)

Unfortunately, the graphical `replmon.exe` tool, part of the Windows Server Support Tools, is no longer available for Windows Server 2008 and newer versions of Windows Server. ADREPLSTATUS moved from being an on-premises tool to a model with on-premises agents and a cloud-powered dashboard. As you can imagine, not many admins feel good about connecting their domain controllers to the internet.

See also

For more information, refer to the *Forcing replication* recipe.

Forcing replication

After troubleshooting failing networking connections, resetting Active Directory's replication topology, and/or removing domain controllers, there may be a need to check proper replication.

As an admin, you can wait for intra-site replication to occur and for inter-site replication to trigger upon the replication schedules, but there's also a way to force replication.

The obvious choice to force replication is to use `repadmin.exe`.

Getting ready

To work with `repadmin.exe`, sign in to a domain controller.

How to do it...

Issue the following command on an elevated **Command Prompt** (`cmd.exe`) on any domain controller:

```
repadmin.exe /syncall /Aped
```

Now, let's see how this works.

How it works...

This command will synchronize all partitions (`/A`) using push notifications (`/p`) in enterprise mode across Active Directory sites (`/e`), using DN instead of DNS names (`/d`).

See also

Please refer to the *Working with repadmin.exe* recipe for more information on `repadmin.exe`.

Managing inbound and outbound replication

On a per-domain controller basis, an Active Directory admin can disable inbound and/or outbound replication. Later, inbound and/or outbound replication can be re-enabled.

The obvious choice for managing inbound and outbound replication is to use `repadmin.exe`.

Getting ready

To work with `repadmin.exe`, sign in to a domain controller.

How to do it...

Issue the following command on an elevated **Command Prompt** (`cmd.exe`) on the domain controller for which you want to disable inbound replication:

```
repadmin.exe /options DC01 +DISABLE_INBOUND_REPL
```

Issue the following command on an elevated **Command Prompt** (`cmd.exe`) on the domain controller for which you want to disable outbound replication:

```
repadmin.exe /options DC01 +DISABLE_OUTBOUND_REPL
```

Issue the following command on an elevated **Command Prompt** (`cmd.exe`) on the domain controller for which you want to re-enable inbound replication:

```
repadmin.exe /options DC01 -DISABLE_INBOUND_REPL
```

Issue the following command on an elevated **Command Prompt** (`cmd.exe`) on the domain controller for which you want to re-enable outbound replication:

```
repadmin.exe /options DC01 -DISABLE_OUTBOUND_REPL
```

How it works...

The `repadmin.exe` commands use the `options` flag for the domain controller to enable and/or disable inbound and/or outbound replication.

The option to disable outbound replication sounds like a fantastic way to perform schema updates and schema extensions on the domain controller that holds the schema master FSMO role, without the need to rebuild the entire Active Directory forest or bring back all domain controllers from backups. Also, the domain controller checks whether outbound replication is enabled before attempting to extend or upgrade the Active Directory schema.

There's more...

When the goal is to enable and/or disable inbound and/or outbound replication on all domain controllers throughout the environment, replace the name of the domain controllers with `*`.

See also

The following recipes provide more information:

- *Extending the schema* in *Chapter 1, Optimizing Forests, Domains and Trusts*
- *Working with repadmin.exe*
- *Forcing replication*

Modifying the tombstone lifetime period

When an object is deleted from Active Directory, it is not actually removed from the database, as this would hinder replication of the deletion. Instead, the object is tombstoned. This tombstone prevents the object from being usable (for sign-ins, for example) and being visible in all common Active Directory tools. It also instructs the garbage collection process on each domain controller to remove the object from the database once the tombstone lifetime period has expired.

Getting ready

To modify the tombstone lifetime period, you'll need to be signed in with an account that is a member of the **Domain Admins** group.

How to do it...

You can modify the tombstone lifetime period using **ADSI Edit** and Windows PowerShell.

Using ADSI Edit

Perform these steps to modify the tombstone lifetime period for an Active Directory domain using **ADSI Edit**:

1. Press Start.
2. Search for **ADSI Edit** and select it from the search results or run `adsiedit.msc`.
3. In the left navigation pane, right-click the **ADSI Edit** node and select **Connect to…** from the menu. The **Connection Settings** window appears.
4. Select the **Select a well known Naming Context:** option and choose **Configuration** from the drop-down list.
5. Click **OK** to connect and close the **Connection Settings** window.
6. In the left navigation pane, expand the `CN=Configuration` node.
7. In the left navigation pane, expand the `CN=Services` node.

8. In the left navigation pane, expand the `CN=Windows NT` node.
9. In the left navigation pane, right-click the `CN=Directory Service` node and select **Properties** from the menu.
10. In the list of attributes, scroll down to the `tombstoneLifetime` attribute.
11. Change the value for the `tombstoneLifetime` attribute and press **OK** when done.
12. Click **OK** to close the **CN=Directory Service Properties** window.
13. Close **ADSI Edit**.

Repeat these steps for all domains in the Active Directory forest.

Using Windows PowerShell

To modify the tombstone lifetime period to `180` days, run the following line of PowerShell code on a system with the Active Directory module for Windows PowerShell installed:

```
Set-ADObject "CN=Directory Service,CN=Windows
NT,CN=Services,CN=Configuration,DC=LucernPub,DC=com" -Replace @
{"tombstoneLifetime"="180"}
```

Replace `DC=LucernPub,DC=com` with a value that represents your environment.

Repeat this line of PowerShell code for all domains in the Active Directory forest.

How it works...

The tombstone lifetime period is set at the domain level and applies to all domain controllers in the Active Directory domain.

The tombstone lifetime period instructs the GC process on each domain controller with the period to wait before removing objects from the database after receiving a `delete` update for an object and processing the update as a tombstone for the object.

The default tombstone lifetime period for an Active Directory domain that has been set up and run by domain controllers running Windows 2000 Server and Windows Server 2003 is 60 days. Active Directory domains that have begun their life run by domain controllers running Windows Server 2008 or newer versions of Windows Server have a default tombstone lifetime period of 180 days.

See also

See the *Checking for and remediating lingering objects* recipe for more information.

Managing strict replication consistency

Strict replication consistency is a mechanism on domain controllers that prevents them from replicating with a replication partner when it suspects an object that is replicated is a **lingering object**. Strict replication consistency is enabled by default but can be disabled.

Getting ready

To manage strict replication, sign in to a domain controller with local administrative privileges.

How to do it...

To disable strict replication consistency on a domain controller, perform the following steps:

1. Press Start.
2. Search for the **Registry Editor** and select it from the search results or run `regedit.exe`.
3. In the top address bar, type `HKLM\SYSTEM\CurrentControlSet\Services\NTDS \Parameters` or navigate to the **Parameters** node in the left navigation pane.
4. In the main pane, right-click an empty space and select **New** and then **DWORD Value** from the menu.
5. Name the new **double-word** (**DWORD**) value `Strict Replication Consistency`.
6. Click **OK**.
7. Double-click the new `Strict Replication Consistency` value.
8. Enter `1` to disable strict replication consistency.
9. Click **OK**.
10. Reboot the domain controller.

To re-enable strict replication consistency on a domain controller, perform the following steps:

1. Open the **Registry Editor** (`regedit.exe`).
2. In the top address bar, type `HKLM\SYSTEM\CurrentControlSet\Services\NTDS \Parameters` or navigate to the **Parameters** node in the left navigation pane.
3. In the main pane, double-click the `Strict Replication Consistency` value.
4. Enter `0` to enable strict replication consistency.
5. Click **OK**.
6. Reboot the domain controller.

How it works...

When a domain controller suspects an object that is replicated inbound is a lingering object and strict replication consistency is enabled, the domain controller will stop replicating with the replication partner it suspects of outbound-replicating the suspected object.

Both inbound replication and outbound replication will be stopped at the specific domain controller.

A lingering object is an object that was tombstoned and then garbage-collected on all domain controllers, but after that point in time was reintroduced by a domain controller that was restored from a backup, image, or snapshot that was older than the tombstone lifetime period.

Strict replication consistency is a per-domain controller setting and is enabled by default.

Upgrading SYSVOL replication from FRS to DFSR

Since Windows 2000 Server, Active Directory features SYSVOL. This shared folder is replicated between all domain controllers in an Active Directory domain and contains commonly accessed files such as GPOs, logon scripts, and logoff scripts, and can hold any file an Active Directory admin wants it to hold.

In legacy Active Directory environments, SYSVOL replication uses FRS. This should be migrated to DFSR.

Getting ready

The deprecation of FRS may or may not be a problem in your Active Directory environment.

If your Active Directory environment has ever featured Windows 2000 Server, Windows Server 2003, or Windows Server 2003 R2-based domain controllers, you have some work to do. Your domain controllers rely on FRS to replicate the contents of the SYSVOL shares between them, even when these domain controllers have long been decommissioned and/or replaced with Windows Server 2008, Windows Server 2008 R2, and/or Windows Server 2012-based domain controllers.

There is a detailed guide available from Microsoft to help you with this task. This *SYSVOL Replication Migration Guide* details the same steps as this recipe to perform migration from FRS to DFSR but also features rollback steps and a command-line reference for `dfsrmig.exe`.

To change the replication of SYSVOL from FRS to DFSR, all domain controllers in the Active Directory domain need to run Windows Server 2008 or a newer version of Windows Server, and the Active Directory **domain functional level** (**DFL**) needs to be Windows Server 2008 or a higher functional level.

Sign in to the domain controller that holds the **Primary Domain Controller Emulator** (**PDCE**) FSMO role, using an account that is a member of the **Domain Admins** group.

How to do it...

There is only one command-line tool available to migrate from FRS replication to DFS replication: `dfsrmig.exe`. This tool guides you through the following states.

The initial state

When you begin upgrading SYSVOL replication from FRS to DFSR, all domain controllers in the Active Directory domain will be in this state. They are ready to begin the migration.

To go from the initial state to the prepared state, use the following command on an elevated **Command Prompt** (`cmd.exe`):

```
dfsrmig.exe /setglobalstate 1
```

The domain controllers will be instructed to do the work to go to the next stage. Use the following command to inspect the process:

```
dfsrmig.exe /getmigrationstate
```

When the output of this command indicates that all domain controllers have successfully migrated to the `Prepared` global state, you're good to go with the next command to get your Active Directory domain toward the next state.

The prepared state

The prepared state configures the DFSR replication service to replicate a copy of the original SYSVOL. When all domain controllers reach the prepared state, DFSR is properly configured, and it has completed an initial synchronization. However, in the prepared state, the replication of SYSVOL still depends on FRS.

To go from the prepared state to the redirected state, use the following command on an elevated **Command Prompt** (`cmd.exe`):

```
dfsrmig.exe /setglobalstate 2
```

The domain controllers will be instructed to do the work to go to the next stage. Use the following command to inspect the process:

```
dfsrmig.exe /getmigrationstate
```

When the output of this command indicates that all domain controllers have successfully migrated to the `Redirected` global state, you're good to go with the next command to get your Active Directory domain toward the next state.

The redirected state

In the redirected state, the live SYSVOL share (mapped to the old `SYSVOL` folder that FRS replicates) is mapped to a new copy of the `SYSVOL` folder, replicated by the DFS replication service. From this point onward, SYSVOL replication depends on DFS replication, but the migration is not quite finished yet.

To go from the redirected state to the eliminated state, use the following command on an elevated **Command Prompt** (`cmd.exe`):

```
dfsrmig.exe /setglobalstate 3
```

The domain controllers will be instructed to do the work to go to the next stage. Use the following command to inspect the process:

```
dfsrmig.exe /getmigrationstate
```

When the output of this command indicates that all domain controllers have successfully migrated to the `Eliminated` global state, read on.

The eliminated state

At the end of the eliminated state, the FRS SYSVOL replica set and the old `SYSVOL` folder are deleted. Not only does SYSVOL replication depend on DFSR, but also, all remnants of SYSVOL FRS replication are gone.

How it works...

Typically, SYSVOL is replicated between domain controllers using Windows NT 4 Server's legacy FRS, unless the Active Directory domain was started on a domain controller running Windows Server 2008 or any newer version of Windows Server. In the latter case, SYSVOL is replicated between domain controllers using the new and more robust DFSR.

Starting with Windows Server 2003 R2, Microsoft began deprecating the use of FRS. In Windows Server 2003 R2, the more efficient and robust DFSR service replaced FRS for replication of DFS folders, but FRS was still used to replicate the `SYSVOL` folder on domain controllers.

DFSR uses a more efficient, scalable, and reliable file replication protocol than FRS. Therefore, it is much faster, especially when you make small changes to large files with **Remote Differential Compression** (**RDC**) enabled. DFSR, additionally, provides built-in health-monitoring tools for ease of monitoring deployments, solving one of the biggest headaches with SYSVOL replication in larger Active Directory environments.

When transitioning domain controllers to Windows Server 2022 from previous versions of Windows Server in environments that still use FRS to replicate SYSVOL, you will run into an error stating that you need to migrate to DFSR first.

See also

See the *Raising the domain functional level to Windows Server 2016* recipe from *Chapter 1*, *Optimizing Forests, Domains, and Trusts,* to raise the DFL if it's not Windows Server 2008 yet.

Checking for and remediating lingering objects

A lingering object is an object that was tombstoned and then garbage-collected on all domain controllers but, after that point in time, the object was reintroduced by a domain controller that was restored from a backup, image, or snapshot that was older than the tombstone lifetime period.

You can periodically check for them.

Getting ready

Sign in to the domain controller that holds the PDCE FSMO role, using an account that is a member of the **Domain Admins** group.

Next, find the `objectGUID` attribute of the domain controller. Use the following command on an elevated **Command Prompt** (`cmd.exe`):

```
dsquery.exe * "CN=DC01,OU=Domain
Controllers,DC=LucernPub,DC=com" -scope base -attr objectguid
```

Replace `DC01` with the hostname of the domain controller.

We'll need the `objectGUID` attribute for the next commands.

How to do it...

Use the following command line on an elevated **Command Prompt** (`cmd.exe`) to scan for lingering objects:

```
repadmin.exe /removelingeringobjects <FQDN of Domain Controller
with suspected lingering objects> <objectGUID of Domain
Controller with correct data> <Distinguished Name of partition
containing lingering objects> /advisory_mode
```

Here's an example of this command line for *Lucerne Publishing* to check for lingering objects between `DC01` and `DC02`, where `DC01` is the DC with correct data and denoted by its GUID (`de235686-7bc1-4412-941a4f6e7e248be1`):

```
repadmin.exe /removelingeringobjects DC02.LucernPub.com
de235686-7bc1-4412-941a4f6e7e248be1 DC=LucernPub,DC=com /
advisory_mode
```

Replace `DC02`, `de235686-7bc1-4412-941a4f6e7e248be1`, and `DC=LucernPub,DC=com` with values that represent your environment.

If the environment suffers from one or more lingering objects, events with event ID `1946` will be logged for each lingering object in the **Directory Services** log on the domain controller with the lingering objects. The string of events with event ID `1946` would be marked by an event with event ID `1938`, marking the start of the detection process, and an event with event ID `1942` containing the final detection summary.

Objects detected as lingering objects might be objects that were inadvertently deleted or mangled on a domain controller or objects that are, indeed, lingering objects on a domain controller.

Note

Check with business representatives, such as department managers, to find out whether lingering objects are indeed lingering objects or otherwise mangled objects. Trust me—the last thing you want to do is delete the user account for the **chief financial officer's** (**CFO's**) secretary or any other important person in the organization.

To remove lingering objects detected by `repadmin.exe` in the previous command line, perform the command again, but this time without the `/advisory_mode` switch.

How it works...

A lingering object usually exists on one domain controller and is not affected by Active Directory replication anymore, since all its replication partners have the tombstone change as the last change for the object and have deleted the object from the database.

Lingering objects can be truly nasty. For instance, an Active Directory admin that was fired on the spot a couple of months ago might suddenly have access again after another Active Directory admin may have restored a domain controller in an environment with a tombstone lifetime period of 60 days.

Active Directory admins might not be aware of lingering objects unless they keep an eye out for unexpected Active Directory behavior. They don't need to be sharp-eyed to notice lingering objects, though.

See also

See the *Managing strict replication consistency* recipe to learn how to avoid lingering objects in the future.

6
Managing Active Directory Users

Users and groups are, undeniably, the bread and butter of **Active Directory** (**AD**). When there is something wrong, missing, or absent in these two object types, service desk personnel will be the first to know because colleagues will ring the number for help. On the other hand, when an error is in a colleague's personal interest, due to lingering privileges or absent **identity and access management** (**IAM**) processes, don't expect a call.

It's imperative to get users right. It is estimated that 20% of all **information technology** (**IT**) costs in typical organizations are related to password resets and account lockouts. As colleagues use their accounts for authentication, any hiccup will inevitably result in a loss of productivity.

A best practice is to cooperate with the **Human Resources** (**HR**) department for user creation and user expiration. HR people know when a contract is (to be) terminated, which can help in setting up account expiration. They also know when a person is on maternity leave or taking a sabbatical. We wouldn't want to automatically delete the accounts of these people based on their apparent inactivity, right?

It's also important to get group memberships right. In most organizations, groups govern access to applications and roles and/or privileges within applications. Colleagues without the right privileges might not be productive. Colleagues with too many privileges might inadvertently cause the application to be unavailable, delete data, or have access to data they shouldn't have access to. These confidentiality, integrity, and availability issues can easily be avoided by revoking access through regular access review processes by application owners.

The following recipes are covered in this chapter:

- Creating a user
- Deleting a user
- Modifying several users at once
- Moving a user
- Renaming a user
- Enabling and disabling a user
- Finding locked-out users
- Unlocking a user
- Managing `userAccountControl`
- Using account expiration

Let's jump into the recipes. Many recipes offer a couple of ways to achieve the same goal. While using **Active Directory Users and Computers** (`dsa.msc`) is still commonplace, some administrators have discovered the benefits of using the **Active Directory Administrative Center** (`dsac.exe`) and its **Windows PowerShell History Viewer** to learn the underlying PowerShell cmdlets of their clicks. PowerShell can then be used to automate everything as efficiently as possible; for complex actions you have to perform three times or more, it's better to automate.

Creating a user

Use this recipe to create a `User` object.

Getting ready

To create a user object, sign in to a domain controller or a member server and/or device with the **Remote Server Administration Tools** (**RSAT**) for Active Directory Domain Services installed.

Sign in with an account that is a member of the **Domain Admins** group or the **Account Operators** group or with an account that is delegated to create user objects in the domain or scope of the **organizational unit** (**OU**) in which user objects are to be created.

How to do it...

There are four ways to create a user object:

- Using **Active Directory Users and Computers**
- Using the **Active Directory Administrative Center**
- Using command-line tools
- Using Windows PowerShell

Using Active Directory Users and Computers

To create a user using **Active Directory Users and Computers**, follow these steps:

1. Press Start.
2. Search for **Active Directory Users and Computers** and select it from the search results or run `dsa.msc`. The **Active Directory Users and Computers** window appears.
3. In the left navigation pane, navigate to the OU or container in which you want to create a user object.

4. Perform one of these actions to open the **New Object - User** screen:

    I. From the Taskbar, click the **New User** icon.

    II. Right-click an empty space in the main window and select **New** and then **User** from the menu.

    III. Right-click the OU or container in which you want to create a new user and select **New** and then **User** from the menu.

5. In the **New Object - User** screen, specify values for the following fields:

    - Specify a value for **Full name** by either typing a full name or by filling the **First name** and **Last name** fields.
    - Specify a value for **User logon name** in the top field of the two available fields. This will create a `userPrincipalName` attribute (a combination of the two top fields) and a `sAMAccountName` attribute (a combination of the two bottom fields, referred to as the **Pre-Windows 2000 user log-on name** field in the **user interface** (**UI**)).

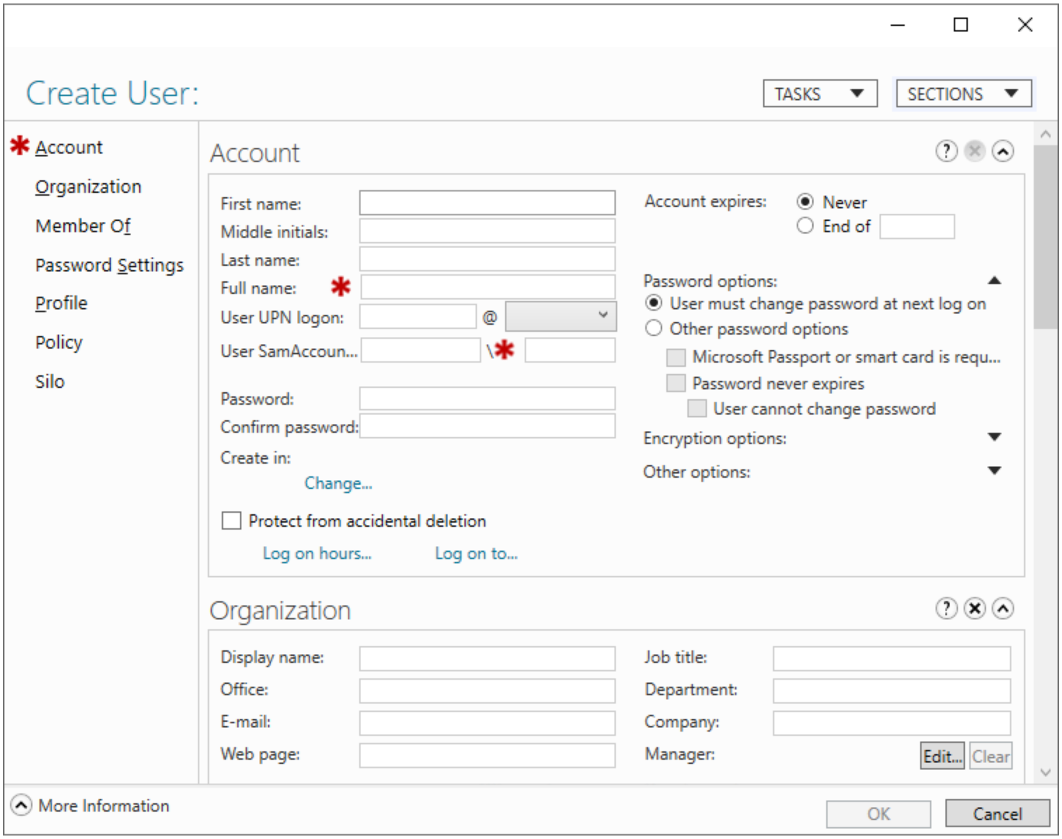
6. Click **Next** >.
7. In the second **New Object - User** screen, specify a **Password** value and then confirm it in the second field. Click **Next** > when done.
8. In the third **New Object - User** screen, click **Finish**.

Using the Active Directory Administrative Center

To create a user using the **Active Directory Administrative Center**, follow these steps:

1. Press Start.
2. Search for **Active Directory Administrative Center** and select it from the search results or run `dsac.exe`. The **Active Directory Administrative Center** window appears.

3. In the left navigation pane, right-click the domain name and select **New** and then **User** from the menu. The **Create User** window appears:



Account expires: ● Never ○ End of

Password options:
● User must change password at next log on
○ Other password options
□ Microsoft Passport or smart card is requ...
□ Password never expires
□ User cannot change password

□ Protect from accidental deletion

Figure 6.1 – Create User window

4. In the **Account** area, specify values for the following fields:

   I. Specify a **Full name** value for the user by either typing a full name or by filling the **First name** and **Last name** fields.

   II. Specify the user's sign-in name in the **User UPN logon** field. This will create a `userPrincipalName` attribute (a combination of the two top fields) and a `sAMAccountName` attribute (a combination of the two bottom fields, referred to as the **User SamAccountName logon** field in the UI).

5. Click **OK**.

Using command-line tools

Use the following command to create a user object in Active Directory:

```
dsadd.exe user 'CN=User,CN=Users,DC=LucernPub,DC=com' -upn
user@lucernpub.com -fn 'User's First Name' -ln 'User's Last
Name' -display 'User' -pwd 'PasswordHere'
```

Replace `DC=LucernPub,DC=com` to represent your environment. Replace the other fields to represent your user's properties.

Using Windows PowerShell

Use the following line of PowerShell code on a system with the Active Directory module for Windows PowerShell installed:

```
New-ADUser -Name User -Path 'CN=Users,DC=LucernPub,DC=com'
-GivenName 'User's First Name' -Surname 'User's Last Name'
-sAMAccountName user
```

Replace `DC=LucernPub,DC=com` to represent your environment. Replace the other fields to represent your user's properties.

How it works...

When you create a user object, some fields are automatically populated and many settings are default settings, including the **User must change password at next logon** setting. These settings are recommended but can be changed after creating a user, either in the UI or through commands.

A descriptive name for every user object is mandatory to ensure the user object can be identified in the graphical interface and in auditing scenarios. Many applications, including Microsoft Exchange Server, use the first, last, and/or full name to create their own default attributes and settings.

When you create a user object, the object consumes a **relative identifier** (**RID**) and **Distinguished Name Tag** (**DNT**). **Pre-Windows 2000 username** is usually the username people in the organization use to sign in, but `userPrincipalName` can also be used in most scenarios. `userPrincipalName` is used when modern authentication protocols are utilized.

There's more...

All organizations create user accounts differently. Configuration steps may vary and may or may not include profile-, Microsoft Exchange Server-, Terminal Server-, or Microsoft 365-specific steps. When additional steps are required, a work instruction is usually present.

Deleting a user

Use this recipe to delete a previously created user object.

Getting ready

To delete a user account, sign in to a domain controller or a member server and/or device with RSAT for Active Directory Domain Services installed.

Sign in with an account that is a member of the **Domain Admins** group or the **Account Operators** group or with an account that is delegated to delete user objects in the domain or scope of the OU where the user account is to be deleted.

How to do it...

There are four ways to delete a user object:

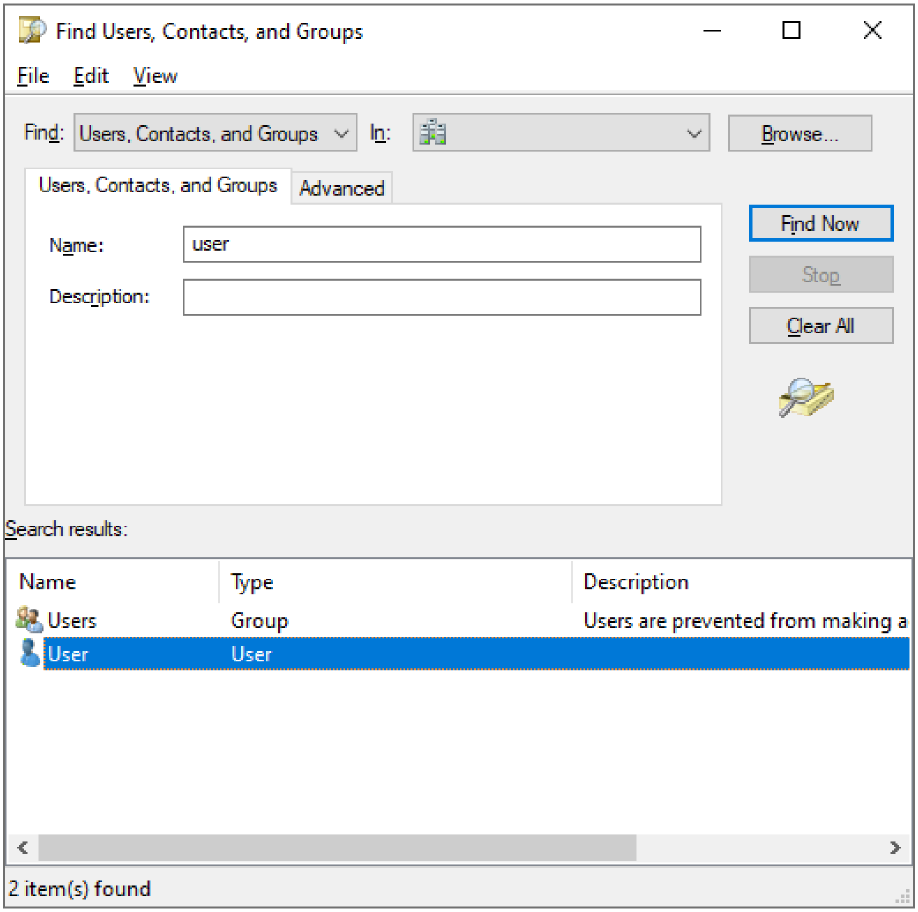
1. Using **Active Directory Users and Computers**
2. Using the **Active Directory Administrative Center**
3. Using command-line tools
4. Using Windows PowerShell

Using Active Directory Users and Computers

To delete a user using **Active Directory Users and Computers**, follow these steps:

1. Press Start.
2. Search for **Active Directory Users and Computers** and select it from the search results or run `dsa.msc`. The **Active Directory Users and Computers** window appears.
3. In the **View** menu, enable **Advanced Features**.
4. In the left navigation pane, select a domain or a specific OU.
5. From the **Action** menu, select **Find…**. The **Find Users, Contacts, and Groups** window appears.

6. In the **Name** field, type the name of the user object you intend to delete and press *Enter*. From the **Search results** list, select the user object:



Find: Users, Contacts, and Groups In:

Name: user

Description:

Search results:

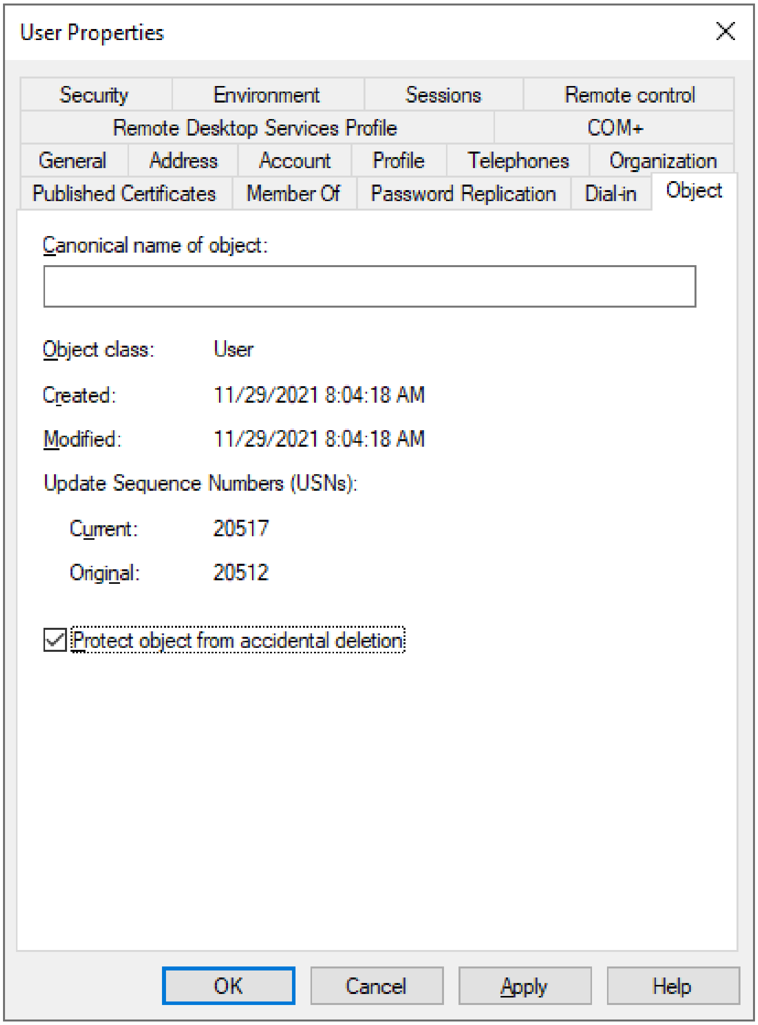
Name	Type	Description
Users	Group	Users are prevented from making a
User	User	

2 item(s) found

Figure 6.2 – Selecting the user object from the Search results list

7. Right-click the user object and select **Properties** from the menu.

8. Navigate to the **Object** tab:



Canonical name of object:

Object class: User

Created: 11/29/2021 8:04:18 AM

Modified: 11/29/2021 8:04:18 AM

Update Sequence Numbers (USNs):

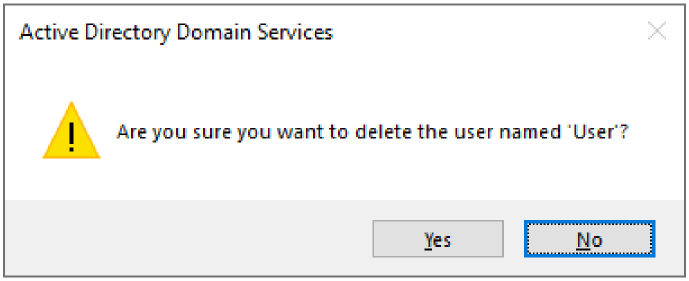
Current: 20517

Original: 20512

☑ Protect object from accidental deletion

Figure 6.3 – Object tab of a user's properties

9. Disable the **Protect object from accidental deletion** option if it is enabled.
10. Click **OK** to close the properties window for the object.
11. Right-click the user object again. This time, select **Delete** from the menu. The **Active Directory Domain Services** pop-up window appears:



Active Directory Domain Services

Are you sure you want to delete the user named 'User'?

Figure 6.4 – Active Directory Domain Services pop-up window

12. Click **Yes** in the **Active Directory Domain Services** pop-up window.

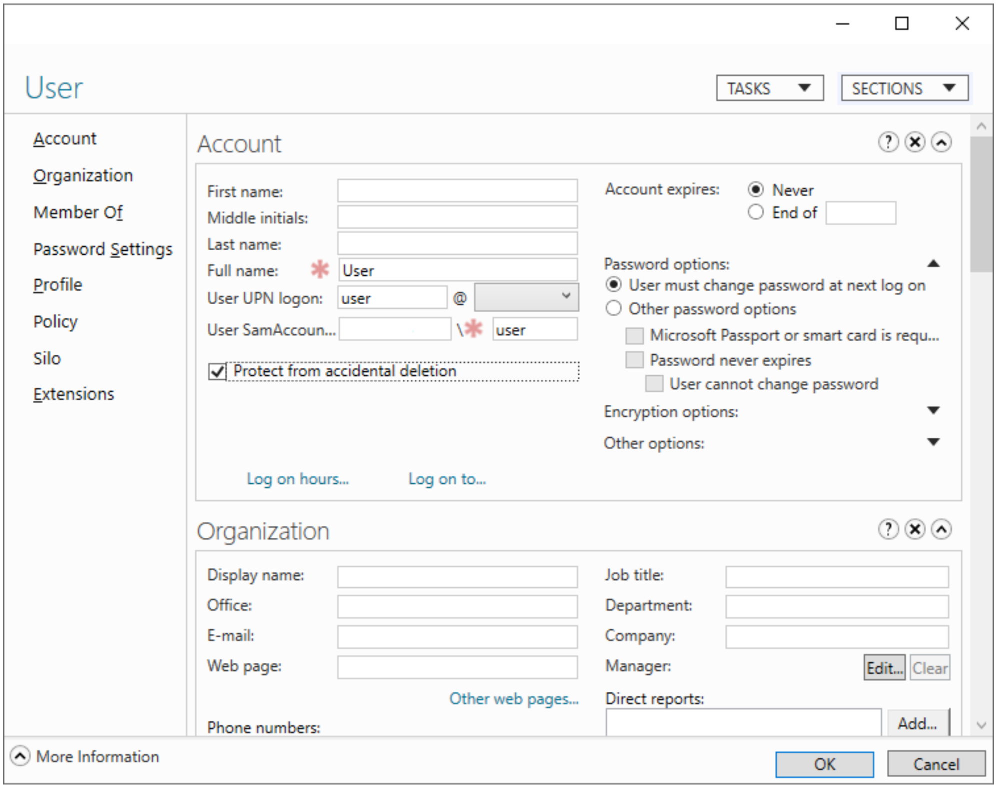
Using the Active Directory Administrative Center

To delete a user using the **Active Directory Administrative Center**, follow these steps:

1. Press Start.
2. Search for **Active Directory Administrative Center** and select it from the search results or run `dsac.exe`. The **Active Directory Administrative Center** window appears.
3. Perform one of these series of actions:

   - In the left navigation pane, switch to the tree view. Navigate to the OU or container where the user object that you intend to delete resides. In the main pane, select the user object.
   - From the main pane menu, under **Global Search**, type the name of the user object you intend to delete and press *Enter*. From the list of **Global Search** results, select the user object.

4. Right-click the user object and select **Properties** from the menu. The user object's properties window appears:



First name:
Middle initials:
Last name:
Full name: * User
User UPN logon: user @
User SamAccoun... * user
☑ Protect from accidental deletion

Account expires: ● Never ○ End of

Password options:
● User must change password at next log on
○ Other password options
□ Microsoft Passport or smart card is requ...
□ Password never expires
□ User cannot change password

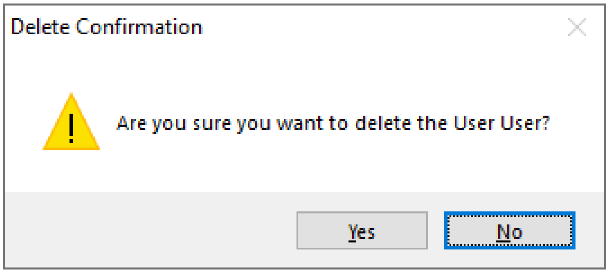
Encryption options:
Other options:

Display name:
Office:
E-mail:
Web page:
Phone numbers:
Job title:
Department:
Company:
Manager:
Direct reports:

Figure 6.5 – User object's properties window

5. Disable the **Protect object from accidental deletion** option if it is enabled.

6. Click the **TASKS** button at the top of the window. Select **Delete** from the menu. The **Delete Confirmation** pop-up window appears:



Are you sure you want to delete the User User?

Figure 6.6 – Delete Confirmation pop-up window

7. Click **Yes** in the **Delete Confirmation** pop-up window.

Using command-line tools

Use the following command to delete a user object in Active Directory:

```
dsrm.exe user 'CN=User,CN=Users,DC=LucernPub,DC=com'
```

Replace `DC=LucernPub,DC=com` to represent your environment. Replace the other fields in the **distinguished name** (**DN**) of the user object to represent your user's properties.

Using Windows PowerShell

Use the following line of PowerShell code on a system with the Active Directory module for Windows PowerShell installed to delete a user object in Active Directory:

```
Remove-ADUser -Identity 'CN=User,CN=Users,DC=LucernPub,DC=com'
```

Replace `DC=LucernPub,DC=com` to represent your environment. Replace the other fields in the DN of the user object to represent your user's properties. Alternatively, you can specify the `objectGUID`, `objectSid`, or `sAMAccountName` attribute to represent the user.

How it works...

When you delete a user object, the object no longer uses its RID, but the RID and the corresponding **security identifier** (**SID**) and DNT in the domain partition cannot be reused.

When you attempt to delete a user object that has the **Protect from accidental deletion** option enabled, you will not be able to delete the object. The option needs to be disabled before this can be done.

Many organizations are wary of deleting user objects because they fear their auditing systems may no longer be able to put a name to the RID or corresponding SID. Instead, most of them opt to disable user objects. Unfortunately, many admins forget to actually delete user objects beyond the auditing retention period, getting stuck with numerous objects that take up space in the Active Directory database, making Active Directory more complex to manage.

When the **Active Directory Recycle Bin** is enabled, the deleted user object emerges in the **Deleted Objects** container.

See also

Refer to the following recipes for more information:

- The *Enabling the Active Directory Recycle Bin* recipe from *Chapter 1*, *Optimizing Forests, Domains, and Trusts*
- The *Enabling and disabling a user* recipe from this chapter

Modifying several users at once

Once a few user objects have been created, a need might arise to modify one attribute for all previously created user objects. Other scopes of user objects might also apply.

Modifying one user object is as simple as double-clicking on it in **Active Directory Users and Computers** or in the **Active Directory Administrative Center**. Modifying multiple objects at once is slightly different, and there are a couple of neat tricks you can use.

Getting ready

To modify user objects, sign in to a domain controller or a member server and/or device with RSAT for Active Directory Domain Services installed.

Sign in with an account that is a member of the **Domain Admins** group or the **Account Operators** group or with an account that is delegated to manage user objects in the domain or scope of the OU where the user objects reside.

How to do it...

There are three ways to modify user objects:

1. Using **Active Directory Users and Computers**
2. Using the **Active Directory Administrative Center**
3. Using PowerShell

Using Active Directory Users and Computers

Active Directory Users and Computers is ideal when you have simple scopes for user objects to modify. Through its selection mechanisms, admins can easily select all users in an OU or container (by pressing *Ctrl* + *A*) or select all users whose names start with *A* manually with the *Shift* button.

To modify multiple users using **Active Directory Users and Computers**, follow these steps:

1. Press Start.
2. Search for **Active Directory Users and Computers** and select it from the search results or run `dsa.msc`. The **Active Directory Users and Computers** window appears.
3. In the left navigation pane, navigate to the OU or container where the user objects reside.
4. In the main pane, select user objects while holding down the *Shift* button.
5. Right-click the objects and select **Properties** from the menu.
6. Change the attribute or attributes you want to modify.
7. Click **OK** when done.

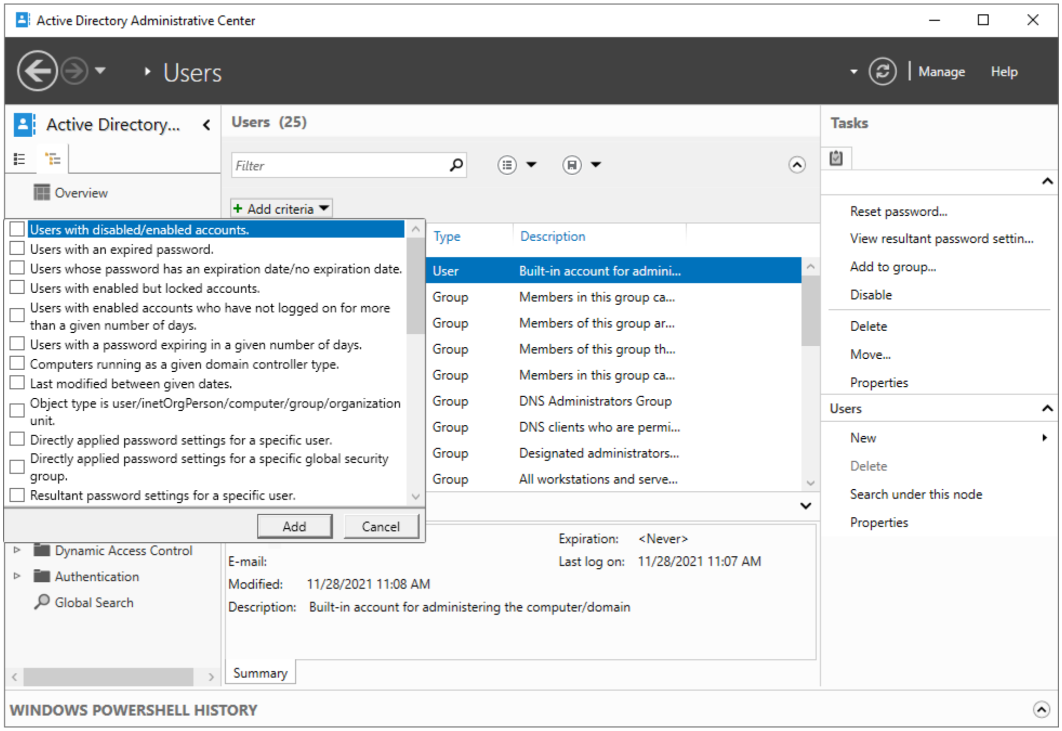
Using the Active Directory Administrative Center

The **Active Directory Administrative Center** allows for filters to scope user objects that you want to modify. Filters can be used on OUs and containers. Of course, you can use the same method for selecting users as you would in **Active Directory Users and Computers**.

To modify a selection of users using the **Active Directory Administrative Center**, follow these steps:

1. Press Start.
2. Search for **Active Directory Administrative Center** and select it from the search results or run `dsac.exe`. The **Active Directory Administrative Center** window appears.
3. In the left navigation pane, switch to the tree view.
4. Select an OU or container to use as the base scope for the filter.
5. In the main pane, expand the top bar.

6. Click the + **Add criteria** button:



□ Users with disabled/enabled accounts.
□ Users with an expired password.
□ Users whose password has an expiration date/no expiration date.
□ Users with enabled but locked accounts.
□ Users with enabled accounts who have not logged on for more than a given number of days.
□ Users with a password expiring in a given number of days.
□ Computers running as a given domain controller type.
□ Last modified between given dates.
□ Object type is user/inetOrgPerson/computer/group/organization unit.
□ Directly applied password settings for a specific user.
□ Directly applied password settings for a specific global security group.
□ Resultant password settings for a specific user.

Type	Description
User	Built-in account for admini...
Group	Members in this group ca...
Group	Members of this group ar...
Group	Members of this group th...
Group	Members in this group ca...
Group	DNS Administrators Group
Group	DNS clients who are permi...
Group	Designated administrators...
Group	All workstations and serve...

E-mail:
Modified: 11/28/2021 11:08 AM
Description: Built-in account for administering the computer/domain
Expiration: <Never>
Last log on: 11/28/2021 11:07 AM

Figure 6.7 – Filters in the Active Directory Administrative Center

7. Add criteria to select user objects you want to modify at once. Use one or more of the built-in filters or scroll down to create a filter based on the user-friendly names of one or more attributes. Click **Add** to add a filter.

> **Note**
>
> When filtering using attributes, you can use matches such as **starts with**, **equals**, **does not equal**, **is empty**, and **is not empty**.

8. In the **Search results** pane, select all user objects that match the filter, by selecting one and then pressing *Ctrl* + *A*.

9. In the right task pane, click **Properties**. The **Multiple Users** window appears:

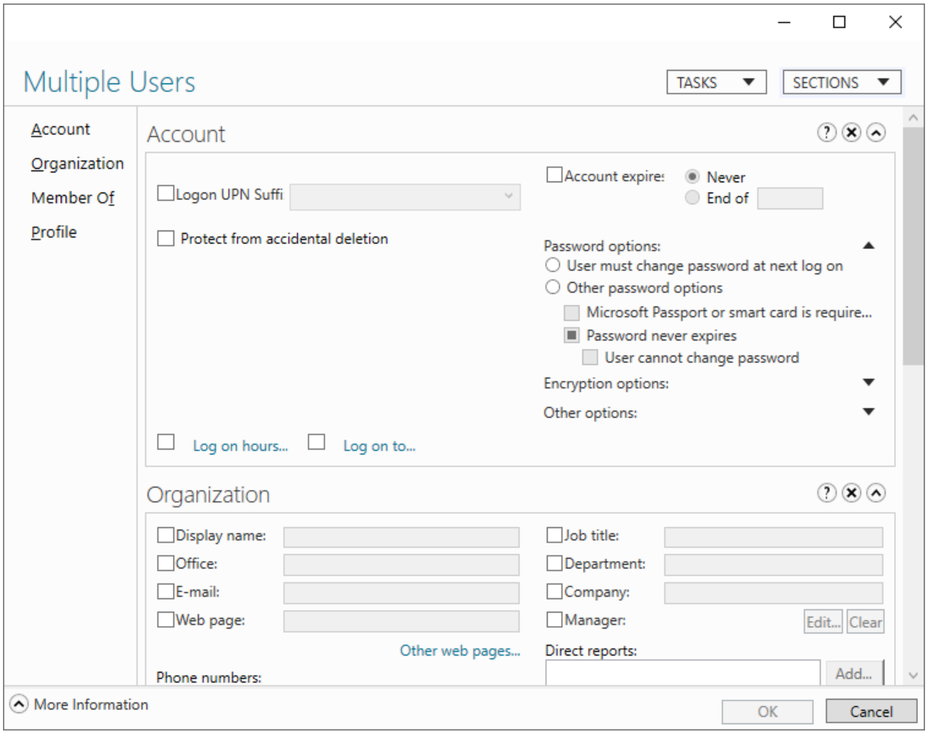


Figure 6.8 – Multiple Users window

10. Change the attribute or attributes you want to modify.

11. Click **OK** when done.

Using Windows PowerShell

The Active Directory module for Windows PowerShell can be used to select multiple user objects. Modifications can then be applied to the scope of users using the piping mechanism in PowerShell.

As an example, use the following lines of PowerShell code on a system with the Active Directory module for Windows PowerShell installed to modify **Prevent from accidental deletion** for all user objects whose `sAMAccountName` attribute starts with `service_`:

```
Get-ADUser -ldapfilter '(sAMAccountName=service_*)' |
Set-ADObject -ProtectedFromAccidentalDeletion $true
```

Since scoping in PowerShell is more advanced, this method is best used for modifying multiple user objects throughout the Active Directory forest and to modify user objects repeatedly.

How it works...

Using filters, several user objects can be modified at once.

Not all attributes can be changed when multiple user objects are selected using **Active Directory Users and Computers** or the **Active Directory Administrative Center**. Typical attributes include those that are unique to a user object, such as `userPrincipalName`, `sAMAccountName`, and `securityIdentifier`.

There's more...

When using the **Active Directory Administrative Center** (`dsac.exe`), the **Windows PowerShell History** feature can be used to find modifications on one user object from the UI in PowerShell. This way, modifications behind | can be found easily. To enable the **Windows PowerShell History** pane, perform these actions:

1. Press Start.
2. Search for **Active Directory Administrative Center** and select it from the search results or run `dsac.exe`. The **Active Directory Administrative Center** window appears.
3. In the bottom bar, expand the **Windows PowerShell History** bar.

Moving a user

After creating a user, you might find that it should live in another OU or container. This recipe describes how to move a user object.

Getting ready

To move a user account, sign in to a domain controller or a member server and/or device with RSAT for Active Directory Domain Services installed.

Sign in with an account that is a member of the **Domain Admins** group or the **Account Operators** group or with an account that is delegated to create and delete user objects in the domain or scope of the OUs where user objects are to be moved.

How to do it...

There are four ways to move a user object:

1. Using **Active Directory Users and Computers**
2. Using the **Active Directory Administrative Center**
3. Using command-line tools
4. Using Windows PowerShell

Using Active Directory Users and Computers

To move a user using **Active Directory Users and Computers**, follow these steps:

1. Press Start.
2. Search for **Active Directory Users and Computers** and select it from the search results or run `dsa.msc`. The **Active Directory Users and Computers** window appears.
3. In the **View** menu, enable **Advanced Features**.
4. Perform one of these series of actions:

    - In the left navigation pane, navigate to the OU or container where the user object that you intend to move resides. In the main pane, right-click the user object and select **Properties** from the menu. The user object's properties window appears.
    - From the **Action** menu, select **Find…**. The **Find Users, Contacts, and Groups** window appears. In the **Name** field, type the name of the user object you intend to move and press *Enter*. From the **Search results** list, select the user object. Right-click the user object and select **Properties** from the menu. The user object's properties window appears.

5. Navigate to the **Object** tab.
6. Disable the **Protect object from accidental deletion** option if it is enabled.
7. Click **OK** to close the properties window for the user object.

8. Right-click the user object again. This time, select **Move...** from the menu. The **Move** pop-up window appears:

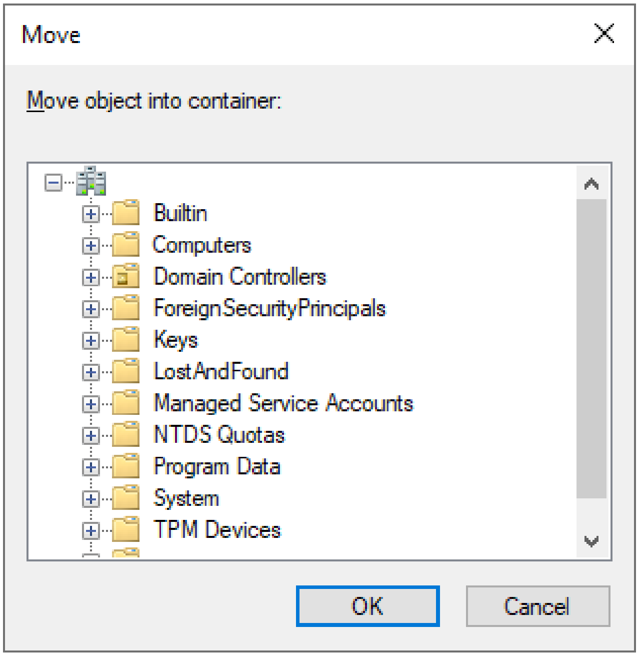


Figure 6.9 – Move pop-up window in Active Directory Users and Computers

9. In the **Move** window, navigate to the OU or container where you want to move the user object to, and select it.
10. Click **OK** to move the user.

If this is an important user object, re-enable the **Protect object from accidental deletion** option.

Using the Active Directory Administrative Center

To move a user using the **Active Directory Administrative Center**, follow these steps:

1. Press Start.
2. Search for **Active Directory Administrative Center** and select it from the search results or run `dsac.exe`. An **Active Directory Administrative Center** window appears.

3. Perform one of these series of actions:

    - In the left navigation pane, switch to the tree view. Navigate to the OU or container where the user object that you intend to move resides. In the main pane, select the user object.
    - From the main pane menu, under **Global Search**, type the name of the user object you intend to move and press *Enter*. From the list of **Global Search** results, select the user object.

4. Right-click the user object and select **Properties** from the menu. The user object's properties window appears.
5. Disable the **Protect object from accidental deletion** option if it is enabled.
6. Click the **TASKS** button at the top of the window. Select **Move…** from the menu. The **Move Confirmation** pop-up window appears, followed by a **Move** pop-up window:

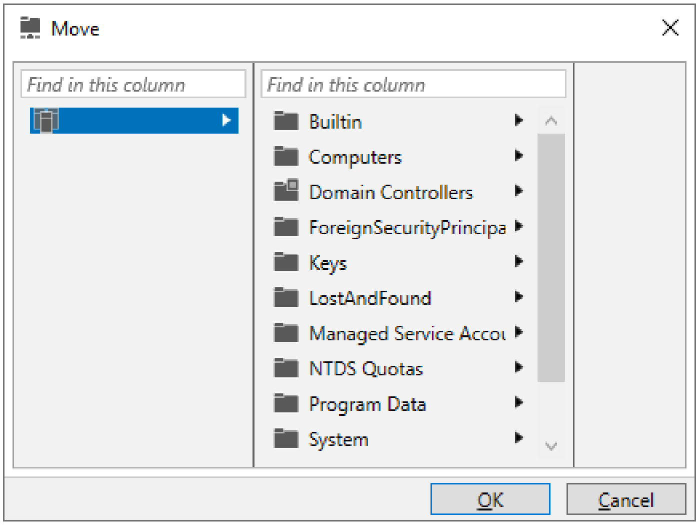


Figure 6.10 – Move pop-up window in the Active Directory Administrative Center

7. In the **Move** window, navigate to the OU or container where you want to move the user object to, and select it.
8. Click **OK** to move the object.

Using command-line tools

Use the following command to move a user object in Active Directory:

```
dsmove.exe 'CN=User,CN=Users,DC=LucernPub,DC=com' -newparent
'OU=Organizational Unit,DC=LucernPub,DC=com'
```

Replace `DC=LucernPub,DC=com` to represent your environment. Replace the other fields in the DN of the user object and a new location to represent your scenario.

Using Windows PowerShell

Use the following line of PowerShell code on a system with the Active Directory module for Windows PowerShell installed:

```
Move-ADObject -Identity:'CN=User,CN=Users,DC=LucernPub,DC=com'
-TargetPath:'OU=Organizational Unit,DC=LucernPub,DC=com'
```

Replace `DC=LucernPub,DC=com` to represent your environment. Replace the other fields in the DN of the user object and a new location to represent your scenario.

How it works...

When you move a user object in Active Directory, it will effectively fall out of the scope of the parent OU or container. Therefore, the user object that is used for this purpose must have (delegated) permissions to delete user objects in the original location of the user object. Also, the user object cannot have the **Protect from accidental deletion** option enabled. This option needs to be disabled to avoid permissions errors.

Renaming a user

Use this recipe to rename a previously created user object.

Getting ready

To rename a user object, sign in to a domain controller or a member server and/or device with RSAT for Active Directory Domain Services installed.

Sign in with an account that is a member of the **Domain Admins** group or the **Account Operators** group or with an account that is delegated to modify user objects in the domain or scope of the OU where the user object resides.

How to do it...

There are four ways to rename a user object:

1. Using **Active Directory Users and Computers**
2. Using the **Active Directory Administrative Center**
3. Using command-line tools
4. Using Windows PowerShell

Using Active Directory Users and Computers

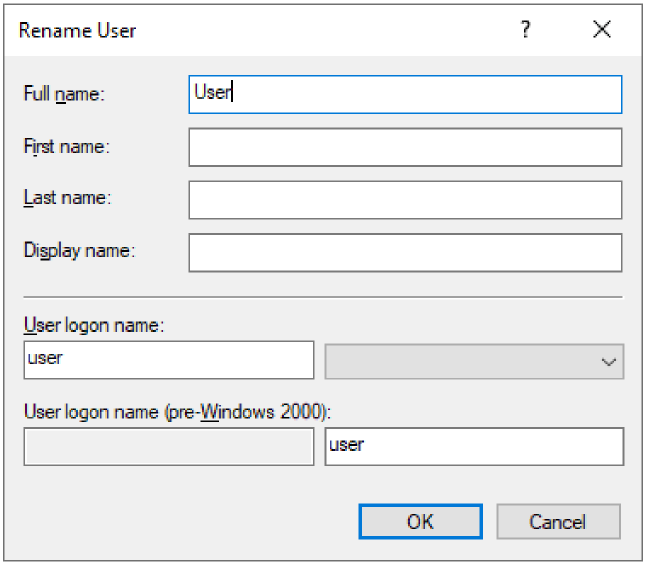
To rename a user using **Active Directory Users and Computers**, follow these steps:

1. Press Start.
2. Search for **Active Directory Users and Computers** and select it from the search results or run `dsa.msc`. The **Active Directory Users and Computers** window appears.
3. Perform one of these actions:

   - In the left navigation pane, navigate to the OU or container where the user object resides. In the main pane, select the user object.
   - From the **Action** menu, select **Find…**. The **Find Users, Contacts, and Groups** window appears. In the **Name** field, type the name of the user object you intend to rename, and press *Enter*. From the **Search results** list, select the user object.

4. Right-click the user object and select **Rename** from the menu.
5. Type a new name for the user object and press *Enter*. The **Rename User** pop-up window appears.

6. In the **Rename User** pop-up window, type new values for other attributes that you want subsequently renamed:



Rename User

Full name: User

First name:

Last name:

Display name:

User logon name: user

User logon name (pre-Windows 2000): user

Figure 6.11 – Renaming a user in Active Directory Users and Computers

7. Click **OK**.

Using the Active Directory Administrative Center

To rename a user using the **Active Directory Administrative Center**, follow these steps:

1. Press Start.
2. Search for **Active Directory Administrative Center** and select it from the search results or run `dsac.exe`. The **Active Directory Administrative Center** window appears.
3. Perform one of these series of actions:

   - In the left navigation pane, switch to the tree view. Navigate to the OU or container where the user object that you intend to move resides. In the main pane, select the user object.
   - From the main pane menu, under **Global Search**, type the name of the user object you intend to move and press *Enter*. From the list of **Global Search** results, select the user object.

4. Right-click the user object and select **Properties** from the menu. The user object's properties window appears.
5. Change one or more of the following fields for the user object:
   - In the **Account** area:
   - **First name**
   - **Last name**
   - **Full name**
   - **User UPN logon**
   - **User SamAccountName logon**
   - In the **Organization** area, change the **Display Name** field.
6. Click **OK**.

Using command-line tools

Use the following command to rename a user object in Active Directory:

```
dsmove.exe 'CN=User,CN=Users,DC=LucernPub,DC=com' -NewName
'User Account'
```

Replace `DC=LucernPub,DC=com` to represent your environment. Replace the other fields in the DN of the user object to represent your scenario.

Using Windows PowerShell

Use the following lines of PowerShell code on a system with the Active Directory module for Windows PowerShell installed:

```
Rename-ADObject -Identity
'CN=User,CN=Users,DC=LucernPub,DC=com' -NewName 'User Account'
```

Replace `DC=LucernPub,DC=com` to represent your environment. Replace the other fields in the DN of the user object to represent your scenario.

How it works...

When you rename a user object, you change its **Canonical Name** (**CN**), its DN, and its `name` attributes. When you perform this action using **Active Directory Users and Computers** (`dsa.msc`), a helpful popup allows you to change any other name-related attributes.

Even though you use the `dsmove.exe` command, when you rename a user object on the command line, the **Protect from accidental deletion** option does not come into effect like it does when you move a user object. You will not need to disable this option first to rename a user object.

Enabling and disabling a user

When you want a person in your organization to no longer be able to sign in interactively, you can disable the corresponding user object. Likewise, when a user object is disabled, you can opt to enable/re-enable it. Use this recipe to perform both actions.

Getting ready

To enable or disable a user object, sign in to a domain controller or a member server and/or device with RSAT for Active Directory Domain Services installed.

Sign in with an account that is a member of the **Domain Admins** group or the **Account Operators** group or with an account that is delegated to modify user objects in the domain or scope of the OU where the user object resides.

How to do it...

There are four ways to enable or disable a user object:

1. Using **Active Directory Users and Computers**
2. Using the **Active Directory Administrative Center**
3. Using command-line tools
4. Using Windows PowerShell

Using Active Directory Users and Computers

To enable or disable a user using **Active Directory Users and Computers**, follow these steps:

1. Press Start.
2. Search for **Active Directory Users and Computers** and select it from the search results or run `dsa.msc`. The **Active Directory Users and Computers** window appears.

3. Perform one of these actions:

    - In the left navigation pane, navigate to the OU or container where the user object resides. In the main pane, select the user object.
    - From the **Action** menu, select **Find…**. The **Find Users, Contacts, and Groups** window appears. In the **Name** field, type the name of the user object you intend to enable or disable, and press *Enter*. From the **Search results** list, select the user object.

4. Right-click the user object and select **Enable account** or **Disable account** from the menu. The **Active Directory Domain Services** pop-up window appears.
5. In the **Active Directory Domain Services** pop-up window, dismiss the **User Object has been disabled** or **User Object has been enabled** message by clicking **OK**.

Using the Active Directory Administrative Center

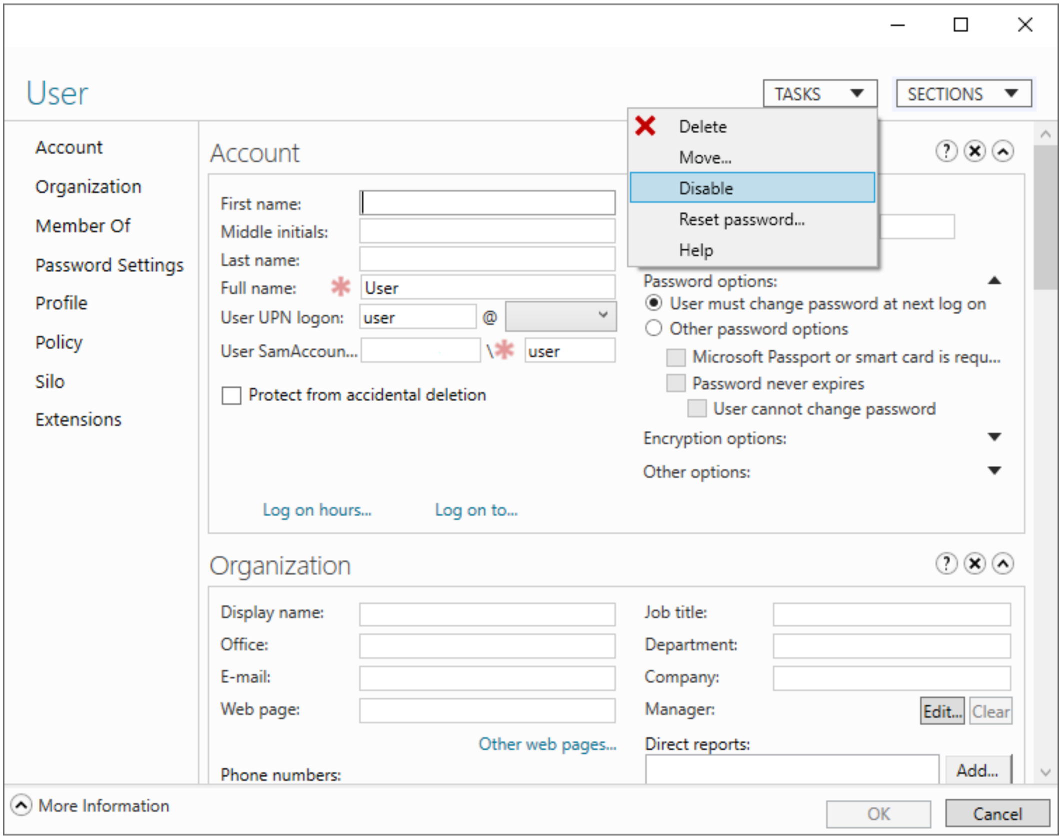
To enable or disable a user using the **Active Directory Administrative Center**, follow these steps:

1. Press Start.
2. Search for **Active Directory Administrative Center** and select it from the search results or run `dsac.exe`. The **Active Directory Administrative Center** window appears.
3. Perform one of these series of actions:

    - In the left navigation pane, switch to the tree view. Navigate to the OU or container where the user object that you intend to enable or disable resides. In the main pane, select the user object.
    - From the main pane menu, under **Global Search**, type the name of the user object you intend to move and press *Enter*. From the list of **Global Search** results, select the user object.

4. Right-click the user object and select **Properties** from the menu. The user object's properties window appears.
5. In the top bar of the user object's properties window, click the **TASKS** button.

6. From the menu, select **Disable** or **Enable**:



First name:

Middle initials:

Last name:

Full name: User

User UPN logon: user @

User SamAccoun... \ user

□ Protect from accidental deletion

Password options:

● User must change password at next log on

○ Other password options

□ Microsoft Passport or smart card is requ...

□ Password never expires

□ User cannot change password

Display name:

Office:

E-mail:

Web page:

Job title:

Department:

Company:

Manager:

Direct reports:

Phone numbers:

Figure 6.12 – Disabling a user in the Active Directory Administrative Center

7. Click **OK**.

Using command-line tools

Use the following command to disable a user object in Active Directory:

```
dsmod.exe user 'CN=User,CN=Users,DC=LucernPub,DC=com' -disabled
yes
```

Replace `DC=LucernPub,DC=com` to represent your environment. Replace the other fields in the DN of the user object to represent your scenario.

Use the following command to (re-)enable a user object in Active Directory:

```
dsmod.exe 'CN=User,CN=Users,DC=LucernPub,DC=com' -disabled no
```

Replace `DC=LucernPub,DC=com` to represent your environment. Replace the other fields in the DN of the user object to represent your scenario.

Using Windows PowerShell

Use the following line of PowerShell code on a system with the Active Directory module for Windows PowerShell installed to disable a user object:

```
Disable-ADAccount -Identity
'CN=User,CN=Users,DC=LucernPub,DC=com'
```

Replace `DC=LucernPub,DC=com` to represent your environment. Replace the other fields in the DN of the user object to represent your scenario.

Use the following line of PowerShell code on a system with the Active Directory module for Windows PowerShell installed to (re-)enable a user object:

```
Enable-ADAccount -Identity
'CN=User,CN=Users,DC=LucernPub,DC=com'
```

Replace `DC=LucernPub,DC=com` to represent your environment. Replace the other fields in the DN of the user object to represent your scenario.

How it works...

When a user object is disabled, it can no longer be used to sign in.

When an account is enabled, you can use the right-click menu in **Active Directory Users and Computers** or use the **TASKS** menu in the user object's properties in the **Active Directory Administrative Center** to disable it. When it's disabled, you can enable it in the same menus. The two menu options are not available at the same time.

There's more...

Even though interactive sign-ins are disabled when a user object is disabled, they can still be used for **Kerberos constrained delegation** (**KCD**) and other delegation scenarios that don't check whether a user object is disabled.

Finding locked-out users

User accounts may get locked out. In this recipe, we will see how to find locked-out accounts.

Getting ready

To find locked-out user accounts, sign in to a domain controller or a member server and/or device with RSAT for Active Directory Domain Services installed.

By default, any user object in Active Directory can be used to find locked-out accounts, as this kind of information is available to the **Everyone** group.

How to do it...

There are two ways to find locked-out users:

1. Using the **Active Directory Administrative Center**
2. Using Windows PowerShell

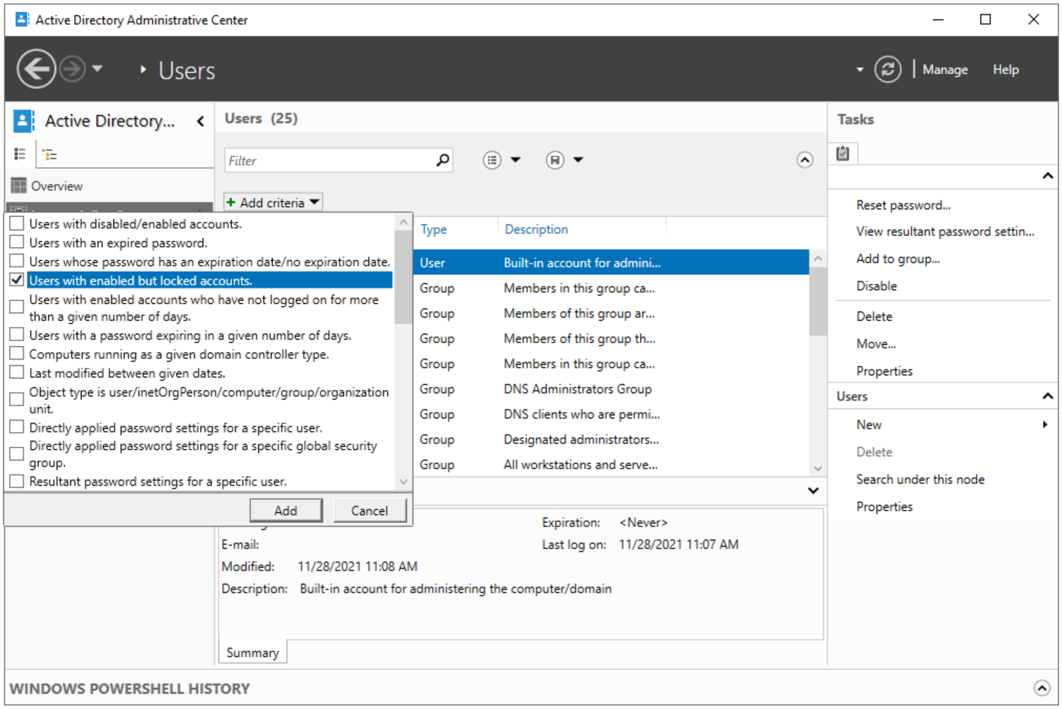
Using the Active Directory Administrative Center

The **Active Directory Administrative Center** allows for filtering locked-out user objects. Filters can be used on OUs and containers.

To find currently locked-out user objects using the **Active Directory Administrative Center**, follow these steps:

1. Press Start.
2. Search for **Active Directory Administrative Center** and select it from the search results or run `dsac.exe`. The **Active Directory Administrative Center** window appears.

3. In the left navigation pane, select an OU or container to use as the base scope for the filter.
4. In the main pane, expand the top bar.
5. Click the + **Add criteria** button.
6. Select the **Users with enabled but locked accounts** criteria:



□ Users with disabled/enabled accounts.
□ Users with an expired password.
□ Users whose password has an expiration date/no expiration date.
☑ Users with enabled but locked accounts.
□ Users with enabled accounts who have not logged on for more than a given number of days.
□ Users with a password expiring in a given number of days.
□ Computers running as a given domain controller type.
□ Last modified between given dates.
□ Object type is user/inetOrgPerson/computer/group/organization unit.
□ Directly applied password settings for a specific user.
□ Directly applied password settings for a specific global security group.
□ Resultant password settings for a specific user.

Type	Description
User	Built-in account for admini...
Group	Members in this group ca...
Group	Members of this group ar...
Group	Members of this group th...
Group	Members in this group ca...
Group	DNS Administrators Group
Group	DNS clients who are permi...
Group	Designated administrators...
Group	All workstations and serve...

Expiration: <Never>
E-mail:
Last log on: 11/28/2021 11:07 AM
Modified: 11/28/2021 11:08 AM
Description: Built-in account for administering the computer/domain

Figure 6.13 – Selecting the Users with enabled but locked accounts criteria

7. Click **Add** to add the filter.
8. The locked-out accounts are displayed in the main pane.

Using Windows PowerShell

Use the following line of PowerShell code on a system with the Active Directory module for Windows PowerShell installed to find locked-out user accounts:

```
Search-ADAccount -LockedOut -UsersOnly | Format-Table
Name,LockedOut -AutoSize
```

Any locked users can be manually unlocked or unlocked using a piping mechanism, as shown in the next recipe.

How it works...

The Default Domain Policy contains password and account lockout policies. By default, the password part of these policies is enabled, but the lockout part is not. For some organizations, this makes sense. Other organizations might consider enabling account lockout, as it prevents brute-force password attacks; when a malicious person uses the maximum number of passwords within a certain time for a user account, the user account is locked out for a specified time. All three metrics can be set in the lockout part of password and account lockout policies.

For example, when an attacker or a clumsy colleague hits a maximum of five wrong passwords attempts in 2 minutes, the user object is locked out for 30 minutes.

From Windows Server 2008 Active Directory onward, fine-grained password and account lockout policies can be used to set the metrics for a specific user object or specific groups.

See also

To unlock the user object(s) found in this recipe, see the *Unlocking a user* recipe.

Unlocking a user

In many organizations with account lock-out policies enabled, the number of passwords that can be mistyped is set to a high value (such as 50), and then the lock-out period is set to indefinitely. This way, a locked-out account hinders the productivity of colleagues since they can't sign in until the account is unlocked. Manual unlocks need to be performed to enable affected colleagues to sign in again.

Getting ready

To modify user objects, sign in to a domain controller or a member server and/or device with RSAT for Active Directory Domain Services installed.

Sign in with an account that is a member of the **Domain Admins** group, the **Account Operators** group, or with an account that is delegated to manage user objects in the domain or scope of the OU where the user object resides.

How to do it...

This recipe shows two ways to unlock a user object:

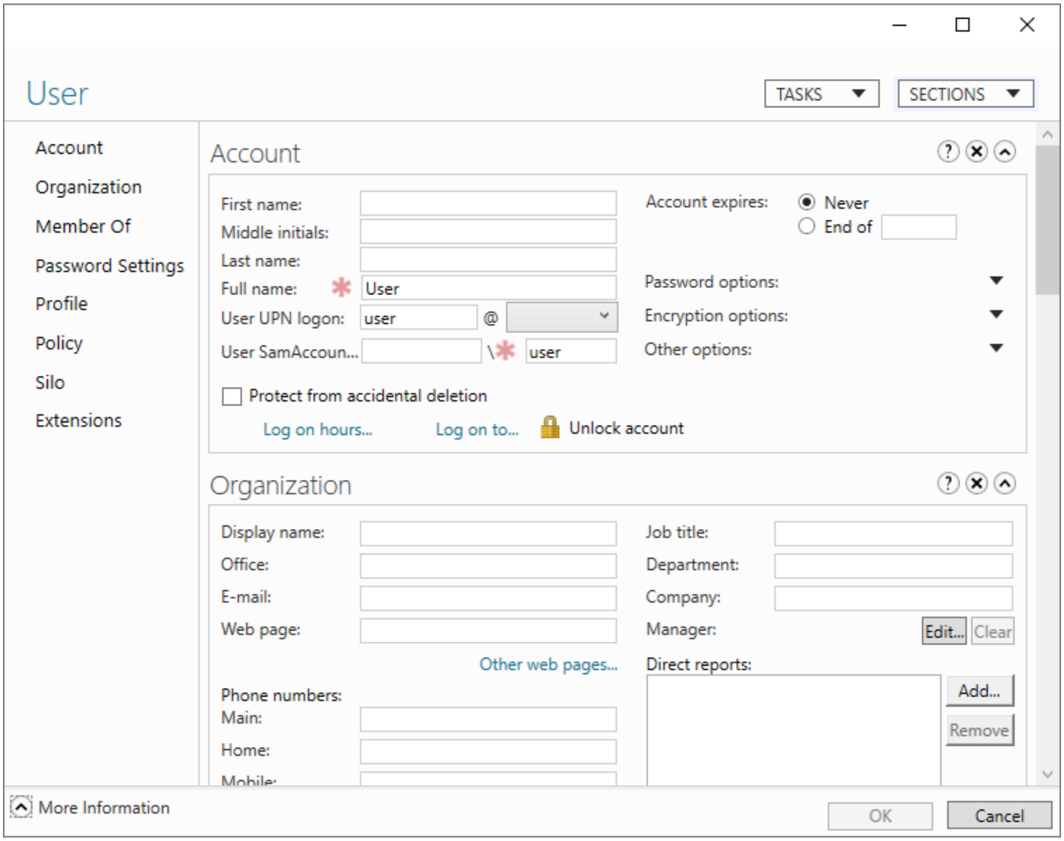
1. Using the **Active Directory Administrative Center**
2. Using Windows PowerShell

Using the Active Directory Administrative Center

To unlock a user object using the **Active Directory Administrative Center**, follow these steps:

1. Press Start.
2. Search for **Active Directory Administrative Center** and select it from the search results or run `dsac.exe`. The **Active Directory Administrative Center** window appears.
3. In the left navigation pane, navigate to the OU or container where the user objects reside.
4. In the main pane, select the user object you want to unlock.
5. Right-click the object and select **Properties** from the menu. The **User** window appears.

6. Click **Unlock account**:



User

Account

First name:
Middle initials:
Last name:
Full name: User
User UPN logon: user @
User SamAccoun... \ user

Account expires: ● Never ○ End of

□ Protect from accidental deletion

Organization

Display name:
Office:
E-mail:
Web page:
Phone numbers:
Main:
Home:
Mobile:

Job title:
Department:
Company:
Manager:
Direct reports:

Figure 6.14 – Unlock account option in the Active Directory Administrative Center

7. Click **OK**.

To unlock all locked-out accounts in a certain OU or container, combine the steps from the *Finding locked-out accounts* recipe with steps 5 through 7.

Using Windows PowerShell

Use the following line of PowerShell code on a system with the Active Directory module for Windows PowerShell installed to unlock a user object:

```
Unlock-ADAccount -Identity
'CN=User,CN=Users,DC=LucernPub,DC=com'
```

Replace `DC=LucernPub,DC=com` to represent your environment. Replace the other fields in the DN of the user object to represent your scenario.

To unlock all locked-out accounts throughout the Active Directory domain, combine the commands from the *Finding locked-out users* recipe with the preceding lines of code in the following way:

```
Search-ADAccount -LockedOut -UsersOnly | Unlock-ADAccount
```

The preceding line of PowerShell code is specifically useful in situations where accounts are locked out after a brute-force password guessing attack, once the initial attack vector has been neutralized.

Managing userAccountControl

Many user objects' settings can be controlled using the `userAccountControl` attribute.

Getting ready

To set the `userAccountControl` attribute for users, sign in to a domain controller or a member server and/or device with RSAT for Active Directory Domain Services installed.

Sign in with an account that is a member of the **Domain Admins** group.

How to do it...

The `userAccountControl` attribute can be managed in the following two ways:

1. Reading the `userAccountControl` attribute
2. Setting the `userAccountControl` attribute

Reading the userAccountControl attribute

There are three ways to read the `userAccountControl` attribute for users:

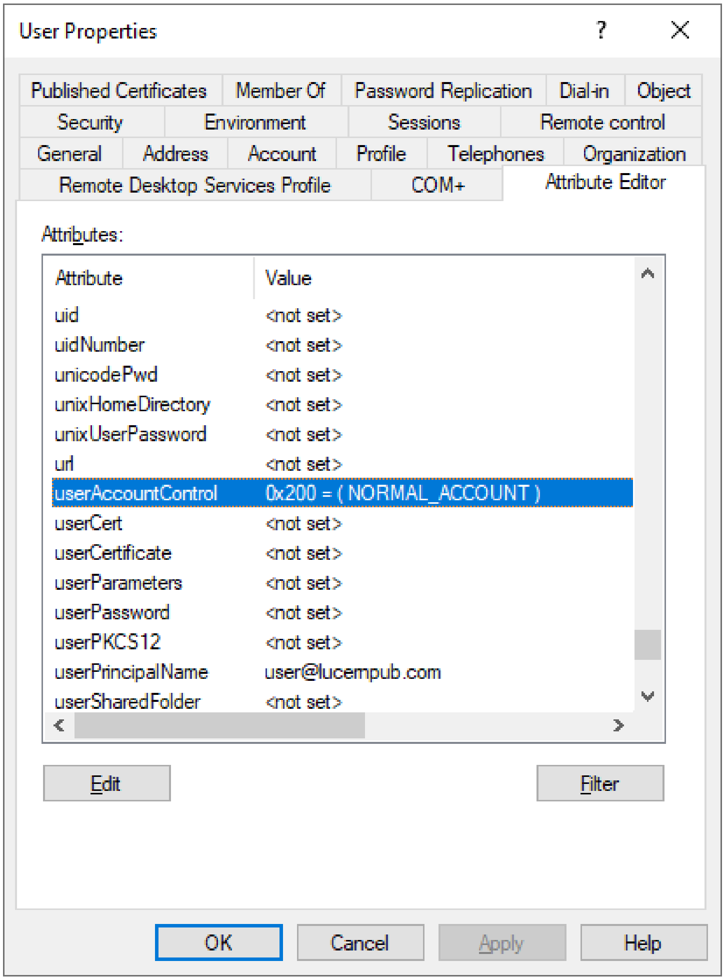
1. Using **Active Directory Users and Computers**
2. Using the **Active Directory Administrative Center**
3. Using Windows PowerShell

Using Active Directory Users and Computers

To read the `userAccountControl` attribute for users using **Active Directory Users and Computers**, follow these steps:

1. Press Start.
2. Search for **Active Directory Users and Computers** and select it from the search results or run `dsa.msc`. The **Active Directory Users and Computers** window appears.
3. Perform one of these series of actions:
    - In the left navigation pane, navigate to the OU or container where the user object resides. In the main pane, select the user object.
    - From the **Action** menu, select **Find…**. The **Find Users, Contacts, and Groups** window appears. In the **Name** field, type the name of the user object whose `userAccountControl` attribute you intend to read, and press *Enter*. From the **Search results** list, select the user object.
4. Right-click the user object and select **Properties** from the menu. The **User Properties** window appears.
5. Navigate to the **Attribute Editor** tab.

6. In the **Attributes** list, scroll down until you reach the `userAccountControl` attribute:



Attribute	Value
uid	<not set>
uidNumber	<not set>
unicodePwd	<not set>
unixHomeDirectory	<not set>
unixUserPassword	<not set>
url	<not set>
userAccountControl	0x200 = (NORMAL_ACCOUNT)
userCert	<not set>
userCertificate	<not set>
userParameters	<not set>
userPassword	<not set>
userPKCS12	<not set>
userPrincipalName	user@lucernpub.com
userSharedFolder	<not set>

Figure 6.15 – userAccountControl attribute for a user object in Active Directory Users and Computers

7. Click **Cancel** to close the **User Properties** window.

Using the Active Directory Administrative Center

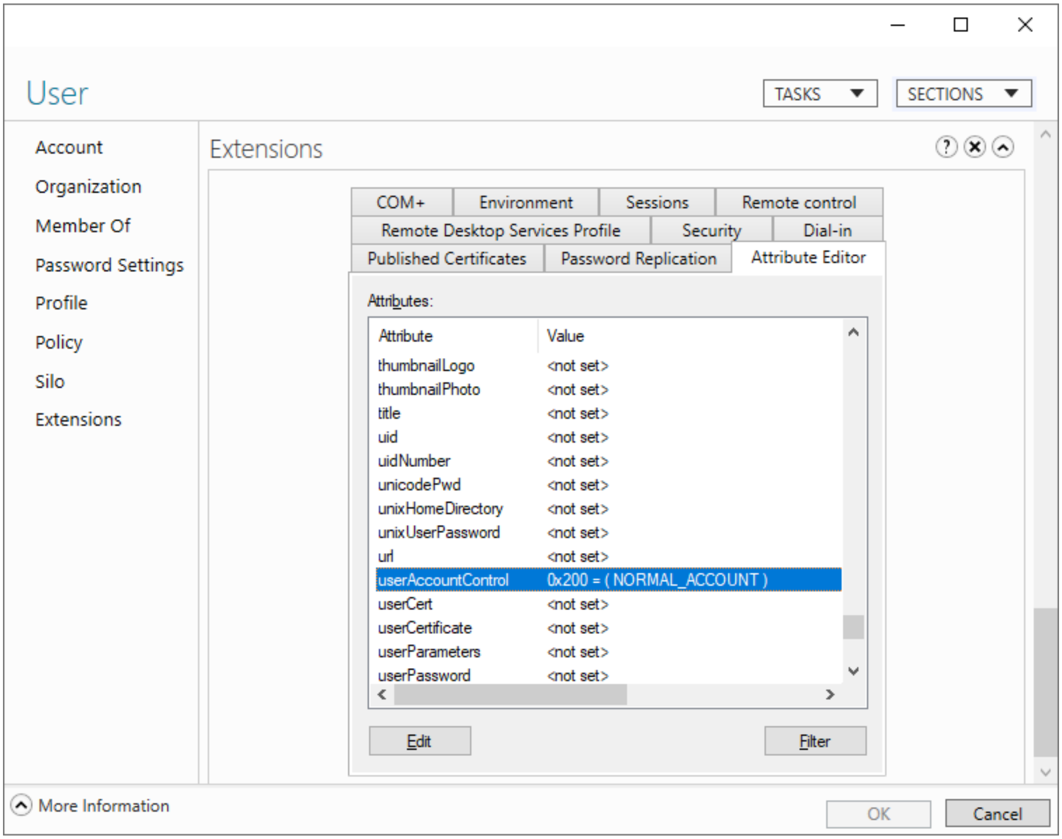
To read the `userAccountControl` attribute for users using the **Active Directory Administrative Center**, follow these steps:

1. Press Start.
2. Search for **Active Directory Administrative Center** and select it from the search results or run `dsac.exe`. The **Active Directory Administrative Center** window appears.
3. Perform one of these series of actions:

    - In the left navigation pane, switch to the tree view. Navigate to the OU or container where the user object whose `userAccountControl` attribute you intend to read resides. In the main pane, select the user object.
    - From the main pane menu, under **Global Search**, type the name of the user object whose `userAccountControl` attribute you intend to read, and press *Enter*. From the list of **Global Search** results, select the user object.

4. Right-click the user object and select **Properties** from the menu. The **User Properties** window appears.
5. In the **User Properties** window, click **Extensions** in the left navigation pane.
6. Navigate to the **Attribute Editor** tab.

7. In the **Attributes** list, scroll down until you reach the `userAccountControl` attribute:



Attribute	Value
thumbnailLogo	<not set>
thumbnailPhoto	<not set>
title	<not set>
uid	<not set>
uidNumber	<not set>
unicodePwd	<not set>
unixHomeDirectory	<not set>
unixUserPassword	<not set>
url	<not set>
userAccountControl	0x200 = (NORMAL_ACCOUNT)
userCert	<not set>
userCertificate	<not set>
userParameters	<not set>
userPassword	<not set>

Figure 6.16 – userAccountControl attribute for a user object in the Active Directory Administrative Center

8. Click **Cancel** to close the **User Properties** window.

Using Windows PowerShell

To read the `userAccountControl` attribute for a specific user object using Windows PowerShell, use the following line of PowerShell code on a system with the Active Directory module for Windows PowerShell:

```
Get-ADUser -Identity 'CN=User,CN=Users,DC=LucernPub,DC=com'
-Properties userAccountControl | Format-Table
name,useraccountcontrol
```

Replace `'CN=User,CN=Users,DC=LucernPub,DC=com'` with the DN property of the user account. This line of PowerShell code returns a table with the display name of the user object and the `userAccountControl` attribute.

To read the `userAccountControl` attribute for all users using Windows PowerShell, use the following line of PowerShell code on a system with the Active Directory module for Windows PowerShell:

```
Get-ADUser -Filter * -Properties userAccountControl | Format-
Table name,useraccountcontrol
```

This line of PowerShell code returns a table with the display names of all user objects and their corresponding `userAccountControl` attributes.

Setting the userAccountControl attribute

There are two ways to set the `userAccountControl` attribute for users:

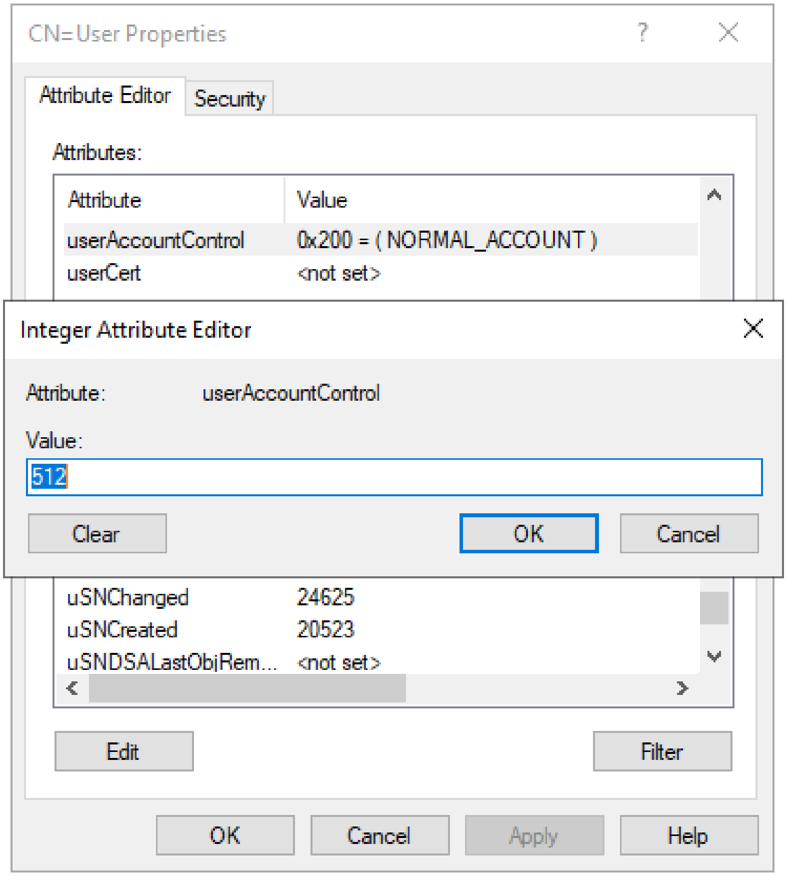
1. Using **ADSI Edit**
2. Using Windows PowerShell

Using ADSI Edit

To set the `userAccountControl` attribute for users using **ADSI Edit**, follow these steps:

1. Press Start.
2. Search for **ADSI Edit** and select it from the search results or run `adsiedit.msc`. The **ADSI Edit** window appears.
3. Right-click **ADSI Edit** in the left navigation pane and select **Connect to…** from the menu.
4. Click **OK** to accept connecting to the **Default Naming Context**.
5. In the left navigation pane, expand the **Default Naming Context**.
6. In the left navigation pane, expand the DN of the domain.
7. In the left navigation pane, expand the container or OU (structure) where the user object resides.
8. In the main pane, select the user object.
9. Next, right-click the user object and select **Properties** from the menu.

10. In the **Attributes** list, scroll down to the `userAccountControl` attribute and select it.
11. Click the **Edit** button:



CN=User Properties

Attributes:

Attribute	Value
userAccountControl	0x200 = (NORMAL_ACCOUNT)
userCert	<not set>
uSNChanged	24625
uSNCreated	20523
uSNDSALastObjRem...	<not set>

Integer Attribute Editor

Attribute: userAccountControl

Value: 512

Figure 6.17 – Editing a user object's userAccountControl attribute in ADSI Edit

12. Type a new value for the `userAccountControl` attribute for the object.
13. Click **OK** to save the new value.
14. Click **OK** to save the changes to the user object.

Using Windows PowerShell

To set the `userAccountControl` attribute for users using Windows PowerShell, use the following lines of PowerShell code on a system with the Active Directory module for Windows PowerShell. Each of the following lines of code represents setting one of the bits in the `userAccountControl` attribute for user objects:

```
Set-ADAccountControl -Identity User -Enable $false
Set-ADAccountControl -Identity User -HomedirRequired $true
Set-ADAccountControl -Identity User -PasswordNotRequired $true
Set-ADAccountControl -Identity User -CannotChangePassword $true
Set-ADAccountControl -Identity User
-AllowReversiblePasswordEncryption $true
Set-ADAccountControl -Identity User -PasswordNeverExpires $true
Set-ADAccountControl -Identity User -MNSLogonAccount $true
Set-ADAccountControl -Identity User -TrustedForDelegation $true
Set-ADAccountControl -Identity User -AccountNotDelegated $true
Set-ADAccountControl -Identity User -UseDESKeyOnly $true
Set-ADAccountControl -Identity User -DoesNotRequirePreAuth
$true
Set-ADAccountControl -Identity User -TrustedToAuthForDelegation
$true
```

Within Windows PowerShell, the `userAccountControl` attribute is not directly changed. Instead, the major implications of changing the attribute define how to achieve this.

How it works...

Many individual settings and options for user objects are stored in the userAccountControl attribute. While administrators can set these options through the UI, they can also be set by modifying the userAccountControl attribute itself.

Additionally, the userAccountControl attribute can also be used in scripts to find user objects with specific sets of values.

The userAccountControl attribute defines a lot of properties for any user and/or computer object. The value for this attribute is built up of bits; every value is a (combination of) 2x value(s):

Name	Value	Value	Value	Description
SCRIPT	1	20	0x00000001	A logon script is executed.
ACCOUNTDISABLE	2	21	0x00000002	The account is disabled.
HOMEDIR_ REQUIRED	8	23	0x00000008	A home folder is required.
LOCKOUT	16	24	0x00000010	
PASSWD_NOTREQD	32	25	0x00000020	A password is not required.
PASSWD_CANT_ CHANGE	64	26	0x00000040	The user cannot change the password.
ENCRYPTED_TEXT_ PWD_ALLOWED	128	27	0x00000080	Store password using reversible encryption.
TEMP_DUPLICATE_ ACCOUNT	256	28	0x00000100	This is an account for users whose primary account is in another domain.
NORMAL_ACCOUNT	512	29	0x00000200	This is a normal, enabled user account.
INTERDOMAIN_ TRUST_ACCOUNT	2048	211	0x00000800	This is a permit to trust an account for a system domain that trusts other domains.
WORKSTATION_ TRUST_ACCOUNT	4096	212	0x00001000	This is a normal computer account.

SERVER_TRUST_ACCOUNT	8192	213	0x00002000	This is a computer account for a domain controller.
DONT_EXPIRE_PASSWORD	65536	216	0x00010000	The password will not expire.
MNS_LOGON_ACCOUNT	131072	217	0x00020000	This is the **Majority Node Set** (**MNS**) logon account, used for clustering.
SMARTCARD_REQUIRED	262144	218	0x00040000	The user is forced to use a smartcard.
TRUSTED_FOR_DELEGATION	524288	219	0x00080000	The service account is trusted for Kerberos delegation.
NOT_DELEGATED	1048576	220	0x00100000	The user will not be delegated to a service even if the service account is set as trusted for Kerberos delegation.
USES_DES_KEY_ONLY	2097152	221	0x00200000	The user uses only **Data Encryption Standard** (**DES**) encryption.
DONT_REQ_PREAUTH	4194304	222	0x00400000	The user does not require Kerberos pre-authentication for logon.
PASSWORD_EXPIRED	8388608	223	0x00800000	The user's password has expired.
TRUSTED_TO_AUTH_FOR_DELEGATION	16777216	224	0x01000000	The account is enabled for delegation.
PARTIAL_SECRETS_ACCOUNT	67108864	226	0x04000000	The account is a **read-only domain controller** (**RODC**).

Table 6.1 - Values for the userAccountControl attribute and their properties

Except for SCRIPT, LOCKOUT, TEMP_DUPLICATE_ACCOUNT, and PASSWORD_EXPIRED, all flags can be set on user objects by an administrator. NORMAL_ACCOUNT doesn't need to be set, as it is set by default on user objects.

SMARTCARD_REQUIRED isn't set this way because it requires certificates and other prerequisites. It would make for an excellent **denial-of-service** (**DoS**) attack vector if this flag could be set.

The INTERDOMAIN_TRUST_ACCOUNT, WORKSTATION_TRUST_ACCOUNT, SERVER_TRUST_ACCOUNT, and PARTIAL_SECRETS_ACCOUNT flags do not apply to user objects.

Using account expiration

User objects can be set to automatically expire.

Getting ready

To set account expiration for a user object, sign in to a domain controller or a member server and/or device with RSAT for Active Directory Domain Services installed.

Sign in with an account that is a member of the **Domain Admins** group or the **Account Operators** group or with an account that is delegated to modify user objects in the domain or scope of the OU where the user object resides.

How to do it...

There are four ways to do this:

1. Using **Active Directory Users and Computers**
2. Using the **Active Directory Administrative Center**
3. Using command-line tools
4. Using Windows PowerShell

The two graphical tools are useful for configuring account expiration on individual accounts, although the **Active Directory Administrative Center** can be used to retrieve the Windows PowerShell cmdlets through the Windows PowerShell History feature.

The command-line tool is particularly useful when you want to configure a time frame (in days) for the account to live on, instead of defining an end date.

Using Active Directory Users and Computers

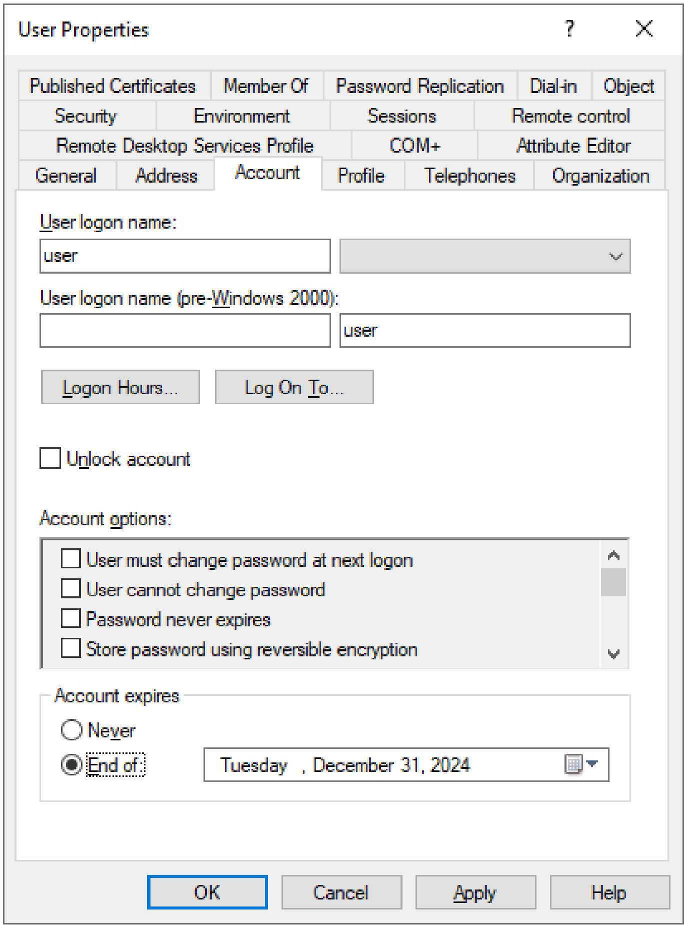
To set account expiration for a user using **Active Directory Users and Computers**, follow these steps:

1. Press Start.
2. Search for **Active Directory Users and Computers** and select it from the search results or run `dsa.msc`. The **Active Directory Users and Computers** window appears.
3. Perform one of these series of actions:

    - In the left navigation pane, navigate to the OU or container where the user object resides that you intend to have automatically expire. In the main pane, select the user object.
    - From the **Action** menu, select **Find…**. The **Find Users, Contacts, and Groups** window appears. In the **Name** field, type the name of the user object you intend to have automatically expire, and press *Enter*. From the **Search results** list, select the user object.

4. Right-click the user object and select **Properties** from the menu. The **User Properties** window appears.
5. Navigate to the **Account** tab.

6. At the bottom of the tab, change the **Never** option in the **Account expires** field to **End of** by selecting the latter option:



User logon name:
user

User logon name (pre-Windows 2000):
user

□ Unlock account

Account options:
□ User must change password at next logon
□ User cannot change password
□ Password never expires
□ Store password using reversible encryption

Account expires
○ Never
● End of: Tuesday , December 31, 2024

Figure 6.18 – Configuring account expiration in Active Directory Users and Computers

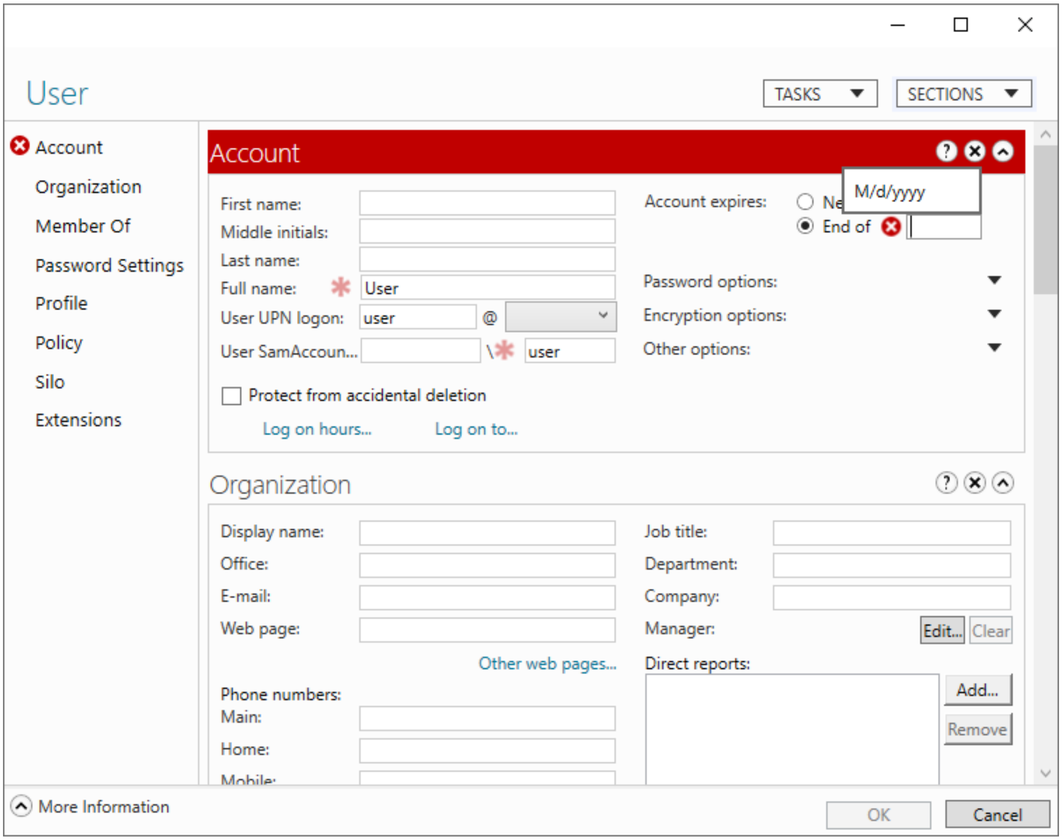
7. Using the calendar icon to the right of the field, select an expiration date for the user object.
8. Click **OK** to save the settings and close the **User Properties** window.

Using the Active Directory Administrative Center

To set account expiration for a user using the **Active Directory Administrative Center**, follow these steps:

1. Press Start.
2. Search for **Active Directory Administrative Center** and select it from the search results or run `dsac.exe`. The **Active Directory Administrative Center** window appears.
3. Perform one of these series of actions:
    - In the left navigation pane, switch to the tree view. Navigate to the OU or container where the user object that you intend to have automatically expire resides. In the main pane, select the user object.
    - From the main pane menu, under **Global Search**, type the name of the user object you intend to have automatically expire, and press *Enter*. From the list of **Global Search** results, select the user object.
4. Right-click the user object and select **Properties** from the menu. The user object's properties window appears.
5. In the top-right corner of the **Account** section, change the **Never** option for **Account expires** to **End of** by selecting the latter option. A red ribbon appears to indicate the **Account** section is missing information.

6. Type the date in the date format specified for the operating system:



Full name: User

User UPN logon: user @

User SamAccoun... \ user

Account expires: ○ Ne ● End of M/d/yyyy

□ Protect from accidental deletion

Figure 6.19 – Configuring account expiration in the Active Directory Administrative Center

7. Below the **End of** field, the **Active Directory Administrative Center** indicates the number of days until the user account expires. Click **OK** to save the settings and close the **User Properties** window.

Using command-line tools

Use the following command to set the days before an account expires in Active Directory:

```
dsmod.exe user 'CN=User,CN=Users,DC=LucernPub,DC=com'
-acctexpires 90
```

Replace DC=LucernPub,DC=com to represent your environment. Replace the other fields in the DN of the user object to represent your scenario.

Using Windows PowerShell

Use the following line of PowerShell code on a system with the Active Directory module for Windows PowerShell installed to set account expiration for a user object:

```
Set-ADAccountExpiration -Identity "CN=User,OU=Organizational
Unit,DC=LucernPub,DC=com" -DateTime "03/01/2027 00:00:00"
```

Replace DC=LucernPub,DC=com to represent your environment. Replace the other fields in the DN of the user object to represent your scenario.

How it works...

User objects can be configured with expiration dates. After this moment in time, the user object can no longer be used to authenticate, including when using KCD.

After the expiration date is set, it is stored in the accountExpires attribute. The 64-bit value stored for this attribute represents the number of 100-nanosecond intervals since January 1, 1601 (**Coordinated Universal Time**, or **UTC**).

Account expiration is a good way to fill in communication gaps between HR and IT in the joiners, movers, and leavers process for IAM.

In an organization where account expiration is used, the end date of the HR contract is typically used to ensure the user object automatically expires when the person would normally leave the organization. Of course, in some organizations, a couple of days are added to allow access to the expenses system and other services the organization offers to people leaving the organization.

There are some caveats, though. For external hires who are normally hired based on project contracts, account expiration may be bothersome. Additionally, people who are on extended leave or travel beyond the end of their contract may be impacted too.

7
Managing Active Directory Groups

In typical Active Directory environments, groups govern access. Groups can be used as distribution lists and/or in **access control lists** (**ACLs**) on file server shares and disks, for delegation in Active Directory itself, and to provide privileged access.

There are several built-in groups, such as the **Domain Admins** group and the **Enterprise Admins** group, which are used in many of the recipes in this book.

For several recipes in this chapter, multiple ways are shown that produce the same outcome. The **Active Directory Users and Computers** (`dsa.msc`) and the **Active Directory Administrative Center** (`dsac.exe`) tools provide graphical means to achieve your goals. **Active Directory Administrative Center**, however, has an important trick up its sleeve; through its **PowerShell History** feature, it provides the ability to see the Windows PowerShell cmdlets behind the clicks. The PowerShell-based methods in this chapter and the additional hints from **Active Directory Administrative Center** provide automation possibilities, while the **Command Prompt** commands provide a way to automate things on even the oldest of domain controllers.

The following recipes are covered in this chapter:

- Creating a group
- Deleting a group
- Managing the direct members of a group
- Managing expiring group memberships
- Changing the scope or type of a group
- Viewing nested group memberships
- Finding empty groups

Creating a group

This recipe demonstrates how to create a group.

Getting ready

To create a group, sign in to a domain controller, a member server, or a device with **Remote Server Administration Tools** (**RSAT**) for Active Directory Domain Services installed.

Sign in with an account that is a member of the **Domain Admins** group, the **Account Operators** group, or with an account that is delegated to create groups in the domain or in the scope of the **Organizational Unit** (**OU**), where the group is to be created.

How to do it...

There are four ways to create a group:

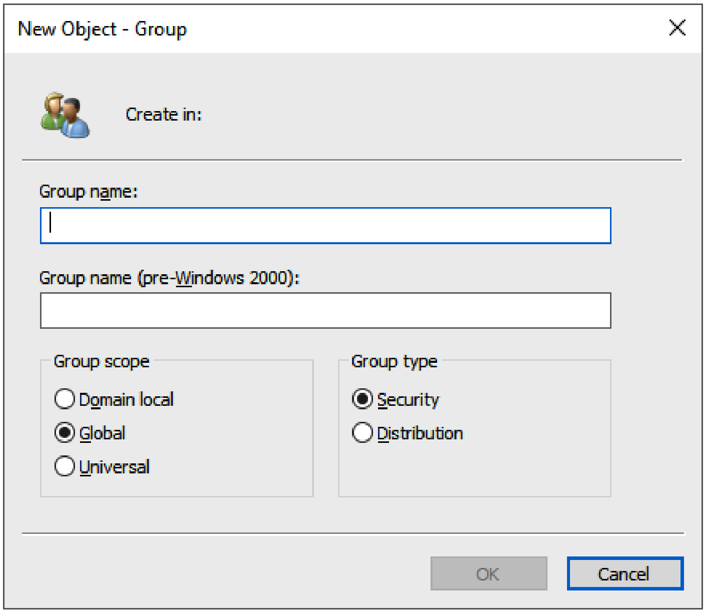
1. Using **Active Directory Users and Computers**
2. Using **Active Directory Administrative Center**
3. Using command-line tools
4. Using Windows PowerShell

Using Active Directory Users and Computers

To create a group using **Active Directory Users and Computers**, perform the following steps:

1. Press Start.
2. Search for **Active Directory Users and Computers** and select it from the search results, or run `dsa.msc`. The **Active Directory Users and Computers** window appears.
3. In the left navigation pane, navigate to the OU or container where you want to create the group object.
4. Perform one of the following actions to open the **New Object - Group** screen:

- From the taskbar, click the **New Group** icon.
- Right-click an empty space in the main window and select **New** and then **Group** from the menu.
- Right-click the OU or container that you want to create a new group in and select **New** and then **Group** from the menu. The **New Object - Group** screen appears:



New Object - Group

Create in:

Group name:

Group name (pre-Windows 2000):

Group scope

○ Domain local

● Global

○ Universal

Group type

● Security

○ Distribution

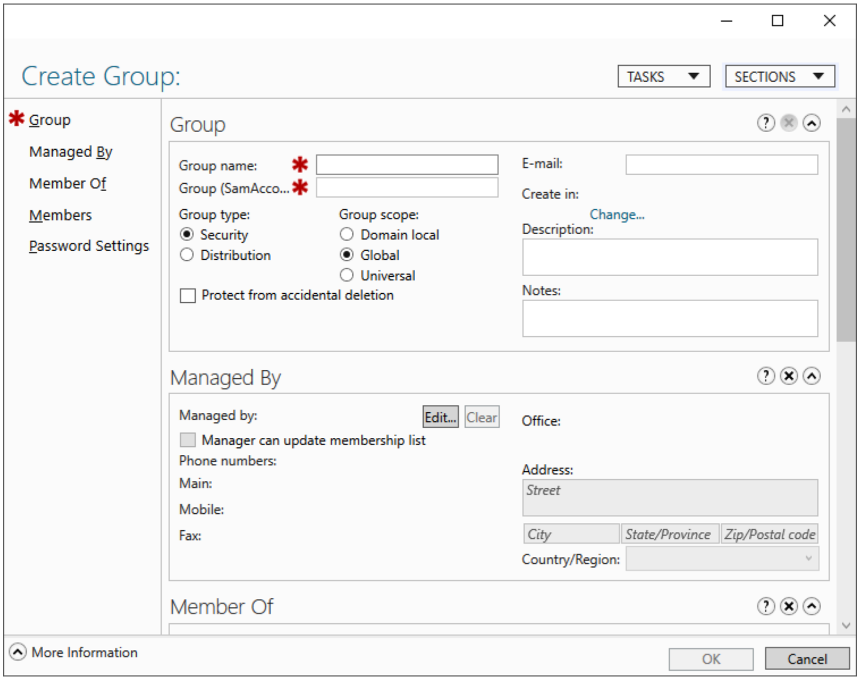
Figure 7.1 – The New Object - Group window in Active Directory Users and Computers

5. In the **New Object - Group** window, specify values for the following fields:
   - Specify **Group name**. The **Group name (pre-Windows 2000)** field will also be filled, based on the name of the group.
   - Specify **Group scope** or accept the default **Global** scope.
   - Specify **Group type** or accept the default **Security** type.
6. Click **OK** to create the group.

Using Active Directory Administrative Center

To create a group using **Active Directory Administrative Center**, perform the following steps:

1. Press Start.
2. Search for **Active Directory Administrative Center** and select it from the search results, or run `dsac.exe`. The **Active Directory Administrative Center** window appears.
3. In the left navigation pane, right-click the domain name and select **New** and then **Group** from the menu. The **Create Group** screen appears:



Create Group:

Group name: *
Group (SamAcco... *
E-mail:
Create in: Change...
Group type:
● Security
○ Distribution
Group scope:
○ Domain local
● Global
○ Universal
□ Protect from accidental deletion
Description:
Notes:

Managed by: Edit... Clear
□ Manager can update membership list
Phone numbers:
Main:
Mobile:
Fax:
Office:
Address:
Street
City | State/Province | Zip/Postal code
Country/Region:

Figure 7.2 – The Create Group window in the Active Directory Administrative Center

4. In the **Create Group** window, specify values for the following fields:

    I. Specify **Group name**. The **Group (SamAccountName) name** field will also be filled, based on the name of the group.

    II. Specify **Group type** or accept the default **Security** type.

    III. Specify **Group scope** or accept the default **Global** scope.

5. Click **OK** to create the group.

Using command-line tools

Use the following command to create a group in Active Directory:

```
dsadd.exe group 'CN=Group,OU=Organizational
Unit,DC=LucernPub,DC=com'
```

Replace `DC=LucernPub,DC=com` to represent your environment. Replace the other fields to represent your group's properties.

Using Windows PowerShell

Use the following line of PowerShell on a system with the Active Directory module for Windows PowerShell installed:

```
New-ADGroup -GroupCategory Security -GroupScope Global -Name
'Group' -Path 'OU=Organizational Unit,DC=LucernPub,DC=com'
-SamAccountName 'Group'
```

Replace `DC=LucernPub,DC=com` to represent your environment. Replace the other fields to represent your group's properties.

When you create a group, some fields are automatically populated and contain default settings. These settings are recommended but can be changed after creating a group, either in the user interface or through commands.

How it works...

This section details group scopes and group types.

Group scopes

There are three group scopes:

- **Global groups**: These groups contain members from their own domain only, but can be granted permissions to resources in any trusting domain.
- **Universal groups**: These groups can contain members from any domain and can be granted permissions to resources in any domain in a specific Active Directory forest.
- **Domain local groups**: These groups can contain members from any trusted domain but are only granted permissions to resources in their own domain.

When deciding what group to create, you might want to consider that global groups and universal groups can be nested into domain local groups, and global groups can be nested into universal groups. Therefore, most organizations opt to create global groups for departments, universal groups for distribution groups and groups that overarch Active Directory trusts, and domain local groups for access rights.

This way, user accounts for people in a finance department can be made members of the finance global group. Then, this group can be nested into several domain local groups that provide read, modify, or full control of the finance department share. The finance universal group is used to send emails to, but it might also be used to access finance-related resources in other Active Directory domains, through the domain local groups in that Active Directory domain.

Group types

There are two group types:

- **Distribution groups**: Distribution groups can be used only with email applications (such as Exchange Server) to send emails to collections of users. Distribution groups are not security-enabled, which means that they cannot be listed in **discretionary access control lists** (**DACLs**).
- **Security groups**: Security groups can provide an efficient way to assign access to resources on your network. Assign user rights to security groups in Active Directory. Assign permissions to security groups for resources.

Distribution groups do not have a **security identifier** (**SID**) and, therefore, can't be used to allow access to resources, except for resources within Microsoft Exchange Server. In comparison, security groups do have SIDs.

If required, you can convert a distribution group into a security group, and vice versa.

Deleting a group

You can use this recipe to delete a previously created group.

Getting ready

To delete a group, sign in to a domain controller, a member server, or a device with RSAT for Active Directory Domain Services installed.

Sign in with an account that is a member of the **Domain Admins** group, the **Account Operators** group, or with an account that is delegated to delete groups in the domain or in the scope of the OU where the group is to be deleted.

How to do it...

There are four ways to delete a group:

1. Using **Active Directory Users and Computers**
2. Using **Active Directory Administrative Center**
3. Using command-line tools
4. Using Windows PowerShell

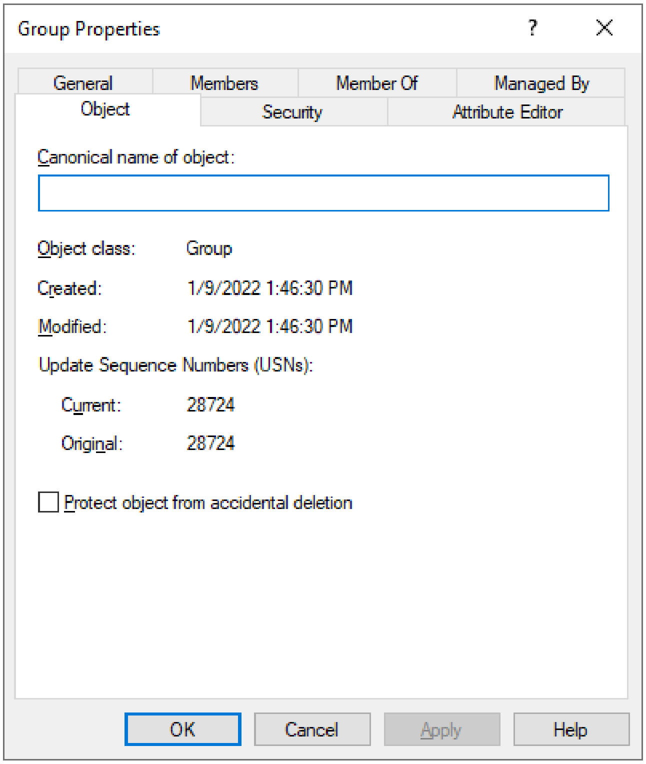
Using Active Directory Users and Computers

To delete a group using **Active Directory Users and Computers**, perform the following steps:

1. Press Start.
2. Search for **Active Directory Users and Computers** and select it from the search results, or run `dsa.msc`. The **Active Directory Users and Computers** window appears.
3. In the **View** menu, enable **Advanced Features**.
4. Perform one of these series of actions:

   - In the left navigation pane, navigate to the OU or container where the group that you intend to delete resides. Then, in the main menu pane, select the group.
   - From the **Action** menu, select **Find…**. In the **Name** field, type in the name of the group you intend to delete, and then press *Enter*. From the list of search results, select the group.

5. Right-click the group and select **Properties** from the menu. The **Group Properties** window appears.
6. Navigate to the **Object** tab.
7. Disable the **Protect object from accidental deletion** option:



Canonical name of object:

Object class: Group

Created: 1/9/2022 1:46:30 PM

Modified: 1/9/2022 1:46:30 PM

Update Sequence Numbers (USNs):

Current: 28724

Original: 28724

☐ Protect object from accidental deletion

Figure 7.3 – The Object tab of a group's properties in Active Directory Users and Computers

8. Click **OK** to close the **Group Properties** window.
9. Right-click the group. This time, select **Delete** from the menu.
10. Click **Yes** in the **Active Directory Domain Services** pop-up window to answer **Are you sure you want to delete the group named 'group'?**.

Using Active Directory Administrative Center

To delete a group using the Active Directory Administrative Center, perform the following steps:

1. Press Start.
2. Search for **Active Directory Administrative Center** and select it from the search results, or run `dsac.exe`. The **Active Directory Administrative Center** window appears.
3. Perform one of these series of actions:
    - In the left navigation pane, switch to the tree view. Navigate to the OU or container where the group that you intend to delete resides. In the main menu pane, select the group.
    - From the main menu pane, under **Global Search**, type in the name of the group you intend to delete, and press *Enter*. From the list of **Global Search** results, select the group.
4. Right-click the group and select **Properties** from the menu. The group's properties window appears.
5. Disable the **Protect object from accidental deletion** option.
6. Click **OK** to close the group's properties window.
7. Right-click the group again; this time, select **Delete** from the menu.
8. Click **Yes** in the **Delete confirmation** pop-up window to answer **Are you sure you want to delete the Group group?**.

Using command-line tools

Ensure the group is not protected from accidental deletion. Then, use the following command to delete the group in Active Directory:

```
dsrm.exe 'CN=Group,OU=Organizational Unit,DC=LucernPub,DC=com'
```

Replace `DC=LucernPub,DC=com` to represent your environment. Replace the other fields to represent your group's properties.

Using Windows PowerShell

Use the following lines of PowerShell on a system with the Active Directory module for Windows PowerShell installed:

```
Set-ADObject -Identity
"CN=Group,OU=OrganizationalUnit,DC=LucernPub,DC=com"
-ProtectedFromAccidentalDeletion $False
Remove-ADObject -Identity 'CN=Group,OU=Organizational
Unit,DC=LucernPub,DC=com'
```

Replace `DC=LucernPub,DC=com` to represent your environment. Replace the other fields to represent your group's properties.

How it works...

When you delete a security group, the object no longer uses its **relative identifier** (**RID**), but the RID and the corresponding SID and **Distinguished Name Tag** (**DNT**) in the domain partition also can't be reused.

When you attempt to delete a group that has the **Protect from accidental deletion** option enabled, you will not be able to delete the object—the option needs to be disabled first.

Managing the direct members of a group

You can use this recipe to manage the direct members of a group.

Getting ready

To manage a group, sign in to a domain controller, a member server, or a device with RSAT for Active Directory Domain Services installed.

Sign in with an account that is a member of the **Domain Admins** group, the **Account Operators** group, or with an account that is delegated to write properties of groups in the domain or in the scope of the OU where the group is to be managed.

How to do it...

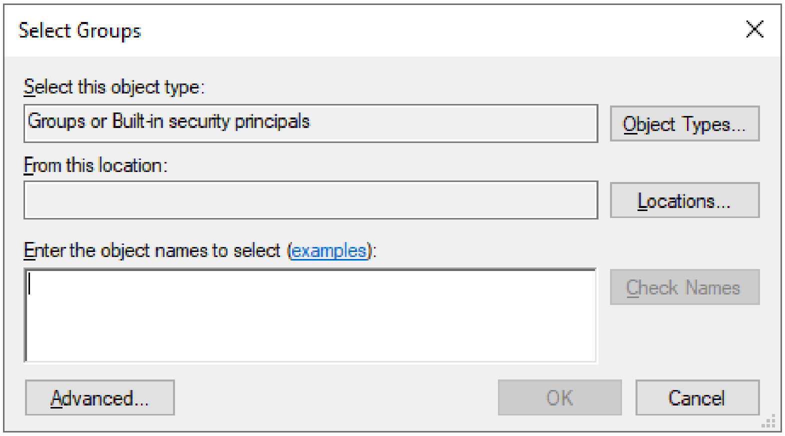
This recipe shows three ways to manage group memberships in Active Directory:

1. Using **Active Directory Users and Computers**
2. Using **Active Directory Administrative Center**
3. Using Windows PowerShell

Using Active Directory Users and Computers

To manage group memberships for a user using **Active Directory Users and Computers**, perform the following steps:

1. Press Start.
2. Search for **Active Directory Users and Computers** and select it from the search results, or run `dsa.msc`. The **Active Directory Users and Computers** window appears.
3. Perform one of these series of actions:
    - In the left navigation pane, navigate to the OU or container where the user that you intend to manage group memberships for resides. In the main menu pane, select the user.
    - From the **Action** menu, select **Find…**. In the **Name** field, type in the name of the user that you intend to manage group memberships for, and then press *Enter*. From the list of **Search results**, select the user.
4. Right-click the user object and select **Add to a group…** from the menu. The **Select Groups** window appears:



Select Groups

Select this object type:
Groups or Built-in security principals

From this location:

Enter the object names to select (examples):

Figure 7.4 – The Select Groups window in Active Directory Users and Computers

5. In the **Select Groups** window, type in the name of the group that you want to add the user account to, otherwise, click the **Advanced…** button to search for the group.
6. Click the **Check Names** button.
7. Click **OK** to add the user to the group.

To manage group memberships for a group using **Active Directory Users and Computers**, perform the following steps:

1. Press Start.
2. Search for **Active Directory Users and Computers** and select it from the search results, or run `dsa.msc`. The **Active Directory Users and Computers** window appears.
3. Perform one of these series of actions:
   - In the left navigation pane, navigate to the OU or container where the group that you intend to manage resides. In the main menu pane, select the group.
   - From the **Action** menu, select **Find…**. In the **Name** field, type in the name of the group that you intend to manage, and then press *Enter*. From the list of search results, select the group.
4. Right-click the group and select **Properties** from the menu.
5. Navigate to the **Members** tab.
6. Perform one of the following actions:
   - Click the **Add…** button to add users, contacts, computers, service accounts, or groups to the group. The **Select Users, Contacts, Computers, Service Accounts, or Groups** window appears. In this window, type in the name of the user account(s) that you want to add to the group, otherwise, click the **Advanced…** button to search for the user account(s). Click the **Check Names** button, and then click **OK** to add the user(s) to the group.
   - Select one or more user objects, contacts, computer objects, service accounts, or groups in the **Members** list. Then, click the **Remove** button to remove the member(s) from the group. In the **Active Directory Domain Services** pop-up window, click **Yes** as the answer to **Do you want to remove the selected member(s) from the group?** to remove the selected member(s).
7. Click **OK** to close the group's properties window and save the changes.

Using Active Directory Administrative Center

To manage group memberships for a user using **Active Directory Administrative Center**, perform the following steps:

1. Press Start.
2. Search for **Active Directory Administrative Center** and select it from the search results, or run `dsac.exe`. The **Active Directory Administrative Center** window appears.
3. Perform one of these series of actions:
    - In the left navigation pane, switch to the tree view. Navigate to the OU or container where the group object resides. In the main menu pane, select the group object.
    - From the main pane menu, under **Global Search**, type in the name of the user object, and then press *Enter*. From the list of **Global Search** results, select the user object.
4. Right-click the user object and select **Add to group…** from the menu.
5. In the **Select Groups** window, type in the name of the group that you want to add the user account to or click the **Advanced…** button to search for the group.
6. Click the **Check Names** button.
7. Click **OK** to add the user to the group.

To manage group memberships for a group using **Active Directory Administrative Center**, perform the following steps:

1. Press Start.
2. Search for **Active Directory Administrative Center** and select it from the search results, or run `dsac.exe`. The **Active Directory Administrative Center** window appears.
3. Perform one of these series of actions:
    - In the left navigation pane, switch to the tree view. Navigate to the OU or container where the group resides. In the main menu pane, select the group.
    - From the main menu pane, under **Global Search**, type in the name of the group, and then press *Enter*. From the list of **Global Search** results, select the group.
4. Right-click the group and select **Properties** from the menu. The group's properties window appears.

5. In the left navigation pane, click **Members**:

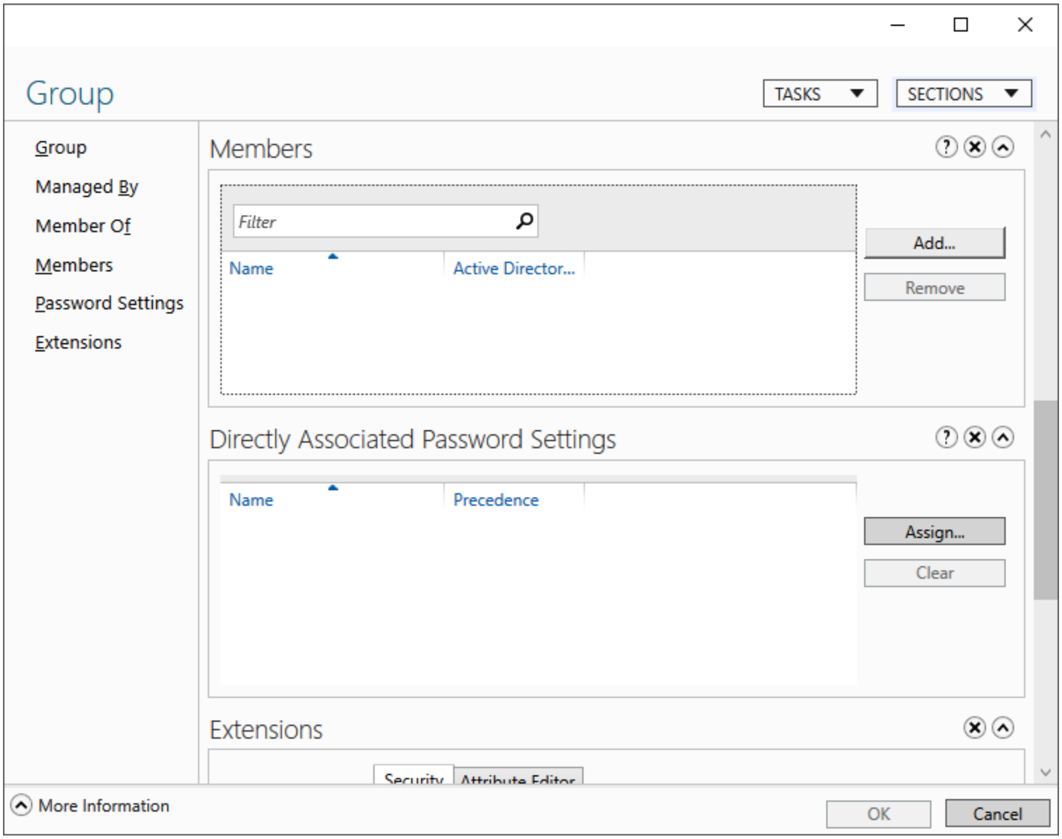


Figure 7.5 – The Members section of a group's properties in the Active Directory Administrative Center

6. In the **Members** section, perform one of the following actions:

   - Click the **Add…** button to add users, contacts, computers, service accounts, or groups to the group. In the **Select Users, Contacts, Computers, Service Accounts, or Groups** pop-up window, type in the name of the user account(s) that you want to add to the group, otherwise, click the **Advanced…** button to search for the user account(s). Click the **Check Names** button, and then click **OK** to add the user to the group.
   - Select one or more user objects, contacts, computer objects, service accounts, or groups in the group's properties window in the list of **Members**. Click **Remove** to remove the member(s) from the group.

7. Click **OK** to close the group's properties window and save the changes.

Using Windows PowerShell

Use the following line of PowerShell to add a user to a group in Active Directory on a system with the Active Directory module for Windows PowerShell installed:

```
Add-ADGroupMember -Identity 'CN=Group,OU=Organizational
Unit,DC=LucernPub,DC=com' -Members 'User'
```

Replace `DC=LucernPub,DC=com` to represent your environment. Replace the other fields to represent your group's and user's properties.

Use the following line of PowerShell to remove a user from a group in Active Directory on a system with the Active Directory module for Windows PowerShell installed:

```
Remove-ADGroupMember -Identity 'CN=Group,OU=Organizational
Unit,DC=LucernPub,DC=com' -Members 'User'
```

Replace `DC=LucernPub,DC=com` to represent your environment. Replace the other fields to represent your group's and user's properties.

How it works...

Groups can have members; members can be user accounts, contacts, computers, service accounts, and other groups. When a group is a member of another group, this is called group nesting.

Members are linked to the group through linked values; the members are stored in the `member` attribute of the group object. The link values are replicated using link-value replication. This ensures that only changed links are replicated between domain controllers, instead of the entire `members` attribute when the Windows Server 2003 **Forest Functional Level** (**FFL**) or a later version is used.

Managing expiring group memberships

Group memberships can alternatively be configured to expire.

Getting ready

To use expiring group memberships, the Active Directory FFL needs to be Windows Server 2012 R2, or a later version.

The optional `Privileged Access Management` feature needs to be enabled. This can be achieved using the following line of PowerShell on a system with the Active Directory module for Windows PowerShell installed:

```
Enable-ADOptionalFeature 'Privileged Access Management Feature'
-Scope ForestOrConfigurationSet -Target lucernPub.com
```

To manage a group, sign in to a domain controller, a member server, or a device with RSAT for Active Directory Domain Services installed.

Sign in with an account that is a member of the **Domain Admins** group, the **Account Operators** group, or with an account that is delegated to write properties of groups in the domain or in the scope of the OU where the group is to be managed.

How to do it...

This feature can only be leveraged using Windows PowerShell.

Normally, to add a group membership to a group, you'd use the following lines of PowerShell:

```
Add-ADGroupMember -Identity 'CN=Group,OU=Organizational
Unit,DC=LucernPub,DC=com' -Members 'User'
```

However, to create an expiring group membership for a group, use the following line of PowerShell on a system with the Active Directory module for Windows PowerShell installed:

```
Add-ADGroupMember -Identity 'CN=Group,OU=Organizational
Unit,DC=LucernPub,DC=com' -Members 'User' -MemberTimeToLive
(New-TimeSpan -Days 14)
```

Replace `DC=LucernPub,DC=com` to represent your environment. Replace the other fields to represent your group's and user's properties.

To view the time-to-live for group memberships, use the following line of PowerShell on a system with the Active Directory module for Windows PowerShell installed:

```
Get-ADGroup 'CN=Group,OU=Organizational
Unit,DC=LucernPub,DC=com' -Property member
-ShowMemberTimeToLive
```

Replace `DC=LucernPub,DC=com` to represent your environment. Replace the other fields to represent your group's properties.

How it works...

Active Directory in Windows Server 2016 offers a new feature that is labeled **expiring links**. This feature configures links in link-value attributes, such as the `member` attribute for a group, with an optional time-to-live value. When the time is up, the link expires; if it is a group membership, then the membership disappears.

This feature can be used to provide temporary group memberships.

Changing the scope or type of a group

You can use this recipe to change the scope and/or type of a group in Active Directory.

Getting ready

To manage a group, sign in to a domain controller, a member server, or a device with RSAT for Active Directory Domain Services installed.

Sign in with an account that is a member of the Domain Admins group, the Account Operators group, or with an account that is delegated to write properties of groups in the domain or in the scope of the OU where the group is to be managed.

How to do it...

This recipe shares four ways to manage the scope or type for a group in Active Directory:

- Using **Active Directory Users and Computers**
- Using **Active Directory Administrative Center**
- Using the Active Directory command-line tools
- Using Windows PowerShell

Using Active Directory Users and Computers

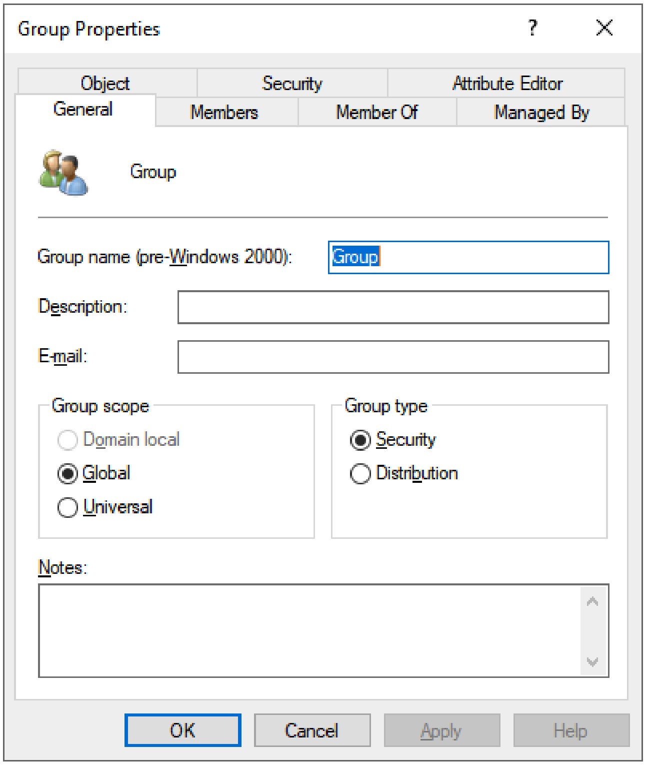
To change the scope or type for a group using **Active Directory Users and Computers**, perform the following steps:

1. Press Start.
2. Search for **Active Directory Users and Computers** and select it from the search results, or run `dsa.msc`. The **Active Directory Users and Computers** window appears.

3. Perform one of these series of actions:

    - In the left navigation pane, navigate to the OU or container where the group that you intend to manage resides. In the main menu pane, select the group.
    - From the **Action** menu, select **Find…**. In the **Name** field, type in the name of the group that you intend to manage, and then press *Enter*. From the list of search results, select the group.

4. Right-click the group and select **Properties** from the menu. The group's properties window appears:



Group name (pre-Windows 2000): Group

Description:

E-mail:

Group scope

○ Domain local
● Global
○ Universal

Group type

● Security
○ Distribution

Notes:

Figure 7.6 – A Group Properties window in Active Directory Users and Computers

5. Change the group's scope in the **Group scope** area, and/or change the group's type in the **Group type** area.
6. Click **OK** to close the group's properties window and save the changes.

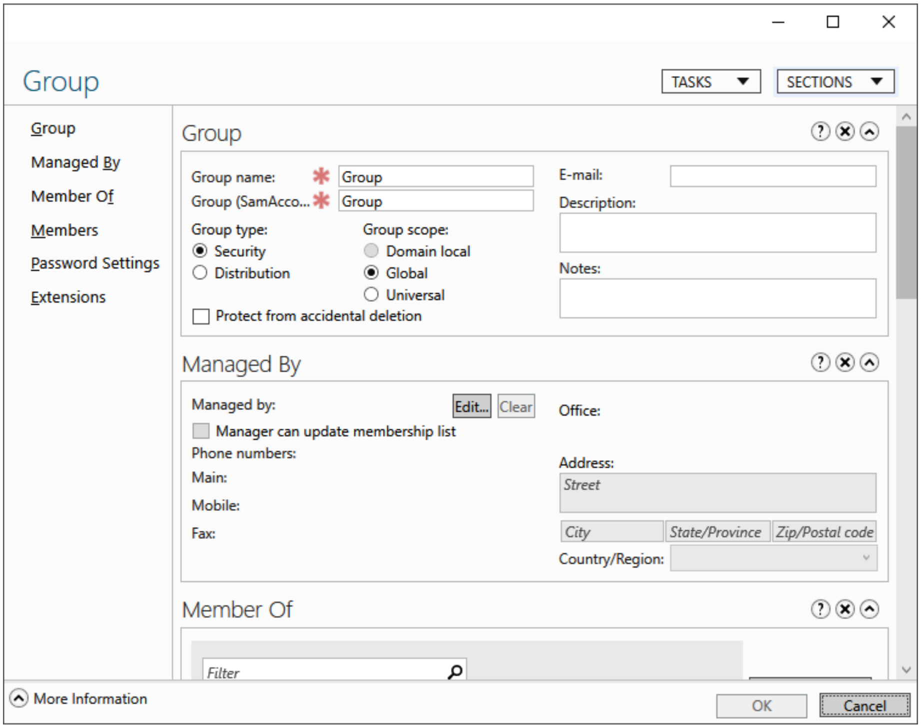
Using Active Directory Administrative Center

To change the scope or type for a group using **Active Directory Administrative Center**, perform the following steps:

1. Press Start.
2. Search for **Active Directory Administrative Center** and select it from the search results, or run `dsac.exe`. The **Active Directory Administrative Center** window appears.
3. Perform one of these series of actions:

   - In the left navigation pane, switch to the tree view. Navigate to the OU or container where the group resides. In the main menu pane, select the group.
   - From the main menu pane, under **Global Search**, type in the name of the group, and then press *Enter*. From the list of **Global Search** results, select the group.

4. Right-click the group and select **Properties** from the menu. The group's properties window appears:



Group name: Group

Group (SamAcco...: Group

E-mail:

Description:

Group type: ● Security ○ Distribution

Group scope: ○ Domain local ● Global ○ Universal

Notes:

□ Protect from accidental deletion

Managed by: Edit... Clear

□ Manager can update membership list

Phone numbers:

Main:

Mobile:

Fax:

Office:

Address: Street City State/Province Zip/Postal code

Country/Region:

Figure 7.7 – A group's properties window in the Active Directory Administrative Center

5. Change the group's type underneath **Group type** and/or change the group's scope underneath **Group scope**.
6. Click **OK** to close the group's properties window and save the changes.

Using command-line tools

Use the following command to change the scope of a group in Active Directory to `Global`:

```
dsmod.exe group 'CN=Group,OU=Organizational
Unit,DC=LucernPub,DC=com' -scope g
```

Use the following command to change the scope of a group in Active Directory to `Universal`:

```
dsmod.exe group 'CN=Group,OU=Organizational
Unit,DC=LucernPub,DC=com' -scope u
```

Use the following command to change the scope of a group in Active Directory to `DomainLocal`:

```
dsmod.exe group 'CN=Group,OU=Organizational
Unit,DC=LucernPub,DC=com' -scope l
```

Use the following command to change the type of group in Active Directory from a distribution group to a security group:

```
dsmod.exe group 'CN=Group,OU=Organizational
Unit,DC=LucernPub,DC=com' -secgrp yes
```

Use the following command to change the type of group in Active Directory from a security group to a distribution group:

```
dsmod.exe group 'CN=Group,OU=Organizational
Unit,DC=LucernPub,DC=com' -secgrp no
```

Replace `DC=LucernPub,DC=com` to represent your environment in the above five command examples. Replace the other fields to represent your group's properties.

Using Windows PowerShell

Use the following line of PowerShell to change the scope of a group in Active Directory to `Global` on a system with the Active Directory module for Windows PowerShell installed:

```
Set-ADGroup -Identity 'CN=Group,OU=Organizational
Unit,DC=LucernPub,DC=com' -GroupScope Global
```

Use the following line of PowerShell to change the scope of a group in Active Directory to `Universal` on a system with the Active Directory module for Windows PowerShell installed:

```
Set-ADGroup -Identity 'CN=Group,OU=Organizational
Unit,DC=LucernPub,DC=com' -GroupScope Universal
```

Use the following line of PowerShell to change the scope of a group in Active Directory to `DomainLocal` on a system with the Active Directory module for Windows PowerShell installed:

```
Set-ADGroup -Identity 'CN=Group,OU=Organizational
Unit,DC=LucernPub,DC=com' -GroupScope DomainLocal
```

Use the following line of PowerShell to change the type of group in Active Directory from a distribution group to a security group on a system with the Active Directory module for Windows PowerShell installed:

```
Set-ADGroup -Identity 'CN=Group,OU=Organizational
Unit,DC=LucernPub,DC=com' -GroupCategory Security
```

Use the following line in PowerShell to change the type of group in Active Directory from a security group to a distribution group on a system with the Active Directory module for Windows PowerShell installed:

```
Set-ADGroup -Identity 'CN=Group,OU=Organizational
Unit,DC=LucernPub,DC=com' -GroupCategory Distribution
```

Replace `DC=LucernPub,DC=com` to represent your environment in the above five Windows PowerShell examples. Replace the other fields to represent your group's properties.

How it works...

This section details group scopes and group types.

Group scopes

There are three group scopes:

- **Global** groups
- **Universal** groups
- **Domain local** groups

Group types

There are two group types, as follows:

- **Distribution** groups
- **Security** groups

You can convert a distribution group into a security group, and vice versa.

See also...

The *Creating a group* recipe earlier in this chapter provides more information on the group scopes and types.

Viewing nested group memberships

This recipe demonstrates how to enumerate all members of a group, even those members in groups that are members of the same group.

Getting ready

To view nested group memberships for a group, sign in to a domain controller, a member server, or a device with RSAT for Active Directory Domain Services installed.

Sign in with a domain account.

How to do it...

To view nested group memberships, double-click the groups listed on the **Members** tab in the properties of a group and look at its members. When groups are heavily nested, though, this becomes tedious fast. A much better approach is to use Windows PowerShell.

Use the following line of PowerShell to enumerate all group memberships in Active Directory for a group on a system with the Active Directory module for Windows PowerShell installed:

```
Get-ADGroupMember -Identity 'CN=Group,OU=Organizational
Unit,DC=LucernPub,DC=com' -Recursive | Out-GridView
```

Replace `DC=LucernPub,DC=com` to represent your environment. Replace the other fields to represent your group's properties.

How it works...

The `Get-ADGroupMember` PowerShell cmdlet offers the `-recursive` switch to view the nested group memberships. When a user object has memberships through various other groups and/or direct memberships, the user object is only returned once.

To make the output readable and sortable, `Out-GridView` displays the output in a graphical user interface. This part can be omitted to display the results in the PowerShell window or changed to output on the clipboard or CSV file.

Finding empty groups

This recipe demonstrates how to find groups without group members. Every object in Active Directory takes up resources. When a group is not used, it may be deleted.

Getting ready

To view group memberships for a group, sign in to a domain controller, a member server, or a device with RSAT for Active Directory Domain Services installed.

Sign in with a domain account.

How to do it...

Use the following line of PowerShell to find all groups without memberships in Active Directory on a system with the Active Directory module for Windows PowerShell installed:

```
Get-ADGroup -Filter * -Properties members | Where-Object {$_.
Members.count -eq 0}  | Out-GridView
```

Replace DC=LucernPub,DC=com to represent your environment. Replace the other fields to represent your group's properties.

How it works...

The Get-ADGroup PowerShell cmdlet is used to get the members attribute. Then, recursively, for each group, the membership count is queried. If this count is zero, it means there are no group members; when this is the case, the group is returned.

To make the output readable and sortable, Out-GridView displays the output in a graphical user interface. This part can be omitted to display the results in the PowerShell window or changed to output on the clipboard or CSV file.

8
Managing Active Directory Computers

Computer objects in Active Directory represent actual devices. When you join a device to an Active Directory domain, it shows up in the database. From there, you can manage it using Active Directory and Group Policy. Joining a computer to the domain can be done in several ways, and it's good to keep certain scenarios at the back of your mind when working with computer objects, such as the integrity of the secure channel and the default ability of any Active Directory user to join devices to the domain.

The following recipes are covered in the chapter:

- Creating a computer
- Deleting a computer
- Joining a computer to the domain
- Renaming a computer
- Testing the secure channel for a computer
- Resetting a computer's secure channel
- Changing the default quota for creating computer objects

Creating a computer

In this recipe, we create a computer object. This object can later be used to attach a device to when it is domain-joined.

Getting ready

To create a computer account, you need to sign in to a domain controller or to a member server, or a device with the **Remote Server Administration Tools** (**RSAT**) for Active Directory Domain Services installed.

Sign in with an account that is a member of the **Domain Admins** group, the **Account Operators** group, or with an account that has delegated privileges to create computer objects in the domain or in the scope of the **Organizational Unit** (**OU**) where the computer account is created.

How to do it...

This recipe describes four ways to create a computer object:

- Using **Active Directory Users and Computers**
- Using the **Active Directory Administrative Center**
- Using command-line tools
- Using Windows PowerShell

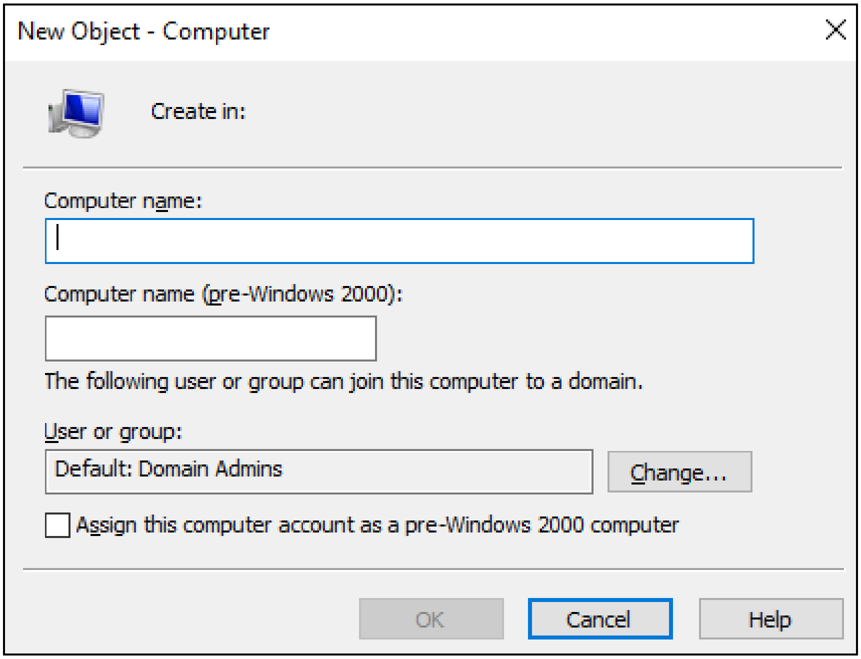
Using Active Directory Users and Computers

To create a computer using **Active Directory Users and Computers**, perform these steps:

1. Press Start.
2. Search for **Active Directory Users and Computers** and select it from the search results or run `dsa.msc`. The **Active Directory Users and Computers** window appears.
3. In the left navigation pane, navigate to the OU or container where you want to create the computer object.

4. Perform one of the following actions to open the **New Object - Computer** window:

    - From the taskbar, click the **New Computer** icon.
    - Right-click an empty space in the main window and select **New,** and then select **Computer** from the menu.
    - Right-click the OU or container that you want to create a new computer in and select **New,** and then select **Computer** from the menu. The **New Object - Computer** window appears:



New Object - Computer

Create in:

Computer name:

Computer name (pre-Windows 2000):

The following user or group can join this computer to a domain.

User or group: Default: Domain Admins

□ Assign this computer account as a pre-Windows 2000 computer

Figure 8.1 – The New Object - Computer window in Active Directory Users and Computers

5. In the **New Object - Computer** window, specify values for the following fields:

    I. Specify the name of the computer.

    II. Change the **Computer name (pre-Windows 2000)** field if you need it to be different from the automatically generated value, based on the value of the **Computer name:** field.

6. Click **OK**.

Using the Active Directory Administrative Center

To create a computer using the **Active Directory Administrative Center**, perform these steps:

1. Press Start.
2. Search for **Active Directory Administrative Center** and select it from the search results or run `dsac.exe`. The **Active Directory Administrative Center** window appears.
3. In the left navigation pane, right-click the domain name and select **New** and then **Computer** from the menu. The **Create Computer:** window appears:

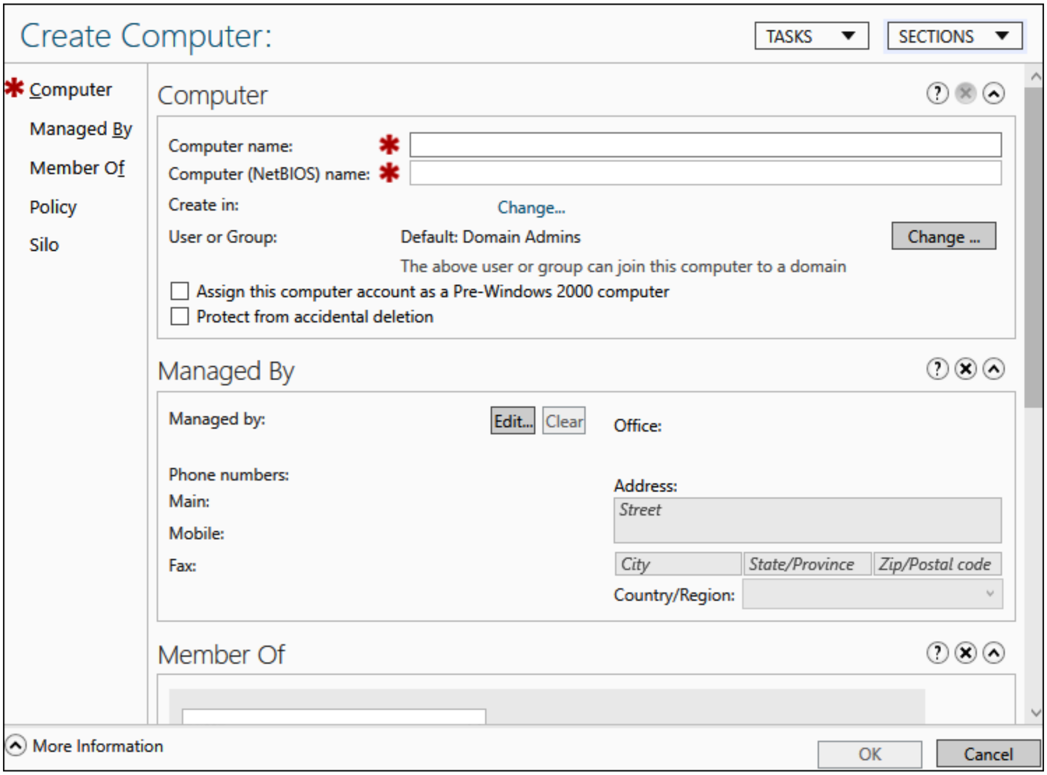


Figure 8.2 – The Create Computer: window in the Active Directory Administrative Center

4. In the **Create Computer:** window, specify values for the following fields:

    I. Specify the name of the computer.

    II. Change the **Computer (NetBIOS) name:** field if you need it to be different from the automatically generated value, based on the value of the **Computer name** field.

5. Click **OK**.

Using command-line tools

Use the following command to create a computer object in Active Directory:

```
dsadd.exe computer
"CN=Computer,CN=Computers,DC=LucernPub,DC=com"
```

Replace `DC=LucernPub,DC=com` to represent your environment. Replace the other fields to represent your computer's properties.

Using Windows PowerShell

Use the following line of PowerShell on a system with the Active Directory module for Windows PowerShell installed:

```
New-ADComputer -Name "Computer" -sAMAccountName "Computer"
-Path "CN=Computers,DC=LucernPub,DC=com"
```

Replace `DC=LucernPub,DC=com` to represent your environment. Replace the other fields to represent your computer's properties.

How it works...

When you create a computer object, an actual device can attach to it when it is domain-joined. This way, the location of the computer object, and any settings that you want the object to have, can be preconfigured. When a device is joined with the same hostname as the computer name, the computer object is attached to it.

When the computer name is over 15 characters, the pre-Windows 2000 computer name is truncated to only use the first 15 characters, by default. It might be a good idea to change the computer name when the data that makes the computer object unique in the Active Directory context is the last part of the computer name. In this case, a good approach would be to manually truncate the first part of the computer name to make room for the unique part.

A descriptive name for every computer object is useful to ensure that the computer object can be identified in management and auditing scenarios. Additionally, the **My Network locations** functionality on domain-joined Windows devices shows the description by default, rather than the hostname.

When you create a computer object, the object consumes a **relative identifier** (**RID**) and **Distinguished Name Tag** (**DNT**) in the domain partition.

There's more...

The **Domain Admins** group is assigned the right to attach an actual device to the computer object, but this can be changed to include any group in Active Directory.

When an organization scales up to multiple political entities with their own admin workforce, it might be a good idea to ensure that the computer objects for a certain entity can only be attached using the credentials given to the members of a specific group for that particular entity.

Deleting a computer

Perform the steps in this recipe to delete a previously created computer object.

Getting ready

To delete a computer account, sign in to a domain controller, a member server, or a device with the RSAT for Active Directory Domain Services installed.

Sign in with an account that is a member of the **Domain Admins** group, the **Account Operators** group, or with an account that has delegated privileges to delete computer objects in the domain or in the scope of the OU where the computer account is to be deleted.

How to do it...

This recipe describes four ways to delete a computer object:

- Using **Active Directory Users and Computers**
- Using the **Active Directory Administrative Center**
- Using command-line tools
- Using Windows PowerShell

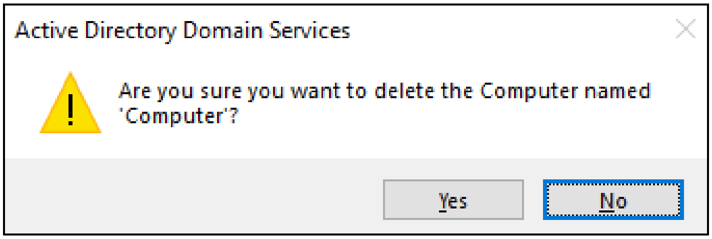
Using Active Directory Users and Computers

To delete a computer using **Active Directory Users and Computers**, perform these steps:

1. Press Start.
2. Search for **Active Directory Users and Computers** and select it from the search results or run `dsa.msc`. The **Active Directory Users and Computers** window appears.
3. In the **View** menu, enable **Advanced Features**.
4. Perform one of these series of actions:

   - In the left navigation pane, navigate to the OU or container where the computer object that you intend to delete resides. In the main menu pane, select the computer object.
   - From the **Action** menu, select **Find**. In the **Name** field, type the name of the computer object that you intend to delete, and then press *Enter*. From the **Search results** list, select the computer object.

5. Right-click the computer object and select **Properties** from the menu. The computer's properties window appears.
6. Navigate to the **Object** tab.
7. Disable the **Protect object from accidental deletion** option.
8. Click **OK** to close the **Properties** window for the object.
9. Right-click the computer object and select **Delete** from the menu. The **Active Directory Domain Services** pop-up window appears:



Active Directory Domain Services

Are you sure you want to delete the Computer named 'Computer'?

Figure 8.3 – The Active Directory Domain Services pop-up window

10. Click **Yes** in the **Active Directory Domain Services** pop-up window in order to answer the **Are you sure you want to delete the Computer named 'Computer'?** question.

Using the Active Directory Administrative Center

To delete a computer using the **Active Directory Administrative Center**, perform these steps:

1. Press Start.
2. Search for **Active Directory Administrative Center** and select it from the search results or run `dsac.exe`. The **Active Directory Administrative Center** window appears.
3. Perform one of the following actions:

   - In the left navigation pane, switch to the tree view. Navigate to the OU or container where the computer object that you intend to delete resides. In the main menu pane, select the computer object.
   - From the main menu pane, under **Global Search**, type the name of the computer object that you intend to delete, and then press *Enter*. From the list of **Global Search** results, select the computer object.

4. Right-click the computer object and select **Properties** from the menu. The computer's properties window appears:

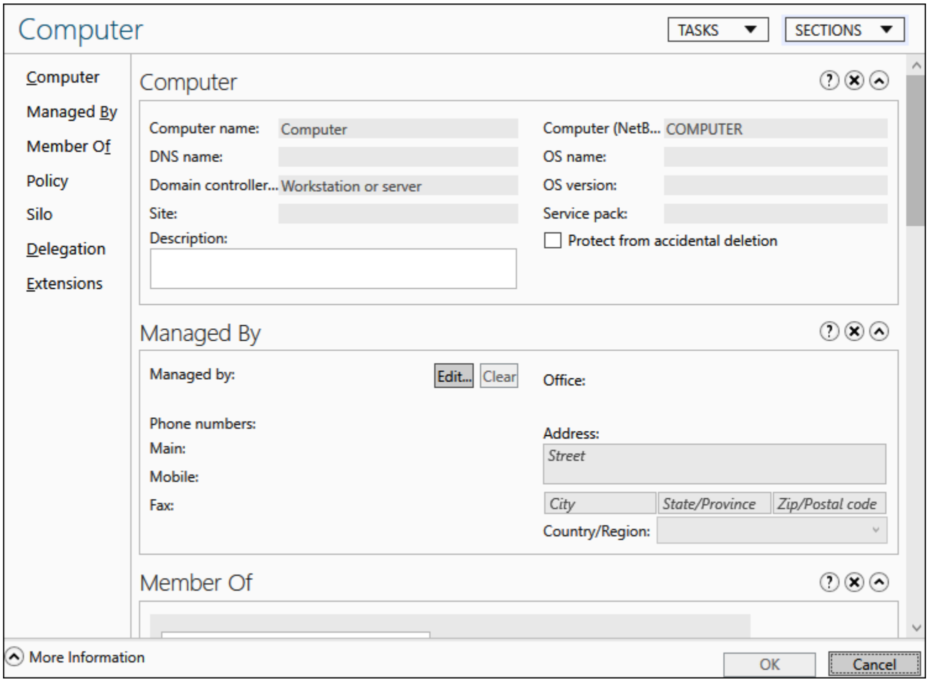
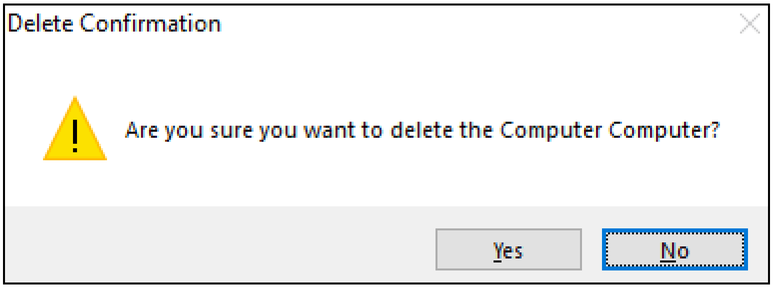

Computer name: Computer
Computer (NetB... COMPUTER
Domain controller... Workstation or server
☐ Protect from accidental deletion

Figure 8.4 – A computer's properties window in the Active Directory Administrative Center

5. Disable the **Protect from accidental deletion** option.
6. Click **OK** to close the computer's properties window.
7. Right-click the computer object and select **Delete** from the list. The **Delete Confirmation** pop-up window appears:



Delete Confirmation

Are you sure you want to delete the Computer Computer?

Figure 8.5 – The Delete Confirmation pop-up window

8. Click **Yes** in the **Delete Confirmation** pop-up window to answer the **Are you sure you want to delete the Computer Computer?** question.

Using command-line tools

Use the following command to delete a computer object in Active Directory:

```
dsrm.exe "CN=Computer,CN=Computers,DC=LucernPub,DC=com"
```

Replace `DC=LucernPub,DC=com` to represent your environment. Replace the other fields to represent your computer's properties.

Using Windows PowerShell

Use the following line of PowerShell on a system with the Active Directory module for Windows PowerShell installed:

```
Remove-ADComputer -Identity
"CN=Computer,CN=Computers,DC=LucernPub,DC=com"
```

Replace `DC=LucernPub,DC=com` to represent your environment. Replace the other fields to represent your computer's properties.

How it works...

When you delete a computer object, the object no longer uses its RID, but the RID and the corresponding SID and DNT in the domain partition can't be reused.

When you attempt to delete a computer object that has the **Protect from accidental deletion** option enabled, you will not be able to delete the object—this option needs to be disabled first.

When the **Active Directory Recycle Bin** option is enabled, the deleted computer object emerges in the **Deleted Objects** container.

See also

Refer to the *Enabling the Active Directory Recycle Bin* recipe in *Chapter 1*, *Optimizing Forests, Domains, and Trusts*.

Joining a computer to the domain

Use this recipe to join a Windows-based computer to an Active Directory domain.

This recipe shows three ways to accomplish this:

- Attach a Windows-based device to a previously created computer object.
- Join a Windows-based device to a domain using the **graphical user interface** (**GUI**).
- Join a Windows-based device to a domain using Windows PowerShell.

Getting ready

To communicate with domain controllers, the device needs to be on the same logical networks as the domain controllers. You might also be able to join a device through a **virtual private network** (**VPN**) connection—when the required traffic is allowed through.

To find domain controllers for the Active Directory domain, the device needs to be able to resolve the name of the Active Directory environment. Configuring devices with a DNS server address through **Dynamic Host Configuration Protocol** (**DHCP**) is the most convenient and, therefore, commonly used method, but this can also be achieved by adding the domain information to the `hosts` file.

When a device is joined to Azure AD, the option to join Active Directory is not available, unless you configure a hybrid Azure AD join with the domain-joined computer object as the source.

To attach a device to a previously created computer object, you'll need the credentials for an account with the privileges to attach the device to the object. If you enter the credentials for an account without the proper group membership, attaching fails.

How to do it...

To attach a Windows-based device to a previously created computer object, simply create the object or have a person with a privileged account create it for you, and then join the device, using either of the following two methods.

Using the GUI

To join a Windows 10-based device to Active Directory, perform the following steps:

1. Press Start.
2. Either click the cog on the left side of the Start menu to access **Settings**, or search for **Settings** and click the **Settings** app from the search results. Alternatively, press the *Start* button on your keyboard and the *X* key simultaneously, and then select **Settings** from the context menu. The **Settings** app appears:

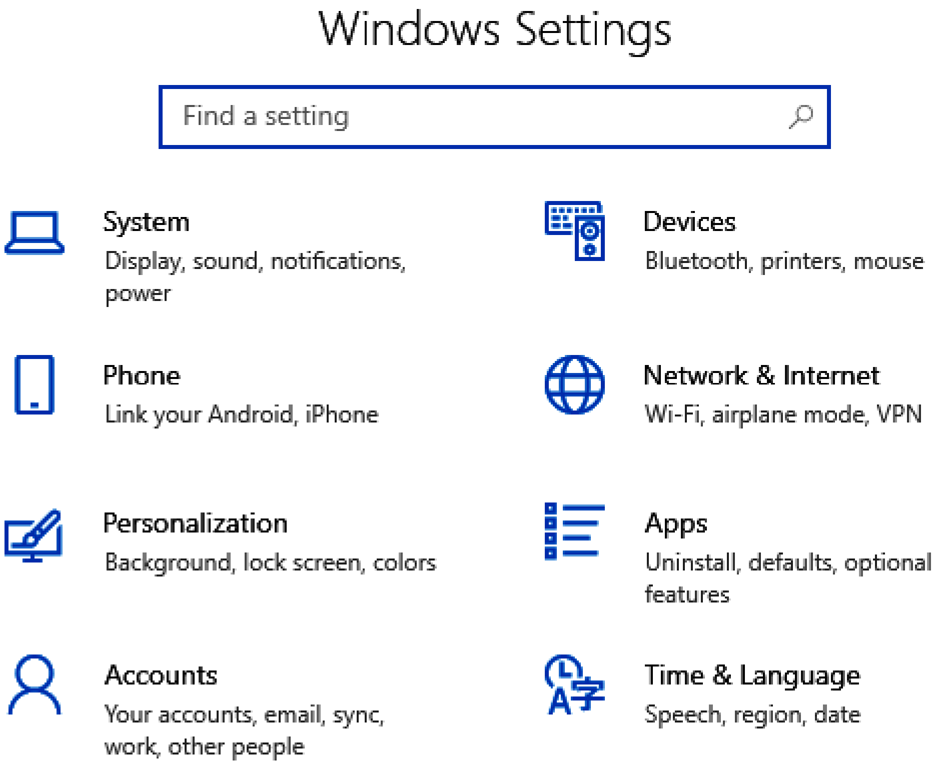

Windows Settings
Find a setting
System
Display, sound, notifications, power
Devices
Bluetooth, printers, mouse
Phone
Link your Android, iPhone
Network & Internet
Wi-Fi, airplane mode, VPN
Personalization
Background, lock screen, colors
Apps
Uninstall, defaults, optional features
Accounts
Your accounts, email, sync, work, other people
Time & Language
Speech, region, date

Figure 8.6 – The Settings app on Windows 10

3. Click **Accounts**.
4. In the left navigation pane, click **Access work or school**.
5. In the main pane, click **+ Connect**. The **Microsoft Account** window appears.
6. In the **Microsoft Account** window, click **Join this device to a local Active Directory domain** under **Alternative Options**. The **Join a Domain** window appears.
7. In the **Join a Domain** window, type the DNS domain name or the NetBIOS name of the Active Directory domain. Then, click **Next**.
8. Type the credentials in the **Windows Security** pop-up window that asks to enter your domain account information to verify that you have permission to connect to the domain. Then, click **OK**.
9. In the **Restart your PC** window, click **Restart now** to restart the PC as a domain-joined device.

Using Windows PowerShell

To add a Windows-based device to an Active Directory domain, use the following line of PowerShell:

```
Add-Computer -DomainName lucernpub.com -Credential LUCERNPUB\
Administrator -Restart
```

Replace `LUCERNPUB\Administrator` with an account having appropriate permissions in your environment. Replace the other fields to represent your environment.

How it works...

When a device is joined to Active Directory, it receives an RID through its corresponding computer object. The device has several permissions in Active Directory, including read permissions in the Active Directory **System Volume** (**SYSVOL**).

By attaching a previously created computer object, all the information for the device is already available to the device when communicating to domain controllers. From the outset, this scenario looks as though it only has advantages.

One piece of information that is needed to attach to the computer object is the secret for the object. As the process needs to work for other operating systems too, the secret is straightforward; it is the NetBIOS name of the computer in all caps, followed by a dollar sign. This represents a disadvantage because when a previously created computer object is not used, its security principle can be misused. Therefore, be sure to have a process in place to clean up stale computer objects.

There's more...

The Add-Computer PowerShell cmdlet offers many more possibilities.

For instance, the -NewName parameter can be used to provide a new hostname for the device before joining it to the Active Directory domain. This way, you may avoid joining many DESKTOP-XXXXXXX devices to Active Directory, but can definitely avoid the reboot for renaming a device.

Another real gem is the -OUPath parameter, which can be used to specify the OU that the computer object is to be placed in. This is particularly useful when you only have privileges to add computer objects to certain OUs as a delegated admin.

See also

For more information, refer to the following recipes:

- The *Creating a computer* recipe
- The *Deleting a computer* recipe
- The *Modifying the default location for new user and computer objects* recipe in *Chapter 4, Managing Containers and Organizational Units*

Renaming a computer

Use this recipe to rename a computer.

Getting ready

To rename a device's hostname, sign in to the device with administrative privileges and you need to have the (delegated) privilege in Active Directory to change the corresponding computer object.

Additionally, the device needs to be on the same logical networks as at least one of the domain controllers. You might be able to rename a device when connected through a VPN connection—when the required traffic is allowed through.

How to do it...

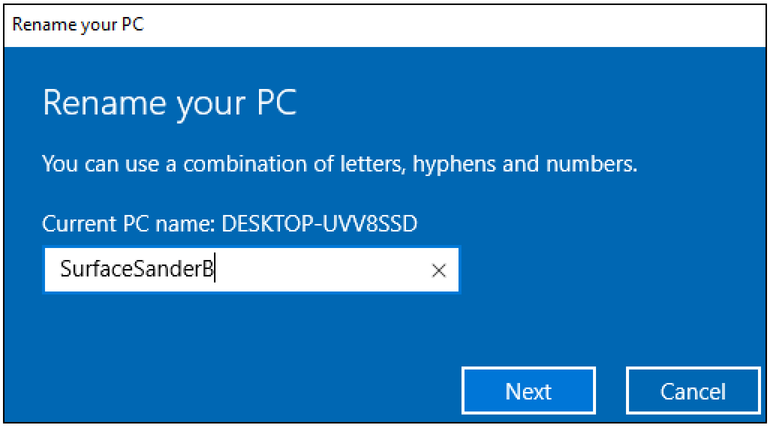
This recipe describes three ways to change the hostname for a Windows 10-based device:

- Using the **Settings** app
- Using the command line
- Using Windows PowerShell

Using the Settings app

To rename a Windows 10-based device, perform the following steps:

1. Press Start.
2. Either click the cog on the left side of the Start menu to access **Settings**, or search for **Settings** and click the **Settings** app from the search results. Alternatively, press the *Start* button on your keyboard and the *X* key simultaneously, and then select **Settings** from the context menu. The **Settings** app appears.
3. In the **Settings** window, click **System**.
4. In the left navigation pane, click **About**.
5. In the main pane, click the **Rename this PC** button under **Device Specifications**. The **Rename your PC** window appears:



Rename your PC

Rename your PC

You can use a combination of letters, hyphens and numbers.

Current PC name: DESKTOP-UVV8SSD

SurfaceSanderB

Figure 8.7 – The Rename your PC window

6. In the **Rename your PC** window, type the new hostname for the device. Then, click **Next**.
7. Click **Restart now**.

Using the command line

To rename a Windows 10-based device, issue the following command line on an elevated **Command Prompt** (`cmd.exe`):

```
netdom.exe renamecomputer localhost /newname NewComputerName /
reboot
```

Replace `NewComputerName` with the computer name you want to assign to the device.

Using Windows PowerShell

To rename a Windows 10-based device, use the following line of Windows PowerShell:

```
Rename-Computer NewComputerName -Restart
```

Replace `NewComputerName` with the computer name you want to assign to the device.

How it works...

Even when a device is domain-joined, the built-in methods to rename the device work. Before restarting to rename the device's hostname, the name of the computer object is changed to match the hostname following a restart.

When you rename the computer object for a device that is actively domain-joined, the domain trust is broken.

Any custom NetBIOS name for the computer object becomes the default name and is truncated to the first 15 characters when the hostname is longer than 15 characters.

There's more...

Using the command-line method, the credentials for the account with the (delegated) privilege in Active Directory to change the corresponding computer object can be specified. This way, you may avoid (temporarily) over-privileging a user account with privileges.

For `netdom.exe`, specify the additional `/userd`, `/passwordd`, `/usero`, and `/passwordo` parameters for the credentials for the privileged user accounts in Active Directory and the local administrator, respectively.

Testing the secure channel for a computer

Use this recipe to test the secure channel for a domain-joined device.

Getting ready

To test the secure channel, the device needs to be on the same logical networks as at least one of the domain controllers. You might be able to test the secure channel when connected through a VPN connection—when the required traffic is allowed through.

When a specific domain controller is specified, ensure the **Netlogon** service is running.

How to do it...

This recipe describes two ways to test the secure channel for a Windows 10-based device:

- Using the command line
- Using Windows PowerShell

Using the command line

To test the secure channel, enter the following command line on an elevated command prompt (`cmd.exe`):

```
nltest.exe /server:DomainControllerName /sc _ query:lucernpub.com
```

Replace `DomainControllerName` with a hostname for a reachable domain controller for the domain and replace `lucernpub.com` with the domain name.

Using Windows PowerShell

To test the secure channel, use the following line of PowerShell:

```
Test-ComputerSecureChannel
```

While `nltest.exe` provides information on the secure channel, `Test-ComputerSecureChannel` simply returns `False` or `True`. To gain more information from `Test-ComputerSecureChannel`, specify the `-Verbose` parameter.

How it works...

Domain-joined devices communicate with domain controllers using the secure channel. This way, these communications are protected using a secret. This secret is stored in Active Directory as an attribute to the computer object, and on the device as an LSA secret.

When these two values no longer match, the device is no longer able to authenticate and, thus, communicate with domain controllers.

See also

For more information, refer to the *Resetting a computer's secure channel* recipe.

Resetting a computer's secure channel

When you test a device's secure channel and it fails, the secure channel can be reset from the computer object using this recipe.

Getting ready

To reset a computer object's secret in the Active Directory object, privileges are needed to allow you to change the computer object. By default, members of the **Domain Admins** and **Account Operators** groups have this privilege. When using the Windows PowerShell method on the device itself, an account is needed that has local administrative privileges on the device and the privileges mentioned previously.

To reset the secure channel, the device needs to be on the same logical networks as at least one of the domain controllers. You might be able to rename a device when connected through a VPN connection—when the required traffic is allowed through.

When a specific domain controller is specified, ensure the **Netlogon** service is running.

How to do it...

This recipe describes four ways to reset the secure channel for a domain-joined device:

- Using **Active Directory Users and Computers**
- Using the **Active Directory Administrative Center**
- Using the command line
- Using Windows PowerShell

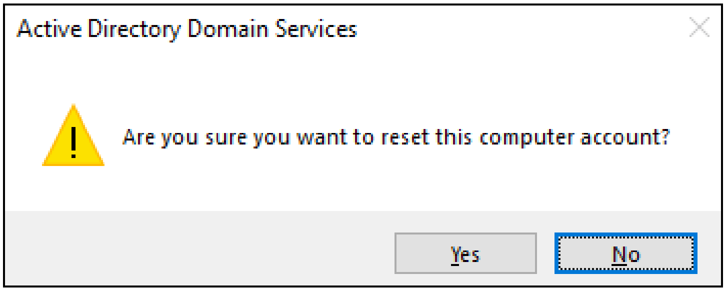
Using Active Directory Users and Computers

To reset a computer object's secure channel using **Active Directory Users and Computers**, perform these steps:

1. Press Start.
2. Search for **Active Directory Users and Computers** and select it from the search results or run `dsa.msc`. The **Active Directory Users and Computers** window appears.
3. Perform one of these series of actions:

    - In the left navigation pane, navigate to the OU or container where the computer object resides that you intend to reset. In the main menu pane, select the computer object.
    - From the **Action** menu, select **Find**. On the left-side of **Find**, select **Computers** from the drop-down menu. In the **Name** field, type the name of the computer object that you intend to reset, and then press *Enter*. From the **Search results** list, select the object.

4. Right-click the computer object and select **Reset Account** from the menu. The **Active Directory Domain Services** pop-up window appears:


Active Directory Domain Services

Are you sure you want to reset this computer account?

Figure 8.8 – The Warning pop-up window in Active Directory Domain Services

5. In the **Active Directory Domain Services** pop-up window, click **Yes** as the answer to the **Are you sure you want to reset this computer account?** question.
6. The secure channel will now be reset. Afterward, the **Active Directory Domain Services** pop-up window appears. In the **Active Directory Domain Services** pop-up window, click **OK** to acknowledge that the account was successfully reset.

Now, sign in to the device for which you reset the secure channel and rejoin it to the domain.

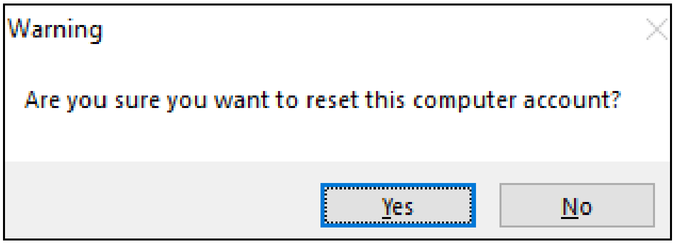
Using the Active Directory Administrative Center

To reset a computer object's secure channel using the **Active Directory Administrative Center**, perform these steps:

1. Press Start.
2. Search for **Active Directory Administrative Center** and select it from the search results or run `dsac.exe`. The **Active Directory Administrative Center** window appears.
3. Perform one of these series of actions:

   - In the left navigation pane, switch to the tree view. Navigate to the OU or container where the computer object resides that you intend to reset. In the main menu pane, select the computer object.
   - From the main menu pane, under **Global Search**, type the name of the computer object that you intend to reset, and then press *Enter*. From the list of **Global Search** results, select the computer object.

4. Right-click the computer object and select **Reset account** from the menu. The **Warning** pop-up window appears:



Warning

Are you sure you want to reset this computer account?

Figure 8.9 – The Warning pop-up window in Active Directory Domain Services

5. In the **Warning** pop-up window, click **Yes** as the answer to the **Are you sure you want to reset this computer account?** question.
6. The secure channel will now be reset. Afterward, the **Reset Computer Account Password** pop-up window appears. In the **Reset Computer Account Password** pop-up window, click **OK** to acknowledge that the account was successfully reset.

Now, sign in to the device for which you reset the computer object and rejoin it to the domain.

Using the command line

To reset a computer object's secure channel, issue the following command line on command prompt (`cmd.exe`) on a domain controller:

```
dsmod.exe computer
"CN=Computer,CN=Computers,DC=LucernPub,DC=com" -reset
```

Replace `DC=LucernPub,DC=com` to represent your environment. Replace the other fields to represent your computer's properties.

Sign in to the device for which you reset the secure channel and rejoin it to the domain.

Using Windows PowerShell

To reset the secure channel from a device that has its secure channel failed, issue the following line of PowerShell on the Windows device:

```
Test-ComputerSecureChannel -Repair
```

Now, sign in to the device for which you reset the secure channel and rejoin it to the domain.

How it works...

By resetting the secure channel for a device, the secret that is stored in Active Directory as an attribute to the computer object and on the device as an LSA secret is changed on both sides.

Another way to reset the secure channel is to remove the device from the Active Directory domain, delete the corresponding computer object, and then rejoin the device to the domain. As this is more time-consuming and requires more effort, resetting the secure channel is the preferred method.

Using **Active Directory Users and Computers**, the **Active Directory Administrative Center**, or the command line in this recipe, the secret in Active Directory is changed. The PowerShell example in this recipe represents a way to perform the action on the Windows device itself.

After changing the secret in Active Directory, the device will need to be rejoined to the domain.

Changing the default quota for creating computer objects

By default, every user object can be an owner of up to 10 computer objects. This means that every non-admin can join up to 10 devices in Active Directory. This recipe details how to change this number.

Getting ready

Sign in with an account that is a member of the **Domain Admins** group.

How to do it...

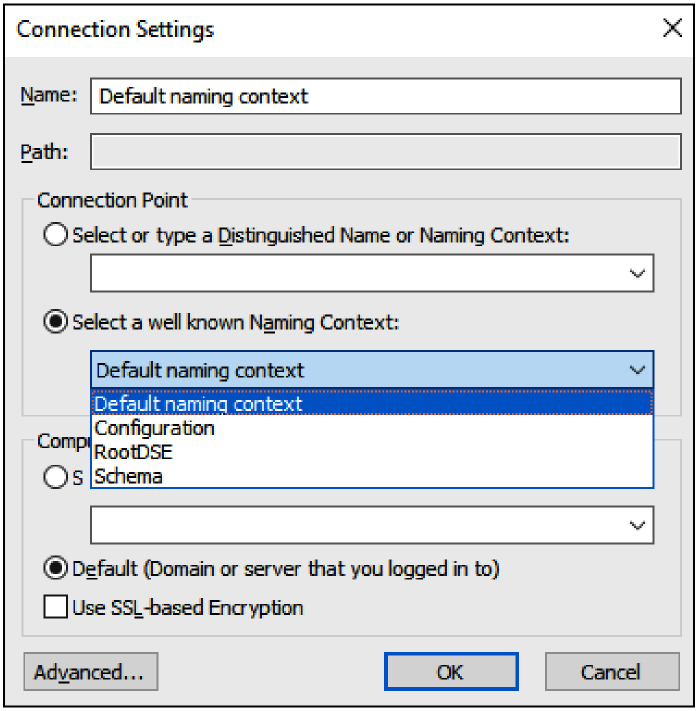
This recipe describes two ways to change the default quota for creating computer objects:

- Using the **Active Directory Service Interfaces** (**ADSI**) Edit tool
- Using Windows PowerShell

Using ADSI Edit

Perform these steps to change the default quota for creating computer objects:

1. Press Start.
2. Search for **ADSI Edit** and select it from the search results or run `adsiedit.msc`. The **ADSI Edit** window appears.
3. In the left pane, right-click the **ADSI Edit** node and select **Connect to** from the menu. The **Connection Settings** pop-up window appears:



Connection Settings

Name: Default naming context

Path:

Connection Point

○ Select or type a Distinguished Name or Naming Context:

● Select a well known Naming Context: Default naming context

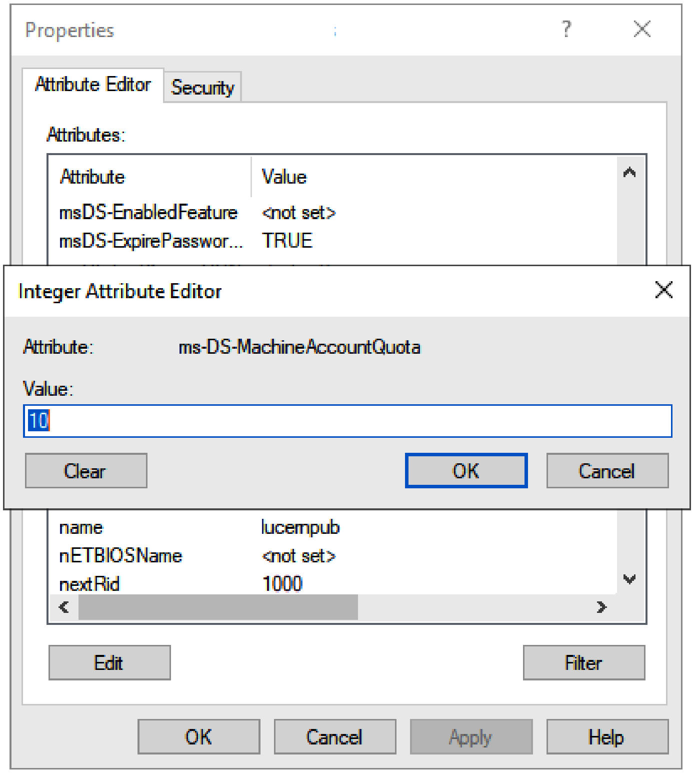
● Default (Domain or server that you logged in to)

□ Use SSL-based Encryption

Figure 8.10 – ADSI Edit's Connection Settings pop-up window

4. In the **Connection Settings** window, click the **Select a well known Naming Context** option and connect to **Default naming context**.
5. Back in the **ADSI Edit** window, in the left navigation pane, expand the **Default naming context** node.
6. Right-click the domain name and select **Properties** from the menu. The domain's properties window appears.
7. On the **Attribute Editor** tab, scroll down to the **ms-DS-MachineAccountQuota** attribute and select it.

8. Click the **Edit** button. The **Integer Attribute Editor** pop-up window appears:



Attributes:

Attribute	Value
msDS-EnabledFeature	<not set>
msDS-ExpirePasswor...	TRUE
name	lucernpub
nETBIOSName	<not set>
nextRid	1000

Integer Attribute Editor

Attribute: ms-DS-MachineAccountQuota

Value: 10

Figure 8.11 – The Integer Attribute Editor pop-up window

9. Type the new value for the attribute in the **Integer Attribute Editor** window.
10. Click **OK** to enter the new value. This closes the **Integer Attribute Editor** window.
11. Click **OK** to save the new value.
12. Close the domain's properties window.

Using Windows PowerShell

To change the default quota for creating computer objects, issue the following lines of PowerShell:

```
Set-ADDomain -Identity Lucernpub.com -Replace @{"ms-DS-
MachineAccountQuota"="Quota"}
```

Replace `Lucernpub.com` with your domain name.

How it works...

By default, every user object can be an owner of up to 10 computer objects. This means that every non-admin can join up to 10 devices in a domain. This behavior is governed by the `ms-DS-MachineAccountQuota` attribute per Active Directory domain.

When this attribute is changed to `0`, only user accounts with privileges to add computer objects to the domain are explicitly allowed to join devices to the domain. By default, only members of the **Domain Admins** and **Account Operators** groups have these privileges, but these privileges can be specifically delegated.

9
Managing DNS

The **Domain Name System** (**DNS**) is an essential service on the internet. It also plays a vital role in Active Directory. DNS offers name resolution, which enables people to navigate functionality based on names instead of IP addresses. It also enables systems to find functionality, such as domain controllers.

Misconfigured DNS records, DNS zones, and DNS servers could result in the loss of functionality, unintentional information disclosure, and an increased vulnerability toward **Meddler-in-the-Middle** (**MitM**) attacks. Domain-joined systems use DNS records to locate domain controllers. Domain controllers delegate privileges based on `serviceprincipalnames` values, which, in turn, are also based on DNS names.

Most domain controllers also offer DNS, but this is not necessary. In many complex networking infrastructures, DNS is not offered by domain controllers but by dedicated DNS servers and dedicated appliances.

The following recipes are covered in this chapter:

- Managing the DNS server role on domain controllers
- Creating a DNS zone
- Managing the DNS zone properties
- Deleting a DNS zone
- Creating a DNS record

- Deleting a DNS record
- Verifying the domain controller SRV DNS records
- Creating a DNS conditional forwarder

Managing the DNS server role on domain controllers

This recipe covers the necessary steps to add and remove the **DNS server role** from domain controllers.

Getting ready

Sign in with an account that has local administrator privileges on the domain controller.

How to do it...

Managing the DNS server role on the domain controllers consists of three steps:

1. Modifying the primary and secondary DNS server addresses on the network interface card(s)
2. Restarting the **Netlogon** service
3. Removing the DNS server role

Modifying the primary and secondary DNS server addresses on the network interface card(s)

DNS is vital to Active Directory, as most, if not all, Active Directory services rely on name resolution. The DNS server addresses configured in the domain controller's **Network Interface Card** (**NIC**) settings allow name resolution for reachability. They also allow the replication of objects within Active Directory among other required services.

To change the IP addresses for the DNS server on a domain controller, perform the following steps:

1. Press Start.
2. Search for `ncpa.cpl` and click the corresponding search result to run it. The **Network Connections** window appears.
3. Right-click the network interface you want to change, and select **Properties** from the context menu.

4. Once the **Properties** window opens, from the list of items, select **Internet Protocol Version 4 (TCP/IPv4)**.
5. Click the **Properties** button under the list.
6. The **Internet Protocol Version 4 (TCP/IPv4) Properties** window appears.
7. Select the **Use the following DNS server addresses:** radio option if it's not already selected.
8. Type in the IPv4 addresses of the dedicated DNS servers and/or appliances for the **Preferred DNS server:** and **Alternate DNS server:** fields.
9. Click **OK** to save the settings, and close the **Internet Protocol Version 4 (TCP/IPv4) Properties** window.
10. Click **Close** to close the **Properties** window for the network interface.
11. Close the **Network Connections** window.

When using Windows PowerShell, the following line in an elevated PowerShell session defines the IPv4 addresses for the primary and secondary DNS servers for the domain controller:

```
Set-DNSClientServerAddress -InterfaceAlias Ethernet
-Serveraddresses ("10.0.0.4","10.0.0.5")
```

Repeat the preceding steps on all domain controllers on which you want to change the DNS configuration.

Restarting the Netlogon service

As a domain controller registers its DNS records when the **Netlogon** service restarts, we restart the **Netlogon** service to have the domain controller register its records with the new DNS servers. To do this, perform the following steps:

1. Press Start.
2. Search for `services.msc` and click the corresponding search result to run it. The **Services** window appears.
3. In the main pane, locate the **Netlogon** service from the list of services.
4. Right-click the **Netlogon** service's name, and select **Restart** from the context menu.
5. Close the **Services** window.

When using Windows PowerShell, the following line in an elevated PowerShell session restarts the **Netlogon** service:

```
Restart-Service -Name Netlogon
```

Repeat the preceding steps on all domain controllers on which you have changed the DNS configuration.

Removing the DNS server role

Once the DNS server settings have been correctly configured, you can now add or remove the DNS server role. Perform the following steps:

1. Press Start.
2. Search for **Server Manager** and click the corresponding search result. The **Server Manager** window appears.
3. In the gray top-level bar of **Server Manager**, click **Manage**.
4. Select **Add Roles and Features** from the menu. The **Add Roles and Features Wizard** window appears.
5. On the **Before You Begin** screen, click **Next** >.
6. On the **Select installation type** screen, select **Role-based or feature-based installation**. Then, click **Next** >.
7. On the **Select destination server** screen, select the local Windows server installation from the server pool list. When done, click **Next** >.
8. On the **Select server roles** screen, select the **DNS Server** role from the list of available roles. Then, click **Next** >. A pop-up screen appears.
9. On the **Add Roles and Features Wizard** pop-up screen, click **Add Features** to add the features that are required for the DNS server role.
10. On the **Select features** screen, click **Next** >.
11. On the **DNS Server** screen, click **Next** >**.**
12. On the **Confirm installation selections** screen, click **Install**.
13. Once the configuration of the DNS server role is complete, click **Close**.

When removing the DNS server role, select **Remove Roles and Features** from the menu in *step 3*. When following the steps in this scenario, you can skip *steps 8* and *10*.

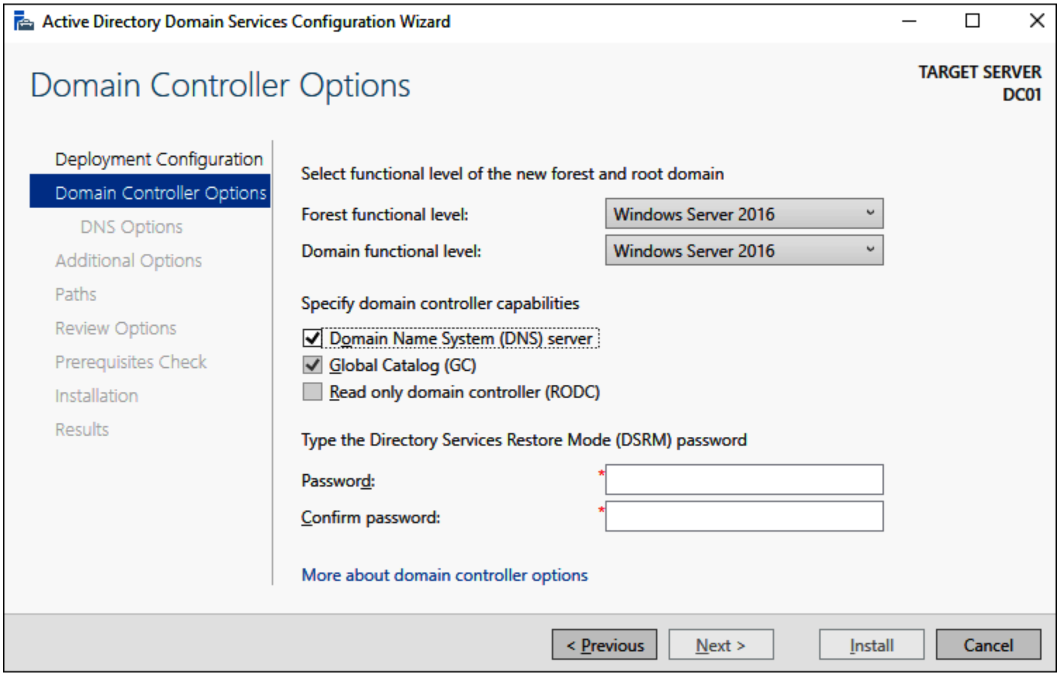
When using PowerShell, perform the following line of Windows PowerShell in an elevated PowerShell session to uninstall the DNS server role:

```
Uninstall-WindowsFeature -Name DNS
```

How it works...

Domain-joined devices rely on DNS services to locate domain controllers. DNS services don't necessarily have to run on domain controllers. However, when changing DNS services to different servers, the DNS server IP addresses should also be updated on all domain-joined devices and on all domain controllers to ensure proper name resolution and name registration.

By default, when a server is promoted to a domain controller, the **Domain Name System (DNS) server** option is checked under **Specify domain controller capabilities** on the **Domain Controller Options** page of the **Active Directory Domain Services Configuration Wizard** screen, as shown in the following screenshot:



Domain Controller Options

TARGET SERVER
DC01

Select functional level of the new forest and root domain

Forest functional level: Windows Server 2016

Domain functional level: Windows Server 2016

Specify domain controller capabilities

☑ Domain Name System (DNS) server
☑ Global Catalog (GC)
☐ Read only domain controller (RODC)

Type the Directory Services Restore Mode (DSRM) password

Password: *

Confirm password: *

More about domain controller options

Figure 9.1 – The Domain Name System (DNS) server option

See also

To make the right choices when promoting a server to a domain controller, please refer to the *Promoting a server to a domain controller* recipe in *Chapter 2*, *Managing Domain Controllers*.

If the DNS server role is added to a domain controller and has not previously been run on any domain controller, be sure to perform the steps of the following recipes in this chapter:

- The *Creating a DNS zone* recipe
- The *Managing the DNS zone properties* recipe
- The *Verifying the domain controller SRV DNS records* recipe

Creating a DNS zone

The basis of DNS is DNS zones. This recipe shows you how to create a DNS zone.

Getting ready

This recipe is applicable to DNS servers running on domain controllers, domain-joined servers, and standalone Windows Server installations. However, these scenarios require different group memberships to gain the administrator privileges that you need to perform the steps in this recipe:

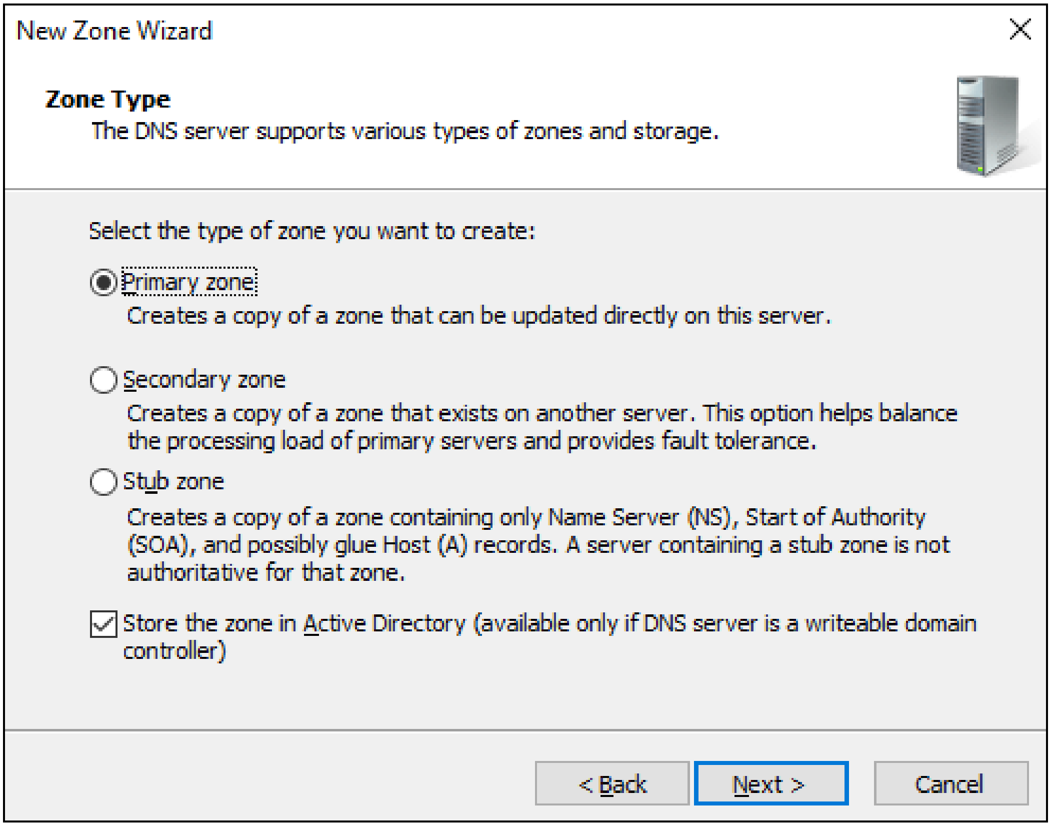
- On a DNS server that runs as a domain controller, sign in with an account that has memberships with the **DNSAdmins**, **Administrators**, **Domain Admins**, and/or **Enterprise Admins** Active Directory groups.
- On a DNS server that is either domain-joined or standalone, sign in with an account with membership to the local **Administrators** group.

How to do it…

To create a DNS zone, perform the following steps:

1. Press Start.
2. Search for `dnsmgmt.msc` and click the corresponding search result to run it. The **DNS Manager** window appears.
3. In the left-hand navigation pane, expand the DNS server's name.
4. Select either **Forward Lookup Zones** or **Reverse Lookup Zones**.

5. Right-click the selected DNS zone type node, and select **New Zone…** from the context menu. The **New Zone Wizard** window appears.
6. Click **Next** > on the **Welcome to the New Zone Wizard** screen.
7. On the **Zone Type** screen, select to create a **Primary zone** instance, a **Secondary zone** instance, or a **Stub zone** instance. When creating a DNS zone on a domain controller, verify that the **Store the zone in Active Directory (available only if DNS server is a writeable domain controller)** option is enabled if you want to have Active Directory replication for DNS zones. Then, click the **Next** > button:


New Zone Wizard

Zone Type

The DNS server supports various types of zones and storage.

Select the type of zone you want to create:

● Primary zone

Creates a copy of a zone that can be updated directly on this server.

○ Secondary zone

Creates a copy of a zone that exists on another server. This option helps balance the processing load of primary servers and provides fault tolerance.

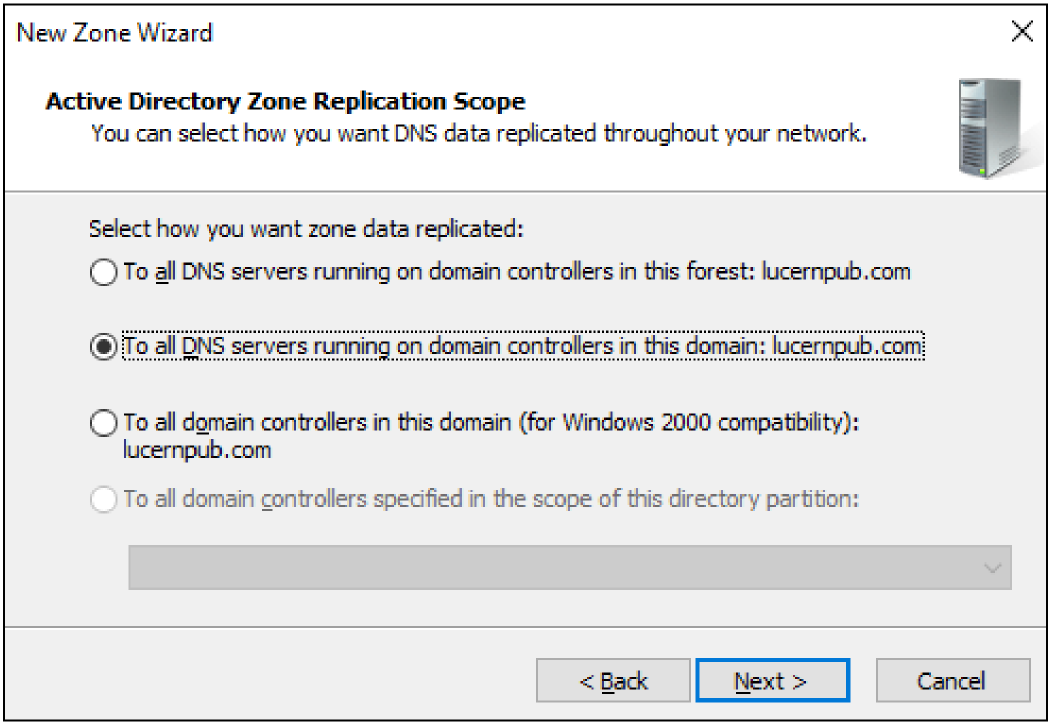
○ Stub zone

Creates a copy of a zone containing only Name Server (NS), Start of Authority (SOA), and possibly glue Host (A) records. A server containing a stub zone is not authoritative for that zone.

☑ Store the zone in Active Directory (available only if DNS server is a writeable domain controller)

Figure 9.2 – The Zone Type page of the New Zone Wizard screen

8. When you select the **Store the zone in Active Directory (available only if DNS server is a writeable domain controller)** option on the preceding screen, the **Active Directory Zone Replication Scope** screen appears:



New Zone Wizard

Active Directory Zone Replication Scope

You can select how you want DNS data replicated throughout your network.

Select how you want zone data replicated:

○ To all DNS servers running on domain controllers in this forest: lucernpub.com

● To all DNS servers running on domain controllers in this domain: lucernpub.com

○ To all domain controllers in this domain (for Windows 2000 compatibility): lucernpub.com

○ To all domain controllers specified in the scope of this directory partition:

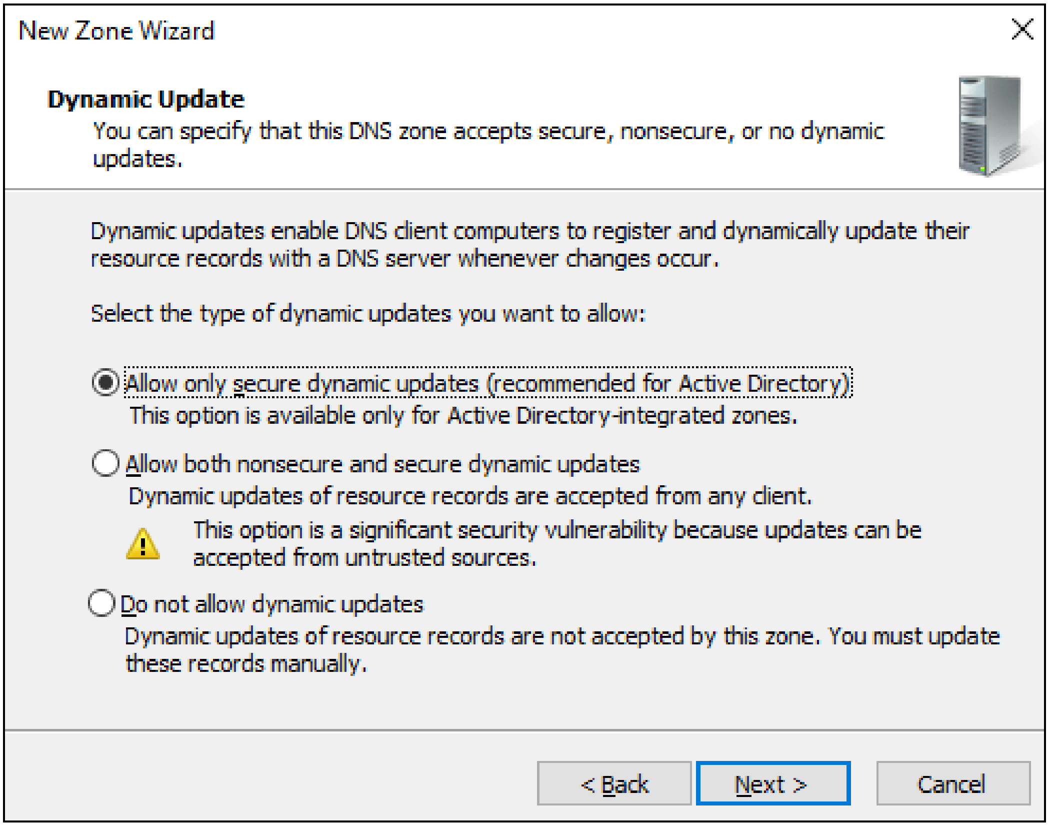
Figure 9.3 – The Active Directory Zone Replication Scope page of the New Zone Wizard screen

9. Select the **To all DNS servers running on domain controllers in this forest: <forestname>** option for **Select how you want zone date replicated:**. Then, click on the **Next** > button.

10. On the **Zone Name** screen, enter the DNS domain name in the **Zone Name:** field and click the **Next** > button.

11. If you didn't select the **Store the zone in Active Directory (available only if DNS server is a writeable domain controller)** option in *step 7*, the **Zone File** screen appears, prompting you for the name and location of where you would like to store the DNS zone file. Selecting this option stores all DNS record information in a standalone file as opposed to being integrated with Active Directory. Select the **Create a new file with this file name:** option, and click **Next** >.

12. When creating a **Secondary zone** instance, the **Master DNS servers** screen appears. Enter the IP addresses for the primary DNS servers for the DNS zone from which the secondary DNS server will be updated. The secondary DNS server holds a read-only copy of the DNS zone information.
13. When creating a primary DNS zone, the **Dynamic Update** screen appears:



New Zone Wizard

Dynamic Update

You can specify that this DNS zone accepts secure, nonsecure, or no dynamic updates.

Dynamic updates enable DNS client computers to register and dynamically update their resource records with a DNS server whenever changes occur.

Select the type of dynamic updates you want to allow:

● Allow only secure dynamic updates (recommended for Active Directory)
This option is available only for Active Directory-integrated zones.

○ Allow both nonsecure and secure dynamic updates
Dynamic updates of resource records are accepted from any client.
⚠ This option is a significant security vulnerability because updates can be accepted from untrusted sources.

○ Do not allow dynamic updates
Dynamic updates of resource records are not accepted by this zone. You must update these records manually.

Figure 9.4 – The Dynamic Update page of the New Zone Wizard screen

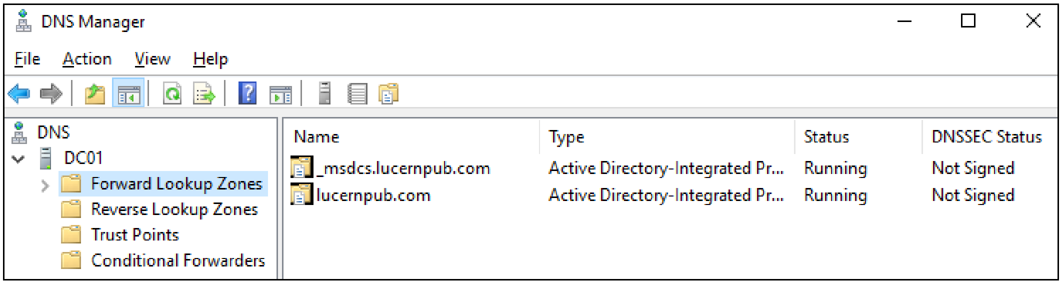
14. Verify that the **Allow only secure dynamic updates** option is enabled when creating a zone on a domain controller, and click the **Next** > button.
15. On the **Completing the New Zone Wizard** screen, click **Finish** to create the DNS zone.
16. Once complete, close the **DNS Manager** window.

How it works...

DNS zones consist of **forward lookup zones** and **reverse lookup zones**:

- A **forward lookup zone** contains one or more records (for example, A, CName, and MX) for a given DNS domain name. When a DNS name is queried, a record from the forward lookup zone provides the name resolution for an IP address.
- A **reverse lookup zone** contains one or more records (for example, PTR) for an IP address range. When an IP address is queried, a record from the reverse lookup zone provides a **Fully Qualified Domain Name** (**FQDN**) that resolves from an IP address.

By default, domain controllers with the DNS server role don't include reverse lookup zones. They include two forward lookup zones: a DNS zone for the domain name and a DNS zone for Active Directory-specific records for the domain name:



Name	Type	Status	DNSSEC Status
_msdcs.lucernpub.com	Active Directory-Integrated Pr...	Running	Not Signed
lucernpub.com	Active Directory-Integrated Pr...	Running	Not Signed

Figure 9.5 – Typical DNS zones on a domain controller

DNS services were originally created with the primary and secondary zone transfer method, for the high availability of DNS records. A **Primary zone** instance is a writable DNS zone, whereas a **Secondary zone** instance holds read-only copies of DNS record information that is replicated from the primary DNS servers. When domain controllers run the DNS server role, Active Directory replication is used to update the DNS record information across all domain controllers. The replication scope can be configured to only include domain controllers in the Active Directory domain or include all domain controllers in the entire Active Directory forest.

A **Stub zone** instance only contains resource records that identify the DNS servers for a given DNS zone. When configuring a stub zone, the IP addresses for the primary DNS servers, from which the zone is to be transferred, need to be provided. Using stub zones, you can use DNS as the name resolution mechanism underneath the Active Directory trusts.

Managing the DNS zone properties

This recipe shows how to change the DNS zone properties.

Getting ready

This recipe is applicable to DNS servers running on domain controllers, domain-joined servers, and standalone Windows Server installations. However, these scenarios require different group memberships to gain the administrator privileges you need to perform the steps in this recipe:

- On a DNS server that runs as a domain controller, sign in with an account that has memberships with the **DNSAdmins**, **Administrators**, **Domain Admins**, and/or **Enterprise Admins** Active Directory groups.
- On a DNS server that is either domain-joined or standalone, sign in with an account with membership to the local **Administrators** group.

How to do it...

To manage a DNS zone, perform the following steps:

1. Press Start.
2. Search for `dnsmgmt.msc` and click the corresponding search result to run it. The **DNS Manager** window appears.
3. In the left-hand navigation pane, expand the DNS server's name.
4. Select the DNS zone you want to manage underneath the **Forward Lookup Zones** or **Reverse Lookup Zones** node.
5. Right-click the zone. Select **Properties** from the context menu. The **Properties** window appears for the DNS zone.

You can manage the following DNS zone settings using the following tabs:

Tab	**DNS zone setting**
General	The type of DNS zone (primary, secondary, or stub)
	Whether the DNS zone is Active Directory-integrated or not
	The dynamic update settings (secure-only, nonsecure and secure, or none)
	Aging and scavenging settings to automatically remove out-of-date records
Start of Authority (SOA)	The email address of the administrator who is responsible for the DNS zone
Zone Transfers	Allow zone transfers to secondary DNS servers
Security	The delegation of control on the DNS zone and its child objects

Table 9.1 – Aspects of a DNS zone and their tabs in the Properties window

6. When finished, click the **OK** button to save the new settings. This closes the **Properties** window for the DNS zone, too.
7. Close the **DNS Manager** window.

How it works...

As a recommendation, implement Active Directory-integrated primary DNS zones, with the Active Directory forest as their scope, configured with **secure dynamic updates** only.

Deleting a DNS zone

This recipe shows how to delete a DNS zone.

Getting ready

This recipe is applicable to DNS servers running on domain controllers, domain-joined servers, and standalone Windows Server installations. However, these scenarios require different group memberships to gain the administrator privileges you need to perform the steps in this recipe:

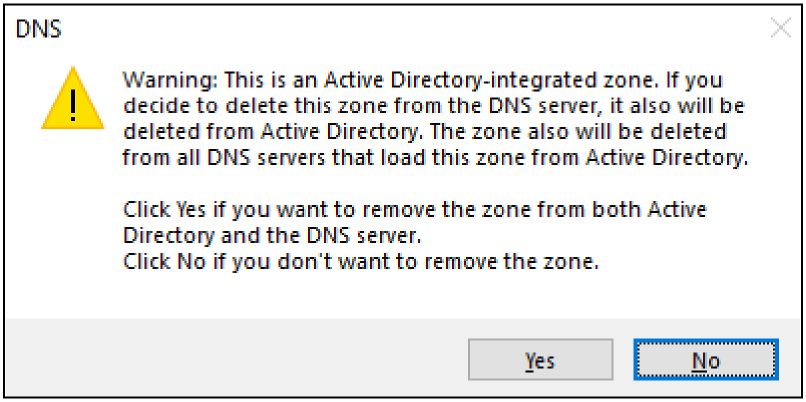
- On a DNS server that runs as a domain controller, sign in with an account that has memberships with the **DNSAdmins**, **Administrators**, **Domain Admins**, and/or **Enterprise Admins** Active Directory groups.
- On a DNS server that is either domain-joined or standalone, sign in with an account with membership to the local **Administrators** group.

How to do it...

To delete a DNS zone, perform the following steps:

1. Press Start.
2. Search for `dnsmgmt.msc` and click the corresponding search result to run it. The **DNS Manager** window appears.
3. In the left-hand navigation pane, expand the DNS server's name.
4. Select the DNS zone you want to manage underneath the **Forward Lookup Zones** or **Reverse Lookup Zones** node.
5. Right-click the zone. Select **Delete** from the context menu. A **DNS** pop-up window appears.

6. In the **DNS** pop-up window, click **Yes** to delete the zone from the server. If the zone is an Active Directory-integrated zone, a second **DNS** pop-up window appears:



DNS

Warning: This is an Active Directory-integrated zone. If you decide to delete this zone from the DNS server, it also will be deleted from Active Directory. The zone also will be deleted from all DNS servers that load this zone from Active Directory.

Click Yes if you want to remove the zone from both Active Directory and the DNS server.
Click No if you don't want to remove the zone.

Figure 9.6 – The DNS pop-up window

7. In the pop-up window, click **Yes** to remove the zone from both Active Directory and the DNS server.
8. Close the **DNS Manager** window.

How it works...

When a DNS server is no longer being queried for records, it is safe to delete the DNS zone from the DNS server and, when applicable, from Active Directory.

Creating a DNS record

An empty DNS zone allows locating domain controllers along with name resolutions for the domain name. In many organizations, other systems and services rely on DNS to be located. For this purpose, create DNS records inside the DNS zone. This recipe shows how to create a DNS record.

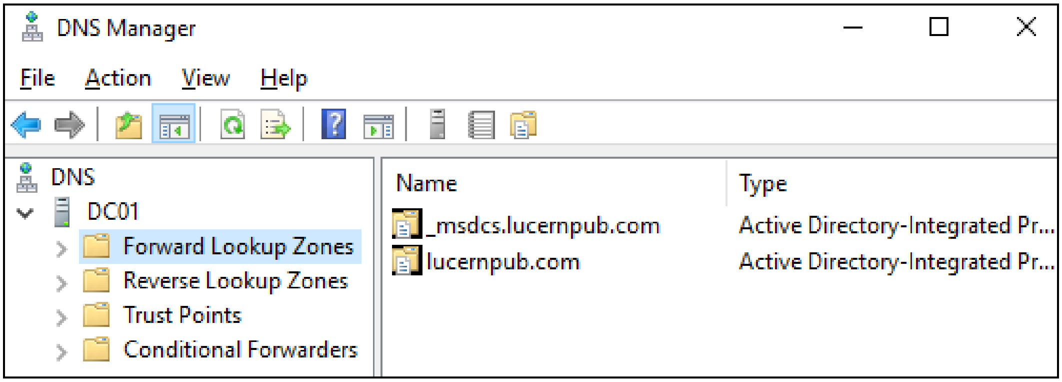
Getting ready

To create a DNS record, delegated permissions on the DNS zones in which to create the DNS record are needed. By default, membership to the **Administrators** group provides access to standalone DNS zones. For Active Directory-integrated zones, delegation can be configured for DNS zones beyond the default **DNSAdmins**, **Administrators**, **Domain Admins**, and/or **Enterprise Admins** Active Directory groups to allow the management of DNS records.

How to do it...

To create a DNS record for a DNS zone, perform the following steps:

1. Press Start.
2. Search for `dnsmgmt.msc` and click the corresponding search result to run it. The **DNS Manager** window appears:



Name	Type
_msdcs.lucernpub.com	Active Directory-Integrated Pr...
lucernpub.com	Active Directory-Integrated Pr...

Figure 9.7 – The DNS Manager window

3. In the left-hand navigation pane, expand the DNS server's name.
4. Expand and select the **Forward Lookup Zones** or **Reverse Lookup Zones** node that you wish to create a DNS record for.
5. Right-click the DNS zone node, or right-click an empty space in the main pane.
6. Select either **New Host (A or AAAA)…**, **New Alias (CNAME)…**, **New Mail Exchanger (MX)…**, or **Other New Records…** from the context menu. Choose **Other New Records…** to create the records that cannot be created using the first three menu items. A pop-up window appears, prompting you to provide the information required to create the DNS record.
7. Provide the required information for the DNS record.
8. Click the **OK** button to create the record, and close the window.
9. When done, close the **DNS Manager** window.

How it works...

Typically, a DNS zone contains DNS records for the **Start of Authority** (**SOA**) and **name servers**. These two aspects of every DNS zone are represented by SOA and NS DNS records.

There are several more types of DNS records. The most commonly used are listed as follows:

Record type	**Purpose**	**Forward lookup zone**	**Reverse lookup zone**
A AAAA	Through the A and AAAA records, the IPv4 and IPv6 addresses can be located for hosts (systems).	Yes	No
PTR	Through the PTR records, a DNS name can be located for hosts (systems) when given an IPv4 or IPv6 address.	No	Yes
CNAME	Through the CNAME records, a name alias can be provided for a host (system) inside the same DNS domain name.	Yes	No
MX	MX records point to mail servers for the DNS domain name. These records can also be used to verify a domain name with Microsoft 365.	Yes	No
TXT	TXT records provide information for the DNS domain name. These records can also be used to verify a domain name with Microsoft 365.	Yes	No
SRV	Service records can be used to locate services provided for the DNS domain name, such as _ftp, _http, and _kerberos, including priority, weight, and port.	Yes	No

Table 9.2 – Typically used DNS record types

When creating an A or AAAA record in a forward lookup zone, the **Create associated pointer (PTR) record** option is provided. To use this option, the reverse lookup zone needs to exist as well as permissions to create a record in the respective reverse lookup zone.

Deleting a DNS record

This recipe shows how to delete a DNS record.

Getting ready

To delete a DNS record, you need delegated permissions on the DNS zones from which you want to delete the DNS record. By default, membership to the **Administrators** group provides access to standalone DNS zones. For Active Directory-integrated zones, delegation can be configured for DNS zones beyond the default **DNSAdmins**, **Administrators**, **Domain Admins**, and/or **Enterprise Admins** Active Directory groups to allow you to manage DNS records.

How to do it...

To delete a DNS record for a DNS zone, perform the following steps:

1. Press Start.
2. Search for `dnsmgmt.msc` and click its corresponding search result to run it. The **DNS Manager** window appears.
3. In the left-hand navigation pane, expand the DNS server's name.
4. Expand the **Forward Lookup Zones** or **Reverse Lookup Zones** node that you wish to delete a DNS record from.
5. On the main pane, right-click the DNS record you want to delete. Select **Delete** from the context menu. A **DNS** pop-up window appears.
6. In the pop-up window, click **Yes** to delete the record from the server.
7. Close the **DNS Manager** window.

How it works...

When a system or service is decommissioned, DNS records pointing to the system and/or service may no longer be required. Delete those records to avoid the following:

- `ServicePrincipalNames`-based authentication issues when reusing names for hosts
- Name collisions when reusing IP addresses
- The slow replication of DNS records between domain controllers and DNS servers

Verifying the domain controller SRV DNS records

This recipe shows how to check whether a domain controller registers its SRV DNS records.

Getting ready

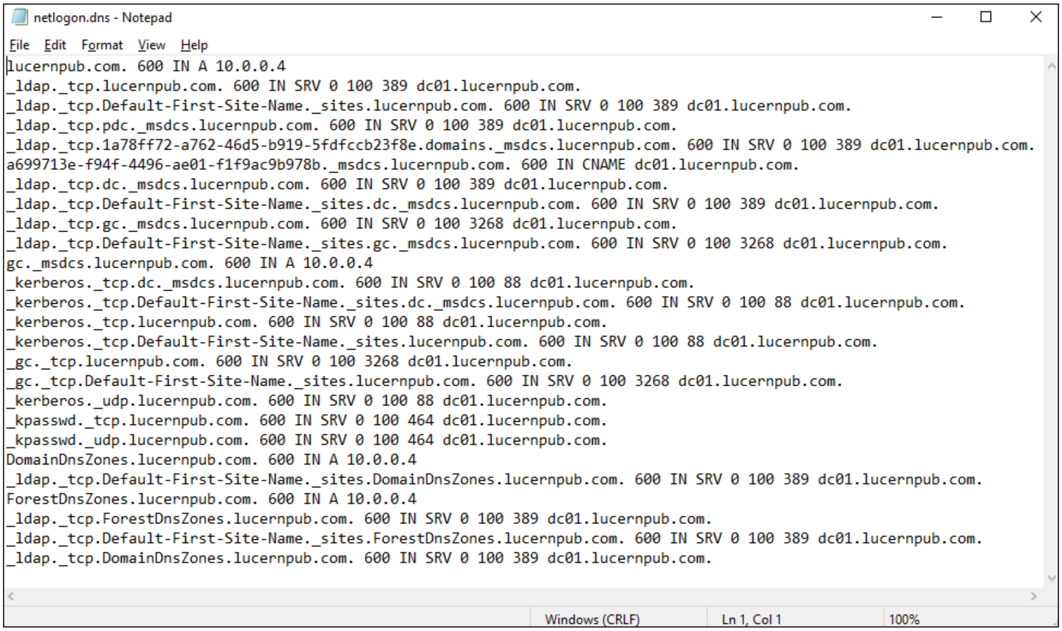
The steps in this recipe require access to the `C:\` drive of a domain controller. To complete the steps in this recipe, sign in with an account that is a member of any of the **Server Operators**, **Administrators**, **Domain Admins**, or **Enterprise Admins** Active Directory groups.

How to do it...

To check whether a domain controller registers its SRV DNS records, perform the following steps:

1. Press Start.
2. Open **File Explorer** by searching for its name. Alternatively, you can search for its executable (`explorer.exe`) in the Start menu or use the **Run...** option in the Start menu to run the executable directly. The File Explorer window appears.
3. Navigate to `C:\Windows\system32\Config`.

4. Double-click the `netlogon.dns` file. Notepad opens with the file, as follows:



```
lucernpub.com. 600 IN A 10.0.0.4
_ldap._tcp.lucernpub.com. 600 IN SRV 0 100 389 dc01.lucernpub.com.
_ldap._tcp.Default-First-Site-Name._sites.lucernpub.com. 600 IN SRV 0 100 389 dc01.lucernpub.com.
_ldap._tcp.pdc._msdcs.lucernpub.com. 600 IN SRV 0 100 389 dc01.lucernpub.com.
_ldap._tcp.1a78ff72-a762-46d5-b919-5fdfccb23f8e.domains._msdcs.lucernpub.com. 600 IN SRV 0 100 389 dc01.lucernpub.com.
a699713e-f94f-4496-ae01-f1f9ac9b978b._msdcs.lucernpub.com. 600 IN CNAME dc01.lucernpub.com.
_ldap._tcp.dc._msdcs.lucernpub.com. 600 IN SRV 0 100 389 dc01.lucernpub.com.
_ldap._tcp.Default-First-Site-Name._sites.dc._msdcs.lucernpub.com. 600 IN SRV 0 100 389 dc01.lucernpub.com.
_ldap._tcp.gc._msdcs.lucernpub.com. 600 IN SRV 0 100 3268 dc01.lucernpub.com.
_ldap._tcp.Default-First-Site-Name._sites.gc._msdcs.lucernpub.com. 600 IN SRV 0 100 3268 dc01.lucernpub.com.
gc._msdcs.lucernpub.com. 600 IN A 10.0.0.4
_kerberos._tcp.dc._msdcs.lucernpub.com. 600 IN SRV 0 100 88 dc01.lucernpub.com.
_kerberos._tcp.Default-First-Site-Name._sites.dc._msdcs.lucernpub.com. 600 IN SRV 0 100 88 dc01.lucernpub.com.
_kerberos._tcp.lucernpub.com. 600 IN SRV 0 100 88 dc01.lucernpub.com.
_kerberos._tcp.Default-First-Site-Name._sites.lucernpub.com. 600 IN SRV 0 100 88 dc01.lucernpub.com.
_gc._tcp.lucernpub.com. 600 IN SRV 0 100 3268 dc01.lucernpub.com.
_gc._tcp.Default-First-Site-Name._sites.lucernpub.com. 600 IN SRV 0 100 3268 dc01.lucernpub.com.
_kerberos._udp.lucernpub.com. 600 IN SRV 0 100 88 dc01.lucernpub.com.
_kpasswd._tcp.lucernpub.com. 600 IN SRV 0 100 464 dc01.lucernpub.com.
_kpasswd._udp.lucernpub.com. 600 IN SRV 0 100 464 dc01.lucernpub.com.
DomainDnsZones.lucernpub.com. 600 IN A 10.0.0.4
_ldap._tcp.Default-First-Site-Name._sites.DomainDnsZones.lucernpub.com. 600 IN SRV 0 100 389 dc01.lucernpub.com.
ForestDnsZones.lucernpub.com. 600 IN A 10.0.0.4
_ldap._tcp.ForestDnsZones.lucernpub.com. 600 IN SRV 0 100 389 dc01.lucernpub.com.
_ldap._tcp.Default-First-Site-Name._sites.ForestDnsZones.lucernpub.com. 600 IN SRV 0 100 389 dc01.lucernpub.com.
_ldap._tcp.DomainDnsZones.lucernpub.com. 600 IN SRV 0 100 389 dc01.lucernpub.com.
```

Figure 9.8 – The typical contents of a netlogon.dns file

5. The `netlogon.dns` file contains all of the DNS records that the domain controller registers when the **Netlogon** service starts or restarts. Verify the following:

   I. The domain name is present.

   II. The domain controller registers the `_ldap._tcp` and `_kerberos._tcp` SRV records for the domain name and for the Active Directory site it is in.

   III. The domain controller registers the `_kerberos._udp`, `_kpasswd._tcp`, and `_kpasswd.udp` SRV records for the domain name.

   IV. The domain controller registers the `_ldap._tcp _pdc` SRV records when it holds the **Primary Domain Controller Emulator** (**PDCE**) **Flexible Single Master Operations** (**FSMO**) role.

   V. The domain controller registers the `_ldap._tcp gc` and `_gc._tcp` SRV records when it is configured as a **Global Catalog** (**GC**) server.

   VI. The domain controller registers the **DomainDNSZones** and **ForestDNSZones** records for the domain name and for the Active Directory site it is in when it is configured with Active Directory-integrated DNS zones.

6. Close the `netlogon.dns` file.
7. Restart the **Netlogon** service.

How it works...

To locate Active Directory Domain Services, the domain controllers add DNS SRV records to the forward lookup zone representing the Active Directory domain name. The `netlogon.dns` file is used by the **Netlogon** service for this purpose. The file contains the information required to register the services offered by the domain controller. This file should contain all the records for the services offered.

Creating a DNS conditional forwarder

For an Active Directory trust, hosts in one Active Directory domain or forest need to be able to resolve names in another Active Directory domain or forest, and vice versa. For this purpose, you can create conditional forwarders in DNS.

Create conditional forwarders on a domain controller in each Active Directory domain or forest.

Getting ready

This recipe is applicable to DNS servers running on domain controllers, domain-joined servers, and standalone Windows Server installations. However, these scenarios require different group memberships to gain the administrator privileges that you need to perform the steps in this recipe:

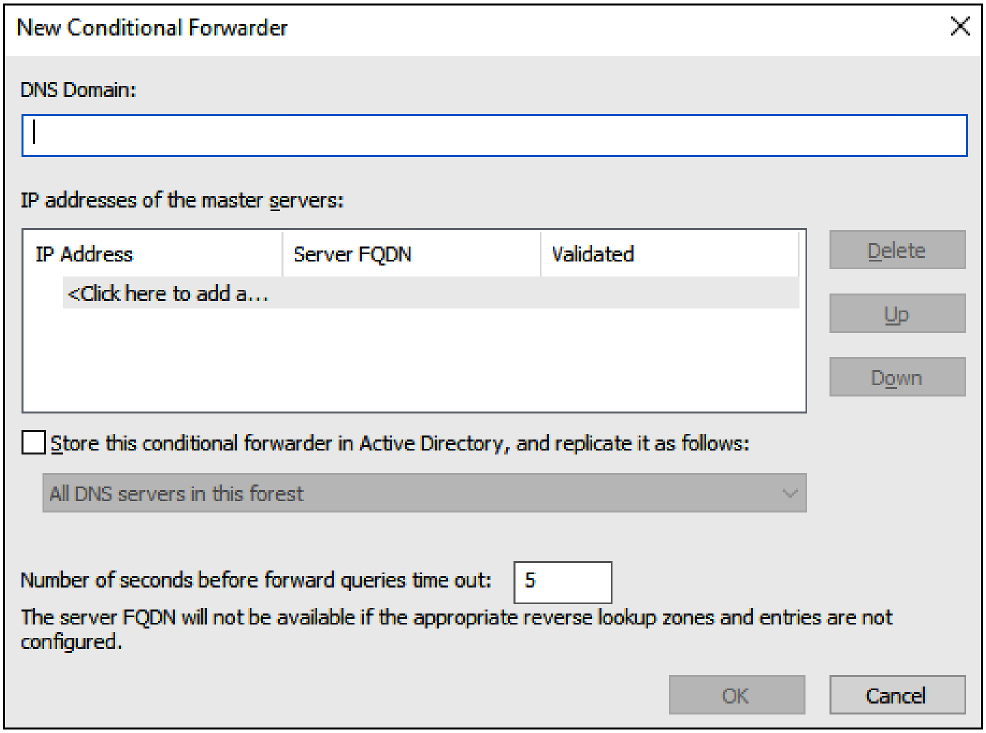
- On a DNS server that runs as a domain controller, sign in with an account that has memberships with the **DNSAdmins**, **Administrators**, **Domain Admins**, and/or **Enterprise Admins** Active Directory groups.
- On a DNS server that is either domain-joined or standalone, sign in with an account with membership to the local **Administrators** group.

How to do it...

To create a **conditional forwarder** in DNS, perform the following steps:

1. Press Start.
2. Search for `dnsmgmt.msc` and click its corresponding search result to run it. The **DNS Manager** window appears.
3. In the left-hand navigation pane, expand the DNS server's name.

4. Right-click the **Conditional Forwarders** node, and select **New Conditional Forwarder…** from the context menu. The **New Conditional Forwarder** window appears:



DNS Domain:

IP addresses of the master servers:

<Click here to add a…

□ Store this conditional forwarder in Active Directory, and replicate it as follows:

All DNS servers in this forest

Number of seconds before forward queries time out: 5

The server FQDN will not be available if the appropriate reverse lookup zones and entries are not configured.

Figure 9.9 – The New Conditional Forwarder screen

5. In the **DNS Domain:** field, enter the DNS domain name that you want to resolve records for. In this example, it would be the Active Directory domain you want to create a trust with.
6. In the **IP addresses of the master servers:** list, add the IP addresses of the reachable domain controllers for the Active Directory domain you want to create a trust with.
7. If the DNS server is also a domain controller, select the **Store this conditional forwarder in Active Directory, and replicate it as follows:** option. Select an appropriate scope to accept the default scope of **All DNS servers in this forest**.
8. Click **OK** to create the conditional forwarder, and close the **New Conditional Forwarder** window.

Repeat the steps of this recipe on the DNS server or domain controller for the Active Directory domain or forest to trust, to configure two-way name resolutions between the Active Directory domains or forests.

How it works...

Conditional forwarders instruct the DNS server to forward queries for hosts with a certain DNS domain name suffix to a predefined list of IP addresses representing DNS servers for the DNS domain name.

To create a conditional forwarder, the IP addresses for the (domain controllers acting as) DNS servers need to be added. When these IP addresses change, the conditional forwarder needs to be updated to reflect the IP address changes.

Conditional forwarders can be stored in Active Directory and replicated to all domain controllers acting as DNS servers. Changes to the configuration of the conditional forwarder can be made centrally.

See also

To create the right Active Directory trust, please refer to the *Creating the right Active Directory trust* recipe in *Chapter 1*, *Optimizing Forests, Domains, and Trusts*.

10 Getting the Most Out of Group Policy

Group Policy allows administrators to manage one device or many thousands of devices and/or servers through a centralized management console. You can use it to secure domain-joined devices, make them more useful for end users, and make them look and feel identical as per your organization's standards.

Granularity in Group Policy objects offers you the ability to manage these settings for devices in the entire Active Directory domain, for an Active Directory site (but please refrain from linking Group Policy objects on the site-level), per **Organizational Unit** (**OU**), and beyond that by using **Windows Management Instrumentation** (**WMI**) filters.

Group Policy has been around since the beginning of Active Directory in Windows 2000 Server. It has seen many improvements in the last two decades, such as Group Policy preferences and many new settings relating to all the new client and server operating system possibilities.

Although currently considered archaic because of trends such as bring your own device and solutions such as **Mobile Device Management** (**MDM**) and **Mobile Application Management** (**MAM**), Group Policy should be considered an essential tool in every Active Directory admin's toolbox.

The following recipes are covered in this chapter:

- Creating a **Group Policy Object** (**GPO**)
- Copying a GPO
- Deleting a GPO
- Modifying the settings of a GPO
- Assigning scripts
- Installing applications
- Linking a GPO to an OU
- Blocking inheritance of GPOs on an OU
- Enforcing the settings of a GPO link
- Applying security filters
- Creating and applying WMI Filters
- Configuring loopback processing
- Refreshing GPO settings
- Restoring a default GPO
- Creating a Group Policy Central Store

Creating a GPO

This recipe shows how to create a GPO.

Getting ready

To create a GPO, sign in to a system with the **Group Policy Management** feature installed with an account that is either of the following:

- A member of the **Domain Admins** group
- Delegated to create GPOs

Besides permission to create GPOs, additional permissions are needed to use a Starter GPO or link a GPO to an OU.

How to do it…

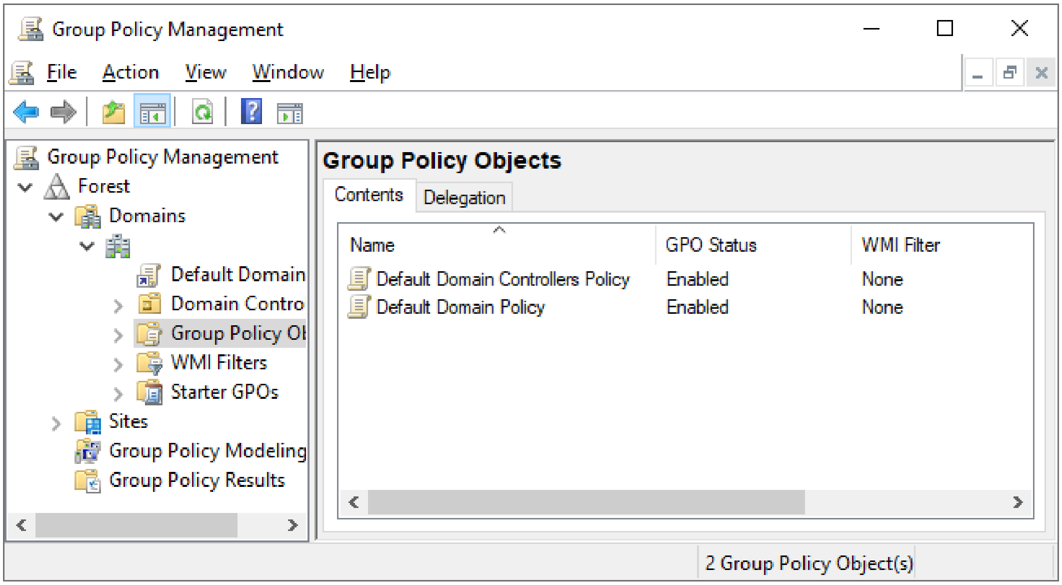
This recipe shows two ways to create a GPO:

- Using **Group Policy Management**
- Using Windows PowerShell

Using Group Policy Management

To create a GPO, perform the following steps:

1. Press Start.
2. Search for **Group Policy Management** and click its search result, or run `gpmc.msc`. The **Group Policy Management** window appears.
3. In the left navigation pane, expand the **Forest** node.
4. Expand the **Domains** node, and then navigate to the domain where you want to create the GPO.
5. Expand the domain name and select the **Group Policy Objects** node:



Name	GPO Status	WMI Filter
Default Domain Controllers Policy	Enabled	None
Default Domain Policy	Enabled	None

Figure 10.1 – The Group Policy Objects node in Group Policy Management

6. Right-click the **Group Policy Objects** node and select **New** from the menu. The **New GPO** pop-up window appears.

7. In the **New GPO** pop-up window, enter the name of the GPO in the **Name:** field. Make sure that you don't select **Source starter GPO**.
8. Click **OK** to create the GPO.

You now have a GPO. However, it doesn't have any settings, and it isn't linked – at least not yet.

Using Windows PowerShell

To create a GPO using Windows PowerShell, use the following line of PowerShell:

```
New-GPO -Name "New GPO Name"
```

Replace `New GPO Name` with the name of the new GPO.

How it works...

Group Policy Management is the tool of choice to manage GPOs.

Regardless of the tool used, GPOs are created in the `Policies` container in Active Directory.

Group Policy Management is a great tool to use when creating one GPO. When creating multiple GPOs or automating creating GPOs, Windows PowerShell proves more efficient.

See also

To get things going with the GPO, refer to the *Linking a GPO to an OU* recipe.

Copying a GPO

This recipe shows how to copy an existing GPO to a new GPO. This is useful when you have configured the perfect GPO but want to adjust just a single setting to apply to legacy versions of the operating system or a test version in another OU.

Getting ready

To copy a GPO, sign in to a system with the **Group Policy Management** feature installed with an account that is either of the following:

- A member of the Domain Admins group
- Delegated to create GPOs and has at least `read` permissions on the GPO to be copied

How to do it...

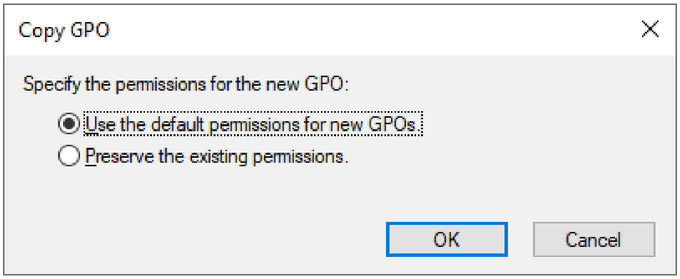
This recipe shows two ways to copy a GPO:

- Using **Group Policy Management**
- Using Windows PowerShell

Using Group Policy Management

To copy a GPO, perform the following steps:

1. Press Start.
2. Search for **Group Policy Management** and click its search result, or run `gpmc.msc`. The **Group Policy Management** window appears.
3. In the left navigation pane, expand the **Forest** node.
4. Expand the **Domains** node, and then navigate to the domain where you want to copy a GPO.
5. Expand the domain name and then expand the **Group Policy Objects** node.
6. Locate the GPO that you want to copy.
7. Right-click the GPO and select **Copy** from the menu.
8. Right-click the **Group Policy Objects** node and select **Paste** from the menu. The **Copy GPO** pop-up window appears:



Copy GPO

Specify the permissions for the new GPO:

● Use the default permissions for new GPOs.

○ Preserve the existing permissions.

Figure10.2 – The Copy GPO pop-up window

9. In the **Copy GPO** pop-up window, select either the **Preserve the existing permissions.** or the **Use the default permissions for new GPOs.** option.
10. Click **OK**. The **Copy** window appears.
11. In the **Copy** window, click **OK** to acknowledge that copying has succeeded.

12. The new GPO is now named **Copy of…**, referencing the source GPO name. Right-click the GPO in the left navigation menu or the main pane and select **Rename** from the menu.
13. Enter the new name of the GPO. Press *Enter* when done.

Using Windows PowerShell

To copy a GPO using Windows PowerShell, use the following line of PowerShell:

```
Copy-GPO -Sourcename "Existing GPO Name" -TargetName "New GPO
Name"
```

Replace `Existing GPO Name` and `New GPO Name` with the names of the GPO to copy and the new GPO. Optionally, add the `-CopyAcl` parameter to switch from the **Use the default permissions for new GPOs.** option to the **Preserve the existing permissions.** option.

How it works…

Copying a GPO is a sure way to create a GPO that is slightly different from an existing GPO.

After copying a GPO, only the name is different when you choose the **Preserve the existing permissions.** option. If you choose the **Use the default permissions for new GPOs.** option, your permissions on top of the default permissions apply, and the account used to copy the GPO would be the Group Policy owner.

There's more…

In a multi-domain Active Directory forest, copying a GPO between the **Group Policy Objects** node in one domain to the **Group Policy Objects** node in another domain is a sure way to keep settings identical across domains. Copying GPOs is especially useful in networking environments consisting of development, testing, acceptance, and production environments to stage GPOs.

Deleting a GPO

This recipe shows how to delete a GPO. As part of this recipe, any GPO links that are present are first deleted to ensure that no stale references occur in multi-domain environments.

Getting ready

To delete a GPO, sign in to a system with the **Group Policy Management** feature installed with an account that is either of the following:

- A member of the **Domain Admins** group
- The current owner of the GPO
- Delegated the **Edit settings, delete, modify security** permission on the GPO

How to do it...

This recipe shows two ways to delete a GPO:

- Using **Group Policy Management**
- Using Windows PowerShell

Using Group Policy Management

To delete a GPO, perform the following steps:

1. Press Start.
2. Search for **Group Policy Management** and click its search result, or run `gpmc.msc`. The **Group Policy Management** window appears.
3. In the left navigation pane, expand the **Forest** node.
4. Expand the **Domains** node, and then navigate to the domain where you want to delete the GPO.
5. Expand the domain name and then expand the **Group Policy Objects** node.
6. Locate the GPO that you want to delete and select it.
7. In the main pane, on the **Scope** tab, check that the GPO is not linked to any GPO by inspecting the field for **The following sites, domains and OUs are linked to this GPO:**. If it's not empty, the GPO is still linked to OUs, sites, and/or domains. Remove these links by right-clicking them and selecting **Delete Link(s)** from the menu before deleting the GPO. In the **Group Policy Management** pop-up window, click **OK**.
8. In the left navigation pane, right-click the GPO and select **Delete** from the menu. The **Group Policy Management** pop-up window appears.
9. In the **Group Policy Management** pop-up window, click **Yes** to answer the **Do you want to delete this GPO and all links to it in this domain? This will not delete links in other domains.** question.

Using Windows PowerShell

To delete a GPO using Windows PowerShell, use the following line of PowerShell:

```
Remove-GPO -Name "Existing GPO Name"
```

Replace `Existing GPO Name` with the name of the GPO to delete.

How it works...

A GPO not linked to an OU, site, or domain and not part of the development or testing process cannot be justified and can be deleted without repercussions.

When you delete a GPO, its links are not automatically deleted when the user account performing the deletion, does not have the permissions to manage the link, or when the link refers to another domain in the forest. GPO links should be removed before deleting a GPO to avoid lingering links.

See also

For more information, refer to the *Linking a GPO to an OU* recipe.

Modifying the settings of a GPO

This recipe shows how to modify settings in a GPO.

Getting ready

To manage settings in a GPO, sign in to a system with the **Group Policy Management** feature installed with an account that is either of the following:

- A member of the **Domain Admins** group
- The current owner of the GPO
- Delegated the **Edit settings** or **Edit settings, delete, modify security** permission on the GPO

How to do it...

Perform the following steps:

1. Press Start.
2. Search for **Group Policy Management** and click its search result or run `gpmc.msc`. The **Group Policy Management** window appears.

3. In the left navigation pane, expand the **Forest** node.
4. Expand the **Domains** node, and then navigate to the domain where you want to modify the GPO.
5. Expand the domain name and then expand the **Group Policy Objects** node.
6. Locate the GPO that you want to manage and select it.
7. In the main pane, on the **Settings** tab, inspect the settings. Use the **show**, **hide**, and **show all** buttons to display the settings under their respective Group Policy setting nodes.
8. In the left navigation pane, right-click the GPO and select **Edit…** from the menu.
9. The **Group Policy Management Editor** window appears:

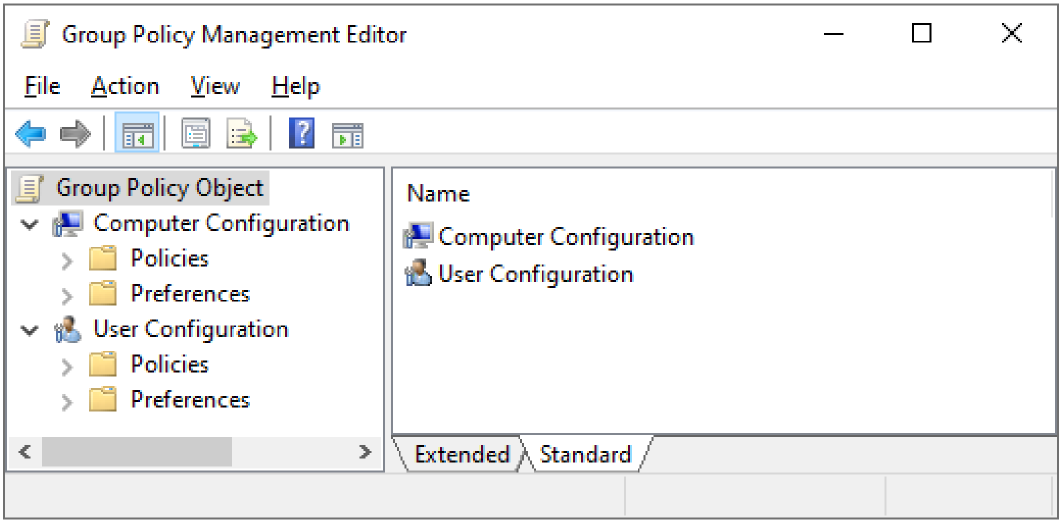


Figure 10.3 – The Group Policy Management Editor window

10. Edit the settings and/or preferences you want to edit in the **Group Policy Management Editor** window.
11. Close the **Group Policy Management Editor** window to save the settings.

How it works…

When you manage the settings and/or preferences in a GPO, they become active immediately. There are slight delays due to replication and the Group Policy background refresh timeout (90 minutes by default but with a 30-minute window of additional time, or 5 minutes for GPOs applied to domain controllers).

Some settings and/or preference patterns cause GPOs to apply slowly. This might slow down device startups and user sign-ins. Here are some tips:

- Avoid configuring settings and policies in one GPO.
- Avoid setting the **Computer Configuration** and **User Configuration** settings in one GPO, unless the GPO targets Remote Desktop Session Hosts.
- When you create a GPO with only **Computer Configuration**, disable **User Configuration** processing, and vice versa.

Assigning scripts

This recipe shows how to assign a logon script using Group Policy.

Getting ready

To manage settings for a GPO, sign in to a system with the **Group Policy Management** feature installed with an account that is either of the following:

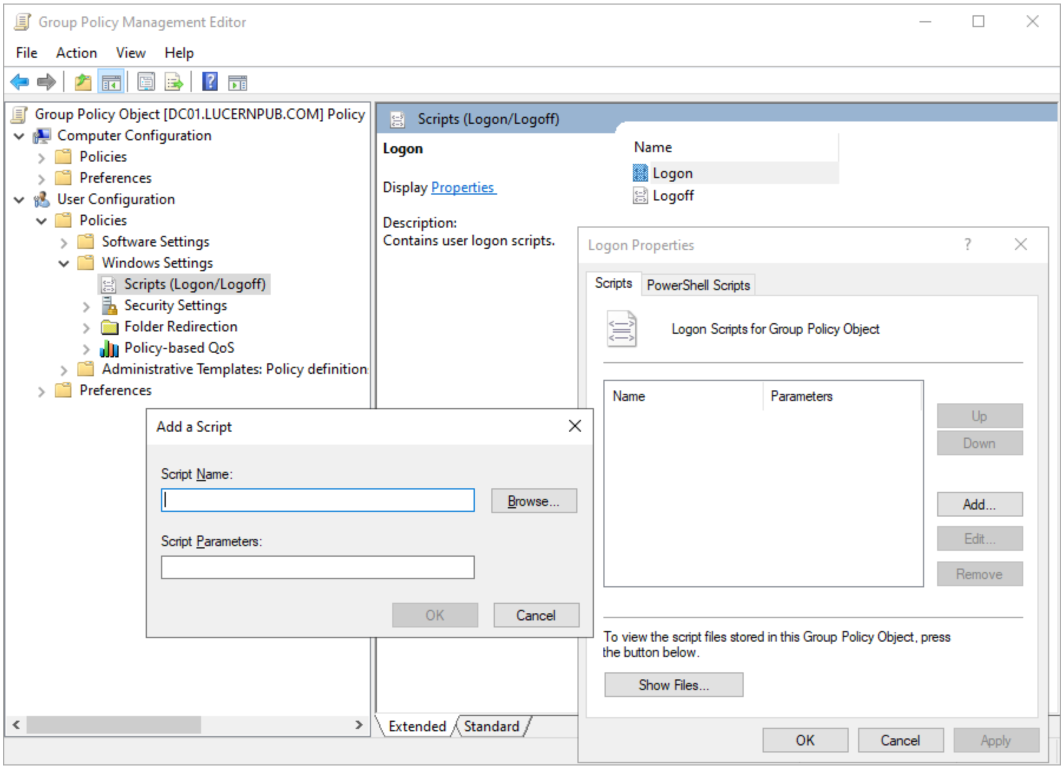
- A member of the **Domain Admins** group
- The current owner of the GPO
- Delegated the **Edit settings** or **Edit settings, delete, modify security** permission on the GPO

How to do it...

Perform the following steps:

1. Press Start.
2. Search for **Group Policy Management** and click its search result, or run `gpmc.msc`. The **Group Policy Management** window appears.
3. In the left navigation pane, expand the **Forest** node.
4. Expand the **Domains** node, and then navigate to the domain where you want to assign a logon script.
5. Expand the domain name and then expand the **Group Policy Objects** node.
6. Locate the GPO that you want to manage and select it.
7. In the left navigation pane, right-click the GPO and select **Edit…** from the menu. The **Group Policy Management Editor** window appears.

8. In the **Group Policy Management Editor** window, expand **User Configuration**, then **Policies**, and **Windows Settings**.
9. Select **Scripts**.
10. In the main pane, right-click the **Logon** script and select **Properties** from the menu. The **Logon Properties** window appears.
11. Click the **PowerShell Scripts** tab to use a Windows PowerShell script as the logon script.
12. In the **Logon Properties** window, click the **Add…** button. The **Add a Script** pop-up window appears:



Description:
Contains user logon scripts.

To view the script files stored in this Group Policy Object, press the button below.

Figure 10.4 – The Add a Script pop-up window

13. Type the executable in the **Script Name:** field or browse to its location.
14. In the **Script Parameters:** field, type any optional script parameters.
15. Click **OK** to save the script settings.

16. Click **OK** to close the **Logon Properties** window.
17. Close the **Group Policy Management Editor** window.

How it works...

This recipe describes four types of script you can use with Group Policy:

- Logon scripts (**User Configuration**)
- Logoff scripts (**User Configuration**)
- Startup scripts (**Computer Configuration**)
- Shutdown scripts (**Computer Configuration**)

It is good practice to reference a script name in the Active Directory **System Volume** (**SYSVOL**). This way, the unavailability of a server doesn't mean the unavailability of scripts used to manage (part of) an environment. By default, when you type a script into the **Script Name:** field, Group Policy assumes that the script is located in the `Scripts` folder in the SYSVOL.

Windows PowerShell scripts require at least Windows 7 or Windows Server 2008 R2.

Installing applications

This recipe shows how to install an application using Group Policy.

Getting ready

To manage settings for a GPO, sign in to a system with the **Group Policy Management** feature installed with an account that is either of the following:

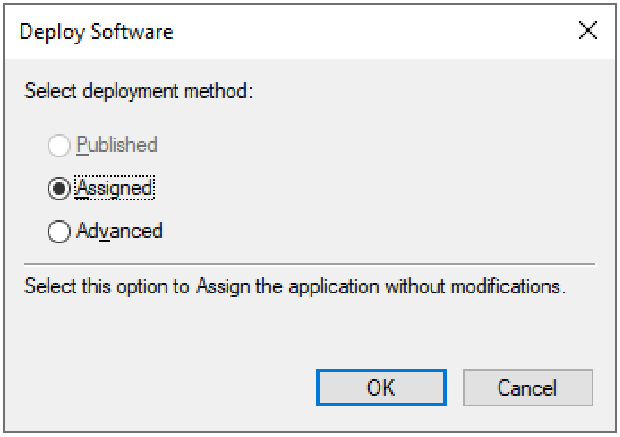
- A member of the **Domain Admins** group
- The current owner of the GPO
- Delegated the **Edit settings** or **Edit settings, delete, modify security** permission on the GPO

How to do it...

Perform the following steps:

1. Press Start.
2. Search for **Group Policy Management** and click its search result, or run `gpmc.msc`. The **Group Policy Management** window appears.

3. In the left navigation pane, expand the **Forest** node.
4. Expand the **Domains** node, and then navigate to the domain where you want to install the application.
5. Expand the domain name and then expand the **Group Policy Objects** node.
6. Locate the GPO that you want to manage and select it.
7. In the left navigation pane, right-click the GPO and select **Edit…** from the menu. The **Group Policy Management Editor** window appears.
8. Expand **Computer Configuration** or **User Configuration**, depending on the kind of object that is the target of the software installation.
9. Expand the **Policies** node, and then the **Software Settings** node.
10. Right-click the **Software Installation** node, select **New**, and then **Package…** from the menu. The **Open** window appears.
11. In the **Open** window, type the **Universal Naming Convention** (**UNC**) path of the network share that has the package for the application with the `.msi` extension – for instance, `\\server.domain.tld\share`. Select the application and click **Open**. The **Deploy Software** pop-up window appears:



Select deployment method:

○ Published

● Assigned

○ Advanced

Select this option to Assign the application without modifications.

Figure 10.5 – The Deploy Software pop-up window for User Configuration

12. In the **Deploy Software** pop-up window, select the deployment method.
13. Click **OK** to save the settings. The package will be listed with its version, its deployment state, and source path.
14. Close the **Group Policy Management Editor** window.

How it works...

Applications can be distributed to devices and user accounts using Group Policy. However, more advanced solutions are used in medium and large organizations, such as Microsoft Endpoint Manager, because of the lack of advanced features, controls, and reporting in Group Policy.

To install an application package, point to the package on a shared folder, which is accessible to the objects you target the GPO to. For users, the default **Everyone – Full Control** share permissions suffice, and default folders on Windows Server installations have **Users – Read NT File System** (**NTFS**) permissions. However, when deploying a package to devices, ensure that the UNC path is accessible to domain computers with read permissions.

Assigned applications will deploy at logon (when this option is selected) or when a user clicks the shortcut for the application or a filetype associated with the application. Published applications (only available under **User Configuration**) will be installed when a user clicks the shortcut for the application or a filetype associated with the application. Published applications can also be installed using **Apps and Features** on domain-joined devices.

Software installation settings are not processed when slow links are detected.

The **Advanced** button can be used to further specify settings for application installation, including but not limited to the presence of other application installations and versioning requirements.

Linking a GPO to an OU

This recipe shows how to link an existing GPO to an OU.

Getting ready

To link a GPO to an OU, sign in to a system with the **Group Policy Management** feature installed with an account that is either of the following:

- A member of the **Domain Admins** group
- The current owner of the GPO, and has the **Link GPOs** permission on the OU(s), site(s), and/or domain(s) where the GPO is to be linked
- Delegated the **Edit settings** or **Edit settings, delete, modify security** permission on the GPO, and has the **Link GPOs** permission on the OU(s), site(s), and/or domain(s) where the GPO is to be linked

How to do it…

Perform the following steps:

1. Press Start.
2. Search for **Group Policy Management** and click its search result or run `gpmc.msc`. The **Group Policy Management** window appears.
3. In the left navigation pane, expand the **Forest** node.
4. Expand the **Domains** node, and then navigate to the domain where you want to link the GPO.
5. Expand the domain name.
6. Navigate to the OU where you want to link an existing GPO.
7. Right-click the OU and select **Link an Existing GPO…** from the menu:

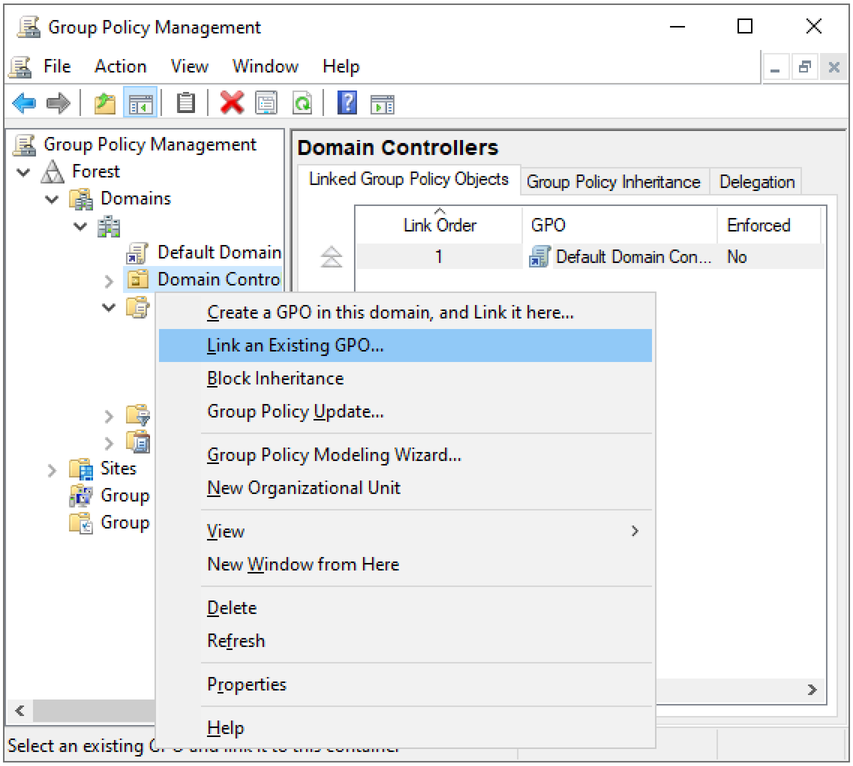


Figure 10.6 – Link an Existing GPO

8. The **Select GPO** window appears. In the window, select the GPO you want to link from the **Group Policy objects:** list.
9. Click **OK** to link the GPO.

How it works...

A GPO is not applied to computer or user objects without a link.

Despite the name *Group Policy*, GPOs cannot be linked to groups.

When a domain-joined device processes Group Policy, it processes GPOs in this order:

1. Locally defined Group Policy settings
2. GPOs linked to sites
3. GPOs linked to the domain the device resides in
4. GPOs linked to the OUs in the OU structure down to the computer object

When a user signs in, Group Policy is processed in the same order, except this time, the domain and OU structure are defined by the location of the user object instead of the computer object.

When there are overlapping settings in different GPOs linked in the preceding order, the last processed GPO with the settings applies.

When multiple GPOs are linked to the same OU, the GPO with the lowest link order is applied, as the settings are applied in the reverse sequence of the link order. Link order can only really be considered a priority when link enforcement and block inheritance don't come into play.

There's more...

GPOs can be linked to sites, domains, and OUs. It's a bad idea to link a GPO to a site due to the nature of intra-domain replication of GPOs.

Blocking inheritance of GPOs on an OU

This recipe shows how to block inheritance of GPOs on an OU.

Getting ready

To block inheritance of GPOs on an OU, sign in to a system with the Group Policy Management feature installed with an account that is either of the following:

- A member of the **Domain Admins** group
- The current owner of the GPO
- Delegated the **Edit settings** or **Edit settings, delete, modify security** permission on the GPO

How to do it...

Perform the following steps:

1. Press Start.
2. Search for **Group Policy Management** and click its search result, or run `gpmc.msc`. The **Group Policy Management** window appears.
3. In the left navigation pane, expand the **Forest** node.
4. Expand the **Domains** node.
5. Expand the domain name.
6. Navigate to the OU where you want to configure inheritance.
7. Right-click the OU and select **Block Inheritance** from the menu.

How it works...

The **Block Inheritance** and **Enforce** settings are two ways to manage how GPOs are processed.

Group Policy inheritance can be used in situations where an OU structure, site, or domain governs settings through GPOs, and where none of these settings are undesired. This way, any GPO will no longer be processed by computer objects and/or user objects in the OU and underlying OU structure that has inheritance blocked.

Managing inheritance is useful where you want to strictly manage test environments. However, it does add complexity to Group Policy and may result in situations that are harder to troubleshoot.

When not all settings are undesired, you can link the GPOs with the desired settings to the OU that has **Block Inheritance** configured.

Enforcing the settings of a GPO Link

This recipe shows how to enforce a GPO link to ensure that its settings always apply.

Getting ready

To enforce a GPO link, sign in to a system with the **Group Policy Management** feature installed with an account that is either of the following:

- A member of the **Domain Admins** group
- The current owner of the GPO
- Delegated the **Edit settings** or **Edit settings, delete, modify security** permission on the GPO

How to do it...

Perform the following steps:

1. Press Start.
2. Search for **Group Policy Management** and click its search result, or run `gpmc.msc`. The **Group Policy Management** window appears.
3. In the left navigation pane, expand the **Forest** node.
4. Expand the **Domains** node, and then navigate to the domain where you want to link the GPO.
5. Expand the domain name.
6. Navigate to the OU where you want to enforce the GPO link.
7. Expand the OU.
8. Right-click the GPO link you want to enforce, and toggle **Enforced** on or off in the menu to enable or disable it. A check mark indicates whether **Enforced** is enabled or not:

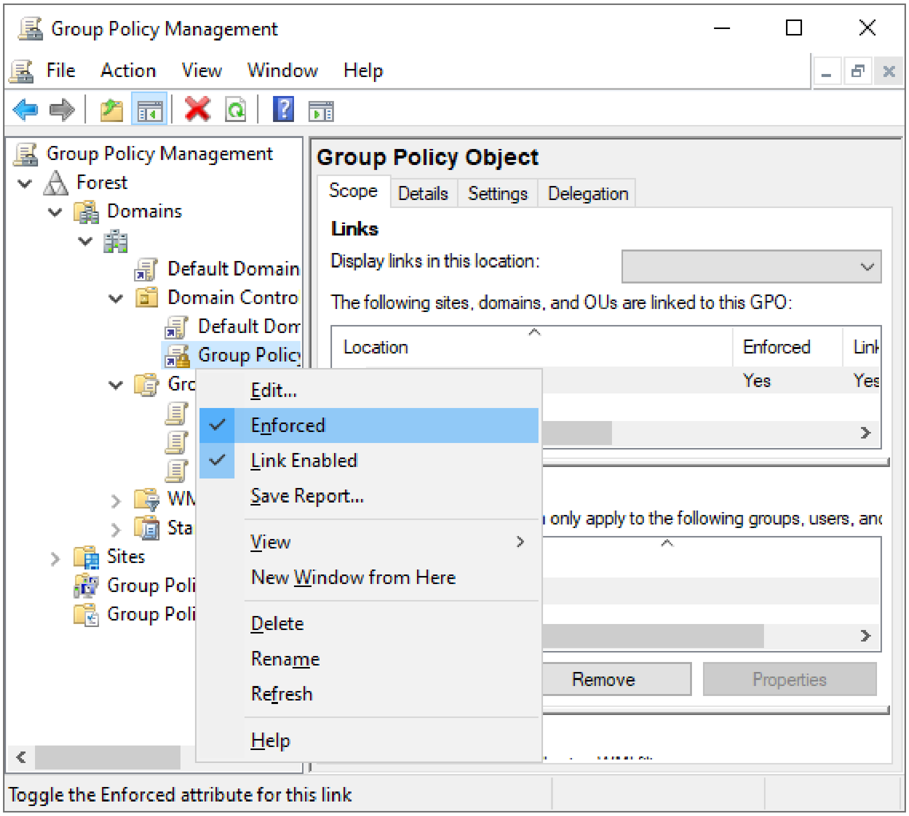


Figure 10.7 – Enforced enabled for a GPO

How it works…

The **Block Inheritance** and **Enforce** settings are two ways to manage how GPOs are processed.

In a complex environment where multiple conflicting GPOs may be applied, enforcing a GPO link ensures the GPO is processed and its settings are applied, regardless of Group Policy inheritance settings. It always takes precedence.

Applying security filters

Group Policy can be linked to OUs but, despite its name, not to groups. However, this recipe shows how a security filter can be applied so that the GPO link only applies to members of a specific group.

Getting ready

To apply security filters on a GPO, sign in to a system with the **Group Policy Management** feature installed with an account that is either of the following:

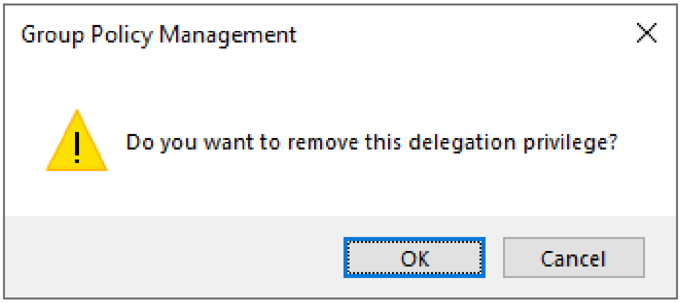
- A member of the **Domain Admins** group
- The current owner of the GPO
- Delegated the **Edit settings** or **Edit settings, delete, modify security** permission on the GPO

How to do it…

Perform the following steps:

1. Press Start.
2. Search for **Group Policy Management** and click its search result, or run `gpmc.msc`. The **Group Policy Management** window appears.
3. In the left navigation pane, expand the **Forest** node.
4. Expand the **Domains** node, and then navigate to the domain where you want to apply security filters to the GPO.
5. Expand the domain name and then expand the **Group Policy Objects** node.
6. Locate the GPO that you want to manage and select it.
7. In the main pane, on the **Scope** tab in the **Security Filtering** area, click the **Add…** button to add a group for security filtering. The **Select User, Computer or Group** window appears.

8. In the **Select User, Computer or Group** window, type the name of a security group and click the **Check Names** button.
9. Click **OK** to add the group to the **Security Filtering** field for the GPO.
10. Select the default **Authenticated Users** entry and click **Remove**. The **Group Policy Management** pop-up window appears:



Group Policy Management

Do you want to remove this delegation privilege?

Figure 10.8 – The Group Policy Management pop-up window

11. In the **Group Policy Management** pop-up window, click **OK** to answer the **Do you want to remove this delegation privilege?** question.

How it works...

By default, all GPOs have the `GpoApply` permission assigned to the **Authenticated Users** group. This permission can be scoped down to a specific group by removing the default permission and adding a scoped group. When the `GpoApply` permission is absent, the settings in the GPO don't apply to each of its GPO links.

Since groups can be nested, security filtering can make Group Policy processing slow, especially in multi-forest environments.

Creating and applying WMI filters

This recipe shows how to apply a WMI filter so that GPOs only apply to specific domain-joined devices and systems. In this recipe, a WMI filter is shown that targets the domain controller with the **Primary Domain Controller Emulator** (**PDCE**) **Flexible Single Master Operations** (**FSMO**) role only.

Getting ready

To create WMI filters on a GPO, sign in to a system with the **Group Policy Management** feature installed with an account that is either of the following:

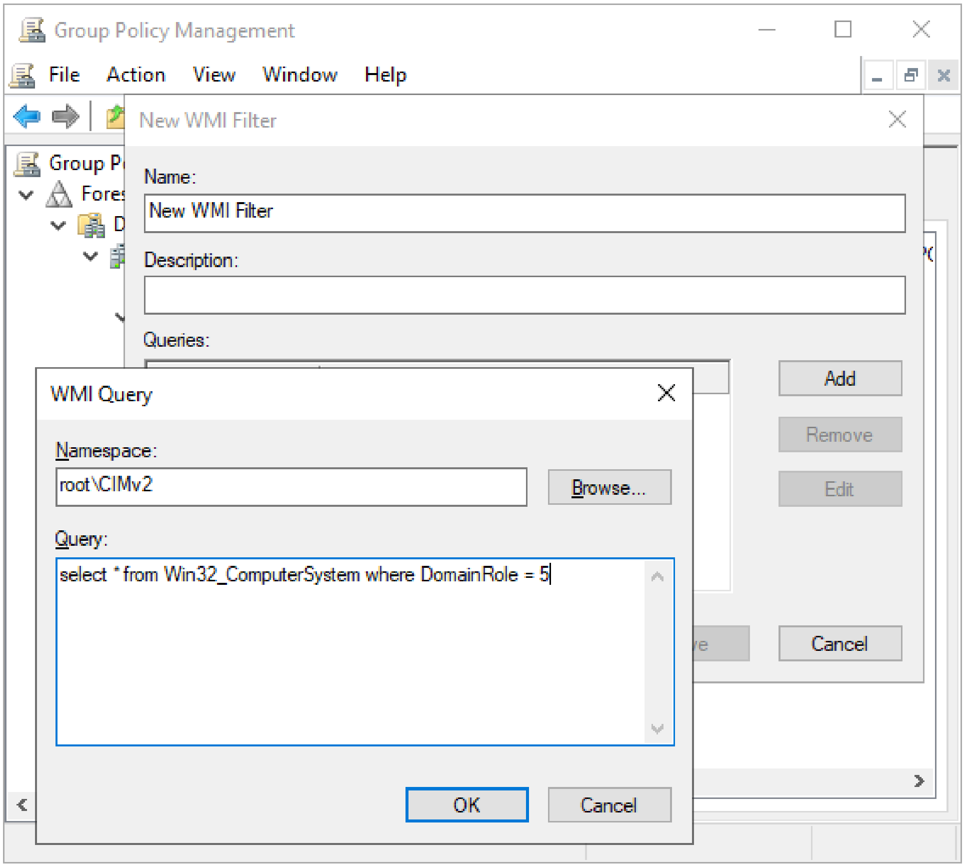
- A member of the **Domain Admins** group
- Delegated the **Edit settings, delete, modify security** permission on the GPO

How to do it…

Perform the following steps:

1. Press Start.
2. Search for **Group Policy Management** and click its search result, or run `gpmc.msc`. The **Group Policy Management** window appears.
3. In the left navigation pane, expand the **Forest** node.
4. Expand the **Domains** node, and then navigate to the domain where you want to apply WMI Filters.
5. Expand the domain name.
6. Right-click the **WMI Filters** node and select **New…** from the menu. The **New WMI Filter** window appears.
7. In the **New WMI Filter** window, enter a name for the new WMI filter in the **Name:** field.
8. Optionally, enter a description for the WMI filter in the **Description:** field.
9. Click **Add**. The **WMI Query** window appears.
10. In the **WMI Query** window, accept the `root\CIMv2`**Namespace:** to target with the WMI filter or click **Browse…** to select another.

11. Enter the WMI query in the **Query:** field:



Name: New WMI Filter

Description:

Queries:

Namespace: root\CIMv2

Query: select * from Win32_ComputerSystem where DomainRole = 5

Figure 10.9 – Enter the WMI query

12. Click **OK** to save the WMI query.
13. Click **Save** to create the WMI filter.
14. In the left navigation pane, navigate to the **Group Policy Objects** node.
15. Select the GPO you want to apply the WMI filter to.
16. In the main pane, on the **Scope** tab in the **WMI Filtering** area, select the WMI filter from **This GPO is linked to the following WMI filter:** drop-down list. The **Group Policy Management** pop-up window appears.
17. In the **Group Policy Management** pop-up window, click **Yes** to answer **Would you like to change the WMI Filter to** [**WMI filter name**].

How it works...

WMI filters can be used to target specific systems in the scope of a GPO throughout all its GPO links, based on the specifications of the device.

Only one WMI filter can be applied at a time for a GPO. Use WMI filters with caution because they can seriously impact the performance of Group Policy processing.

There's more...

A WMI filter is very useful for targeting only the domain controller holding the PDCE FSMO role for a Group Policy that sets the Windows time service to synchronize time with a reliable external time source.

Refreshing GPO settings

This recipe shows how to refresh Group Policy settings on domain-joined hosts after changing and/or adding GPOs.

This recipe shows two ways to refresh GPO settings:

- Using **Group Policy Management**, centrally
- Using the command line on a domain-joined host

Getting ready

To refresh Group Policy settings using **Group Policy Management**, sign in to a system with the **Group Policy Management** feature installed with an account that is a member of the **Domain Admins** group.

To refresh Group Policy settings using the command line on a domain-joined host, sign in to the host with an account that has local administrator privileges.

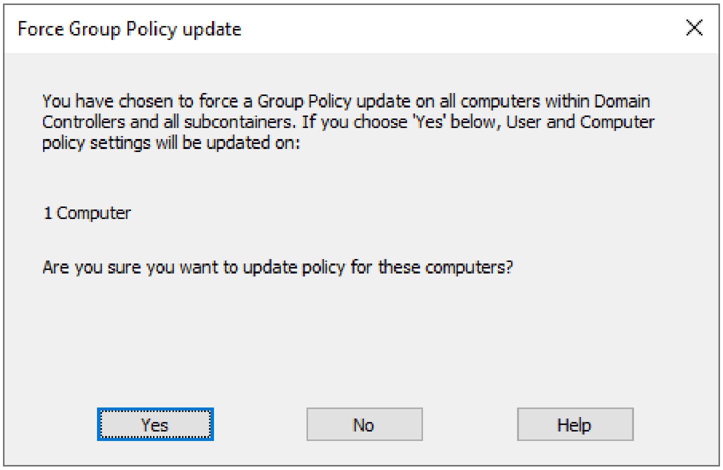
How to do it...

To refresh the GPO settings, choose between the two ways – using Group Policy Management and using the command line on a domain-joined host.

Using Group Policy Management

Perform these steps to refresh the GPO settings for an OU using **Group Policy Management**:

1. Press Start.
2. Search for **Group Policy Management** and click its search result, or run `gpmc.msc`. The **Group Policy Management** window appears.
3. In the left navigation pane, expand the **Forest** node.
4. Expand the **Domains** node, and then navigate to the domain where you want to link the GPO.
5. Expand the domain name.
6. Navigate to the OU where you want to refresh settings.
7. Right-click the OU and select **Group Policy Update…** from the menu. The **Force Group Policy update** window appears:



Force Group Policy update

You have chosen to force a Group Policy update on all computers within Domain Controllers and all subcontainers. If you choose 'Yes' below, User and Computer policy settings will be updated on:

1 Computer

Are you sure you want to update policy for these computers?

Figure 10.10 – The Force Group Policy update window

8. Click **Yes**. The **Remote Group Policy update results** window appears.
9. Review the results and click **Close** when done.

Using the command line on a domain-joined host

Use the following command to update Group Policy settings on an elevated Command Prompt (`cmd.exe`) on a domain-joined host:

```
gpupdate.exe
```

If you want to refresh all Group Policy settings, instead of merely the settings that have changed, add the `/Force` parameter to this command.

How it works...

The Group Policy client on domain-joined hosts typically refreshes Group Policy settings in the background. By default, GPOs are refreshed in the background every 90 minutes, with a 30-minute window of additional time to avoid clobbering the domain controllers. Domain controllers refresh GPOs every 5 minutes.

Using the methods in this recipe, the refresh is immediate. This allows for expedited testing of Group Policy settings and/or a more passionate troubleshooting.

Configuring loopback processing

This recipe shows how to configure Group Policy loopback processing.

Getting ready

To configure Group Policy loopback processing, sign in to a system with the **Group Policy Management** feature installed with an account that is either of the following:

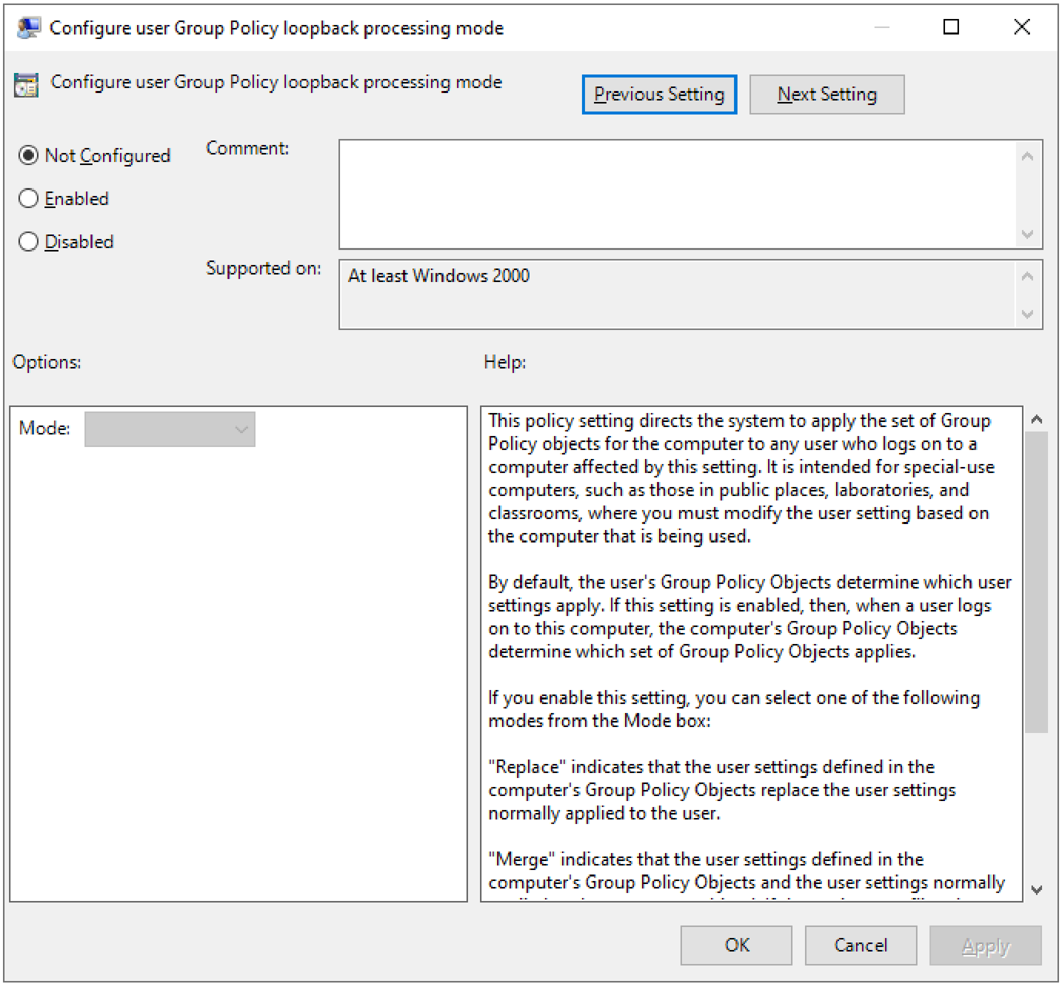
- A member of the **Domain Admins** group
- The current owner of the GPO
- Delegated the **Edit settings** or **Edit settings, delete, modify security** permission on the GPO

How to do it...

Perform the following steps:

1. Press Start.
2. Search for **Group Policy Management** and click its search result, or run `gpmc.msc`. The **Group Policy Management** window appears.
3. In the left navigation pane, expand the **Forest** node.
4. Expand the **Domains** node, and then navigate to the domain where you want to create the GPO.
5. Expand the domain name and then expand the **Group Policy Objects** node.
6. Locate the GPO that you want to manage and select it.

7. In the left navigation pane, right-click the GPO and select **Edit…** from the menu. The **Group Policy Management Editor** window appears.
8. In the **Group Policy Management Editor** window, expand **Computer Configuration**, **Policies**, **Administrative Templates**, **System**, and finally, **Group Policy**.
9. In the main pane, locate the **Configure user Group Policy loopback processing mode** setting and double-click it. The **Configure user Group Policy loopback processing mode** properties window appears:


Configure user Group Policy loopback processing mode
Configure user Group Policy loopback processing mode
Previous Setting
Next Setting
Comment:
Options:
Help:
OK
Cancel
Apply

● Not Configured

○ Enabled

○ Disabled

Supported on: At least Windows 2000

Mode:

This policy setting directs the system to apply the set of Group Policy objects for the computer to any user who logs on to a computer affected by this setting. It is intended for special-use computers, such as those in public places, laboratories, and classrooms, where you must modify the user setting based on the computer that is being used.

By default, the user's Group Policy Objects determine which user settings apply. If this setting is enabled, then, when a user logs on to this computer, the computer's Group Policy Objects determine which set of Group Policy Objects applies.

If you enable this setting, you can select one of the following modes from the Mode box:

"Replace" indicates that the user settings defined in the computer's Group Policy Objects replace the user settings normally applied to the user.

"Merge" indicates that the user settings defined in the computer's Group Policy Objects and the user settings normally

Figure 10.11 – The Configure user Group Policy loopback processing mode window

10. In the **Configure user Group Policy loopback processing mode** window, change the default **Not Configured** value to **Enabled**.
11. From the **Mode:** drop-down menu, select either **Merge** or **Replace** (default).
12. Click **OK** to save the setting.
13. Close the **Group Policy Management Editor** window.

How it works...

In Group Policy loopback processing mode, Group Policy processing is changed. With default settings, the computer object processes the **Computer Configuration** parts of GPOs and the user object processes the **User Configuration** parts of GPOs. In loopback processing mode, though, Group Policy processing is different. It can be configured in two ways:

- **Merge mode**: In merge mode, any GPOs that are applicable to the user object will be processed first. The GPOs applicable to the computer object will be applied afterward. This way, the user account will still correctly process any settings applicable to the account, but user settings in GPOs configured with loopback processing and linked to the computer account override when there are conflicts.
- **Replace mode**: In replace mode, any GPOs that are applicable to the user object will not be processed. Only the settings within the GPOs configured with loopback processing and linked to the computer object will be applied.

Group Policy loopback processing is ideal for the implementations of Remote Desktop Session Hosts and other Remote Desktop, Terminal Server, and **Virtual Desktop Infrastructure** (**VDI**) implementations. The user settings configured in the GPOs linked to the computer objects only apply when users sign in to these hosts but not when they sign in to other domain-joined devices.

Restoring a default GPO

This recipe shows how to restore the **Default Domain Policy** and the **Default Domain Controllers Policy** to default settings.

Getting ready

To restore the **Default Domain Policy** and the **Default Domain Controllers Policy** to default settings, sign in to a non-read-only domain controller with an account that is a member of the **Domain Admins** group.

How to do it...

The `dcgpofix.exe` command-line utility can be used to restore the **Default Domain Policy** and the **Default Domain Controllers Policy** to their default settings.

Restoring the Default Domain Policy

Use the following command to restore the **Default Domain Policy** to its default settings on an elevated Command Prompt (`cmd.exe`):

```
dcgpofix.exe /target:Domain
```

Type *Y* followed by *Enter* twice to continue and restore the **Default Domain Policy**.

Restoring the Default Domain Controllers Policy

Use the following command to restore the **Default Domain Controllers Policy** to its default settings on an elevated Command Prompt (`cmd.exe`):

```
dcgpofix.exe /target:DC
```

Type *Y* followed by *Enter* twice to continue to restore the **Default Domain Controllers Policy**.

Restoring the Default Domain Policy and the Default Domain Controllers Policy

Use the following command to restore the **Default Domain Policy** and the **Default Domain Controllers Policy** to their default settings on an elevated Command Prompt (`cmd.exe`):

```
dcgpofix.exe /target:BOTH
```

Type *Y* followed by *Enter* twice to restore the **Default Domain Policy** and the **Default Domain Controllers Policy.**

How it works...

When you want to revert to the default settings for the **Default Domain Policy** and the **Default Domain Controllers Policy**, or if you've deleted them, they can be easily restored using the `dcgpofix.exe` command-line utility.

There's more...

As there have been changes to the **Default Domain Policy** and the **Default Domain Controllers Policy** across versions of Windows Server and the Active Directory schema, `dcgpofix.exe` checks the schema version. If the schema level is different from the current domain controller operating system, `dcgpofix.exe` displays the following message:

```
The Active Directory schema version for this domain and the
version supported by this tool do not match. The GPO can be
restored using the /ignoreschema command-line parameter.
However, it is recommended that you try to obtain an updated
version of this tool that might have an updated version of the
Active Directory schema. Restoring a GPO with an incorrect
schema might result in unpredictable behavior.
```

In this case, add the `/ignoreschema` parameter to the previous command-line examples.

Creating the Group Policy Central Store

This recipe shows how to configure the Group Policy Central Store in the SYSVOL to optimize Group Policy authoring and replication.

Getting ready

Implement or locate a default Windows client device with Microsoft Office and any other software that supports Group Policy management. Install language packs for the languages that are used by admins in your organization. Update this system with the latest available Windows updates.

To create the Group Policy Central Store, sign in to a non-read-only domain controller or access the SYSVOL over the network with an account that is a member of the Domain Admins group.

How to do it...

Perform the following steps:

1. Sign in to the default Windows client device for your organization.
2. Press Start.
3. Search for **File Explorer** and select it from the search results, or run `explorer.exe`. The **File Explorer** window appears.
4. Navigate to the `Windows` folder – typically, `C:\Windows`.

5. Locate the `PolicyDefinitions` folder.
6. Right-click the `PolicyDefinitions` folder and select **Copy** from the menu.
7. Navigate the current **File Explorer** window to the Active Directory SYSVOL – for instance, `\\lucernpub.com\SYSVOL\lucernpub.com`.
8. In the SYSVOL, navigate to the `Policies` folder.
9. Right-click an empty space in the main pane and select **Paste** from the menu.
10. Navigate to the `C:\Program Files (x86)\Microsoft Group Policy` folder, if it exists.
11. This folder may contain multiple versions of downloaded Group Policy administrative templates. Copy the `PolicyDefinitions` folder in the latest version folder and paste it into the same location by repeating *steps 7 through 9*.

How it works...

Since Windows Server 2008 and Windows Vista, Group Policy settings in the **Administrative Templates** parts of GPOs are represented on the filesystem by the `*.admx` and `*.adml` files. The first type of file defines settings. The latter type of file provides language-dependent labels, so administrators can work together seamlessly using different languages.

Beyond the language benefit, the new file types also allow a central store to store all Group Policy settings and language settings in the SYSVOL. This way, files for configured settings no longer have to be stored with each GPO in the SYSVOL, only once in the SYSVOL. This optimizes Group Policy replication between domain controllers significantly.

There's more...

Creating the Group Policy Central Store requires a process that is revisited when new software versions are introduced within an organization. Overwrite the `*.adml` and `*.admx` files with the newer versions.

11 Securing Active Directory

From a business perspective, Active Directory needs to be an available, confidential attribute store with absolute integrity. The security measures in this chapter detail how to achieve a higher level of confidentiality and integrity.

The following recipes are covered in this chapter:

- Applying fine-grained password and account lockout policies
- Backing up and restoring GPOs
- Backing up and restoring the Active Directory database
- Working with Active Directory snapshots
- Managing the DSRM passwords on domain controllers
- Protecting important objects from accidental deletion
- Implementing **Local Administrator Password Solution** (**LAPS)**
- Managing deleted objects
- Working with **group Managed Service Accounts** (**gMSAs**)

- Configuring diagnostic logging
- Configuring the advanced security audit policy
- Resetting the **KRBTGT** secret
- Using the **Security Configuration Wizard** (**SCW**) to secure domain controllers
- Leveraging the **Protected Users** group
- Putting authentication policies and authentication policy silos to good use
- Configuring Extranet Smart Lockout

Applying fine-grained password and account lockout policies

Active Directory comes with a built-in password policy. Admins can configure stricter password policies and account lockout policies. This way, privileged accounts, such as members of the **Domain Admins** group, can be configured with more secure password and account lockout settings. This recipe shows how.

Getting ready

To apply fine-grained password and account lockout policies, sign in to a domain controller or a member server and/or device with the **Remote Server Administration Tools** (**RSAT**) for Active Directory Domain Services installed.

Sign in with an account that is a member of the **Domain Admins** group, or with an account that is delegated to manage fine-grained password and account lockout policies in the domain.

Fine-grained password and account lockout policies require the Windows Server 2008 **Domain Functional Level** (**DFL**), or a higher version of the DFL.

How to do it...

This recipe shares two ways to manage fine-grained password and account lockout policies:

- Using the **Active Directory Administrative Center**
- Using the Active Directory module for Windows PowerShell

Using the Active Directory Administrative Center

Perform these steps to create and apply a fine-grained password and account lockout policy in the **Active Directory Administrative Center**:

1. Press Start.
2. Search for **Active Directory Administrative Center** and select it from the search results, or run `dsac.exe`. The **Active Directory Administrative Center** window appears.
3. In the left navigation window, switch to the tree view.
4. Expand the domain name and navigate to **Password Settings Container** under the **System** container:

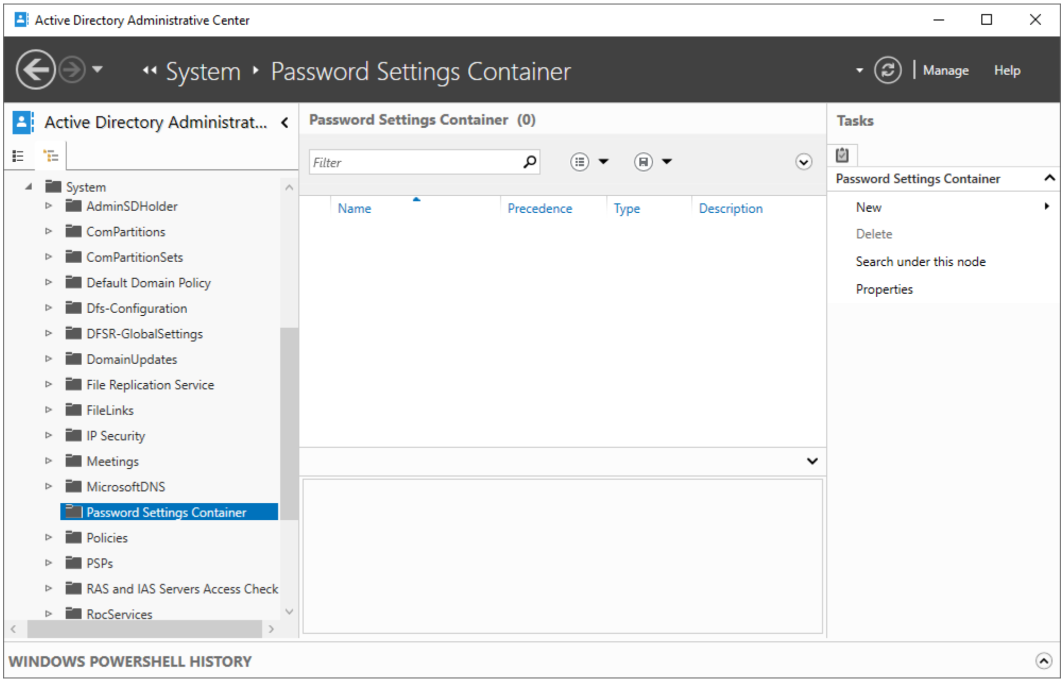


Figure 11.1 – Password Settings Container in the Active Directory Administrative Center

5. In the **Tasks** pane, under **Password Settings Container**, click **New** and then **Password Settings** in the menu. The **Create Password Settings** screen appears.
6. In the **Create Password Settings** screen, in the **Password Settings** section, fill in values for the **Name** and **Precedence** fields.

7. Select every setting that you want to apply to meet the requirements. Only one policy may apply to any user account, so define all the settings that you want to apply.
8. In the **Directly applies to** section, press the **Add…** button. The **Select Users or Groups** pop-up window appears.
9. In the **Select Users or Groups** pop-up window, type the name of the user account(s) and/or groups you want the fine-grained password policy to directly apply to, or click the **Advanced** button to search for the user account(s) and/or groups.
10. Click **Check Names**.
11. Click **OK** to have the fine-grained password policy directly apply to the user(s) and/or group(s).
12. Click **OK** to create the fine-grained password and account lockout policy.

Using the Active Directory Module for Windows PowerShell

To create a fine-grained password and account lockout policy, use the following lines of PowerShell on a system with the Active Directory module for Windows PowerShell installed:

```
New-ADFineGrainedPasswordPolicy PolicyName
-ComplexityEnabled $true -LockoutDuration "00:30:00"
-LockoutObservationWindow "00:30:00" -LockoutThreshold
"5" -MaxPasswordAge "42.00:00:00" -MinPasswordAge
"1.00:00:00" -MinPasswordLength "7" -PasswordHistoryCount
"24" -Precedence "1" -ReversibleEncryptionEnabled $false
-ProtectedFromAccidentalDeletion $true
```

To apply a fine-grained password and account lockout policy to a group, use the following line of PowerShell on a system with the Active Directory module for Windows PowerShell installed:

```
Add-ADFineGrainedPasswordPolicySubject -Identity PolicyName
-Subjects GroupName
```

Replace `GroupName` with the group or individual user account you want to apply the fine-grained password and account lockout policy to.

How it works...

Active Directory comes with a built-in password policy. This is the password policy that is automatically set at the Active Directory domain level. This default policy does not enable account lockout.

The password policy applies when the password is changed and when it is set by the admin. Account lockout policies observe bad password attempts. When a bad password is typed, it is added to the bad password count. When this count reaches the limit within the time specified by the observation period, the account is locked for the duration of the time-out period. Accounts can be configured to remain locked indefinitely. In this case, accounts need to be unlocked by a person using their admin account or otherwise privileged account.

Admins can configure stricter password policies and account lockout policies at the domain level, but they can also configure these policies granularly per user account and/or per group.

A fine-grained password and account lockout policy replaces the policy on the domain level completely. Define all the settings that you want to have applied because all other password policies are ignored.

Only one password policy can apply at one time to a user object. When multiple fine-grained password policies are applied, the policy applied to a user account directly applies. When a user account has memberships in multiple groups, the password policy with the lowest value for precedence is applied. The precedence value can be interpreted as a priority. Specifying unique precedence values for password policies is key to having the right policy applied.

In scenarios where lockout is configured identically over every password policy, the maximum lifetime for the password makes for an excellent precedence value.

There's more...

When you're unsure which fine-grained password and account lockout policy applies, look at the user account's `msDS-ResultantPSO` attribute. It exposes the reference to the fine-grained password and account lockout policy that is applied.

The `Get-ADUserResultantPasswordPolicy` PowerShell cmdlet can also be used for this purpose.

Backing up and restoring GPOs

Group Policy Management does not offer to roll back changes in **Group Policy Objects** (**GPOs**). However, when, in the process of modifying GPOs, a step is added to create a backup of the GPOs, inadvertent settings can be rolled back by restoring a previous backup.

This recipe shows what that step would look like and how to restore a GPO.

Getting ready

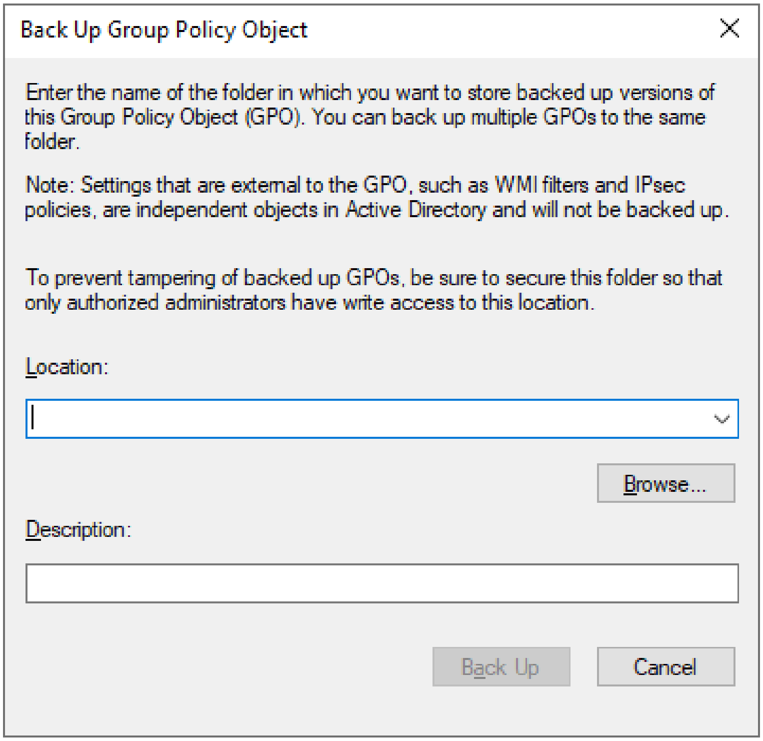
Sign in to a domain controller or a member server and/or device with the RSAT for Active Directory Domain Services installed. Ideally, the domain controller or member server runs Windows Server 2012, or a newer version of Windows Server.

Sign in with an account that is a member of the **Domain Admins** group. In contrast to the delegation of creating, linking, managing, editing, and reading GPOs, backing up and restoring GPOs cannot be delegated.

How to do it...

Perform these steps to back up GPOs:

1. Press Start.
2. Search for **Group Policy Management** and select it from the search results, or run `gpmc.msc`. The **Group Policy Management** window appears.
3. In the left navigation pane, expand the forest, then the **Domains** node, and then the domain for which you want to backup GPOs.
4. Select the **Group Policy Objects** node and right-click it.
5. From the menu, select **Back Up All...**. The **Back Up Group Policy Object** window appears:



Back Up Group Policy Object

Enter the name of the folder in which you want to store backed up versions of this Group Policy Object (GPO). You can back up multiple GPOs to the same folder.

Note: Settings that are external to the GPO, such as WMI filters and IPsec policies, are independent objects in Active Directory and will not be backed up.

To prevent tampering of backed up GPOs, be sure to secure this folder so that only authorized administrators have write access to this location.

Location:

Description:

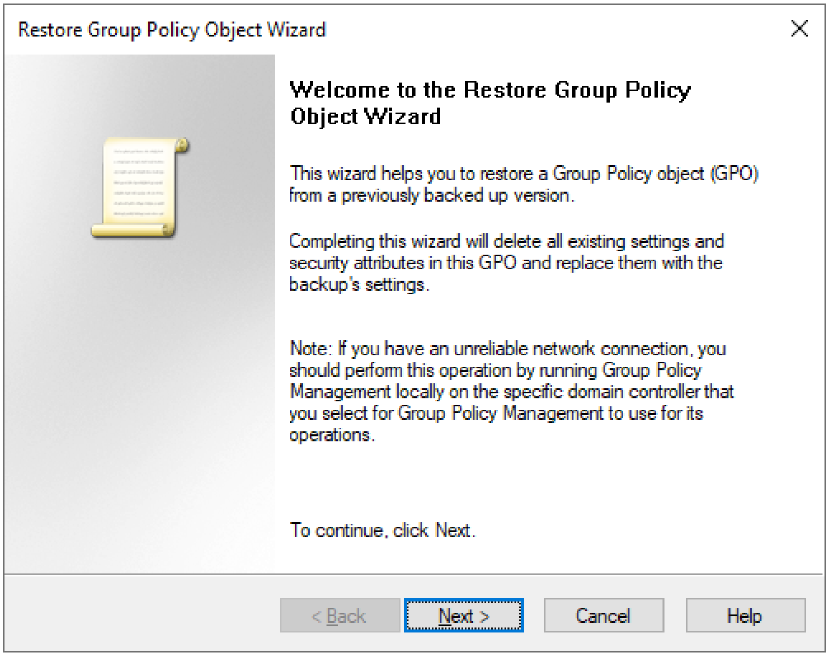
Figure 11.2 – The Back Up Group Policy Object window

6. In the **Back Up Group Policy Object** window, in the **Location:** field, provide a storage location to store the backups. Use the **Browse...** button to locate and/or create a folder for Group Policy backups.
7. Provide a description in the **Description:** field.
8. Click the **Back Up** button.
9. In the **Backup** window, review the statistics of the backup and click **OK** when done.

Perform these steps to restore inadvertent settings for a GPO:

1. Press Start.
2. Search for **Group Policy Management** and select it from the search results, or run `gpmc.msc`. The **Group Policy Management** window appears.
3. In the left navigation pane, expand the forest, then the **Domains** node, and then the domain for which you want to restore a GPO.
4. Expand the **Group Policy Objects** node and locate the GPO you want to restore from a previous backup.

5. Select the GPO to restore.
6. Right-click the GPO and select **Restore from Backup…** from the menu. The **Restore Group Policy Object Wizard** window appears:



Restore Group Policy Object Wizard

Welcome to the Restore Group Policy Object Wizard

This wizard helps you to restore a Group Policy object (GPO) from a previously backed up version.

Completing this wizard will delete all existing settings and security attributes in this GPO and replace them with the backup's settings.

Note: If you have an unreliable network connection, you should perform this operation by running Group Policy Management locally on the specific domain controller that you select for Group Policy Management to use for its operations.

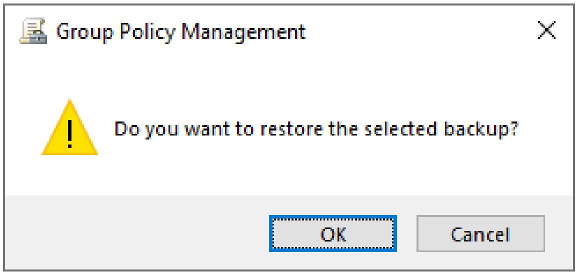
To continue, click Next.

Figure 11.3 – The Restore Group Policy Object Wizard

7. On the **Welcome to the Restore Group Policy Object Wizard** screen, click **Next** >.
8. On the **Backup Location** screen, type the location of previous backups or use the **Browse...** button to look it up.
9. Click **Next** >.
10. On the **Source GPO** screen, select the GPO that you want to restore.
11. Click **Next** >.
12. On the **Completing the Restore Group Policy Object Wizard** screen, click **Finish** to start the restoration.
13. In the **Restore** window, review the outcome, and click **OK** when done.

Perform these steps to restore an inadvertently deleted GPO:

1. Press Start.
2. Search for **Group Policy Management** and select it from the search results, or run `gpmc.msc`. The **Group Policy Management** window appears.
3. In the left navigation pane, expand the forest, then the **Domains** node, and then the domain for which you want to restore a GPO.
4. Select the **Group Policy Objects** node and right-click it.
5. From the menu, select **Manage Backups…**. The **Manage Backups** window appears.
6. In the **Manage Backups** window, click the **Browse** button to navigate to the folder that contains the previous backups. Click **OK** with the folder selected.
7. Select the GPO(s) that you want to restore from the backup.
8. Click **Restore** to start the restoration. The **Group Policy Management** pop-up window appears:



Do you want to restore the selected backup?

Figure 11.4 – The Group Policy Management pop-up window

9. In the **Group Policy Management** pop-up window, click **OK**.
10. In the **Restore** window, review the outcome, and click **OK** when done.
11. In the **Manage Backups** window, click **Close**.

How it works…

When you create backups of GPOs, the settings for these objects are stored in a backup file. Then, when inadvertent changes are made to these objects, they can be restored from the backup file.

Depending on the purpose of the GPO backup, the location to store the backups can be on the domain controller or in a remote location. Storing on a domain controller may be a good option for a fast restore test, duplicating group policies between Active Directory forests, and versioning. However, for true backups, always use a remote location.

Alternatively, you can implement the **Advanced Group Policy Management** (**AGPM**) tool.

There's more...

Backups for GPOs can only be restored in the same Active Directory forest. To recreate GPOs from one Active Directory forest to another, use the **Export** and **Import** functionality in **Group Policy Management**.

Backing up and restoring Active Directory

To avoid a situation where Active Directory, the backbone of every Microsoft-oriented networking infrastructure, is irreversibly lost, Active Directory backups should be performed. Additionally, restores should be performed regularly in an isolated networking environment to ensure backups can be restored and procedures are up to date and familiar to admins.

This recipe shows how to create backups of Active Directory using Windows Backup.

Getting ready

To make a backup of a domain controller, sign in to the domain controller with a user account that is a member of the **Domain Admins** group or the **Backup Operators** group.

To restore a domain controller, you need to know the **Directory Services Restore Mode** (**DSRM**) password for the domain controller.

The Windows Backup feature needs to be installed. Use the following PowerShell one-line in an elevated PowerShell window to do so:

```
Install-WindowsFeature Windows-Server-Backup
```

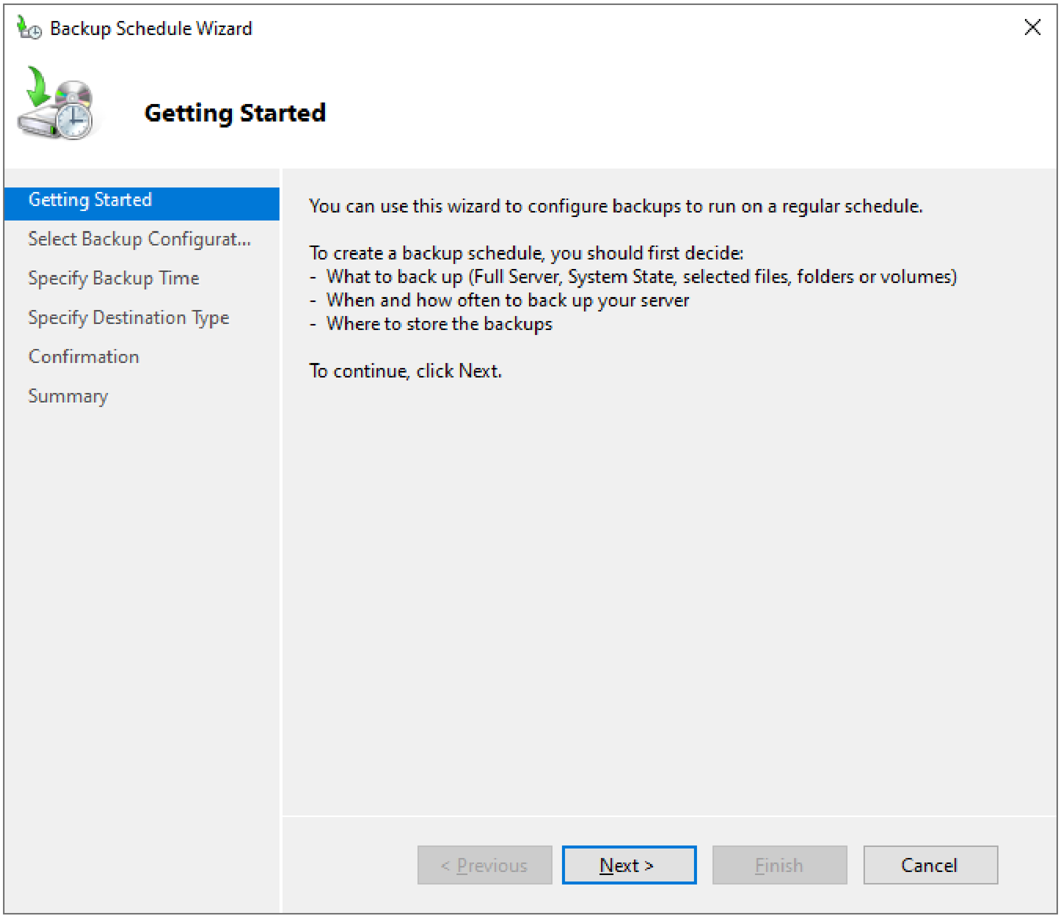
To avoid any dependencies on Active Directory-integrated network and file access, make sure you back up to a dedicated (USB) hard drive for physical domain controllers or to a separate **Logical Unit Number** (**LUN**) in the virtualization fabric for virtual domain controllers. When working with USB hard disks, purchase at least two devices for off-site storage options and replacement in case of failure scenarios.

How to do it...

Perform these steps to create backups of a domain controller:

1. Sign in to the domain controller.
2. Plug in the dedicated hard drive or attach the LUN to which you want to back up. Install drivers, if necessary.
3. Press Start.

4. Search for **Windows Server Backup** and select it from the search results, or run `wbadmin.exe`. The **wbadmin - [Windows Server Backup (Local)]** window appears.
5. In the left navigation pane, select **Local Backup**.
6. In the **Actions** pane on the right side of the **Windows Server Backup** window, click **Backup Schedule…**. The **Backup Schedule Wizard** window appears:



You can use this wizard to configure backups to run on a regular schedule.

To create a backup schedule, you should first decide:
- What to back up (Full Server, System State, selected files, folders or volumes)
- When and how often to back up your server
- Where to store the backups

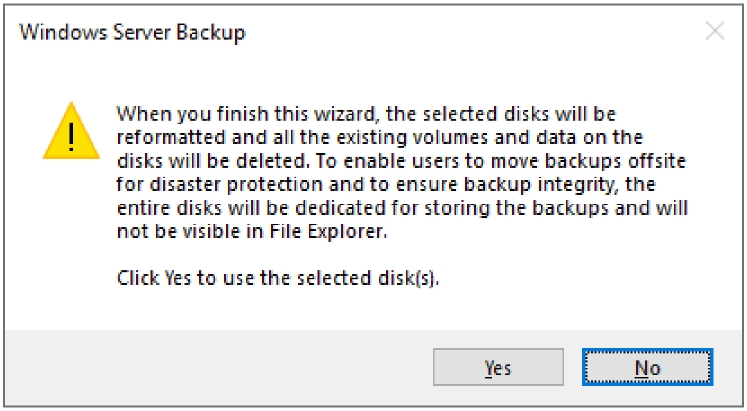
To continue, click Next.

Figure 11.5 – The Getting Started screen of the Backup Schedule Wizard

7. On the **Getting Started** screen, click **Next >**.
8. On the **Select Backup Configuration** screen, select **Full Server (recommended)**.

   Alternatively, select **Custom**, but in this case, always select **System state** on the **Select Items for Backup** screen as part of the backup configuration if you want to be able to restore the domain controller functionality from backup.

9. Click **Next** >.
10. On the **Specify Backup Time** screen, select **Once a day**. Choose a time that is outside the typical opening hours or working day(s) in your organization. If you have other processes running out of hours, be sure not to clash with them.
11. Click **Next** >.
12. On the **Specify Destination Type** screen, select **Back up to a hard disk that is dedicated for backups (recommended)**.
13. Click **Next** >.
14. On the **Select Destination Disk** screen, select the removable disk to backup to, and click **Next** >. The **Windows Server Backup** pop-up window appears:



Windows Server Backup

When you finish this wizard, the selected disks will be reformatted and all the existing volumes and data on the disks will be deleted. To enable users to move backups offsite for disaster protection and to ensure backup integrity, the entire disks will be dedicated for storing the backups and will not be visible in File Explorer.

Click Yes to use the selected disk(s).

Figure 11.6 – The Windows Server Backup pop-up window

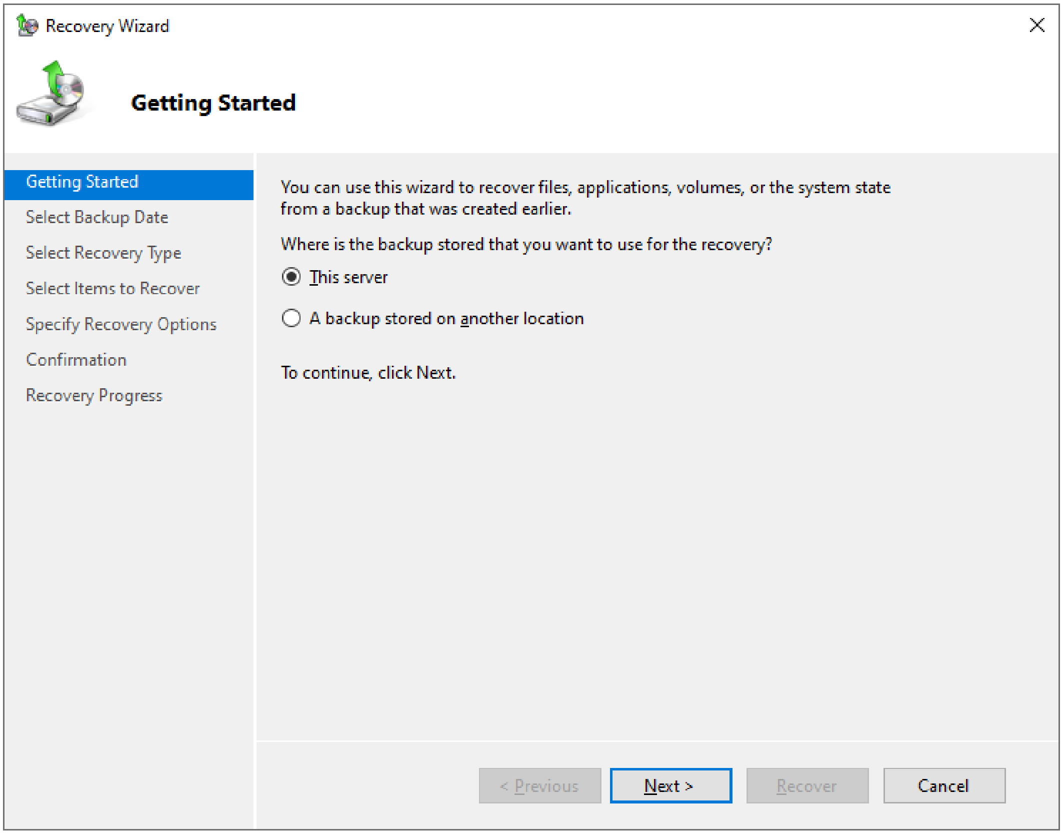
15. Click **Yes** to acknowledge that the selected disk will be reformatted and that all the existing data on the disk will be deleted.
16. On the **Confirmation** screen, click **Finish**.
17. On the **Summary** screen, click **Close**.

Windows Server Backup can also be used on Command Prompt (`cmd.exe`). To create an instant system state backup to a hard disk attached as `F:\`, use the following command on an elevated Command Prompt (`cmd.exe`):

```
wbadmin.exe start systemstatebackup -backuptarget:F:
```

Perform these steps to restore a backup of a domain controller:

1. Start the domain controller in DSRM.
2. Sign in to the domain controller with the username `Administrator` and the DSRM password as the password.
3. Plug in the dedicated hard drive or LUN where you want to restore from.
4. Start **Server Manager** (`servermanager.exe`) if it doesn't start automatically.
5. From the **Tools** menu on the top gray bar, choose **Windows Server Backup**.
6. In the left navigation pane, select **Local Backup**.
7. In the action pane on the right side of the **Windows Server Backup** screen, click **Recover…**. The **Recovery Wizard** appears:



You can use this wizard to recover files, applications, volumes, or the system state from a backup that was created earlier.

Where is the backup stored that you want to use for the recovery?

● This server

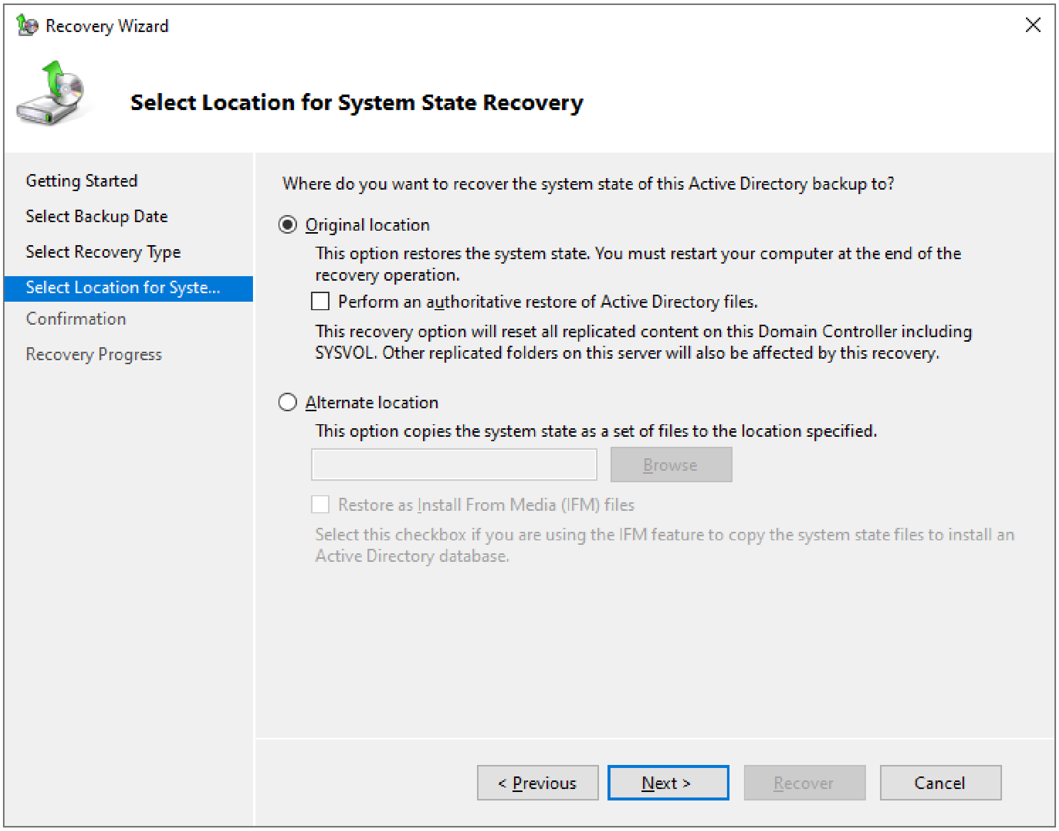
○ A backup stored on another location

To continue, click Next.

Figure 11.7 – The Getting Started screen of the Recovery Wizard

8. On the **Getting Started** screen, select the **This server** option and click **Next >**.
9. On the **Select Backup Date** screen, select the backup to restore by selecting **Backup date:** from the calendar and, optionally, by selecting the time the backup was created for the **Time:** field.
10. Click **Next >**.
11. On the **Select Recovery Type** screen, select **System state** and click **Next >**.

    You reach the **Select Location for System State Recovery** screen:



Select Location for System State Recovery

Where do you want to recover the system state of this Active Directory backup to?

● Original location

This option restores the system state. You must restart your computer at the end of the recovery operation.

□ Perform an authoritative restore of Active Directory files.

This recovery option will reset all replicated content on this Domain Controller including SYSVOL. Other replicated folders on this server will also be affected by this recovery.

○ Alternate location

This option copies the system state as a set of files to the location specified.

□ Restore as Install From Media (IFM) files

Select this checkbox if you are using the IFM feature to copy the system state files to install an Active Directory database.

Figure 11.8 – The Select Location for System State Recovery screen of the Recovery Wizard

12. On the **Select Location for System State Recovery** screen, select **Original location**. Optionally, select the **Perform an authoritative restore of Active Directory files.** option. Click **Next >**.
13. On the **Confirmation** screen, click **Finish**.
14. After restoration, restart the domain controller normally.

The Command Prompt (cmd.exe) is also available in this scenario. Use the following command to restore a system state backup created on February 2nd, 2022 at 9PM:

```
wbadmin.exe start systemstaterecovery -version:02/02/2022-21:00
```

Add the -AuthSysvol parameter if you want to restore the domain controller authoritatively. After restoration, restart the domain controller normally.

How it works...

By creating backups of the system state, all the information to restore a domain controller is copied off the system and onto removable media. This way, when a domain controller becomes non-functional, the backup can be used to restore the functionality to a new Windows Server installation or boot up from the backup media to restore the entire domain controller.

Windows Server Backup uses volume shadow copies with the Active Directory VSS Writer to make a backup of the Active Directory files while they are in use. This way, there is no need to stop the **Active Directory Domain Services** service to make a consistent backup. In most third-party backup applications, this functionality is called **application-consistent backups**.

Domain controllers can be restored authoritatively or non-authoritatively. When restored authoritatively, the restored domain controller will take the role of an authoritative replication partner for Active Directory and SYSVOL replication; all domain controllers will assume that its versions of the database and the SYSVOL are the most up to date. When restoring non-authoritatively, the domain controller reports itself as a new Active Directory replication partner and replicates from other domain controllers, ignoring any changes it might have made before being restored.

The DSRM administrator password for the domain controller is stored on the system and provides the ability to sign in with a local administrator account when the Active Directory Domain Services service is not running. When the service runs on a domain controller, this password cannot be used. The DSRM password is initially configured when the domain controller is configured.

As modern malware attacks on environments involve invalidating backups, ensure to store backups for domain controllers off the network, and ideally off-site.

See also...

If you don't know or have forgotten the DSRM password, see the *Managing the DSRM passwords on domain controllers* recipe.

Working with Active Directory snapshots

This recipe shows how to work with Active Directory snapshots as an alternative to having to restore entire backups for a domain controller to restore only a handful of objects.

Getting ready

To work with snapshots for a domain controller, sign in to a domain controller with a user account that is a member of the **Domain Admins** group or the **Backup Operators** group.

How to do it...

1. To make a snapshot, type the following command in an elevated Command Prompt (`cmd.exe`) window:

```
ntdsutil.exe "activate instance ntds" "snapshot" "create" q
q
```

2. To view all snapshots, type the following command in an elevated Command Prompt (`cmd.exe`) window to get a numbered list of all available snapshots:

```
ntdsutil.exe "activate instance ntds" "snapshot" "list all"
q q
```

3. To mount a snapshot, type the following command in an elevated Command Prompt (`cmd.exe`) window, using the **Universally unique identifier** (**UUID**) of the snapshot you want to mount from the previous command, with or without the curly brackets:

```
ntdsutil.exe "activate instance ntds" "snapshot" "mount
UUID" q q
```

The preceding command will output the folder where the database is mounted.

4. Run the following command to expose it as an **Lightweight Directory Access Protocol** (**LDAP**) store:

```
dsamain.exe -dbpath "Location from previous command"
-ldapport PortNumber
```

Keep this command running for as long as you want the LDAP endpoint for the snapshot to be available.

Choose a port number in the dynamic port range. As `389` is already in use by the Active Directory instance, port `5389` or `55389` would make for an excellent port number.

To look up information, use the cmdlets in the Active Directory module for Windows PowerShell. Specify the `-Server` parameter, and type the hostname of the server and the port number separated by a colon, as follows:

```
Get-ADComputer -Identity * -Server Localhost:PortNumber
```

Alternatively, you can use **ADSI Edit** (`adsiedit.msc`) or Windows Sysinternals' AD Explorer tool to connect to the snapshot.

How it works...

Snapshots for Active Directory can be useful in scenarios where an organization has a need to compare information from a certain point in time (the time the snapshot was taken) with information from another point in time (for instance, today).

Creating an Active Directory snapshot requires the Volume Shadow Copy functionality and a functional Active Directory VSS writer. These features are available by default on Windows Server. However, if anything is not working, check that first.

There's more...

When you want to transfer information between a snapshot and the Active Directory, use tools such as `ldifde.exe` and `csvde.exe`. There is no native tooling available to perform these kinds of actions.

See also

To change the password for the administrator in DSRM, perform the steps from the next recipe.

Managing the DSRM passwords on domain controllers

This recipe shows how to manage the password to sign in to domain controllers when the Active Directory Domain Services service is not running or the domain controller is in DSRM.

Getting ready

To manage the DSRM password on a domain controller, sign in to a domain controller with a user account that is a member of the **Domain Admins** group, the **Backup Operators** group, or the **Server Operators** group.

For the scenario where the DSRM Administrator password is automatically synchronized with an account in Active Directory, create a disabled user account with a strong password. Document the password in a password vault. Additionally, ensure all domain controllers run Windows Server 2008 or newer versions of Windows Server and are replicating properly.

How to do it...

This recipe shows two routes:

1. Manually resetting the DSRM administrator password
2. Synchronizing the DSRM administrator password

Manually resetting the DSRM administrator password

To manually reset the DSRM administrator password on a domain controller, type the following commands in an elevated Command Prompt (`cmd.exe`) window when the domain controller is running properly:

```
ntdsutil.exe
> set dsrm password
> reset password on server null
```

Type or paste the password to use as the DSRM administrator password and then type the following commands:

```
> quit
> quit
```

Document the password in a password vault.

Synchronizing the DSRM administrator password

To synchronize the DSRM administrator password on a domain controller with a user account in Active Directory, type the following command in an elevated Command Prompt (`cmd.exe`) window when the domain controller is running and replicating properly:

```
ntdsutil.exe "set dsrm password" "sync from domain account
DSRMSyncUser" quit quit
```

Replace `DSRMSyncUser` with the `sAMAccountName` attribute of the user object whose password you want to have synchronized as the DSRM administrator password.

As a recommendation, use a different user object in Active Directory for each domain controller.

How it works...

When the Active Directory Domain Services service is not running, the domain controller is in DSRM, or the domain controller is non-functional, you'll need a way to sign in to the domain controller. As the Active Directory database is not available, a special password is maintained for the domain controller-local built-in Administrator account.

The password can be managed in two ways:

- The manual reset scenario
- The domain account password synchronization scenario

In the first scenario, the password is set manually per domain controller and is then to be documented in a password vault.

In the second scenario, the domain controller is instructed to synchronize the password from a specific Active Directory account. This password is then set manually on the user account and to be documented. in a password vault.

A recommended practice is to have different passwords for each domain controller. Another recommended practice is to reset the DSRM administrator passwords yearly.

Protecting important objects from accidental deletion

To avoid having to reboot domain controllers and restore individual items, important objects in Active Directory can be protected from accidental deletion.

Getting ready

To protect important objects from accidental deletion, sign in to a domain controller with a user account that is a member of the **Domain Admins** group.

How to do it...

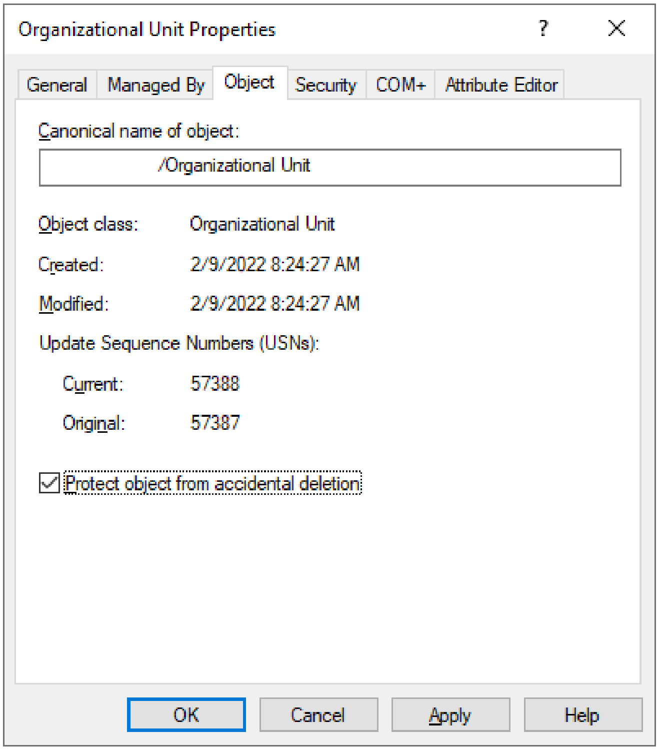
This recipe shows three ways to protect an **Organizational Unit** (**OU**) from accidental deletion:

- Using **Active Directory Users and Computers**
- Using the **Active Directory Administrative Center**
- Using Windows PowerShell

Using Active Directory Users and Computers

Perform these steps to protect an OU from accidental deletion using **Active Directory Users and Computers**:

1. Press Start.
2. Search for **Active Directory Users and Computers** and click its search result, or run `dsa.msc`. The **Active Directory Users and Computers** window appears.
3. From the **View** menu, enable the **Advanced Features** option.
4. In the left navigation pane, navigate to the OU you want to protect from accidental deletion and select it.
5. Right-click the OU and select **Properties** from the menu.
6. Navigate to the **Object** tab.
7. Check the **Protect object from accidental deletion** checkbox:



Canonical name of object: /Organizational Unit

Object class: Organizational Unit

Created: 2/9/2022 8:24:27 AM

Modified: 2/9/2022 8:24:27 AM

Update Sequence Numbers (USNs):

Current: 57388

Original: 57387

☑ Protect object from accidental deletion

Figure 11.9 – The Protect object from accidental deletion option for an OU

8. Click **OK**.

Using the Active Directory Administrative Center

Perform these steps to protect an OU from accidental deletion using the Active Directory Administrative Center:

1. Press Start.
2. Search for **Active Directory Administrative Center** and click its search result, or run `dsac.exe`. The **Active Directory Administrative Center** window appears.
3. In the left navigation pane, switch to the tree view.
4. If necessary, expand the tree to locate the OU you want to protect from accidental deletion and select it.
5. Right-click the OU and select **Properties** from the menu.
6. Check the **Protect from accidental deletion** checkbox.
7. Click **OK**.

Using Windows PowerShell

To protect an OU from accidental deletion, use the following line of PowerShell on a system with the Active Directory module for Windows PowerShell installed:

```
Set-ADOrganizationalUnit -Identity "OU=Organizational
Unit,DC=LucernPub,DC=com" -ProtectedFromAccidentalDeletion $true
```

Replace `OU=Organizational Unit,DC=LucernPub,DC=com` with the OU you want to protect from accidental deletion.

How it works...

When objects are protected from accidental deletion, they cannot be deleted or moved. This prevents common errors when managing Active Directory but works best when combined with regular backups and backup tests.

User objects, computer objects, **Managed Service Accounts** (**MSAs**), **group Managed Service Accounts** (**gMSAs**), groups, password settings objects, and OUs can be protected against accidental deletion, but GPOs can't.

When the **Protect from accidental deletion** option is enabled for an object, the security descriptors are changed:

- The object itself gains a `Deny` access control entry for `Everyone` to have `Delete` access. When the object is an OU, it also gains a `Deny` access control entry for `Everyone` to have the `Delete Tree` access.
- In the parent object, a `Deny` access control entry is set for `Everyone` to have `Delete Child` access.

The **Protect from accidental deletion** option is not an attribute, but it is exposed as a property in Windows PowerShell.

There's more...

Microsoft recommends protecting all OUs from accidental deletion. However, several OUs in the Active Directory domain are not protected automatically and may not have been created with the right settings enabled.

To protect all OUs from accidental protection, use the following line of PowerShell on a system with the Active Directory module for Windows PowerShell installed:

```
Get-ADOrganizationalUnit -filter {name -like "*"}
-Properties ProtectedFromAccidentalDeletion | Where-
Object {$ _ .ProtectedFromAccidentalDeletion -eq $false} |
Set-ADOrganizationalUnit -ProtectedFromAccidentalDeletion $true
```

The preceding line of PowerShell typically only needs to be run once to protect all OUs.

Implementing LAPS

Microsoft's free LAPS allows admins to periodically change the password for the local administrator password on domain-joined devices. This recipe shows how to implement and use it.

Getting ready

First, download LAPS from `http://aka.ms/LAPS`. Download the `*.msi` file that corresponds to the client operating system architecture(s) used in the organization. Most likely, this will be x64. Place the file on a share.

Ensure that all domain controllers in the environment run Windows Server 2003 with Service Pack 1 or a newer version of Windows Server.

If your organization places devices in the default **Computers** container, move the computer objects that you want to be part of LAPS from this container to an OU dedicated to devices.

How to do it...

There are two sides to LAPS – implementing it and managing it.

Implementing LAPS

Implementing LAPS requires four steps:

Extending the schema

Perform these steps to extend the Active Directory schema with LAPS extensions:

1. Sign in to a domain controller or Windows Server-based management server that has .NET Framework 4.0 installed (or a newer version of .NET Framework) with an account that is a member of the **Schema Administrators** group.
2. Double-click the MSI installer to install LAPS on Windows Server.
3. Follow the instructions to install LAPS.
4. Right-click Start.
5. Choose **Windows PowerShell (Admin)** from the menu.
6. Type the following two lines of PowerShell to import LAPS PowerShell module and extend the Active Directory schema:

```
Import-Module AdmPwd.PS
Update-AdmPwdADSchema
```

This adds the `mS-MCS-AdmPwd` and `mS-MCS-AdmPwdExpirationTime` attributes to the schema for computer objects.

Setting permissions

Next, set permissions in Active Directory to enable devices to write to the new `mS-MCS-AdmPwd` and `mS-MCS-AdmPwdExpirationTime` attributes. Perform these steps:

1. Sign in to the domain controller or Windows Server-based management server that has LAPS installed, with an account that is a member of the **Domain Admins** group or is delegated full control over the OU containing the devices in the scope for LAPS (and its child OUs).

2. Right-click Start.
3. Choose **Windows PowerShell (Admin)** from the menu.
4. Type the following two lines of PowerShell to import LAPS PowerShell module and set permissions on the OU that contains computer objects for devices that are in scope:

```
Import-Module AdmPwd.PS
Set-AdmPwdComputerSelfPermission -OrgUnit "OU ShortName"
```

Do not run the preceding PowerShell command on the entire directory, as it would also include domain controllers. Repeat the last line of PowerShell from *step 4* to bring additional OUs into scope if they are not sub-containers of previously included OUs.

Creating the GPO to install the LAPS client-side extensions

Perform these steps to install the LAPS **Client-Side Extensions** (**CSEs**):

1. Sign in to a system with **Group Policy Management** (`gpmc.msc`) installed, with an account that is either a member of the **Domain Admins** group, the current owner of an existing GPO, or delegated the **Edit settings** or **Edit settings, delete, modify security permission** on an existing GPO.
2. Press Start.
3. Search for **Group Policy Management** and click its search result, or run `gpmc.msc`. The **Group Policy Management** window appears.
4. In the left navigation pane, navigate to the **Group Policy objects** node.
5. Locate the GPO that you want to use and select it, or right-click the **Group Policy Objects** node and select **New** from the menu.
6. Right-click the GPO and select **Edit…** from the menu. The **Group Policy Management Editor** window appears.
7. In the main pane of the **Group Policy Management Editor** window, expand the **Computer Configuration** node, then **Policies**, and then the **Software Settings** node.
8. Right-click the **Software Installation** node and select **New** from the menu, and then **Package…**.

9. In the **Open** screen, browse to the network share that has the LAPS MSI package. Select the application and click **Open**.
10. In the **Deploy Software** pop-up screen, select **Assigned**.
11. Click **OK** to save the settings. The package will be listed with its version, its deployment state, and source path.
12. In the left navigation window, expand the **Administrative Templates** node and then **LAPS** node.
13. Double-click the **Enable local admin password management** setting and enable it.
14. Click **OK**.
15. Double-click the **Do not allow password expiration time longer than required by policy** setting and enable it.
16. Click **OK**.
17. Close the **Group Policy Management Editor** window.

Linking the GPO to OUs with devices

Perform these steps to link the GPO to OUs with devices in scope for the LAPS:

1. Sign in to a system with the **Group Policy Management** Console feature installed, with an account that is either a member of the **Domain Admins** group or the current owner of the GPO, and has the **Link GPOs** permission on the OU(s), site(s), and/or domain(s) where the GPO is to be linked, or is delegated the **Edit settings** or **Edit settings, delete, modify security** permissions on the GPO, and has the **Link GPOs** permission on the OU(s) where the GPO is to be linked.
2. Press Start.
3. Search for **Group Policy Management** and click its search result or run `gpmc.msc`. The **Group Policy Management** window appears.
4. In the left navigation pane, navigate to the OU where you want to link the LAPS GPO.
5. Right-click the OU and select **Link an Existing GPO…** from the menu.
6. In the **Select GPO** window, select **LAPS GPO**.
7. Click **OK** to link the GPO.

Repeat *steps 5–7* to link the LAPS GPO to all OUs that require the LAPS GPO. Take **block inheritance** into account for OUs by linking the LAPS GPO specifically to include all devices in its scope.

Managing passwords

After LAPS is implemented, the passwords in the LAPS' Active Directory attributes can be viewed and managed by accounts that have the **All extended rights** permission on computer objects in scope. By default, only members of the **Domain Admins** and **Enterprise Admins** security groups have this permission. The LAPS UI is the preferred tool to manage passwords.

Viewing an administrator password

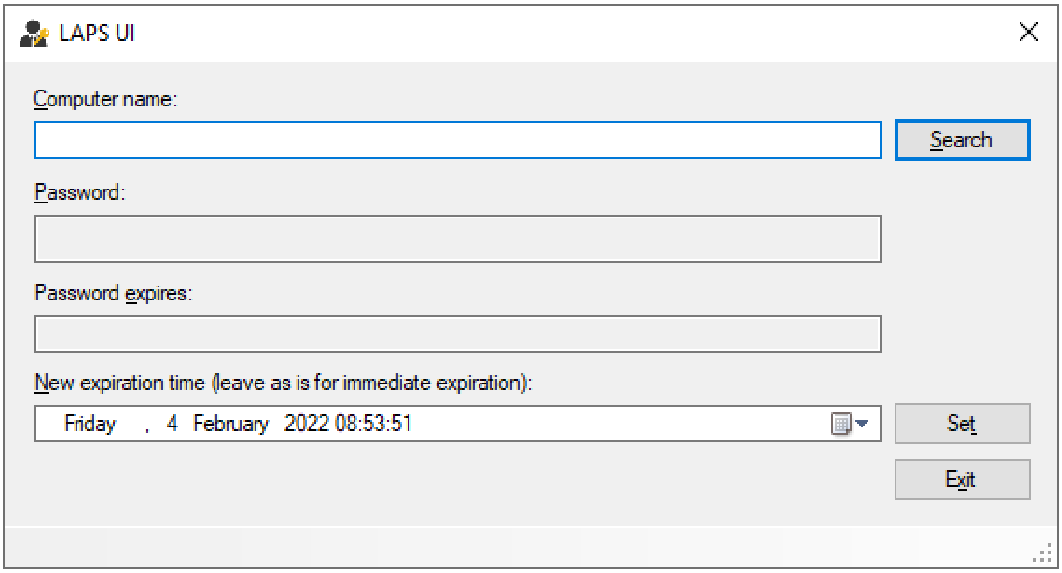
Perform these steps to view an administrator password:

1. Sign in to the domain controller or Windows Server-based management server that has LAPS installed with an account that is a member of the **Enterprise Admins** group or the **Domain Admins** group.
2. Press Start.
3. Search for **LAPS UI** and select it from the search results. The **LAPS UI** window appears.
4. In the **LAPS UI** window, enter a device name in the **Computer name:** field or use the **Search** button to search for a device.
5. The password is shown in the **Password:** field, together with when the password expires.
6. Press **Exit** to close the LAPS UI.

Resetting an administrator password

Perform these steps to reset an administrator password:

1. Sign in to the domain controller or Windows Server-based management server that has LAPS installed with an account that is a member of the **Enterprise Admins** group or the **Domain Admins** group.
2. Press Start.
3. Search for **LAPS UI** and select it from the search results. The **LAPS UI** window appears:



Computer name:

Password:

Password expires:

New expiration time (leave as is for immediate expiration):

Friday , 4 February 2022 08:53:51

Figure 11.10 – The LAPS UI window

4. In the **LAPS UI** window, enter a device name in the **Computer name:** field or use the **Search** button to search for a device.
5. Press the **Set** button to immediately expire the password and have the LAPS CSE on the device communicate with Active Directory to set a new password and reset the **Password expires:** time frame.
6. Press **Exit** to close the **LAPS UI** window.

How it works...

LAPS has three components:

- The LAPS GPO instructs domain-joined devices in scope with settings.
- The LAPS CSEs set and exchange clear-text passwords with Active Directory, based on the GPO.
- The LAPS attributes for computer objects in Active Directory store passwords and expiration time frames for LAPS. Devices have `SELF` permissions to write to these attributes.

Passwords for local administrator accounts are stored in clear text in the `mS-MCS-AdmPwd` attribute. By default, only members of the Enterprise Admins and **Domain Admins** security groups have access to these attributes, although the permissions to managing passwords can be delegated in the following ways:

- By delegating the **All Extended Rights** permission on one or more OUs containing computer objects of devices in scope of the LAPS
- By delegating `CONTROL_ACCESS` or `READ` permissions to the `mS-MCS-AdmPwd` attribute for computer objects on one or more OUs

See also

Refer to the *Installing applications* and *Linking a GPO to an OU* recipes in *Chapter 10, Getting the Most Out of Group Policy*, for more information on these topics.

Managing deleted objects

This recipe shows how to manage deleted objects in an Active Directory environment with the Active Directory Recycle Bin enabled.

Getting ready

To manage deleted objects, sign in to a domain controller, a Windows Server-based management server with the RSAT for Active Directory Domain Services installed, or a Windows installation with the RSAT installed with an account that is a member of the **Domain Admins** group.

How to do it...

This recipe shares two ways to manage deleted objects:

- Using the **Active Directory Administrative Center**
- Using Windows PowerShell

Using the Active Directory Administrative Center

To manage deleted objects using the **Active Directory Administrative Center**, perform these steps:

1. Press Start.
2. Search for **Active Directory Administrative Center** and select it from the search results, or run `dsac.exe`. The **Active Directory Administrative Center** window appears.

3. In the left navigation pane, switch to the tree view.
4. Navigate to the **Deleted Objects** container.
5. Perform one of these actions:

    I. When the object to restore is an OU, expand the **Deleted Objects** container and select the OU to restore. Right-click it and select **Restore** from the menu.

    II. When the object to restore is a user object, contact, computer object, or group, select it in the main pane. Right-click it and select **Restore** from the menu.

Using Windows PowerShell

To view the deleted objects for a domain, use the following line of PowerShell on a system with the Active Directory module for Windows PowerShell installed:

```
Get-ADObject -ldapFilter:"(msDS-LastKnownRDN=*)"
-IncludeDeletedObjects
```

To restore a deleted object, use the following line of PowerShell on a system with the Active Directory module for Windows PowerShell installed:

```
Get-ADObject -Filter {displayName -eq "DisplayNameOfTheObject"}
-IncludeDeletedObjects | Restore-ADObject
```

Replace `DisplayNameOfTheObject` with the display name of the object to restore.

How it works...

When the Active Directory Recycle Bin is not enabled, objects that are deleted are tombstoned. This allows domain controllers to replicate the deletion. When the Active Directory Recycle Bin is enabled, deleted objects are not tombstoned immediately but recycled first.

In this state, deleted objects are shown in the **Deleted Objects** container. Objects can be restored from this container to their original location or a different container, including all their group memberships and other attributes.

There's more...

In the **Active Directory Administrative Center**, multiple objects can be selected in the main view by selecting them, or by using the *Shift* and *Ctrl* keys. Then, the selection can be used to restore multiple objects at once.

See also

For more information on the Active Directory Recycle Bin, refer to the *Enabling the Active Directory Recycle Bin* recipe from *Chapter 1*, *Optimizing Forests, Domains, and Trusts*.

Working with gMSAs

gMSAs are managed domain accounts that you use to help secure services. This recipe shows how to work with gMSAs.

Getting ready

To create gMSAs, the Active Directory domain needs to have at least one domain controller running Windows Server 2012 or a newer version.

gMSAs can only be used to run services on domain-joined hosts running Windows Server 2012 and newer versions, or Windows 8 and newer versions.

For the automatic password and **Service Principal Name** (**SPN**) management functionality, the domain needs to run at least Windows Server 2008 R2 DFL.

As gMSAs depend on the Key Distribution Service on domain controllers, prepare the forest by running the following line of PowerShell on a system with the Active Directory module for Windows PowerShell:

```
Add-KdsRootKey -EffectiveImmediately
```

How to do it...

1. To create a gMSA, use the following line of PowerShell on a system with the Active Directory module for Windows PowerShell installed:

   ```
   New-ADServiceAccount -Name MSAName -DNSHostName MSAName.
   domain.tld -PrincipalsAllowedToRetrieveManagedPassword
   "CN=AppServer1,CN=Computers,DC=LucernPub,DC=com"
   ```

2. To install the gMSA on an application server so that it can be assigned to run a service, application, or application pool, use the following lines of PowerShell in an elevated Windows PowerShell window:

   ```
   Install-WindowsFeature RSAT-AD-PowerShell
   Install-ADServiceAccount -Identity MSAName
   Uninstall-WindowsFeature RSAT-AD-PowerShell
   ```

In the previous PowerShell examples, replace `MSAName` with the `sAMAccountName` attribute of the gMSA, and replace `CN=AppServer1,CN=Computers,DC=LucernPub,DC=com` with the domain-joined host on which you want to use the gMSA.

How it works...

Service accounts are notoriously hard for admins to get right. User objects are reused for this purpose, and they are typically over-privileged and not sufficiently secured, and admins rarely change the passwords for these accounts out of fear of breaking functionality.

Managed Service Accounts (**MSAs**) were introduced in Windows Server 2008 R2 to solve this problem. In Windows Server 2012, MSAs were superseded by gMSAs. Since then, when you create this type of object as an admin, you create a gMSA by default.

The main difference between an MSA and a gMSA is that a gMSA can be used as a service account on more than one server, where an MSA is limited to one server.

gMSAs are `msDS-GroupManagedServiceAccount` objects. They are not based on user objects but on computer objects. Just as with computer objects, they are prohibited from signing in interactively to systems and automatically change their passwords every 30 days by default. This makes them much more secure than service accounts based on user objects.

gMSAs use a password that is stored in the `msDS-ManagedPassword` attribute of the object. Only domain-joined servers that are listed in the `msDS-GroupMSAMembership` attribute are provided access to the attribute by the Key Distribution Service on domain controllers.

Although the line of PowerShell to create a gMSA specifies the root key to be effective immediately with the `-EffectiveImmediately` switch, you will actually have to wait 10 hours for it to become active. This ensures that there is ample time to replicate the information to other domain controllers.

There's more...

The interval that gMSAs use to change their passwords can be controlled using the `msDS-ManagedPasswordInterval` attribute. Even if password changes for computer objects has been turned off, gMSAs will continue to change their passwords. If the interval can be higher in your environment because strict regulations don't apply, set the attribute to a higher value (in days) after creating gMSAs.

Configuring diagnostic logging

This recipe shows how to configure diagnostic logging to provide additional information when troubleshooting Active Directory problems.

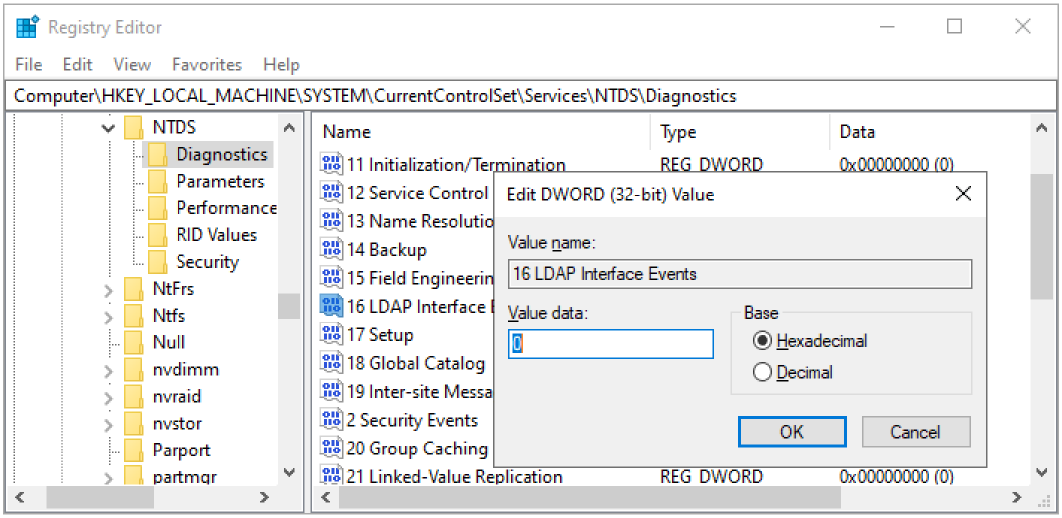
Getting ready

To configure diagnostic logging on a domain controller, sign in to it using an account that is a member of the **Domain Admins** group.

How to do it…

Perform these steps to configure diagnostic logging:

1. Press Start.
2. Search for **Registry Editor** and click its search result, or run `regedit.exe`. The **Registry Editor** window appears.
3. In the location bar at the top of the **Registry Editor** window, type `HKLM\SYSTEM\CurrentControlSet\Services\NTDS\Diagnostics`.
4. In the main pane, select the area that you want to enable diagnostic logging for.
5. Right-click the name of the logging area and select **Modify…** from the menu. The **Edit DWORD (32-bit) Value** pop-up window appears:



Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NTDS\Diagnostics

Name	Type	Data
11 Initialization/Termination	REG_DWORD	0x00000000 (0)
12 Service Control		
13 Name Resolutio		
14 Backup		
15 Field Engineerin		
16 LDAP Interface E		
17 Setup		
18 Global Catalog		
19 Inter-site Messa		
2 Security Events		
20 Group Caching		
21 Linked-Value Replication	REG_DWORD	0x00000000 (0)

Value name: 16 LDAP Interface Events

Value data: 0

Base: ● Hexadecimal ○ Decimal

Figure 11.11 – Configuring diagnostic logging for 16 LDAP Interface Events

6. In the **Value data:** field, enter the logging level value.
7. Click **OK**.
8. Close the **Registry Editor** window.
9. Restart the **Active Directory Domain Services** service or the domain controller for the changes to take effect.

How it works...

With default settings, domain controllers record critical events and error events in the **Directory Service** log only. To configure domain controllers to record other events to help diagnose and resolve problems, increase the logging level in the registry.

There are 27 information areas that diagnostic logging can be configured for:

1. **Knowledge Consistency Checker (KCC)**
2. **Security Events**
3. **ExDS Interface Events**
4. **MAPI Interface Events**
5. **Replication Events**
6. **Garbage Collection**
7. **Internal Configuration**
8. **Directory Access**
9. **Internal Processing**
10. **Performance Counters**
11. **Initialization/Termination**
12. **Service Control**
13. **Name Resolution**
14. **Backup**
15. **Field Engineering**
16. **LDAP Interface Events**
17. **Setup**
18. **Global Catalog**
19. **Intersite Messaging**

20. **Group Caching**
21. **Linked-Value Replication**
22. **DS RPC Client**
23. **DS RPC Server**
24. **DS Schema**
25. **Transformation Engine**
26. **Claims-Based Access Control**
27. **PDC Password Update Notifications**

These areas can each be configured with their own logging level:

Logging level	**Value**
None (default)	`0`
Minimal	`1`
Basic	`2`
Extensive	`3`
Verbose	`4`
Internal	`5`

Table 11.1 – Logging levels and their corresponding values

Logging can degrade the performance of domain controllers. Therefore, logging levels should be configured with the value `0`, unless you are troubleshooting or monitoring an issue.

Configuring the advanced security audit policy

This recipe shows how to configure the advanced security audit policy.

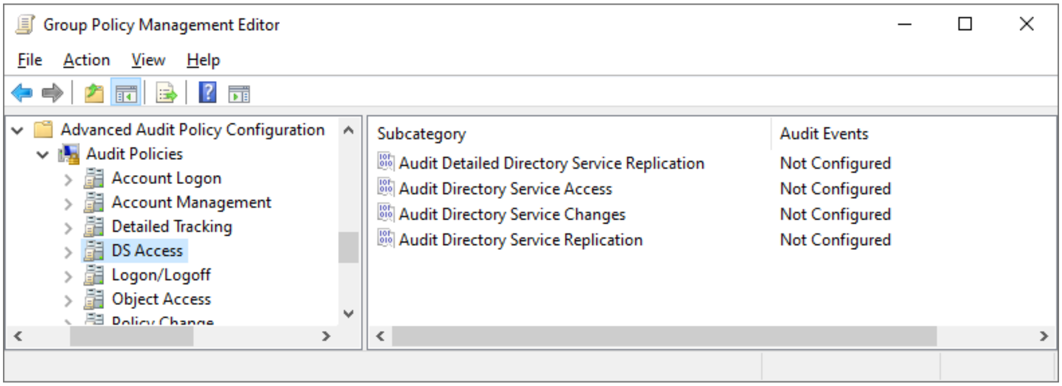
Getting ready

To configure the advanced security audit policy, sign in to a domain controller with a user account that is a member of the **Domain Admins** group.

How to do it...

Perform these steps to configure the advanced security audit policy:

1. Press Start.
2. Search for **Group Policy Management** and click its search result, or run `gpmc.msc`. The **Group Policy Management** window appears.
3. In the left navigation pane, navigate to the **Domain Controllers** OU for the domain in which you want to configure the advanced security audit policy and expand it.
4. Select **Default Domain Controllers Policy**. The **Group Policy Management Console** pop-up window may appear, notifying you that you have selected a link to a GPO and that changes you make here are global to it and will impact all other locations where it is applied. Click **OK**.
5. Right-click **Default Domain Controllers Policy** and select **Edit…** from the menu. The **Group Policy Management Editor** window opens.
6. In the left navigation window, expand **Computer Configuration**, **Policies**, **Windows Settings**, **Security Settings**, **Advanced Audit Policy Configuration**, **Audit Policies**, and then **DS Access**:



Subcategory	Audit Events
Audit Detailed Directory Service Replication	Not Configured
Audit Directory Service Access	Not Configured
Audit Directory Service Changes	Not Configured
Audit Directory Service Replication	Not Configured

Figure 11.12 – The DS Access subcategories in the Group Policy Management Editor window

7. In the main pane, double-click the **Audit Directory Service Changes** setting. The **Audit Directory Service Changes Properties** window opens.
8. Check the **Configure the following auditing events:** option. Then, check to audit **Success** and/or **Failure** audit events.
9. Click **OK** to save the settings and close the **Audit Directory Service Changes Properties** screen.
10. Close the **Group Policy Management Editor** window.

How it works...

Microsoft introduced the advanced security audit policy in Active Directory in Windows Server 2008 R2. This feature offers more granular auditing options in 10 categories:

- Account Logon
- Account Management
- Detailed Tracking
- DS Access
- Logon/Logoff
- Object Access
- Policy Change
- Privilege Use
- System
- Global Object Access Auditing

For each of these categories, several auditing options are available. When these options are enabled, additional entries are added to the event log with the source **Microsoft Windows security auditing**.

A recommended practice is to copy auditing events from event logs on the domain controller to a centralized **Security Incident and Event Management** (**SIEM**) solution.

Resetting the KRBTGT secret

This recipe shows how to reset the password of the **KRBTGT** account.

Getting ready

To reset the password for the **KRBTGT** account, sign in to a domain controller with a user account that is a member of the **Domain Admins** group.

How to do it...

Perform the following line of PowerShell:

```
Set-ADAccountPassword -Identity (Get-ADUser krbtgt).
DistinguishedName -Reset -NewPassword (ConvertTo-SecureString
"Rand0mCompl3xP@ssw0rd!" -AsPlainText -Force)
```

Replace `Rand0mCompl3xP@ssw0rd!` with the new complex password for the **KRBTGT** user object.

How it works...

Each Active Directory domain in a multi-domain environment has its own **KRBTGT** account used by all fully writable domain controllers. Each read-only domain controller has its own **KRBTGT_*** account.

The password hash for the **KRBTGT** account is used as the secret to encrypt all Kerberos tickets.

The password for **KRBTGT** is set during the creation of an Active Directory domain. Microsoft has only automatically reset the secret on the **KRBTGT** account for Active Directory domains when the DFL was upgraded to Windows Server 2008, or beyond.

When the **KRBTGT** secret is found out, a malicious person will not just be able to read all Kerberos authentication traffic. When a malicious person wants to attain a foothold in Active Directory, the most common way to do so is by forging tickets, as in a golden ticket attack. As ticket control in Active Directory is on the client side, malicious people can (re) use forged Kerberos tickets for as long as 10 years.

To prevent the **KRBTGT** secret from being found out, the only way to lock out malicious persons using forged Kerberos tickets is to regularly reset the password for **KRBTGT** with different values.

The process for signing tickets is designed to handle password changes without locking out legitimate users. Any ticket that has been signed before the password change will use the fallback method provided for the **Ticket Granting Ticket** (**TGT**) lifetime. This lifetime, by default, is 7 days.

Replication convergence may take time throughout a large Active Directory environment.

Therefore, the password for **KRBTGT** needs to be reset twice with an interval in between. Microsoft's recommendation is to reset the password for **KRBTGT** twice a year.

There's more...

Microsoft offers a script in the TechNet Gallery to automate the processes of generating complex passwords, changing a password, and checking for proper replication:

`https://github.com/microsoft/New-KrbtgtKeys.ps1`

Using the SCW to secure domain controllers

This recipe shows how to secure domain controllers running older versions of Windows Server, using the Windows Server SCW and Group Policy.

Getting ready

To secure domain controllers using the SCW, sign in to a domain controller with a user account that is a member of the **Domain Admins** group.

The Security Configuration Wizard was removed from Windows Server 2016 and is not present in Windows Server versions since this version. Features are secured by default. This recipe applies to full installations of the following Windows Server versions:

- Windows Server 2008
- Windows Server 2008 R2
- Windows Server 2012
- Windows Server 2012 R2

How to do it

Securing domain controllers using the Windows Server SCW and Group Policy consists of two steps:

1. Securing a representative domain controller using the SCW
2. Rolling out the security settings to all domain controllers using Group Policy

Secure a representative domain controller using the SCW

Perform these steps on a domain controller in a test environment to test the settings in the context of routine administration processes, before rolling the settings out to all production domain controllers:

1. Press Start.
2. Search for **Server Manager** and click its search result, or run `servermanager.exe`. The **Server Manager** window appears.
3. From the **Tools** menu in the top gray pane, select **Security Configuration Wizard**. The **Security Configuration Wizard** window appears.
4. On the **Welcome** screen, click **Next**.

5. On the **Configuration Action** screen, select the **Create a new security policy** option.
6. Click **Next**.
7. On the **Select Server** screen, select the local server. Click **Next** >.
8. On the **Role-based Service Configuration** screen, click **Next** >.
9. On the **Select Server Roles** screen, click **Next** >.
10. On the **Select Server Features** screen, click **Next** >.
11. On the **Select Administration and Other Options** screen, click **Next** >.
12. On the **Select Additional Services** screen, click **Next** >.
13. On the **Handling Unspecified Services** screen, select the **Disable the service** option and click **Next** >.
14. On the **Confirm Service Changes** screen, click **Next** >.
15. On the **Network Security** screen, click **Next** >.
16. On the **Network Security Rules** screen, click **Next** >.
17. On the **Registry Settings** screen, click **Next** >.
18. On the **Require SMB Security Signatures** screen, check the properties of the Windows Server installation. These properties determine the SMB signing settings.
19. Click **Next** >.
20. On the **Require LDAP Signing** screen, select the **Windows 2000 Service Pack 3 or later** option.
21. Click **Next** >.
22. On the **Outbound Authentication Methods** screen, check the **Domain Accounts** option.
23. Click **Next** >.
24. On the **Outbound Authentication using Domain Accounts** screen, select both options to require NTLM version 2.
25. Click **Next** >.
26. On the **Inbound Authentication Methods** screen, deselect all options unless your environment contains devices running Windows XP.
27. Click **Next** >.
28. On the **Registry Settings Summary** screen, click **Next** >.
29. On the **Auditing Policy** screen, click **Next** >.

30. On the **System Auditing Policy** screen, select **Audit successful and unsuccessful activities**.
31. Click **Next** >.
32. On the **Audit Policy Summary** screen, click **Next** >.
33. On the **Save Security Policy** screen, click **Save**.
34. Save the file with its `.xml` extension.
35. On the **Apply Security Policy** screen, click **Next** >.
36. On the **Application Complete** screen, click **Finish**.

Roll out the security settings to all domain controllers using Group Policy

Run the following command line on an elevated Command Prompt (`cmd.exe`) to convert the settings file into a GPO:

```
scwcmd.exe transform /p:"C:\Windows\security\msscw\Policies\test.
xml" /g:"Domain Controller Security Settings"
```

The preceding command creates a GPO with the name **Domain Controller Security Settings**. Change `Domain Controller Security Settings` in the preceding PowerShell example to adhere to your organization's GPO naming convention, if needed.

Next, link the new GPO to the Domain Controllers' OU by performing these steps:

1. Search for **Group Policy Management** and click its search result, or run `gpmc.msc`. The **Group Policy Management** window appears.
2. In the left navigation pane, expand the **Forest** node.
3. Expand the **Domains** node, and then navigate to the domain where you want to link the GPO.
4. Expand the domain name.
5. Navigate to the **Domain Controllers** OU.
6. Right-click the OU and select **Link an Existing GPO…** from the menu.
7. In the **Select GPO** window, select the GPO you want to link from the **Group Policy objects:** list.
8. Click **OK** to link the GPO.

How it works...

The Windows Server SCW guides admins through the following settings:

- Permitted server roles and server features
- Permitted remote access
- Permitted services
- SMB and LDAP settings
- Auditing settings

This way, the wizard allows for straightforward management of these settings.

While the settings can be applied on a per-domain controller basis, a Group Policy can be applied with the settings, after the file is converted to a GPO using `scwcmd.exe`. After that, the GPO can be linked to the Domain Controllers' OU to apply the settings to all domain controllers.

Leveraging the Protected Users group

This recipe shows how the **Protected Users** security group can be used to protect privileged and sensitive accounts.

Getting ready

To use the **Protected Users** security group, ensure the domain runs the Windows Server 2012 R2 DFL or a newer version of the level. Also, be aware that the protections offered by the **Protected Users** security group only apply when accounts that are members of the group are used on devices running Windows 8.1 or newer, and Windows Server 2012 R2 or newer.

To manage the **Protected Users** security group, sign in to a domain controller or a member server and/or device with the RSAT for Active Directory Domain Services installed.

Sign in with an account that is a member of the **Domain Admins** group, the **Account Operators** group, or with an account that is delegated to manage groups in the domain or in the scope of the OU.

How to do it…

This recipe shares three ways to manage group memberships in Active Directory:

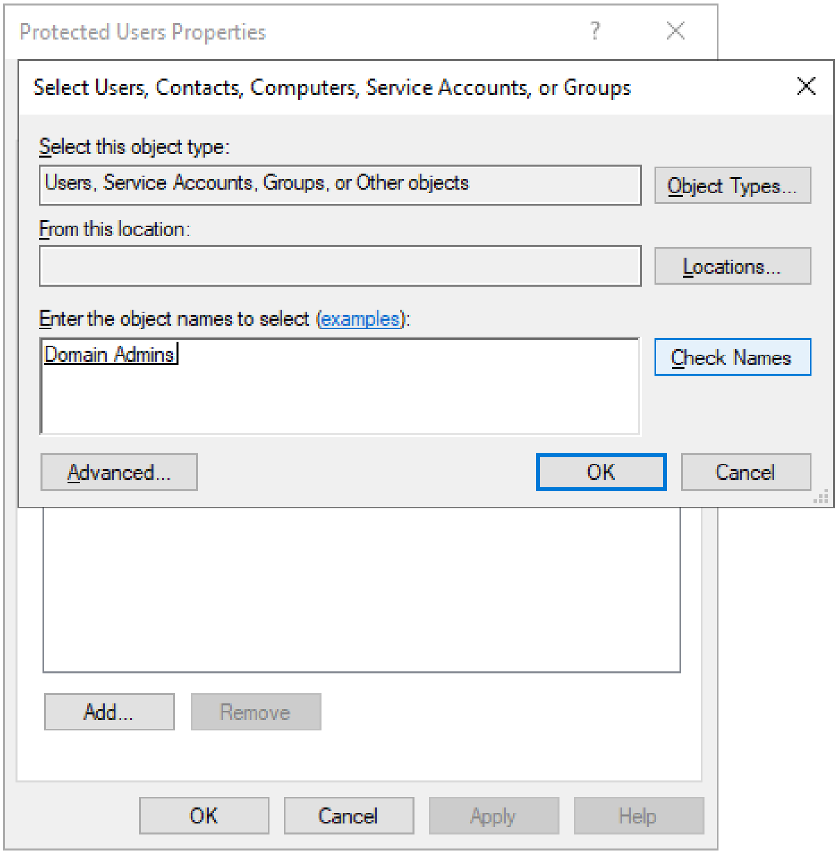
- Using **Active Directory Users and Computers**
- Using the **Active Directory Administrative Center**
- Using Windows PowerShell

Using Active Directory Users and Computers

Perform these steps to add user accounts to the **Protected Users** security group using **Active Directory Users and Computers**:

1. Press Start.
2. Search for **Active Directory Users and Computers** and click its search results, or run `dsa.msc`. The **Active Directory Users and Computers** window appears.
3. From the **Action** menu, select **Find…**. In the **Name** field, type **Protected Users** and press *Enter*. From the **Search results:** list, select the group.
4. Right-click the group and select **Properties** from the menu. The **Protected Users Properties** window appears.
5. Navigate to the **Members** tab.
6. Click **Add…** to add users, contacts, computers, service accounts, or groups to the group. The **Select Users, Contacts, Computers, Service Accounts, or Groups** window appears.
7. In the **Select Users, Contacts, Computers, Service Accounts, or Groups** window, type the name of the object(s) you want to add to the group or click the **Advanced** button to search for the object(s).

8. Click **Check Names**:



Protected Users Properties
Select Users, Contacts, Computers, Service Accounts, or Groups
Select this object type:
Users, Service Accounts, Groups, or Other objects
Object Types...
From this location:
Locations...
Enter the object names to select (examples):
Domain Admins
Check Names
Advanced...
OK
Cancel
Add...
Remove
OK
Cancel
Apply
Help

Figure 11. 13 – Check Names in the Select Users, Contacts, Computers, Service, or Groups window

9. Click **OK** to add the user, contact, computer, service account, or group to the **Protected Users** group.

10. Click **OK** to close the **Protected Users Properties** window and save the changes.

Using the Active Directory Administrative Center

Perform these steps to add user accounts to the **Protected Users** security group using the **Active Directory Administrative Center**:

1. Press Start
2. Open the **Active Directory Administrative Center** (`dsac.exe`).
3. From the main pane menu, under **Global Search**, type the name of the group and press *Enter*.
4. From the list of **Global Search** results, select the group.
5. Right-click the group and select **Properties** from the list.
6. In the left navigation pane, click **Members**.
7. Click **Add…** to add users, contacts, computers, service accounts, or groups to the group.
8. In the **Select Users, Contacts, Computers, Service Accounts, or Groups** window, type the name of the user account(s) you want to add to the group or click the **Advanced** button to search for them.
9. Click **Check Names**.
10. Click **OK** to add the user to the group.
11. Click **OK** to close the **Group Properties** window and save the changes.

Using Windows PowerShell

Use the following line of PowerShell to add a user to the **Protected Users** security group in Active Directory on a system with the Active Directory module for Windows PowerShell installed:

```
Add-ADGroupMember -Identity "CN=Protected
Users,CN=Users,DC=LucernPub,DC=com" -Members "User"
```

Replace `User` with the name of the user, contact, computer, service account, or group you want to add to the **Protected Users** security group.

How it works…

The **Protected Users** security group is a new feature in Active Directory since Windows Server 2012 R2. Accounts that are members of the group lose the following abilities:

- Use cached logons.

- Use outdated authentication protocols, such as NTLM, digest authentication, and CredSSP.
- Use weak encryption algorithms, such as DES and RC4, for Kerberos pre-authentication
- Be delegated as part of both **Kerberos Constrained Delegation** (**KCD**) and Kerberos unconstrained delegation
- Use and renew their Kerberos **Ticket Granting Ticket (TGT)** for longer than 240 minutes, compared to the default 10-hour validity and 7-day renewal periods

The preceding protections are non-configurable.

Membership of the **Protected Users** security group is ideal for privileged and sensitive user accounts, such as members of the Domain Users security group. Don't make service accounts, MSAs, gMSAs, or computer objects members of the **Protected Users** security group, as it may break their functionality.

See also

To configure the protections that membership of the **Protected Users** security group brings in a more granular way, refer to the *Putting authentication policies and authentication policy silos to good use* recipe that follows.

Putting authentication policies and authentication policy silos to good use

This recipe shows how to use authentication policies and authentication policy silos.

Getting ready

To use authentication policies and authentication policy silos, ensure the domain runs the Windows Server 2012 R2 DFL or a newer version of the level. Also, be aware that the protections offered by authentication policies and authentication policy silos only apply when accounts are used on devices running Windows 8.1 or newer versions of Windows, and Windows Server 2012 R2 or newer versions of Windows Server.

To manage authentication policies and authentication policy silos, sign in to a domain controller or a member server and/or device with the RSAT for Active Directory Domain Services installed. Sign in with an account that is a member of the **Domain Admins** group.

How to do it...

Putting authentication policies and authentication policy silos to good use consists of five steps:

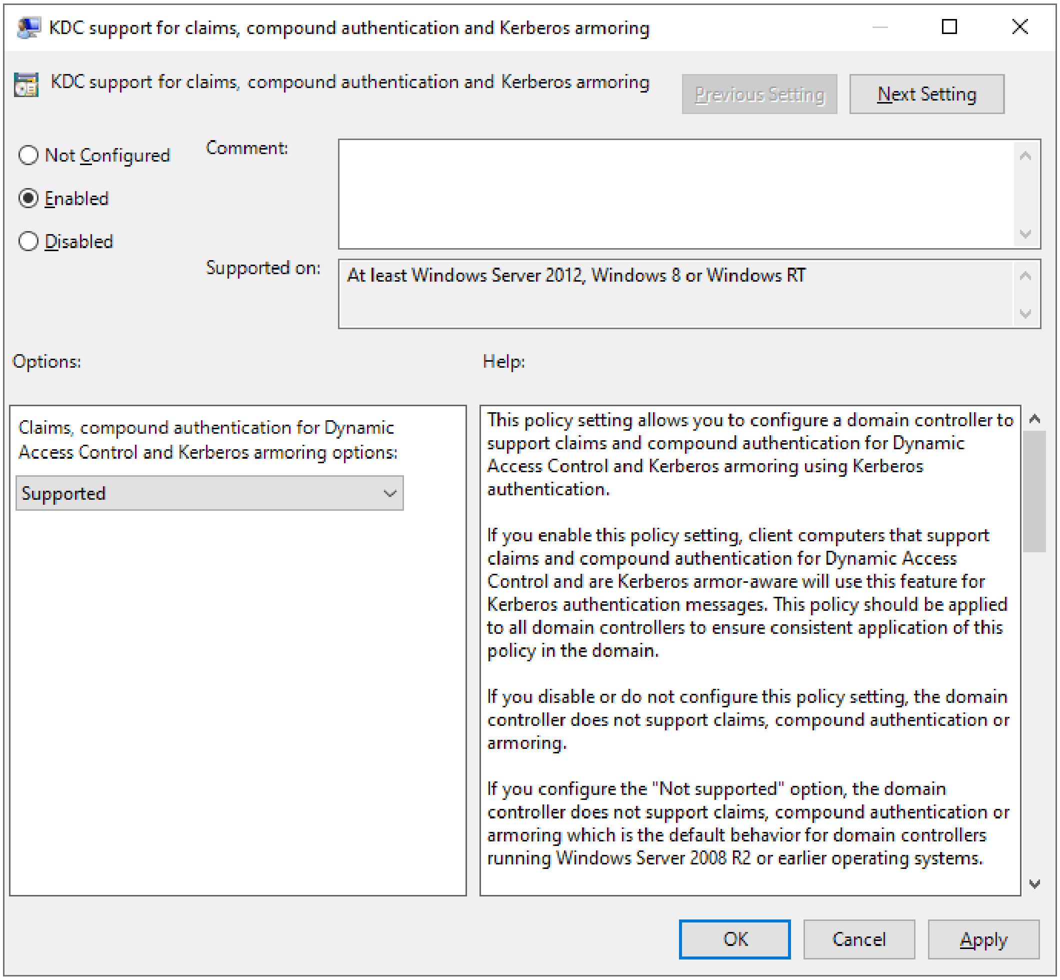
- Enabling domain controller support for claims
- Enabling compound claims on devices in scope of an authentication policy
- Creating an authentication policy
- Creating an authentication policy silo
- Assigning the authentication policy silo

Enabling domain controller support for claims

Perform these steps to enable domain controller support for claims:

1. Press Start.
2. Search for **Group Policy Management** and select it from the search results, or run `gpmc.msc`. The **Group Policy Management** window appears.
3. In the left navigation pane, expand the **Forest** node.
4. Expand the **Domains** node, and then navigate to the domain where you want to enable compound claims on devices.
5. Expand the domain name.
6. Right-click the **Group Policy Objects** node and select **New** from the menu. The **New GPO** pop-up window appears.
7. In the **New GPO** pop-up window, enter the name of the GPO. Ensure you don't select a Starter GPO.
8. Click **OK** to create the GPO.
9. In the main pane, locate the GPO that you want to manage and select it.
10. Right-click the GPO and select **Edit** from the menu. The **Group Policy Management Editor** window appears.

11. In the **Group Policy Management Editor** window, expand **Computer Configuration**, then **Policies**, **Administrative Settings**, and **System**.
12. Select **KDC**.
13. In the main pane, right-click the **KDC support for claims, compound authentication and Kerberos armoring** setting and select **Edit** from the menu.
14. Select the **Enabled** option, as shown in the following screenshot:



KDC support for claims, compound authentication and Kerberos armoring

KDC support for claims, compound authentication and Kerberos armoring

○ Not Configured
● Enabled
○ Disabled

Comment:

Supported on: At least Windows Server 2012, Windows 8 or Windows RT

Options:

Claims, compound authentication for Dynamic Access Control and Kerberos armoring options: Supported

Help:

This policy setting allows you to configure a domain controller to support claims and compound authentication for Dynamic Access Control and Kerberos armoring using Kerberos authentication.

If you enable this policy setting, client computers that support claims and compound authentication for Dynamic Access Control and are Kerberos armor-aware will use this feature for Kerberos authentication messages. This policy should be applied to all domain controllers to ensure consistent application of this policy in the domain.

If you disable or do not configure this policy setting, the domain controller does not support claims, compound authentication or armoring.

If you configure the "Not supported" option, the domain controller does not support claims, compound authentication or armoring which is the default behavior for domain controllers running Windows Server 2008 R2 or earlier operating systems.

Figure 11.14 – Enable KDC support for claims, compound authentication and Kerberos armoring

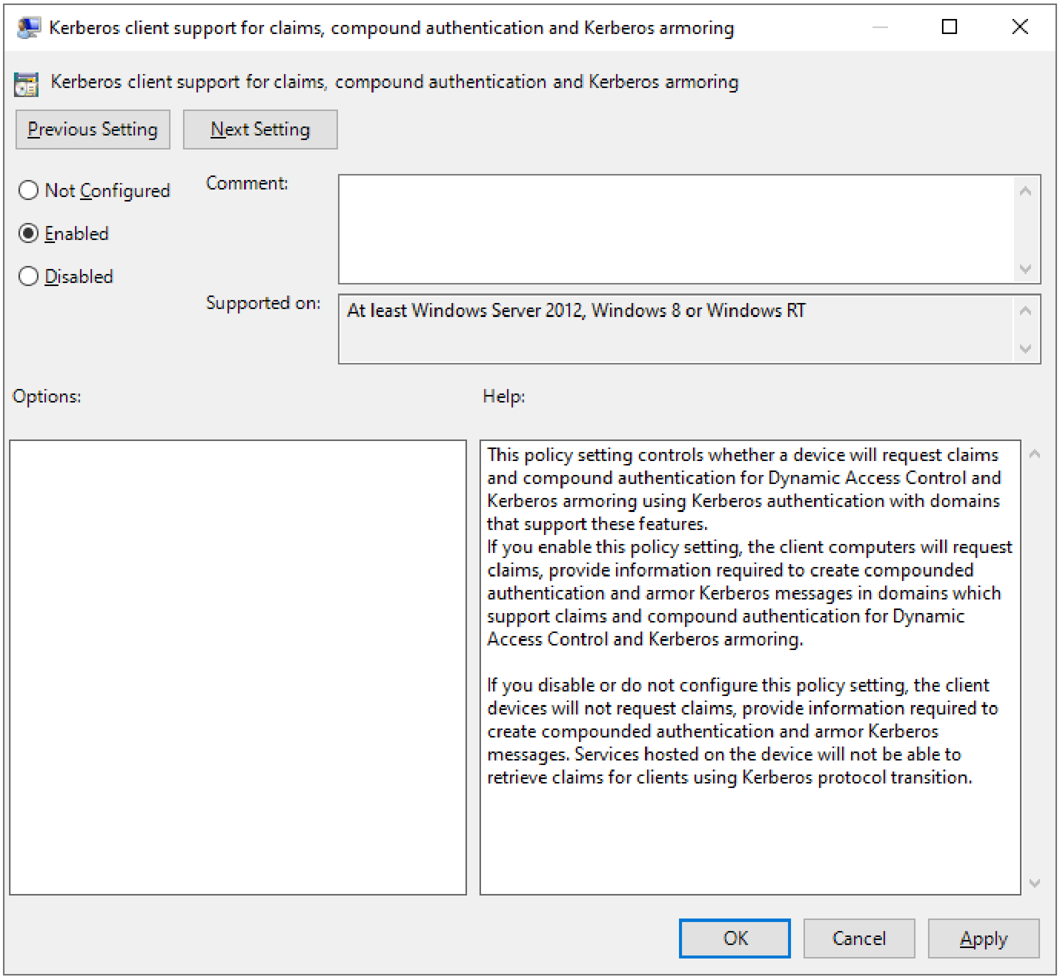
15. Click **OK** to close the **Group Policy Management Editor** window.
16. Link this Group Policy to the **Domain Controllers** OU by right-clicking the OU in the left navigation pane of the **Group Policy Management** window and selecting **Link an Existing GPO…** from the menu. The **Select GPO** window appears.
17. In the **Select GPO** window, select the GPO you want to link from the **Group Policy objects:** list.
18. Click **OK** to link the GPO. This closes the **Select GPO** window.

Enabling compound claims on devices in scope for an authentication policy

Perform these steps to enable compound claims on devices in scope for an authentication policy.

Create another Group Policy by repeating *steps 6–10* from the previous list of steps, and start editing it by repeating *steps 11–18* from the previous list of steps, with the following differences:

1. In the left navigation pane of the **Group Policy Management Editor** window, expand **Computer Configuration**, **Policies**, **Administrative Settings**, and **System**.
2. Select **Kerberos**.
3. In the main pane, right-click the **Kerberos client support for claims, compound authentication and Kerberos armoring** setting and select **Properties** from the menu.
4. Select the **Enabled** option, as shown in the following screenshot:



Kerberos client support for claims, compound authentication and Kerberos armoring

Kerberos client support for claims, compound authentication and Kerberos armoring

○ Not Configured
● Enabled
○ Disabled

Comment:

Supported on: At least Windows Server 2012, Windows 8 or Windows RT

Options:

Help:

This policy setting controls whether a device will request claims and compound authentication for Dynamic Access Control and Kerberos armoring using Kerberos authentication with domains that support these features.
If you enable this policy setting, the client computers will request claims, provide information required to create compounded authentication and armor Kerberos messages in domains which support claims and compound authentication for Dynamic Access Control and Kerberos armoring.

If you disable or do not configure this policy setting, the client devices will not request claims, provide information required to create compounded authentication and armor Kerberos messages. Services hosted on the device will not be able to retrieve claims for clients using Kerberos protocol transition.

Figure 11.15 – Kerberos client support for claims, compound authentication and Kerberos armoring

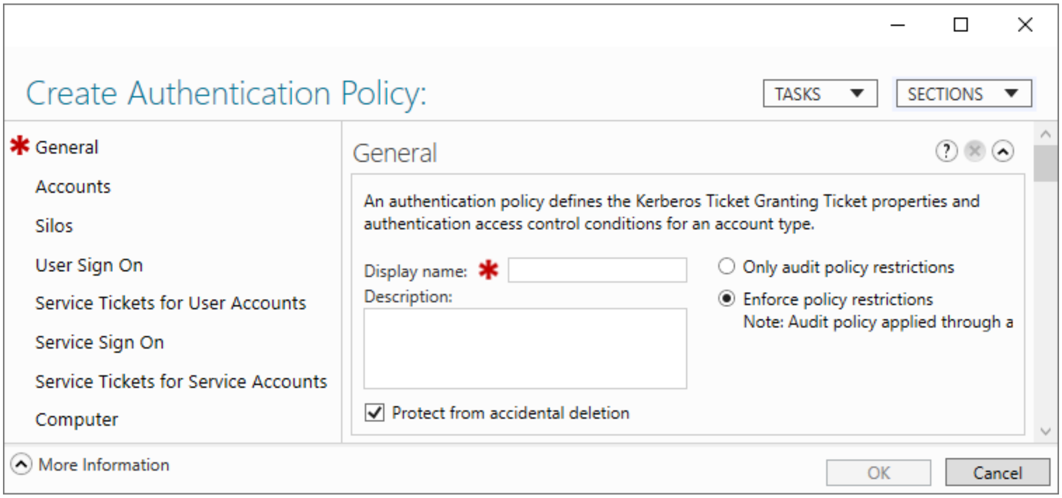
5. Click **OK** to close the **Group Policy Management Editor** window.
6. Link the second Group Policy to the OU(s) with devices in scope, or to the domain by repeating *steps 16–18* from the previous list of steps.

To restrict administrators from using certain devices, do not apply the preceding GPO to the OU(s) containing these devices.

Creating an authentication policy

Perform these steps to create an authentication policy:

1. Press Start.
2. Search for **Active Directory Administrative Center** and click its search result, or run `dsac.exe`. The **Active Directory Administrative Center** window appears.
3. In the left navigation pane, click **Authentication**.
4. In the main pane, select the **Authentication Policies** node.
5. In the **Tasks** pane on the right, click **New** under **Authentication Policies** and select **Authentication Policy** from the menu. The **Create Authentication Policy:** window appears:



An authentication policy defines the Kerberos Ticket Granting Ticket properties and authentication access control conditions for an account type.

○ Only audit policy restrictions

● Enforce policy restrictions
Note: Audit policy applied through a

☑ Protect from accidental deletion

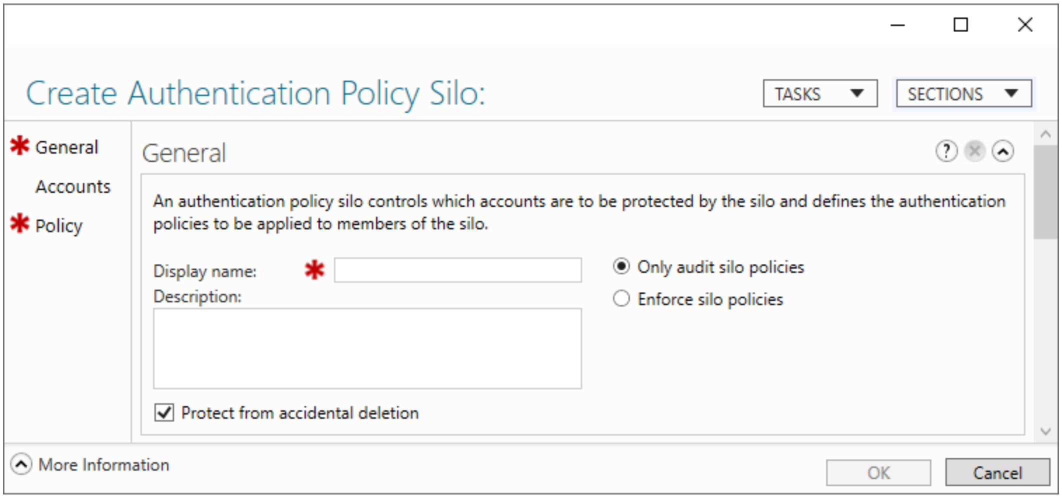
Figure 11.16 – The Create Authentication Policy: window

6. Provide a name for the authentication policy in the **Display name:** field.
7. Optionally, you can also provide a description.
8. In the left navigation list, click **User Sign On**.
9. Select the settings you want to configure, such as the **Specify a Ticket Granting Ticket lifetime for user accounts.** option. Then, select a value between 45 and 2147483647 (2^{31}-1) for **Ticket-Granting-Ticket Lifetime (minutes):** to limit the lifetime of the TGT for objects in scope of this authentication policy.
10. Click **OK** to close the **Create Authentication Policy** window and save its settings.

Creating an authentication policy silo

Perform these steps to create an authentication policy silo:

1. Press Start.
2. Search for **Active Directory Administrative Center** and click its search result, or run `dsac.exe`. The **Active Directory Administrative Center** window appears.
3. In the left navigation pane, click **Authentication**.
4. In the main pane select the **Authentication Policy Silos** node.
5. In the **Tasks** pane on the right, click **New** under **Authentication Policy Silos** and select **Authentication Policy Silo** from the menu. The **Create Authentication Policy Silo:** window appears:


Create Authentication Policy Silo:

General

An authentication policy silo controls which accounts are to be protected by the silo and defines the authentication policies to be applied to members of the silo.

Display name: *

Description:

● Only audit silo policies

○ Enforce silo policies

☑ Protect from accidental deletion

Figure 11.17 – The Create Authentication Policy Silo: window

6. Provide a name for the authentication policy silo in the **Display name:** field.
7. Optionally, you can also provide a description.
8. As the behavior for this authentication policy silo, select the **Enforce silo policies** option.
9. In the left navigation pane, select **Accounts**.
10. In the **Permitted Accounts** list, add one or more objects for which you want the policy silo to apply. Use the **Add…** button to add one or more objects. The **Select Users, Computers, or Service Accounts** pop-up window appears.

11. In the **Select Users, Computers, or Service Accounts** pop-up window, type the name of the user account(s), computer(s), and/or service account(s) you want the authentication policy silo to permit access to, or click the **Advanced** button to search for the user account(s), computer(s), and/or service account(s).
12. Click **Check Names**.
13. Click **OK** to close the pop-up window and add the object(s) to the **Permitted Accounts** list.
14. In the left navigation pane, select **Policy**.
15. Select the **Use a single policy for all principals that belong to this authentication policy silo** option.
16. In the **The authentication policy that applies to all accounts in this silo:** field, select the authentication policy you created in the previous steps from the drop-down list.
17. Click **OK** to close the **Create Authentication Policy Silo:** window and save its settings.

Assigning the authentication policy silo

Perform these steps to assign the authentication policy silo:

1. Press Start.
2. Search for **Active Directory Administrative Center** and click its search result, or run `dsac.exe`. The **Active Directory Administrative Center** window appears.
3. In the left navigation pane, expand **Authentication** and select the **Authentication Policy Silos** node.
4. In the main pane, select the authentication policy silo you created earlier.
5. Right-click it and select **Properties** from the menu.
6. In the left pane, click **Accounts**.
7. Double-click the first item in the **Permitted Accounts** list to open its properties. The properties window for the account will now open.
8. In the left pane, click **Silo**.
9. In the **Authentication Policy Silo** section, select the **Assign Authentication Policy Silo** option. Use the drop-down list for **Authentication Policy Silo:** to select the authentication policy silo created earlier.

10. Click **OK**. This saves the authentication policy silo DN in the object's `msDS-AssignedAuthNPolicySilo` attribute.
11. Repeat *steps 5–8* for all other accounts in the **Permitted Accounts** list for the authentication policy silo.
12. Click **OK** to close the **Create Authentication Policy Silo** window.

How it works…

Using authentication policies and authentication policy silos is a perfect way to set the scene for Microsoft's **Privileged Access Workstation** strategy to prevent people from signing in with their privileged account to devices other than their secure ones. This way, lateral movement toward admin (cached) credentials is hugely limited, benefiting the overall security posture of an organization.

Authentication policies define policies but do not assign these policies to accounts, whereas authentication policy silos assign policies to accounts. An authentication policy can be assigned through many authentication policy silos, when the policies need to be the same but for different accounts.

When a person tries to sign in with an account that is governed by an authentication policy on a device that does not support claims, compound authentication, and Kerberos armoring, signing in to the device is prohibited with the following error:

```
Your account is configured to prevent you from using this PC.
Please try another PC.
```

If this locks out people with legitimate interests, add the device to the scope of the GPO that enables the **Kerberos client support for claims, compound authentication and Kerberos armoring** setting.

Configuring Extranet Smart Lockout

This recipe shows how to configure Extranet Smart Lockout on an **Active Directory Federation Services** (**AD FS**) farm running Windows Server 2016 or newer versions.

Getting ready

When using AD FS on Windows Server, ensure that at least the June 2018 cumulative update for Windows Server 2016 (KB4284880 – `https://support.microsoft.com/en-us/help/4284880/windows-10-update-kb4284880` – and OS Build 14393.2312) is installed on all AD FS servers in the AD FS farm.

Sign in with an account that is an AD FS administrator. By default, members of the **Domain Admins** group have the required permissions.

Sign in to the primary AD FS server when the AD FS farm is using the **Windows Internal Database** (**WID**) as its replication model, or any AD FS server when the AD FS farm leverages SQL Server as its configuration database.

How to do it...

To enable Extranet Smart Account Lockout for an AD FS farm running SQL Server, run the following lines of PowerShell to update the AD FS Artifact Store. These three lines of PowerShell do not need to be run on AD FS farms using WID:

```
$cred = Get-Credential
Import-Module ADFS
Update-AdfsArtifactDatabasePermission -Credential $cred
```

To enable Extranet Smart Account Lockout for an AD FS farm, run the following lines of PowerShell:

```
Set-AdfsProperties -ExtranetLockoutThreshold 10
Set-AdfsProperties -ExtranetObservationWindow (New-Timespan
-minutes 5)
Set-AdfsProperties -EnableExtranetLockout $true
Set-AdfsProperties -ExtranetLockoutMode AdfsSmartLockoutEnforce
Restart-Service adfssrv
```

How it works...

When adding AD FS to an environment running Active Directory, the last thing you want is for AD FS to have a negative impact on the overall information security of the environment.

AD FS adds Extranet Lockout to the Active Directory (fine-grained) password and account lockout policies to prevent malicious persons from locking out accounts with incorrect password attempts from the internet. Instead, when the AD FS Extranet Lockout threshold is reached, the account cannot be used for the period of the AD FS Extranet Lockout duration to authenticate to AD FS-integrated resources. Authenticating to other resources will work without a hitch.

To prevent this scenario of malicious persons locking out accounts in Active Directory through AD FS, enable Extranet Smart Lockout. With this feature enabled, IP addresses for successful authentications by users are logged as familiar IPs in the `AccountActivity` table for the account in the AD FS configuration database. For this IP address, the regular AD FS threshold applies, and each legitimate user may still lock themselves out, as always.

A difference flow applies when authentications start to fail for the account from unfamiliar IP addresses. The failed authentication count for those IP addresses is incremented, and when the lockout threshold is reached, authentication attempts from those specific unfamiliar IP address are locked out. However, legitimate users do not experience any lockouts from their familiar IP addresses.

Microsoft recommends using stricter lockout settings for AD FS (Smart Account Lockout) than for Active Directory (fine-grained) password and account lockout policies to make sure that AD FS authentication attempts don't lock out accounts in Active Directory itself.

12
Managing Certificates

Certificates, as part of a **Public Key Infrastructure** (**PKI**), have several practical uses:

- They can be used to encrypt network traffic.
- They can be used to sign in.
- They can be used to sign code.
- They can be used to encrypt files and folders.

Certificates are based on cryptographic **public keys** and **private keys**. This key pair is generated upon the creation of a certificate by a **Certification Authority** (**CA**). When used to encrypt data, files, and folders, only with the right private key can data be decrypted that was encrypted with a given public key, and vice versa. When used to sign in, only the right key can be used to verify the identity.

Every certificate has a subject and a lifespan. A typical lifespan is one year. However, once a certificate is issued to you or a system, it can be revoked during its lifespan. Every time a certificate action is performed, the certificate revocation status is checked with the CA that issued the certificate. CAs offer a **Certificate Revocation List** (**CRL**) for this purpose.

A certificate can also be renewed. In this case, a new certificate is generated and issued.

In Microsoft-oriented networking environments, **Active Directory Certificate Services** (**AD CS**) can be used as a CA. It can be deployed to issue and revoke certificates. AD CS can function as a standalone CA but can also be configured as an Active Directory-integrated CA. This latter deployment scenario offers several benefits.

In this chapter, we cover the following recipes:

- Deciding between your own CA and a public CA
- Setting up a CA
- Setting up an online responder
- Removing a certificate template
- Duplicating and editing a certificate template
- Requesting a web server certificate
- Issuing domain controller certificates
- Managing certificate autoenrollment
- Revoking a certificate
- Decommissioning a CA

Deciding between your own CA and a public CA

When you need a certificate, you can choose between a certificate issued by your own CA implementation or a public CA.

How to do it...

To decide between a certificate issued by your own CA implementation or a public CA, use the flow chart in the following figure:

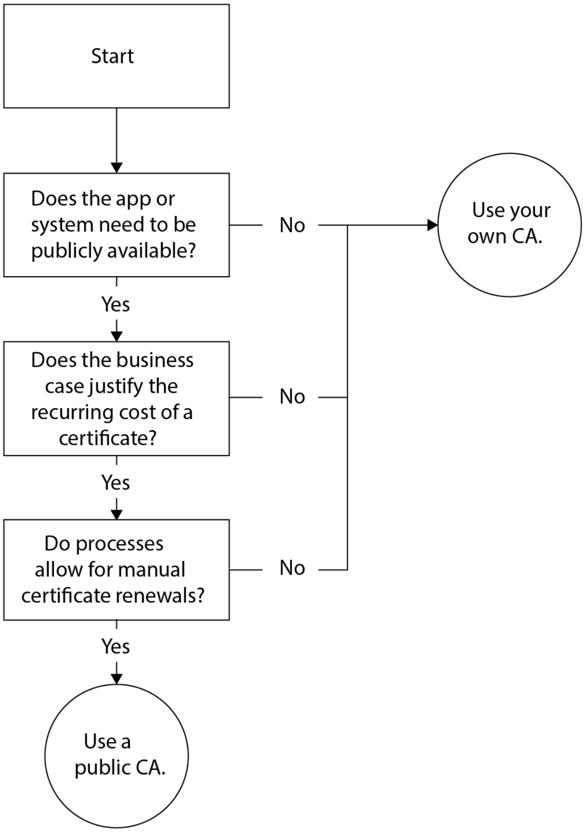

Start
Does the app or system need to be publicly available?
No
Use your own CA.
Yes
Does the business case justify the recurring cost of a certificate?
No
Yes
Do processes allow for manual certificate renewals?
No
Yes
Use a public CA.

Figure 12.1 – A flowchart to decide between your own CA and a public CA

How it works...

There are two main types of CAs:

- Public CAs
- Private CAs

Public CAs are trusted by operating systems, browsers, applications, and services. For this purpose, public CAs are part of the *list of trusted root CAs* that comes bundled with the Windows and Windows Server operating systems.

Of course, maintaining a place on the list of trusted root CAs involves rigorous security measures. Public CAs will only conduct business with legitimate organizations. As an organization, you will be asked to show proof of existence, tax registration, or incorporation.

The measures taken by public CAs cost money. Therefore, a certificate from most public CAs costs money. This is a recurring cost, as the lifespan of public certificates on the internet is configured to a maximum of 13 months.

Private CAs are not widely trusted. When you want to trust certificates issued by private CAs, you'll need to add the CA certificate to the **Trusted Root Certificate Authorities** Certificate Store for your system, browser, apps, and/or service accounts.

An AD CS implementation within your own networking infrastructure is a private CA. When it is integrated with Active Directory, your own CA can offer automatic renewals of certificates.

See also

If you want to use certificates from your own CA, check out the following recipes in this chapter:

- *Setting up a CA*
- *Setting up an online responder*

If you decide you no longer need your own CA, check the *Decommissioning a certification authority* recipe in this chapter.

There's more...

DigiCert (`https://digicert.com`) is a public CA that issues certificates that are widely trusted. Let's Encrypt (`https://letsencrypt.org`) is a public CA that also issues certificates that are trusted in most scenarios. However, Let's Encrypt issues their certificates for free, as they are an open source CA.

Setting up a CA

If you decide to issue certificates from your own CA, use this recipe to set it up using AD CS.

Getting ready

Sign in with local administrator privileges to a Windows Server installation that you intend to use as a CA.

If you intend to implement an enterprise CA, ensure the server is domain-joined and you're signed in with a domain account that is a member of the **Enterprise Admins** group. Ensure that Active Directory replication works adequately.

How to do it…

Setting up a CA consists of the following steps:

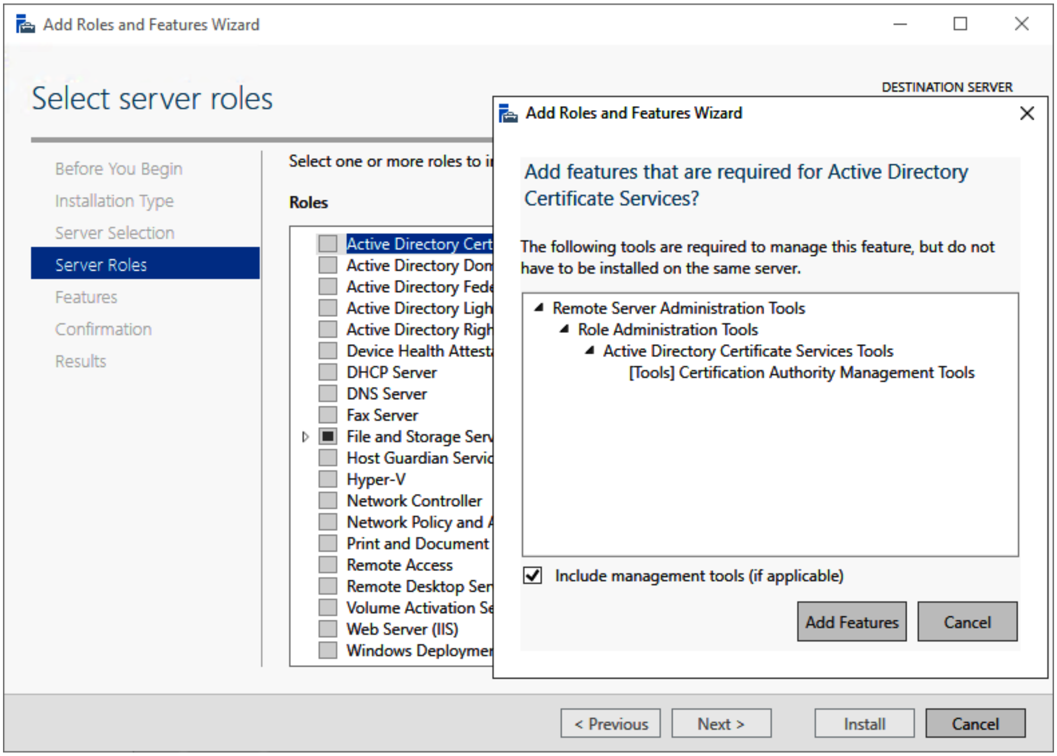
1. Installing the AD CS role
2. Configuring the CA

Installing the AD CS role

Perform these steps to install the AD CS role using the **Add Roles and Features** wizard:

1. Press Start.
2. Search for **Server Manager** or run `servermanager.exe`. The **Server Manager** window appears.
3. In the gray top bar of **Server Manager**, click **Manage**.
4. Select **Add Roles and Features** from the menu. The **Add Roles and Features Wizard** window appears.
5. On the **Before you begin** screen, click **Next** >.
6. On the **Select installation type** screen, select **Role-based or feature-based installation** and click **Next** >.
7. On the **Select destination server** screen, select the local Windows Server installation from the server pool list and click **Next** >.

8. On the **Select server roles** screen, select the **Active Directory Certificate Services** role from the list of available roles. The **Add Roles and Features Wizard** pop-up screen appears:



Add features that are required for Active Directory Certificate Services?

The following tools are required to manage this feature, but do not have to be installed on the same server.

- Remote Server Administration Tools
  - Role Administration Tools
    - Active Directory Certificate Services Tools
      - [Tools] Certification Authority Management Tools

☑ Include management tools (if applicable)

Figure 12.2 – The Add Roles and Features Wizard pop-up window

9. On the pop-up screen, click the **Add Features** button to add features that are required for AD CS. This installs **Certification Authority Management Tools** and then closes the pop-up window.

10. Click **Next** >.

11. On the **Select features** screen, click **Next** >.

12. On the **Active Directory Certificate Services** screen, click **Next** >.

13. On the **Select role services** screen, click **Next** >.

14. On the **Confirm installation selections** screen, click **Install**.

15. When configuration of the AD CS server role is done, click **Close** to close **Add Roles and Features Wizard**.

Alternatively, you can install the AD CS role using the following line of PowerShell in an elevated Windows PowerShell window:

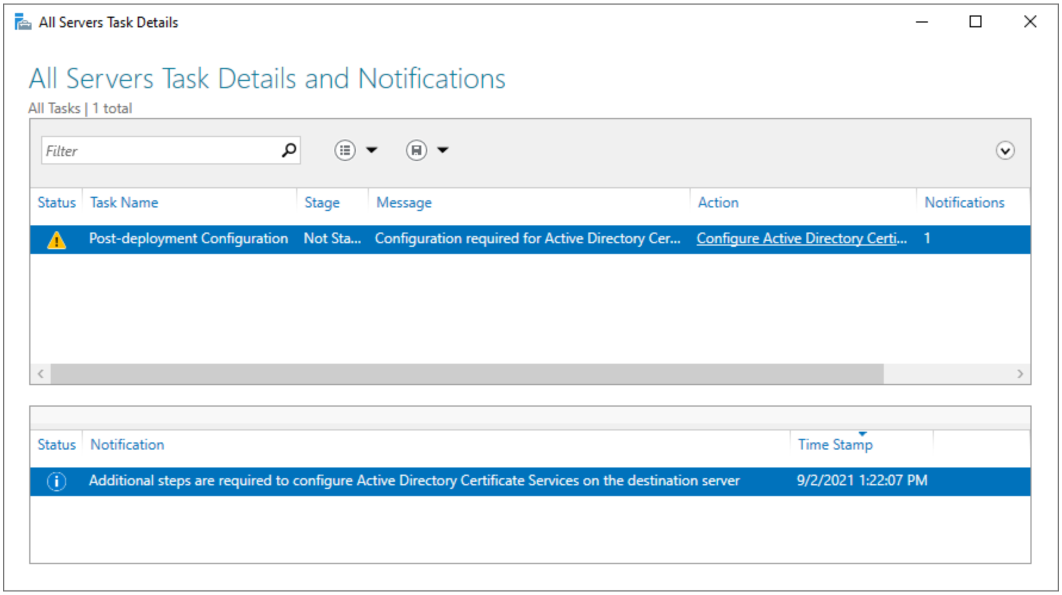
```
Install-WindowsFeature ADCS-Cert-Authority
-IncludeManagementTools
```

The preceding line of PowerShell is the preferred way to install the AD CS role on Server Core installations.

Configuring the CA

Perform these steps to configure the CA:

1. Press Start.
2. Search for **Server Manager** or run `servermanager.exe`. If the **Server Manager** window is still open, return to it.
3. In the left navigation pane, click **AD CS**.
4. Click the **More…** link in the yellow ribbon titled **Configuration required for Active Directory Certificate Services at server**. The **All Servers Task Details and Notifications** screen appears:



All Servers Task Details

All Servers Task Details and Notifications

All Tasks | 1 total

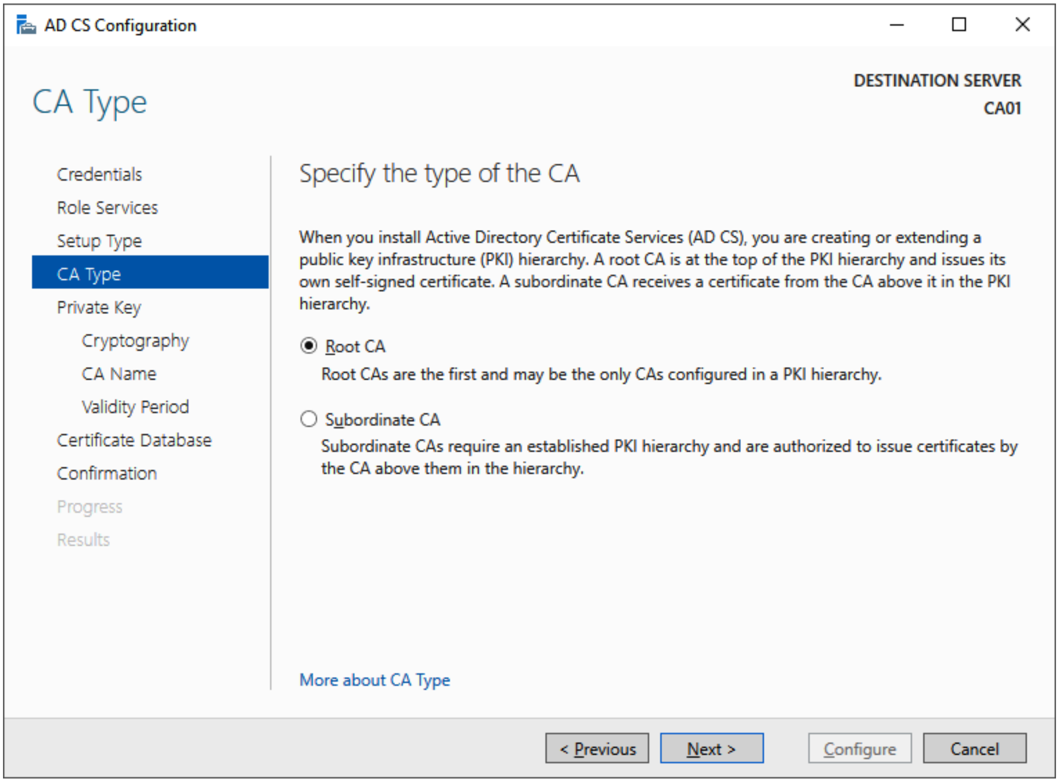
Status	Task Name	Stage	Message	Action	Notifications
⚠	Post-deployment Configuration	Not Sta...	Configuration required for Active Directory Cer...	Configure Active Directory Certi...	1

Status	Notification	Time Stamp
ⓘ	Additional steps are required to configure Active Directory Certificate Services on the destination server	9/2/2021 1:22:07 PM

Figure 12.3 – The All Servers Task Details and Notifications window

5. Click the **Configure Active Directory Certificate Services on the destination server** link in the **Action** column. The **AD CS Configuration** window appears.
6. On the **Specify credentials to configure role services** screen, click **Next** >.

   On the **Select Role Services to configure** screen, select the **Certification Authority** from the list of role services available to be configured and click **Next** >.
7. On the **Specify the setup type of the CA** screen, select the **Enterprise CA** or the **Standalone CA** option. To configure an enterprise CA, the Windows Server installation needs to be domain-joined.
8. Click **Next** >.
9. On the **Specify the type of the CA** screen, select the **Root CA** or **Subordinate CA** option. If this is the first CA you're configuring and you're not aiming for a multi-tier CA hierarchy, select the **Root CA** option:



Specify the type of the CA

When you install Active Directory Certificate Services (AD CS), you are creating or extending a public key infrastructure (PKI) hierarchy. A root CA is at the top of the PKI hierarchy and issues its own self-signed certificate. A subordinate CA receives a certificate from the CA above it in the PKI hierarchy.

● Root CA

Root CAs are the first and may be the only CAs configured in a PKI hierarchy.

○ Subordinate CA

Subordinate CAs require an established PKI hierarchy and are authorized to issue certificates by the CA above them in the hierarchy.

Figure 12.4 – Specifying the type of CA

10. Click **Next** >.
11. On the **Specify the type of private key** screen, click **Next** >.
12. On the **Specify the cryptographic option** screen, click **Next** >.
13. On the **Specify the name of the CA** screen, specify a meaningful name in the **Common name for this CA:** field. This name will be used to denote the CA for all certificates.
14. Click **Next** >.
15. On the **Specify the validity period** screen, click **Next** >.
16. On the **Specify the database locations** screen, click **Next** >.
17. On the **Confirmation** screen, click **Configure**.
18. On the **Results** screen, click **Close**.

Alternatively, you can configure the CA with default settings using the following line of PowerShell in an elevated Windows PowerShell window:

```
Install-AdcsCertificationAuthority -CAType StandaloneRootCA
```

To configure an enterprise root CA, use the following line in PowerShell:

```
Install-AdcsCertificationAuthority -CAType EnterpriseRootCA
```

The preceding lines of PowerShell are the preferred ways to configure a CA on Server Core installations.

How it works...

There are four ways you can install a CA with AD CS:

- A standalone root CA
- A standalone subordinate CA
- An enterprise root CA
- An enterprise subordinate CA

An **enterprise CA** is Active Directory-integrated. This means that AD CS stores additional information in Active Directory that allows systems and people within your Active Directory environment to request and (automatically) renew certificates, and that makes issued certificates trusted within the environment. A **standalone CA** is not integrated with Active Directory. For standalone CAs, the certificate with the public key for the root certificate needs to be manually distributed to devices and persons that use certificates issued by the CA. This certificate needs to be placed in their Trusted Root Certification Authorities Certificate Store.

A **root CA** acts as the top CA in a hierarchy of CAs. A **subordinate CA** is a CA that is below the root CA in that hierarchy.

By default, a Windows Server 2022-based CA uses an SHA-256 RSA private key from the Microsoft Software Key Storage Provider with a 2,048-bit key length. Its default validity period is 5 years.

There's more...

Once you've installed the AD CS role on a Windows Server installation, its domain membership can no longer be changed. It cannot be joined to a domain from a workgroup and it cannot be removed from an Active Directory domain.

Setting up an online responder

When a lot of certificates are revoked on a CA, certification revocation checks may become slow. In this case, you can add the **Online Responder** role service to the CA. Use this recipe to configure the Online Responder role on an existing CA.

Getting ready

Sign in with local administrator privileges to the Windows Server-based CA that you intend to add the Online Responder feature to.

How to do it...

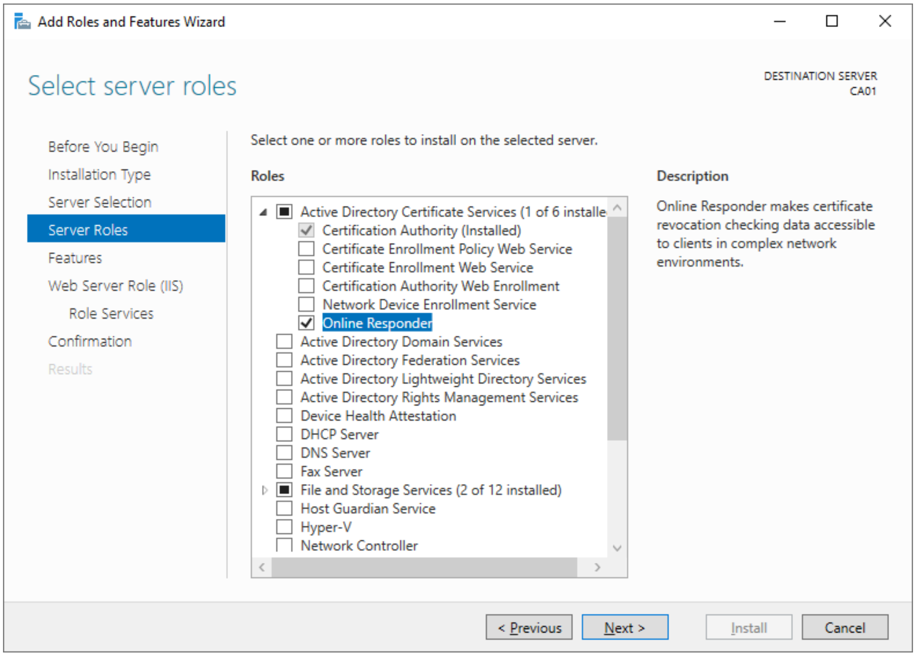
Setting up an online responder consists of the following steps:

1. Installing the Online Responder role service
2. Configuring the online responder

Installing the Online Responder role service

Perform these steps to configure an online responder for the CA:

1. Press Start.
2. Search for **Server Manager** or run `servermanager.exe`. The **Server Manager** window appears.
3. In the gray top bar of **Server Manager**, click **Manage**.
4. Select **Add Roles and Features** from the menu.
5. The **Add Roles and Features Wizard** window appears.
6. On the **Before you begin** screen, click **Next >**.
7. On the **Select installation type** screen, select **Role-based or feature-based installation** and click **Next >**.
8. On the **Select destination server** screen, select the local Windows Server installation from the server pool list and click **Next >**.
9. On the **Select server roles** screen, expand the **Active Directory Certificate Services** role.
10. Select the **Online Responder** role service:



Select server roles

DESTINATION SERVER
CA01

Select one or more roles to install on the selected server.

Roles

■ Active Directory Certificate Services (1 of 6 installe
☑ Certification Authority (Installed)
□ Certificate Enrollment Policy Web Service
□ Certificate Enrollment Web Service
□ Certification Authority Web Enrollment
□ Network Device Enrollment Service
☑ Online Responder
□ Active Directory Domain Services
□ Active Directory Federation Services
□ Active Directory Lightweight Directory Services
□ Active Directory Rights Management Services
□ Device Health Attestation
□ DHCP Server
□ DNS Server
□ Fax Server
■ File and Storage Services (2 of 12 installed)
□ Host Guardian Service
□ Hyper-V
□ Network Controller

Description

Online Responder makes certificate revocation checking data accessible to clients in complex network environments.

Figure 12.5 – The Online Responder role service selected on the Select server roles screen

The **Add Roles and Features Wizard** pop-up screen appears.

11. In the pop-up screen, click the **Add Features** button to add features that are required for the online responder. This installs the **Online Responder Tools**, the **IIS Management Console**, and default Web Server features, and then closes the pop-up window.
12. On the **Select features** screen, click **Next** >.
13. On the **Web Server Role (IIS)** screen, click **Next** >.
14. On the **Confirm installation selections** screen, click **Install**.
15. On the **Results** screen, click **Close**.

Alternatively, you can install the Online Responder role service using the following line of PowerShell in an elevated Windows PowerShell window:

```
Add-WindowsFeature Adcs-Online-Cert
```

The preceding line of PowerShell is the preferred way to install the Online Responder role service on Server Core installations.

Configuring the online responder

With the role service installed, we can configure the online responder. Perform these steps:

1. Press Start.
2. Search for **Server Manager** or run `servermanager.exe`. If the **Server Manager** window is still open, return to it.
3. In the left navigation pane, click **AD CS**.
4. Click the **More…** link in the yellow ribbon titled **Configuration required for Active Directory Certificate Services at server**. The **All Servers Task Details and Notifications** screen appears.
5. Click the **Configure Active Directory Certificate Services on the destination server** link in the **Action** column. The **AD CS Configuration** window appears.
6. On the **Specify credentials to configure role services** screen, click **Next** >.
7. On the **Select Role Services to configure** screen, select **Online Responder** in the list of role services available to be configured and click **Next** >.
8. On the **Confirmation** screen, click **Configure**.
9. On the **Results** screen, click **Close**.

Alternatively, you can configure the Online Responder role service using the following line of PowerShell in an elevated Windows PowerShell window:

```
Install-AdcsOnlineResponder
```

The preceding line of PowerShell is the preferred way to configure an online responder on Server Core installations.

How it works...

When a certificate is used, the CA is contacted to check whether the certificate was revoked during its lifetime. When a certificate is revoked, it means the certificate should no longer be used and should be considered compromised. Each certificate carries the location of the revocation information in its CRL distribution point.

By default, the CRL is shared through CRL distribution points via the `CertEnroll` file share and the `CertEnroll` IIS virtual folder.

However, the CRL can achieve quite an extensive length over the years. As the CRL is downloaded in its entirety from the distribution point every time a certificate is used, this can strain the CA and lead to slow CRL checks.

An online responder can be configured to remove the need to download the CRL in its entirety. Instead, when the revocation information is needed, the online responder checks its cached copy of the CRL and answers using the **Online Certificate Status Protocol** (**OCSP**), described in RFC 6960.

See also

OCSP is explained in more detail in RFC 6960: `https://datatracker.ietf.org/doc/html/rfc6960`.

Removing a certificate template

On enterprise CAs, the Active Directory integration offers **certificate template** functionality. These templates define the certificates that are issued by the CA. Use this recipe to remove a certificate template that was issues by an enterprise CA.

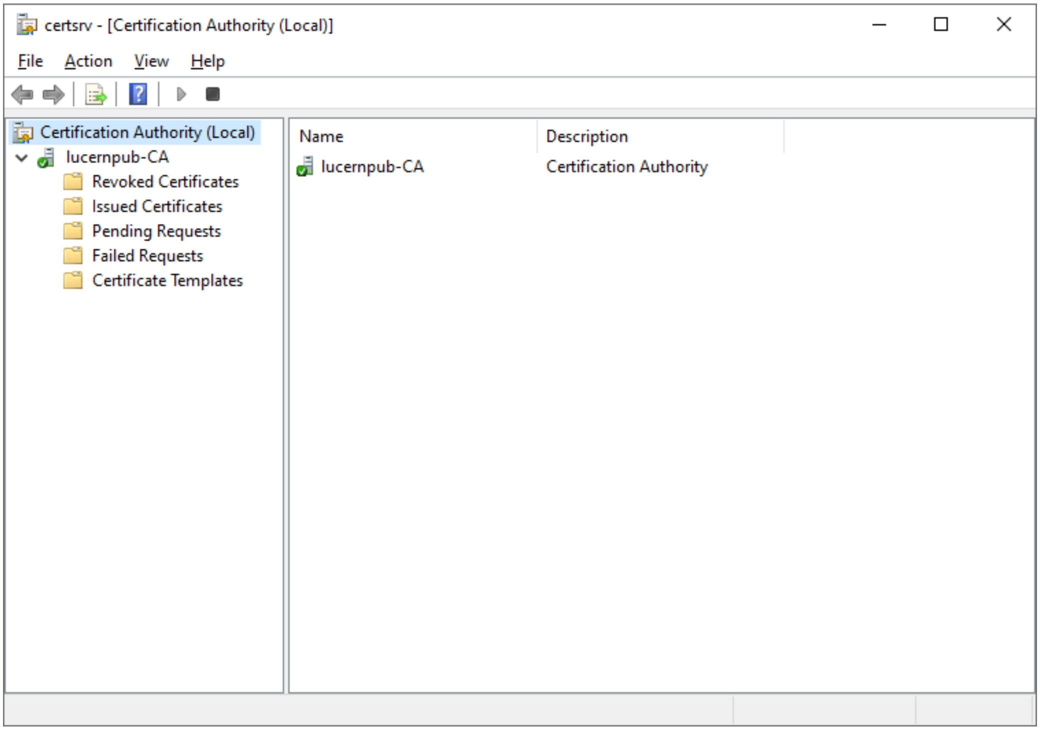
Getting ready

Sign in to the enterprise CA with a domain account that is a member of the **Enterprise Admins** group.

How to do it...

Perform the following steps to manage certificate templates:

1. Press Start.
2. Search for the **Certification Authority** management console or run `certsrv.msc`.
3. The **Certification Authority** window appears:



Name	Description
lucernpub-CA	Certification Authority

Figure 12.6 – The Certification Authority management console

4. In the left navigation pane, expand the node representing the CA.
5. Click the **Certificate Templates** node. The main pane lists the certificate templates that this CA issues certificates for. Usual certificate templates that are available by default include **User**, **Computer**, and **Web Server** templates.
6. Right-click a certificate template here and select **Delete** from the context menu. This will not delete the certificate template or any certificates issued based on the template.
7. Close the **Certification Authority** management console.

Alternatively, you can use Windows PowerShell on a Windows Server installation configured as a CA to remove a certificate template:

```
Remove-CATemplate -Name 'Name of no longer needed certificate'
```

Using PowerShell, you can automate the removal of certificate templates or remove certificate templates using filters.

How it works…

Certificate templates act as templates for certificates. Certificate templates are stored in Active Directory.

The **Certificate Templates** node in the **Certification Authority** management console provides a list of the certificate templates that can be requested from the CA and, thus, can be issued from the CA.

You might want to remove a certificate template for a CA when you don't want a particular CA issuing that type of certificate. Another CA can then be used to issue that type of certificate. This is useful when you want to start issuing user certificates from a new CA with an optimized CRL distribution point or when you want dedicated CAs for certain types of certificates.

Duplicating and editing a certificate template

Certificate templates can be edited. However, it is a bad idea to edit the built-in certificate templates. Use this recipe to duplicate and edit a certificate template to be issued from an enterprise CA.

Getting ready

Sign in to the enterprise CA with a domain account that is a member of the **Enterprise Admins** group.

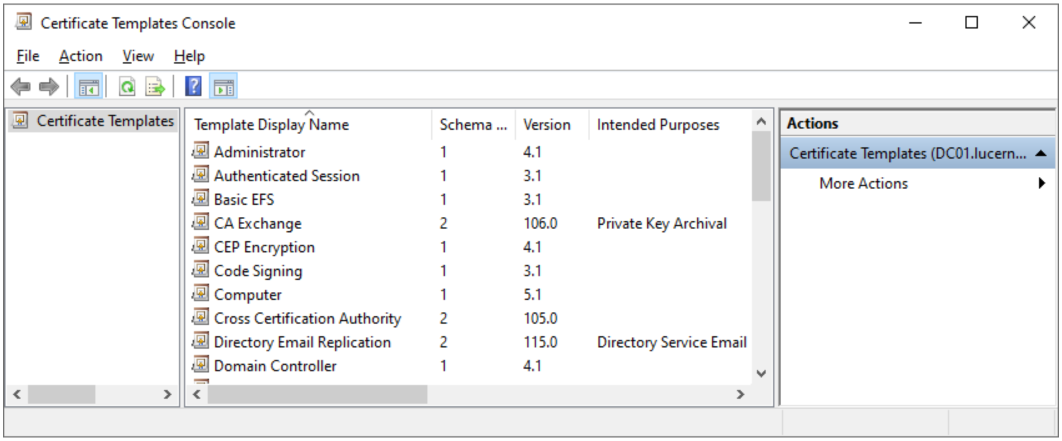
How to do it…

Perform the following steps to manage certificate templates:

1. Press Start.
2. Search for the **Certification Authority** management console or run `certsrv.msc`.

3. In the left navigation pane, expand the node representing the CA. Right-click the **Certificate Templates** node and select **Manage** from the context menu to manage certificate templates. Alternatively, run `certtmpl.msc`.

   The **Certificate Templates Console** (`certtmpl.msc`) window appears:


Certificate Templates Console
File
Action
View
Help
Certificate Templates
Actions
Certificate Templates (DC01.lucern...
More Actions

Template Display Name	Schema ...	Version	Intended Purposes
Administrator	1	4.1	
Authenticated Session	1	3.1	
Basic EFS	1	3.1	
CA Exchange	2	106.0	Private Key Archival
CEP Encryption	1	4.1	
Code Signing	1	3.1	
Computer	1	5.1	
Cross Certification Authority	2	105.0	
Directory Email Replication	2	115.0	Directory Service Email
Domain Controller	1	4.1	

Figure 12.7 – The Certificate Templates Console window

4. If you want to change the audience for a certificate template, right-click the certificate template in the main pane and select **Properties**. The **Security** tab of the certificate's properties allows you to change the permissions for users and computers in Active Directory. You can allow or deny **Full Control**, **Read**, **Write**, **Enroll**, and/or **Autoenroll** permissions to the certificate template.
5. If you want to change other settings for a certificate template, right-click a certificate template in the main pane and select **Duplicate Template** from the context menu to create a copy of a template and change the properties of the duplicate template.
6. Close the **Certificate Templates Console** window when done.

When you've duplicated a certificate template, you'll need to instruct at least one CA to issue the certificate template. Perform these steps to do so:

1. Switch to the **Certification Authority** management console, or open it on the CA from which you want to issue certificates using the newly created certificate template.
2. In the left navigation pane, expand the node representing the CA.
3. Right-click the **Certificate Templates** node and select **New**, and then **Certificate Template to Issue** from the context menu. The **Enable Certificate Template** window appears.
4. Select the newly created certificate template from the list and click **OK**.

Alternatively, you can use Windows Powershell on a Windows Server installation configured as a CA to add the certificate template to a CA:

```
Add-CATemplate -Name 'Name of certificate template to issue'
```

Using PowerShell, you can automate adding certificate templates or add multiple certificate templates at once.

How it works...

Certificate templates act as templates for certificates. Certificate templates are stored in Active Directory.

The **Certificate Templates Console** provides a way to manage certificate templates in Active Directory. Changes to certificate templates are picked up by all enterprise CAs.

As software may rely on requesting certain types of certificates with certain settings, it is recommended to not alter the built-in certificate templates. Instead, duplicating a certificate template to meet your needs and naming the certificate template appropriately is the way to go.

Deny permissions overrule allow permissions. Denying permissions in certificate templates is not recommended, as it adds complexity and makes issuing and managing certificates harder.

Requesting a web server certificate

One of the most common certificate uses is to encrypt web traffic. A **web server certificate** can be used for this purpose. This recipe shows how to request a web server certificate from an enterprise CA from a domain-joined Windows server.

Getting ready

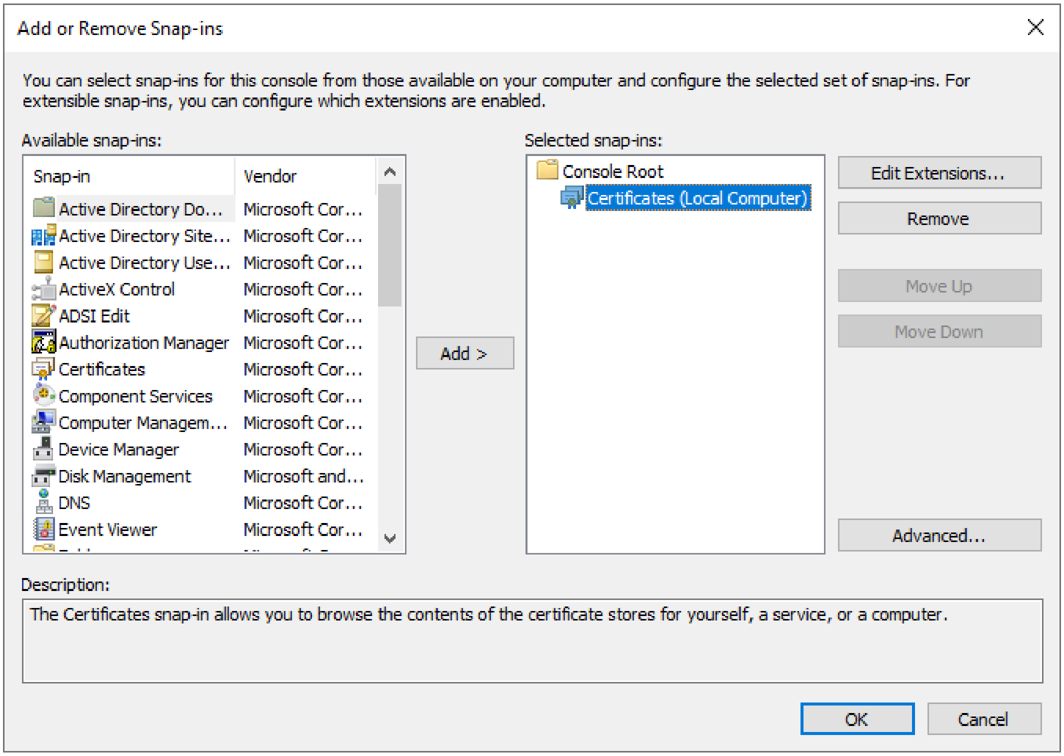
Sign in to the web server with a domain account.

How to do it...

Perform these steps to request a web server certificate from an enterprise CA:

1. Press Start.
2. Type `mmc.exe` and press the *Enter* key.
3. This starts an empty Microsoft Management Console window titled **Console1**.

4. From the **File** menu, click **Add/Remove Snap-in…**. The **Add or Remove Snap-ins** pop-up window appears.
5. Select **Certificates** from the **Available snap-ins** list and click **Add >**.
6. Select **Computer account** and click **Next >**.
7. Select **Local Computer** and click **Finish**. The **Certificates (Local Computer)** snap-in is now visible in the **Selected snap-ins** column:



Add or Remove Snap-ins

You can select snap-ins for this console from those available on your computer and configure the selected set of snap-ins. For extensible snap-ins, you can configure which extensions are enabled.

Available snap-ins:

Snap-in	Vendor
Active Directory Do…	Microsoft Cor…
Active Directory Site…	Microsoft Cor…
Active Directory Use…	Microsoft Cor…
ActiveX Control	Microsoft Cor…
ADSI Edit	Microsoft Cor…
Authorization Manager	Microsoft Cor…
Certificates	Microsoft Cor…
Component Services	Microsoft Cor…
Computer Managem…	Microsoft Cor…
Device Manager	Microsoft Cor…
Disk Management	Microsoft and…
DNS	Microsoft Cor…
Event Viewer	Microsoft Cor…

Selected snap-ins:

- Console Root
  - Certificates (Local Computer)

Description:

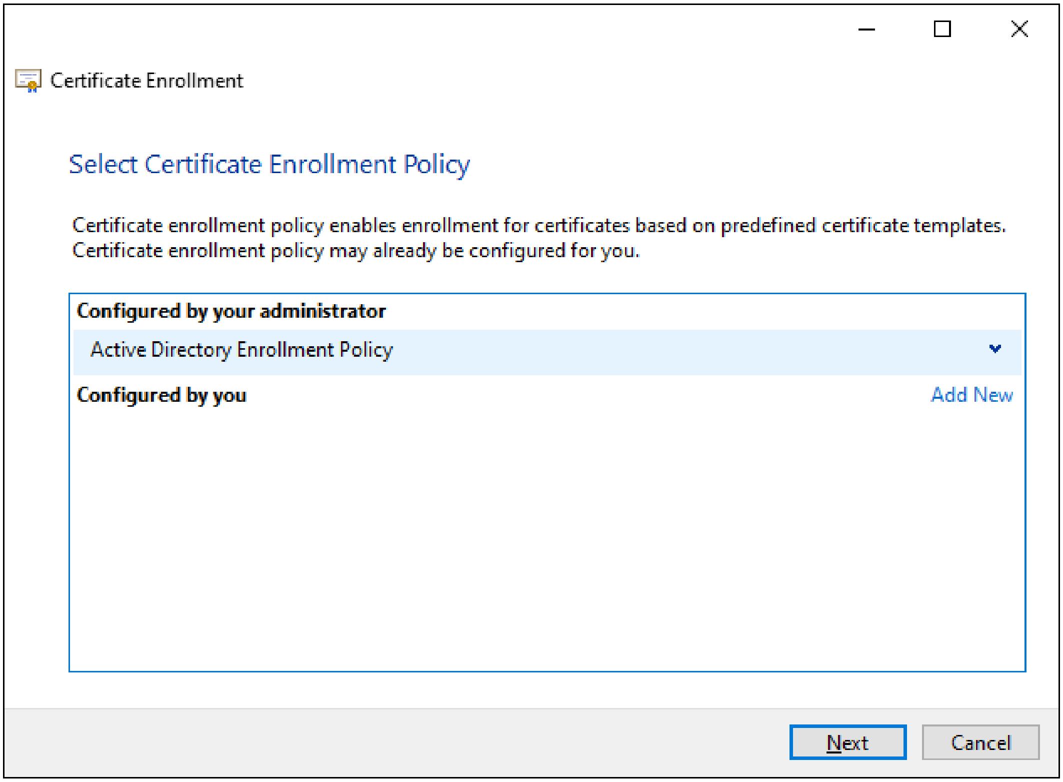
The Certificates snap-in allows you to browse the contents of the certificate stores for yourself, a service, or a computer.

Figure 12.8 – Adding or removing snap-ins

8. Click **OK**. This closes the **Add or Remove Snap-ins** pop-up window.
9. In the left pane of the **Console1** window, expand **Certificates (Local Computer)**, then **Personal**, and then **Certificates**.
10. In the **Action** pane to the right, click **More Actions**, select **All Tasks** from the context menu, and then click **Request New Certificate…**.

The **Certificate Enrollment** window appears.

11. On the **Before you begin** screen, click **Next**.
12. On the **Select Certificate Enrollment Policy** screen, ensure that **Active Directory Enrollment Policy** is selected:



Certificate Enrollment

Select Certificate Enrollment Policy

Certificate enrollment policy enables enrollment for certificates based on predefined certificate templates.
Certificate enrollment policy may already be configured for you.

Configured by your administrator

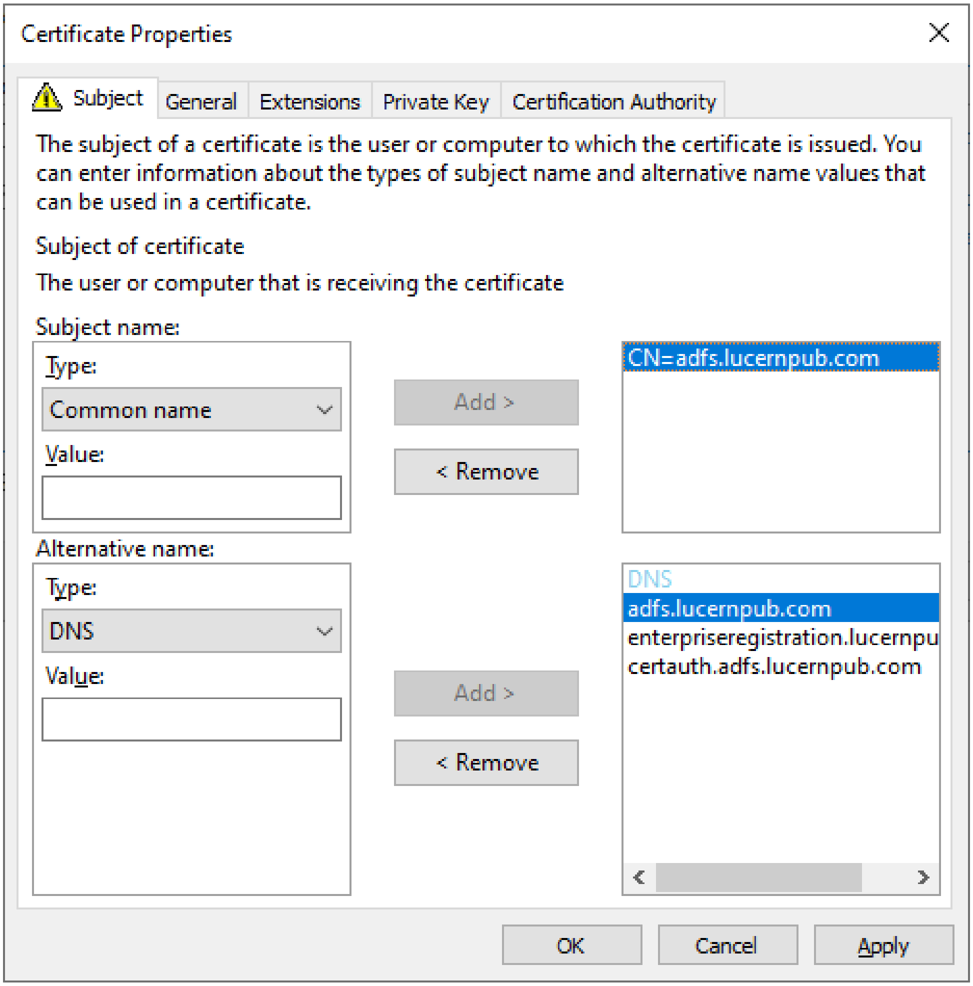
Active Directory Enrollment Policy

Configured by you

Figure 12.9 – The Select Certificate Enrollment Policy window for certificate enrollment

13. Click **Next**.
14. On the **Request certificates** screen, select the **Web Server** certificate. Click the **More information is required to enroll for this certificate. Click here to configure settings.** link underneath the web server certificate.
15. The **Certificate Properties** screen appears.

16. On the **Subject** tab, add the way or ways the web server is going to be addressed. Specify the fully qualified DNS name as the **Common Name** type in the **Subject Name:** area and as the **DNS** name in the **Alternative name:** area. Add more DNS names for the web server, too:



Certificate Properties

Subject | General | Extensions | Private Key | Certification Authority

The subject of a certificate is the user or computer to which the certificate is issued. You can enter information about the types of subject name and alternative name values that can be used in a certificate.

Subject of certificate

The user or computer that is receiving the certificate

Subject name:

Type: Common name

Value:

CN=adfs.lucernpub.com

Alternative name:

Type: DNS

Value:

DNS
adfs.lucernpub.com
enterpriseregistration.lucernpu
certauth.adfs.lucernpub.com

Figure 12.10 – Certificate Properties when requesting the default web server certificate

17. Click **OK** when done. This closes the **Certificate Properties** window.
18. On the **Request certificates** screen, click **Enroll**.
19. On the **Certificate Installation Results** screen, click **Finish**.

How it works...

Web servers typically use a certificate that is stored in the local computer's personal certificate store.

The **web server certificate template** is the quintessential certificate to encrypt TLS traffic between web servers and web browsers. For this purpose, its *key usage* is set to **digital signature** and **key encipherment**. Its application policy is configured as **server authentication**. These settings allow the web server to verify its identity and encrypt traffic.

The web server certificate template is not readily available from an enterprise CA. This prevents any domain-joined device from requesting a certificate and acting as a web server. To allow domain-joined devices to request the web server certificate, allow the **Enroll** permission on the web server certificate template to the computer account(s) for the web server or a specific group containing web servers in Active Directory.

See also

See also the *Duplicating and editing a certificate template* recipe in this chapter.

Issuing domain controller certificates

Domain controllers use certificates. However, the certificate may not fulfill all of the requirements set out for it. This recipe shows how to issue the right certificates to domain controllers.

Getting ready

To issue **Kerberos Authentication** certificates to domain controllers, the CA needs to run Windows Server 2003 or a newer version.

Sign in to the enterprise CA with a domain account that is a member of the **Enterprise Admins** group.

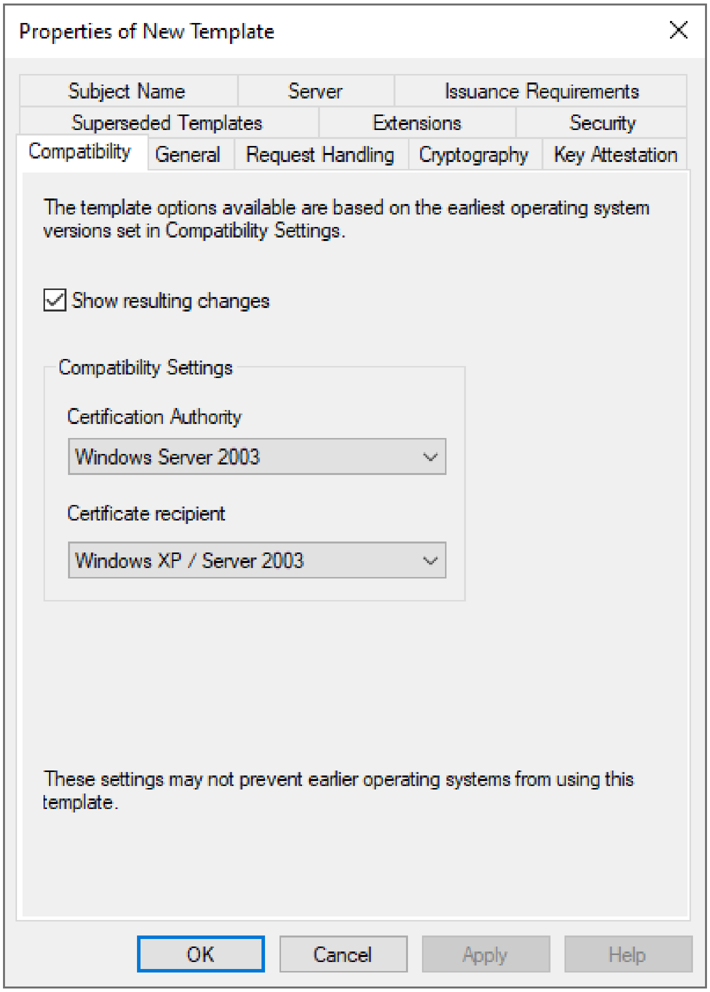
To issue the necessary certificates for Windows Hello for Business, all domain controllers that request the new certificate template need to run Windows Server 2016. The CA needs to run at least Windows Server 2008 R2 in this scenario.

How to do it...

Before enabling the **certificate autoenrollment policy** through Group Policy, configure the Kerberos Authentication certificate template to supersede the domain controller and domain controller authentication certificate templates.

Perform these steps to do so:

1. Press Start.
2. Search for the **Certification Authority** management console or run `certsrv.msc`. The **Certificate Authority** window appears.
3. In the left navigation pane, expand the node representing the CA. Right-click the **Certificate Templates** node and select **Manage** from the context menu to manage certificate templates.
4. The **Certificate Templates Console** (`certtmpl.msc`) window appears.
5. In the main pane, select the **Kerberos Authentication** certificate template.
6. Right-click the certificate template and select **Duplicate Template** from the context menu. The **Properties of New Template** window appears:



Properties of New Template

The template options available are based on the earliest operating system versions set in Compatibility Settings.

☑ Show resulting changes

Compatibility Settings

Certification Authority

Windows Server 2003

Certificate recipient

Windows XP / Server 2003

These settings may not prevent earlier operating systems from using this template.

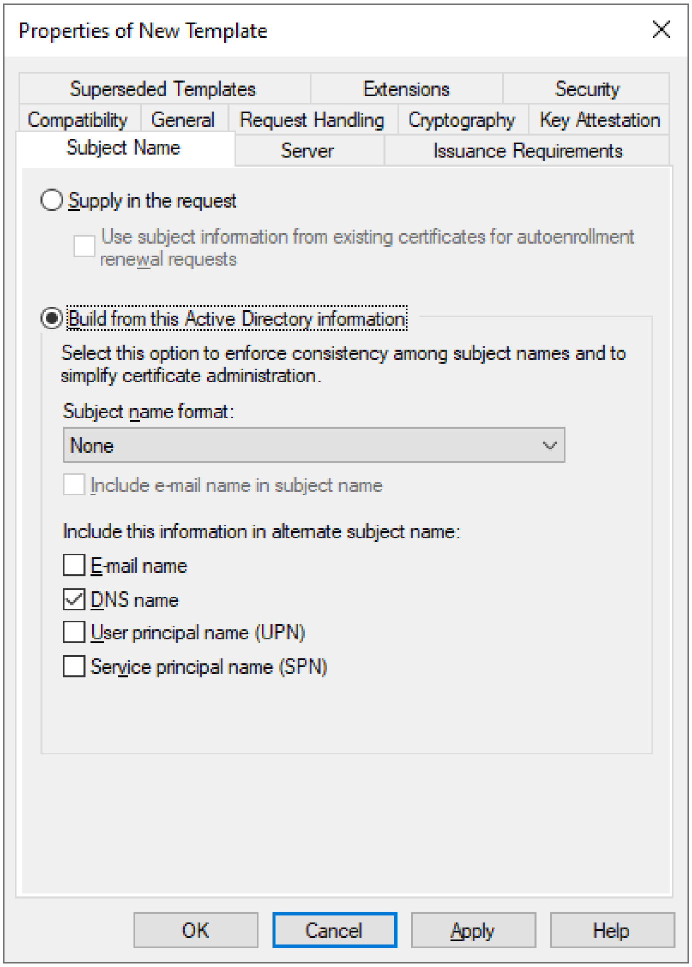
Figure 12.11 – The Properties of New Template window

7. On the **Compatibility** tab, make the following changes in the **Compatibility Settings** area:

    I. Change the value for **Certification Authority** to at least **Windows Server 2008 R2**, specifying the earliest Windows Server version acting as the CA that will issue this certificate template. The **Resulting changes** pop-up window appears. Click **OK** to dismiss it.

    II. Change the value for **Certification recipient** to at least **Windows 7 / Windows Server 2008 R2**, specifying the earliest Windows Server version acting as the CA that will issue this certificate template. The **Resulting changes** pop-up window appears. Click **OK** to dismiss it.

8. Navigate to the **Subject Name** tab.
9. Ensure the settings for the certificate template's subject name are configured as shown in the following screenshot:



○ Supply in the request

□ Use subject information from existing certificates for autoenrollment renewal requests

● Build from this Active Directory information

Select this option to enforce consistency among subject names and to simplify certificate administration.

Subject name format: None

□ Include e-mail name in subject name

Include this information in alternate subject name:

□ E-mail name

☑ DNS name

□ User principal name (UPN)

□ Service principal name (SPN)

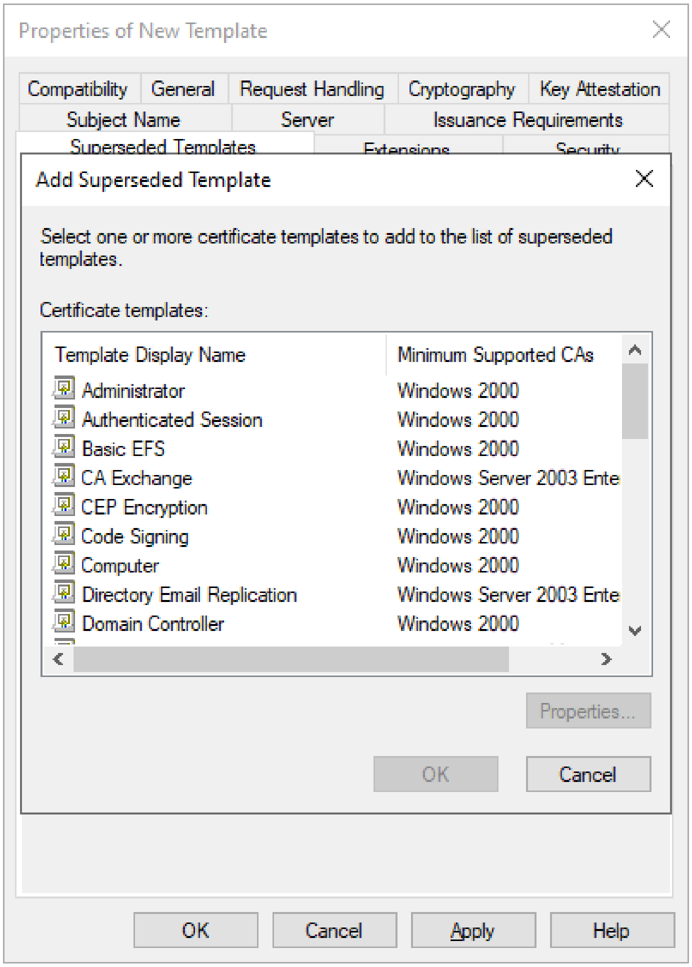
Figure 12.12 – The Subject Name tab for the new certificate template

10. Navigate to the **Cryptography** tab.
11. On the **Cryptography** tab, make the following changes:

    I. Change **Provider Category** to **Key Storage Provider**. This should change **Algorithm name** to **RSA** and the **Minimum key size** value to `2048`.

    II. Change the value for the request hash to **SHA256**.

12. Navigate to the **Superseded Templates** tab.
13. Click the **Add…** button. The **Add Superseded Template** pop-up window appears:



Select one or more certificate templates to add to the list of superseded templates.

Certificate templates:

Template Display Name	Minimum Supported CAs
Administrator	Windows 2000
Authenticated Session	Windows 2000
Basic EFS	Windows 2000
CA Exchange	Windows Server 2003 Ente
CEP Encryption	Windows 2000
Code Signing	Windows 2000
Computer	Windows 2000
Directory Email Replication	Windows Server 2003 Ente
Domain Controller	Windows 2000

Figure 12.13 – The Add Superseded Template pop-up window

14. Select the following certificate templates using the *Shift* button to select multiple templates:

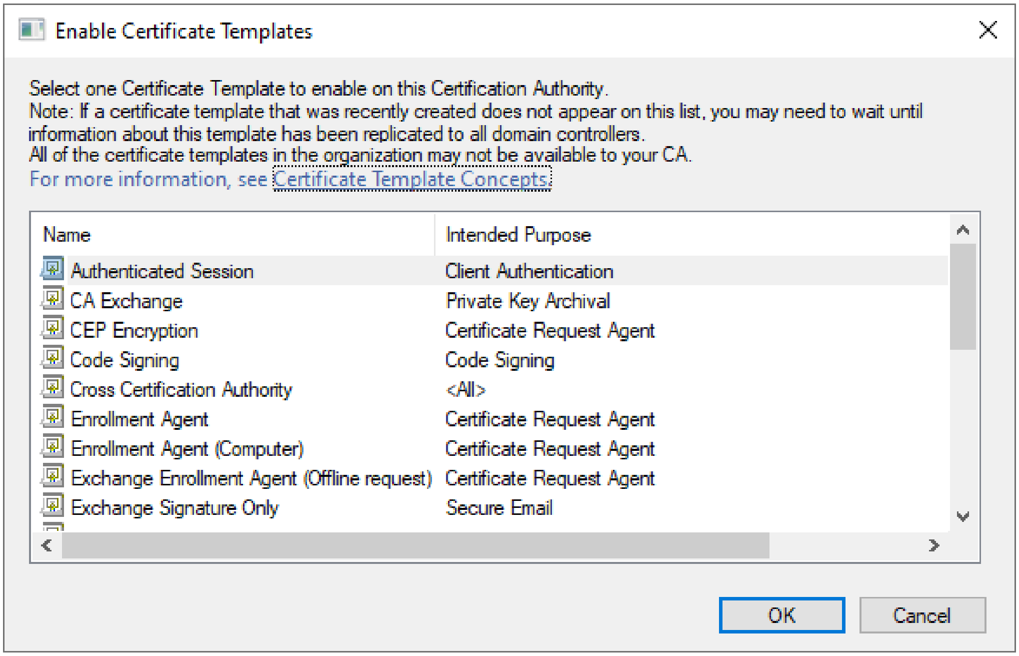
   I. **Directory Email Replication**

   II. **Domain Controller**

   III. **Domain Controller Authentication**

   IV. **Kerberos Authentication**

15. Click **OK**.
16. Navigate to the **General** tab.
17. Provide a meaningful name for the certificate template in the **Template display name:** and **Template name:** fields.
18. Click **OK** to save the new template. The **Properties of New Template** window closes.
19. In the **Certification Authority** management console, select the **Certificate Templates** node in the left navigation menu.
20. From the **Action** menu, select **New**, and then click **Certificate Template to Issue**. The **Enable Certificate Templates** window appears:



Select one Certificate Template to enable on this Certification Authority.
Note: If a certificate template that was recently created does not appear on this list, you may need to wait until information about this template has been replicated to all domain controllers.
All of the certificate templates in the organization may not be available to your CA.
For more information, see Certificate Template Concepts.

Name	Intended Purpose
Authenticated Session	Client Authentication
CA Exchange	Private Key Archival
CEP Encryption	Certificate Request Agent
Code Signing	Code Signing
Cross Certification Authority	<All>
Enrollment Agent	Certificate Request Agent
Enrollment Agent (Computer)	Certificate Request Agent
Exchange Enrollment Agent (Offline request)	Certificate Request Agent
Exchange Signature Only	Secure Email

Figure 12.14 – The Enable Certificate Templates window

21. Select the newly created certificate template from the list of available certificate templates and click **OK**.

With these steps completed, disable the superseded certificate templates using the steps in the *Removing a certificate template* recipe.

With these steps completed, perform the steps for *configuring the group policy for autoenrollment* in the next recipe.

Any previously enrolled certificate based on the superseded certificate templates will be removed and replaced by a certificate based on the new certificate template.

How it works...

Domain controllers use certificates for several purposes:

- To verify their identities as domain controllers for the Active Directory domain
- To provide **smart card authentication**
- To encrypt traffic when acting as a host offering the secure **Lightweight Directory Access Protocol** (**LDAPS**)

Optionally, they can use their certificates for IPsec communications.

All certificates used by domain controllers are stored in the local computer's personal certificate store.

Throughout the history of Active Directory, several superseding certificate configurations have been issued to domain controllers. The **domain controller certificate template**, initially requested by Windows 2000 Server-based domain controllers and issued by Windows 2000 Server-based CAs, was superseded by the **domain controller authentication certificate template**. Since Windows Server 2008, the **Kerberos Authentication certificate template** is recommended to issue to domain controllers.

The domain controller certificate template is a v1 template. It cannot be modified. The domain controller authentication certificate template is a v2 template. It can be modified but does not support the new Microsoft **Cryptographic API** (**CAPI**) with the latest encryption and hashing algorithms. The Kerberos Authentication certificate is a v3 template. Unlike the v2 template, v3 templates and beyond can use the latest cryptographic abilities.

Since Windows Server 2012, v4 templates allow the option to renew with the same key on the **Request Handling** tab of the **Properties** for a certificate template.

LDAP and LDAPS are typically used by non-domain-joined devices and services. LDAPS provides encryption based on TLS, whereas LDAP doesn't provide encryption of the traffic exchanged with the domain controller.

For offering LDAPS, by default, a domain controller uses a self-signed certificate with a validity period of one year.

To provide a valid certificate for this purpose, a proper certificate should be enrolled. The Kerberos Authentication certificate template provides the necessary certificate for this purpose, too.

The Kerberos Authentication certificate template is recommended, as it includes both the Active Directory domain name and the domain controller's fully qualified domain name as its subject and, by default, supports the following purposes:

- Server authentication
- Client authentication
- Smart card logon
- Key Distribution System (KDS) authentication

For Windows Hello for Business, a new feature in Windows 10, the built-in Kerberos Authentication certificate template needs to be updated to comply with the certificate template settings outlined in Microsoft Docs.

The certificate templates that are superseded by the new certificate template are hard-coded for a domain controller to autoenroll. The enrollment for these certificates occurs, despite the lack of an autoenrollment policy. However, to have new certificate templates autoenroll, an autoenrollment policy needs to be created using Group Policy.

Managing certificate autoenrollment

Certificate autoenrollment allows admins to configure users and computers to automatically enroll and renew certificates. This recipe shows how to configure certificate autoenrollment.

Getting ready

Certificate autoenrollment is a feature of enterprise CAs. It cannot be configured on a standalone CA.

For certificate autoenrollment to work, the CA that issues the certificate needs to run Windows Server 2003 or a newer version. Active Directory needs to run the Windows Server 2003 schema or a newer version.

To create a **Group Policy Object** (**GPO**) and manage its settings, sign in to a system with the Group Policy Management Console installed with an account that is a member of the **Domain Admins** group.

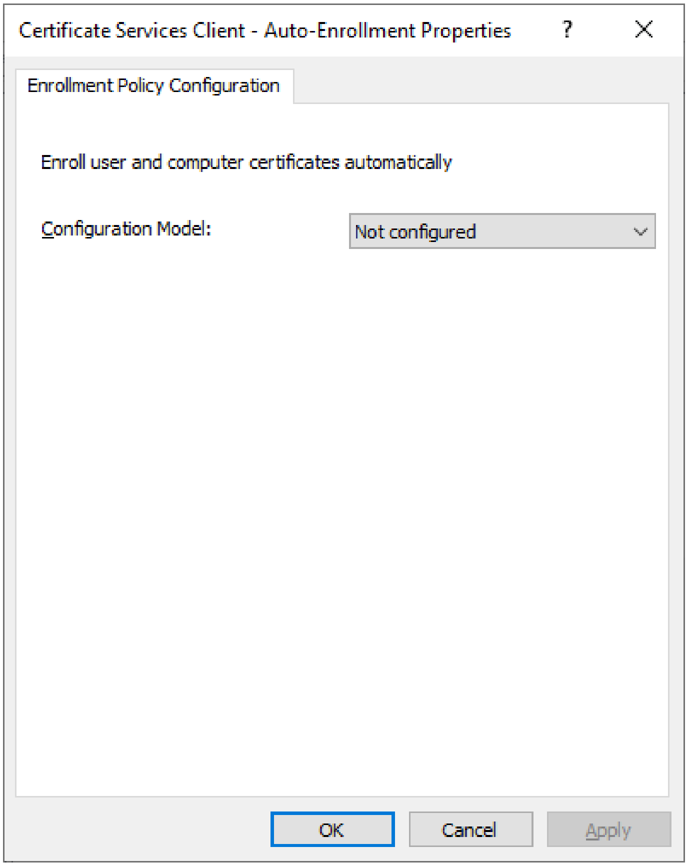
To configure certificates for autoenrollment, sign in to an enterprise CA with an account that is a member of the **Enterprise Admins** group.

How to do it...

Certificate autoenrollment is based on Group Policy. Perform these steps to configure Group Policy settings for autoenrollment:

1. Press Start.
2. Search for **Group Policy Management** or run `gpmc.msc`. The **Group Policy Management** window appears.
3. In the left navigation pane, expand the **Forest** node.
4. Expand the **Domains** node, and then navigate to the domain where you want to create the GPO.
5. Expand the domain name and select the **Group Policy Objects** node.
6. Right-click the **Group Policy Objects** node and select **New** from the context menu. The **New GPO** pop-up window appears.
7. Enter the name for the GPO.
8. Ensure that you don't select **Starter GPOs**.
9. Click **OK** to create the GPO.
10. Locate the newly created GPO in the main pane.
11. Right-click the GPO and select **Edit** from the context menu. The **Group Policy Management Editor** window appears.
12. Expand either the **Computer Configuration** or the **User Configuration** node, depending on whether you want to configure autoenrollment for user accounts or devices.
13. Expand the **Policies** node.
14. Expand the **Windows Settings** node.
15. Expand the **Security Settings** node.
16. Select the **Public Key Policies** node.

17. In the main pane, right-click the **Certificate Services client – Auto-Enrollment** setting and select **Properties** from the context menu. The **Certificate Services Client – Auto-Enrollment Properties** window appears:



Certificate Services Client - Auto-Enrollment Properties

Enrollment Policy Configuration

Enroll user and computer certificates automatically

Configuration Model: Not configured

Figure 12.15 – The Certificate Services Client – Auto-Enrollment Properties window

18. Change the **Configuration Model:** setting to **Enabled**.

19. Select the **Renew expired certificates, update pending certificates, and remove revoked certificates** option.

20. Select the **Update certificates that use certificate templates** option.

21. Click **OK** to save the Group Policy setting and close the **Certificate Services Client – Auto-Enrollment Properties** window.

22. Close the **Group Policy Management Editor** window.

23. In the **Group Policy Management** window, in the left navigation pane, select the **Domain Controllers** OU.
24. Right-click the **Domain Controllers** OU and select **Link an Existing GPO…** from the context menu. The **Select GPO** pop-up window appears.
25. Select the newly created GPO from the **Group Policy objects:** list and click **OK**.

How it works...

When the Group Policy setting is enabled within the **Computer Configuration** part of a GPO, a computer will automatically enroll certificates that are configured for **autoenrollment** when the following occurs:

- The device reboots.
- A Group Policy refreshes in the background.
- The `certutil.exe -pulse` command is performed.

When the Group Policy setting is enabled within the **User Configuration** part of a GPO, the device will automatically enroll a user certificate for the user account when the following occurs:

- The user signs in interactively with the account.
- A Group Policy refreshes in the background.
- The `certutil.exe -pulse` command is performed.

See also

To extend the schema to meet the requirements for certificate autoenrollment, refer to the *Extending the schema* recipe in *Chapter 1*, *Optimizing Forests, Domains and Trusts*.

To change the settings for a certificate template to allow autoenrollment, refer to the **Security** tab in the *Duplicating and editing a certificate template* recipe.

Revoking a certificate

Every certificate issued by a CA is stored in its database. From there, it can be reissued and revoked. This recipe shows how to revoke a previously issued certificate.

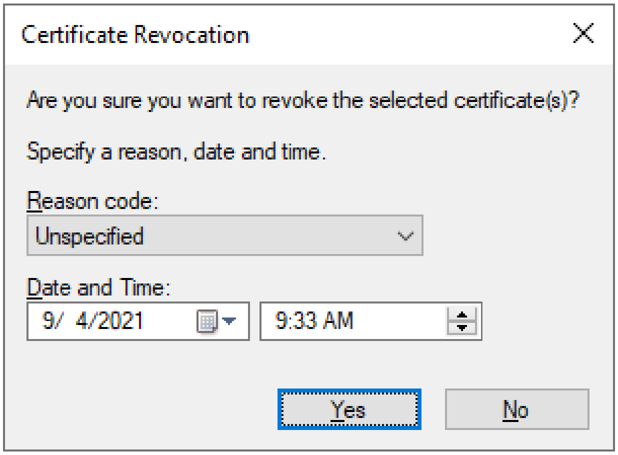
Getting ready

Sign in with local administrator privileges when the CA on which you want to revoke one or more certificates is a standalone CA. Sign in with a domain account that is a member of the **Enterprise Admins** group when the CA on which you want to revoke one or more certificates is an enterprise CA.

How to do it...

Perform these steps to revoke a previously issued certificate:

1. Press Start.
2. Search for the **Certification Authority** management console or run `certsrv.msc`.
3. In the left navigation pane, expand the node representing the CA.
4. Click the **Issued Certificates** node.
5. In the main pane, select the certificate you want to revoke.
6. Right-click the certificate. From the context menu, select **All Tasks** and then **Revoke Certificate**. The **Certificate Revocation** pop-up window appears:



Certificate Revocation

Are you sure you want to revoke the selected certificate(s)?

Specify a reason, date and time.

Reason code: Unspecified

Date and Time: 9/ 4/2021 9:33 AM

Figure 12.16 – The Certificate Revocation pop-up window

7. Select a **Reason code** option from the list.
8. Select a date and time as the end of the validity time period for the certificate. The certificate will not be trusted beyond this date and time. Click **Yes**.
9. In the left navigation pane, click the **Revoked Certificates** node. The revoked certificate should appear in the list of revoked certificates in the main pane.

10. In the left navigation pane, right-click the **Revoked Certificates** node. From the context menu, select **All Tasks** and then click **Publish**. The **Publish CRL** pop-up window appears.
11. Click **OK**. This closes the **Publish CRL** pop-up window and publishes a new CRL.
12. Close the **Certification Authority** management console.

How it works...

A certificate is valid for its validity period, as specified in the certificate template and certificate. However, a certificate can be revoked before the end of the validity period.

Every time a certificate action is performed, the certificate revocation status is checked with the CA that issued the certificate. The CA offers a CRL for this purpose.

When you revoke a certificate, it is important to publish a new CRL that includes the revocation information for the newly revoked certificate(s). Otherwise, when the revocation status for the certificate is checked, the certificate is not included in the CRL and remains valid.

In the **Certification Authority** management console, multiple certificates can be selected using the *Shift* and *Ctrl* keys. This way, multiple certificates can be revoked at once.

Decommissioning a CA

When a CA is no longer needed, it can be decommissioned. This recipe shows how to decommission a CA.

Getting ready

Sign in with local administrator privileges to the standalone CA you intend to decommission.

If you intend to decommission an enterprise CA, sign in with a domain account that is a member of the **Enterprise Admins** group. Ensure that Active Directory replication works adequately.

How to do it...

Perform these steps to decommission a CA:

1. Press Start.
2. Search for **Server Manager** or run `servermanager.exe`. The **Server Manager** window appears.

3. In the gray top bar of **Server Manager**, click **Manage**.
4. Select **Remove Roles and Features** from the menu. The **Remove Roles and Features Wizard** window appears.
5. On the **Before you begin** screen, click **Next** >.
6. On the **Select destination server** screen, select the local Windows Server installation from the server pool list and click **Next** >.
7. On the **Remove server roles** screen, deselect the **Active Directory Certificate Services** role. The **Remove Roles and Features Wizard** pop-up window appears.
8. Click **Remove Features** to remove the features and close the pop-up window.
9. Click **Next** >.
10. On the **Remove features** screen, click **Next** >.
11. On the **Confirm removal selections** screen, click **Remove**.
12. On the **Removal progress** screen, click **Close**.

Alternatively, you can decommission the CA using the following line of PowerShell in an elevated Windows PowerShell window:

```
Uninstall-WindowsFeature ADCS-Cert-Authority
-IncludeManagementTools
```

The preceding line of PowerShell is the preferred way to decommission a CA on Server Core installations.

How it works...

Just like demoting a domain controller, decommissioning a CA revolves around removing the server role from the Windows Server installation. The **Remove Roles and Features** wizard performs all the configurations needed.

Ensure that all certificates issued by the CA to be decommissioned are either expired, revoked, or removed. Renew any necessary certificates from another CA before decommissioning a CA. Otherwise, certificates that may be required for continued operations will no longer be trusted.

13 Managing Federation

Active Directory Domain Services (**AD DS**) has been around for 20 years. Its interactions are based on protocols—such as **New Technology LAN Manager** (**NTLM**) and Kerberos—that Microsoft has invented and/or expanded on. In fact, these protocols originated before some companies were even connected to the internet era; they were intended for safe networks. However, today, there's a need for open protocols that are usable on all networks, allowing for interactions without technology boundaries. **Active Directory Federation Services** (**AD FS**) allows for these interactions.

AD FS was initially purposed for organization-to-organization collaboration without a need to set up and maintain Active Directory trusts. Recently, it gained traction as a common way to implement **single sign-on** (**SSO**) between AD DS on-premises and Azure AD.

The following recipes will be covered in this chapter:

- Choosing the right AD FS farm deployment method
- Installing the AD FS server role
- Setting up an AD FS farm with **Windows Internal Database** (**WID**)
- Setting up an AD FS farm with SQL Server
- Adding additional AD FS servers to an AD FS farm

- Removing AD FS servers from an AD FS farm
- Creating a **Relying Party Trust** (**RPT**)
- Deleting an RPT
- Configuring branding
- Migrating a WID-based AD FS farm to SQL Server
- Setting up a **web application proxy** (**WAP**)
- Decommissioning a WAP

Choosing the right AD FS farm deployment method

Before implementing AD FS, it's useful to have a plan for configuring it to integrate with the existing networking infrastructure, strategy, and intended use in the organization. Use this recipe to make the right choices.

Getting ready

Before choosing the right AD FS farm deployment method, it's a good idea to get to know the organization. Its size and its intended use of AD FS set the required number of AD FS servers. Its network layout may determine the available bandwidth between data centers and the AD FS servers in these data centers.

Many organizations consolidate individual SQL Server installations into more centralized SQL Server clusters and Always On availability groups. All three SQL Server implementation models are supported by the AD FS database(s), but some caveats exist.

You can expect organizations to have preferences regarding Windows Server operating system versions. Due to licensing constraints, organizations may want to stick with a previous version of Windows Server for AD FS. It is important that you are aware of these preferences.

Create an inventory of applications that the organization wants to make available through AD FS. Depending on the applications you want to make available through AD FS authentication, features such as Artifact Resolution might be required.

How to do it…

If the organization employs fewer than 1,500 people and/or fewer than 1,500 people use the AD FS farm at peak moments, then the organization only needs one AD FS server to meet the required capacity.

When more capacity or redundancy is required, additional choices are to be made.

When deploying an AD FS farm on Windows Server 2016, Windows Server 2019, or Windows Server 2022, a SQL Server cluster or SQL Server Always On availability group to host the AD FS database has to be created when:

- More than 30 **Structured Query Language** (**SQL**) servers are required
- AD FS management should be performed on any AD FS server
- Advanced AD FS features (such as **Artifact resolution** and **Token replay detection**) are required
- The organization prefers SQL Server to host the database

Otherwise, the AD FS farm may be deployed with WID. In the WID deployment model, changes in the database are replicated between the primary AD FS server and the other AD FS servers.

How it works…

There are three types of AD FS deployments:

- A single AD FS server using WID
- Using WID on each AD FS server in the farm, replicating through SQL Server
- Using a SQL server

Using WID, a single AD FS server offers a cost-effective AD FS solution but no redundancy and capacity for approximately 1,500 concurrent sessions at peak moments.

When deploying multiple AD FS servers, these servers need a way to exchange the AD FS farm information.

The WID deployment model can scale up to 30 servers in the latest versions of Windows Server, but in previous versions, it couldn't scale that far:

Windows Server version	**WID scale limit**
Windows Server 2008 Windows Server 2008 **Release 2** (**R2**) Windows Server 2012 Windows Server 2012 R2	5 AD FS servers
Windows Server 2016 Windows Server 2019 Windows Server 2022	30 AD FS servers

Table 13.1 – WID scale limits per Windows Server version

The preceding scale limits only apply to AD FS servers, not to AD FS proxies or WAPs attached to the AD FS farm.

The first common downside of the WID deployment model is that a primary AD FS server is in charge of (or has authority over) AD FS farm information. This means that AD FS can only be managed from the primary server in the WID deployment model.

The second common downside is that each AD FS server needs to be able to communicate with the primary AD FS server to replicate the AD FS farm information in the AD FS configuration database.

In comparison, the SQL Server deployment model doesn't suffer from these downsides. As long as the SQL Server implementation can handle requests from the AD FS farm in a timely fashion, the AD FS farm can continue to scale out.

When using a SQL server for the AD FS farm information, an AD FS artifact store is created next to the AD FS configuration database. This second database is used for the AD FS Artifact Resolution feature. Some **electronic identification and trust services** (**eIDAS**) implementations and other applications require this functionality.

Token Replay Detection is another AD FS feature that is only available when using a SQL server for AD FS farm information. When Token Replay Detection is a hard requirement from a security point of view, stick with the SQL Server deployment model.

Unfortunately, as the SQL Server deployment model leverages a SQL server, licensing fees may apply for the SQL Server installation and SQL Server **Client Access Licenses** (**CALs**). This makes the SQL Server deployment model a costly implementation model.

There's more...

After deploying an AD FS farm with WID, switching to the SQL Server deployment model is still possible.

See also

For more information, refer to the following recipes:

- *Setting up an AD FS farm with WID*
- *Setting up an AD FS farm with SQL Server*

Installing the AD FS server role

This recipe shows how to install the AD FS server role on prospective AD FS servers. This is the first of two steps to set up an AD FS farm.

Getting ready

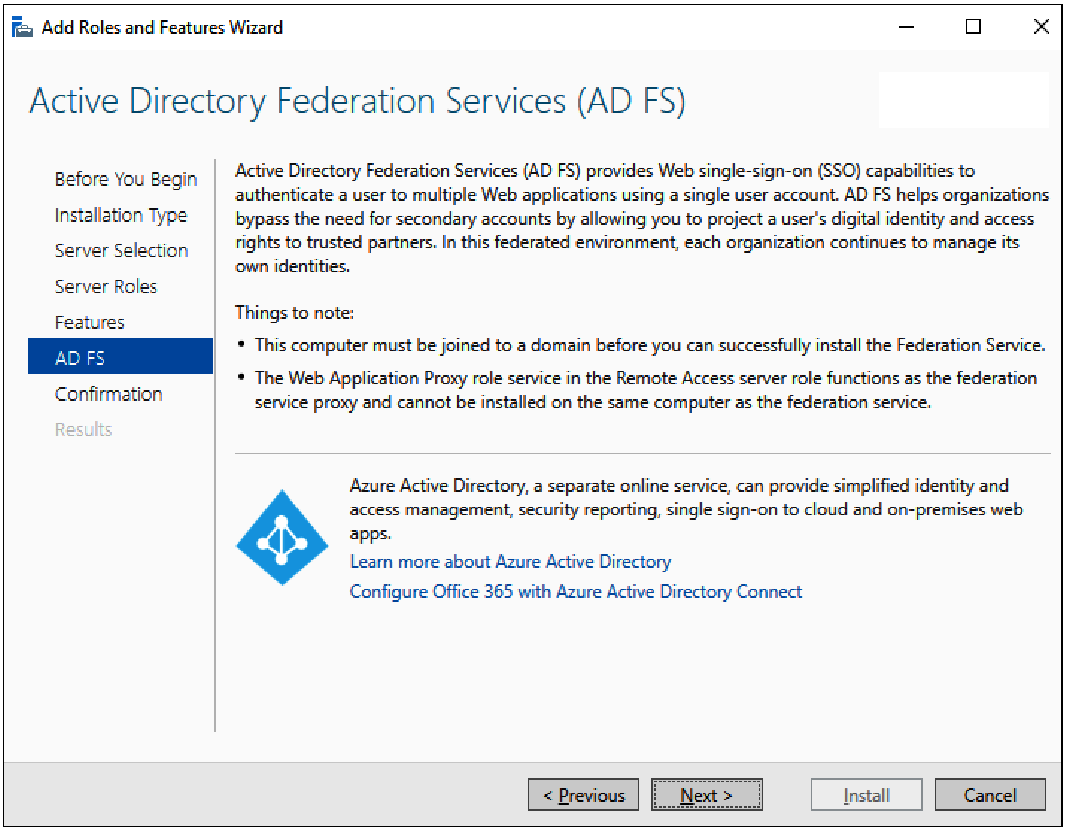
Sign in with local administrator privileges to a domain-joined Windows Server installation that you intend to use as an AD FS server for your organization.

How to do it...

To install the AD FS server role using the wizard, perform these steps:

1. Press Start.
2. Search for **Server Manager** and click its search result or run `servermanager.exe`. The **Server Manager** window appears.
3. In the gray top bar of **Server Manager**, click **Manage**.
4. Select **Add Roles and Features** from the menu. The **Add Roles and Features Wizard** window appears.
5. On the **Before you begin** screen, click **Next** >.
6. On the **Select installation type** screen, select **Role-based or feature-based installation**. Then, click **Next** >.
7. On the **Select destination server** screen, select the local Windows Server installation from the server pool list. Click **Next** > afterward.
8. On the **Select server roles** screen, select the **Active Directory Federation Services** role from the list of available roles. Then, click **Next** >.

9. On the **Select features** screen, click **Next** > to advance to the **Active Directory Federation Services (AD FS)** screen:



Active Directory Federation Services (AD FS)

Active Directory Federation Services (AD FS) provides Web single-sign-on (SSO) capabilities to authenticate a user to multiple Web applications using a single user account. AD FS helps organizations bypass the need for secondary accounts by allowing you to project a user's digital identity and access rights to trusted partners. In this federated environment, each organization continues to manage its own identities.

Things to note:

- This computer must be joined to a domain before you can successfully install the Federation Service.
- The Web Application Proxy role service in the Remote Access server role functions as the federation service proxy and cannot be installed on the same computer as the federation service.

Azure Active Directory, a separate online service, can provide simplified identity and access management, security reporting, single sign-on to cloud and on-premises web apps.

Learn more about Azure Active Directory

Configure Office 365 with Azure Active Directory Connect

Figure 13.1 – Active Directory Federation Services (AD FS) screen

10. On the **Active Directory Federation Services (AD FS)** screen, click **Next** >.
11. On the **Confirm installation selections** screen, click **Install**.
12. After the AD FS server role configuration completes, click **Close**.

You can also perform this task using PowerShell. Use the following line of PowerShell to install the AD FS server role in an elevated window:

```
Install-WindowsFeature ADFS-Federation -IncludeManagementTools
```

The preceding line also installs the AD FS Management tools.

How it works...

The AD FS server role adds the necessary files to configure a Windows Server as an AD FS server as part of an AD FS farm.

Its management tools include the AD FS module for Windows PowerShell and the AD FS Management tools. Unfortunately, these tools cannot be used remotely.

Setting up an AD FS farm with WID

This recipe shows how to set up an AD FS farm with WID. This is *Step 2* of setting up an AD FS farm.

Getting ready

In the organization, ensure that there is a consensus on the name of the AD FS farm.

Additionally, ensure that a **Transport Layer Security** (**TLS**) certificate is available for the AD FS farm name or request one from a **certification authority** (**CA**). Install the certificate in the **Personal** certificate store for the local machine.

To set up an AD FS farm, ensure that a domain-joined Windows Server installation is available to commission as an AD FS server and has the AD FS server role installed. Additionally, ensure the AD FS farm name is resolvable to this server in the appropriate **Domain Name System** (**DNS**) zones.

Sign in with an account that is a member of the **Domain Admins** group.

How to do it...

Setting up an AD FS farm with WID consists of two steps:

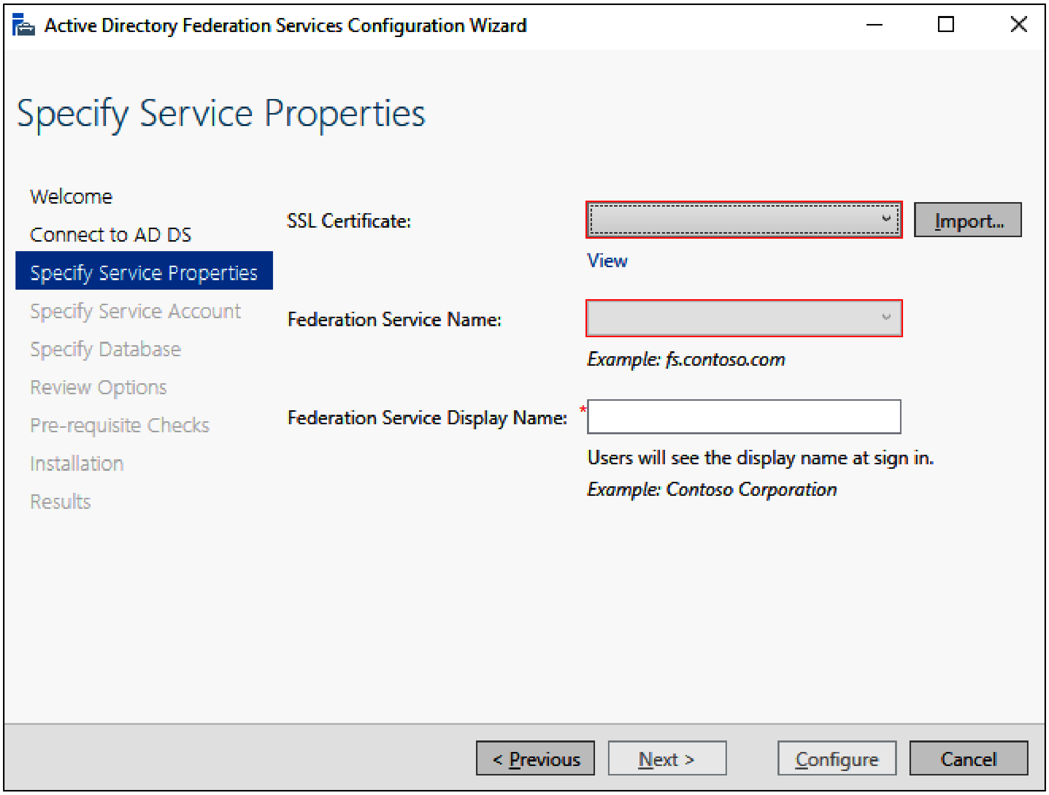
1. Configuring AD FS
2. Checking that AD FS was properly configured

Configuring AD FS

To set up an AD FS farm with WID, perform these steps:

1. Press Start.
2. Search for **Server Manager** and click the search result or run `servermanager.exe`. The **Server Manager** window appears.

3. In the gray top bar of **Server Manager**, click the warning sign to see a list of warnings and alerts.
4. Select the **Post-deployment Configuration** for the AD FS server role and click the **Configure the federation service on this server** link. The **Active Directory Federation Services Configuration Wizard** window appears.
5. On the **Welcome** screen, select the **Create the first federation server in a federation server farm** option and click **Next** >.
6. On the **Connect to Active Directory Domain Services** screen, click **Next** > as you have already signed in with an account that has administrative privileges. You reach the **Specify Service Properties** screen:



Example: fs.contoso.com

Users will see the display name at sign in.

Example: Contoso Corporation

Figure 13.2 – Specify Service Properties screen

7. On the **Specify Service Properties** screen, select the installed TLS certificate from the drop-down list. The **Federation Service Name** field is automatically selected from the **Subject** field of the certificate. If the certificate is a wildcard certificate, type the name. Then, type a name in the **Federation Service Display Name** field—for instance, `Lucern Publishing`. Click **Next** > when done.

8. On the **Specify Service Account** screen, select the **Create a Group Managed Service Account** option and type the account name for the **group Managed Service Account** (**gMSA**)—for instance, `ADFSgMSA`. Then, click **Next** >.
9. On the **Specify Configuration Database** page, acknowledge the **Create a database on this server using Windows Internal Database** default option by clicking **Next** >.
10. On the **Review Options** screen, click **Next** >.
11. On the **Prerequisite Checks** screen, click **Configure**.
12. On the **Results** screen, click **Close**.

Alternatively, use the following lines of PowerShell on a system with the AD FS Management tools and the Active Directory module for Windows PowerShell installed, while logged on with an account that is a member of the **Domain Admins** group:

```
New-ADServiceAccount -Name ADFSgMSA -DNSHostName adfsgmsa.
lucernpub.com
$ADFSFarmName = "adfs.lucernpub.com"
$Thumb = (Get-ChildItem -path cert:\LocalMachine\My | Where-
Object {$_.Subject -match $ADFSFarmName}).Thumbprint
Install-AdfsFarm -CertificateThumbprint
$thumb -FederationServiceName $ADFSFarmName
-GroupServiceAccountIdentifier lucernpub.com\ADFSgMSA$
```

Change the values for the gMSA username (`ADFSgMSA`), the AD FS farm name (`adfs.lucernpub.com`), and the domain name (`lucernpub.com`) to match your account naming convention and networking environment. As the gMSA object type originates from the computer object type, do not omit `$` from the end of the object name in the script.

Checking that AD FS was properly configured

After setting up the AD FS farm, the following **Uniform Resource Locators** (**URLs**) should be accessible from a web browser:

- `https://ADFSFarmName/FederationMetadata/2007-06/FederationMetadata.xml`
- `http://ADFSFarmName/adfs/probe`

How it works...

An AD FS farm handles authentication requests only when it is addressed with its farm name. For instance, *Lucern Publishing* might choose `adfs.lucernpub.com`, `sts.lucernpub.com`, or `signin.lucernpub.com` as its farm name.

A certificate is required as AD FS exchanges traffic for authentication requests using only TLS. This certificate may originate from a locally managed CA or a public CA, but it must always include the AD FS farm name in the **Subject** or **Subject Alternative Name** field.

AD FS offers services beyond end-user authentication requests. Ideally, when requesting a certificate, include the following URLs for AD FS farms with AD FS servers running Windows Server 2012 R2 and newer versions of Windows Server:

- `certauth.adfsfarmname.domain.tld`
- `enterpriseregistration.domain.tld`

The certificate for AD FS must meet the following requirements:

- It uses a non-**Cryptographic Next Generation** (**CNG**)-generated private key
- It uses **Secure Hashing Algorithm 2** (**SHA-2**) as the hashing algorithm
- It uses a 3,072-bit key length (or larger)

AD FS can use a user object dedicated as its service account or a gMSA. The latter option is the preferred option. gMSAs automatically change their password every 30 days, thereby adding additional security to the solution, when the following requirements are met:

- At least one domain controller runs Windows Server 2012 or a newer version of Windows Server
- Active Directory runs with the Windows Server 2008 **Forest Functional Level** (**FFL**) or above

To replicate the AD FS configuration database, by default, the AD FS servers in the farm communicate with the primary server every 5 minutes using **Transmission Control Protocol** (**TCP**) port `80`. Ensure that the networking ports are available between AD FS servers, on top of TCP port `443` for authentication requests and the networking ports that Microsoft recommends between AD FS and domain controllers.

There's more...

Microsoft's recommendation is to install AD FS servers as Server Core installations, as AD FS servers offer end-user services and are at an increased risk of being compromised. The lines of PowerShell in this recipe can be used to configure the AD FS farm in this scenario.

While AD FS can be installed and configured on a domain controller and can coexist with other roles, this is not a recommended practice from a security point of view.

See also

To add AD FS servers to the AD FS farm created in this recipe, look at the *Adding additional AD FS servers to an AD FS farm* recipe.

To migrate an AD FS farm using WID to SQL Server, perform the steps in the *Migrating a WID-based AD FS farm to SQL Server* recipe.

Setting up an AD FS farm with SQL Server

In this recipe, you learn how to set up an AD FS farm with a SQL Server-based backend. This is *Step 2* of setting up an AD FS farm and an alternative to the steps described in the *Setting up an AD FS farm with WID* recipe.

Getting ready

In the organization, ensure that there is a consensus on the name of the AD FS farm.

Ensure a TLS certificate is available for the AD FS farm name, or request one from a CA. Install the certificate in the **Personal** certificate store for the local machine.

To set up an AD FS farm, ensure that a domain-joined Windows Server installation is available to commission as an AD FS server and has the AD FS server role installed. Additionally, ensure the AD FS farm name is resolvable to this server in the appropriate DNS zones.

Have a domain-joined SQL server available on the network that is also resolvable through DNS and reachable by the proposed AD FS server(s).

Perform all the following steps with an account that has these specific permissions:

- Domain administrator privileges in Active Directory through membership of the **Domain Admins** group
- Local administrator privileges on the Windows Server installation, which is intended as the first AD FS server—this is a default permission for **Domain Admins** when the server is domain-joined
- **System administrator** (**SysAdmin**, or **sa**) privileges on the SQL server

How to do it...

To set up an AD FS farm with an SQL Server, perform the following steps:

1. Create a gMSA.
2. Create a script.
3. Create AD FS databases.
4. Configure AD FS.
5. Check that AD FS was properly configured.

Creating a gMSA

To create a gMSA, use the following line of PowerShell on a domain controller or any other domain-joined system with the Active Directory module for Windows PowerShell installed:

```
New-ADServiceAccount -Name ADFSgMSA -DNSHostName ADFSgMSA.
lucernpub.com -PrincipalsAllowedToRetrieveManagedPassword
ADFS01
```

Change the values for the gMSA username (`ADFSgMSA`) and the AD FS server name (`ADFS01`) to match your account naming convention and networking environment.

Creating a script

On the prospective AD FS server with the AD FS server role installed, use the following lines of PowerShell to create SQL scripts that configure databases on the SQL server:

```
New-Item "C:\ADFSSQLScript" -Type Directory
Export-AdfsDeploymentSQLScript -DestinationFolder "C:\
ADFSSQLScript" -ServiceAccountName LUCERNPUB\ADFSgMSA$
```

Change the values for the folder (`C:\ADFSSQLScript`) and the gMSA username (`LUCERNPUB\ADFSgMSA`). As the gMSA object type originates from the computer object type, do not omit `$` from the end of the object name in the script.

This creates two files in the specified folder:

- `CreateDB.sql`
- `Set-Permissions.sql`

Creating AD FS databases

Copy both files to the SQL server and then perform the following steps on the SQL server or any other system with **Microsoft SQL Server Management Studio** installed:

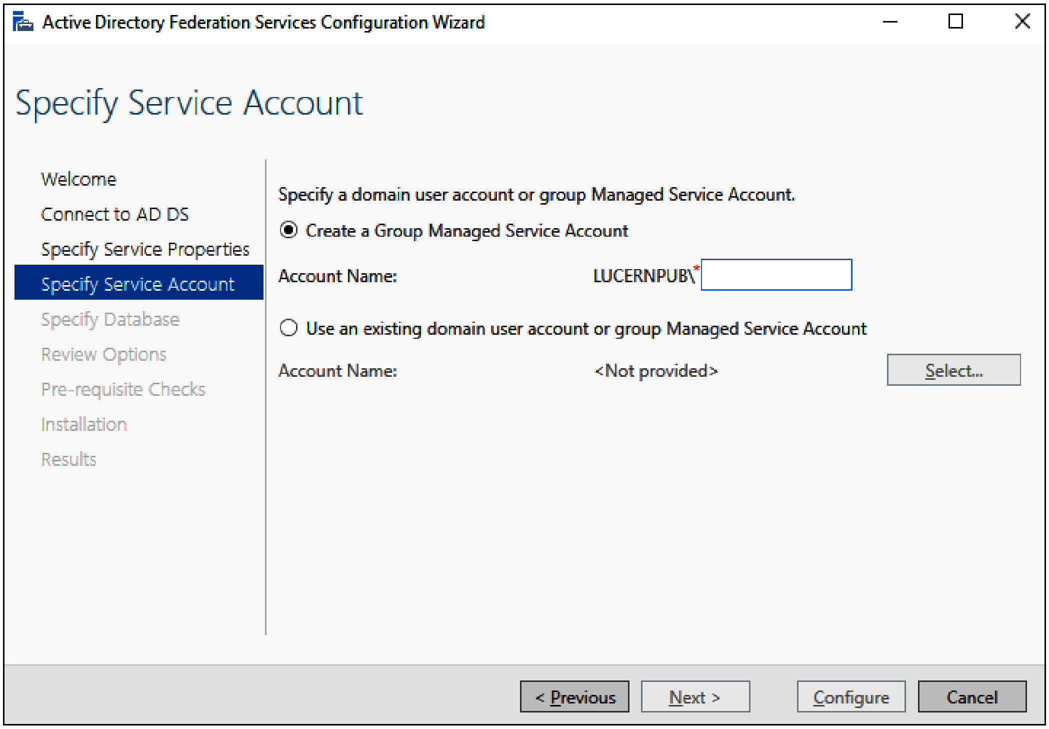
1. Press Start.
2. Search for **Microsoft SQL Server Management Studio** and click its search result or run `ssms.exe`. The **Microsoft SQL Server Management Studio** window appears with a **Connect to Server** pop-up window.
3. In the **Connect to Server** pop-up window, connect to the SQL database engine by providing the necessary information for the **Server name:** and **Authentication:** fields. Click **Connect**.
4. From the **File** menu, click the **Open** option. Then, click **File… (Ctrl + O)**. The **Open File** pop-up window appears.
5. Browse to the `CreateDB.sql` script. Select it and click **Open**.
6. From the **Query** menu, click **Execute (F5)**. This creates **AdfsArtifactStore** and **AdfsConfigurationV4** databases.
7. From the **File** menu, click the **Open** option. Then, click **File… (Ctrl + O)** again.
8. This time, browse to the `SetPermissions.sql` script and open it.
9. From the **Query** menu, click the **Execute (F5)** option. This creates a SQL login for the gMSA, grants it permissions to connect to SQL, and configures the **db_owner** and **db_genevaservice** role memberships scoped to the two previously created databases.

Configuring AD FS

To set up an AD FS farm with an SQL Server, perform these steps:

1. Press Start.
2. Search for **Server Manager** and click its search result or run `servermanager.exe`. The **Server Manager** window appears.
3. In the gray top bar of **Server Manager**, click the warning sign to see a list of warnings and alerts.
4. Select the **Post-deployment Configuration** for the AD FS server role and click the **Configure the federation service on this server** link. The **Active Directory Federation Services Configuration Wizard** window appears.

5. On the **Welcome** screen, select **Create the first federation server in a federation server farm** and then click **Next** >.
6. On the **Connect to Active Directory Domain Services** screen, click **Next** > as you have already signed in with an account that has administrative privileges.
7. On the **Specify Service Properties** screen, select the installed TLS certificate from the drop-down list. The **Federation Service Name** field is automatically selected from the **Subject** field of the certificate. If the certificate is a wildcard certificate, type the name. Then, type the **Federation Service Display Name** value—for instance, `Lucern Publishing`. Click **Next** > when done. You reach the **Specify Service Account** screen:



Active Directory Federation Services Configuration Wizard

Specify Service Account

Specify a domain user account or group Managed Service Account.

● Create a Group Managed Service Account

Account Name: LUCERNPUB*

○ Use an existing domain user account or group Managed Service Account

Account Name: <Not provided>

Figure 13.3 – Specify Service Account screen

8. On the **Specify Service Account** screen, select the **Use an existing domain user account or group Managed Service Account** option.

9. Click the **Select…** button. The **Select User or Service Account** pop-up window appears.
10. Type in the account name for the gMSA—for instance, `ADFSgMSA`—and then click **Check Names**. Click **OK** to select a gMSA and close the **Select User or Service Account** pop-up window.
11. Click **Next** >.
12. On the **Specify Configuration Database** screen, select the **Specify the location of a SQL Server database** option. Specify the SQL server's name in the **Database Host Name:** field. Then, specify **Database Instance:** or leave it blank to use the default instance. Click **Next** >.
13. On the **Confirm Overwrite** screen, select the **Overwrite existing AD FS configuration database data** option. Then, click **Next** >.
14. On the **Review Options** screen, click **Next** >.
15. On the **Prerequisite Checks** screen, click **Configure**.
16. On the **Results** screen, click **Close**.

Alternatively, use the following lines of PowerShell to set up an AD FS farm with the SQL Server:

```
$ADFSFarmName = "adfs.lucernpub.com"
$Thumb = (Get-ChildItem -path cert:\LocalMachine\My | Where-
Object {$_.Subject -match $ADFSFarmName}).Thumbprint
Install-ADFSFarm -CertificateThumbPrint
$thumb -FederationServiceDisplayName "Lucern
Publishing" -FederationServiceName $ADFSFarmName
-GroupServiceAccountIdentifier "LUCERNPUB\ADFSgMSA$"
-OverwriteConfiguration -SQLConnectionString "Data
Source=SQL01.lucernpub.com;IntegratedSecurity=True"
```

Change the values for the gMSA username (`LUCERNPUB\ADFSgMSA`), the AD FS farm name (`adfs.lucernpub.com`), the SQL server (`SQL01`), and the domain name (`lucernpub.com`) to match your account naming convention and networking environment. As the gMSA object type originates from the computer object type, do not omit `$` from the end of the object name in the script.

Checking that AD FS was properly configured

After configuring AD FS, the following URLs should be accessible from a web browser:

- `https://ADFSFarmName/FederationMetadata/2007-06/FederationMetadata.xml`
- `http://ADFSFarmName/adfs/probe`

How it works...

The certificate plays an essential role in setting up an AD FS farm with WID as AD FS exchanges traffic for authentication requests using only TLS.

When using an SQL Server, however, the service account is paramount because it is used to run the AD FS service on AD FS servers in the AD FS farm and connect to the SQL Server backend.

The `CreateDB.sql` script creates two databases: an **AdfsConfigurationV4** database, which stores the AD FS farm settings, and an **AdfsArtifactStore** database, which is used for AD FS Artifact Resolution.

The service account specified when running the script from `Export-AdfsDeploymentSQLScript` is added as a login to the SQL Server installation and is granted privileges:

- It is configured with the **public** role membership, server-wide.
- It is configured with the **db_owner** role membership on the **AdfsArtifactStore** and **AdfsConfigurationV4** databases.
- It is configured with the **db_genevaservice** role membership on the **AdfsArtifactStore** and **AdfsConfigurationV4** databases.

There's more...

In some environments, complex SQL connection strings might be returned by an SQL Server administrator in response to AD FS SQL scripts. In these cases, it is wise to test the SQL connection string manually on the AD FS servers before implementation. You can use the following lines of PowerShell for this purpose:

```
$conn = New-Object System.Data.SqlClient.SqlConnection
$conn.ConnectionString = "Data Source=SQL01.lucernpub.
com:Port;Integrated Security=True"
# If no error occurs here, then connection was successful.
```

```
$conn.Open();
$conn.Close();
```

Replace `SQL01.lucernpub.com` with the DNS name of the SQL Server database engine and replace `Port` with the port number on which the database engine listens. The default port number for the SQL Server is TCP `1433`.

See also

To add AD FS servers to the AD FS farm created in this recipe, look at the *Adding additional AD FS servers to an AD FS farm* recipe.

For more information on managing gMSAs, look at the *Working with group Managed Service Accounts* recipe in *Chapter 11*, *Securing Active Directory*.

Adding additional AD FS servers to an AD FS farm

This recipe demonstrates how to add additional AD FS servers to an AD FS farm.

Getting ready

After setting up the AD FS farm by implementing the first AD FS server, use this recipe to add additional AD FS servers to the farm.

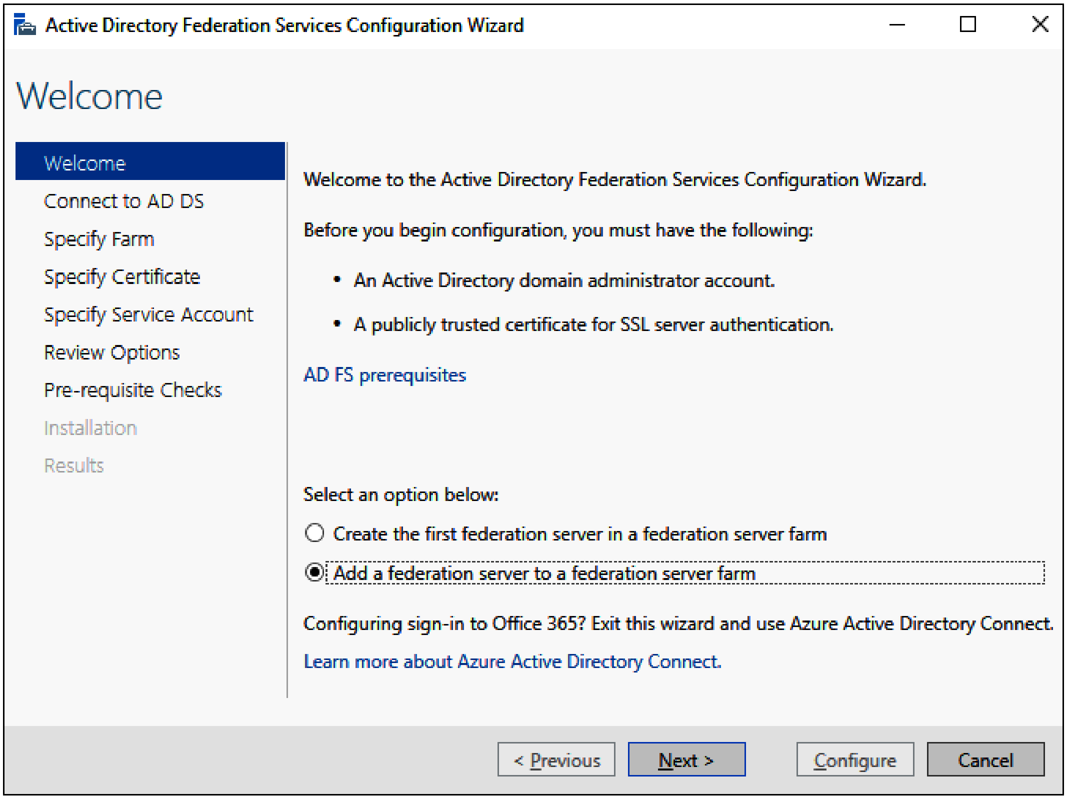
Before you begin, ensure that:

- The same TLS certificate used when configuring the first AD FS server is available for any additional AD FS servers in the same AD FS farm. Install the certificate in the **Personal** certificate store for the local machine.
- The proposed AD FS server is a domain-joined Windows Server installation, and you are signed in with a domain account that is a member of the **Domain Admins** group.
- The AD FS farm name resolves to all AD FS servers in the AD FS farm in the appropriate DNS zones.
- The primary AD FS server is reachable for AD FS farms using WID, and an SQL server is available and reachable for AD FS farms using an SQL Server as their backends.
- The AD FS server role is installed on the prospective AD FS server.

How to do it...

Perform the following steps per AD FS server that you want to add to the AD FS farm:

1. Press **Start**.
2. Search for **Server Manager** and click its search result or run `servermanager.exe`. The **Server Manager** window appears.
3. In the gray top bar of **Server Manager**, click the warning sign to see a list of warnings and alerts.
4. Select the **Post-deployment Configuration** for the AD FS server role and click the **Configure the federation service on this server** link. The **Active Directory Federation Services Configuration Wizard** window appears.
5. On the **Welcome** screen, select **Add a federation server to a federation server farm**:



Welcome to the Active Directory Federation Services Configuration Wizard.

Before you begin configuration, you must have the following:

- An Active Directory domain administrator account.
- A publicly trusted certificate for SSL server authentication.

AD FS prerequisites

Select an option below:

○ Create the first federation server in a federation server farm

● Add a federation server to a federation server farm

Configuring sign-in to Office 365? Exit this wizard and use Azure Active Directory Connect.

Learn more about Azure Active Directory Connect.

Figure 13.4 – Add a federation server to a federation server farm option

6. Click **Next** >.

7. On the **Connect to Active Directory Domain Services** screen, click **Next** > as you have already signed in with an account that has administrative privileges.
8. On the **Specify Farm** screen, choose from one of the following options:
    - Select the **Specify the primary federation server in an existing farm using Windows Internal Database** option and type the DNS name of the primary AD FS server in the **Primary Federation Server:** field. Click **Next** > afterward.
    - Select the **Specify the database location for an existing farm using SQL Server** option and specify the DNS name of the SQL Server in the **Database Host Name:** field. Optionally, specify a value for the **Database Instance:** field. Click **Next** > afterward.
9. On the **Specify SSL Certificate** screen, select the installed TLS certificate from the drop-down list and click **Next** >.
10. On the **Specify Service Account** screen, click **Select…**. The **Select User or Service Account** pop-up window appears.
11. Type in the account name for the gMSA—for instance, `ADFSgMSA`—and then click **Check Names**. Click **OK** to select a gMSA and close the **Select User or Service Account** pop-up window.
12. Click **Next** >.
13. On the **Review Options** screen, click **Next** >.
14. On the **Prerequisite Checks** screen, click **Configure**.
15. On the **Results** screen, click **Close**.

Alternatively, use the following lines of PowerShell while logged on with an account that is a member of the **Domain Admins** group:

```
$ADFSFarmName = "adfs.lucernpub.com"
$Thumb = (Get-ChildItem -path cert:\LocalMachine\My | Where-
Object {$_.Subject -match $ADFSFarmName}).Thumbprint
Add-AdfsFarmNode -CertificateThumbprint $thumb
-GroupServiceAccountIdentifier lucernpub.com\ADFSgMSA$
-SQLConnectionString "Data Source=SQL01.lucernpub.
com;Integrated Security=True"
```

Change the values for the gMSA username (`ADFSgMSA`), the AD FS farm name (`adfs.lucernpub.com`), the SQL server (`SQL01.lucernpub.com`), and the domain name (`lucernpub.com`) to match your previously set-up AD FS farm. As the gMSA object type originates from the computer object type, do not omit `$` from the end of the object name in the script.

How it works...

When an AD FS farm exists, AD FS servers can be added by specifying a primary AD FS server for AD FS farms using WID or a database for AD FS farms using an SQL Server.

Each AD FS server replicates the database from the primary AD FS server in an environment using WID. Only this server has read/write access to the database. The AD FS Management tools can only be run on the primary server. The primary server is located through DNS, and the WID is replicated using TCP port `80`, by default.

In an environment using an SQL Server, each AD FS server communicates with the SQL server for the database. Each server has read/write access to the database. The AD FS Management tools can be run on each AD FS server.

When multiple AD FS servers are added to an AD FS farm, use a load-balancer solution to distribute authentication requests over the servers. Use the `/adfs/probe` endpoint to check if an AD FS server is available in the load-balancer configuration.

Removing AD FS servers from an AD FS farm

This recipe shows how to remove AD FS servers from an AD FS farm.

Getting ready

Sign in with a user account that is a member of the **Domain Admins** group.

How to do it...

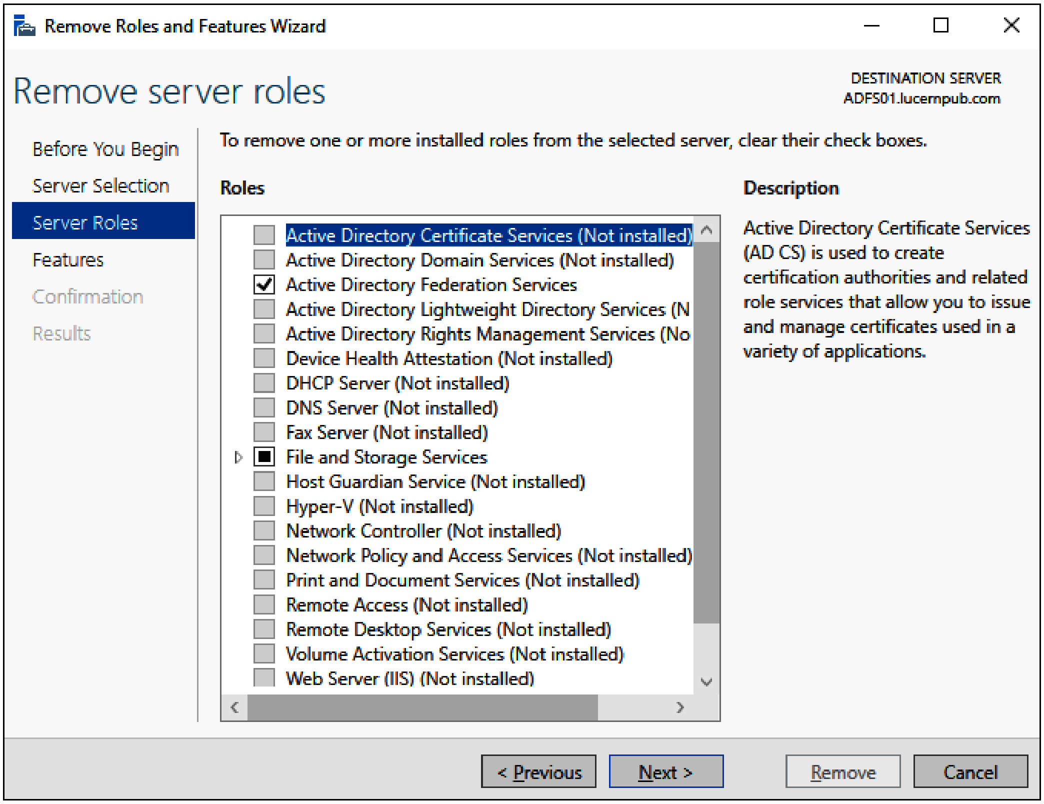
Perform the following steps:

1. Press **Start**.
2. Search for **Server Manager** and click its search result or run `servermanager.exe`. The **Server Manager** window appears.

3. From the gray top banner, click **Manage**. From the menu, select **Remove Roles and Features**.

   The **Remove Roles and Features Wizard** window appears.

4. On the **Before you begin** screen, click **Next** >.
5. On the **Select destination server** screen, acknowledge the local server by clicking **Next** >. You reach the **Remove server roles** screen:



Remove server roles

DESTINATION SERVER
ADFS01.lucernpub.com

To remove one or more installed roles from the selected server, clear their check boxes.

Roles

- □ Active Directory Certificate Services (Not installed)
- □ Active Directory Domain Services (Not installed)
- ☑ Active Directory Federation Services
- □ Active Directory Lightweight Directory Services (N
- □ Active Directory Rights Management Services (No
- □ Device Health Attestation (Not installed)
- □ DHCP Server (Not installed)
- □ DNS Server (Not installed)
- □ Fax Server (Not installed)
- ▷ ■ File and Storage Services
- □ Host Guardian Service (Not installed)
- □ Hyper-V (Not installed)
- □ Network Controller (Not installed)
- □ Network Policy and Access Services (Not installed)
- □ Print and Document Services (Not installed)
- □ Remote Access (Not installed)
- □ Remote Desktop Services (Not installed)
- □ Volume Activation Services (Not installed)
- □ Web Server (IIS) (Not installed)

Description

Active Directory Certificate Services (AD CS) is used to create certification authorities and related role services that allow you to issue and manage certificates used in a variety of applications.

Figure 13.5 – Remove server roles screen

6. On the **Remove server roles** screen, deselect the **Active Directory Federation Services** role. Afterward, click **Next** >.
7. If the AD FS server is part of an AD FS farm running WID, then, on the **Remove features** screen, deselect the **Windows Internal Database** feature and click **Next** >. If the AD FS farm uses an SQL Server, click **Next** > on the **Remove features** screen.
8. On the **Confirm removal selections** screen, click **Remove**.
9. When removal completes, click **Close** on the **Removal Progress** page.

How it works...

For most server roles and features in Windows Server, the decommissioning of the configured Windows Server installation occurs when the role or feature is removed.

When the AD FS role is removed from a server, it is automatically removed from the AD FS farm, and its AD FS role and its management tools are uninstalled.

Decommissioning AD FS by removing the server role does not remove other settings and infrastructure configurations, such as the TLS certificate, the service account from AD DS, domain membership of the server, and any memberships in the pools of load balancers. These items need to be removed or reconfigured manually afterward.

There's more...

When you decommission the last AD FS server in an AD FS farm, the farm is decommissioned. Remove any service accounts, load-balancer configurations, firewall rules, certificates, and DNS records pertaining to the former AD FS farm.

When you decommission an AD FS farm that leverages an SQL Server, then also decommission the corresponding **ADFSConfiguration** and **ArtifactStore** databases and service account login on the SQL server(s).

Creating an RPT

This recipe shows how to create an RPT in AD FS.

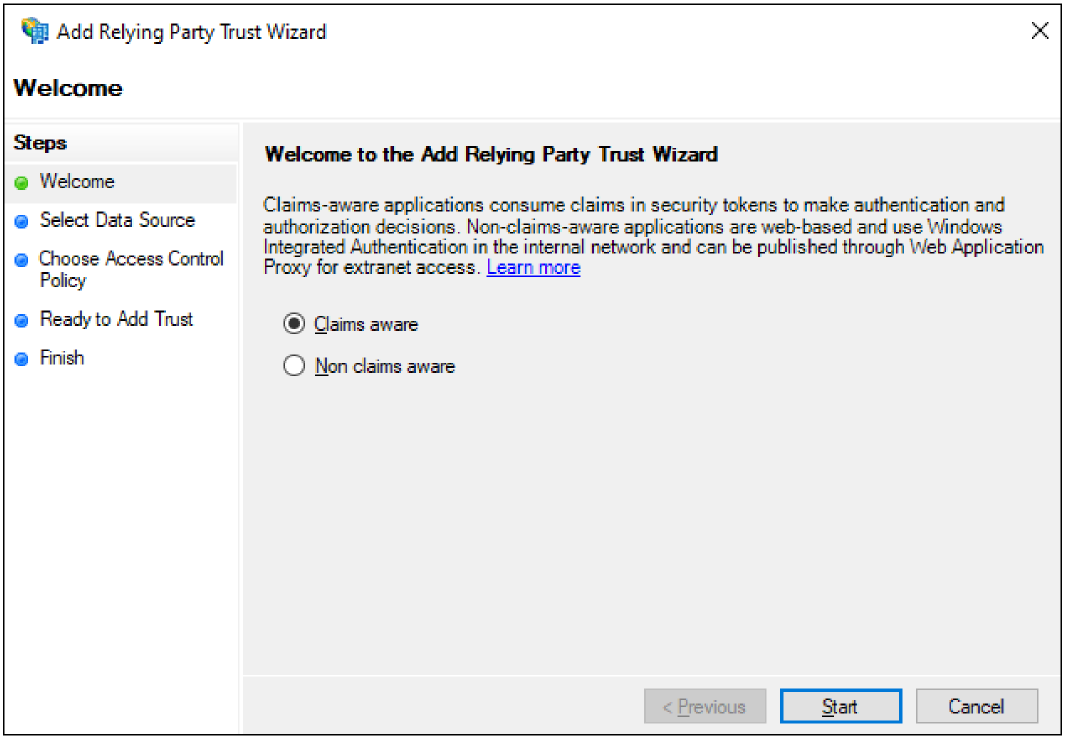
Getting ready

In an AD FS farm that uses WID, sign in to the primary AD FS server. Sign in with an account that is a member of the **Domain Admins** group.

How to do it...

Perform these steps to create an RPT:

1. Press Start.
2. Search for **AD FS Management** and click its search result or run `%windir%\ADFS\Microsoft.IdentityServer.msc`. The **AD FS** window appears.
3. In the right-hand **Actions** pane or from the **Action** menu, click **Add Relying Party Trust…**. The **Add Relying Party Trust Wizard** window appears:



Add Relying Party Trust Wizard

Welcome

Welcome to the Add Relying Party Trust Wizard

Claims-aware applications consume claims in security tokens to make authentication and authorization decisions. Non-claims-aware applications are web-based and use Windows Integrated Authentication in the internal network and can be published through Web Application Proxy for extranet access. Learn more

● Claims aware

○ Non claims aware

Figure 13.6 – Add Relying Party Trust Wizard window

4. On the **Welcome** screen, select the **Claims aware** option, and then click **Start**.
5. Choose between these three options:

   I. **Import data about the relying party published online or on a local network:**– Type the required information in the **Federation metadata address (host name or URL):** field.

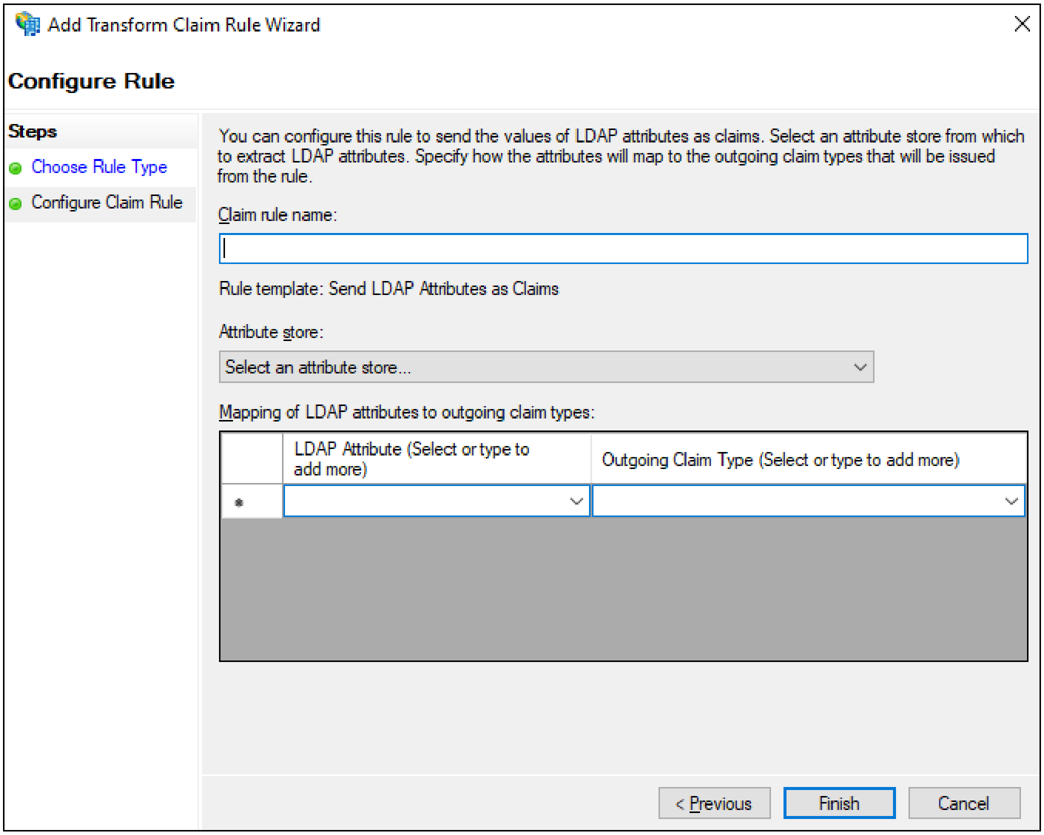
   II. **Import data about the relying party from a file:** – Type the filename in the **Federation metadata file location:** field.

   III. **Enter data about the relying party manually:**– For this RPT, the data is typed manually.

6. Click **Next** >.
7. On the **Specify Display Name** screen, enter a name for the RPT in the **Display Name:** field. Optionally, specify additional information in the **Notes:** field. Click **Next** > afterward.

8. On the **Configure Certificate** screen, optionally select a token encryption certificate by clicking **Browse…**, locating a certificate with a `.cer`, `.sst`, or `.p7b` file extension, and clicking **Open**. Click **Next** >.
9. On the **Configure URL** screen, select protocols for the RPT. Select the **Enable support for the WS-Federation Passive protocol** option and/or the **Enable support for SAML 2.0 Web SSO protocol** option and enter the required information. When done, click **Next** >.
10. On the **Configure Identifiers** screen, enter a value or URL for the **Relying party trust identifier:** field, and click **Add**. Then, click **Next** >.
11. On the **Choose Access Control Policy** screen, either accept the default **Permit Everyone** access control policy or select a different access control policy. Then, click **Next** >.
12. On the **Ready to Add Trust** screen, click **Next** >.
13. On the **Finish** screen, keep the **Configure claims issuance policy for this application** option enabled, and then click **Close**. The **Add Relying Party Trust Wizard** window closes and the **Edit Claims Issuance Policy** window for the RPT appears.
14. Click the **Add Rule…** button to add a claims issuance rule. The **Add Transform Claim Rule Wizard** window appears.
15. On the **Choose Rule Type** screen, select a claim rule template from the **Claim rule template:** drop-down list, and then click **Next** >.

    You are presented with the **Configure Rule** screen:



Add Transform Claim Rule Wizard

Configure Rule

You can configure this rule to send the values of LDAP attributes as claims. Select an attribute store from which to extract LDAP attributes. Specify how the attributes will map to the outgoing claim types that will be issued from the rule.

Claim rule name:

Rule template: Send LDAP Attributes as Claims

Attribute store: Select an attribute store...

Mapping of LDAP attributes to outgoing claim types:

	LDAP Attribute (Select or type to add more)	Outgoing Claim Type (Select or type to add more)
*		

Figure 13.7 – Configure Rule screen

16. On the **Configure Rule** screen, configure claims to be issued in the claim token for the RPT.
17. Click **Finish** to close the **Configure Rule** screen and save the claim rule.
18. In the **Edit Claims Issuance Policy** window for the RPT, click **OK** to save the issuance transform rules for the RPT.

How it works...

Without RPTs, an AD FS farm doesn't provide functionality. By adding one or more RPTs, an AD FS administrator could unlock SSO access to one or more applications, systems, and/or services.

An RPT adds a relationship between an application, system, and/or service and the AD FS farm. In this relationship, the application trusts claims issued by the AD FS farm. When an authenticated user presents claims to the application, the application uses the content of the claims to perform authentication and authorization.

Optionally, claims can be encrypted. When a TLS certificate is exchanged between the administrator of the application and the AD FS administrator, a token encryption certificate can be configured on both sides and used to encrypt claims that are exchanged. This is useful to further protect the contents of claims beyond the default TLS connection that is used to exchange claims.

Considering regulations such as the **General Data Protection Regulation** (**GDPR**) and the **California Consumer Privacy Act** (**CCPA**), applications, systems, and services focus on the safe exchange of data and require encrypted claims. The AD FS server encrypts claims with the certificate's private key. The application decrypts the claim token with the public key of the certificate.

The **Import data about the relying party published online or on a local network:** option should be your preferred option to add RPTs, as it adds the ability to monitor and—optionally—update information for RPTs automatically.

Alas, some claims-aware applications, systems, and services do not offer federation metadata online, and the **Enter data about the relying party manually:** option needs to be used. In this case, when editing claims issuance rules, do not type in the fields below **Mapping of LDAP attributes to outgoing claim types:** on the **Configure Rule** screen of the **Add Transform Claim Rule Wizard** window. Instead, select them from their respective drop-down lists.

Deleting an RPT

This recipe shows how to delete an RPT in AD FS.

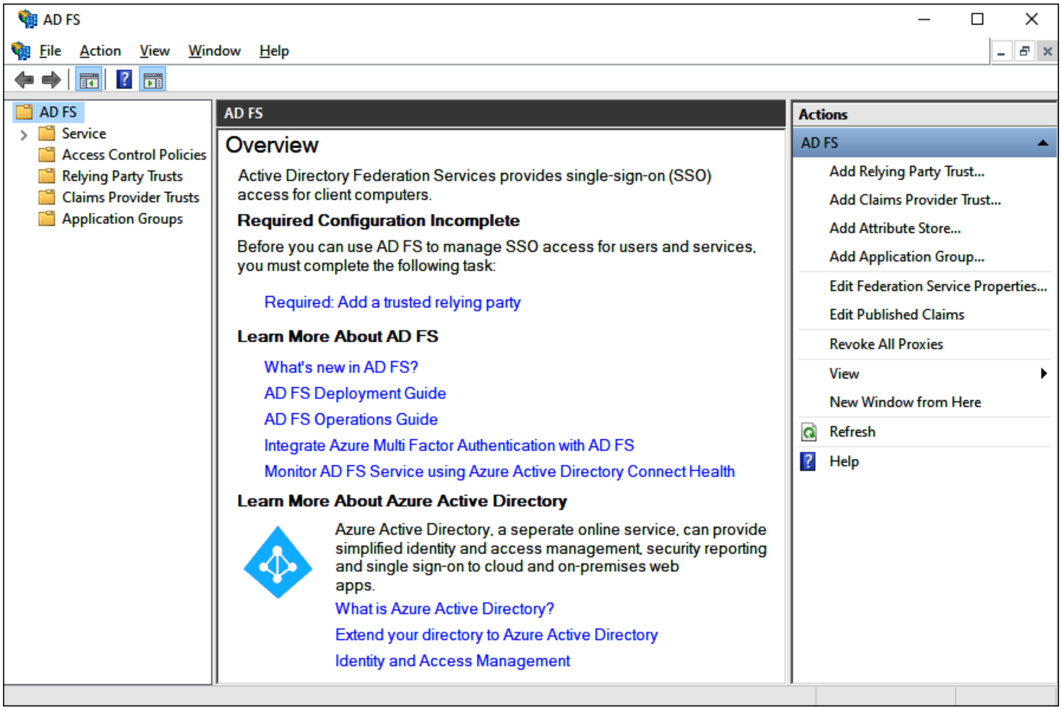
Getting ready

In an AD FS farm that uses WID, sign in to the primary AD FS server. Sign in with an account that is a member of the **Domain Admins** group.

How to do it...

Perform these steps to delete an RPT:

1. Press Start.
2. Search for **AD FS Management** and click its search result or run `%windir%\ADFS\Microsoft.IdentityServer.msc`. The **AD FS** window appears:



Overview

Active Directory Federation Services provides single-sign-on (SSO) access for client computers.

Required Configuration Incomplete

Before you can use AD FS to manage SSO access for users and services, you must complete the following task:

Required: Add a trusted relying party

Learn More About AD FS

What's new in AD FS?

AD FS Deployment Guide

AD FS Operations Guide

Integrate Azure Multi Factor Authentication with AD FS

Monitor AD FS Service using Azure Active Directory Connect Health

Learn More About Azure Active Directory

Azure Active Directory, a seperate online service, can provide simplified identity and access management, security reporting and single sign-on to cloud and on-premises web apps.

What is Azure Active Directory?

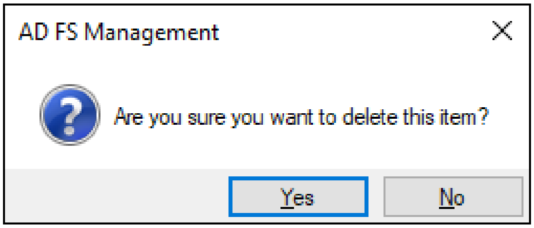
Extend your directory to Azure Active Directory

Identity and Access Management

Figure 13.8 – AD FS window

3. In the left navigation pane, expand the **Relying Party Trusts** node.
4. In the main pane, select an RPT that you want to delete.

5. In the right-hand **Actions** pane, click **Delete**, or right-click the RPT and select **Delete** from the menu. The **AD FS Management** pop-up window appears:



AD FS Management

Are you sure you want to delete this item?

Figure 13.9 – AD FS Management pop-up window

6. In the **AD FS Management** pop-up window, click **Yes** as the answer to the **Are you sure you want to delete this item?** question.

How it works...

When an RPT is no longer needed, it can be removed from the AD FS farm. Without an RPT, end-user access to the application is lost.

Configuring branding

This recipe shows how to apply your organization's branding to AD FS sign-in pages.

Getting ready

Ask the marketing team for your organization to produce the following files:

- One logo, ideally 280 pixels wide and 60 pixels high
- One big background picture, ideally 1,420 pixels wide and 1,200 pixels tall, not exceeding 200 **kilobytes** (**KB**) in size
- Disclaimer text

Place these files in a folder on the AD FS server—for instance, in the `C:\Style` folder.

In an AD FS farm that uses WID, sign in to the primary AD FS server and place the files in a folder on this server. Sign in with an account that is a member of the **Domain Admins** group.

How to do it...

Perform the following steps:

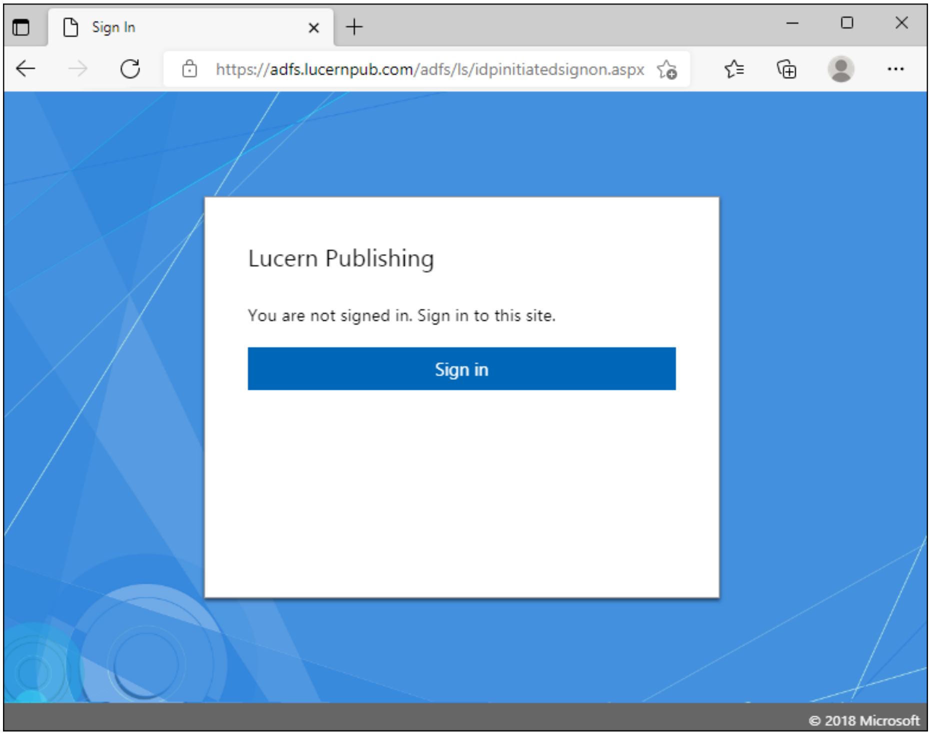
1. Right-click Start and select **Windows PowerShell (Admin)** from the menu. The elevated **Windows PowerShell** window appears.
2. Use the following command to switch the appearance of the AD FS farm's login pages to a paginated experience with a centered **user interface** (**UI**):

```
Set-AdfsGlobalAuthenticationPolicy
-EnablePaginatedAuthenticationPages $true
```

3. Type `Y` followed by *Enter* to confirm.
4. Use the following command to enable the `/adfs/ls/idpinitiatedsignon.aspx` page:

```
Set-AdfsProperties -EnableIdpInitiatedSignonPage $true
```

This allows you to see the branding in action. Navigate to the page to see the default branding; it resembles the following screenshot:



Sign In

https://adfs.lucernpub.com/adfs/ls/idpinitiatedsignon.aspx

Lucern Publishing

You are not signed in. Sign in to this site.

Figure 13.10 – Default AD FS branding

5. Perform the following command to create a custom theme based on the default theme:

```
New-AdfsWebTheme –Name custom -SourceName default
```

6. Perform the following command to change the AD FS farm name to the organization's logo:

```
Set-AdfsWebTheme -TargetName custom -Logo @{path="C:\
Style\logo.png"}
```

7. Perform the following command to change the background:

```
Set-AdfsWebTheme -TargetName custom –Illustration @
{path="C:\Style\background.jpg"}
```

8. Perform the following command to add disclaimer text:

```
Set-AdfsGlobalWebContent –SignInPageDescriptionText
"<p>By logging on, you gain access to services. When
using these services, rules apply as stated in the
protocol. Unauthorized access is prohibited.</p>"
```

9. Perform the following command to switch from the default theme to a custom theme:

```
Set-AdfsWebConfig -ActiveThemeName custom
```

10. Navigate to the `/adfs/ls/idpinitiatedsignon.aspx` page to see the custom branding.
11. Perform the following command to disable the page:

```
Set-AdfsProperties –EnableIdpInitiatedSignonPage $False
```

12. Communicate the layout to the marketing department to get a sign-off.

How it works...

Colleagues may encounter AD FS sign-in pages when they can't use Kerberos to sign in, when they can't use a previous session, or when they have to provide additional information such as **multi-factor authentication** (**MFA**). To make AD FS sign-in pages an organizational experience, branding can be applied.

AD FS supports themes. By adding a custom theme, themes can be switched easily, allowing for quick rollbacks if needed.

Starting with AD FS on Windows Server 2016, the `/adfs/ls/idpinitiatedsignon.aspx` page is no longer enabled by default. This adds to information security because malicious persons outside the organization can use the page to discover RPTs.

If you are starting with AD FS on Windows Server 2019, then a paginated experience with a centered UI is available by default. For Windows Server 2016, the experience needs to be downloaded from GitHub first.

Migrating a WID-based AD FS farm to an SQL Server

This recipe shows how to migrate a WID-based AD FS farm to an SQL Server.

Getting ready

Have a domain-joined SQL server available on the network that is also resolvable through DNS and reachable by the proposed AD FS server(s).

Perform all the following steps with an account that has these specific permissions:

- Domain administrator privileges in Active Directory through membership of the **Domain Admins** group
- Local administrator privileges on the primary AD FS server in the AD FS farm
- **sa** privileges on the SQL server

How to do it…

To migrate a WID-based AD FS farm to the SQL Server, perform the following steps on the primary AD FS server:

1. Create a script.
2. Create AD FS databases.
3. Detach databases from the AD FS server.
4. Copy database files.
5. Attach databases to the SQL server.
6. Reconfigure AD FS.
7. Remove the WID feature.

Then, perform *Steps 2*, *5*, and *6* for all other AD FS servers in the AD FS farm.

Creating a script

On the primary AD FS server, use the following lines of PowerShell to create SQL scripts that configure databases on the SQL server:

```
New-Item "C:\ADFSSQLScript" -Type Directory


Export-AdfsDeploymentSQLScript -DestinationFolder "C:\
ADFSSQLScript" -ServiceAccountName LUCERNPUB\ADFSgMSA$
```

Change the values for the folder (`C:\ADFSSQLScript`). When the WID-based AD FS farm uses a user account to run the **Active Directory Federation Services** service on AD FS servers, replace the gMSA username (`LUCERNPUB\ADFSgMSA`) and omit `$` from the end. When the WID-based AD FS farm uses a gMSA, add `$` at the end of the object name in the script.

This creates two files in the specified folder:

- `CreateDB.sql`
- `Set-Permissions.sql`

Creating AD FS databases

Copy both files to the SQL server and then perform the following steps on the SQL server or any other system with **Microsoft SQL Server Management Studio** installed:

1. Press Start.
2. Search for **Microsoft SQL Server Management Studio** and click its search result or run `ssms.exe`. The **Microsoft SQL Server Management Studio** window appears with a **Connect to Server** pop-up window.
3. In the **Connect to Server** pop-up window, connect to the SQL database engine by providing the necessary information for the **Server name:** and **Authentication:** fields. Click **Connect**.
4. From the **File** menu, click the **Open** option. Then, click **File... (Ctrl + O)**. The **Open File** pop-up window appears.
5. Browse to the `CreateDB.sql` script. Select it and click **Open**.
6. From the **Query** menu, click **Execute (F5)**. This creates **AdfsArtifactStore** and **AdfsConfigurationV4** databases.

7. From the **File** menu, click the **Open** option. Then, click **File… (Ctrl + O)** again.
8. This time, browse to the `SetPermissions.sql` script and open it.
9. From the **Query** menu, click the **Execute (F5)** option. This creates a SQL login for the gMSA, grants it permissions to connect to SQL, and configures the **db_owner** and **db_genevaservice** role memberships scoped to the two previously created databases.

Detaching databases from the AD FS server

On the primary AD FS server, use the following line of PowerShell to stop the **Active Directory Federation Services** service:

```
Stop-Service adfssrv
```

Wait for the service to stop. Then, run the following lines of PowerShell to detach AD FS databases from WID:

```
$connection = New-Object -TypeName System.Data.SqlClient.
SqlConnection
$connection.ConnectionString = 'Server=np:\.
pipeMICROSOFT##WIDtsqlquery;Database=master;Trusted_
Connection=True;'
$connection.Open()
$command = $connection.CreateCommand()
$command.CommandText = "EXEC sp_detach_db @dbname =
N'AdfsArtifactStore';"
$result = $command.ExecuteReader()
$connection.close()
$connection.Open()
$command = $connection.CreateCommand()
$command.CommandText = "EXEC sp_detach_db @dbname =
N'AdfsConfigurationV4';"
$result = $command.ExecuteReader()
$connection.close()
```

The database files are safely detached and can now be copied to the SQL server.

Copying database files

Copy the following four files from the `C:\Windows\WID\Data` folder:

- `AdfsConfigurationV4.mdf`
- `AdfsConfigurationV4_log.ldf`
- `AdfsArtifactStore.mdf`
- `AdfsArtifactStore.ldf`

Attaching databases to the SQL server

Copy files to the SQL server from the `C:\Program Files\Microsoft SQL Server\MSSQL.15\MSSQL\DATA\` folder and perform these steps:

1. Press Start.
2. Search for **Microsoft SQL Server Management Studio** and click its search result or run `ssms.exe`. The **Microsoft SQL Server Management Studio** window appears with the **Connect to Server** pop-up window in front of it.
3. In the **Connect to Server** pop-up window, connect to the SQL database engine by providing the necessary information for the **Server name:** and **Authentication:** fields. Click **Connect**.
4. In the left navigation pane, expand **Databases**.
5. Right-click **the AdfsArtifactStore** database and select **Attach** from the menu. The **Attach Database** window appears.
6. In the **Attach Database** window, click **Add**.
7. Browse to the `AdfsArtifactStore.mdf` file.
8. Click **OK**.
9. Right-click the **AdfsConfigurationV4** database and select **Attach** from the menu. The **Attach Database** window appears.
10. In the **Attach Database** window, click **Add**.
11. Browse to the `AdfsConfigurationV4.mdf` file.
12. Click **OK**.

Reconfiguring AD FS

Run the following lines of PowerShell on the AD FS server to reconfigure the AD FS service to use databases on the SQL server:

```
$Conn= Get-WmiObject -namespace root/ADFS -class
SecurityTokenService
$Conn.ConfigurationdatabaseConnectionstring="data source=SQL01.
lucernpub.com; initial catalog=adfsconfiguration;integrated
security=true"
$Conn.put()
```

Replace `SQL01.lucernpub.com` with the hostname of the SQL server hosting the AD FS databases.

Then, start the **Active Directory Federation Services** service with the following line of PowerShell:

```
Start-Service adfssrv
```

Lastly, change the connection for the **AdfsArtifactStore** database:

```
Set-adfsproperties -artifactdbconnection "data source= SQL01.
lucernpub.com; initial catalog=adfsartifactstore;integrated
security=true"
Restart-Service adfssrv
```

Replace `SQL01.lucernpub.com` with the hostname of the SQL server hosting the AD FS databases.

Removing the WID feature

As the WID feature is no longer needed on the AD FS server, run the following line of PowerShell to remove it:

```
Uninstall-WindowsFeature Windows-Internal-Database
```

On all other AD FS servers in the farm, perform the steps from the *Detaching databases from the AD FS server*, *Reconfiguring AD FS*, and *Removing the WID feature* sections.

How it works...

When the WID deployment model for the AD FS farm no longer satisfies the organization's needs, the AD FS farm can be switched to use an SQL Server installation.

The AD FS configuration database stores all the settings for the AD FS farm. It contains all the settings, RPTs, published applications, token-signing and token-encryption certificates, and claims issuance policies. To avoid recreating and reconfiguring these settings, the AD FS configuration database needs to be migrated when switching between the WID-based and the SQL Server-based AD FS deployment model.

For AD FS farms running Windows Server 2019 and newer versions of Windows Server, the AD FS configuration database is named **ADFSConfigurationV4**. In Windows Server 2016, the database was named **ADFSConfigurationV3**, and in Windows Server 2012 R2, the database was named **ADFSConfiguration**. Change the database name in the lines of PowerShell in this recipe when migrating AD FS farms running previous versions of Windows Server.

The AD FS artifact store contains artifacts that applications, systems, and services may collect from the AD FS farm when their RPTs are configured with Artifact Resolution.

When databases are in use, they can't be copied without the risk of data corruption. Therefore, any database to be copied needs to be paused or detached.

Setting up a WAP

This recipe shows how to set up a WAP to publish an AD FS farm to the internet.

Getting ready

After setting up the AD FS farm, add one or more WAPs.

Before you begin, ensure the following:

- The same TLS certificate used on AD FS servers in the AD FS farm is available on the intended WAP. Install the certificate in the **Personal** certificate store for the local machine.
- The AD FS farm name is resolvable to AD FS servers for the WAP.
- The AD FS farm name is resolvable to the WAP from the internet.

How to do it...

Setting up a WAP consists of three steps:

1. Installing the **Web Application Proxy** feature
2. Configuring the WAP
3. Checking the proper WAP configuration

Sign in with a local administrator account on the WAP.

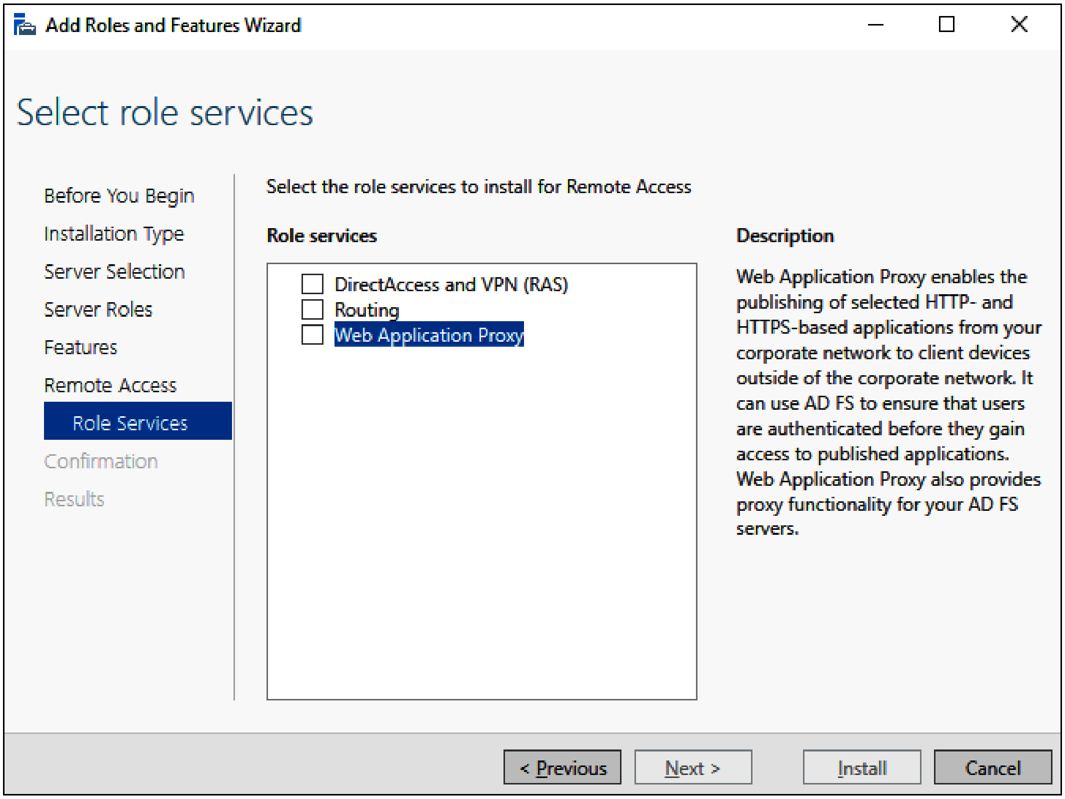
Installing the Web Application Proxy feature

Perform the following steps:

1. Press **Start**.
2. Search for **Server Manager** and click the search result or run `servermanager.exe`. The **Server Manager** window appears.
3. In the gray top bar of **Server Manager**, click **Manage**.
4. Select **Add Roles and Features** from the menu.

   The **Add Roles and Features Wizard** window appears.
5. On the **Before you begin** screen, click **Next** >.
6. On the **Select installation type** screen, select **Role-based or feature-based installation**. Afterward, click **Next** >.
7. On the **Select destination server** screen, select the local Windows Server installation from the server pool list. Click **Next** > when done.
8. On the **Select server roles** screen, select the **Remote Access** role from the list of available roles. Then, click **Next** >.
9. On the **Select features** screen, click **Next** >.

10. On the **Remote Access** screen, click **Next** >. You reach the **Select role services** screen:



Add Roles and Features Wizard

Select role services

Select the role services to install for Remote Access

Role services

□ DirectAccess and VPN (RAS)
□ Routing
□ Web Application Proxy

Description

Web Application Proxy enables the publishing of selected HTTP- and HTTPS-based applications from your corporate network to client devices outside of the corporate network. It can use AD FS to ensure that users are authenticated before they gain access to published applications. Web Application Proxy also provides proxy functionality for your AD FS servers.

Figure 13.11 – Select role services screen

11. On the **Select role services** screen, select the **Web Application Proxy** feature. The **Add Roles and Features Wizard** pop-up window appears.
12. In the **Add Roles and Features Wizard** pop-up window, click **Add Features** to add the **Group Policy Management**, **RAS Connection Manager Administration Kit (CMAK)**, and **Remote Server Administration Tools** features for remote access.

13. In the **Add Roles and Features Wizard** window, on the **Select role services** screen, click **Next** >.
14. On the **Confirm installation selections** screen, click **Install**.
15. When the configuration of the **Web Application Proxy** server role service is done, click **Close** on the **Installation Progress** screen.

Alternatively, this task can be accomplished using Windows PowerShell. Use the following line of PowerShell to install the **Web Application Proxy** server role service in an elevated window:

```
Install-WindowsFeature Web-Application-Proxy
-IncludeManagementTools
```

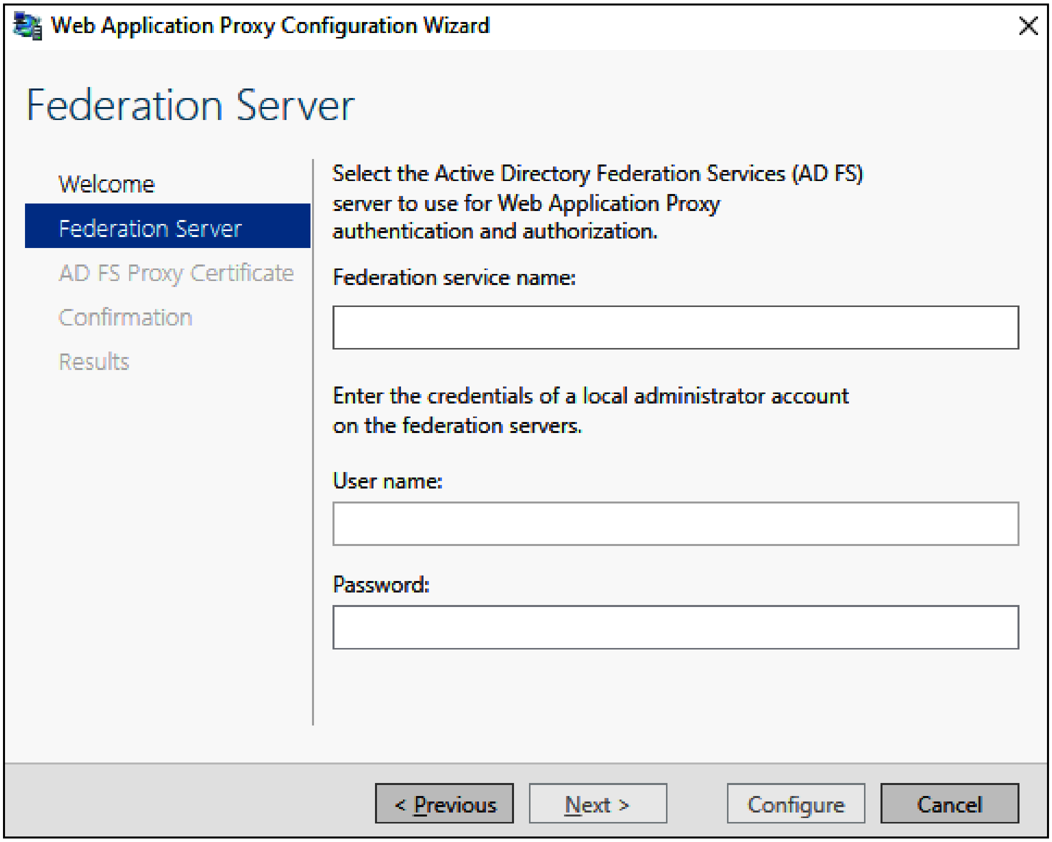
The management tools include **Group Policy Management**, **RAS Connection Manager Administration Kit (CMAK)**, and **Remote Server Administration Tools** features for remote access.

Configuring the WAP

Perform the following steps:

1. Press Start.
2. Search for **Server Manager** and click the search result or run `servermanager.exe`. The **Server Manager** window appears.
3. In the gray top bar of **Server Manager**, click the warning sign to see a list of warnings and alerts.
4. Select the **Post-deployment Configuration** for the **Web Application Proxy** server role service and then click the **Open the Web Application Proxy Wizard** link. The **Web Application Proxy Configuration Wizard** window appears.

5. On the **Welcome** screen, click **Next** >. You reach the **Federation Server** screen:



Web Application Proxy Configuration Wizard

Federation Server

Select the Active Directory Federation Services (AD FS) server to use for Web Application Proxy authentication and authorization.

Federation service name:

Enter the credentials of a local administrator account on the federation servers.

User name:

Password:

Figure 13.12 – Federation Server screen

6. On the **Federation Server** screen, type the AD FS farm name—for instance, `adfs.lucernpub.com`—in the **Federation service name:** field. Next, go to **Enter the credentials of a local administrator account on the federation servers** and fill in the respective fields. Click **Next** > afterward.

7. On the **AD FS Proxy Certificate** screen, choose **Select a certificate to be used by the AD FS proxy:** from the drop-down list and click **Next** >.

8. On the **Confirmation** screen, click **Configure**.

9. On the **Results** screen, click **Close**.

Alternatively, this task can be accomplished using Windows PowerShell. Use the following lines of PowerShell in an elevated window:

```
$ADFSFarmName = "adfs.lucernpub.com"

$Thumb = (Get-ChildItem -path cert:\LocalMachine\My | Where-
Object {$_.Subject -match $ADFSFarmName}).Thumbprint

Install-WebApplicationProxy -CertificateThumbprint $Thumb
-FederationServiceName $ADFSFarmName
```

Change the values for the AD FS farm name (`adfs.lucernpub.com`) to match your previously set-up AD FS farm.

Checking the proper WAP configuration

After configuring the WAP, the following URL should be accessible from the internet:

- `https://ADFSFarmName/FederationMetadata/2007-06/FederationMetadata.xml`

How it works...

WAP servers can be used to publish the AD FS farm on the internet safely. For authentication requests toward the AD FS farm from the internet, a WAP functions as a reverse proxy for AD FS servers, terminating the connection yet relaying authentication requests to the AD FS servers.

When an authentication request comes in via a WAP, the `insidecorporatenetwork` claim is set to `false`. This claim is leveraged in AD FS access control policies to distinguish between outside and inside clients. Administrators can require MFA for outside clients using the built-in access control policy by using a configured MFA adapter.

WAP servers have a certificate-based relationship with the AD FS farm. Using `MS-ADFSPIP`, a certificate is obtained and automatically renewed using TCP port `443` between the WAP and AD FS servers only.

There's more...

By default, the `/adfs/probe` endpoint is not accessible on WAP servers.

Configure Windows Defender Firewall to allow traffic to the endpoint using the following line of PowerShell in an elevated Windows PowerShell window on the WAP:

```
New-NetFirewallRule -Name Allow_HTTP -DisplayName "AD FS HTTP
Services" -Protocol TCP -LocalPort 80 -Profile Any -Action
Allow
```

Allowing **HyperText Transfer Protocol** (**HTTP**) traffic makes WAP servers monitorable as part of a load balancer's backend pool.

Decommissioning a WAP

This recipe shows how to decommission a WAP.

Getting ready

Sign in with a local administrator account on the WAP.

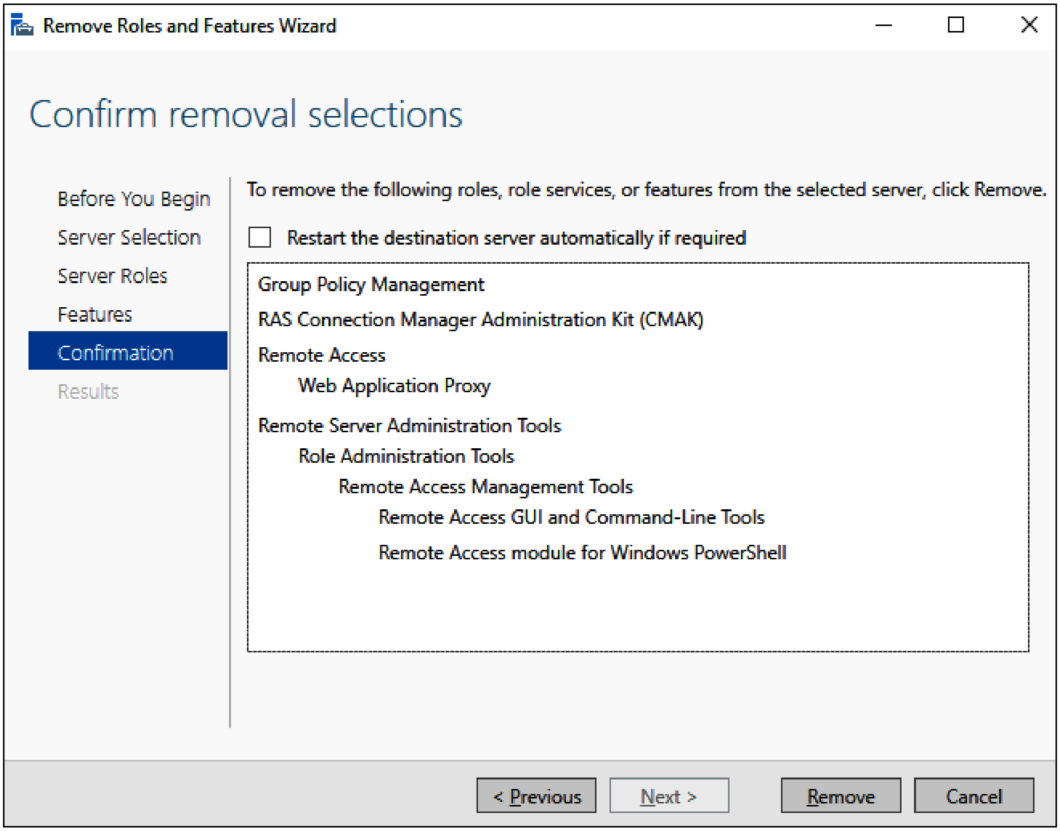
How to do it...

Perform the following steps:

1. Press Start.
2. Search for **Server Manager** and click its search result or run `servermanager.exe`. The **Server Manager** window appears.
3. In the gray top bar of **Server Manager**, click **Manage**.
4. Select **Remove Roles and Features** from the menu.
5. On the **Before you begin** screen, click **Next** >.
6. On the **Select installation type** screen, click **Next** >.
7. On the **Remove server roles** screen, deselect the **Remote Access** role.

   The **Remove features that require Remote Access** pop-up window appears.
8. In the **Remove features that require Remote Access** pop-up window, click **Remove Features** to remove **Remote Server Administration Tools** for remote access.
9. Click **Next** >.

10. On the **Remove features** screen, deselect the **Group Policy Management** feature and the **RAS Connection Manager Administration Kit (CMAK)** feature, unless these features are required for other functionalities and you'd like to keep them.
11. Click **Next** >. The **Confirm removal selections** screen appears:



Confirm removal selections

To remove the following roles, role services, or features from the selected server, click Remove.

□ Restart the destination server automatically if required

Group Policy Management
RAS Connection Manager Administration Kit (CMAK)
Remote Access
    Web Application Proxy
Remote Server Administration Tools
    Role Administration Tools
        Remote Access Management Tools
            Remote Access GUI and Command-Line Tools
            Remote Access module for Windows PowerShell

Figure 13.13 – Confirm removal selections screen

12. On the **Confirm removal selections** screen, click **Remove**.
13. On the **Removal progress** screen, click **Close**.

Finally, restart the server.

How it works...

For most **Server Roles** and features in Windows Server, decommissioning of the configured Windows Server installation occurs when a role or feature is removed.

When the **Remote Access** role is removed from a server, the server stops functioning as a WAP.

Decommissioning a WAP by removing the **Web Application Proxy** server role service does not remove other settings and infrastructure configurations, such as a TLS certificate and any memberships of load-balancer pools. These configuration items need to be removed or reconfigured manually afterward.

14
Handling Authentication in a Hybrid World (AD FS, PHS, PTA, and DSSO)

Azure Active Directory (**Azure AD**) is Microsoft's cloud-based identity and access management service. Organizations can register for an Azure AD tenant, where they can store and use the information on their identities.

Hybrid identity is Microsoft Marketing speak for connecting an on-premises Active Directory environment to Azure AD. When done correctly, the hybrid identity implementation allows end users to authenticate to both on-premises and cloud-based applications, systems, and services:

- When accessing **Windows NT Lan Manager** (**NTLM**)-based, **lightweight directory access protocol** (**LDAP**)-based, and Kerberos-integrated applications, systems, and services, the on-premises Active Directory takes care of authentication and authorization. These protocols are designed for safe networks and have been offering **Single Sign-On** (**SSO**) for decades.
- When accessing cloud-based applications, systems, and services, it is ill-advised to use the protocols for safe networks or other proprietary protocols. Many cloud service providers use open authentication protocols and interfaces, such as **Web Services Federation Language** (**WS-Fed**), **Security Assertion Markup Language** (**SAML**), OAuth2, OpenID Connect, and **System for Cross-domain Identity Management** (**SCIM**). Both **Active Directory Federation Services** (**AD FS**) and Azure AD offer these protocols and interfaces.

Hybrid identity consists of two distinct areas:

- Authentication
- Synchronization

This chapter discusses the first area in terms of AD FS, **Password Hash Synchronization** (**PHS**), **Pass-through Authentication** (**PTA**), and Seamless SSO. Synchronization is discussed in *Chapter 15*, *Handling Synchronization in a Hybrid World (Azure AD Connect)*.

The following recipes are covered in this chapter:

- Signing up for Azure AD
- Choosing the proper authentication method
- Verifying your DNS domain name
- Implementing PHS with Express Settings

- Implementing PTA and Seamless SSO
- Implementing SSO using AD FS
- Managing AD FS with Azure AD Connect
- Implementing Azure Traffic Manager for AD FS geo-redundancy
- Migrating from AD FS to PTA for SSO to Office 365
- Making PTA (geo)redundant

Choosing the right authentication method

This recipe shows how to choose the right authentication method between Active Directory and the Azure AD tenant.

Getting ready

To make a choice, you'll need to understand the following characteristics of your organization:

- Is your organization OK with synchronizing secrets to the (public) cloud for end users?
- Does your organization already have a federation solution and use claims-based applications inside your organization or cloud-based applications, systems, and/or services?
- Does your organization rely on on-premises multi-factor authentication solutions?
- Is your organization's **Security Incident and Event Monitoring** (**SIEM**) solution cloud-aware?
- Do people in your organization use Internet Explorer, Edge, or another browser as their default browser?

How to do it...

Use the following flowchart to choose the proper authentication method for your organization:

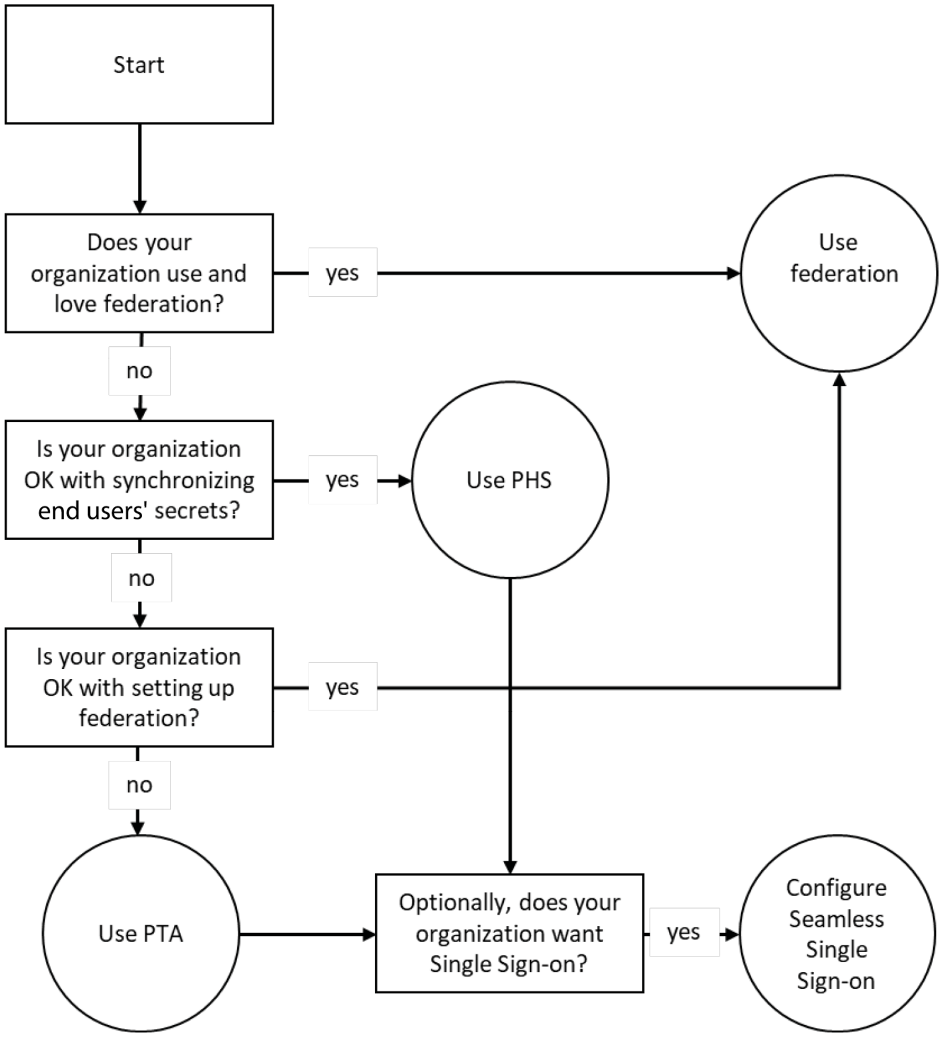

Start
Does your organization use and love federation?
yes
Use federation
no
Is your organization OK with synchronizing end users' secrets?
yes
Use PHS
no
Is your organization OK with setting up federation?
yes
no
Use PTA
Optionally, does your organization want Single Sign-on?
yes
Configure Seamless Single Sign-on

Figure 14.1 – A flowchart to choose the proper authentication method

How it works…

In the following subsections, you'll find information about the pros and cons of each authentication method.

AD FS and PingFederate

When combined with AD FS or PingFederate, Azure AD offers a claims-based trust between Azure AD and the organization's federation solution. This offers a claims-based SSO experience.

When AD FS or PingFederate is in use within an organization, the knowledge and processes are already in place to make connecting with Azure AD a success.

Since an AD FS implementation for Azure AD requires public connectivity for federation metadata exchange, using AD FS as the authentication method in a redundant setup requires at least five on-premises systems. These are listed as follows:

- At least two AD FS servers
- At least two Web Application Proxy servers, ideally in a perimeter network
- At least one Azure AD Connect installation

For organizations, additional training for personnel alongside the additional processes to keep AD FS optimized have proven too steep unless organizations have already deployed AD FS or another federation solution, such as PingFederate.

PTA

PTA enables organizations to continue having an SSO experience on their domain-joined devices, even those running Windows versions before Windows 10. Additionally, PTA provides audit trail capabilities that are quite similar to AD FS, without the infrastructure overhead typically associated with implementing a highly available AD FS infrastructure.

PTA relies on PTA agents. One or more PTA agents are installed within the network boundary. These agents maintain outbound TLS connections to the Azure Service Bus. Azure AD places authentication requests on the service bus, where the PTA agents pick them up and process them against the on-premises domain controllers. Then, they put the signals for successful authentication back on the service bus for Azure AD to pick up.

PHS

PHS is the default authentication method when using Azure AD Connect with the default settings. While Microsoft labeled this scenario as "Password Hash Sync," they've made sure that neither the password itself nor the hash of the password stored for the user object in Active Directory are synchronized to Azure AD. Rather, a hash of the hash of the password is synchronized.

PHS is the only scenario where Active Directory is not leveraged for actual authentications out of the three authentication scenarios. Therefore, no authentication events appear in the Windows event log of the domain controllers for sign-ins to cloud applications. When a SIEM solution is present within the networking infrastructure, it is paramount that it can also integrate with Azure AD to have access to the logs from there.

Seamless SSO

When the optional **Enable single sign-on** setting is enabled in Azure AD Connect, next to PHS or PTA, it instructs Azure AD Connect to offer Kerberos authentication using a specific computer object in the local Active Directory for this feature: `AzureADSSOACC`. This account is in the default **Computers** container. The object creates two **Service Principal Names** (**SPNs**) defining the Azure AD URLs. The account credentials are then shared by Azure AD Connect with Azure AD as the shared Kerberos secret to encrypt the interchanged Kerberos packets.

Using the SSO authentication method, people authenticating to Azure AD on domain-joined devices communicate using the Kerberos protocol to Azure AD's authentication service, just like they would with an on-premises federation service, except this service runs in Microsoft's data centers.

Cloud-only

Of course, organizations can always choose to manually implement accounts in Azure AD that have no relationship with the user accounts in Active Directory. These accounts authenticate to Azure AD's authentication service.

There's more...

The preceding authentication methods can be mixed, matched, and staged, but moving from one authentication method to another might not always yield the expected outcome. For example, moving from PHS to PTA does not clear the previously synchronized hashes of password hashes.

Therefore, it's a good plan to start any hybrid identity journey by deciding what authentication method to use.

Signing up for Azure AD

The recipes in this chapter are based on Azure AD. If your organization doesn't have an Azure AD tenant, use this recipe to create one.

Getting ready

You'll need a valid email address and phone number to sign up for Azure AD.

How to do it

Sign up for Azure AD by following the steps on the Microsoft website:

`https://signup.microsoft.com/Signup?OfferId=B07A1127-DE83-4a6d-9F85-2C104BDAE8B4.`

Provide a valid email address.

How it works...

The preceding link creates a non-expiring tenant with a 30-day Microsoft Office 365 E3 trial license. You specify an account with the **Global administrator** role during the process of signing up.

The URL in this recipe includes a form. It provides a way to name the Azure AD tenant. This is a plus because many other signup methods base the tenant's name on the domain name in the email address of the person signing up.

Note that you will not receive unsolicited email messages from Microsoft, and Microsoft does not share your data with its partners.

Verifying your DNS domain name

Any hybrid identity journey starts with verifying your DNS domain name in Azure AD. This recipe explains how to do this.

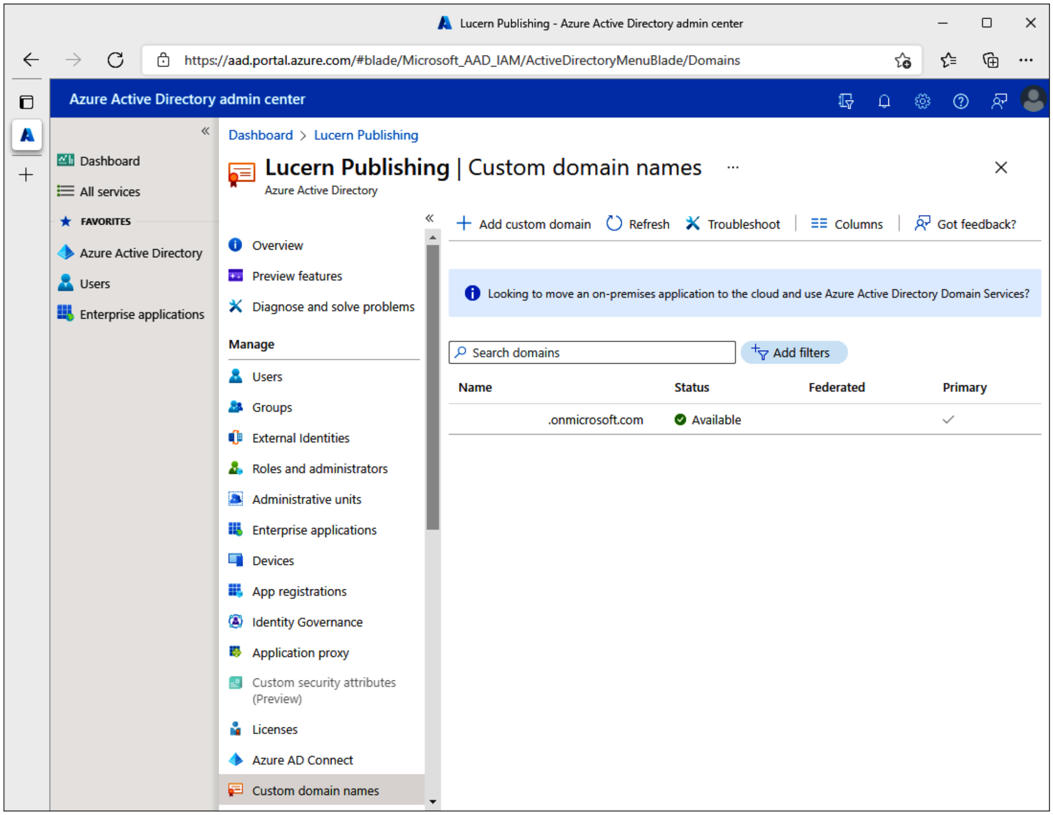
Getting ready

If the organization uses the Azure AD **Privileged Identity Management** (**PIM**) feature, activate the **Global administrator** role or the **Domain Name administrator** role in advance.

How to do it…

Perform the following steps in the Azure AD Tenant:

1. Navigate to `https://aad.portal.azure.com` in your browser.
2. Sign in with an account that has the **Global administrator** role or the **Domain Name administrator** role assigned to it.
3. Perform multi-factor authentication when prompted.
4. In the left-hand navigation pane, click **Azure Active Directory**.
5. In the Azure AD pane, click **Custom domain names**:



Looking to move an on-premises application to the cloud and use Azure Active Directory Domain Services?

Name	Status	Federated	Primary
.onmicrosoft.com	Available		✓

Figure 14.2 – Custom domain names in the Azure Active Directory admin center window

6. In the top-level bar, click **+ Add custom domain**. The **Custom domain name** blade appears on the right-hand side.
7. Enter the name of the DNS domain name that you want to add to the **Custom domain name** field.

8. Click the **Add domain** button at the bottom of the blade. You are then presented with the **Custom domain name** pane to verify the DNS name.
9. Use the information on this pane to create the DNS records that will prove your organization owns the DNS domain.
10. After creating the records, click the **Verify** button at the bottom of the pane.

How it works...

Perform the steps in this recipe before attempting to implement any of the following recipes in the remainder of this book. People in your organization might have already accepted invitations from people in other Azure AD tenants to work together. For instance, people already have Azure AD accounts based on their email addresses in Power BI. After a DNS domain name is verified in Azure AD, these accounts come into the scope of the Azure AD tenant.

When you first synchronize between Active Directory and Azure AD, these accounts are automatically reconfigured with the new authentication method. If you synchronize before scooping up these user objects, people might end up with two accounts, which defeats the purpose of SSO. If you synchronize without a verified DNS domain name, all user objects in Azure AD will be appended by the tenant name, for instance, `lucernpub.onmicrosoft.com`. AD FS isn't available as an authentication method without first verifying a DNS domain name.

If your organization's on-premises Active Directory does not utilize a publicly routable UPN suffix, for instance, `lucernpub.local`, perform the following steps:

1. Decide on an appropriate DNS domain name for your organization.

   If your organization uses email, use the DNS domain name that is used for email purposes.
2. Register the DNS domain name if your organization does not own it on the internet.
3. Verify the DNS domain name in the Azure AD tenant by adding the required TXT or MX records in the public DNS zone for the DNS domain name.
4. Add the DNS domain name as a UPN suffix in Active Directory.
5. Change the `userPrincipalName` attribute for each user object in the Active Directory to use the new DNS domain name as the UPN suffix.

Microsoft recommends configuring user objects with identical primary email addresses and `userPrincipalName` attributes. That way, when people in your organization are asked to authenticate using a method that specifies the @sign, it doesn't matter what they enter.

Implementing PHS with Express Settings

This recipe shows you how to configure PHS as the authentication method toward Azure AD, using Azure AD Connect Express Settings.

This recipe assumes your organization already possesses an Active Directory domain and Azure AD tenant.

Getting ready

Dedicate at least one domain-joined Windows Server system on the internal network as the host for Azure AD Connect for your organization. As this Windows Server will have a SQL Server Express database hosted on it, be sure not to combine this role with sensitive or overburdened hosts.

Ensure all accounts in the on-premises Active Directory are configured with a publicly routable `userPrincipalName` suffix, such as `lucernpub.com`. Ensure the DNS domain name(s) that are part of the `userPrincipalName` attributes for user accounts are owned by your organization on the internet and configured as verified DNS domain name(s) in your organization's Azure AD tenant.

Additionally, ensure that the Windows Server that will run Azure AD Connect meets the following requirements:

- It can communicate with the internet without passing proxies.
- It is running Windows Server 2016 or later.
- It is domain-joined.
- It has the **Internet Explorer Enhanced Security Configuration** (**IE ESC**) feature turned off.

If proxies need passing, take the appropriate measures by making a proxy exception or configuring a proxy for Azure AD Connect in its configuration file.

Sign in as a local administrator on the Windows Server installation.

Download the latest version of Azure AD Connect from the following URL:

`https://www.microsoft.com/en-us/download/details.aspx?id=47594`

As part of the following steps, you'll need to enter the credentials for these accounts:

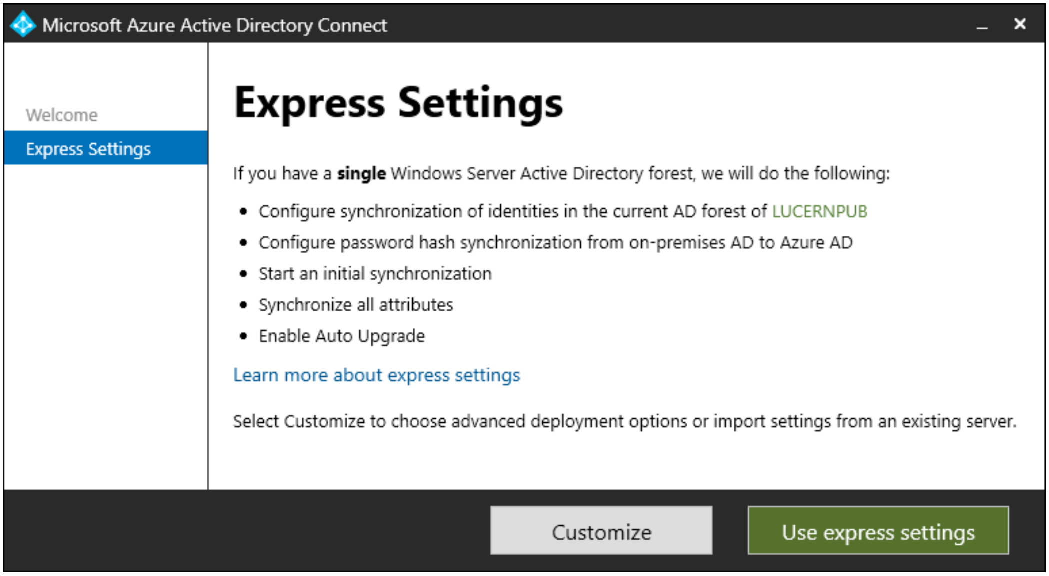
- An account in Active Directory that is a member of the **Enterprise Admins** group
- An account in Azure AD that has the **Global administrator** role or the **Hybrid Identity administrator** role assigned

If the organization uses the Azure AD PIM feature, activate the **Global administrator** role or the **Hybrid Identity administrator** role in advance.

How to do it...

Perform the following steps on the dedicated Windows Server installation that you want to use to configure Azure AD Connect with Express Settings:

1. Double-click `AzureADConnect.msi`. The **Microsoft Azure Active Directory Connect** window appears.
2. On the **Welcome** screen, select the **I agree to the license terms and privacy notice.** option.
3. Click **Next**. You are presented with the **Express Settings** screen:



Express Settings

If you have a **single** Windows Server Active Directory forest, we will do the following:

- Configure synchronization of identities in the current AD forest of LUCERNPUB
- Configure password hash synchronization from on-premises AD to Azure AD
- Start an initial synchronization
- Synchronize all attributes
- Enable Auto Upgrade

Learn more about express settings

Select Customize to choose advanced deployment options or import settings from an existing server.

Figure 14.3 – Express Settings in the Microsoft Azure Active Directory Connect window

4. On the **Express Settings** screen, click **Use express settings**.
5. On the **Connect to Azure AD** screen, enter the credentials of an account in Azure AD that has been assigned the **Global administrator** role or the **Hybrid Identity administrator** role.
6. Click **Next**.
7. When prompted, perform multi-factor authentication.

8. On the **Connect to AD DS** screen, enter the credentials for an account that is a member of the **Enterprise Admins** group in your on-premises Active Directory.
9. Click **Next**.
10. On the **Azure AD Sign-in** screen, review the DNS domain names for the Azure AD tenant. Ensure those domains that your organization uses have been verified in Azure AD.
11. Click **Next**.
12. On the **Ready to configure** screen, click **Install**.
13. When the installation completes, click **Exit** to close the **Microsoft Azure Active Directory Connect** window and start synchronization with Azure AD.

How it works...

When you install and configure Azure AD Connect with **Express Settings**, the following configuration is performed:

- All user objects, groups, Windows 10-based computer objects, contacts, and `InetOrgPerson` objects in the Active Directory forest, that the Azure AD Connect installation belongs to, are configured to be synchronized to Azure AD.
- All user objects and `InetOrgPerson` objects have hashes of their password hashes synchronized to Azure AD.
- Azure AD Connect's **Automatic Upgrade** feature is enabled.

Do not use **Express Settings** in the following situations:

- If your organization uses multiple Active Directory forests
- If your organization has over 100,000 objects, including groups and computer objects

Implementing PTA and Seamless SSO

This recipe shows how to configure Azure AD Connect with PTA and Seamless SSO.

Getting ready

To implement PTA, you'll need to sign in with an account that is a local administrator on the server dedicated to Azure AD Connect. As part of the steps of this recipe, you'll need to enter the credentials for the following accounts:

- An account in Active Directory that is a member of the **Enterprise Admins** group
- An account in Azure AD that has the **Global administrator** role assigned

Ensure the Windows Server that will run Azure AD Connect meets the following requirements:

- It can communicate with the internet without passing proxies.
- It is running Windows Server 2016 or later.
- It is domain-joined.
- It has the IE ESC feature turned off.

If proxies need passing, take appropriate measures by making a proxy exception or configuring a proxy for Azure AD Connect in its configuration file.

Ensure that the accounts in the on-premises Active Directory in scope of a synchronization to Azure AD are configured with a publicly routable `userPrincipalName` suffix, such as `lucernpub.com`. Ensure the DNS domain name(s) representing the UPN suffix(es) are owned by your organization on the internet and are configured as verified DNS domain name(s) in your organization's Azure AD tenant.

Download the latest version of Azure AD Connect from the following URL:

`https://www.microsoft.com/en-us/download/details.aspx?id=47594`

How to do it...

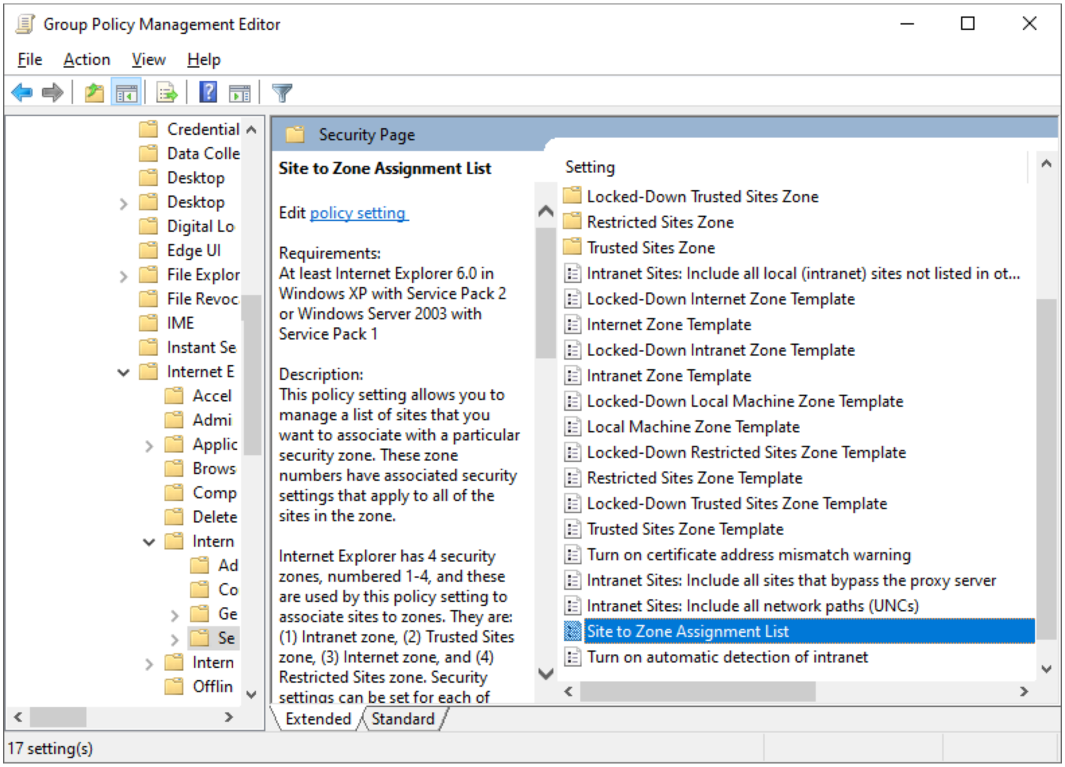
Configuring Azure AD Connect with PTA and Seamless SSO consists of the following high-level steps:

1. Adding the Azure AD Authentication service to the intranet sites
2. Configuring Azure AD Connect

Adding the Azure AD Authentication service to the intranet sites

Perform these steps to add the Azure AD Authentication service to Internet Explorer's and Edge's **Intranet Sites** in a **Group Policy Object** (**GPO**) that is targeted to the same **Organizational Unit** (**OU**) filtering scope as in Azure AD Connect:

1. Sign in to a system that has the Group Policy Management feature installed.
2. Press Start.
3. Search for **Group Policy Management**, and click its search result or run `gpmc.msc`. The **Group Policy Management** window appears.
4. In the left-hand navigation pane, navigate to the **Group Policy Objects** node.
5. If your organization already manages the intranet sites through Group Policy, locate the corresponding GPO and select it. Alternatively, right-click the **Group Policy Objects** node and select **New** from the menu. When creating a new GPO, provide a name for the new GPO and click **OK**.
6. Right-click the GPO and select **Edit…** from the menu. The **Group Policy Management Editor** window appears.
7. In the left-hand navigation pane, navigate to **User Configuration**, **Policies**, **Administrative Templates**, **Windows Components**, **Internet Explorer**, **Internet Control Panel**, and, lastly, **Security Page**.
8. On the main pane, select the **Site to Zone Assignment List** setting:



Site to Zone Assignment List

Edit policy setting

Requirements:
At least Internet Explorer 6.0 in Windows XP with Service Pack 2 or Windows Server 2003 with Service Pack 1

Description:
This policy setting allows you to manage a list of sites that you want to associate with a particular security zone. These zone numbers have associated security settings that apply to all of the sites in the zone.

Internet Explorer has 4 security zones, numbered 1-4, and these are used by this policy setting to associate sites to zones. They are: (1) Intranet zone, (2) Trusted Sites zone, (3) Internet zone, and (4) Restricted Sites zone. Security settings can be set for each of

Setting
- Locked-Down Trusted Sites Zone
- Restricted Sites Zone
- Trusted Sites Zone
- Intranet Sites: Include all local (intranet) sites not listed in ot...
- Locked-Down Internet Zone Template
- Internet Zone Template
- Locked-Down Intranet Zone Template
- Intranet Zone Template
- Locked-Down Local Machine Zone Template
- Local Machine Zone Template
- Locked-Down Restricted Sites Zone Template
- Restricted Sites Zone Template
- Locked-Down Trusted Sites Zone Template
- Trusted Sites Zone Template
- Turn on certificate address mismatch warning
- Intranet Sites: Include all sites that bypass the proxy server
- Intranet Sites: Include all network paths (UNCs)
- Site to Zone Assignment List
- Turn on automatic detection of intranet

Figure 14.4 – The Site to Zone Assignment List setting

9. Double-click the **Site to Zone Assignment List** setting.
10. Select **Enabled**.
11. Click the **Show…** button. The **Show Contents** window will appear.
12. In the **Value name** field, enter the following URL:

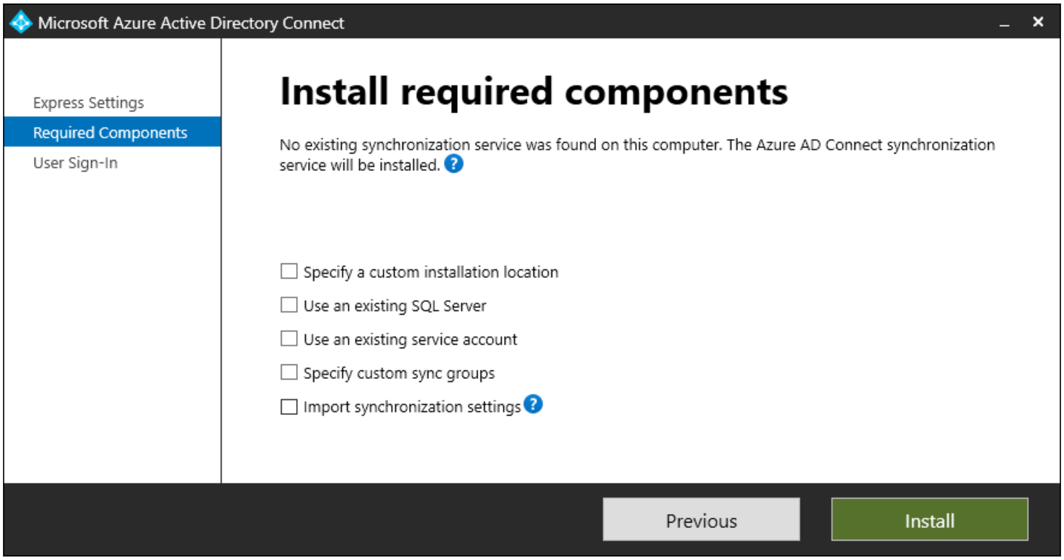
```
https://autologon.microsoftazuread-sso.com
```

13. In the **value** field, assign **1**, as this value corresponds to the **Intranet Sites** zone.
14. Click **OK** to save the settings, and close the **Show Contents** window.
15. Click **OK** to close the Group Policy setting.
16. Close the **Group Policy Management Editor** window.
17. After you have created a new GPO, in the left-hand navigation pane of the **Group Policy Management** window, navigate to the OU where you want to link the GPO. Right-click the OU and select **Link an existing GPO…** from the menu. In the **Select GPO window**, select the GPO and click **OK**.

Configuring Azure AD Connect

Perform the following steps on the server running Azure AD Connect:

1. Double-click `AzureADConnect.msi`. The **Microsoft Azure Active Directory Connect** window appears.
2. On the **Welcome** screen, select the **I agree to the license terms and privacy notice.** option.
3. Click **Next**.
4. On the **Express Settings** screen, click **Customize**. You are presented with the **Install required components** screen:


Microsoft Azure Active Directory Connect

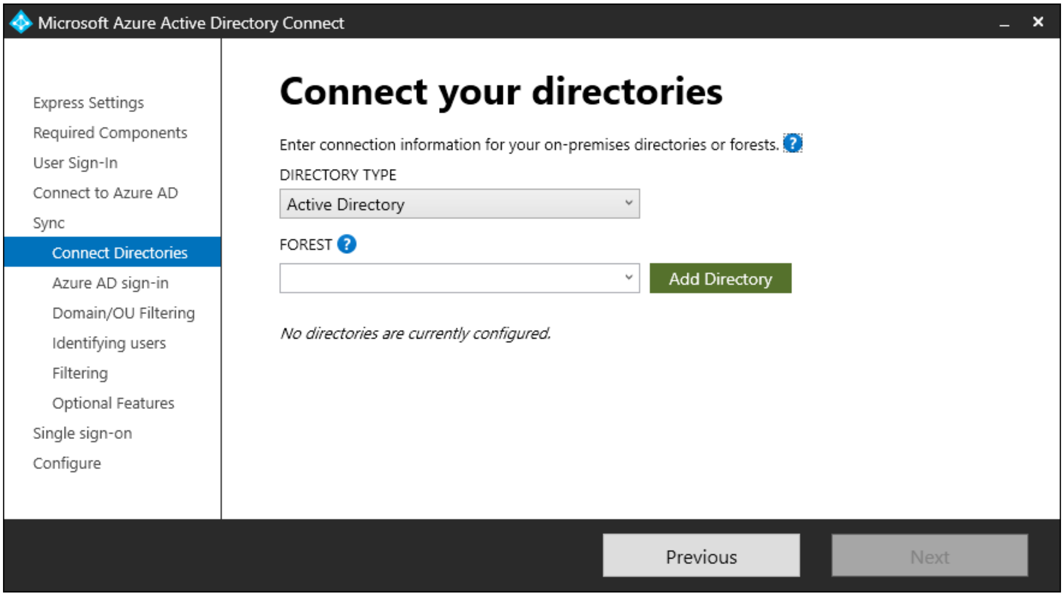
Install required components

No existing synchronization service was found on this computer. The Azure AD Connect synchronization service will be installed.

□ Specify a custom installation location
□ Use an existing SQL Server
□ Use an existing service account
□ Specify custom sync groups
□ Import synchronization settings

Figure 14.5 – The Install required components screen

5. On the **Install required components** screen, click **Install**.
6. On the **User sign-in** screen, select the **Pass-through authentication** option followed by the **Enable single sign-on** option.
7. Click **Next**.
8. On the **Connect to Azure AD** screen, enter the credentials of an account in Azure AD that has been assigned the **Global administrator** role.
9. Click **Next**.
10. Perform multi-factor authentication when prompted. You are presented with the **Connect your directories** screen:



Connect your directories

Enter connection information for your on-premises directories or forests.

DIRECTORY TYPE: Active Directory

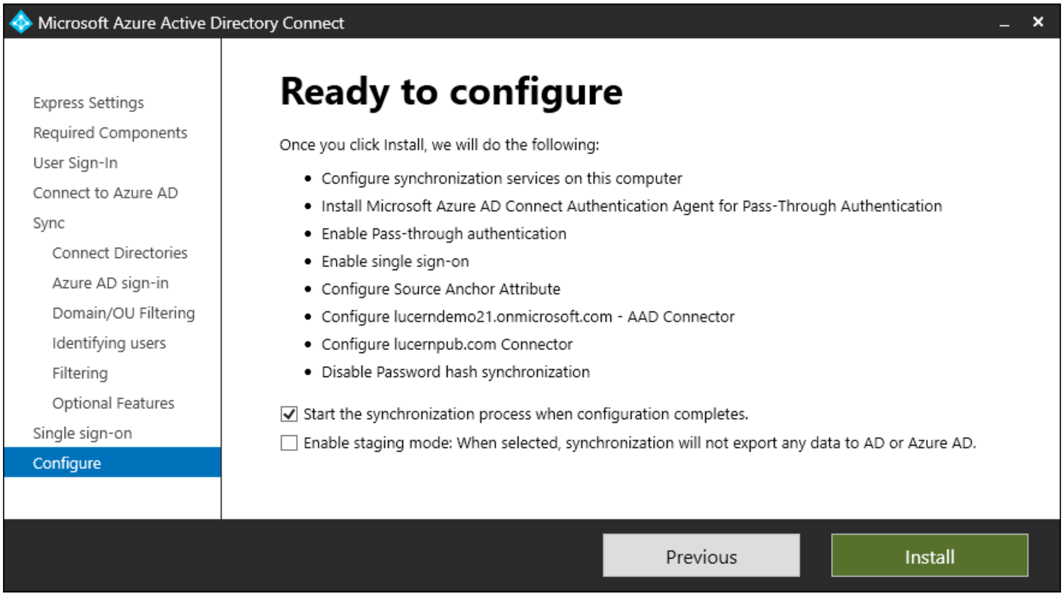
FOREST:

No directories are currently configured.

Figure 14.6 – The Connect your directories screen

11. On the **Connect your directories** screen, click **Add directory**. The **AD forest account** pop-up window will appear.
12. Ensure the **Create new AD account** option has been selected.
13. Specify the credentials of an account that is a member of the **Domain Admins** group for the Active Directory domain in which PTA is configured. Alternatively, specify an account that is a member of the **Enterprise Admins** group in the Active Directory forest that contains the domain in which PTA will be configured. This will create the account in Active Directory that Azure AD Connect will use to connect to the directory.
14. Click **OK**.
15. Back in the **Microsoft Azure Active Directory Connect** window, click **Next**.
16. On the **Azure AD Sign-in configuration** screen, click **Next**.
17. On the **Domain and OU Filtering** screen, click **Next**.
18. On the **Uniquely identifying your users** screen, click **Next**.
19. On the **Filter users and devices** screen, click **Next**.
20. On the **Optional features** screen, click **Next**.
21. On the **Enable single sign-on** screen, click the **Enter credentials** button. A **Windows Security** pop-up window appears to enter the credentials for the specified forest.

22. Specify the credentials of an account that is a member of the **Domain Admins** group for the Active Directory domain for which Seamless SSO will be configured. Alternatively, specify an account that is a member of the **Enterprise Admins** group in the Active Directory forest that contains the domain in which Seamless SSO will be configured.
23. Click **OK**.
24. Back in the **Microsoft Azure Active Directory Connect** windows, click **Next**. You are presented with the **Ready to configure** screen:



Ready to configure

Once you click Install, we will do the following:

- Configure synchronization services on this computer
- Install Microsoft Azure AD Connect Authentication Agent for Pass-Through Authentication
- Enable Pass-through authentication
- Enable single sign-on
- Configure Source Anchor Attribute
- Configure lucerndemo21.onmicrosoft.com - AAD Connector
- Configure lucernpub.com Connector
- Disable Password hash synchronization

☑ Start the synchronization process when configuration completes.

□ Enable staging mode: When selected, synchronization will not export any data to AD or Azure AD.

Figure 14.7 – The Ready to configure screen

25. On the **Ready to configure** screen, click **Install**.
26. On the **Configuration complete** screen, click **Exit** to close the **Microsoft Azure Active Directory Connect** window and start synchronization with Azure AD.

How it works...

When enabling PTA in Azure AD Connect, the Windows Server running Azure AD Connect gets the PTA agent installed and configured.

The PTA agent picks up authentication requests from the Azure Service Bus, processes them, and puts the authentication results back on the Azure Service Bus. This way, the PTA agent merely requires an outbound HTTPS connection.

To allow Seamless SSO, people using domain-joined devices authenticate to the Azure AD Authentication Service with their domain accounts using Kerberos. By default, Kerberos is not available for all websites in a browser for security reasons. The Azure AD Authentication service needs to be added to the **Intranet Sites** list of Internet Explorer and Edge to allow it.

Mozilla Firefox does not automatically allow Kerberos authentication. Add the `https://autologon.microsoftazuread-sso.com` URL to `network.negotiate-auth.trusted-URI's` on Firefox's `about:config` settings page.

Google Chrome does not automatically allow Kerberos authentication. Add `https://autologon.microsoftazuread-sso.com` to the `AuthNegotiateDelegateWhitelist` or `AuthServerWhitelist` settings.

There's more...

Install any additional PTA agents on additional Windows Server installations. Microsoft recommends that you have at least two additional PTA agents.

Implementing SSO using AD FS

This recipe shows how to configure an AD FS farm, consisting of one AD FS server and one publicly available Web Application Proxy for SSO.

Getting ready

While the recipes in *Chapter 13*, *Managing Federation*, showed how to build an AD FS farm, for this recipe, we'll use the built-in capability of Azure AD Connect to configure two Windows Server 2022 installations as an AD FS server and Web Application Proxy, respectively.

You'll need one domain-joined Windows Server installation running Windows Server 2016 or a newer version of Windows Server to install Azure AD Connect. Ensure this Windows Server can communicate with the internet without having to pass proxies, is domain-joined, and has IE ESC turned off. If proxies need passing, take the appropriate measures by making a proxy exception or configuring a proxy for Azure AD Connect in its configuration file.

For this recipe, you'll need two domain-joined Windows Server installations, running Windows Server 2012 R2 or newer versions of Windows Server:

- One AD FS server that will use a **Windows Internal Database** (**WID**) to host the AD FS configuration database
- One Web Application Proxy

The two Windows Server installations need to be resolvable and reachable on TCP port `5985` by the Windows Server you'll use for the Azure AD Connect installation.

To configure the AD FS farm, the following prerequisites need to be met:

- The AD FS farm name needs to be resolvable to the AD FS server in the internal DNS zone.
- The AD FS farm name needs to be resolvable to the Web Application Proxy in the public DNS zone.
- A publicly valid TLS certificate needs to be available for the AD FS farm name as a file in a password-protected `*.pfx` file, in addition to the private key.

Ensure the accounts in the on-premises Active Directory that are in scope for synchronization with Azure AD are configured with a publicly routable `userPrincipalName` suffix, such as `lucernpub.com`. Ensure the DNS domain name(s) representing the UPN suffix(es) are owned by your organization on the internet and are configured as verified DNS domain name(s) in your organization's Azure AD tenant.

Download the latest version of Azure AD Connect from `https://www.microsoft.com/en-us/download/details.aspx?id=47594`.

To connect Active Directory to Azure AD and set up the AD FS farm, you'll need to sign in with an account that is a member of the local **Administrators** group on the server dedicated to Azure AD Connect. As part of the steps of this recipe, you'll need to enter the credentials for the following accounts:

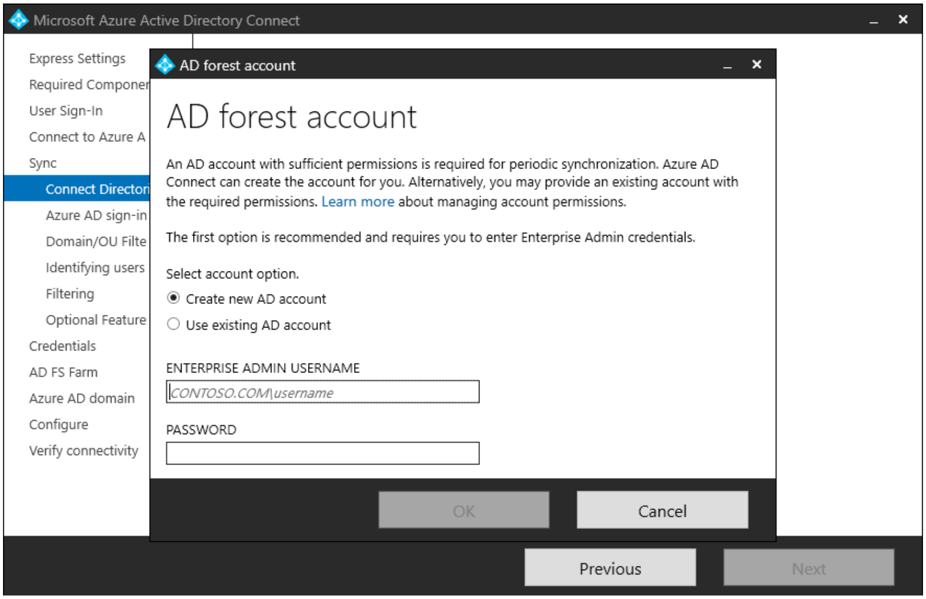
- An account in Active Directory that is a member of the **Enterprise Admins** group
- An account in Azure AD that has the **Global administrator** role or the **Hybrid Identity administrator** role assigned

If the organization uses the Azure AD PIM feature, activate the **Global administrator** role or the **Hybrid Identity administrator** role in advance.

How to do it...

Perform the following steps on the Windows Server you have dedicated to Azure AD Connect:

1. Double-click `AzureADConnect.msi`. The **Microsoft Azure Active Directory Connect** screen will appear.
2. On the **Welcome** screen, select the **I agree to the license terms and privacy notice.** option.
3. Click **Continue**.
4. On the **Express Settings** screen, click **Customize**.
5. On the **Install required components** screen, click **Install**.
6. On the **User Sign-in** screen, select **Federation with AD FS** as the sign-in method.
7. Click **Next**.
8. On the **Connect to Azure AD** screen, enter the credentials of an account in Azure AD that has been assigned the **Global administrator** role.
9. Click the **Next** button.
10. When prompted, perform multi-factor authentication.
11. On the **Connect your directories** screen, click **Add directory**. The **AD forest account** pop-up window appears:



AD forest account

An AD account with sufficient permissions is required for periodic synchronization. Azure AD Connect can create the account for you. Alternatively, you may provide an existing account with the required permissions. Learn more about managing account permissions.

The first option is recommended and requires you to enter Enterprise Admin credentials.

Select account option.

● Create new AD account

○ Use existing AD account

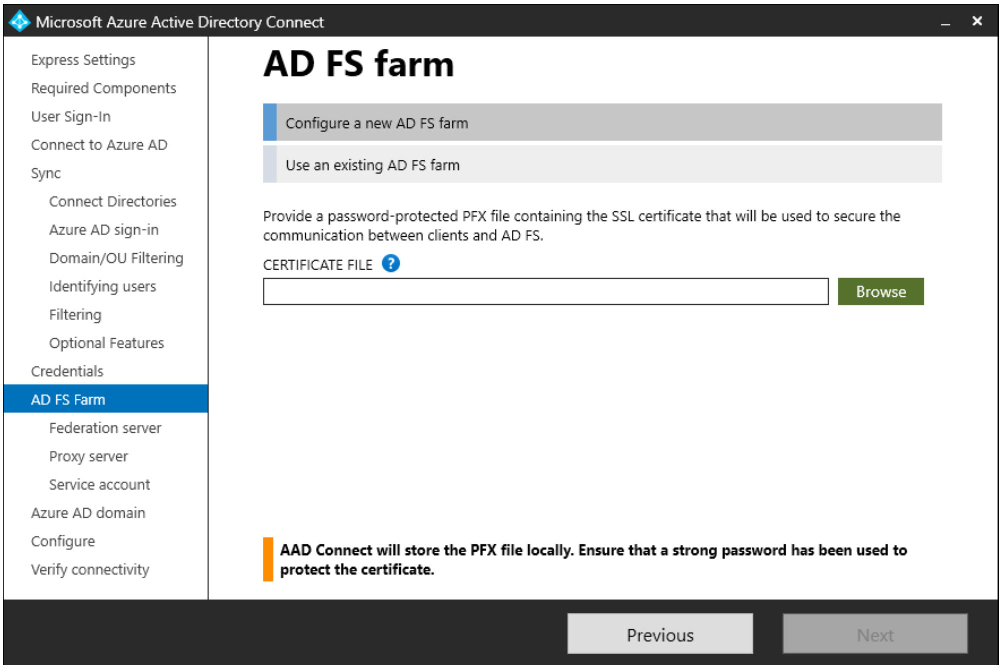
ENTERPRISE ADMIN USERNAME

CONTOSO.COM\username

PASSWORD

Figure 14.8 – The AD forest account pop-up window

12. Ensure the **Create new AD account** option has been selected.
13. Enter the credentials of an account that is a member of the **Enterprise Admins** group in the Active Directory forest you want to add.
14. When done, click **OK**.
15. Back in the **Microsoft Azure Active Directory Connect** screen, click **Next**.
16. On the **Azure AD sign-in configuration** screen, click **Next**.
17. On the **Domain and OU filtering** screen, click **Next**.
18. On the **Uniquely identifying your users** screen, click **Next**.
19. On the **Filter users and devices** screen, click **Next**.
20. On the **Optional features** screen, click **Next**.
21. On the **Domain administrator** screen, enter the credentials of an account that is a member of the **Domain Admins** group for the Active Directory domain in which AD FS will be deployed or configured. Alternatively, enter the credentials of an account that is a member of the **Enterprise Admins** group in the Active Directory forest that contains the domain in which the AD FS server will be deployed or configured.
22. Click **Next**. You reach the **AD FS farm** screen:



Microsoft Azure Active Directory Connect

AD FS farm

Configure a new AD FS farm

Use an existing AD FS farm

Provide a password-protected PFX file containing the SSL certificate that will be used to secure the communication between clients and AD FS.

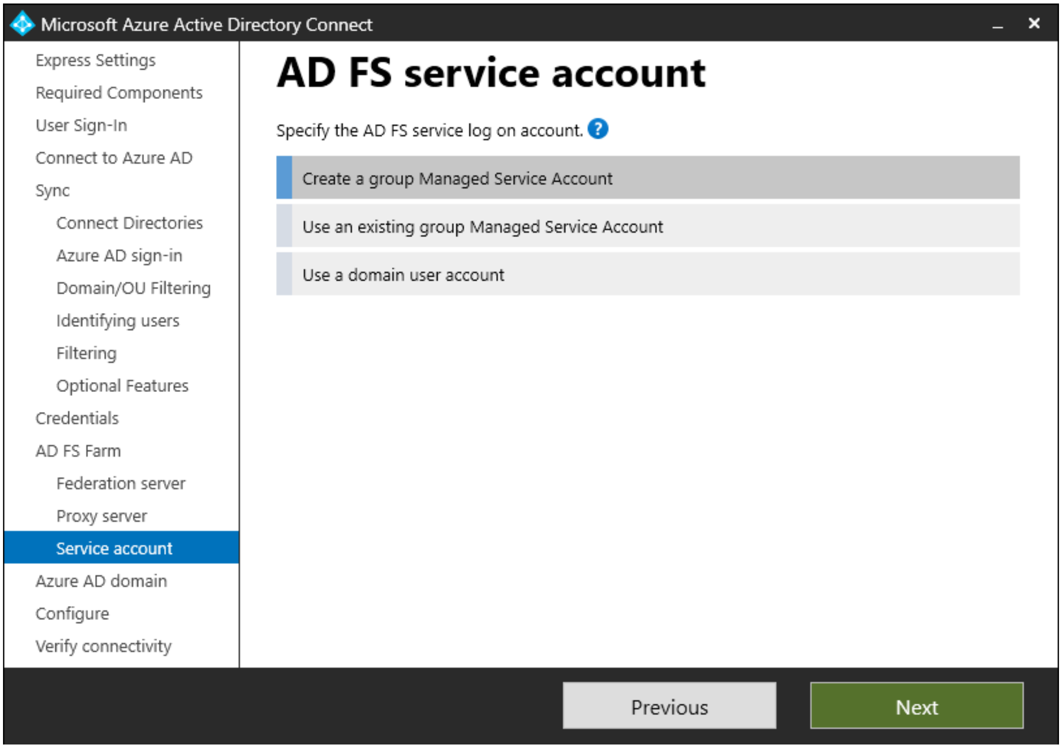
CERTIFICATE FILE

AAD Connect will store the PFX file locally. Ensure that a strong password has been used to protect the certificate.

Figure 14.9 – The AD FS farm screen

23. On the **AD FS farm** screen, select the **Configure a new AD FS farm** ribbon.
24. Ensure the **Provide a password-protected PFX certificate file** option has been selected.
25. Click the **Browse** button. The **Open** window appears.
26. Navigate to the file location of the certificate file, and click **Open** to select it. The **Open** window closes, and the **Password** pop-up window appears.
27. In the **Password** pop-up window, enter the password for the PFX file.
28. Click **OK**.
29. Back in the **Microsoft Azure Active Directory Connect** screen, select the correct **Subject name** instance from the available subject names in the certificate.
30. Click **Next**.
31. On the **AD FS servers** screen, type the fully qualified domain name(s) of the domain-joined Windows Server you dedicated as the AD FS server in the AD FS farm. Click the **Add** button to add the AD FS server to the AD FS farm.
32. Click **Next**.
33. On the **Web application proxy servers** screen, type the fully qualified domain name of the domain-joined Windows Server you dedicated as the Web Application Proxy server in the AD FS farm. Click the **Add** button to add the Web Application Proxy servers to the AD FS farm.

34. Click **Next**. You reach the **AD FS service account** screen:



AD FS service account

Specify the AD FS service log on account.

Create a group Managed Service Account

Use an existing group Managed Service Account

Use a domain user account

Figure 14.10 – The AD FS service account screen

35. On the **AD FS service account** screen, select the **Create a group Managed Service Account** ribbon.
36. Click **Next**.
37. On the **Azure AD domain** screen, select the DNS domain name that is verified in Azure AD and represents the `userPrincipalName` suffix of the user objects in scope.
38. Click **Next**.
39. On the **Ready to configure** screen, click **Install**.
40. On the **Configuration complete** screen, click **Next** to continue to verify the federation settings.
41. On the **Verify federation connectivity** screen, ensure that the **I have created DNS A records or DNS AAAA records that allow clients to resolve my federation service from the intranet.** and **I have created DNS A records that allow clients to resolve my federation service from the extranet.** options have been selected.

42. Click **Verify**.
43. On the **Verify federation connectivity** screen, click **Exit** to close the **Microsoft Azure Active Directory Connect** window and start synchronization with Azure AD.

How it works...

When the network ports for WMI Management (TCP `5985`) are open between the proposed AD FS server and the Web Application Proxy server, Azure AD Connect can automatically configure the hosts with their role and the required configuration to federate with Azure AD.

First, Azure AD Connect connects to the Azure AD tenant using the credentials supplied by the **Global administrator** account. Then, the Active Directory forest is connected, using the credentials of the account that is a member of the Enterprise Admins group.

As AD FS is selected as the sign-in method, the Microsoft Azure AD Connect Configuration wizard presents the flow to configure the AD FS farm, consisting of one AD FS server and one Web Application Proxy.

The AD FS farm will be configured with the following details:

- WID to store and replicate the AD FS Configuration database
- A **group Managed Service Account** (**gMSA**) as the AD FS service account
- A **relying party trust** (**RPT**) between the AD FS farm and Azure AD

After configuration, Azure AD Connect verifies the name resolution of the AD FS farm, using both the internal and public DNS zones.

There's more...

When using the recipes in *Chapter 13*, *Managing Federation*, an AD FS farm can be tailored to the organization's specific needs. For instance, the Web Application Proxy server can be standalone instead of domain-joined.

When reusing an existing AD FS farm with Azure AD Connect, specify **Use an existing AD FS farm** instead of **Configure a new AD FS farm** in *step 23*. Instead of setting up an AD FS farm, the ensuing steps validate the AD FS farm and configure the RPT.

Managing AD FS with Azure AD Connect

This recipe explores how to manage an AD FS farm with Azure AD Connect.

Getting ready

For this recipe, you'll need the following:

- A properly configured AD FS farm running Windows Server 2012 R2 or a newer version of Windows Server
- A properly configured Azure AD Connect installation, capable of communicating with the AD FS server(s) and Web Application Proxy server(s) in the AD FS farm using TCP `5985`

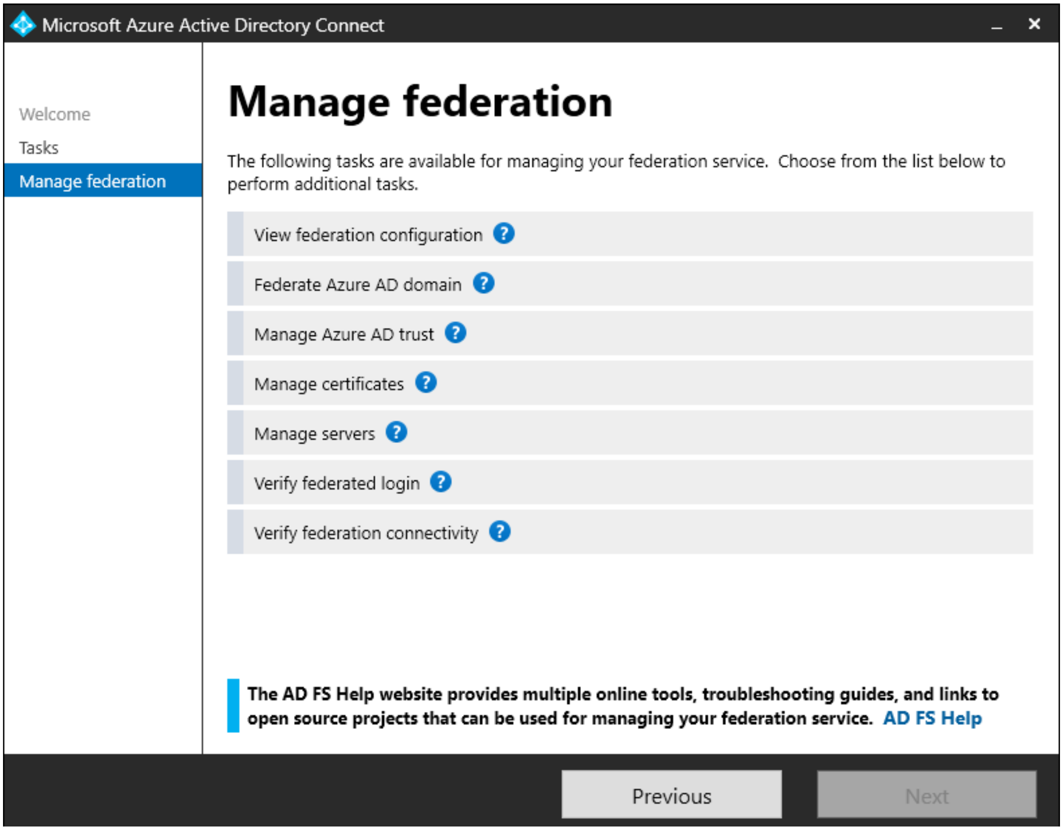
Sign in to the Windows Server installation with Azure AD Connect with an account that is a member of the local **Administrators** group.

If the organization uses the Azure AD PIM feature, activate the **Global administrator** role or the **Hybrid Identity administrator** role in advance.

How to do it...

First, perform the following steps:

1. Open **Azure AD Connect** from the desktop.
2. In the **Welcome to Azure AD Connect** screen, click **Configure**.
3. In the **Additional tasks** screen, select the **Manage federation** ribbon.
4. Click **Next**. You are presented with the **Manage federation** screen:



Manage federation

The following tasks are available for managing your federation service. Choose from the list below to perform additional tasks.

- View federation configuration
- Federate Azure AD domain
- Manage Azure AD trust
- Manage certificates
- Manage servers
- Verify federated login
- Verify federation connectivity

The AD FS Help website provides multiple online tools, troubleshooting guides, and links to open source projects that can be used for managing your federation service. AD FS Help

Figure 14.11 – The Manage federation screen

5. Choose between the available actions.

Perform these steps before each action that is shown in the following subsections.

Resetting the Azure AD trust

To reset the RPT between the AD FS farm and Azure AD, perform the following steps:

1. Select the **Manage Azure AD trust** ribbon.
2. Click **Next**.
3. On the **Connect to Azure AD** screen, enter the credentials of an account in Azure AD that has been assigned the **Global administrator** role or the **Hybrid Identity administrator** role.
4. Perform multi-factor authentication when prompted.
5. On the **Azure AD trust management tasks** screen, select the **Reset Azure AD and AD FS trust** option.
6. Click **Next**.
7. On the **Overview of task** screen, click **Next**.
8. On the **Connect to AD FS** screen, enter the credentials for an account that is a member of the local **Administrators** group on the primary AD FS server or a member of the **Domain Admins** group in the domain.
9. On the **Ready to configure** screen, click **Configure**.
10. On the **Configuration complete** screen, click **Exit** to close the **Microsoft Azure Active Directory Connect** window and restart synchronization.

Federating an additional Azure AD domain

To add a verified DNS domain name in Azure AD as a federated domain in AD FS, perform the following steps:

1. Select the **Federate Azure AD domain** ribbon.
2. Click **Next**.
3. On the **Connect to Azure AD** screen, enter the credentials of an account in Azure AD that has been assigned the **Global administrator** role or the **Hybrid Identity administrator** role.
4. On the **Connect to AD FS** screen, enter the credentials for an account that is a member of the local **Administrators** group on the primary AD FS server or a member of the **Domain Admins** group in the domain.

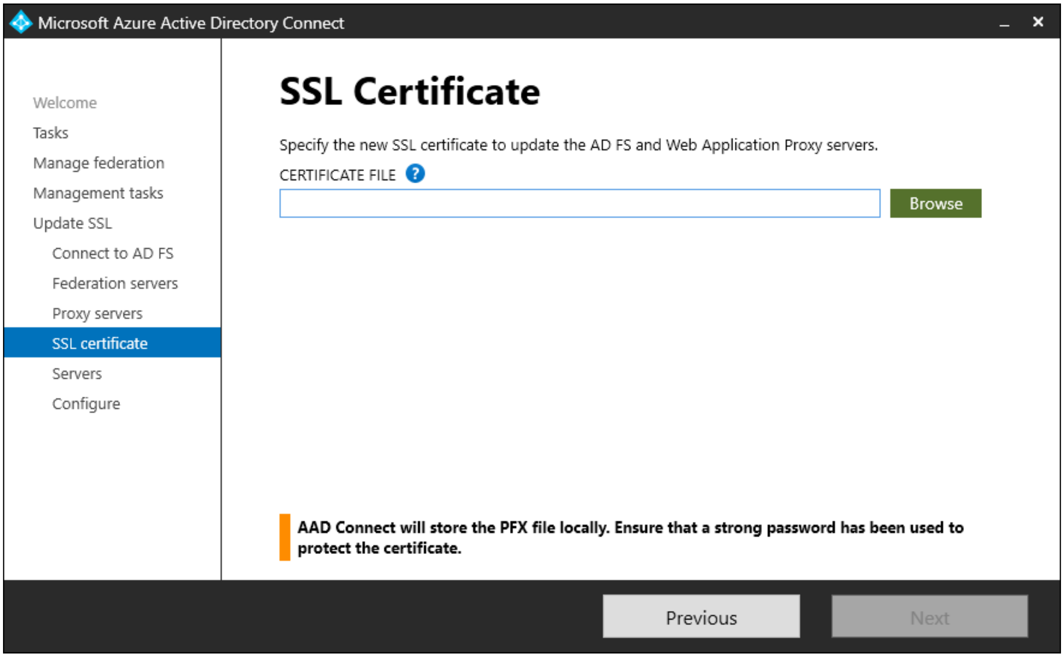
5. On the **Azure AD Domain** screen, select the additional Azure AD domain to federate with the Azure AD Connect-managed AD FS farm.
6. Click **Next**.
7. On the **Ready to Configure** screen, click **Configure**.
8. On the **Configuration complete** screen, click **Exit** to close the **Microsoft Azure Active Directory Connect** window and restart synchronization.

Updating the AD FS SSL certificate

To update the AD FS service communication certificate, with a few simple clicks, on all AD FS servers and Web Application Proxy servers in the Azure AD Connect-managed AD FS farm, perform the following steps:

1. Select the **Manage certificates** ribbon.
2. Click **Next**.
3. On the **Certificate management tasks** screen, select the **Update SSL certificate** option.
4. Click **Next**.
5. On the **Connect to AD FS servers** screen, enter the credentials for an account that is a member of the local **Administrators** group on the primary AD FS server or a member of the **Domain Admins** group in the domain.
6. On the **AD FS servers** screen, check that all of the AD FS servers are present. Use the **Add** button to add any missing AD FS servers. Use the **Remove** link next to the AD FS servers that are offline or no longer part of the AD FS farm.
7. Click **Next**.
8. On the **Web Application Proxy servers** screen, check that all Web Application Proxy servers are present. Use the **Add** button to add missing Web Application Proxies. Use the **Remove** link next to Web Application Proxies that are offline or no longer part of the AD FS farm.

9. Click **Next**. You are presented with the SSL Certificate screen:



SSL Certificate

Specify the new SSL certificate to update the AD FS and Web Application Proxy servers.

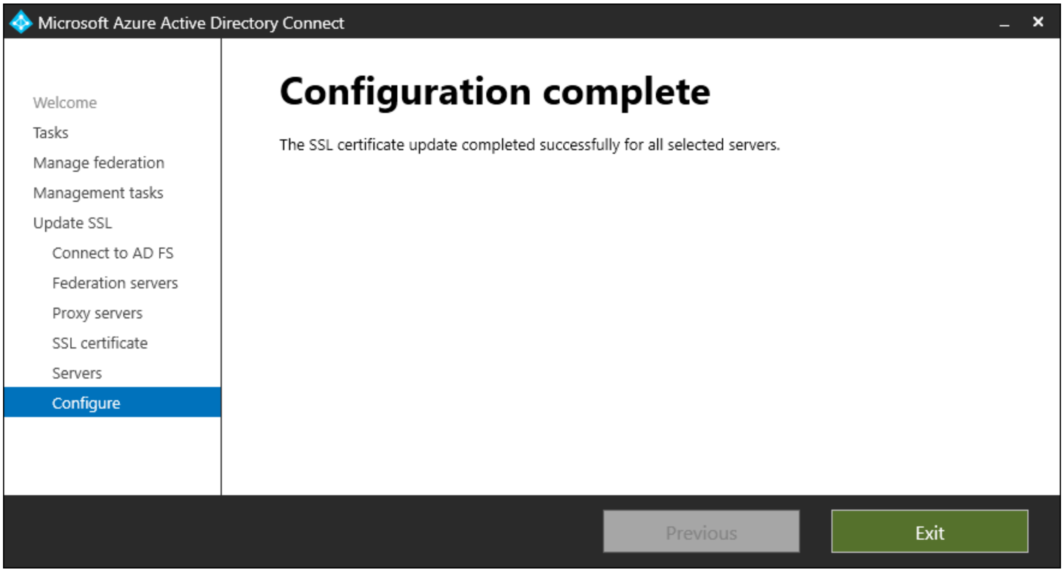
CERTIFICATE FILE

AAD Connect will store the PFX file locally. Ensure that a strong password has been used to protect the certificate.

Figure 14.12 – The SSL Certificate screen

10. On the **SSL Certificate** screen, click the **Browse** button. The **Open** window appears.
11. Navigate to the file location of the certificate file, and click **Open** to select it. The **Open** window closes, and the **Password** pop-up window appears.
12. In the **Password** pop-up window, enter the password for the PFX file.
13. Click **OK**.
14. Back in the **Microsoft Azure Active Directory Connect** screen, click **Next**.
15. On the **Select servers for SSL certificate update** screen, check that all AD FS servers and Web Application Proxy servers that are connected to the AD FS farm have been selected.

16. Click **Next**.
17. On the **Ready to Configure** screen, click **Configure**. After configuration, the **Configuration complete** screen appears:



Configuration complete

The SSL certificate update completed successfully for all selected servers.

Figure 14.13 – The Configuration complete screen

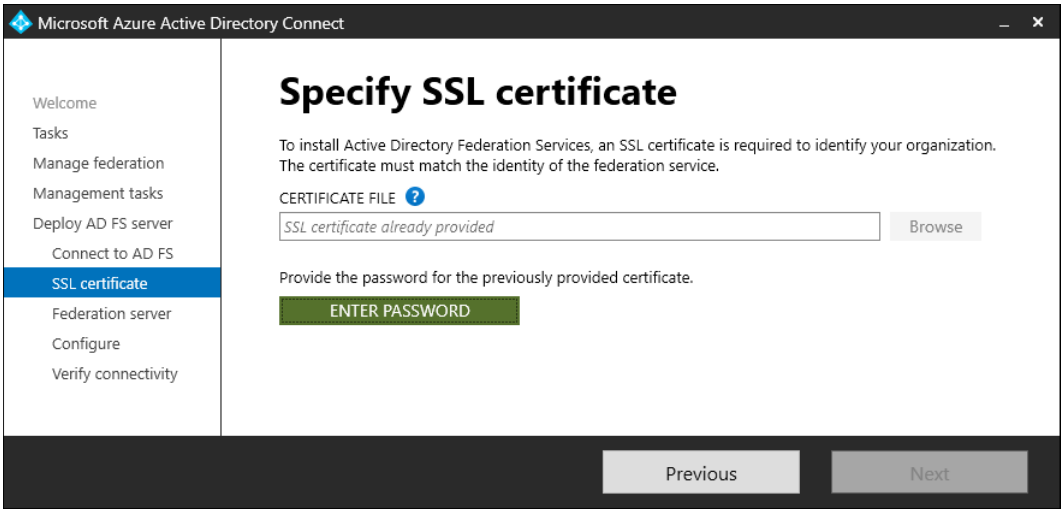
18. On the **Configuration complete** screen, click **Exit** to close the **Microsoft Azure Active Directory Connect** window and restart synchronization.

Deploying an additional AD FS server

To deploy an additional AD FS server to the Azure AD Connect-managed AD FS farm, perform the following steps:

1. Select the **Manage servers** ribbon.
2. Click **Next**.
3. On the **Server management tasks** screen, select the **Deploy an AD FS server** option.
4. Click **Next**.

5. On the **Connect to AD FS** screen, enter the credentials for an account that is a member of the local **Administrators** group on the primary AD FS server or a member of the **Domain Admins** group in the domain. You will reach the **Specify SSL certificate** screen:



Specify SSL certificate

To install Active Directory Federation Services, an SSL certificate is required to identify your organization. The certificate must match the identity of the federation service.

CERTIFICATE FILE: *SSL certificate already provided*

Provide the password for the previously provided certificate.

Figure 14.14 – The Specify SSL certificate screen

6. On the **Specify SSL certificate** screen, click the **ENTER PASSWORD** button. The **Password** pop-up window appears.
7. Enter the password for the PFX file.
8. Click **OK**.
9. Back in the **Microsoft Azure Active Directory Connect** screen, click **Next**.
10. On the **AD FS Server** screen, enter the server name or IP address of the domain-joined Windows Server installation to be added as an AD FS server to the AD FS farm.
11. Click **Add**.
12. Click **Next**.
13. On the **Ready to Configure** screen, click **Configure**.
14. On the **Configuration complete** screen, click **Next** to continue to verify the federation settings.
15. On the **Verify federation connectivity**, ensure that both **I have created DNS A records or DNS AAAA records that allow clients to resolve my federation service from the intranet.** and **I have created DNS A records that allow clients to resolve my federation service from the extranet.** are selected.

16. Click **Verify**.
17. On the **Verify federation connectivity** screen, click **Exit** to close the **Microsoft Azure Active Directory Connect** window and restart synchronization.

Deploying an additional Web Application Proxy server

Perform the following steps to deploy an additional Web Application Proxy server to the Azure AD Connect-managed AD FS farm:

1. Select the **Manage servers** ribbon.
2. Click **Next**.
3. On the **Server management tasks** screen, select the **Deploy a Web Application Proxy server** option.
4. Click **Next**.
5. On the **Overview of Web Application Proxy** screen, click **Next**.
6. On the **Connect to AD FS** screen, enter the credentials for an account that is a member of the local **Administrators** group on the primary AD FS server or a member of the **Domain Admins** group in the domain.
7. On the **Specify SSL certificate** screen, click the **ENTER PASSWORD** button. The **Password** pop-up window appears.
8. Enter the password for the PFX file.
9. Click **OK**.
10. Back in the **Microsoft Azure Active Directory Connect** screen, click **Next**.
11. On the **Web Application Proxy server** screen, enter the server name or IP address of the domain-joined Windows Server installation to be added as a Web Application Proxy server to the AD FS farm.
12. Click **Add**.
13. Click **Next**.
14. On the **Ready to Configure** screen, click **Configure**.
15. On the **Configuration complete** screen, click **Next** to continue to verify the federation settings.
16. On the **Verify federation connectivity**, ensure that both **I have created DNS A records or DNS AAAA records that allow clients to resolve my federation service from the intranet.** and **I have created DNS A records that allow clients to resolve my federation service from the extranet.** are selected.

17. Click **Verify**.
18. On the **Verify federation connectivity** screen, click **Exit** to close the **Microsoft Azure Active Directory Connect** window and restart synchronization.

Verifying the federated login

To verify federated logins, perform the following steps:

1. Select the **Verify federated login** ribbon.
2. Click **Next**.
3. On the **Connect to Azure AD** screen, enter the credentials of an account in Azure AD that has been assigned the **Global administrator** role or the **Hybrid Identity administrator** role.
4. On the **Verify federated login** screen, enter the credentials of a federated user account.
5. Click **Verify**.
6. On the **Verify federated login** screen, click **Exit** to close the **Microsoft Azure Active Directory Connect** window and restart synchronization.

How it works...

Azure AD Connect can help administrators to manage AD FS farms. For this functionality, the AD FS farm does not need to have been set up using the **Microsoft Azure Active Directory Connect** wizard to be manageable with Azure AD Connect. This is because existing AD FS farms can also be onboarded to Azure AD Connect by using the **Use an existing AD FS farm** option when configuring **Federation with AD FS** as the sign-in method for one or more UPN suffixes in Active Directory.

First, the RPT information will be backed up for all actions that concern the RPT between Azure AD and AD FS.

Implementing Azure Traffic Manager for AD FS geo-redundancy

This recipe shows how to implement a geo-redundant AD FS deployment consisting of two AD FS servers and two Web Application Proxies, equally distributed over two geographically dispersed data centers.

Getting ready

For this recipe, we'll assume that an Active Directory domain exists with domain controllers in a networking environment consisting of two separate, geographically dispersed data centers. Each data center is defined as an Active Directory site. The traffic required for Active Directory replication is allowed, as is TCP `80` between the AD FS servers.

Perform the steps from the *Installing the AD FS server role* recipe in *Chapter 13*, *Managing Federation*, to install the AD FS server role on two Windows Server installations, each running in a separate, geographically dispersed data center. Perform the steps from the *Setting up an AD FS farm with Windows Internal Database* recipe in *Chapter 13*, *Managing Federation*, to set up an AD FS farm with WID on one of the two AD FS servers. Perform the steps from the *Adding additional AD FS servers to an AD FS farm* recipe in *Chapter 13*, *Managing Federation*, to add the second AD FS server. Perform the steps from the *Setting up a Web Application Proxy* recipe in *Chapter 13*, *Managing Federation*, to set up both Web Application Proxies.

Provide inbound access for both TCP `80` and TCP `443` to each Web Application Proxy, using separate IP addresses. Create DNS records for the public IP addresses of the two Web Application Proxies. For instance, when the AD FS farm URL is `sts.lucernpub.com`, dedicate `sts1.lucernpub.com` to the external IPv4 address of the first Web Application Proxy and `sts2.lucernpub.com` to the second.

How to do it...

Implementing geo-redundancy for AD FS consists of three steps:

1. Configuring the Web Application Proxy servers for probing
2. Configuring Azure Traffic Manager
3. Adding DNS records

Configuring the Web Application Proxy servers for probing

Run the following lines of Windows PowerShell on each of the Web Application Proxy servers to configure them for probing:

```
New-NetFirewallRule -Name Allow_HTTP -DisplayName "AD FS HTTP
Services" -Protocol TCP -Localport 80 -Profile Any -Action
Allow
```

This allows traffic using TCP `80` to the Web Application Proxy servers.

Configuring Azure Traffic Manager

To configure Azure Traffic Manager, perform the following steps:

1. Sign in to the Microsoft Azure Portal at `https://portal.azure.com` with an account that has sufficient permissions to create a resource group or add resources to an existing resource group.
2. If the account is associated with multiple tenants, choose the right tenant by clicking on the name or email address of the account in the upper-right corner of the Azure portal. Select the tenant you want to use from the bottom of the context menu.
3. Click the big green plus sign in the left-hand navigation menu to add products and services to your tenant.
4. In the **Search services and marketplace** field, search for **Traffic Manager Profile**. Then, select it.
5. Click **Create**. The **Create Traffic Manager profile** pane appears.
6. Type and select the following information:

    I. Type the name of your Traffic Manager profile, for instance, `LucernSTS`. This will be appended by `trafficmanager.net` to become the **Fully-qualified Domain Name** (**FQDN**) of the Traffic Manager profile.

    II. Select a **Routing method** option from the drop-down list.

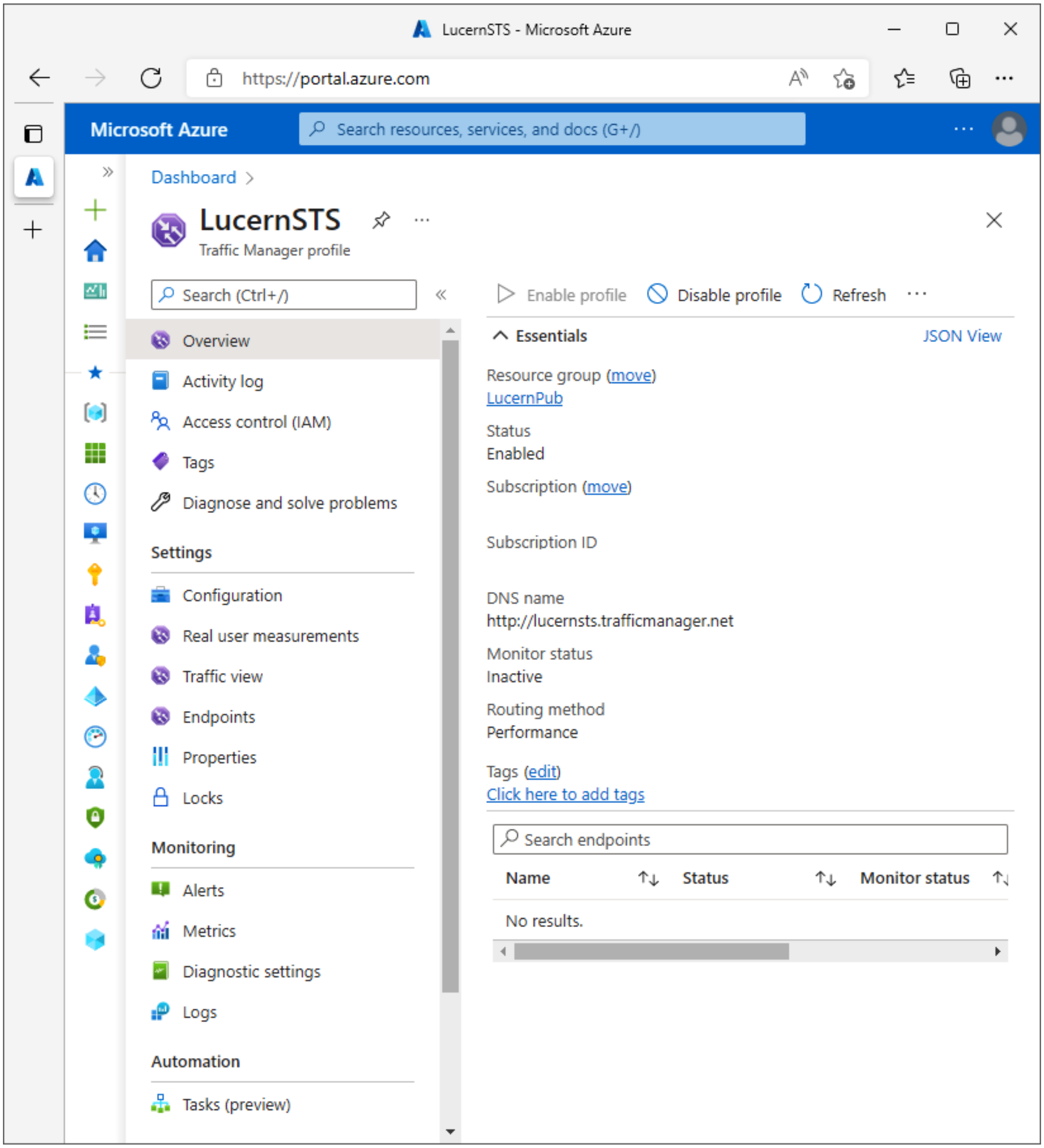
    III. Select a **Subscription** option from the drop-down list.

    IV. Create a new **Azure Resource Manager** (**ARM**) **Resource group**. If you already have a resource group for Traffic Manager profiles and other load-balancing/high-availability resources in the subscription, reuse that by selecting it from the drop-down list.

    V. Select a **Resource group location** from the drop-down list when creating a new resource group. Otherwise, continue with the next step.

7. Click **Create** at the bottom of the pane. You will be redirected back to the Azure Portal dashboard.

8. In the **Deployment succeeded** callout, click **Go to resource**. You are redirected to the **Overview** pane of the Traffic Manager profile:



Resource group (move)
LucernPub

Status
Enabled

Subscription (move)

Subscription ID

DNS name
http://lucernsts.trafficmanager.net

Monitor status
Inactive

Routing method
Performance

Tags (edit)
Click here to add tags

Search endpoints

Name	Status	Monitor status
No results.		

Figure 14.15 – The Overview pane of the Traffic Manager profile

9. In the left-hand navigation pane, click **Configuration** under **Settings**. The **Configuration** pane appears.

10. In the **Configuration** pane, click the **Path** field under **Endpoint monitor settings**. Change it to `/adfs/probe/`.
11. Click **Save** at the top of the blade.
12. In the left-hand navigation blade, click **Endpoints**.
13. Follow the **+ Add** link at the top of the pane. The **Add endpoint** blade appears.
14. From the **Type** drop-down list, select **External endpoint**.
15. Type something meaningful as the name.
16. For the FQDN, enter the DNS name you assigned to the external IP address of the load balancer or Web Application Proxy from the first location.
17. Select a **Location** option from the drop-down list. This determines the end user device locations to be directed to this endpoint.
18. Click **Add**.

Repeat *steps 13* to *18* to add the endpoint for the other Web Application Proxy.

Adding DNS records

Next, add a DNS CNAME record for the AD FS farm name to the Azure Traffic Manager URL in the external DNS zone. For instance, `sts.lucernpub.com` would point to `lucernsts.trafficmanager.net`.

How it works…

Azure Traffic Manager is an Azure-based user traffic load-balancing solution. It can direct user traffic between the IP addresses associated with the endpoints for Azure Virtual Machines, Azure services, and on-premises networks. Traffic Manager uses DNS to direct client requests to the most appropriate endpoint in its configuration, based on the traffic routing method of your choice and the health of the endpoints. It provides automatic failover.

Traffic Manager offers failover, performance, weighted, priority, geographic, multi-value, and subnet routing methods for client requests originating from outside the organization. Active Directory sites offer routing to the closest AD FS server inside the organization for client requests from inside the organization.

Azure Traffic Manager probes the `/adfs/probe/` URLs on the Web Application Proxy servers to determine whether the hosts are available to handle authentication requests. When Traffic Manager determines the endpoint can handle authentication requests, it provides the endpoint's DNS address to the device based on the routing method. If the endpoint is deemed degraded or is offline, Azure Traffic Manager does not return its DNS name to the device.

There's more...

This method to make AD FS geo-redundant is very cost-efficient, as Azure Traffic Manager will merely consume a small amount of money for probing endpoints (per endpoint) and redirects (per million requests). However, it does not offer resilience when an AD FS server is down, as the solution merely probes the Web Application Proxy servers' availability.

Migrating from AD FS to PTA for SSO to Office 365

This recipe shows how to change the sign-in method from federation with AD FS to PTA and Seamless SSO.

Getting ready

Ensure the organization has not implemented heavy customizations to the `onload.js` page of the AD FS sign-in pages or relies on on-premises multi-factor authentication solutions.

To configure the sign-in method within Azure AD Connect, you'll need to sign in with an account that is a local administrator on the server dedicated to Azure AD Connect. As part of the following steps, you'll need to enter the credentials for these accounts:

- An account in Active Directory that is a member of the **Enterprise Admins** group
- An account in Azure AD that has the **Global Administrator** role or the **Hybrid Identity Administrator** role assigned

Ensure the Windows Server running Azure AD Connect can communicate with the internet without having to pass proxies and has IE ESC turned off.

If the organization uses the Azure AD PIM feature, activate the **Global administrator** role or the **Hybrid Identity administrator** role in advance.

How to do it...

Migrating from AD FS to PTA consists of the following high-level steps:

- Adding the Azure AD Authentication service to the intranet sites zone
- Configuring Azure AD Connect
- Checking domains in the Azure Portal
- Deleting the **Microsoft Office 365 Identity Platform** RPT

Adding the Azure AD Authentication service to the intranet sites zone

Perform the following steps to add the Azure AD Authentication service to Internet Explorer's and Edge's **Intranet Sites** list in a new GPO that is targeted to the same OU filtering scope as in Azure AD Connect:

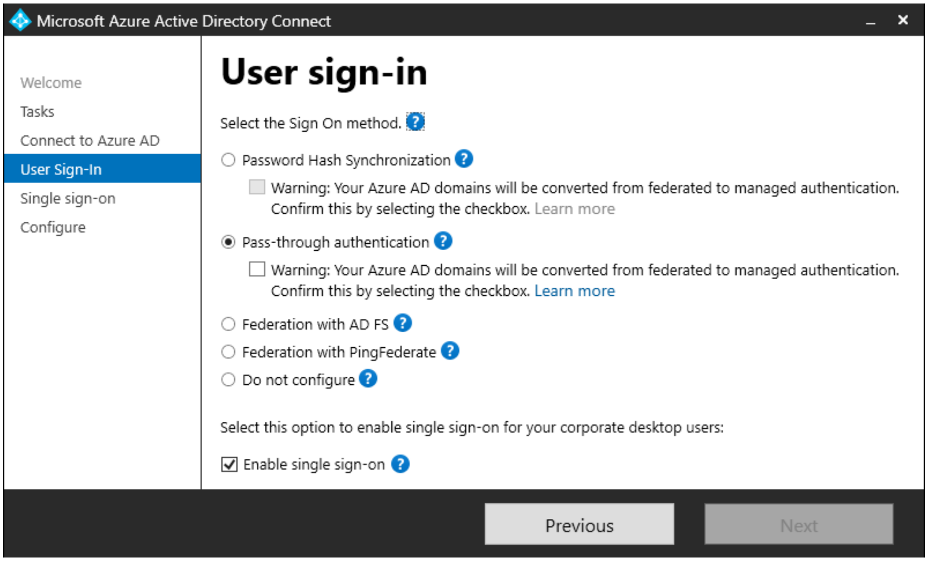
1. Sign in to a system with the Group Policy Management feature installed.
2. Press Start.
3. Search for **Group Policy Management**, and click its search result or run `gpmc.msc`. The **Group Policy Management** window appears.
4. In the left-hand navigation pane, navigate to the **Group Policy Objects** node.
5. If your organization already manages the list of intranet sites through Group Policy, locate the corresponding GPO and select it. Alternatively, right-click the **Group Policy Objects** node and select **New** from the menu.
6. Right-click the GPO, and select **Edit…** from the menu. The **Group Policy Management Editor** window appears.
7. In the left-hand navigation pane, navigate to **User Configuration**, then **Policies**, then **Administrative Templates**, **Windows Components**, **Internet Explorer**, **Internet Control Panel**, and, lastly, **Security Page**.
8. In the main pane, select the **Site to Zone Assignment List** setting.
9. Double-click the **Site to Zone Assignment List** setting.
10. Select **Enabled**.
11. Click the **Show…** button. The **Show Contents** window appears.
12. In the **Value name** field, enter the following URL: `https://autologon.microsoftazuread-sso.com`.
13. In the **value** field, assign **1**, as this value corresponds to the **Intranet Sites** zone.

14. Click **OK**.
15. Click **OK** to close the Group Policy setting.
16. Close the **Group Policy Management Editor** window.
17. When you have created a new GPO, in the left-hand navigation pane of the **Group Policy Management** window, navigate to the OU where you want to link the GPO. Right-click the OU and select **Link an existing GPO…** from the menu.

Configuring Azure AD Connect

Perform the following steps on the server running Azure AD Connect:

1. Open **Azure AD Connect** from the desktop.
2. In the **Welcome to Azure AD Connect** screen, click **Configure**.
3. In the **Additional tasks** screen, select the **Change user sign-in** ribbon.
4. Click **Next**.
5. On the **Connect to Azure AD** screen, enter the credentials of an account in Azure AD that has been assigned the **Global administrator** role.
6. Click **Next**. The **User sign-in** screen appears.
7. On the **User sign-in** screen, select the **Pass-through authentication** option. Additionally, you'll notice the **Enable single sign-on** option has been automatically selected, too:



User sign-in

Select the Sign On method.

○ Password Hash Synchronization

□ Warning: Your Azure AD domains will be converted from federated to managed authentication. Confirm this by selecting the checkbox. Learn more

● Pass-through authentication

□ Warning: Your Azure AD domains will be converted from federated to managed authentication. Confirm this by selecting the checkbox. Learn more

○ Federation with AD FS

○ Federation with PingFederate

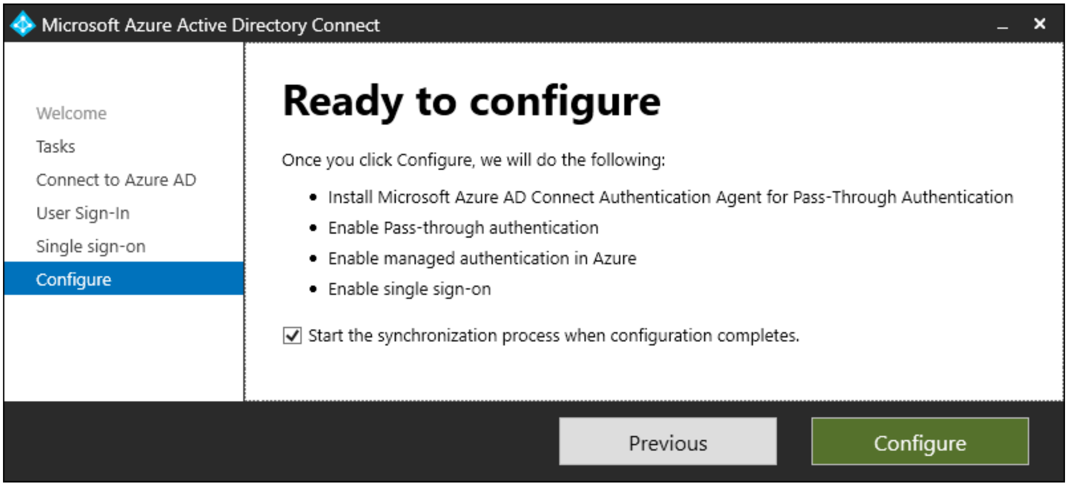
○ Do not configure

Select this option to enable single sign-on for your corporate desktop users:

☑ Enable single sign-on

Figure 14.16 – The User sign-in screen

8. Select the **Warning: Your Azure AD domains will be converted from federated to managed authentication. Confirm this by selecting the checkbox.** option.
9. Click **Next**.
10. On the **Enable single sign-on** screen, click the **Enter credentials** button. A **Windows Security** pop-up window appears, allowing you to enter the credentials of the specified forest.
11. Enter the credentials of an account that is a member of the **Domain Admins** group for the Active Directory domain for which Seamless SSO will be configured, or an account that is a member of the **Enterprise Admins** group in the Active Directory forest, which contains the domain in which Seamless SSO will be configured.
12. Click **OK**.
13. Back in the **Microsoft Azure Active Directory Connect** window, click **Next**. The **Ready to configure** screen appears:



Ready to configure

Once you click Configure, we will do the following:

- Install Microsoft Azure AD Connect Authentication Agent for Pass-Through Authentication
- Enable Pass-through authentication
- Enable managed authentication in Azure
- Enable single sign-on

☑ Start the synchronization process when configuration completes.

Figure 14.17 – The Ready to configure screen

14. On the **Ready to configure** screen, click **Configure**.
15. On the **Configuration complete** screen, click **Exit** to close the **Microsoft Azure Active Directory Connect** window and restart synchronization.

Checking domains in the Azure portal

Perform the following steps to check whether all federated DNS domain names have successfully migrated to PTA:

1. Navigate a browser to `https://aad.portal.azure.com`.
2. Sign in with an account that has the **Global administrator**, **Hybrid Identity administrator**, or **Domain Name administrator** roles assigned to it.
3. Perform multi-factor authentication when prompted.
4. In the left-hand navigation pane, click **Azure Active Directory**.
5. In the Azure AD navigation pane, click **Azure AD Connect**.
6. In the main pane for Azure AD Connect, under **User sign-in**, you should see that 0 domains use **Federation**, that all custom DNS domain names use **Seamless single sign-on**, and that there is one **Pass-through Authentication** agent.

Deleting the Office 365 Identity Platform RPT

Perform the following steps to disable federation on the AD FS side by deleting the **Microsoft Office 365 Identity Platform** RPT:

1. Sign in to the AD FS server with an account that is a member of the **Domain Admins** group.
2. Open **AD FS Management** (`Microsoft.IdentityServer.msc`).
3. In the left-hand navigation pane, under the **AD FS** node, expand the **Relying Party Trusts** node.
4. In the main pane, select the **Microsoft Office 365 Identity Platform** RPT.
5. In the right-hand **Actions** pane, click **Delete**, or right-click the RPT and select **Delete** from the menu.
6. In the **AD FS Management** pop-up window, click **Yes** as the answer to the question, **Are you sure you want to delete this item?**.

How it works...

PTA offers a sign-in method with less overhead. For organizations that adopted Microsoft 365 services early on, the migration from the federation with AD FS as the authentication method to PTA with Seamless SSO as the authentication method is clear.

However, if your organization has heavy customizations in the `onload.js` file of the AD FS sign-in pages, the Azure AD sign-in pages might not fit your organization's needs. If your organization has implemented on-premises multi-factor authentication solutions, expand these solutions into Azure AD, migrate the MFA settings for users to Azure AD, or start over with Azure MFA to continue to have the same level of security going forward.

When enabling PTA in Azure AD Connect, the Windows Server running Azure AD Connect gets the PTA agent installed. Install any additional PTA agents on additional Windows Server installations. Microsoft recommends that you have at least two additional PTA agents.

To allow Seamless SSO, people using domain-joined devices authenticate to the Azure AD Authentication service with their domain accounts using Kerberos. By default, Kerberos is not available for all websites in a browser for security reasons. The Azure AD Authentication service needs to be added to the **Intranet Sites** list of Internet Explorer and Edge to allow it.

Mozilla Firefox does not automatically allow Kerberos authentication. Add the `https://autologon.microsoftazuread-sso.com` URL to `network.negotiate-auth.trusted-URI's` in Firefox's `about:config` settings page.

Google Chrome does not automatically allow Kerberos authentication. Add `https://autologon.microsoftazuread-sso.com` to the `AuthNegotiateDelegateWhitelist` or `AuthServerWhitelist` settings.

Changing the user sign-in method in Azure AD Connect makes no changes to AD FS. Instead, the RPT between AD FS and Azure AD, aptly named **Microsoft Office 365 Identity Platform**, needs to be deleted to properly clean up after migrating all of the DNS domain names from AD FS to PTA.

There's more...

The AD FS farm can be decommissioned when there are no more RPTs in AD FS.

To do this, perform these recipes in *Chapter 13*, *Managing Federation*:

- *Decommissioning a Web Application Proxy*
- *Removing AD FS servers from an AD FS farm*

Making PTA (geo)redundant

This recipe shows how to install additional PTA agents to make PTA (geo)redundant.

Getting ready

To register additional PTA agents with Azure AD, you'll need to sign in with an account that is a local administrator on the Windows Server installation you plan to run the PTA agent on. As part of the following steps, you'll need to enter the credentials for these accounts:

- An account in Active Directory that is a member of the **Enterprise Admins** group
- An account in Azure AD that has the **Global administrator** role assigned

Ensure the Windows Server running the additional PTA agent is domain-joined, able to communicate with the internet without having to pass proxies, is running Windows Server 2016 or a newer version of Windows Server, and has IE ESC turned off.

Download the PTA agent from the following URL:

`aka.ms/getauthagent`

If the organization uses the Azure AD PIM feature, activate the **Global administrator** role in advance.

How to do it...

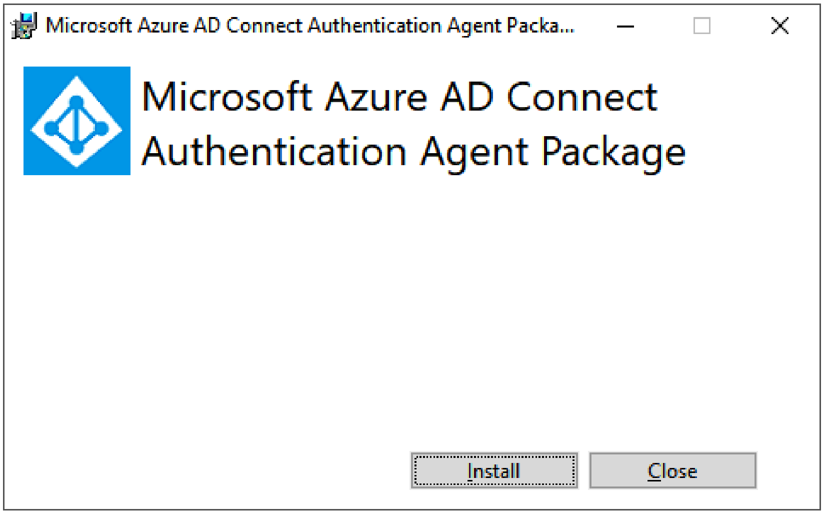
Adding an additional PTA agent consists of the following two steps:

1. Installing and configuring the PTA agent
2. Checking proper installation and configuration

Installing and configuring the PTA agent

Perform the following steps on the additional PTA agent:

1. Double-click `AADConnectAuthAgentSetup.exe`. The **Microsoft Azure AD Connect Authentication Agent Package** window appears:



Microsoft Azure AD Connect Authentication Agent Packa...
Microsoft Azure AD Connect Authentication Agent Package
Install
Close

Figure 14.18 – The Microsoft Azure AD Connect Authentication Agent Package window

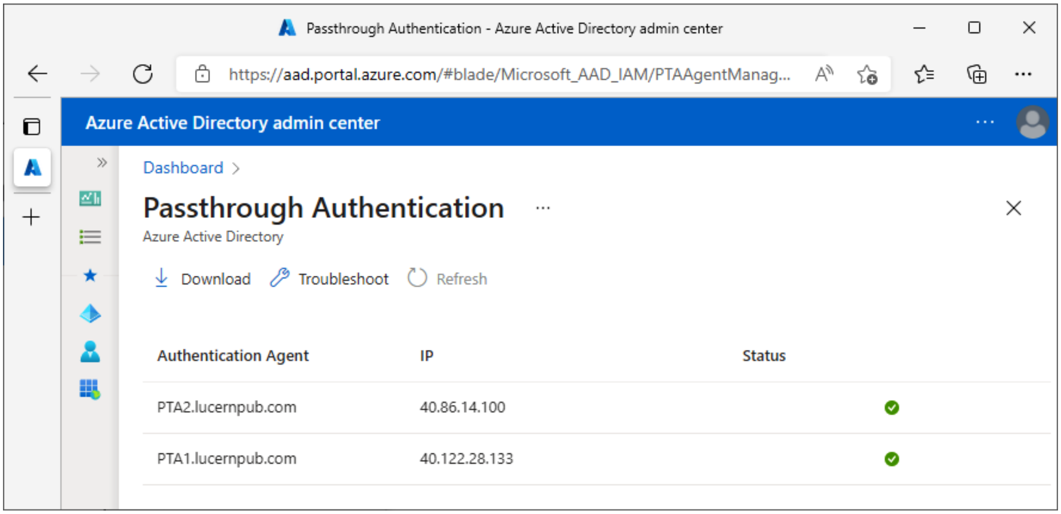
2. Click **Install**. The **Sign in to your account** screen appears.
3. Sign in with an account in Azure AD that has the **Global administrator** role assigned.
4. Perform multi-factor authentication when prompted.
5. Back in the **Microsoft Azure AD Connect Authentication Agent Package** window, click **Close** to close the window and finish the installation.

Checking proper installation and configuration

To check the installation and configuration of the PTA agent in Azure AD, perform the following steps:

1. In a browser, navigate to `portal.azure.com`.
2. Sign in with an account in Azure AD that has the **Global administrator**, **Hybrid Identity administrator**, or **Global Reader** roles assigned.
3. Perform multi-factor authentication when prompted.
4. In the left-hand navigation pane, click **Azure Active Directory**.
5. In the Azure AD navigation pane, click **Azure AD Connect**.

   In the main pane for Azure AD Connect, under **USER SIGN-IN**, you should see multiple agents **Enabled** for **Pass-through authentication**.
6. Follow the **Pass-through authentication** link. This redirects to the **Passthrough Authentication** pane:


Passthrough Authentication - Azure Active Directory admin center
https://aad.portal.azure.com/#blade/Microsoft_AAD_IAM/PTAAgentManag...
Azure Active Directory admin center
Dashboard >
Passthrough Authentication
Azure Active Directory
Download
Troubleshoot
Refresh

Authentication Agent	IP	Status
PTA2.lucernpub.com	40.86.14.100	✓
PTA1.lucernpub.com	40.122.28.133	✓

Figure 14.19 – The Passthrough Authentication pane in the Azure Active Directory admin center

In the **Passthrough Authentication** pane, you should see your PTA agents with their IP addresses, indicating their outside IP addresses and status. You should see green checks.

How it works...

By default, the Windows Server running Azure AD Connect, on which PTA was configured, acts as the first and only PTA agent.

PTA agents can be added to make PTA (geo)redundant and make authentication for end users not dependent on one Windows Server installation.

So, how many PTA agents should you place? Microsoft recommends a minimum of three PTA agents.

Each PTA agent is equipped with its own certificate, for which it uses the private key to decrypt authentication requests. The Azure AD Authentication service puts each authentication request on the Azure Service Bus, specifically encrypted for each registered PTA agent. Therefore, registering unnecessary PTA agents or PTA agents across tremendously slow networking connections might mean more authentication requests on the Azure Service Bus and, thus, delays for end users authenticating using PTA.

15
Handling Synchronization in a Hybrid World (Azure AD Connect)

The previous chapter discussed authentication in a hybrid world and touched upon **Azure AD Connect**. This chapter provides in-depth recipes for Azure AD Connect, as it is both Microsoft's recommended synchronization tool and the most used synchronization tool, used by 99% of all tenants worldwide.

The recipes in this chapter contain configuration items that are not available when **Express Settings** is used with Azure AD Connect. The **Customize** button is the key to the functionality outlined here.

The following recipes are covered in this chapter:

- Choosing the right source anchor attribute for user objects
- Configuring staging mode
- Switching to a staging-mode server

- Configuring domain and OU filtering
- Configuring Azure AD app and attribute filtering
- Configuring hybrid Azure AD join
- Configuring device writeback
- Configuring password writeback
- Configuring group writeback
- Changing passwords for Azure AD Connect service accounts

Choosing the right source anchor attribute for user objects

This recipe shows how to choose the right source anchor attribute for user objects in Azure AD Connect.

Getting ready

To make a choice, you need to know the following characteristics of your organization:

- Is your organization's current Active Directory environment a multi-forest environment?
- Is your organization currently consolidating Active Directory domains and/or planning to acquire other organizations and configure these into the current hybrid identity environment in scope for synchronization by Azure AD Connect?
- Does your organization already have a federation solution and use claims-based applications, either inside your organization or with cloud-based applications, systems, and/or services?
- Does your organization currently use or plan to use a third-party solution that leverages the content of the `mS-DS-ConsistencyGUID` attribute?

How to do it...

Use the following flowchart to choose the right source anchor attribute for your organization:

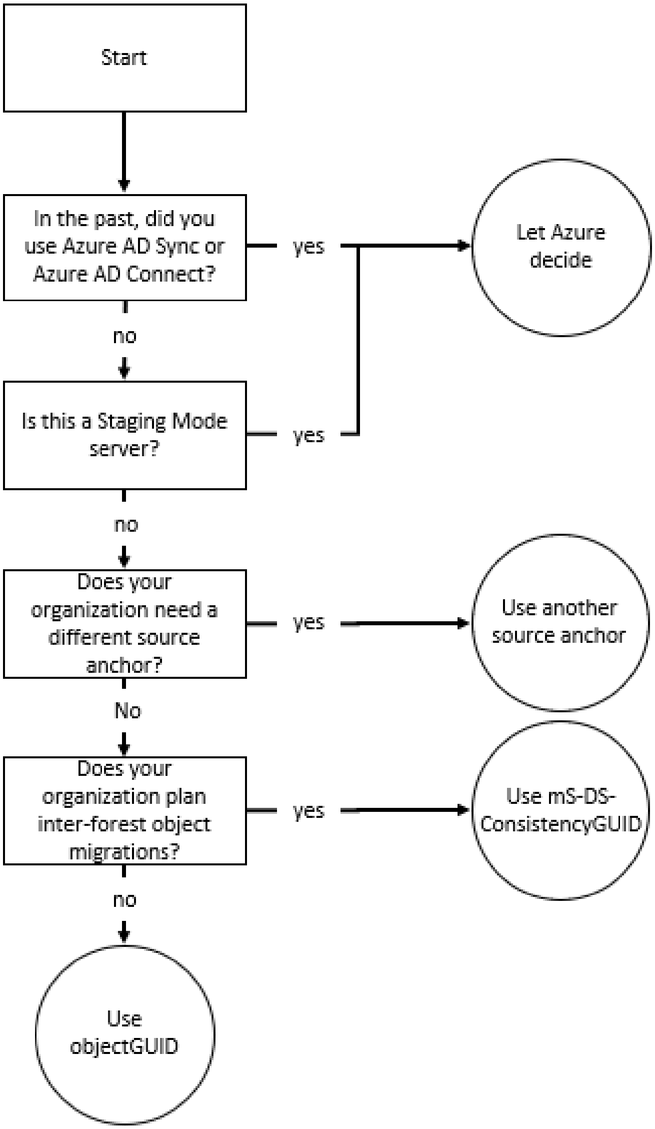

Start
In the past, did you use Azure AD Sync or Azure AD Connect?
yes
Let Azure decide
no
Is this a Staging Mode server?
yes
no
Does your organization need a different source anchor?
yes
Use another source anchor
No
Does your organization plan inter-forest object migrations?
yes
Use mS-DS-ConsistencyGUID
no
Use objectGUID

Figure 15.1 – Flowchart to choose the right source anchor attribute

How it works...

In hybrid identity, where Active Directory and Azure AD work together, an attribute needs to be agreed upon to be the end-to-end identifier. In Azure AD Connect, this is called a source anchor.

For synchronization purposes, this attribute needs to be immutable, meaning it doesn't change during the lifetime of an object and is unique. That's why email addresses and surnames make for bad source anchor attributes: an email address might change when someone gets married, and a surname might not be unique throughout the organization.

Beyond immutability and uniqueness, the attribute value for a source anchor attribute must be fewer than 60 characters in length; must be either a string, integer, or binary; should not be case-sensitive and should avoid values that may vary by case; and should be assigned when an object is created.

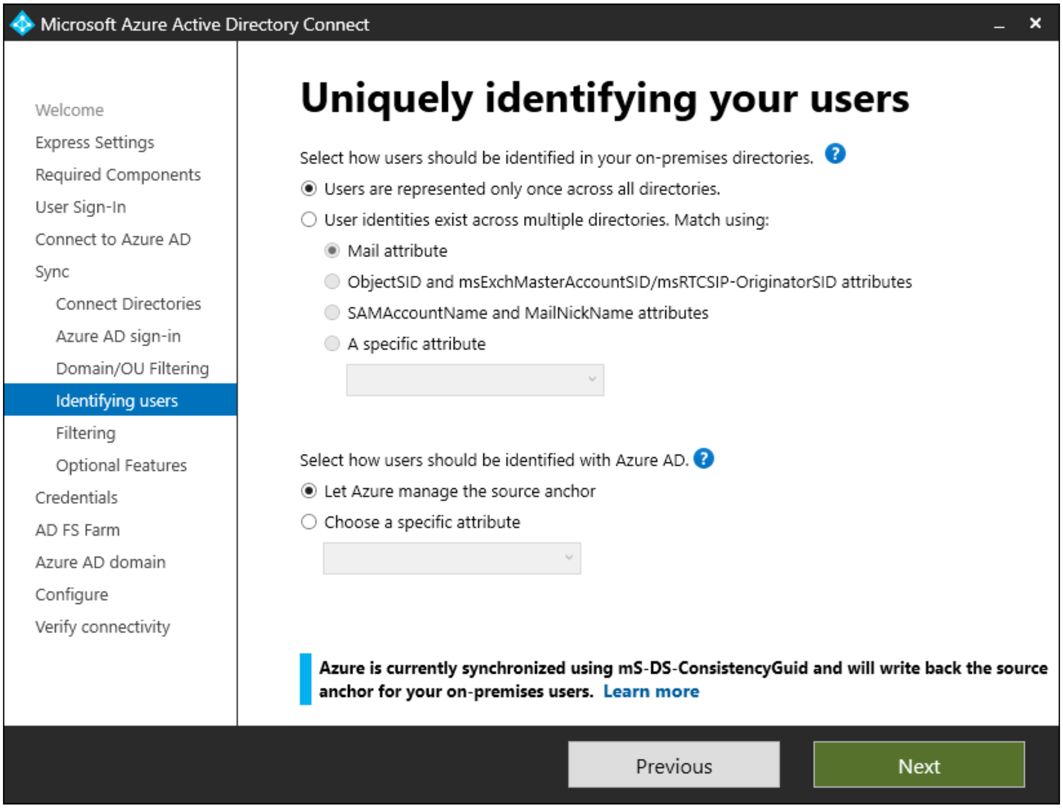
If you have a single Active Directory forest on-premises, then an attribute you could use is `objectGUID`. If you foresee Active Directory migrations or consolidations, you should use `mS-DS-ConsistencyGUID` as the source anchor attribute.

The source anchor attribute for groups is always the `mS-DS-ConsistencyGUID` attribute.

In Azure AD Connect, during initial configuration only, when **Customize** is chosen on the **Express Settings** screen, there are two options on the **Uniquely identifying your users** screen under **Select how users should be identified with Azure AD.**, as follows:

- **Let Azure manage the source anchor**
- **Choose a specific attribute**

These options are shown in the following screenshot:



Uniquely identifying your users

Select how users should be identified in your on-premises directories.

● Users are represented only once across all directories.
○ User identities exist across multiple directories. Match using:
● Mail attribute
○ ObjectSID and msExchMasterAccountSID/msRTCSIP-OriginatorSID attributes
○ SAMAccountName and MailNickName attributes
○ A specific attribute

Select how users should be identified with Azure AD.

● Let Azure manage the source anchor
○ Choose a specific attribute

Azure is currently synchronized using mS-DS-ConsistencyGuid and will write back the source anchor for your on-premises users. Learn more

Figure 15.2 – Uniquely identifying your users screen

When you select the **Let Azure manage the source anchor** option, the Azure AD Connect wizard queries your Azure AD tenant to retrieve the Active Directory attribute used as the source anchor attribute in a previous Azure AD Connect installation (if any). If this information is available, Azure AD Connect uses the same Active Directory attribute.

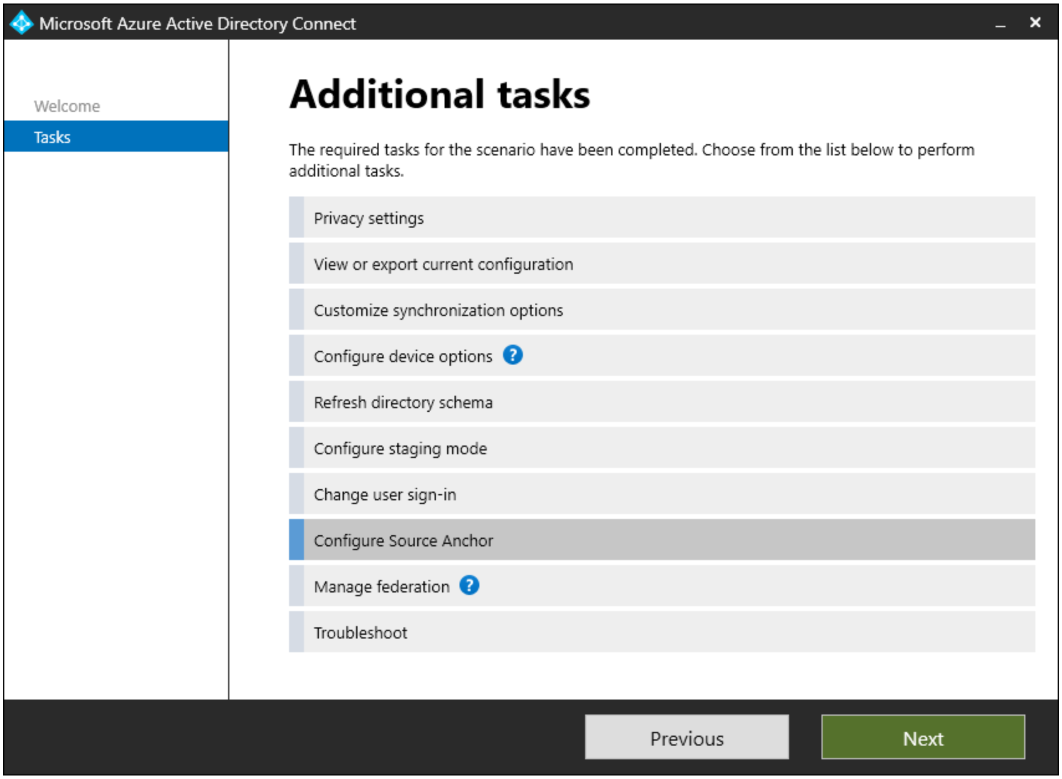
If information about the source anchor attribute isn't available, Azure AD Connect checks the state of the `mS-DS-ConsistencyGUID` attribute in Active Directory using the **Lightweight Directory Access Protocol** (**LDAP**). If the attribute isn't configured on any object in the directory, Azure AD Connect configures `mS-DS-ConsistencyGUID` as the source anchor attribute. If the attribute is configured on one or more objects in the directory, Azure AD Connect concludes the attribute is being used by other applications and is not suitable as the source anchor attribute. In such cases, the wizard falls back to using `objectGUID` as the source anchor attribute.

When using **Express Settings**, the **Select how users should be identified with Azure AD.** choice is not available and the preceding logic always applies.

The value stored in the `mS-DS-ConsistencyGUID` attribute is a Base64 representation of the value of the `objectGUID` attribute. Azure AD Connect writes this attribute during the first export to Active Directory, based on the value of the `objectGUID` attribute for the object when it first comes into the scope of Azure AD Connect.

If the source anchor attribute needs to be changed from the `mS-DS-ConsistencyGUID` attribute to the `objectGUID` attribute, then you must uninstall and reinstall Azure AD Connect.

If the source anchor attribute needs to be changed from the `objectGUID` attribute to the `mS-DS-ConsistencyGUID` attribute, use the **Configure Source Anchor** additional task in Azure AD Connect:



Additional tasks

The required tasks for the scenario have been completed. Choose from the list below to perform additional tasks.

Figure 15.3 – Configure Source Anchor additional task

Any settings for the source anchor need to be reconfigured on Azure AD Connect installations in staging mode too.

There's more...

When using **Customize** and creating your own Azure AD Connect service account, make sure the account is delegated to write the `mS-DS-ConsistencyGUID` attribute for user objects in the scope of Azure AD Connect.

Configuring staging mode

This recipe provides tips for configuring Azure AD Connect in staging mode.

Getting ready

To implement one or more staging-mode servers, you need to meet the same requirements as when implementing the actively synchronizing Azure AD Connect installation. In short, you need to do the following:

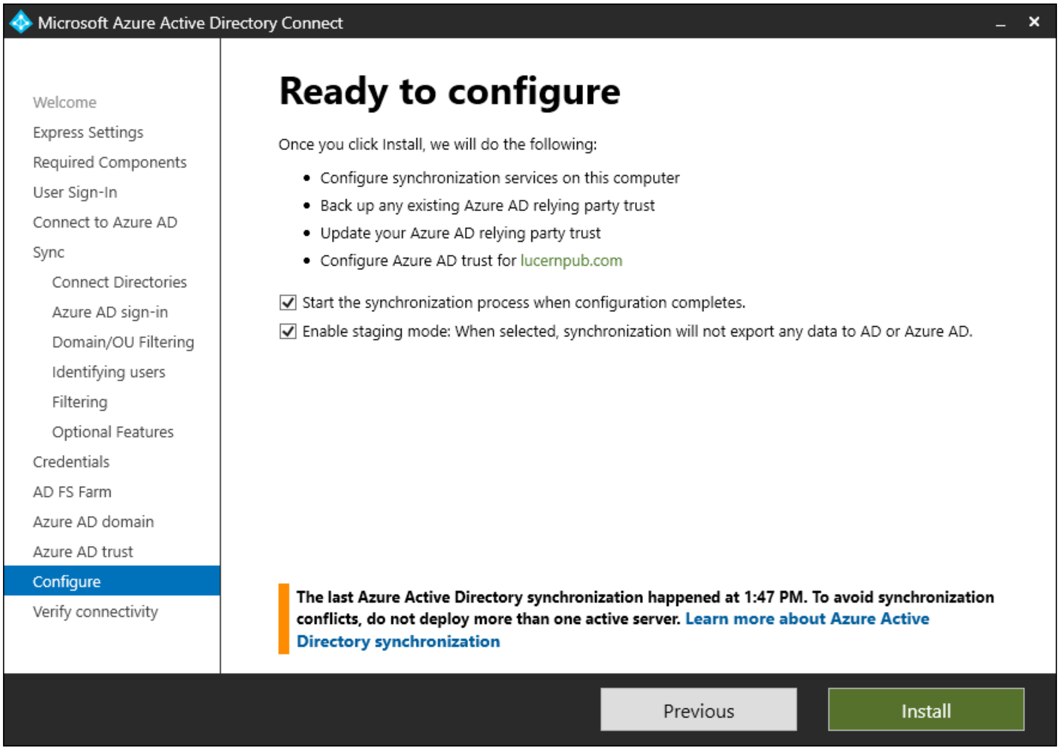
- Sign in with an account that is a local administrator account on the server. As part of the process, credentials for the following accounts need to be specified:
    - An account in Active Directory that is a member of the **Enterprise Admins** group
    - An account in Azure AD that has the **Global administrator** or **Hybrid Identity administrator** role assigned
- Ensure that the following are true of the Windows Server that you intend to configure as an Azure AD Connect staging-mode server:
    - Is able to communicate with the internet without having to pass proxies
    - Is running Windows Server 2016, or a newer version of Windows Server
    - Is domain-joined
    - Has **Internet Explorer Enhanced Security Configuration** (**IE ESC**) turned off

Download the latest version of Azure AD Connect from `https://www.microsoft.com/en-us/download/details.aspx?id=47594`.

If the organization uses the Azure AD **Privileged Identity Management** (**PIM**) feature, activate the **Global administrator** or **Hybrid Identity administrator** role in advance.

How to do it...

Configure an additional Windows Server with Azure AD Connect. Perform identical steps to those you performed when configuring the actively synchronizing Azure AD Connect installation or import the settings from the actively synchronizing Azure AD Connect installation, with one exception. On the **Ready to Configure** screen, enable the **Enable staging mode: When selected, synchronization will not export any data to AD or Azure AD.** option, before clicking **Install**:



Ready to configure

Once you click Install, we will do the following:

- Configure synchronization services on this computer
- Back up any existing Azure AD relying party trust
- Update your Azure AD relying party trust
- Configure Azure AD trust for lucernpub.com

☑ Start the synchronization process when configuration completes.

☑ Enable staging mode: When selected, synchronization will not export any data to AD or Azure AD.

The last Azure Active Directory synchronization happened at 1:47 PM. To avoid synchronization conflicts, do not deploy more than one active server. Learn more about Azure Active Directory synchronization

Figure 15.4 – Ready to configure screen

How it works...

Staging mode offers the ability to configure additional Azure AD Connect installations for an Azure AD tenant. To provide the same level of integrity for multiple Azure AD Connect installations as for a single Azure AD Connect installation, two processes occur:

- Only one Azure AD Connect installation is the actively synchronizing Azure AD Connect installation. It performs **import**, **synchronization**, and **export** *profiles* during synchronization cycles.

- The other Azure AD Connect installations are only synchronizing changes into their databases, but not out to Active Directory or Azure AD. They perform **import** and **synchronization** profiles during synchronization cycles but no **export** profiles.

Staging-mode servers can be useful in the following scenarios:

- As cold standbys to the actively synchronizing Azure AD Connect installation. In this scenario, staging-mode servers offer a way to query the metaverse or create backups of the configuration, without impacting the performance of the actively synchronizing Azure AD Connect installation.
- As a way to perform life cycle management of Azure AD Connect.

There is one big downside to the staging-mode approach: a lot of configuration changes need to be performed on each Azure AD Connect installation to retain the required level of integrity to achieve the preceding benefits. For each of the recipes in this chapter, a remark is made if a configuration change is needed on the actively synchronizing Azure AD Connect installation only, or also on all staging-mode Azure AD Connect installations.

> **Note**
>
> It is important to configure staging-mode Azure AD Connect installations in an identical way to the actively synchronizing Azure AD Connect installation.

There are two ways to ensure identical configuration between Azure AD Connect installations:

1. Using the **Problem Steps Recorder** (**PSR**) tool (`psr.exe`) to record actions when configuring the actively synchronizing Azure AD Connect installation and performing the same configuration steps for staging-mode servers
2. Exporting the configuration of the actively synchronizing Azure AD Connect installation and importing it during the configuration of the staging-mode server

The precise moment for the synchronization cycle within the default time frame and the service account used as the Azure AD Connector account will be different between the actively synchronizing Azure AD Connect installation and staging-mode Azure AD Connect installations. This is no reason for concern.

When configuring a staging-mode server, there is no direct need to use the exact same accounts on the **Connect to Azure AD** and **Connect to Active Directory** screens.

The AD Connector account will be different between the actively synchronizing Azure AD Connect installation and staging-mode Azure AD Connect installations when you let Azure AD Connect create an account. If you manually create a service account, ensure each Azure AD Connect installation uses its own AD Connector account to avoid reuse, permissions issues, and lockouts.

> **Note**
>
> When you manually create service accounts, delegate the appropriate permissions in Active Directory to a group and make your newly created service accounts members of the group. This avoids misconfiguration of individual accounts and adheres to Microsoft's recommended practices for Active Directory **Delegation of Control**.

When using a dedicated **Structured Query Language** (**SQL**) server on the network to store the Azure AD Connect configuration and metaverse, do not reuse the database between Azure AD Connect installations; each Azure AD Connect installation requires its own database.

See also

The Azure AD Connect configuration documenter report generator can be used to compare existing configurations between Azure AD Connect configurations:

`https://github.com/Microsoft/AADConnectConfigDocumenter`

Switching to a staging-mode server

This recipe shows how to switch the actively synchronizing Azure AD Connect installation and a staging-mode installation.

Getting ready

Sign in with an account that is a local administrator on the actively synchronizing Azure AD Connect installation. Also, sign in to the staging-mode installation that you want to switch to.

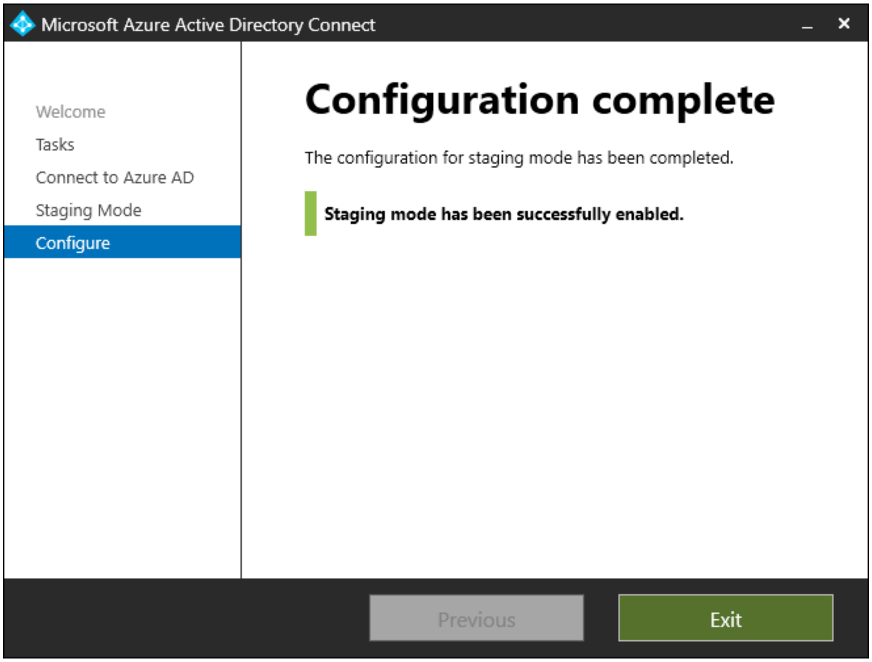
As part of the switch, you need to enter credentials for an account in Azure AD that has the **Global administrator** or **Hybrid Identity administrator** role assigned.

If the organization uses the Azure AD PIM feature, activate the **Global administrator** or **Hybrid Identity administrator** role in advance.

How to do it...

To switch the actively synchronizing Azure AD Connect installation and a staging-mode installation, perform these steps on the actively synchronizing Azure AD Connect installation, if this installation is still operable:

1. Open **Azure AD Connect** from the desktop. The **Microsoft Azure Active Directory Connect** window appears.
2. On the **Welcome to Azure AD Connect** screen, click **Configure**.
3. From the **Additional Tasks** list, select the **Configure staging mode** ribbon.
4. Click **Next**.
5. On the **Connect to Azure AD** screen, sign in with an Azure AD-based account with the **Global administrator** or **Hybrid Identity administrator** role assigned.
6. Perform **multi-factor authentication** (**MFA**) when prompted.
7. On the **Configure staging mode** screen, select the **Enable staging mode** option.
8. Click **Next**.
9. On the **Ready to configure** screen, click **Configure**.
10. On the **Configuration complete** screen, click **Exit** to close the **Microsoft Azure Active Directory Connect** window:



Configuration complete

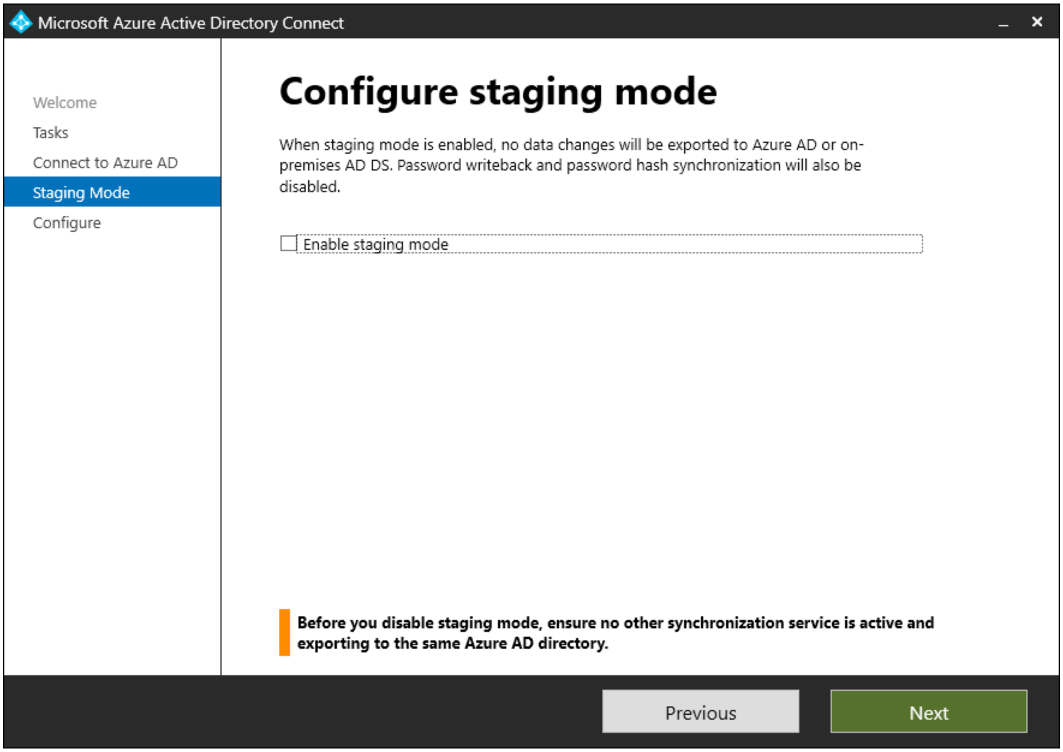
The configuration for staging mode has been completed.

Staging mode has been successfully enabled.

Figure 15.5 – Configuration complete screen

Perform the following steps on the staging-mode Azure AD Connect installation that you want to become the actively synchronizing Azure AD Connect installation:

1. Open **Azure AD Connect** from the desktop. The **Microsoft Azure Active Directory Connect** window appears.
2. On the **Welcome to Azure AD Connect** screen, click **Configure**.
3. From the **Additional Tasks** list, select the **Configure staging mode** ribbon.
4. Click **Next**.
5. On the **Connect to Azure AD** screen, sign in with an Azure AD-based account with the **Global administrator** or **Hybrid Identity administrator** role assigned.
6. Perform MFA when prompted.
7. On the **Configure staging mode** screen, deselect the **Enable staging mode** option:



Configure staging mode

When staging mode is enabled, no data changes will be exported to Azure AD or on-premises AD DS. Password writeback and password hash synchronization will also be disabled.

□ Enable staging mode

Before you disable staging mode, ensure no other synchronization service is active and exporting to the same Azure AD directory.

Figure 15.6 – Enable staging mode option on the Configure staging mode screen

8. Click **Next**.
9. On the **Ready to configure** screen, click **Configure**.
10. On the **Configuration complete** screen, click **Exit** to close the **Microsoft Azure Active Directory Connect** window and to restart synchronization.

How it works...

There can only be one Azure AD Connect installation actively synchronizing to an Azure AD tenant. A staging-mode server can be switched to being the actively synchronizing installation.

The first part of the switch is configuring the actively synchronizing Azure AD Connect installation as an additional staging-mode server. After these steps, there is no actively synchronizing Azure AD Connect installation for the Azure AD tenant.

Don't wait too long to configure one of the Azure AD Connect installations as the actively synchronizing Azure AD Connect installation; synchronization cycles may be missed, resulting in out-of-date objects and attributes.

Configuring domain and OU filtering

This recipe shows how to configure domain and **Organizational Unit** (**OU**) filtering in Azure AD Connect to filter a set of objects that are synchronized to Azure AD.

Getting ready

To configure the **Domain and OU filtering** functionality in Azure AD Connect, you need to know the following characteristics of your organization:

- In which domains, OUs, and containers are the end users for my organization stored?
- Where are the objects that the organization doesn't want to be synchronized to Azure AD?

To configure domain and OU filtering within Azure AD Connect, sign in with an account that is a local administrator on the server running Azure AD Connect. You sign in with an account that is a local administrator on the server. As part of the process, the credentials for the following accounts need to be specified:

- An account in Active Directory that is a member of the **Enterprise Admins** group
- An account in Azure AD that has the **Global administrator** or **Hybrid Identity administrator** role assigned

If the organization uses the Azure AD PIM feature, activate the **Global administrator** or **Hybrid Identity administrator** role in advance.

How to do it...

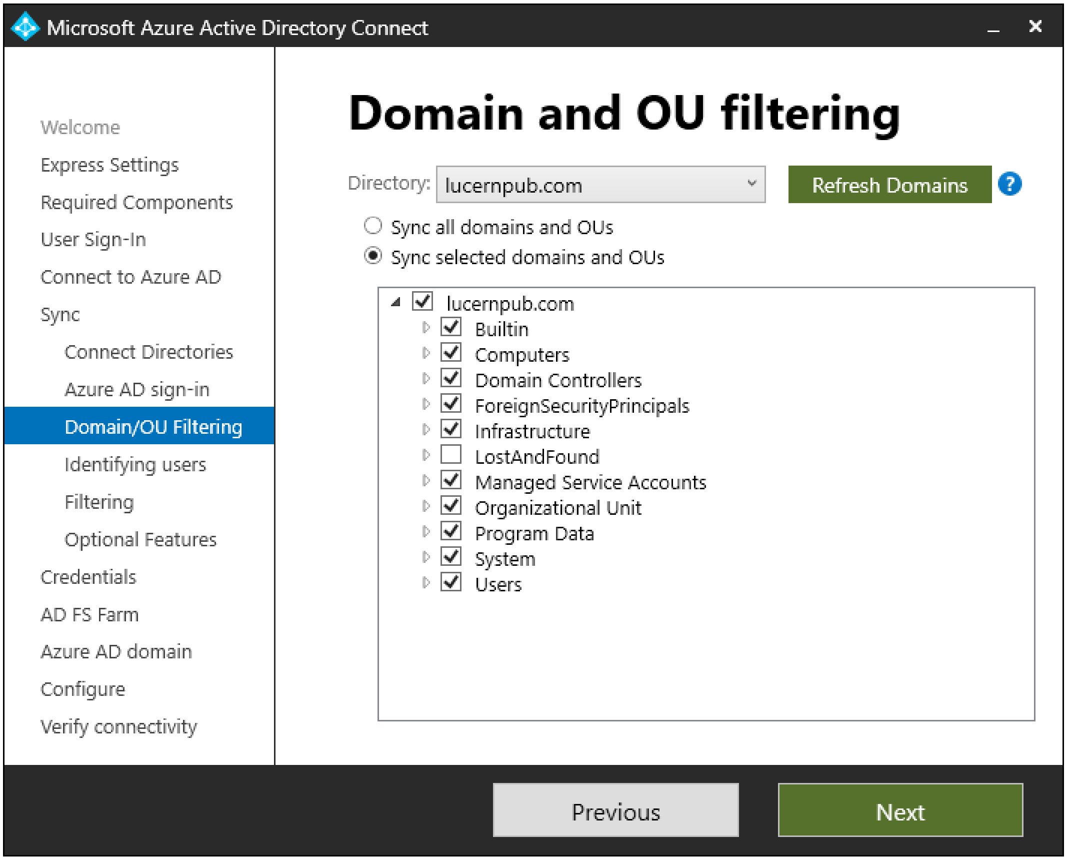
Configuring domain and OU filtering can be handled in two scenarios:

- When configuring Azure AD Connect initially
- When reconfiguring Azure AD Connect

Configuring Azure AD Connect initially

When initially configuring Azure AD Connect, ensure to click **Customize** on the **Express Settings** screen of Azure AD Connect. You encounter the **Domain and OU filtering** screen, regardless of whether you are configuring Azure AD Connect for **password hash synchronization** (**PHS**), **pass-through authentication** (**PTA**), or federation:

1. Select the **Sync selected domains and OUs** option instead of the default **Sync all domains and OUs** option.
2. Make appropriate changes to a selection of domains, OUs, and containers:



Domain and OU filtering

Directory: lucernpub.com

○ Sync all domains and OUs
● Sync selected domains and OUs

☑ lucernpub.com
- ☑ Builtin
- ☑ Computers
- ☑ Domain Controllers
- ☑ ForeignSecurityPrincipals
- ☑ Infrastructure
- □ LostAndFound
- ☑ Managed Service Accounts
- ☑ Organizational Unit
- ☑ Program Data
- ☑ System
- ☑ Users

Figure 15.7 – Domain and OU filtering screen

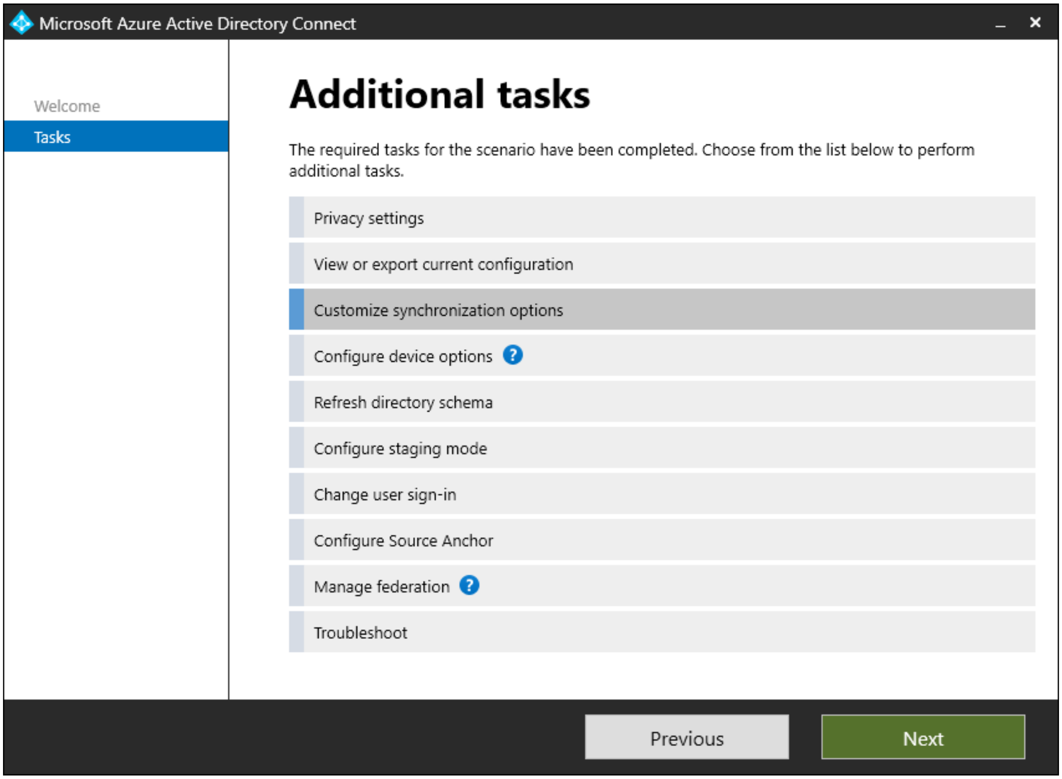
3. Click **Next** to continue configuring Azure AD Connect.

Reconfiguring Azure AD Connect

Perform these steps to reconfigure Azure AD Connect with domain and OU filtering:

1. Open **Azure AD Connect** from the desktop. The **Microsoft Azure Active Directory Connect** window appears.
2. On the **Welcome to Azure AD Connect** screen, click **Configure**.

3. On the **Additional tasks** screen, select the **Customize synchronization options** ribbon:



Additional tasks

The required tasks for the scenario have been completed. Choose from the list below to perform additional tasks.

- Privacy settings
- View or export current configuration
- Customize synchronization options
- Configure device options ?
- Refresh directory schema
- Configure staging mode
- Change user sign-in
- Configure Source Anchor
- Manage federation ?
- Troubleshoot

Figure 15.8 – Customize synchronization options ribbon on the Additional tasks screen

4. Click **Next**.
5. On the **Connect to Azure AD** screen, sign in with an Azure AD-based account with the **Global administrator** or **Hybrid Identity administrator** role assigned.
6. Perform MFA when prompted.
7. Click **Next**.
8. When you want to remove entire Active Directory forests from the scope of Azure AD Connect, remove them on the **Connect your directories** screen.
9. Click **Next**.

10. On the **Domain and OU filtering** screen, select the **Sync selected domains and OUs** option instead of the default **Sync all domains and OUs** option, or when you are revisiting the **Domain and OU filtering** screen, ensure the **Sync selected domains and OUs** option is selected.
11. Make appropriate changes to a selection of domains, OUs, and containers.
12. Click **Next**.
13. On the **Optional features** screen, click **Next**.
14. On the **Ready to configure** screen, click **Configure**.
15. On the **Configuration complete** screen, click **Exit** to close the **Microsoft Azure Active Directory Connect** window and to restart synchronization.

How it works...

The **Domain and OU Filtering** screen presents an option to select and deselect Active Directory domains in a forest, OUs, and containers.

When you remove a forest, ensure to also remove or reconfigure any service account used by Azure AD Connect in that forest.

> **Note**
>
> There is a difference between unselecting a parent OU and selecting one or more of its child OUs, and selecting a parent OU but unselecting one or more of its child OUs. In the latter scenario, new child OUs will be automatically in scope for synchronizing, while in the first scenario, they will not.

When objects that the organization doesn't want to be synchronized to Azure AD are located in OUs with objects that the organization wants to be synchronized, create new OUs.

When objects fall out of the scope of Azure AD Connect, they will automatically be deleted in Azure AD. They will remain in the Azure AD Recycle Bin for 30 days, after which time, the objects are permanently gone.

Any setting for the scope of objects needs to be configured on Azure AD Connect installations in staging mode, too.

Configuring Azure AD app and attribute filtering

This recipe shows how to configure Azure AD app and attribute filtering in Azure AD Connect to filter a set of attributes for objects that are synchronized to Azure AD.

Getting ready

To configure the **Azure AD app and attribute filtering** feature in Azure AD Connect, you need to know the following characteristics of your organization:

- What is the Office 365 functionality my organization is going to use?
- Which attributes for my end users, groups, services, and devices am I allowed to synchronize to Azure AD in terms of regulatory compliance?

To configure the **Azure AD app and attribute filtering** feature within Azure AD Connect, sign in with an account that is a local administrator account on the server running Azure AD Connect. You sign in with an account that is a local administrator on the server.
As part of the process, credentials for the following accounts need to be specified:

- An account in Active Directory that is a member of the **Enterprise Admins** group
- An account in Azure AD that has the **Global administrator** or the **Hybrid Identity administrator** role assigned

If the organization uses the Azure AD PIM feature, activate the **Global administrator** or **Hybrid Identity administrator** role in advance.

How to do it…

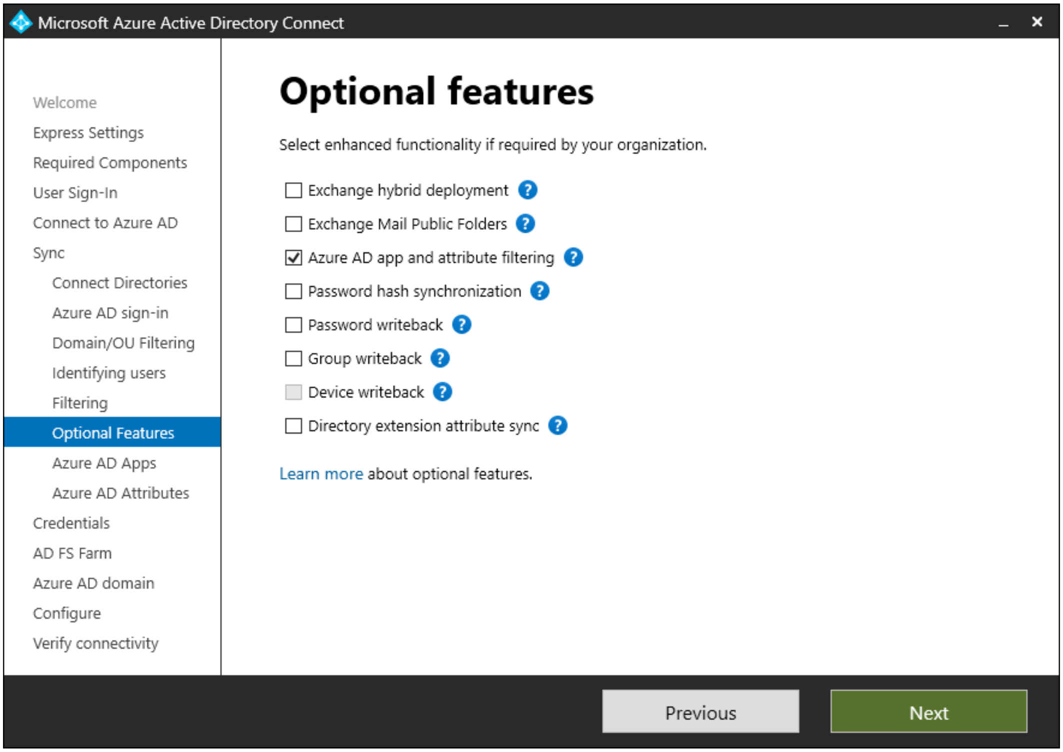
Configuring Azure AD app and attribute filtering can be handled in two scenarios:

- When configuring Azure AD Connect initially
- When reconfiguring Azure AD Connect

Configuring Azure AD Connect initially

When initially configuring Azure AD Connect, ensure to click **Customize** on the **Express Settings** screen of Azure AD Connect. You encounter the **Azure AD app and attribute filtering** screen, regardless of whether you're configuring Azure AD Connect for PHS, PTA, or federation:

1. On the **Optional features** screen, select the **Azure AD app and attribute filtering** option:



Optional features

Select enhanced functionality if required by your organization.

- ☐ Exchange hybrid deployment
- ☐ Exchange Mail Public Folders
- ☑ Azure AD app and attribute filtering
- ☐ Password hash synchronization
- ☐ Password writeback
- ☐ Group writeback
- ☐ Device writeback
- ☐ Directory extension attribute sync

Learn more about optional features.

Figure 15.9 – Optional features screen

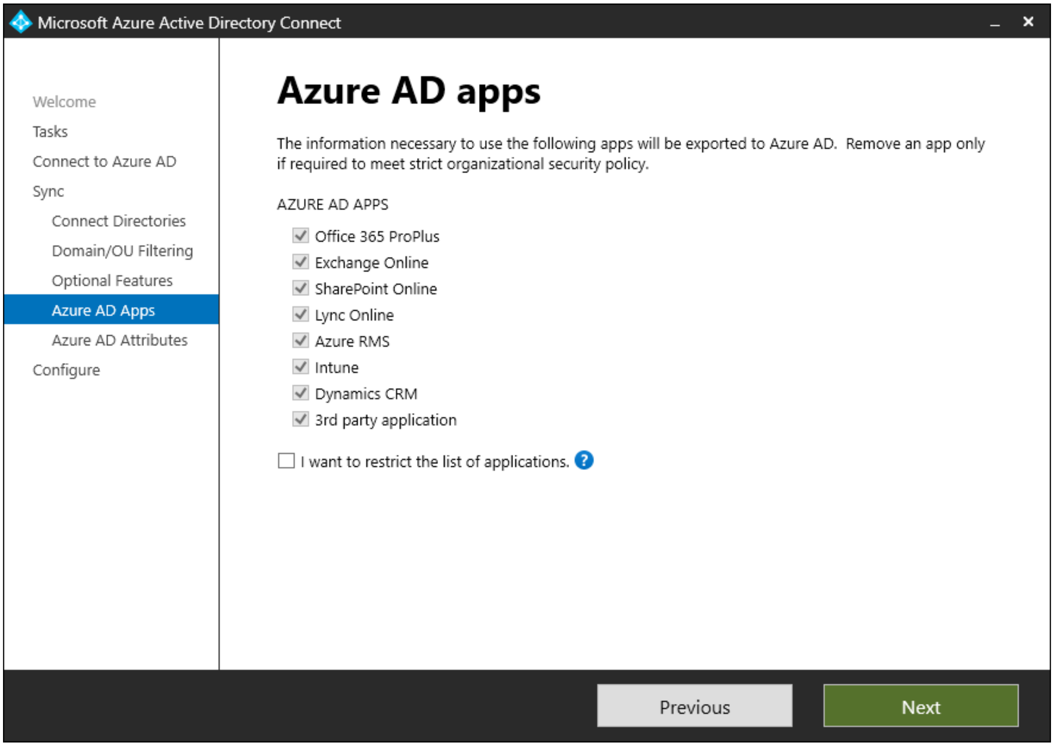
2. Click **Next**.
3. On the **Azure AD apps** screen, select the **I want to restrict the list of applications.** option.
4. This will remove the grayed-out selections for **Office 365 ProPlus**, **Exchange Online**, **SharePoint Online**, **Lync Online**, **Azure RMS**, **Intune**, **Dynamics CRM**, and **3rd party application** options.

5. Select at least one Azure AD app.
6. Click **Next**.
7. On the **Ready to configure** screen, click **Configure**.
8. On the **Configuration complete** screen, click **Exit** to close the **Microsoft Azure Active Directory Connect** window and to start the initial synchronization.

Reconfiguring Azure AD Connect

Perform the following steps to reconfigure Azure AD Connect with **Azure AD app and attribute filtering**:

1. Open **Azure AD Connect** from the desktop. The **Microsoft Azure Active Directory Connect** window appears.
2. On the **Welcome to Azure AD Connect** screen, click **Configure**.
3. On the **Additional tasks** screen, select the **Customize synchronization options** ribbon.
4. Click **Next**.
5. On the **Connect to Azure AD** screen, sign in with an Azure AD-based account with the **Global administrator** or **Hybrid Identity administrator** role assigned.
6. Perform MFA when prompted.
7. On the **Connect your directories** screen, click **Next**.
8. On the **Domain and OU filtering** screen, click **Next**.
9. On the **Optional features** screen, select the **Azure AD app and attribute filtering** option.
10. Click **Next** to advance to the **Azure AD apps** screen:



Azure AD apps

The information necessary to use the following apps will be exported to Azure AD. Remove an app only if required to meet strict organizational security policy.

AZURE AD APPS

☑ Office 365 ProPlus
☑ Exchange Online
☑ SharePoint Online
☑ Lync Online
☑ Azure RMS
☑ Intune
☑ Dynamics CRM
☑ 3rd party application

□ I want to restrict the list of applications.

Figure 15.10 – Azure AD apps screen

11. On the **Azure AD apps** screen, select the **I want to restrict the list of applications.** option.
12. This will remove the grayed-out selections for **Office 365 ProPlus**, **Exchange Online**, **SharePoint Online**, **Lync Online**, **Azure RMS**, **Intune**, **Dynamics CRM**, and **3rd party application** options.
13. Select at least one Azure AD app.
14. Click **Next**.
15. On the **Ready to configure** screen, click **Configure**.
16. On the **Configuration complete** screen, click **Exit** to close the **Microsoft Azure Active Directory Connect** window and to restart synchronization.

How it works...

Increasing regulations, including **European Union**'s (**EU**) **General Data Protection Regulation** (**GDPR**), empower organizations to protect the privacy of people. While it can be argued that synchronizing from Active Directory to Azure AD serves the same purposes in an employer-employee relationship, other organizations may have difficulties rationalizing the synchronization of over 150 attributes for users, groups, contacts, and devices. Azure AD Connect offers the **Azure AD app and attribute filtering** feature to address the needs of this latter kind of organization.

In contrast to stopping the synchronization of objects using the **Domain and OU filtering** functionality, whereby the object eventually disappears, when organizations stop synchronizing certain attributes for objects using Azure AD app and attribute filtering, these attributes aren't (automatically) removed from Azure AD. When organizations want certain attributes to be cleared, PowerShell needs to be used, although some attributes—such as the ones where synchronized (and rehashed) password hashes are stored—cannot be cleared.

A lack of integrity of values for attributes that are no longer synchronized might lead to problems when the organization starts adopting additional services and/or applications. Orphaned attribute values might lead to the undesired behavior of a service and/or an application, or at least part of it.

The **Azure AD app and attribute filtering** feature uses built-in **comma-separated values** (**CSV**) files that act as templates for sets of attributes that Microsoft online services use. There are ready-to-use templates for organizations to deploy Office 365 **Professional Plus** (**ProPlus**) applications on users' devices, Exchange Online, and SharePoint Online, among others.

Any setting for the scope of attributes needs to be configured on Azure AD Connect installations in staging mode, too.

Using the **Azure AD app and attribute filtering** feature does not result in unsupported synchronization rules or an unsupported state for Azure AD Connect.

Configuring hybrid Azure AD join

This recipe shows how to configure hybrid Azure AD join to synchronize device properties for domain-joined devices from Active Directory to Azure AD.

Getting ready

To configure hybrid Azure AD join in Azure AD Connect, you need to know the following characteristics of your organization:

- What are the operating systems in use in the organization? Which attributes for the devices am I allowed to synchronize?
- Which Azure AD Connect installation is the non-staging-mode server? (Only applicable if the organization has multiple Azure AD Connect servers.)

To configure hybrid Azure AD join in Azure AD Connect, you need to sign in with an account that is a local administrator account on a server dedicated to Azure AD Connect.

As part of the process, the credentials for the following accounts need to be specified:

- An account in Active Directory that is a member of the **Enterprise Admins** group
- An account in Azure AD that has the **Global administrator** or **Hybrid Identity administrator** role assigned

If the organization uses the Azure AD PIM feature, activate the **Global administrator** or **Hybrid Identity administrator** role in advance.

Azure AD Connect needs to be initially configured, although the chosen authentication method doesn't matter as hybrid Azure AD join works with PHS, PTA, and federation. PHS and PTA need to be configured in combination with the **Enable single sign-on** setting when non-Windows 10 devices are to be hybrid Azure AD-joined.

The Active Directory schema version needs to be Windows Server 2012 R2 (level 69) or higher.

To deploy the **Workplace Join for non-Windows 10 computers** package, download it from here:

`https://www.microsoft.com/en-us/download/details.aspx?id=53554`

How to do it…

Configuring hybrid Azure AD join consists of these steps:

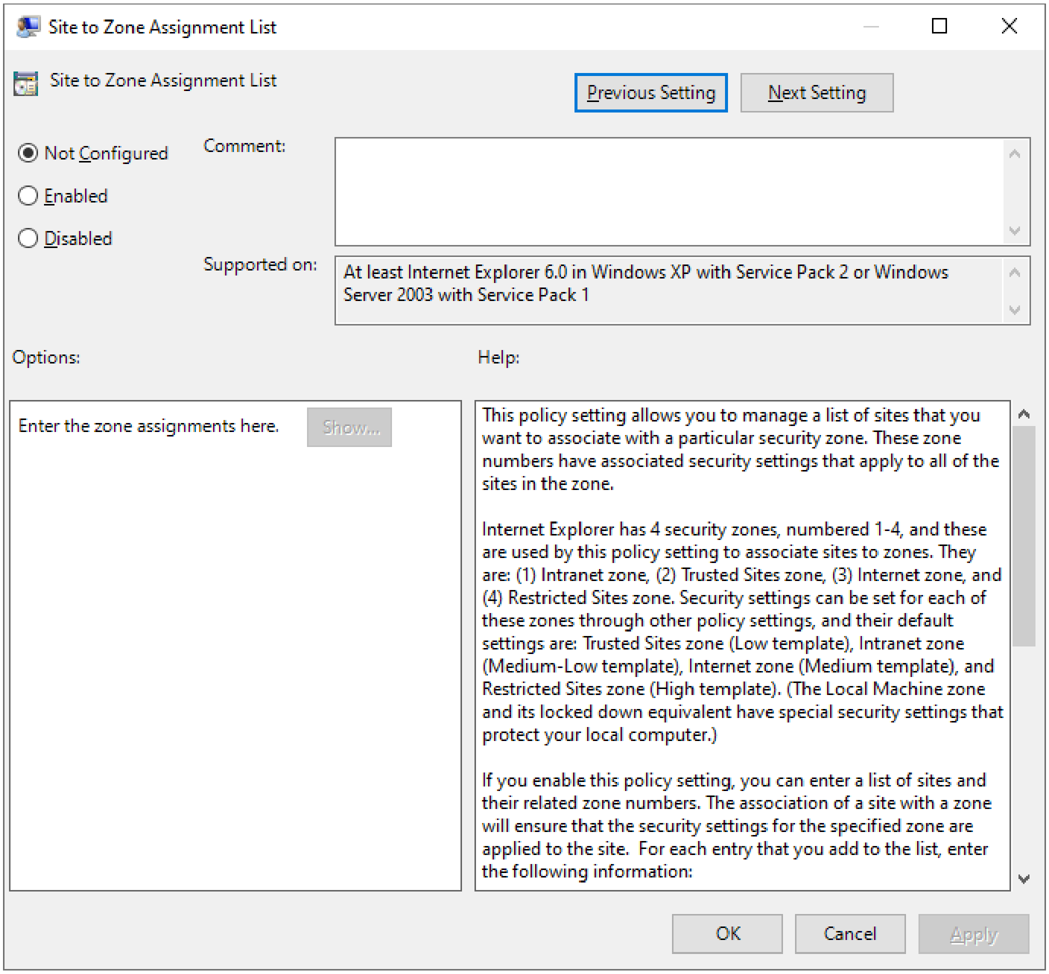
1. Adding the Azure AD Device Registration Service to intranet sites
2. Distributing workplace join for non-Windows 10 computers
3. Setting the **Group Policy Object** (**GPO**) to register for down-level Windows devices
4. Linking the Group Policy to the right OUs
5. Configuring hybrid Azure AD join in Azure AD Connect

Steps 2-4 are only required when the organization runs older versions of Windows, such as Windows 8.1. The package needs to be deployed before the device can understand the Group Policy setting.

Adding the Azure AD Device Registration Service to intranet sites

Perform the following steps to add the Azure AD authentication service to IE's and Edge's **Intranet Sites** list in a new GPO that is targeted to the same OU filtering scope as in Azure AD Connect:

1. Sign in to a system with the **Group Policy Management** feature installed.
2. Press Start.
3. Search for **Group Policy Management** and click its search result or run `gpmc.msc`. The **Group Policy Management window** appears.
4. In the left-hand pane, navigate to the **Group Policy Objects** node.
5. If your organization already manages the **Intranet Sites** list through Group Policy, locate the corresponding Group Policy object and select it. Alternatively, right-click the **Group Policy Objects** node and select **New** from the menu. When creating a new GPO, provide a name for the new GPO and click **OK**.
6. Right-click the Group Policy object and select **Edit…** from the menu. The **Group Policy Management Editor** window appears.
7. In the left-hand navigation pane, expand the **User Configuration** node, then **Policies**, then **Administrative Templates**, **Windows Components**, **Internet Explorer**, **Internet Control Panel**, and—lastly—**Security Page**.
8. Double-click the **Site to Zone Assignment List** setting. The **Site to Zone Assignment List** window appears:



● Not Configured
○ Enabled
○ Disabled

Supported on: At least Internet Explorer 6.0 in Windows XP with Service Pack 2 or Windows Server 2003 with Service Pack 1

Enter the zone assignments here.

This policy setting allows you to manage a list of sites that you want to associate with a particular security zone. These zone numbers have associated security settings that apply to all of the sites in the zone.

Internet Explorer has 4 security zones, numbered 1-4, and these are used by this policy setting to associate sites to zones. They are: (1) Intranet zone, (2) Trusted Sites zone, (3) Internet zone, and (4) Restricted Sites zone. Security settings can be set for each of these zones through other policy settings, and their default settings are: Trusted Sites zone (Low template), Intranet zone (Medium-Low template), Internet zone (Medium template), and Restricted Sites zone (High template). (The Local Machine zone and its locked down equivalent have special security settings that protect your local computer.)

If you enable this policy setting, you can enter a list of sites and their related zone numbers. The association of a site with a zone will ensure that the security settings for the specified zone are applied to the site. For each entry that you add to the list, enter the following information:

Figure 15.11 – Site to Zone Assignment List window

9. Select **Enabled**.
10. Click the **Show…** button. The **Show Contents** window appears.
11. In the **Value name** field, enter the following **Uniform Resource Locator** (**URL**):

```
https://device.login.microsoftonline.com
```

12. In the **value** field, assign `1`, as this value corresponds to the **Intranet Sites** zone.
13. In the next **Value name** field, enter the following URL:

```
https://autologon.microsoftazuread-sso.com
```

14. In the **value** field, also assign `1` to this URL.

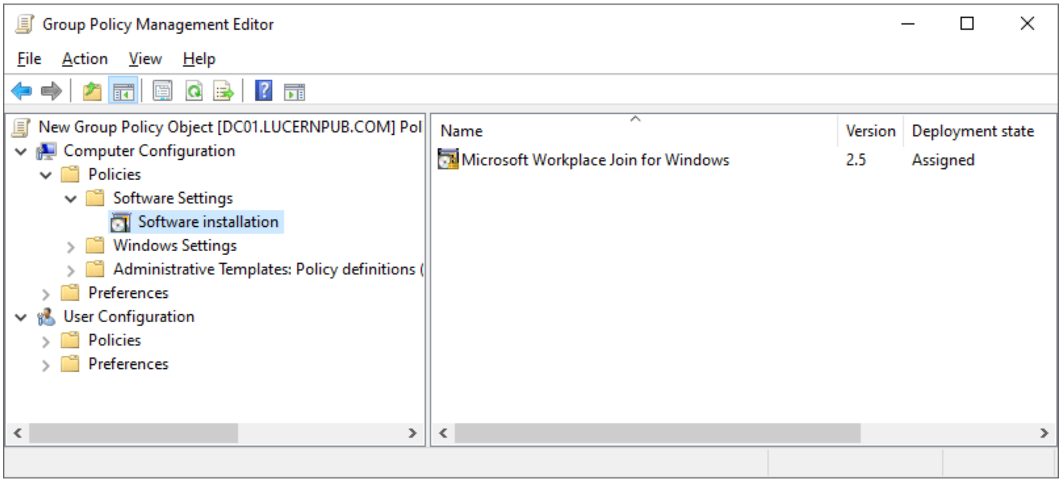
15. Click **OK**.
16. Click **OK** to close the Group Policy setting.
17. Close the **Group Policy Management Editor** window.
18. In the left-hand navigation pane, navigate to the OU where you want to link the GPO.
19. Right-click the OU and select **Link an existing GPO…** from the menu. The **Select GPO** window appears.
20. In the **Select GPO** window, select a GPO.
21. Click **OK** to link the GPO.

Distributing Workplace Join for non-Windows 10 computers

Perform the following steps to install the **Workplace Join** package for non-Windows 10 computers:

1. Press Start.
2. Search for `Group Policy Management` and click its search result or run `gpmc.msc`. The **Group Policy Management window** appears.
3. In the left-hand pane, navigate to the **Group Policy Objects** node.
4. Right-click the **Group Policy Objects** node and select **New** from the menu. The **New GPO** pop-up window appears
5. Enter a name for the new Group Policy object.
6. Click **OK**.
7. In the left-hand navigation pane, right-click the new GPO and select **Edit** from the menu. The **Group Policy Management Editor** window appears.
8. In the left-hand navigation pane, expand the **Computer Configuration** node, then **Policies**, and then **Software Settings**.
9. Right-click the **Software Installation** node and select **New** from the menu, and then **Package…**. The **Open** window appears.
10. In the **Open** window, browse to the network share that has the **Microsoft Installer** (**MSI**) package for the application. Select the application and click **Open**. The **Deploy Software** pop-up screen appears.
11. On the **Deploy Software** pop-up screen, select **Assigned** as the deployment method.
12. Click **OK** to save the settings.

13. In the **Group Policy Management Editor** window, the package will be listed with its version, its deployment state, and its source path:



Name	Version	Deployment state
Microsoft Workplace Join for Windows	2.5	Assigned

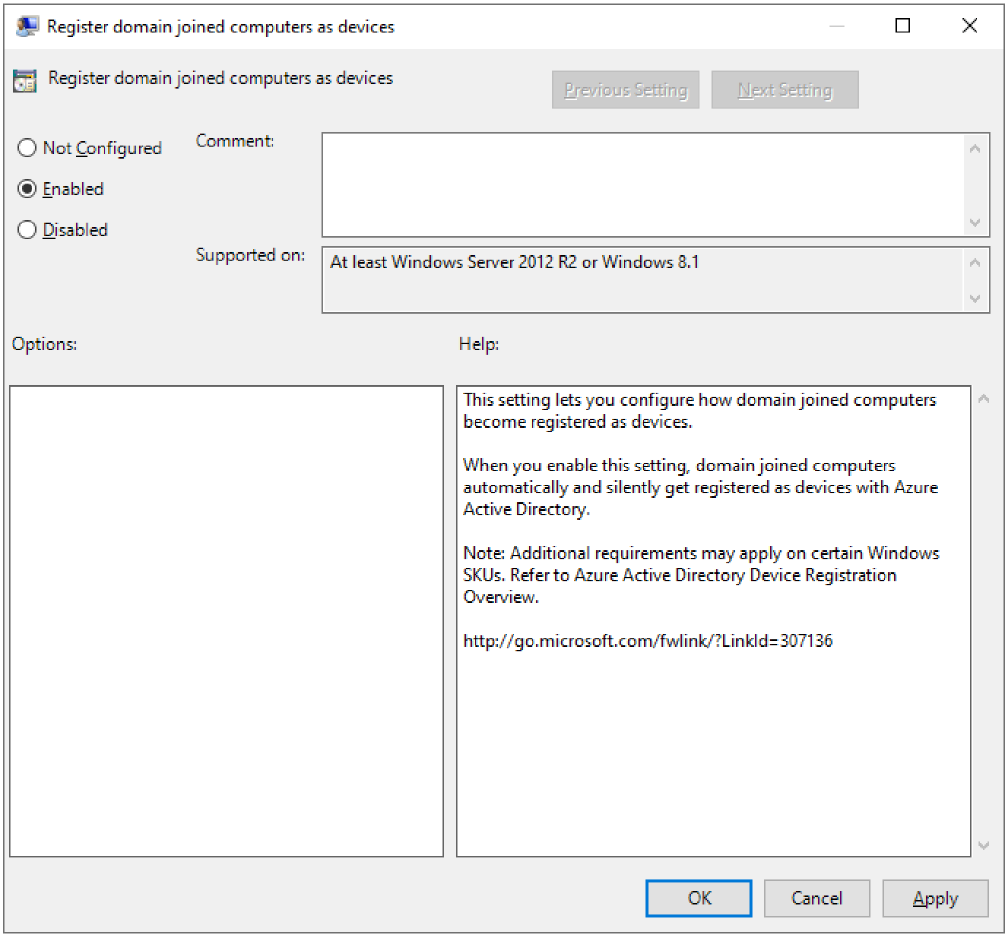
Figure 15.12 – Microsoft Workplace Join for Windows package

Setting the Group Policy to register for down-level Windows devices

The same Group Policy object can be reused to configure down-level Windows installations for hybrid Azure AD join if only down-level Windows devices are present in the environment:

1. In the newly created Group Policy object, expand the **Administrative Templates** node under **Policies** and **Computer Configuration**.
2. Expand the **Windows Components** node.
3. Expand the **Device Registration** node.
4. In the main pane, select the **Register domain-joined computers as devices** setting.
5. Right-click the setting and select **Edit** from the menu. The **Register domain joined computers as devices** window appears.

6. Select the **Enabled** option:



Supported on: At least Windows Server 2012 R2 or Windows 8.1

This setting lets you configure how domain joined computers become registered as devices.

When you enable this setting, domain joined computers automatically and silently get registered as devices with Azure Active Directory.

Note: Additional requirements may apply on certain Windows SKUs. Refer to Azure Active Directory Device Registration Overview.

http://go.microsoft.com/fwlink/?LinkId=307136

Figure 15.13 – Enabled option in the Register domain joined computers as devices window

7. Click **OK**.
8. Close the **Group Policy Management Editor** window.

Linking the Group Policy to the right OUs

Link the newly created Group Policy object to OUs containing devices:

1. In the left-hand navigation pane of the **Group Policy Management** window, navigate to the OU where you want to link an existing GPO.
2. Right-click the OU and select **Link an existing GPO…** from the menu.

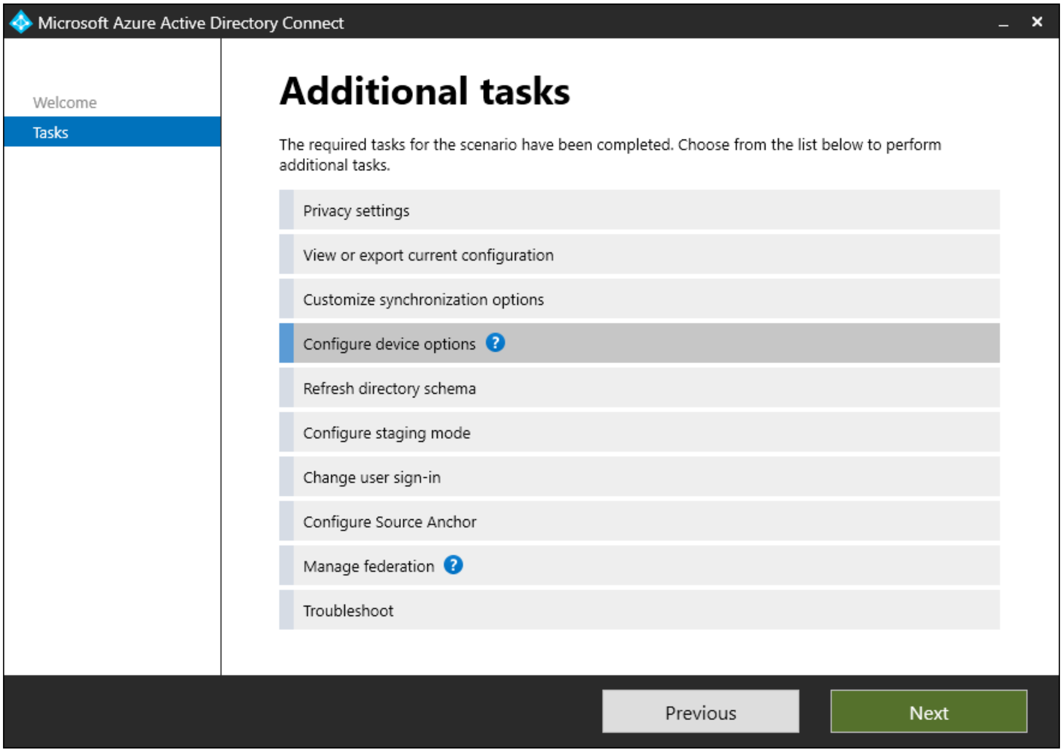
3. In the **Select GPO** window, select the newly created GPO.
4. Click **OK** to link the GPO.

Repeat the preceding steps for all OUs with Windows 8.1-based devices in scope for hybrid Azure AD join.

Configuring hybrid Azure AD join in Azure AD Connect

Perform the following steps on the Windows Server running Azure AD Connect:

1. Open **Azure AD Connect** from the desktop. The **Microsoft Azure Active Directory Connect** window appears.
2. On the **Welcome to Azure AD Connect** screen, click **Configure**.
3. On the **Additional tasks** screen, select the **Configure device options** ribbon:



Microsoft Azure Active Directory Connect

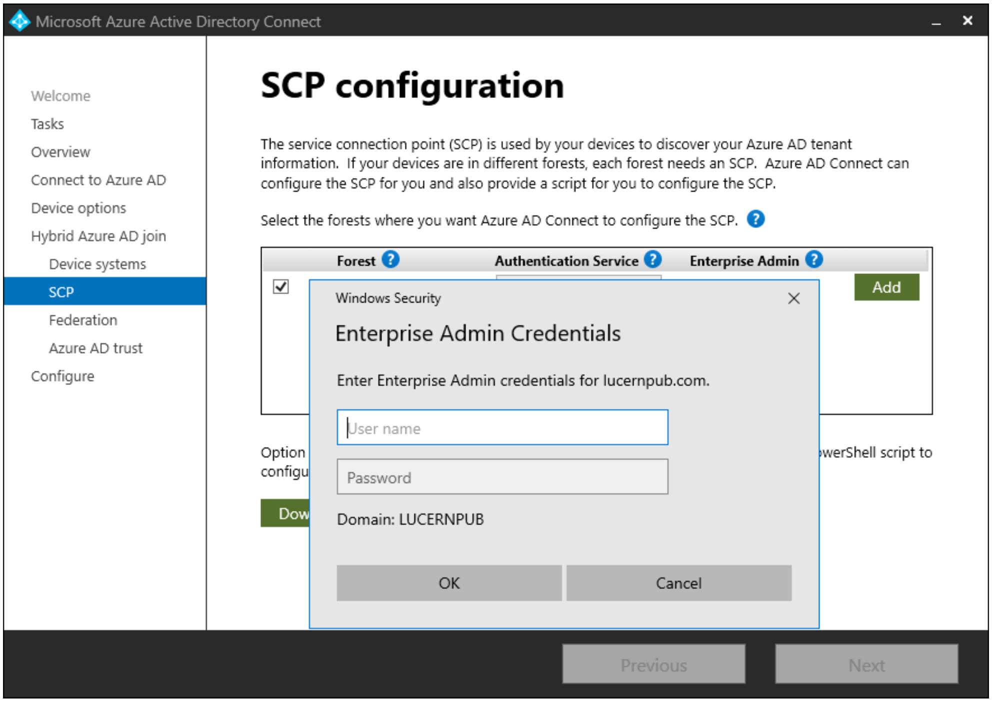
Additional tasks

The required tasks for the scenario have been completed. Choose from the list below to perform additional tasks.

Figure 15.14 – Configure device options ribbon

4. Click **Next**.
5. On the **Overview** screen, click **Next**.

6. On the **Connect to Azure AD** screen, sign in with an Azure AD-based account with the **Global administrator** or **Hybrid Identity administrator** role assigned.
7. Perform MFA when prompted.
8. On the **Device options** screen, select **Configure Hybrid Azure AD join**.
9. Click **Next**.
10. On the **Device operating systems** screen, select the **Windows 10 or later domain-joined devices.** option.
11. If your organization is planning to also support Windows down-level domain-joined devices, select the **Supported Windows downlevel domain-joined devices.** option too.
12. Click **Next**.
13. On the **SCP configuration** screen, select the Active Directory forest to configure hybrid Azure AD join for.
14. In the **Authentication Service** column, select **Azure Active Directory**.
15. Click the **Add** button in the **Enterprise Admin** column. The **Windows Security** pop-up window appears:



Microsoft Azure Active Directory Connect

SCP configuration

The service connection point (SCP) is used by your devices to discover your Azure AD tenant information. If your devices are in different forests, each forest needs an SCP. Azure AD Connect can configure the SCP for you and also provide a script for you to configure the SCP.

Select the forests where you want Azure AD Connect to configure the SCP.

Forest	Authentication Service	Enterprise Admin
☑		Add

Windows Security

Enterprise Admin Credentials

Enter Enterprise Admin credentials for lucernpub.com.

User name

Password

Domain: LUCERNPUB

Figure 15.15 – Windows Security pop-up window

16. Enter credentials for an account in Active Directory that is a member of the **Enterprise Admins** group and then click **OK**. This dismisses the pop-up window.
17. Click **Next**.
18. On the **Ready to configure** screen, click **Configure**.
19. On the **Configuration complete** screen, click **Exit** to close the **Microsoft Azure Active Directory Connect** window and to restart synchronization.

How it works...

Hybrid Azure AD join leverages the domain-join bond a device has with Active Directory to create a device object in Azure AD. A device object is created and then attached to the device, leveraging its **Trusted Platform Module** (**TPM**) device as a secure enclave to store the private key for a certificate that seals the bond with Azure AD.

For hybrid Azure AD join, Active Directory on-premises is the source, and Azure AD is the destination.

Note

After a device bonds with Azure AD, it no longer updates its data in Azure AD. For instance, if a device has its name changed or its operating system upgraded, these changes aren't reflected in Azure AD. A **mobile device management** (**MDM**) solution such as Microsoft Intune can offer this functionality.

Older versions of Windows, such as Windows 8.1, require the **Workplace Join for non-Windows 10 computers** package. They also require a **security token service** (**STS**). For this, configure the **Seamless Single Sign-on** option when PHS or PTA is used as the authentication method; the Azure AD authentication service will then be used as the STS. With **Active Directory Federation Services** (**AD FS**) as the authentication method, the AD FS servers act as STSs.

Older versions of Windows, and Windows 10 devices up to version 1607, require a Group Policy setting to start the process of attaching to the device object in Azure AD. Newer versions of Windows 10 no longer require the Group Policy setting and will try attaching to their corresponding device objects unless the Group Policy setting is specifically configured as **Disabled**.

Older versions of Windows Server running Group Policy Management or Windows servers utilizing an out-of-date centralized policy store might not show the **Register domain-joined computers as devices** setting but may instead show the **Register domain joined computer as device** setting. The label for the setting has been changed; however, this does not impact the effectiveness of the Group Policy.

Azure AD join can also be configured in **System Center Configuration Manager** (**ConfigMgr**). The setting in ConfigMgr overrules the setting in any GPO.

In terms of conditional access, a hybrid Azure AD-joined device may constitute a trusted device, just as with a compliant device that has earned its trust through health attestation and enforced configuration settings by the organization's MDM solution. Many organizations choose to use more relaxed access policies for users on compliant devices, but the question is whether a domain-joined device is as trustworthy as a compliant device.

Hybrid Azure AD join only needs to be configured on one Azure AD Connect installation. If you run multiple Azure AD Connect installations, perform the steps on the actively synchronizing Azure AD Connect installation (the non-staging-mode server).

Configuring device writeback

This recipe shows how to configure the **Device writeback** feature in Azure AD Connect.

Getting ready

To configure the **Device writeback** feature in Azure AD Connect, you need to know the following characteristics of your organization:

- In which forest are we going to write device objects? (Only applicable if your organization has multiple forests in scope for Azure AD Connect.)
- Which Azure AD Connect installation is the non-staging-mode server? (Only applicable if your organization has multiple Azure AD Connect servers.)

The **Device writeback** feature requires Azure AD Premium P1 licenses or a Microsoft license that includes the P1 license, such as Azure AD Premium P2, **Enterprise Mobility + Security** (**EMS**) E3, EMS A3, Microsoft 365 E3, Microsoft 365 E5, and Microsoft 365 Business Premium licenses.

To configure device writeback in Azure AD Connect, you need to sign in with a domain account that is configured as a local administrator account on a server dedicated to Azure AD Connect. As part of the process, credentials for the following accounts need to be specified:

- An account in Active Directory that is a member of the **Enterprise Admins** group
- An account in Azure AD that has the **Global administrator** or **Hybrid Identity administrator** role assigned

If the organization uses the Azure AD PIM feature, activate the **Global administrator** or **Hybrid Identity administrator** role in advance.

Azure AD Connect needs to be initially configured, although the chosen authentication method is inconsequential as the **Device writeback** feature works with PHS, PTA, and federation.

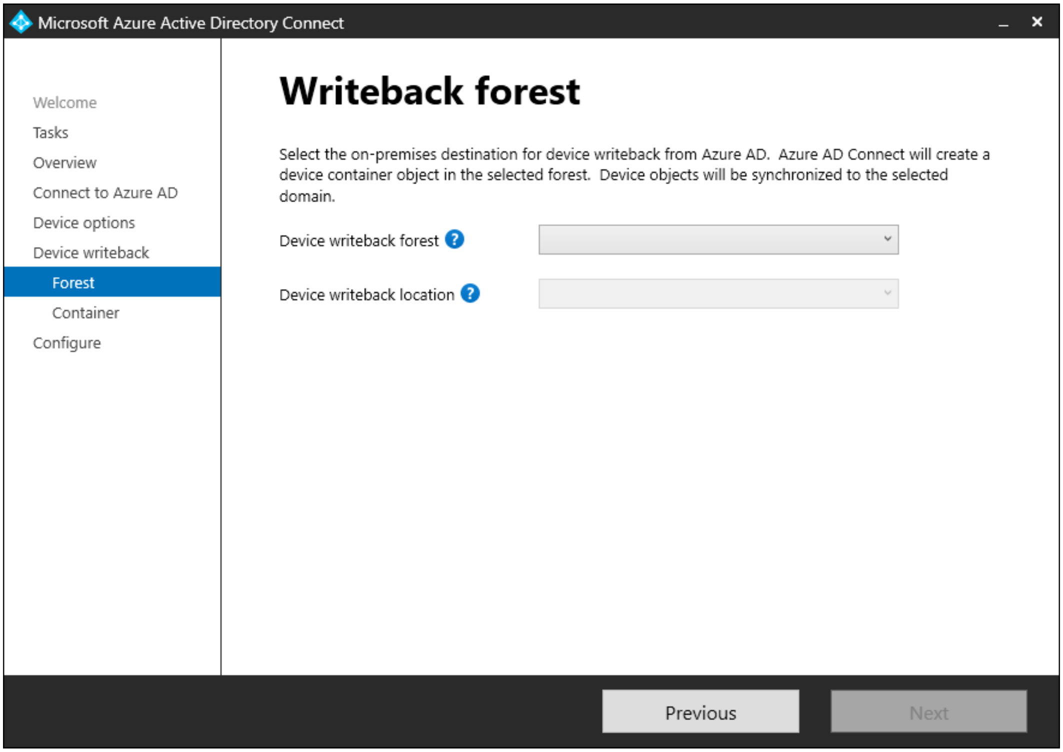
The Active Directory schema version needs to be Windows Server 2012 R2 (level 69), or higher.

How to do it...

Perform the following steps on the Windows Server running Azure AD Connect:

1. Open **Azure AD Connect** from the desktop. The **Microsoft Azure Active Directory Connect** window appears.
2. On the **Welcome to Azure AD Connect** screen, click **Configure**.
3. On the **Additional tasks** screen, select the **Configure device options** ribbon.
4. Click **Next**.
5. On the **Overview** screen, click **Next**.
6. On the **Connect to Azure AD** screen, sign in with an Azure AD-based account with the **Global administrator** or **Hybrid Identity administrator** role assigned.
7. Perform MFA when prompted.
8. On the **Device options** screen, select **Configure device writeback**.
9. Click **Next**.

10. On the **Writeback forest** screen, select the on-premises device writeback forest from the drop-down list:



Writeback forest

Select the on-premises destination for device writeback from Azure AD. Azure AD Connect will create a device container object in the selected forest. Device objects will be synchronized to the selected domain.

Device writeback forest

Device writeback location

Figure 15.16 – Writeback forest screen

11. Click **Next**.
12. On the **Device container** screen, select the **I will provide Enterprise Admin credentials.** option and enter credentials.
13. On the **Ready to configure** screen, click **Configure**.
14. On the **Configuration complete** screen, click **Exit** to close the **Microsoft Azure Active Directory Connect** window and to restart synchronization.

How it works…

Device writeback is an optional feature.

When people register their devices with Azure AD, their actions create device objects in Azure AD. In AD FS, these objects can be used in claims issuance rules. This way, when an Azure AD-joined device is used outside of one of your organization's locations, a claims issuance rule can be created to not require MFA, even though the device is external and other outside users are still required to perform MFA.

For the **Device writeback** feature, Azure AD is the source, and Active Directory on-premises is the destination.

Device writeback does not support multi-domain and multi-forest topologies, as devices must be located in the same forest as the users. Administrators can only define one domain to write back devices to.

When configuring device writeback, Azure AD Connect creates a `RegisteredDevices` container in the selected Active Directory domain to store written-back devices.

The objects placed in the `RegisteredDevices` container are of the `mS-DS-Device` type. This type of object does not have an `objectSid` attribute and therefore cannot be used on-premises other than in claims issuance rules in AD FS.

Device writeback only needs to be configured on one Azure AD Connect installation. If you run multiple Azure AD Connect installations, perform the steps on the actively synchronizing Azure AD Connect installation (the non-staging-mode server). However, the service accounts for staging-mode Azure AD Connect servers need to be configured with the same permissions as the service account of the actively synchronizing Azure AD Connect installation. The **I will configure it using the PowerShell script before continuing.** option on the **Device Container** screen results in a `CreateDeviceContainer.ps1` file with hints on how to configure the service account(s) of the staging-mode server(s):

```
$adConnectorAccount = "lucernpub.com\MSOL_9eee26c52012"
$registeredDevicesDN =
"CN=RegisteredDevices,DC=LucernPub,DC=com"
$userAcl = $adConnectorAccount + ":GRGWCCDCSDDT"
dsacls.exe $registeredDevicesDN /G $userAcl /I:T > $null
```

Replace the values for the AD Connector account (`MSOL_9eee26c52012`) and domain name (both `lucernpub.com` and `DC=LucernPub,DC=com`) in the preceding example.

Configuring password writeback

As an addition to the **Self-service Password Reset** and **Change Password** functionality in Azure AD, this recipe shows how to configure the **Password writeback** feature in Azure AD Connect.

Getting ready

To configure the **Password writeback** feature in Azure AD Connect, you need to know the following about your organization:

- Does my organization allow employees to reset their passwords from outside the organization?

To delegate permissions to Azure AD Connect service accounts, sign in with an account that is a member of the **Enterprise Admins** group in the Active Directory forest for which you are configuring password writeback to a Windows Server that has the **Active Directory Users and Computers** remote server administration tool installed.

To configure the **Password writeback** feature within Azure AD Connect, sign in with an account that is a local administrator account on a server dedicated to Azure AD Connect. As part of the process, credentials for the following accounts need to be specified:

- An account in Active Directory that is a member of the **Enterprise Admins** group
- An account in Azure AD that has the **Global administrator** or **Hybrid Identity administrator** role assigned

If the organization uses the Azure AD PIM feature, activate the **Global administrator** or **Hybrid Identity administrator** role in advance.

The **Password writeback** functionality requires Azure AD Premium P1 licenses or a Microsoft license that includes the P1 license, such as Azure AD Premium P2, EMS E3, EMS A3, Microsoft 365 E3, Microsoft 365 E5, and Microsoft 365 Business Premium licenses.

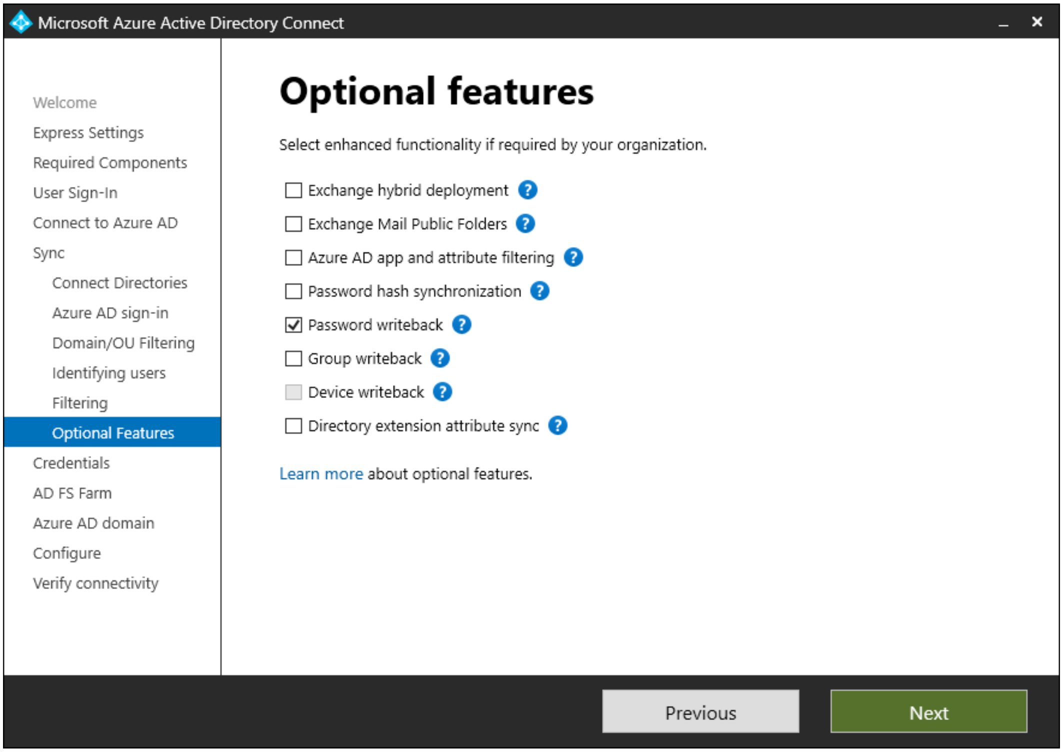
How to do it...

Configuring the **Password writeback** feature can be handled in two scenarios:

1. When configuring Azure AD Connect initially
2. When reconfiguring Azure AD Connect

Configuring Azure AD Connect initially

When initially configuring Azure AD Connect, ensure to click **Customize** on the **Express Settings** screen of Azure AD Connect. You encounter the **Optional features** screen, regardless of whether you are configuring Azure AD Connect for PHS, PTA, or federation:



Optional features

Select enhanced functionality if required by your organization.

□ Exchange hybrid deployment
□ Exchange Mail Public Folders
□ Azure AD app and attribute filtering
□ Password hash synchronization
☑ Password writeback
□ Group writeback
□ Device writeback
□ Directory extension attribute sync

Learn more about optional features.

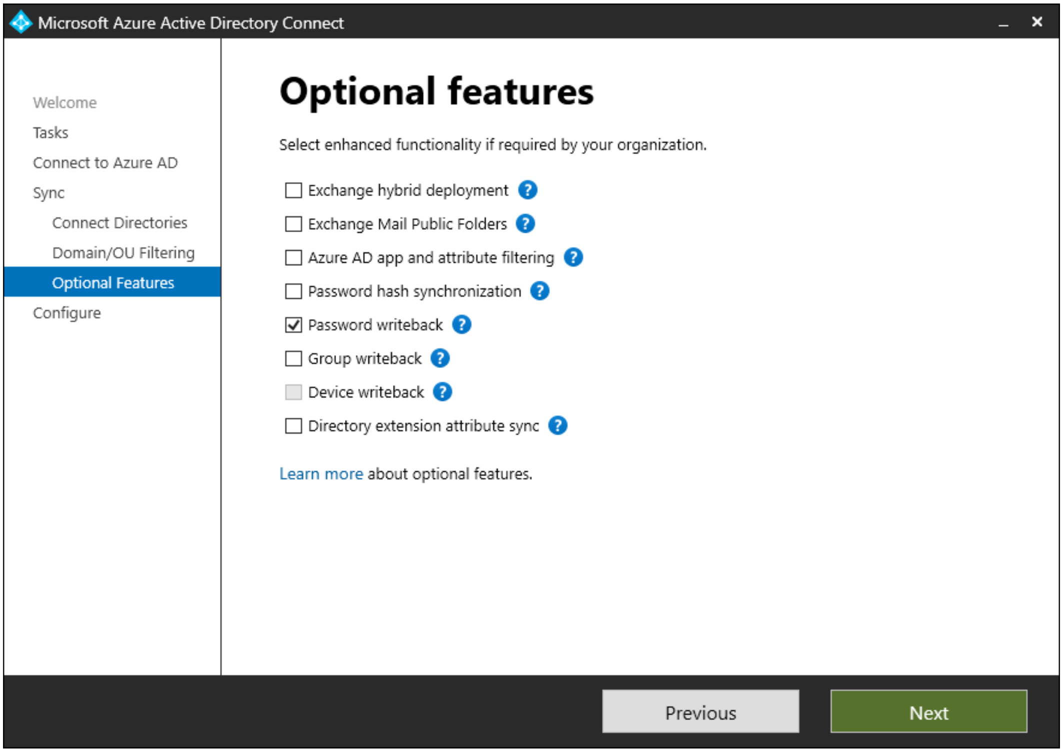
Figure 15.17 – Optional features screen

1. On the **Optional features** screen, select the **Password writeback** option.
2. Press **Next**.
3. On the **Ready to configure** screen, click **Configure**.
4. On the **Configuration complete** screen, click **Exit** to close the **Microsoft Azure Active Directory Connect** window and to start the initial synchronization.

Reconfiguring Azure AD Connect

Perform the following steps to reconfigure Azure AD Connect with password writeback:

1. Open **Azure AD Connect** from the desktop. The **Microsoft Azure Active Directory Connect** window appears.
2. On the **Welcome to Azure AD Connect** screen, click **Configure**.
3. On the **Additional tasks** screen, select the **Customize synchronization options** ribbon.
4. Click **Next**.
5. On the **Connect to Azure AD** screen, sign in with an Azure AD-based account with the **Global administrator** or **Hybrid Identity administrator** role assigned.
6. Perform MFA when prompted.
7. On the **Connect your directories** screen, click **Next**.
8. On the **Domain and OU filtering** screen, click **Next**.
9. On the **Optional features** screen, select the **Password writeback** option:



Optional features

Select enhanced functionality if required by your organization.

☐ Exchange hybrid deployment
☐ Exchange Mail Public Folders
☐ Azure AD app and attribute filtering
☐ Password hash synchronization
☑ Password writeback
☐ Group writeback
☐ Device writeback
☐ Directory extension attribute sync

Learn more about optional features.

Figure 15.18 – Selecting the Password writeback feature on the Optional features screen

10. Press **Next**.
11. On the **Ready to configure** screen, click **Configure**.
12. On the **Configuration complete** screen, click **Exit** to close the **Microsoft Azure Active Directory Connect** window and to restart synchronization.

How it works…

Password writeback is an optional feature.

The **Password writeback** functionality in Azure AD Connect offers add-on functionality to the **Self-service Password Reset** and **Change Password** functionality in Azure AD for organizations that use both Active Directory and Azure AD in a hybrid identity setup.

> **Note**
>
> When **Password writeback** is not enabled in Azure AD Connect or when accounts are not in scope for Azure AD Connect, people who use the **Self-service Password Reset** functionality or the **Change Password** functionality in either their MyProfile or Office 365 experience receive an error that passwords can't be changed right now.

With **Password writeback**, Azure AD places password reset requests on Azure Service Bus for Azure AD Connect to pick up. Azure AD Connect executes a password reset request, followed by a password change request against the on-premises domain controller that holds the **Primary Domain Controller Emulator** (**PDCE**) **Flexible Single Master Operations** (**FSMO**) role, and signals the result back to Azure AD, by placing it back on the Azure Service Bus. For password changes, only a password change request is processed in the preceding process. This way, all password and account lock-out policies apply to self-service password reset requests and change password requests.

Password writeback only needs to be configured on one Azure AD Connect installation. If you run multiple Azure AD Connect installations, perform the steps on the actively synchronizing Azure AD Connect installation (the non-staging-mode server).

AD Connector accounts that Azure AD Connect automatically provisions are automatically delegated the right permissions. When using your own configured AD Connector accounts, provide accounts with the permissions to reset passwords, change passwords, and write the `pwdLastSet` attribute on OUs with user objects in the scope of Azure AD Connect.

Configuring group writeback

This recipe shows how to enable the **Group writeback** feature in Azure AD Connect.

Getting ready

To configure group writeback in Azure AD Connect, you need to know the following characteristics of your organization:

- In which OU are we going to write back group objects?
- Which accepted domain name will be appended to Office 365 groups? (Only applicable if your organization has multiple **Domain Name System** (**DNS**) domain names and accepted domains.)

To configure the **Group writeback** feature in Azure AD Connect, you need to sign in with an account that is a local administrator account on a server dedicated to Azure AD Connect. As part of the process, credentials for the following accounts need to be specified:

- An account in Active Directory that is a member of the **Enterprise Admins** group
- An account in Azure AD that has the **Global administrator** or **Hybrid Identity administrator** role assigned

If the organization uses the Azure AD PIM feature, activate the **Global administrator** or **Hybrid Identity administrator** role in advance.

Azure AD Connect needs to be initially configured, although the chosen authentication method is inconsequential as the **Group writeback** feature works with PHS, PTA, and federation.

The **Exchange Hybrid** option needs to have already been selected.

If your organization needs groups written back to an OU that doesn't exist yet, create one.

The **Password writeback** functionality requires the following:

- A configured hybrid deployment between your Exchange Server organization and Exchange Online
- Azure AD Premium P1 licenses or a Microsoft license that includes the P1 license, such as Azure AD Premium P2, EMS E3, EMS A3, Microsoft 365 E3, Microsoft 365 E5, and Microsoft 365 Business Premium licenses

How to do it…

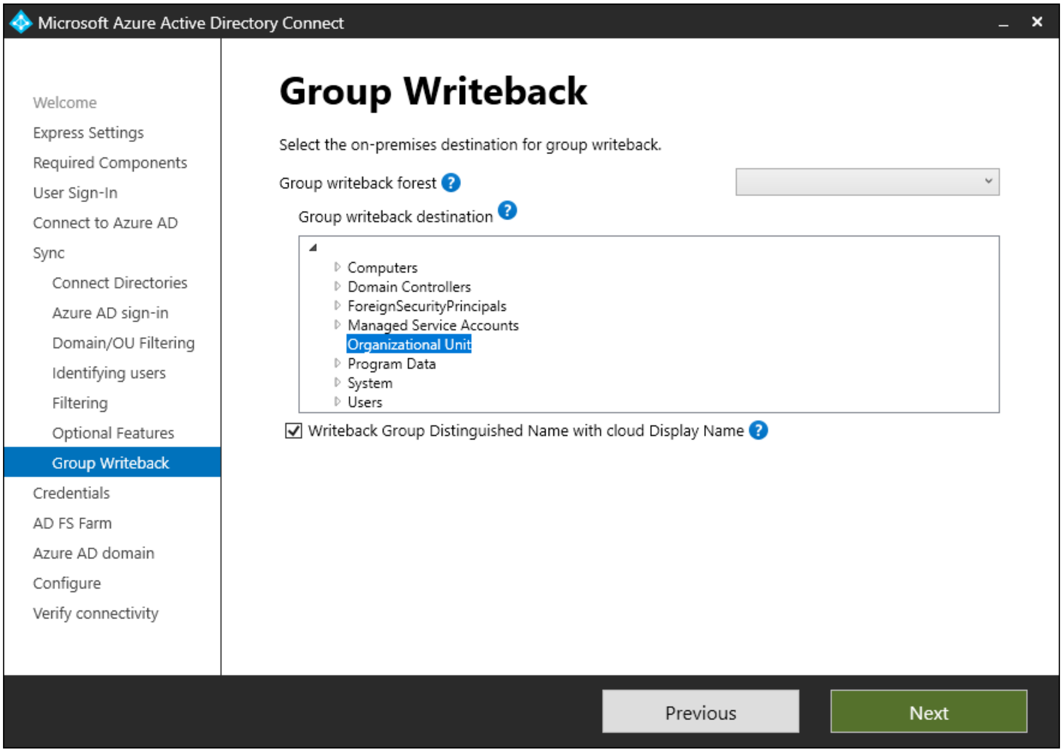
Configuring the **Group writeback** feature can be handled in two scenarios:

1. When configuring Azure AD Connect initially
2. When reconfiguring Azure AD Connect

Configuring Azure AD Connect initially

When initially configuring Azure AD Connect, ensure to click **Customize** on the **Express Settings** screen of Azure AD Connect. You encounter the **Optional Features** screen, regardless of whether you're configuring Azure AD Connect for PHS, PTA, or federation:

1. On the **Optional features** screen, select the **Group writeback** option.
2. Press **Next**.
3. On the **Group Writeback** screen, select the Active Directory OU to store group objects that will be written back:



Group Writeback

Select the on-premises destination for group writeback.

Group writeback forest

Group writeback destination

- Computers
- Domain Controllers
- ForeignSecurityPrincipals
- Managed Service Accounts
- Organizational Unit
- Program Data
- System
- Users

☑ Writeback Group Distinguished Name with cloud Display Name

Figure 15.19 – Group Writeback screen

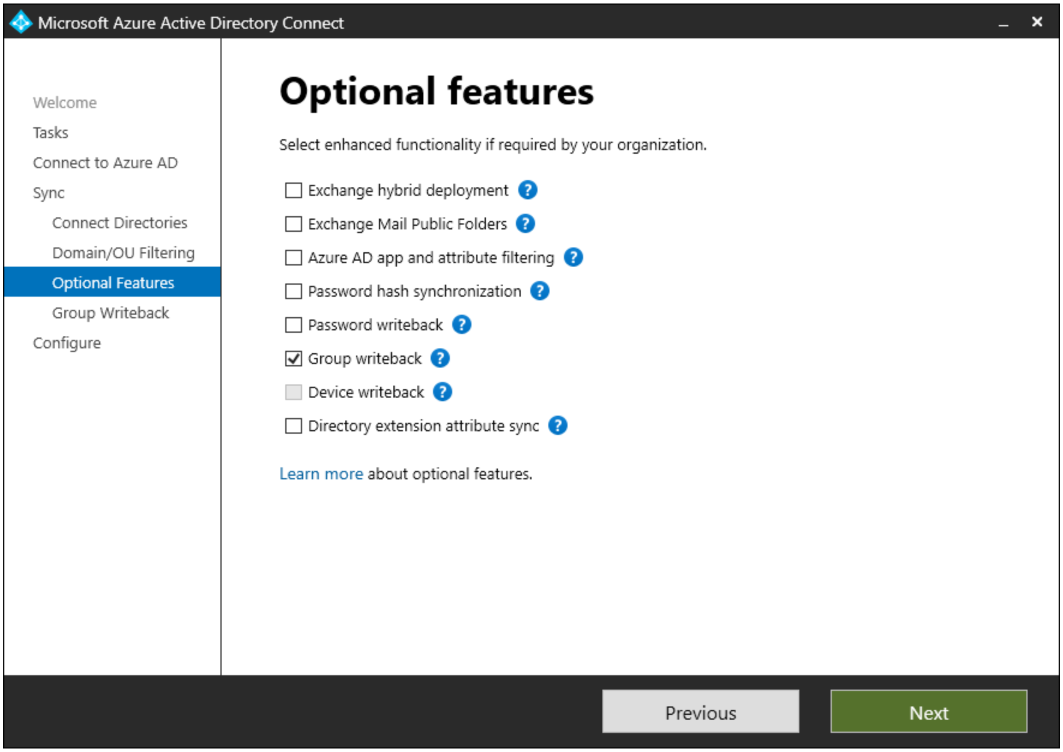
4. Optionally, select the **Writeback Group Distinguished Name with cloud Display Name** option.
5. Click **Next**.
6. On the **Ready to configure** screen, click **Configure**.
7. On the **Configuration complete** screen, click **Exit** to close the **Microsoft Azure Active Directory Connect** window and to start the initial synchronization.

Reconfiguring Azure AD Connect

Perform the following steps to reconfigure Azure AD Connect with the **Group writeback** feature:

1. Open **Azure AD Connect** from the desktop. The **Microsoft Azure Active Directory Connect** window appears.
2. On the **Welcome to Azure AD Connect** screen, click **Configure**.
3. On the **Additional tasks** screen, select the **Customize synchronization options** ribbon.
4. Click **Next**.
5. On the **Connect to Azure AD** screen, sign in with an Azure AD-based account with the **Global administrator** or **Hybrid Identity administrator** role assigned.
6. Perform MFA when prompted.
7. On the **Connect your directories** screen, click **Next**.
8. On the **Domain and OU filtering** screen, click **Next**.

9. On the **Optional features** screen, select the **Group writeback** option:



Microsoft Azure Active Directory Connect

Optional features

Select enhanced functionality if required by your organization.

- □ Exchange hybrid deployment
- □ Exchange Mail Public Folders
- □ Azure AD app and attribute filtering
- □ Password hash synchronization
- □ Password writeback
- ☑ Group writeback
- □ Device writeback
- □ Directory extension attribute sync

Learn more about optional features.

Figure 15.20 – Group writeback option on the Optional features screen

10. Click **Next**.
11. On the **Writeback** screen, select the Active Directory OU to store group objects that will be written back.
12. Click **Next**.
13. On the **Ready to configure** screen, click **Configure**.
14. On the **Configuration complete** screen, click **Exit** to close the **Microsoft Azure Active Directory Connect** window and to restart synchronization.

How it works...

Group writeback is an optional feature.

The **Group writeback** feature in Azure AD Connect offers add-on functionality to the **Microsoft 365 Groups** functionality in Azure AD for organizations that use both Active Directory and Azure AD in a hybrid exchange setup.

When **Group writeback** is enabled, Microsoft 365 groups are written back to Active Directory as mail-enabled distribution groups. This way, people in your organization with mailboxes on on-premises implementations of Exchange Server 2013 Cumulative Update 8, and newer versions of Exchange Server, can send email messages to these groups and receive email messages from these groups.

If an Active Directory forest does not have its schema extended with Exchange Server schema extensions, it is not eligible for the **Group writeback** feature, and the option will be grayed out in Azure AD Connect.

Exchange servers in your organization need to run the following versions of Exchange Server as a minimum:

- Exchange Server 2013 Cumulative Update 8
- Exchange Server 2016 Cumulative Update 1

Changing passwords for Azure AD Connect service accounts

This recipe shows how to manually change passwords for your Azure AD Connect service account(s).

Getting ready

To reconfigure Azure AD Connect, you need to sign in with an account that is a local administrator account on a server dedicated to Azure AD Connect. As part of the following steps, you need to enter credentials for these accounts:

- An account in Azure AD that has the **Global administrator** role assigned
- An account in Active Directory that is a member of the **Domain Admins** group, for each domain in which Seamless SSO is configured (only applicable when Seamless SSO is configured)

If the organization uses the Azure AD PIM feature, activate the **Global administrator** role in advance.

Azure AD Connect needs to be initially configured.

How to do it…

Perform the steps outlined next to change passwords for the following Azure AD Connect service accounts:

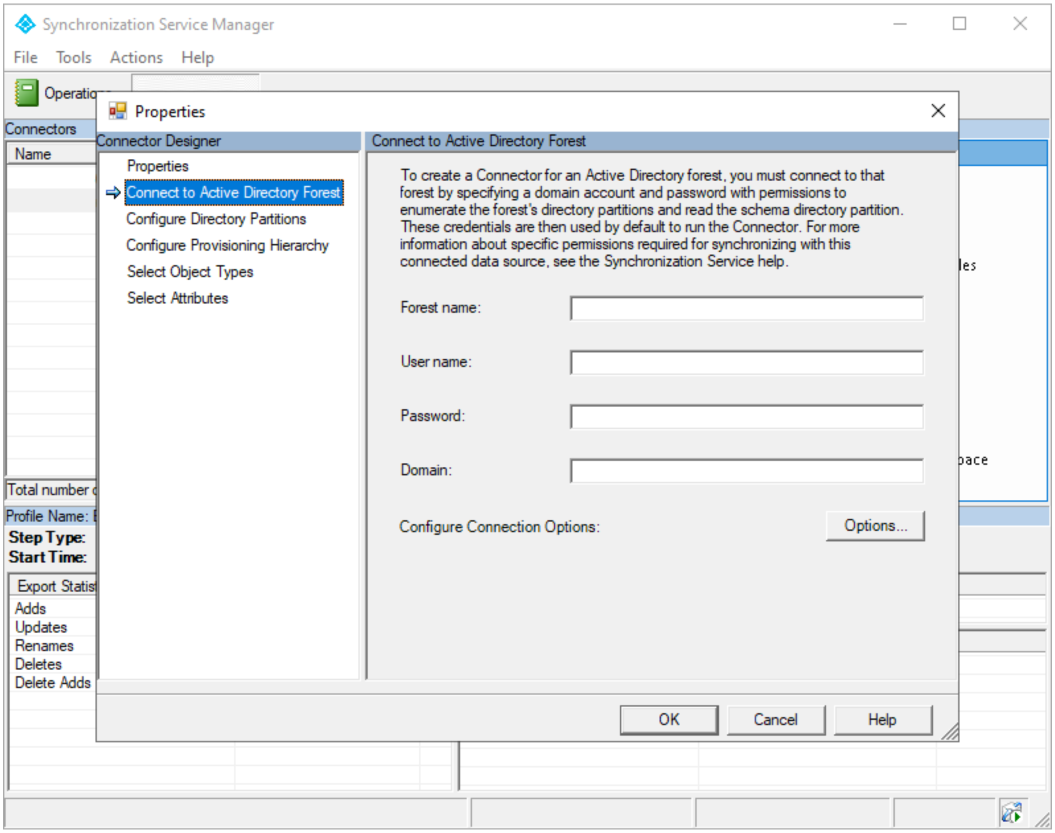
1. The AD Connector account—the service account connecting to Active Directory.
2. The Azure AD Connector account—the service account connecting to Azure AD
3. The computer account for Seamless SSO

Changing the password for the AD Connector account

Perform these steps to change the password for the AD Connector account:

1. Press Start.
2. Search for **Synchronization Service** and click its search result.
   The **Synchronization Service Manager** window appears.
3. Go to the **Connectors** tab.
4. Select an **AD Connector** type.
5. From the **Actions** menu, select **Properties**. The **Properties** window appears.

6. In the left navigation pane, select the **Connect to Active Directory Forest** node:



To create a Connector for an Active Directory forest, you must connect to that forest by specifying a domain account and password with permissions to enumerate the forest's directory partitions and read the schema directory partition. These credentials are then used by default to run the Connector. For more information about specific permissions required for synchronizing with this connected data source, see the Synchronization Service help.

Figure 15.21 – Properties of the AD Connector

7. Enter a new password for the AD Connector account in the **Password:** field.
8. Click **OK**. The **Synchronization Service Manager** pop-up window appears. Click **OK**.
9. Close the **Synchronization Service Manager** window.
10. Press Start again.
11. Search for **Services** and click its search result or run `services.msc`.
12. In the main pane, select the **Microsoft Azure AD Sync** service.
13. Right-click the service and select **Restart** from the menu.
14. Close the **Services** window.

Perform the preceding steps on all Azure AD Connect installations for all AD Connectors to reset their passwords.

Changing the password for the Azure AD Connector account

Perform the following steps to change the password for the Azure AD Connector account:

1. Right-click the Start button and select **Windows PowerShell (Admin)** from the menu.
2. Run the following lines of PowerShell:

   ```
   Import-Module ADSync
   Add-ADSyncAADServiceAccount
   ```

   A pop-up window appears.
3. Provide credentials of an account that has the **Global administrator** role assigned to it.
4. Perform MFA when prompted.

Perform the preceding steps on all Azure AD Connect installations.

Changing the password for the computer account for Seamless SSO

Perform the following steps to change the password for the `AzureADSSOACC` computer account in Active Directory:

1. Right-click the Start button and select **Windows PowerShell (Admin)** from the menu.
2. Run the following lines of PowerShell:

   ```
   Import-Module "C:\Program Files\Microsoft Azure Active
   Directory Connect\AzureADSSO.psd1"
   New-AzureADSSOAuthenticationContext
   ```

   A pop-up window appears.
3. Provide credentials of an account that has the **Global administrator** role assigned to it.
4. Click **OK**.
5. Perform MFA when prompted.
6. Run the following line of PowerShell in the same PowerShell session as the preceding lines:

   ```
   Get-AzureADSSOStatus | ConvertFrom-Json
   ```

This shows domains where Seamless SSO is configured and provides a list of domains to change the password in.

7. Run the following line of PowerShell in the same PowerShell session as the preceding lines:

   ```
   $cred = Get-Credentials
   ```

   A pop-up window appears.

8. Enter credentials of an account that is a member of the **Domain Admins** group in the Active Directory forest. The Active Directory domain in which to reset the password for the `AzureADSSOACC` computer object is derived from the account information.

9. Click **OK**.

10. Run the following line of PowerShell in the same PowerShell session as the preceding lines:

    ```
    Update-AzureADSSOForest -onPremCredentials $cred
    ```

    The preceding command updates the Kerberos decryption key for the `AzureADSSOACC` computer object and updates the information in Azure AD.

Perform *Steps 7-10.* for each of the domains from the output of *Step 6*.

How it works...

Microsoft recommends changing passwords for the AD Connector account and Azure AD Connector account at least every year, as this then makes it harder for attackers to gain and maintain a foothold in your organization's systems.

Microsoft recommends changing the password for the `AzureADSSOACC` computer object monthly.

By default, Azure AD Connect runs with a **Virtual Service Account** (**VSA**). This account does not need to have a password, so it doesn't need changing. Optionally, during the initial installation of Azure AD Connect, a **group Managed Service Account** (**gMSA**) can be configured to run Azure AD Connect. gMSAs change their passwords every 30 days, by default.

The AD Connector account

The service account to connect to Active Directory is stored for the Active Directory **Management Agent** (**MA**). After you successfully change the password for the service account (and thus pass the password requirements), change the account password in the configuration for the MA in the Synchronization Service Manager **user interface** (**UI**).

The Azure AD Connector account

The service account to connect to Azure AD is stored for the Azure AD MA. Azure AD Connect does not let you create a password for the service account itself but offers a Windows PowerShell cmdlet to take care of everything for you.

The computer account for Seamless SSO

When the optional **Enable single sign-on** setting is enabled in Azure AD Connect, it instructs Azure AD Connect to use Kerberos authentication using a specific computer object in the local Active Directory: `AzureADSSOACC`. This account is located in the default **Computers** container.

As the password for this account is also stored in Azure AD for the Azure AD authentication services to be able to decrypt the Kerberos packages, it requires a little more work to reset. Azure AD Connect offers a Windows PowerShell cmdlet to take care of this.

16
Hardening Azure AD

Azure AD is a Microsoft cloud-based **Identity and Access Management** (**IAM**) solution. Over the years, many features have been added to the platform to address the needs of its millions of customers worldwide. Many of these features were security features that weren't turned on by default. For newer Azure AD tenants, some of the security features are turned on by default.

This chapter shows how to configure an Azure AD tenant to increase its confidentiality, integrity, and availability. Some of these features and functionalities might hinder productivity, so you might not want to make changes without communicating these first.

The recipes in this chapter start with recipes any administrator can apply to harden any Azure AD tenant. Then, recipes are covered that require Azure AD Premium P1 licenses. At the end of the chapter, two recipes require Azure AD Premium P2 licenses, and one recipe requires at least one **Enterprise Mobility + Security** (**EMS**) E5 license.

The following recipes are covered in this chapter:

- Setting contact information
- Preventing non-privileged users from accessing the Azure portal
- Viewing all privileged users in Azure AD
- Preventing users from registering or consenting to apps

- Preventing users from inviting guests
- Allowing and blocking invitations for Azure AD **Business to Business** (**B2B**)
- Configuring Azure AD join and Azure AD registration
- Configuring Intune auto-enrollment upon Azure AD join
- Choosing between Security defaults and **Conditional Access**
- Configuring Conditional Access
- Accessing Azure AD Connect Health
- Configuring Azure AD Connect Health for **AD Federation Services** (**AD FS**)
- Configuring Azure AD Connect Health for **AD Domain Services** (**AD DS**)
- Configuring Azure AD **Privileged Identity Management** (**PIM**)
- Configuring Azure AD Identity Protection
- Implementing Defender for Identity

> **Note**
>
> We have gone to great lengths to ensure that the contents of this chapter are as future-proof as possible. However, as Azure AD is a cloud service, screen texts, panes, blades, and functionality in this chapter may change between the time of writing and when you read it, but the strategy and direction of Microsoft will still hold true.

Setting contact information

This recipe shows how to set contact information for the tenant.

Getting ready

When you work in a team, create a distribution list to receive important updates for the Azure AD tenant.

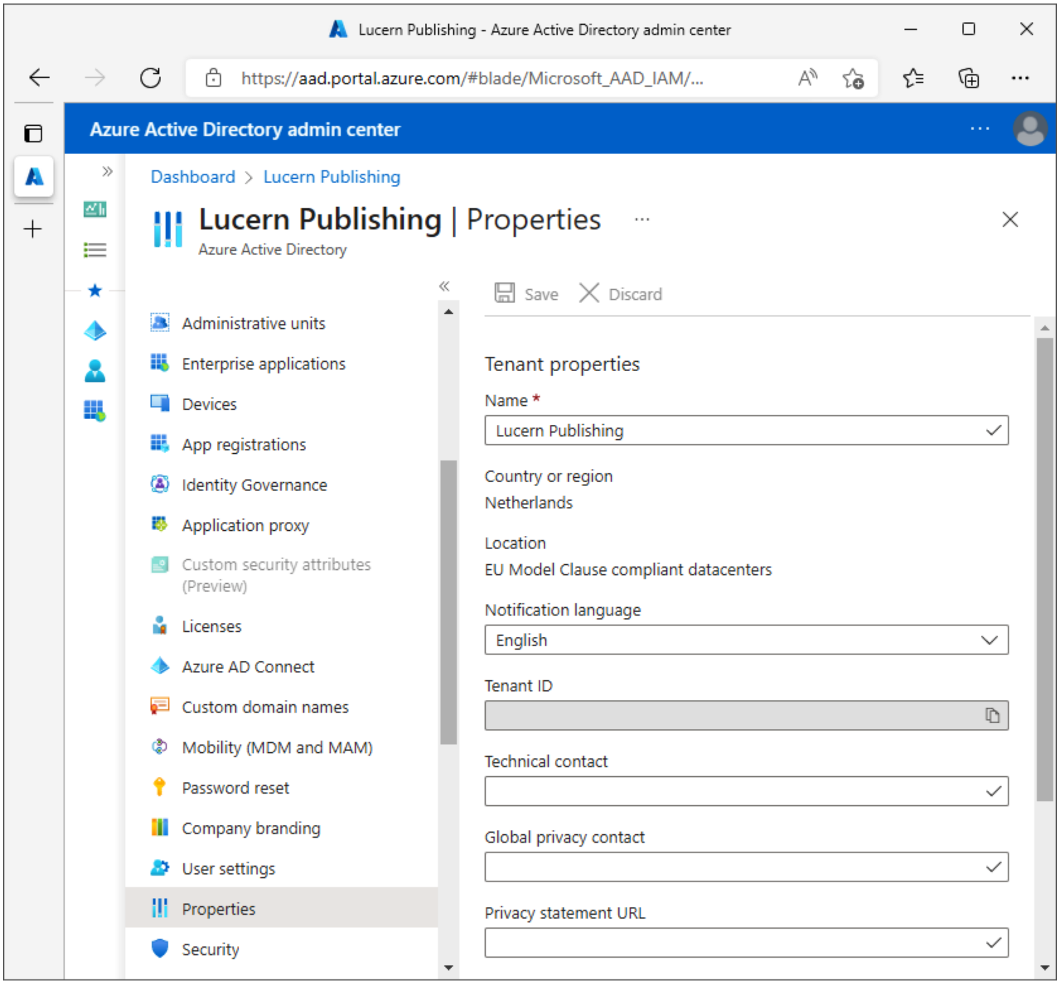
Find out who the contact is for privacy within the organization and where the organization has publicly published its privacy policy.

To complete this recipe, sign in to the Azure AD tenant with an account that has the **Global administrator** role assigned. If the organization uses the Azure AD PIM feature, activate the **Global administrator** role in advance.

How to do it...

Perform the following steps to set contact information:

1. Navigate your browser to `https://aad.portal.azure.com`.
2. Sign in with an account in Azure AD that has the **Global administrator** role assigned.
3. Perform **multi-factor authentication** (**MFA**) when prompted.
4. In the left navigation pane, click **Azure Active Directory**.
5. In the **Azure Active Directory** navigation pane, click **Properties** to navigate to the **Properties** pane for the tenant:



Tenant properties

Name *
Lucern Publishing

Country or region
Netherlands

Location
EU Model Clause compliant datacenters

Notification language
English

Tenant ID

Technical contact

Global privacy contact

Privacy statement URL

Figure 16.1 – Properties pane

6. In the **Properties** pane, fill in the information for your organization.
7. Click **Save** at the top of the **Properties** pane.

How it works...

Azure AD tenants may experience hostile takeovers; attackers brute-force passwords for users or reuse leaked credentials and then make their way to privileged accounts to take over the tenant. The percentage of tenants that experience this problem is tiny, but the scale of Azure AD means it happens to a couple of thousand tenants per day.

When Microsoft detects a tenant is compromised, it contacts the owner of the tenant to help them regain control. However, for a lot of tenants, the contact information is not provided or it's out of date. Information for the **Technical contact** entry on the **Properties** pane is only used for these purposes.

Information for the **Global privacy contact** and **Privacy statement URL** entries are used when a person consents to permissions for an app your organization owns.

See also

To set up an Azure AD tenant, see the *Signing up for Azure AD* recipe in *Chapter 14*, *Handling Authentication in a Hybrid World (AD FS, PHS, PTA, and DSSO)*.

Preventing non-privileged users from accessing the Azure portal

This recipe shows how to restrict access to the Azure portal for non-privileged users to make it only available to privileged users.

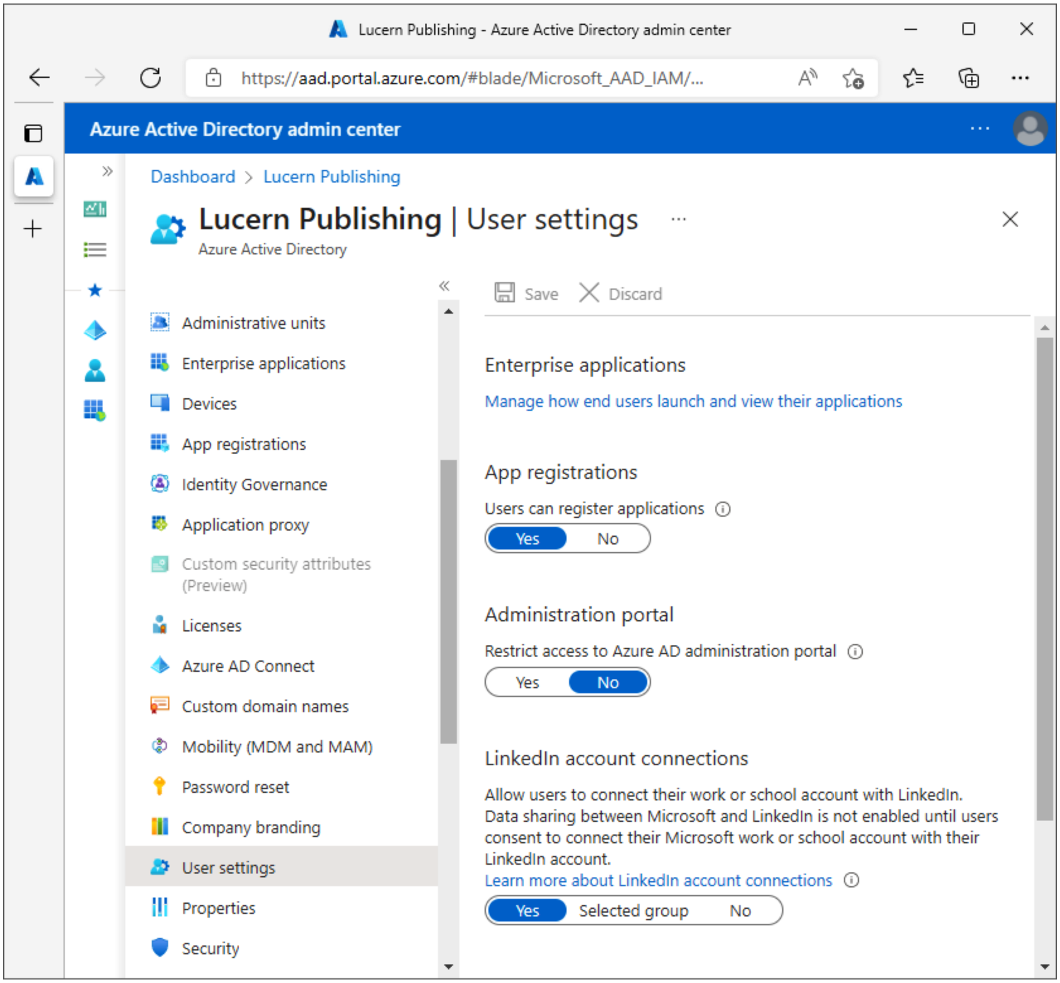
Getting ready

To complete this recipe, sign in to the Azure AD tenant with an account that has the **Global administrator** role assigned. If the organization uses the Azure AD PIM feature, activate the **Global administrator** role in advance.

How to do it...

Perform these steps to restrict access of non-privileged users to the Azure AD portal:

1. Navigate your browser to `https://aad.portal.azure.com`.
2. Sign in with an account in Azure AD that has the **Global administrator** role assigned.
3. Perform MFA when prompted.
4. In the left navigation pane, click **Azure Active Directory**.
5. In the **Azure Active Directory** navigation pane, click **User settings** to go to the **User settings** pane for the tenant:



Lucern Publishing - Azure Active Directory admin center

https://aad.portal.azure.com/#blade/Microsoft_AAD_IAM/...

Azure Active Directory admin center

Dashboard > Lucern Publishing

Lucern Publishing | User settings

Azure Active Directory

Save Discard

Enterprise applications

Manage how end users launch and view their applications

App registrations

Users can register applications ⓘ

[✓ Yes] No

Administration portal

Restrict access to Azure AD administration portal ⓘ

Yes [✓ No]

LinkedIn account connections

Allow users to connect their work or school account with LinkedIn. Data sharing between Microsoft and LinkedIn is not enabled until users consent to connect their Microsoft work or school account with their LinkedIn account.

Learn more about LinkedIn account connections ⓘ

[✓ Yes] Selected group No

Figure 16.2 – User settings pane

6. In the **User settings** pane, change the **Restrict access to Azure AD administration portal** setting to **Yes**.
7. Click **Save** at the top of the **User settings** pane.

How it works...

In Active Directory, by default, any user can access the contents of a directory to look up objects. Password (hashes) are off limits, of course; some organizations have limited this default access.

As many organizations synchronize user objects from Active Directory to Azure AD, many Azure AD tenants contain personal data of people inside organizations. By default, anyone can sign in to the Azure portal with any browser and view a list of users, including many of their attributes, except—of course—passwords and password hashes. Because Azure AD is a cloud-based resource, by default, anyone may access this information from any location.

This is the big difference with Active Directory: looking up objects in Azure AD doesn't require a connection to an internal network, whereas Active Directory does.

For regulatory compliance with the **General Data Protection Regulation** (**GDPR**) and other privacy regulations, organizations may want to restrict the Azure AD administration portal experience.

Changing the default **No** value to **Yes** for **Restrict access to Azure AD administration portal** restricts all non-privileged users from accessing the administration portal but does not restrict such access using PowerShell or another client, such as Visual Studio.

Privileges in Azure AD are defined by roles. When a person requires access to the Azure portal experience, assign the person one or more roles.

Viewing all privileged users in Azure AD

This recipe shows two ways to view all privileged users in Azure AD through the Microsoft Graph **application programming interface** (**API**).

Getting ready

To complete this recipe, sign in to the Azure AD tenant with an account that has the **Global administrator** role assigned. If the organization uses the Azure AD PIM feature, activate the **Global administrator** role in advance.

When using the PowerShell method, install the `Microsoft.Graph` PowerShell module first. Use the following line of PowerShell on a Windows or Windows Server system that runs Windows PowerShell `5.0` or higher in an elevated Windows PowerShell window:

```
Install-Module Microsoft.Graph
```

Press `Yes` twice.

How to do it…

You can view all privileged users in Azure AD by executing the following lines of PowerShell on the device where you installed the `Microsoft.Graph` PowerShell module:

```
Import-Module Microsoft.Graph
Connect-MgGraph -scopes RoleManagement.Read.Directory,User.
ReadBasic.All
```

Sign in with an account in Azure AD that has the **Global administrator** role assigned. Perform MFA when prompted.

On the **Permissions requested** screen, click **Accept** to consent to the requested Graph API permissions. Then, run the following lines of PowerShell:

```
$roles = Get-MgDirectoryRole
ForEach($role in $roles) {
 $users = Get-MgDirectoryRoleMember -DirectoryRoleId $role.Id
 ForEach($user in $users) {
 $Hashtable =[ordered]@{
  RoleName = $role.DisplayName
  userPrincipalName = $user.AdditionalProperties.
userPrincipalName
  }
  [PSCustomObject]$Hashtable
 }
}
```

The output is a list of Azure AD privileged roles and `userPrincipalName` attributes of user accounts with the roles.

How it works...

The Azure AD portal provides information on roles and their members, but it doesn't provide a clear overview of all the roles in use unless the Azure AD tenant is equipped with Azure AD Premium P2 licenses. This makes it hard to get a complete overview of all users with privileged Azure AD roles.

The lines of PowerShell in this recipe show this information. The first line creates an array of roles. Then, each role is cycled through for members, which are then added to a hash table. The last line of PowerShell displays the contents of the hash table.

Preventing users from registering or consenting to apps

This recipe shows how to prevent users from consenting to apps.

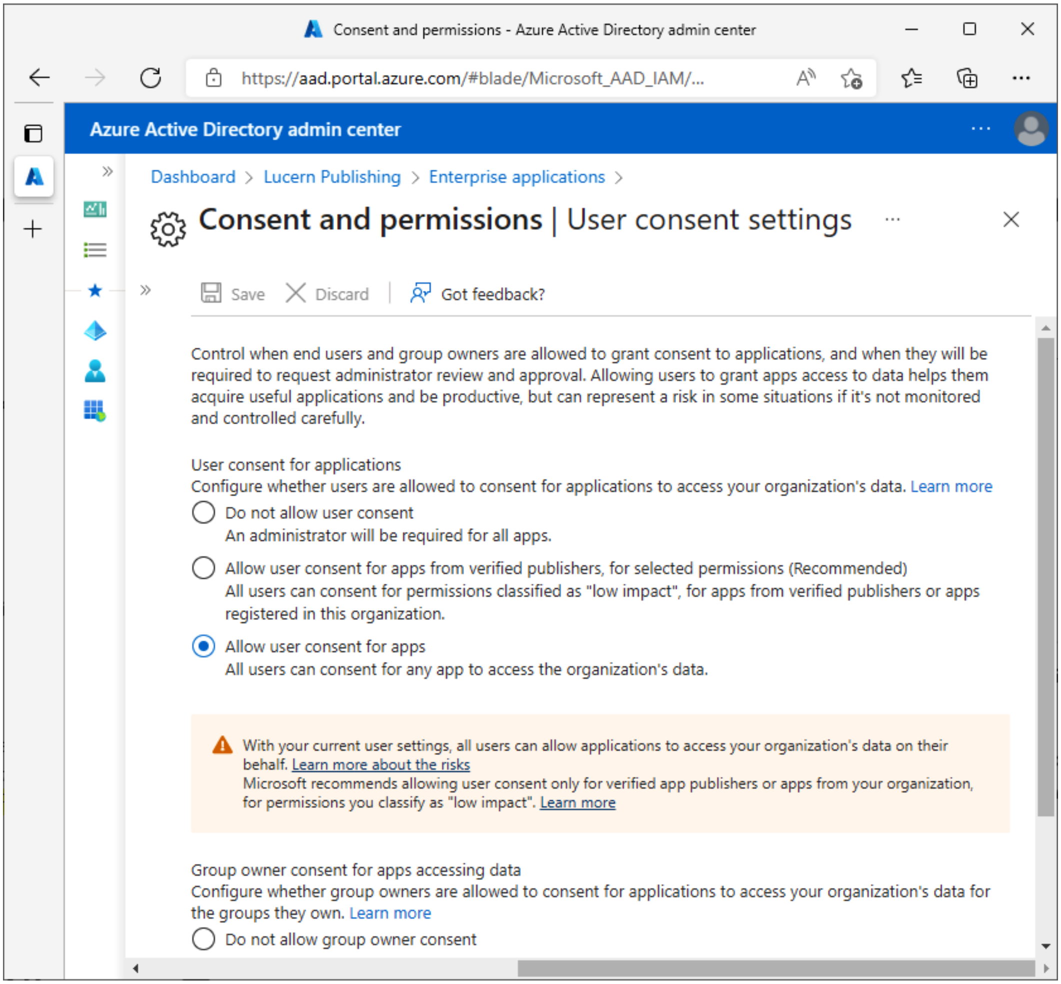
Getting ready

To complete this recipe, sign in to the Azure AD tenant with an account that has the **Global administrator** role assigned to it. If the organization uses the Azure AD PIM feature, activate the **Global administrator** role in advance.

How to do it...

Perform the following steps to prevent users from consenting to apps:

1. Navigate your browser to `https://aad.portal.azure.com`.
2. Sign in with an account in Azure AD that has the **Global administrator** role assigned.
3. Perform MFA when prompted.
4. In the left navigation pane, click **Azure Active Directory**.
5. In the **Azure Active Directory** navigation pane, click **User settings**.
6. In the **User settings** pane, change the **Users can register applications** setting to **No**.
7. Click **Save** at the top of the pane.
8. In the left navigation pane, click **Azure Active Directory** again.
9. In the **Azure Active Directory** navigation pane, select **Enterprise applications**.
10. In the **Enterprise applications** pane, click **Consent and permissions** to navigate to the **Consent and permissions | User consent settings** pane for the tenant:



Consent and permissions - Azure Active Directory admin center
https://aad.portal.azure.com/#blade/Microsoft_AAD_IAM/...
Azure Active Directory admin center
Dashboard > Lucern Publishing > Enterprise applications >
Save
Discard
Got feedback?

Consent and permissions | User consent settings

Control when end users and group owners are allowed to grant consent to applications, and when they will be required to request administrator review and approval. Allowing users to grant apps access to data helps them acquire useful applications and be productive, but can represent a risk in some situations if it's not monitored and controlled carefully.

User consent for applications
Configure whether users are allowed to consent for applications to access your organization's data. Learn more

○ Do not allow user consent
An administrator will be required for all apps.

○ Allow user consent for apps from verified publishers, for selected permissions (Recommended)
All users can consent for permissions classified as "low impact", for apps from verified publishers or apps registered in this organization.

● Allow user consent for apps
All users can consent for any app to access the organization's data.

With your current user settings, all users can allow applications to access your organization's data on their behalf. Learn more about the risks
Microsoft recommends allowing user consent only for verified app publishers or apps from your organization, for permissions you classify as "low impact". Learn more

Group owner consent for apps accessing data
Configure whether group owners are allowed to consent for applications to access your organization's data for the groups they own. Learn more

○ Do not allow group owner consent

Figure 16.3 – User settings for Enterprise applications

11. In the **Consent and permissions | User consent settings** pane, select the **Do not allow user consent** and the **Do not allow group owner consent** options.
12. Click **Save** at the top of the pane.

How it works...

By default, people with user accounts in an Azure AD tenant can consent to any application they want to use with their Azure AD account. For administrators, this might be an unwanted scenario, because they might want to do the following:

- Provide administrator consent to certain applications so that end users don't have to.
- Lock down the Azure AD tenant by only allowing applications through administrator consent.
- Avoid application sprawl in the Azure AD tenant.

To prevent users from consenting to apps and other ways to add apps, the Azure AD tenant should be turned off in two locations inside the Azure Active Directory admin center. The first option prevents users from registering custom-developed applications for use with the Azure AD tenant, while the second option prevents users from consenting to third-party multi-tenant applications that access user data in the organization's tenant.

When turned off, the administrative overhead of Azure AD increases. This should especially be taken into consideration in large environments. Application management may be delegated to people in the **Application administrator** role.

Preventing users from inviting guests

This recipe shows how to prevent people in the Azure AD tenant from inviting guests through Azure AD B2B.

Getting ready

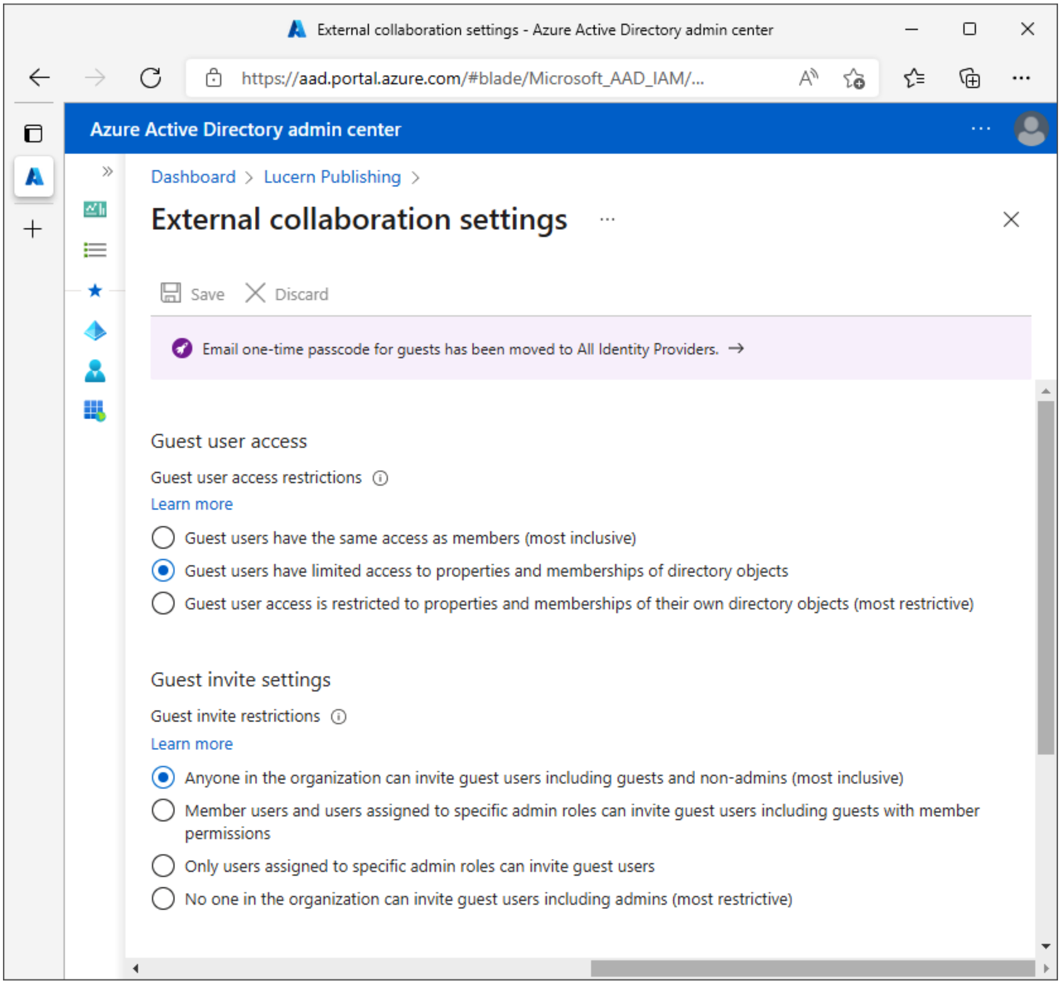
To complete this recipe, sign in to the Azure AD tenant with an account that has the **Global administrator** role assigned to it. If the organization uses the Azure AD PIM feature, activate the **Global administrator** role in advance.

How to do it...

Perform these steps to prevent users from inviting guests:

1. Navigate your browser to `https://aad.portal.azure.com`.
2. Sign in with an account in Azure AD that has the **Global administrator** role assigned.

3. Perform MFA when prompted.
4. In the left navigation pane, click **Azure Active Directory**.
5. In the **Azure Active Directory** navigation pane, click **User settings**.
6. Follow the **Manage external collaboration settings** link to navigate to the **External collaboration settings** pane:


External collaboration settings - Azure Active Directory admin center
https://aad.portal.azure.com/#blade/Microsoft_AAD_IAM/...
Azure Active Directory admin center
Dashboard > Lucern Publishing >
Save
Discard

External collaboration settings

Email one-time passcode for guests has been moved to All Identity Providers. →

Guest user access

Guest user access restrictions ⓘ

Learn more

○ Guest users have the same access as members (most inclusive)

● Guest users have limited access to properties and memberships of directory objects

○ Guest user access is restricted to properties and memberships of their own directory objects (most restrictive)

Guest invite settings

Guest invite restrictions ⓘ

Learn more

● Anyone in the organization can invite guest users including guests and non-admins (most inclusive)

○ Member users and users assigned to specific admin roles can invite guest users including guests with member permissions

○ Only users assigned to specific admin roles can invite guest users

○ No one in the organization can invite guest users including admins (most restrictive)

Figure 16.4 – External collaboration settings pane

7. In the **External collaboration settings** pane, under **Guest invite settings**, select the **Only users assigned to specific admin roles can invite guest users** option to move away from the default **Anyone in the organization can invite guest users including guests and non-admins (most inclusive)** option.
8. Click **Save** at the top of the pane.

How it works...

People in your organization can invite other people outside of your organization to collaborate in Azure AD-integrated cloud applications, services, and systems.

The default **Anyone in the organization can invite guest users including guests and non-admins (most inclusive)** setting allows people to invite their personal accounts. When such a person leaves the organization, access may still be provided based on the personal account, and this may lead to a data leak.

When people outside your organization successfully redeem the invitation, they appear as **Guest** objects instead of **Member** objects.

There's more...

When the Azure AD tenant already contains guests, selecting the **Guest user access is restricted to properties and memberships of their own directory objects (most restrictive)** option in the **External collaboration settings** pane may be a smart decision.

See also

For more information on how to take control of Azure AD accounts that may have been created by your colleagues redeeming invitations, see the *Verifying your DNS domain name* recipe in *Chapter 14, Handling Authentication in a Hybrid World (AD FS, PHS, PTA, and DSSO).*

To allow guest invites but limit organizations to which invitations can be sent, see the *Allowing and blocking invitations for Azure AD B2B* recipe.

Allowing and blocking invitations for Azure AD B2B

This recipe shows how to allow or block **Domain Name System** (**DNS**) domain names for Azure AD B2B invitations.

Getting ready

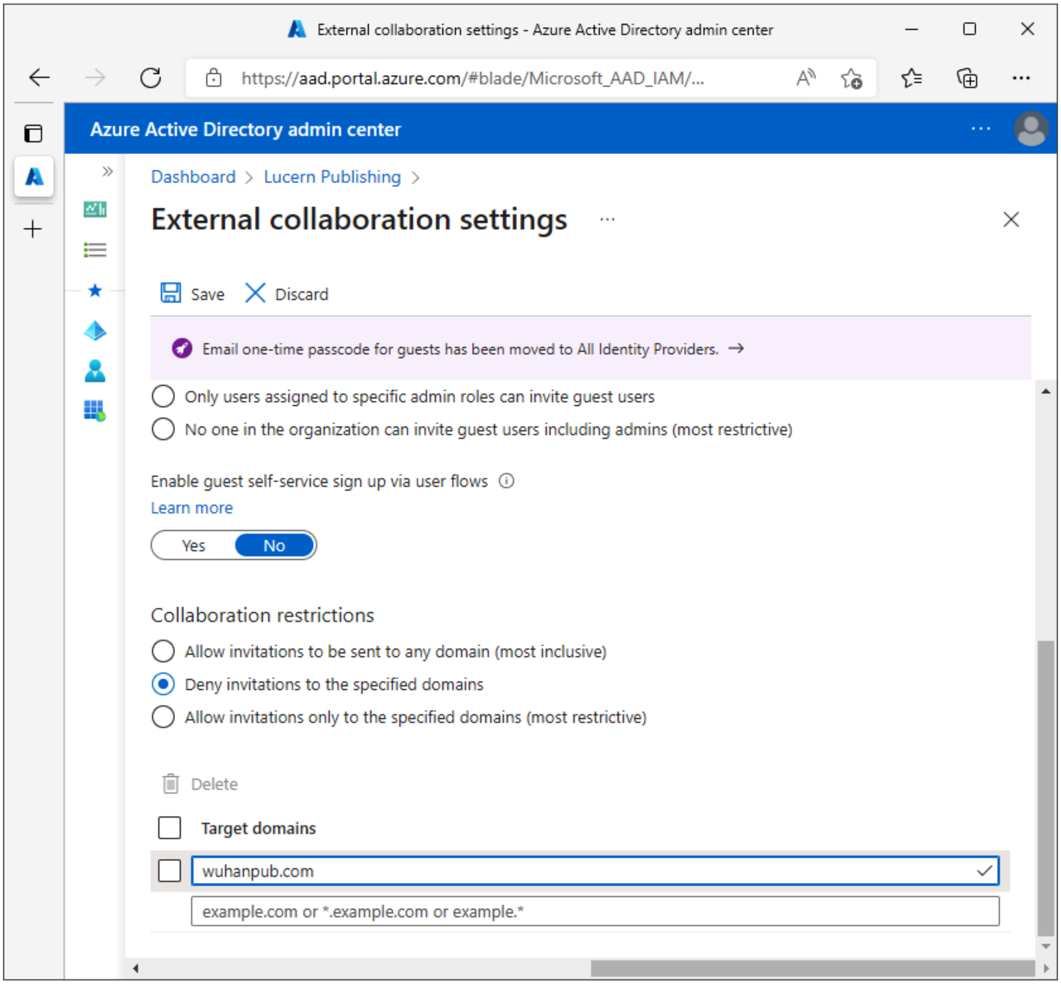
To complete this recipe, sign in to the Azure AD tenant with an account that has the **Global administrator** role assigned to it. If the organization uses the Azure AD PIM feature, activate the **Global administrator** role in advance.

How to do it...

Perform the following steps to allow or block Azure AD B2B invitations:

1. Navigate your browser to `https://aad.portal.azure.com`.
2. Sign in with an account in Azure AD that has the **Global administrator** role assigned.
3. Perform MFA when prompted.
4. In the left navigation pane, click **Azure Active Directory**.
5. In the **Azure Active Directory** navigation pane, click **User settings**.
6. Follow the **Manage external collaboration settings** link.
7. In the **External collaboration settings** pane, under **Collaboration restrictions**, select either the **Deny invitations to the specified domains** option or the **Allow invitations only to the specified domains (most restrictive)** option.

8. Type the DNS domain name(s) to block (with the **Deny invitations to the specified domains** option selected) or allow (with the **Allow invitations only to the specified domains (most restrictive)** option selected):


External collaboration settings - Azure Active Directory admin center
https://aad.portal.azure.com/#blade/Microsoft_AAD_IAM/...
Azure Active Directory admin center
Dashboard > Lucern Publishing >
Save
Discard
Delete

External collaboration settings

Email one-time passcode for guests has been moved to All Identity Providers. →

○ Only users assigned to specific admin roles can invite guest users
○ No one in the organization can invite guest users including admins (most restrictive)

Enable guest self-service sign up via user flows
Learn more
Yes [✓ No]

Collaboration restrictions
○ Allow invitations to be sent to any domain (most inclusive)
● Deny invitations to the specified domains
○ Allow invitations only to the specified domains (most restrictive)

□ Target domains
□ wuhanpub.com
example.com or *.example.com or example.*

Figure 16.5 – Collaboration restrictions on the External collaboration settings pane

9. Click **Save** at the top of the pane.

How it works...

Not collaborating might not be an option for every organization.

When an organization wants to collaborate with other organizations through Azure AD B2B, blocklisting can be used to ensure people in the organization cannot invite people from certain non-approved organizations. For this functionality, use the **Deny invitations to the specified domains** option. Invitations to email addresses with DNS domain names specified will not be sent and logged.

When an organization wants to collaborate with other organizations through Azure AD B2B, allowlisting can be used to ensure people in your organization can only invite people from approved organizations. For this functionality, use the **Allow invitations only to the specified domains (most restrictive)** option. Invitations to email addresses with other DNS domain names will not be sent or logged.

From a strategy perspective, allowing is preferred over blocking because it is the most restrictive option; however, it may lead to increased administrative overhead.

Configuring Azure AD join and Azure AD registration

This recipe shows how to limit the Azure AD join and Azure AD registration features for your organization, and allow the Enterprise State Roaming functionality.

Getting ready

To complete this recipe, sign in to the Azure AD tenant with an account that has the **Global administrator** role assigned to it. If the organization uses the Azure AD PIM feature, activate the **Global administrator** role in advance.

Configuring additional accounts with local administrator privileges on Azure AD-joined devices and enabling Enterprise State Roaming requires Azure AD Premium P1 licenses or Microsoft licenses that include the P1 license, such as Azure AD Premium P2, EMS E3, EMS A3, Microsoft 365 E3, or Microsoft 365 Business licenses.

How to do it...

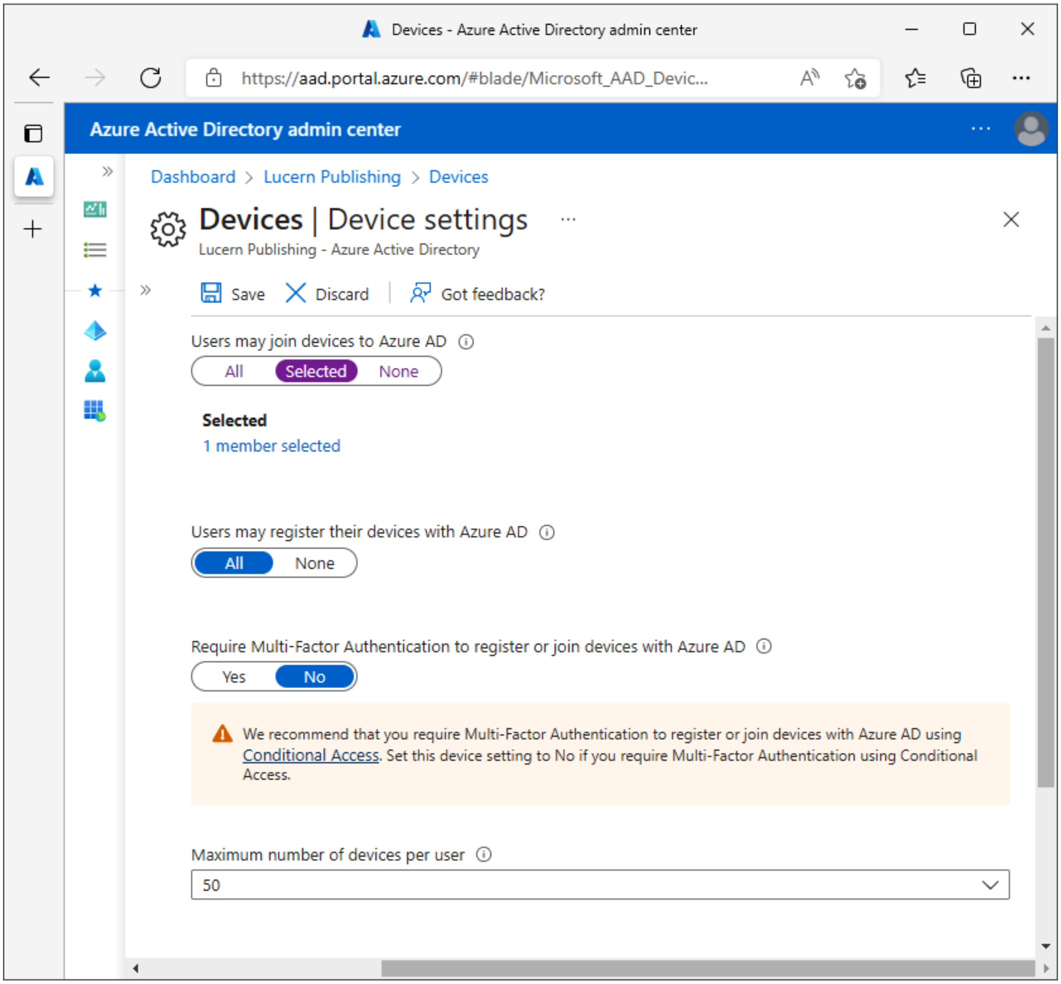
Configuring the Azure AD join and Azure AD registration features consists of these three distinct configuration changes:

- Limiting who can join Azure AD devices
- Limiting who can register Azure AD devices
- Enabling Enterprise State Roaming (Azure AD Premium only)

Limiting who can join Azure AD devices

Perform the following steps to limit the Azure AD join feature for your organization:

1. Navigate your browser to `https://aad.portal.azure.com`.
2. Sign in with an account in Azure AD that has the **Global administrator** role assigned.
3. Perform MFA when prompted.
4. In the left navigation pane, click **Azure Active Directory**.
5. In the **Azure Active Directory** navigation pane, click **Devices**.
6. In the **Devices** navigation pane, click **Device settings** to navigate to the **Devices | Device settings** pane.
7. Change the setting for **Users may join devices to Azure AD** from **All** to **Selected** or **None**.
8. When you've selected **Selected**, specify a group or multiple accounts that are allowed to join Azure AD devices by clicking the **No member selected** text. The **Members allowed to join devices** pane appears. Click **+ Add**. The **Add members** blade appears. Select or search for members to add. Click **Select** at the bottom of the blade. Click **OK** at the bottom of the **Members allowed to join devices** pane to return to the **Devices | Device settings** pane:


Devices - Azure Active Directory admin center
https://aad.portal.azure.com/#blade/Microsoft_AAD_Devic...
Azure Active Directory admin center
Dashboard > Lucern Publishing > Devices
Devices | Device settings
Lucern Publishing - Azure Active Directory
Save
Discard
Got feedback?

Users may join devices to Azure AD
All [✓ Selected] None

Selected
1 member selected

Users may register their devices with Azure AD
[✓ All] None

Require Multi-Factor Authentication to register or join devices with Azure AD
Yes [✓ No]

We recommend that you require Multi-Factor Authentication to register or join devices with Azure AD using Conditional Access. Set this device setting to No if you require Multi-Factor Authentication using Conditional Access.

Maximum number of devices per user
50

Figure 16.6 – Users may join devices to Azure AD option on the Devices | Device settings pane

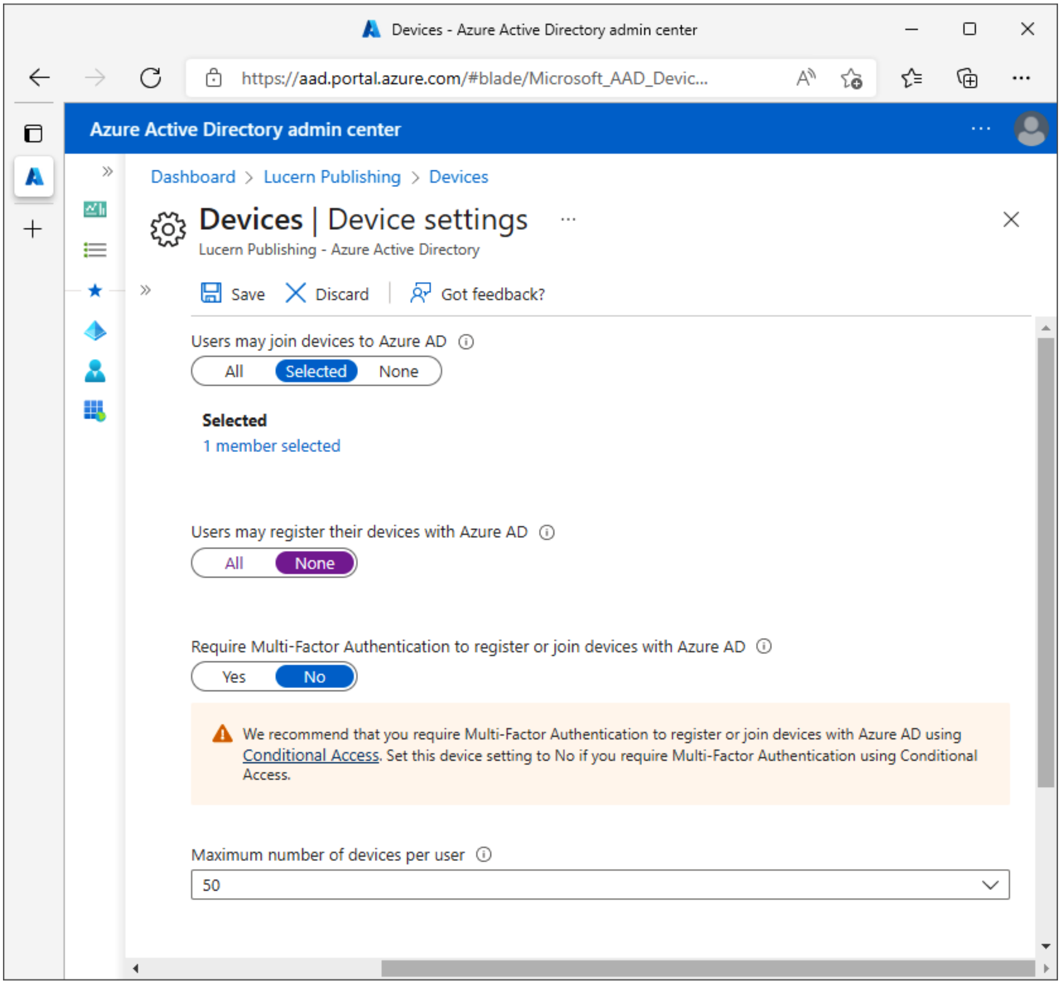
9. Click **Save** at the top of the **Devices | Device settings** pane.

Limiting who can register Azure AD devices

Perform the following steps to limit who can register Azure AD devices:

1. Navigate your browser to `https://aad.portal.azure.com`.
2. Sign in with an account in Azure AD that has the **Global administrator** role assigned.
3. Perform MFA when prompted.

4. In the left navigation pane, click **Azure Active Directory**.
5. In the **Azure Active Directory** navigation pane, click **Devices**.
6. In the **Devices** navigation pane, click **Device settings** to navigate to the **Devices | Device settings** pane.
7. Change the setting for **Users may register their devices with Azure AD** from **All** to **None**:


Devices - Azure Active Directory admin center
https://aad.portal.azure.com/#blade/Microsoft_AAD_Devic...
Azure Active Directory admin center
Dashboard > Lucern Publishing > Devices
Save
Discard
Got feedback?

Devices | Device settings

Lucern Publishing - Azure Active Directory

Users may join devices to Azure AD: All [✓ Selected] None

Selected
1 member selected

Users may register their devices with Azure AD: All [✓ None]

Require Multi-Factor Authentication to register or join devices with Azure AD: Yes [✓ No]

We recommend that you require Multi-Factor Authentication to register or join devices with Azure AD using Conditional Access. Set this device setting to No if you require Multi-Factor Authentication using Conditional Access.

Maximum number of devices per user: 50

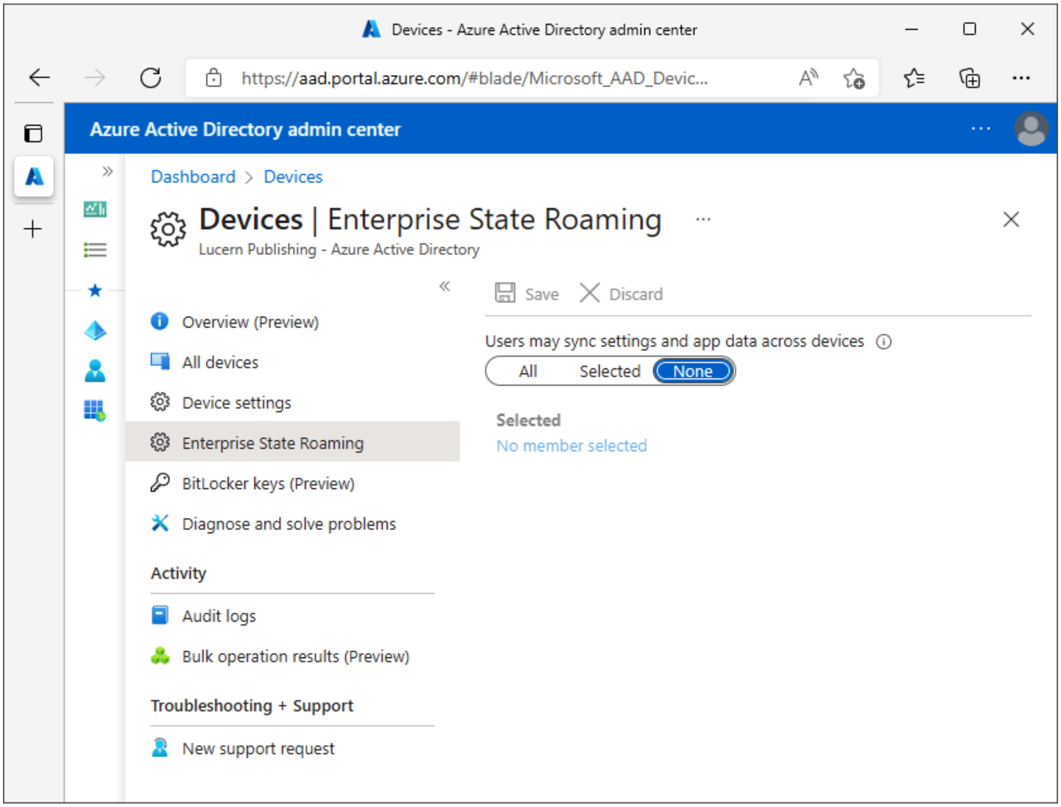
Figure 16.7 – Users may register their devices with Azure AD on the Devices | Device settings pane

8. Click **Save** at the top of the **Devices | Device Settings** pane.

Enabling Enterprise State Roaming

Perform the following steps to enable Enterprise State Roaming:

1. Navigate your browser to `https://aad.portal.azure.com`.
2. Sign in with an account in Azure AD that has the **Global administrator** role assigned.
3. Perform MFA when prompted.
4. In the left navigation pane, click **Azure Active Directory**.
5. In the **Azure Active Directory** navigation pane, click **Devices**.
6. In the **Devices** navigation pane, click **Enterprise State Roaming** to navigate to the **Devices | Enterprise State Roaming** pane:


Devices - Azure Active Directory admin center
https://aad.portal.azure.com/#blade/Microsoft_AAD_Devic...
Azure Active Directory admin center
Dashboard > Devices
Devices | Enterprise State Roaming
Lucern Publishing - Azure Active Directory
Save
Discard
Overview (Preview)
All devices
Device settings
Enterprise State Roaming
BitLocker keys (Preview)
Diagnose and solve problems
Activity
Audit logs
Bulk operation results (Preview)
Troubleshooting + Support
New support request

Users may sync settings and app data across devices: All | Selected | [✓ None]

Selected: No member selected

Figure 16.8 – Devices | Enterprise State Roaming pane

7. Change the setting for **Users may sync settings and app data across devices** from **None** to **All** or **Selected**.

8. When you've selected **Selected**, specify a group or multiple accounts that are allowed to use the Enterprise State Roaming functionality by clicking the **No member selected** text. The **Members allowed to sync settings and app data** pane appears. Click **+ Add**. The **Add members** blade appears. Select or search for members to add. Click **Select** at the bottom of the blade. Click **OK** at the bottom of the **Members allowed to sync settings and app data** pane to return to the **Devices | Enterprise State Roaming** pane.
9. Click **Save** at the top of the **Devices | Enterprise State Roaming** pane.

How it works...

Windows 10 and Windows 11 allow people to join their devices to Azure AD, marking it as an organizational device. End users who perform this action, by default, gain the same benefits as people using domain-joined devices, in terms of **single sign-on** (**SSO**) to Azure AD-integrated cloud applications and being able to sign in to other Azure AD-joined devices. With an Azure AD Premium license assigned, end users can also benefit from Enterprise State Roaming for their settings.

When a device is domain-joined, it cannot be Azure AD-joined by the end user.

Android and iOS-based devices don't offer the Azure AD join capability, but these devices can be registered with Azure AD to gain the same benefits as Azure AD-joined and domain-joined devices.

When you have either the **Enrollment with Microsoft Intune** or **Device Management for Office 365** settings enabled, the **Users may register their devices with Azure AD** option is grayed out. In such cases, devices use the Azure AD device registration service for compliance.

Enterprise State Roaming is a Windows feature that securely synchronizes user settings and application settings data to the Microsoft cloud. When people enabled with this feature switch from one Azure AD-joined device to another, they retain their settings across these devices. By using Group Policy and **mobile device management** (**MDM**) solutions, such as Microsoft Intune, administrators can manage settings in scope for synchronization.

See also

To hybrid Azure AD join devices to Azure AD, based on their domain-join status, see the *Configuring hybrid Azure AD join* recipe in *Chapter 15*, *Handling Synchronization in a Hybrid World (Azure AD Connect)*.

Configuring Intune auto-enrollment upon Azure AD join

This recipe shows how to configure auto-enrollment in Microsoft Intune for MDM and **mobile application management** (**MAM**) upon Azure AD join.

Getting ready

To complete this recipe, sign in to the Azure AD tenant with an account that has the **Global administrator** role assigned to it. If the organization uses the Azure AD PIM feature, activate the **Global administrator** role in advance.

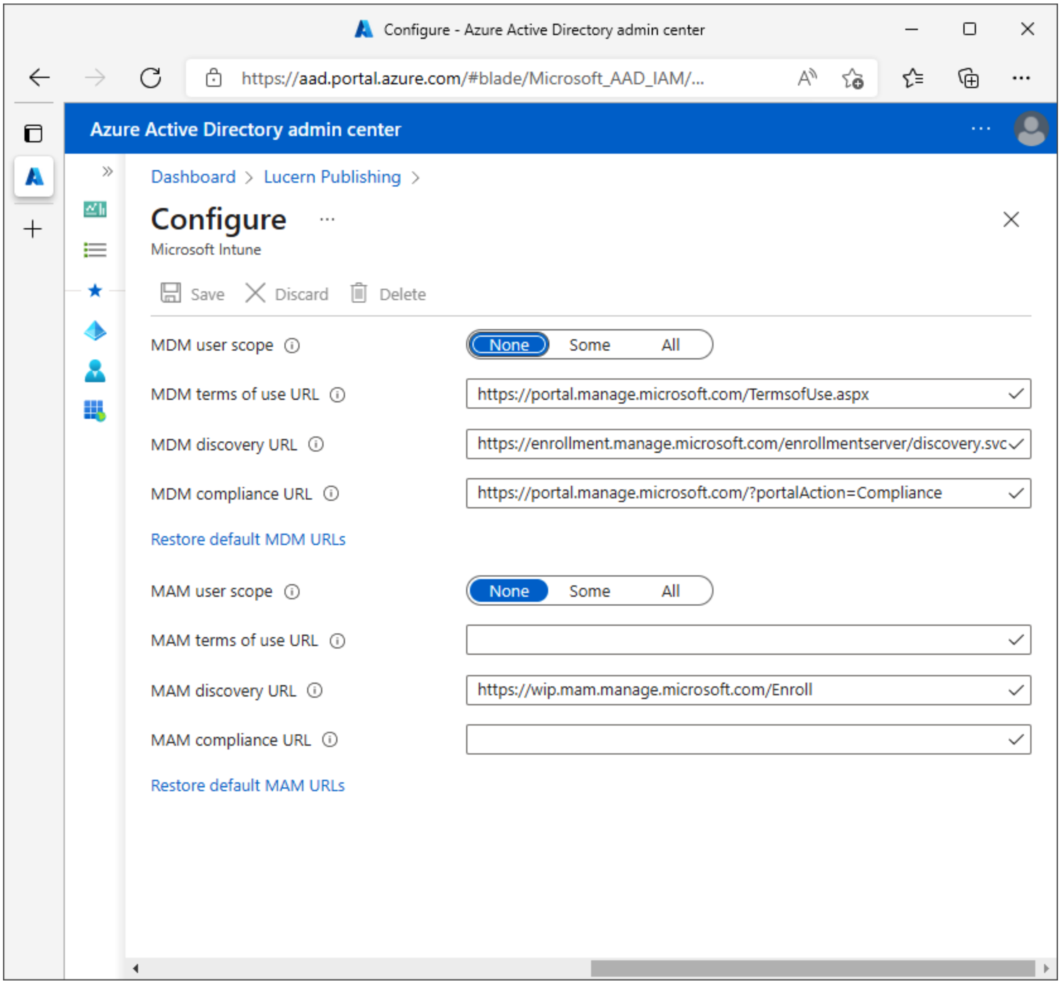
An MDM solution, such as Microsoft Intune, needs to be configured for the Azure AD tenant. This recipe shows how to configure auto-enrollment for Intune, but when the URLs for your organization's alternative MDM solution are known, the default URLs can be replaced to meet your organization's needs.

How to do it...

Perform these steps to configure Intune auto-enrollment upon Azure AD join:

1. Navigate your browser to `https://aad.portal.azure.com`.
2. Sign in with an account in Azure AD that has the **Global administrator** role assigned.
3. Perform MFA when prompted.
4. In the left navigation pane, click **Azure Active Directory**.
5. In the **Azure Active Directory** navigation pane, click **Mobility (MDM and MAM)** to navigate to the **Mobility (MDM and MAM)** pane.

6. In the **Mobility (MDM and MAM)** pane, click **Microsoft Intune** to navigate to the **Configure** pane:



Configure - Azure Active Directory admin center

https://aad.portal.azure.com/#blade/Microsoft_AAD_IAM/...

Azure Active Directory admin center

Dashboard > Lucern Publishing >

Configure ...

Microsoft Intune

Save Discard Delete

MDM user scope ⓘ [✓ None] Some All

MDM terms of use URL ⓘ https://portal.manage.microsoft.com/TermsofUse.aspx

MDM discovery URL ⓘ https://enrollment.manage.microsoft.com/enrollmentserver/discovery.svc

MDM compliance URL ⓘ https://portal.manage.microsoft.com/?portalAction=Compliance

Restore default MDM URLs

MAM user scope ⓘ [✓ None] Some All

MAM terms of use URL ⓘ

MAM discovery URL ⓘ https://wip.mam.manage.microsoft.com/Enroll

MAM compliance URL ⓘ

Restore default MAM URLs

Figure 16.9 – Configure pane for Microsoft Intune

7. In the **Configure** pane, change the **MDM user scope** setting to **All**.
8. Change the **MAM user scope** setting to **All**.
9. Click **Save** at the top of the **Configure** pane.

How it works…

MDM and MAM are modern endpoint-management solutions that manage devices regardless of their operating system or location. MDM manages settings for complete devices, whereas MAM manages settings for applications on these devices, such as Outlook. A device can be MDM-managed and MAM-managed but does not need to be both to work.

When Intune auto-enrollment is configured, Azure AD-joined devices, hybrid Azure AD-joined devices, and Azure AD-registered devices are in scope.

Choosing between Security defaults and Conditional Access

Azure AD offers the **Security defaults** feature to offer default security settings for any Azure AD tenant. For administrators of tenants with Azure AD Premium licenses, using **Conditional Access** offers more flexibility in some respects. This recipe shows how to decide between using Security defaults and Conditional Access.

Getting ready

To make a choice on which to use, you'll need to know the following characteristics of your organization:

- Has your organization assigned Azure AD Premium P1 licenses to all people in scope for security measures?
- Has your organization assigned Azure AD Premium P2 licenses to all people in scope for security measures?
- Does your organization use any applications, services, and/or systems that rely on user accounts in Azure AD instead of security principals?

How to do it...

Use the following flowchart to make the right choice between Security defaults and Conditional Access:

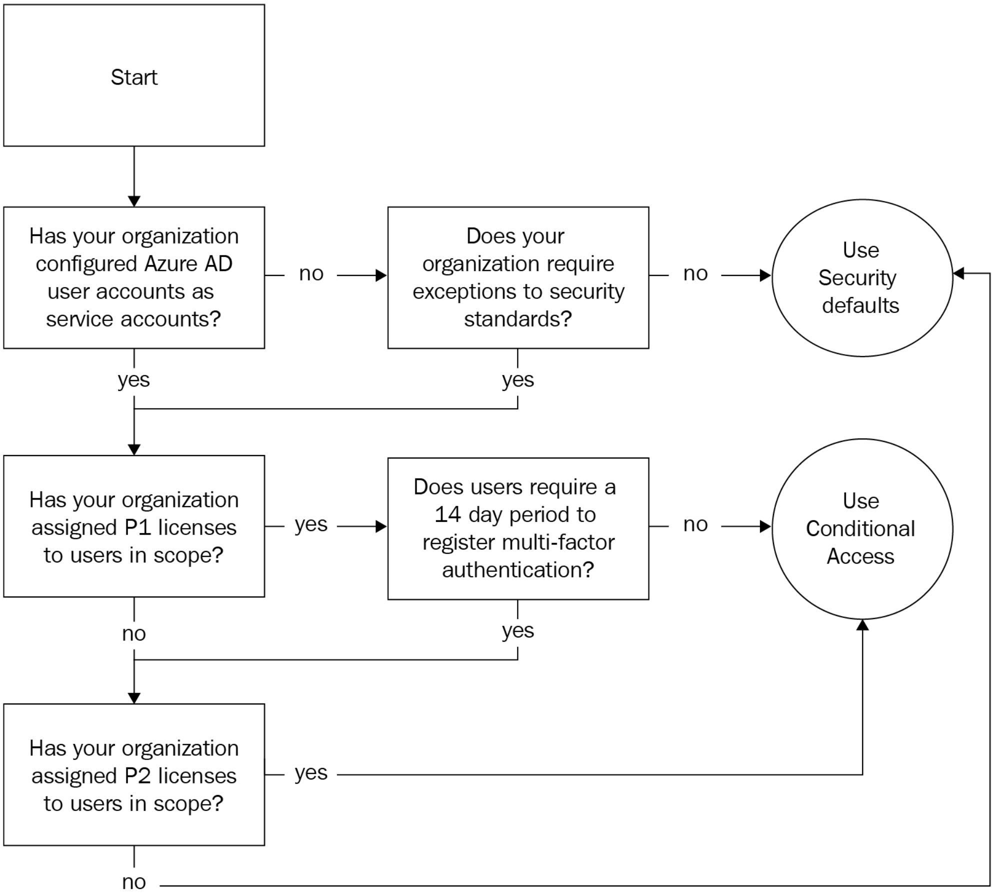

Start
Has your organization configured Azure AD user accounts as service accounts?
no
Does your organization require exceptions to security standards?
no
Use Security defaults
yes
yes
Has your organization assigned P1 licenses to users in scope?
yes
Does users require a 14 day period to register multi-factor authentication?
no
Use Conditional Access
no
yes
Has your organization assigned P2 licenses to users in scope?
yes
no

Figure 16.10 – Flowchart to decide between security defaults and Conditional Access

How it works...

Azure AD Security defaults are enabled by default for new Azure AD tenants. They offer the following security settings:

- All users in Azure AD are required to register for MFA. They have a 14-day period to register their security information, starting at the first interactive sign-in to Azure AD and Azure AD-integrated apps, services, and systems.

- Users with Azure AD privileged roles assigned are required to perform MFA at every sign-in.
- Legacy authentication protocols are blocked.
- Users that are flagged by the Azure AD Identity Protection functionality as risky users are required to perform MFA.
- Access to the Azure portal is restricted.

As an alternative to the Security defaults, organizations with Azure AD Premium P1 and/or Azure AD Premium P2 licenses can adopt Conditional Access policies. These policies are more flexible because they allow exclusions and conditions in the aforementioned security defaults. Conditional Access policies need to be configured from scratch.

Typical exclusions in Conditional Access policies apply to service accounts, emergency access accounts, and multi-functional printers.

When a person's user account comes into the scope of a Conditional Access policy that contains a condition for them to perform MFA, initial MFA registration is triggered. In this case, the person needs to register for MFA to specify security information to perform MFA during the next interactive sign-in.

However, when MFA is required as part of the Security defaults or as part of an MFA registration policy within Azure AD Identity Protection, users do not need to register during the next interactive sign-in but may postpone registering for up to 14 days.

The Conditional Access functionality requires Azure AD Premium P1 licenses or Microsoft licenses that include the P1 license, such as Azure AD Premium P2, EMS E3, EMS A3, Microsoft 365 E3, or Microsoft 365 Business licenses.

Managing the Identity Protection functionality requires Azure AD Premium P2 licenses or Microsoft licenses that include the P2 license, such as EMS E5 or Microsoft 365 E5 licenses.

Security defaults and Conditional Access policies are mutually exclusive.

Configuring Conditional Access

This recipe shows how to switch from Security defaults to Conditional Access and configure Conditional Access policies. As three example policies, we will perform the following configurations:

- All users can access an Azure AD-integrated application only when they perform MFA.
- All users can access any Azure AD-integrated applications only when they use a hybrid Azure AD-joined device when they are visiting sensitive countries on business trips.
- No users can use legacy authentication.

Getting ready

To complete this recipe, sign in to the Azure AD tenant with an account that has the **Global administrator** or **Conditional Access administrator** role assigned to it. If the organization uses the Azure AD PIM feature, activate the **Global administrator** or **Conditional Access administrator** role in advance.

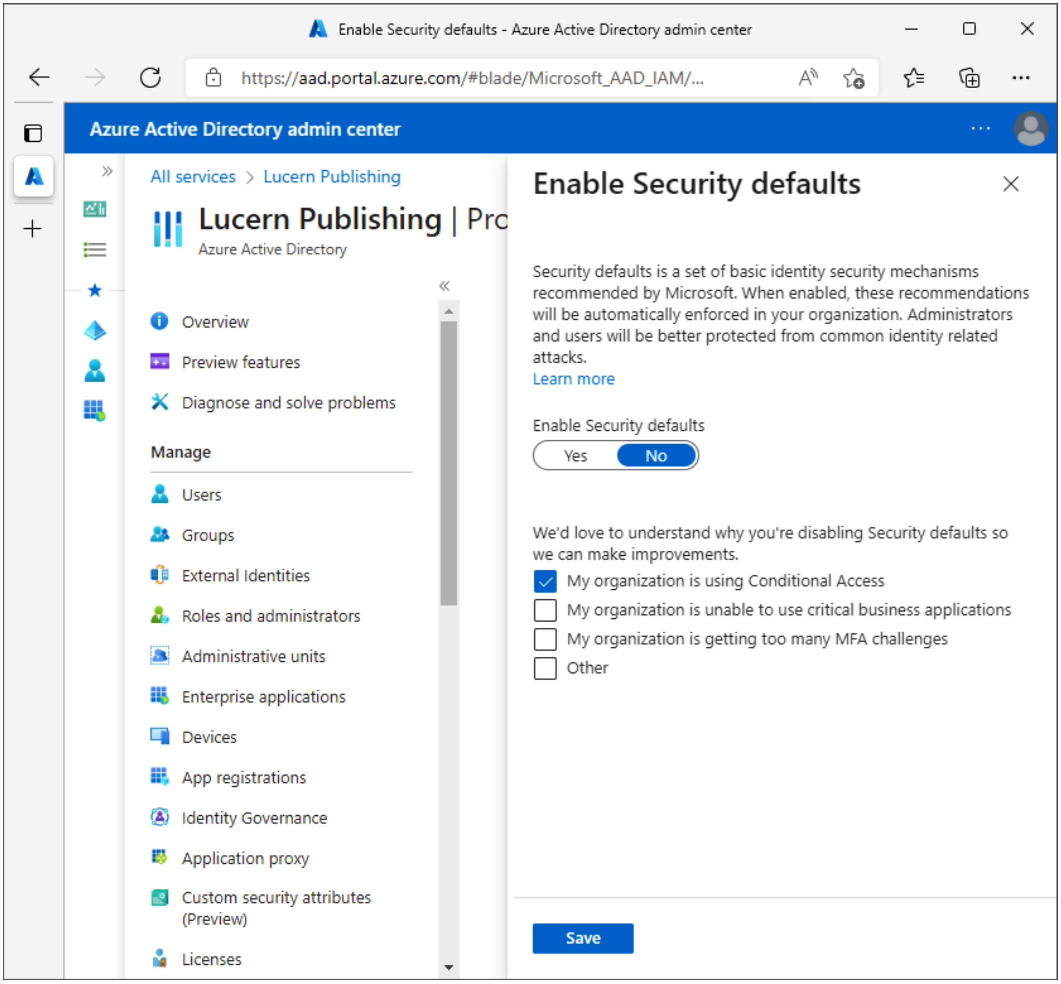
The Conditional Access functionality requires Azure AD Premium P1 licenses or Microsoft licenses that include the P1 license, such as Azure AD Premium P2, EMS E3, EMS A3, Microsoft 365 E3, or Microsoft 365 Business licenses.

How to do it...

Perform the following steps to configure Conditional Access to allow all users to access an Azure AD-integrated application only when they perform MFA:

1. Navigate your browser to `https://aad.portal.azure.com`.
2. Sign in with an account in Azure AD that has the **Global administrator** role assigned.
3. Perform MFA when prompted.
4. In the left navigation pane, click **Azure Active Directory**.
5. In the **Azure Active Directory** navigation pane, click **Properties**.
6. At the bottom of the **Properties** pane, follow the **Manage Security defaults** link. The **Enable Security defaults** blade appears.

7. Change the **Enable Security defaults** setting to **No**. As the reason for disabling Security defaults, select the **My organization is using Conditional Access** option:


Enable Security defaults - Azure Active Directory admin center
https://aad.portal.azure.com/#blade/Microsoft_AAD_IAM/...
Azure Active Directory admin center
All services > Lucern Publishing
Lucern Publishing | Pro
Azure Active Directory
Overview
Preview features
Diagnose and solve problems
Manage
Users
Groups
External Identities
Roles and administrators
Administrative units
Enterprise applications
Devices
App registrations
Identity Governance
Application proxy
Custom security attributes (Preview)
Licenses

Enable Security defaults

Security defaults is a set of basic identity security mechanisms recommended by Microsoft. When enabled, these recommendations will be automatically enforced in your organization. Administrators and users will be better protected from common identity related attacks.
Learn more

Enable Security defaults

Yes [✓ No]

We'd love to understand why you're disabling Security defaults so we can make improvements.

☑ My organization is using Conditional Access

☐ My organization is unable to use critical business applications

☐ My organization is getting too many MFA challenges

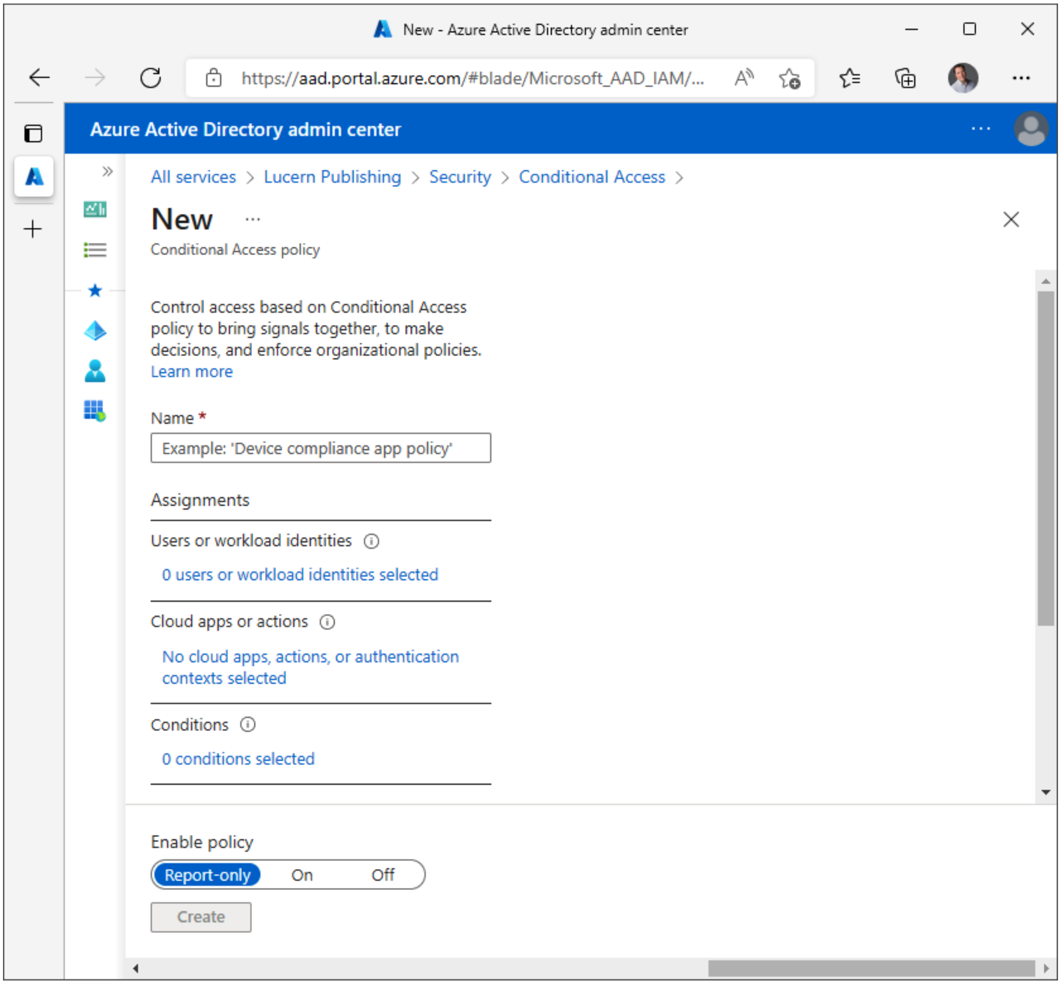
☐ Other

Save

Figure 16.11 – Disabling the Security defaults on the Enable Security defaults blade

8. Click **Save** at the bottom of the **Enable Security defaults** blade.
9. In the **Azure Active Directory** navigation pane, click **Security**.
10. In the **Security** navigation pane, click **Conditional Access** to go to the **Conditional Access | Policies** pane.

11. In the **Conditional Access | Policies** pane, click **+ New policy**. Select **Create new policy** from the context menu. The **New** pane appears:


New - Azure Active Directory admin center
https://aad.portal.azure.com/#blade/Microsoft_AAD_IAM/...
Azure Active Directory admin center
All services > Lucern Publishing > Security > Conditional Access >

New ...

Conditional Access policy

Control access based on Conditional Access policy to bring signals together, to make decisions, and enforce organizational policies.
Learn more

Name *
Example: 'Device compliance app policy'

Assignments

Users or workload identities
0 users or workload identities selected

Cloud apps or actions
No cloud apps, actions, or authentication contexts selected

Conditions
0 conditions selected

Enable policy
Report-only | On | Off

Create

Figure 16.12 – New pane

12. To create the policy that requires MFA, make the following changes:

I. Provide a **Name** value for the new Conditional Access policy.

II. In the **Assignments** region, click **0 users or workload identities selected**. Then, select the **All users** option.

- III. In the **Assignments** region, click **No cloud apps, actions, or authentication contexts selected**. Then, select the **Select Apps** option. A **Select** blade will appear. Select an Azure AD-integrated application that you want to require MFA for from the list or search for it. Click **Select** at the bottom of the blade.
- IV. In the **Access Controls** region, click **0 controls selected** under **Grant**. The **Grant** blade appears. Select **Require multi-factor authentication** as the control. Click **Select** at the bottom of the **Grant** pane.
- V. Select **On** for the **Enable policy** setting.

13. Click **Create** at the bottom of the **New** pane to create a Conditional Access policy and return to the **Conditional Access | Policies** pane.

Perform these steps to configure Conditional Access to allow all users to access any Azure AD-integrated applications only when they use a hybrid Azure AD-joined device when they are visiting sensitive countries:

1. Navigate your browser to `https://aad.portal.azure.com`.
2. Sign in with an account in Azure AD that has the **Global administrator** role or the **Conditional Access administrator** role assigned.
3. Perform MFA when prompted.
4. In the left navigation pane, click **Azure Active Directory**.
5. In the **Azure Active Directory** navigation pane, click **Security**.
6. In the **Security** navigation pane, click **Conditional Access** to go to the **Conditional Access | Policies** pane.
7. In the **Conditional Access** navigation pane, click **Named locations** to navigate to the **Conditional Access | Named Locations** pane.
8. In the **Conditional Access | Named locations** pane, click **+ Countries location**. The **New location (Countries)** blade appears.
9. Specify a name for the named location in the **Name** field.
10. Select the **Include unknown countries/areas** option.
11. Select countries that are not deemed trustworthy by your organization.
12. Click **Create** at the bottom of the **New location (Countries)** blade.
13. Click **Policies** at the top of the **Conditional Access** navigation pane.

14. In the **Conditional Access | Policies** pane, click **+ New policy**. Select **Create new policy** from the context menu. The **New** pane appears.

    I. Provide a **Name** value for the new Conditional Access policy.

    II. In the **Assignments** region, click **0 users or workload identities selected**. Then, select the **All users** option.

    III. In the **Assignments** region, click **No cloud apps, actions, or authentication contexts selected**. Then, select the **All Apps** option.

    IV. In the **Conditions** region, click **0 conditions selected**. Under **Locations**, click **Not configured**. Change the **Configure** setting to **Yes**. Click **Exclude**. The **Selected locations** option is selected by default. Click **None**. The **Select** blade appears. Select a named location with untrustworthy countries. Click **Select** at the bottom of the **Select** blade. Click **Policies** to resume configuring the Conditional Access policy.

    V. In the **Access Controls** region, click **0 controls selected** under **Grant**. The **Grant** blade appears. Select **Require Hybrid Azure AD joined device** as the control. Click **Select** at the bottom of the **Grant** pane.

    VI. Select **On** for the **Enable policy** setting.

15. Click **Create** at the bottom of the **New** pane to create a Conditional Access policy and return to the **Conditional Access | Policies** pane.

Perform the following steps to block legacy authentication:

1. Navigate your browser to `https://aad.portal.azure.com`.
2. Sign in with an account in Azure AD that has the **Global administrator** role or the **Conditional Access administrator** role assigned.
3. Perform MFA when prompted.
4. In the left navigation pane, click **Azure Active Directory**.
5. In the **Azure Active Directory** navigation pane, click **Security**.
6. In the **Security** navigation pane, click **Conditional Access** to go to the **Conditional Access | Policies** pane.
7. In the **Conditional Access | Policies** pane, click **+ New policy**. Select **Create new policy** from the context menu. The **New** pane appears.

    I. Provide a **Name** value for the new Conditional Access policy.

    II. In the **Assignments** region, click **0 users or workload identities selected**. Then, select the **All users** option.

III. In the **Assignments** region, click **No cloud apps, actions, or authentication contexts selected**. Then, select the **All Apps** option.

IV. In the **Conditions** region, click **0 conditions selected**. Under **Client apps**, click **Not configured**. The **Client apps** blade appears. Change the **Configure** setting to **Yes**. Deselect the options under **Modern authentication clients**. Click **Done** at the bottom of the **Client apps** blade.

V. In the **Access Controls** region, click **0 controls selected** under **Grant**. The **Grant** blade appears. Select **Block access** as the control. Click **Select** at the bottom of the **Grant** pane.

VI. Select **On** for the **Enable policy** setting.

8. Click **Create** at the bottom of the **New** pane to create a Conditional Access policy and return to the **Conditional Access | Policies** pane.

How it works...

In on-premises networks, access is typically governed by group memberships; when you authenticate on a domain-joined system, the group account is a member that dictates the level of access. Dynamic Access Control and authentication policies, introduced in Windows Server 2012 and Windows Server 2012 R2, showed the possibilities of **attribute-based access control** (**ABAC**) with claims in Kerberos.

For cloud applications, services, and systems, an even more granular form of access control is needed. Microsoft introduced controls organizations need with Conditional Access. Access can be allowed or denied, per Azure AD account and/or group, and per Azure AD-integrated applications, including on-premises claims-based applications and applications that are published through Azure AD Application Proxy.

Access can be allowed or denied based on conditions, such as sign-in risk, the device used, and its state (marked as compliant or hybrid Azure AD-joined), the location from where the authentication originates, and/or the client app used and its state (approved or non-approved).

Access can be subject to performing MFA. Additionally, access to SharePoint can be limited to read-only or non-download through authentication contexts. The possibilities to control access based on these conditions are virtually endless.

To make the most of Conditional Access policies, it's good to know that, without Security defaults being applied, everyone has access to everything, at any time, anywhere. Therefore, every time an administrator tries to add a policy that enforces this same principle, the **Save** button at the bottom of the Conditional Access policy pane is grayed out.

The **All Users** assignment in Conditional Access contains all users, including guest users.

To avoid locking out administrators, Conditional Access features a **What If** feature. Before creating a policy, the **What If** feature allows you to input conditions and run a preliminary analysis of the impact. Additionally, Conditional Access policies can be configured as **Report-only**. Assigning this state to a Conditional Access policy allows administrators to view the impact of a policy through Azure AD workbooks. Using this process, policies can be applied without inadvertently denying access.

In terms of naming, it's wise to start with a consistent naming convention for Conditional Access policies. As the organization embraces the possibilities, many policies may be created. A naming convention with the environment (`D`, `T`, `A`, or `P`), app name(s), scope, and conditions is recommended, resulting in the following Conditional Access policy names:

- P – All Users – HR application – Require MFA
- P – All Users – All applications – Compliant devices only in untrustworthy countries
- P – All Users – All applications – Block Legacy Authentication

See also

To use device state with Conditional Access, refer to the *Configuring Azure AD join and Azure AD registration* recipe in this chapter and the *Configuring hybrid Azure AD join* recipe in *Chapter 15*, *Handling Synchronization in a Hybrid World (Azure AD Connect).*

Accessing Azure AD Connect Health

This recipe shows the benefits of using Azure AD Connect Health to monitor and troubleshoot a hybrid identity implementation.

Getting ready

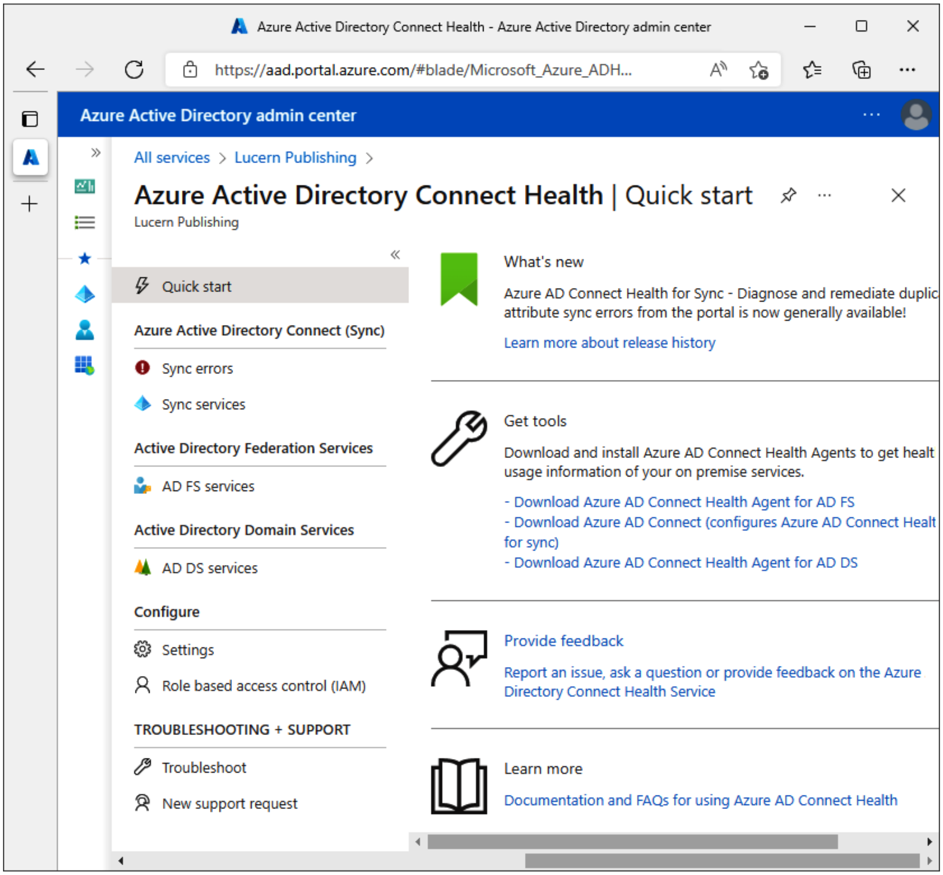
To complete this recipe, sign in to Azure AD with an account that has the **Global administrator** role assigned to it. If the organization uses the Azure AD PIM feature, activate the **Global administrator** role in advance. Access to the **Azure AD Connect Health** dashboard can be delegated through its **role-based access control** (**RBAC**) IAM settings.

The Azure AD Connect Health functionality requires Azure AD Premium P1 licenses or Microsoft licenses that include the P1 license, such as Azure AD Premium P2, EMS E3, EMS A3, Microsoft 365 E3, or Microsoft 365 Business licenses.

How to do it...

Perform these steps:

1. Navigate your browser to `https://aad.portal.azure.com`.
2. Sign in with an account in Azure AD that has the **Global administrator** role assigned.
3. Perform MFA when prompted.
4. In the left navigation pane, click **Azure Active Directory**.
5. In the **Azure Active Directory** navigation pane, click **Azure AD Connect** to navigate to the **Azure AD Connect** pane.
6. In the **Azure AD Connect** main pane, under **Health and analytics**, follow the **Azure AD Connect Health** link to go to the **Azure Active Directory Connect Health | Quick start** pane:



Azure Active Directory Connect Health - Azure Active Directory admin center

https://aad.portal.azure.com/#blade/Microsoft_Azure_ADH...

Azure Active Directory admin center

All services > Lucern Publishing >

Azure Active Directory Connect Health | Quick start

Lucern Publishing

What's new

Azure AD Connect Health for Sync - Diagnose and remediate duplic attribute sync errors from the portal is now generally available!

Learn more about release history

Get tools

Download and install Azure AD Connect Health Agents to get healt usage information of your on premise services.

- Download Azure AD Connect Health Agent for AD FS
- Download Azure AD Connect (configures Azure AD Connect Healt for sync)
- Download Azure AD Connect Health Agent for AD DS

Provide feedback

Report an issue, ask a question or provide feedback on the Azure Directory Connect Health Service

Learn more

Documentation and FAQs for using Azure AD Connect Health

Quick start
Azure Active Directory Connect (Sync)
Sync errors
Sync services
Active Directory Federation Services
AD FS services
Active Directory Domain Services
AD DS services
Configure
Settings
Role based access control (IAM)
TROUBLESHOOTING + SUPPORT
Troubleshoot
New support request

Figure 16.13 – Azure Active Directory Connect Health | Quick start pane

7. In the **Azure Active Directory Connect Health** navigation pane, click **Sync errors** to examine synchronization errors.
8. In the **Azure Active Directory Connect Health** navigation pane, click **Sync services** to examine Azure AD Connect installations associated with the Azure AD tenant.
9. In the **Azure Active Directory Connect Health** navigation pane, click **Settings** to optionally disable automatic upgrades of Azure AD Connect Health agents with the latest version, and/or allow Microsoft access to the tenant's health data for troubleshooting purposes.

How it works...

By default, the Azure AD Connect Health agent is installed with Azure AD Connect, and, by default, it provides data to Azure AD on the synchronization status.

Data is available in the Azure AD Connect Health dashboard in the Azure portal but is only accessible when at least one Azure AD Premium P1 license is attached to the Azure AD tenant per configured Azure AD Connect installation.

The Azure AD Connect Health dashboard provides information on the following:

- Azure AD Connect synchronization services:
    - Azure AD Connect servers in use:
        - Their status
        - Their alerts
        - Their Azure AD Connect versions, database settings, and service accounts
        - Their operating systems, domains, time zones, last reboots, machine types, and dimensions in terms of **central processing unit** (**CPU**) and physical memory
        - The Azure AD Connect Health Agent version
        - The time of their last export to Azure AD
        - Their run-profile latency in seconds
- Settings for uploaded data; whether to upload all error logs

- Azure AD Connect synchronization errors:
    - Viewing of the following synchronization errors:
        - Duplicate attribute errors
        - Data mismatch errors
        - Data validation failures
        - Large attribute errors
        - Federated domain changes
        - Existing administrator role conflicts
        - Exporting the preceding errors
    - Notification settings for errors

Microsoft's strategy for Azure AD Connect Health is that administrators can troubleshoot hybrid identity upon being notified of problems in their environment.

There's more...

By default, people with accounts in the **Global administrator** role receive notifications for Azure AD Connect Health. The list of email addresses can be expanded to include email addresses for people and/or distribution lists. Perform the following steps to configure email addresses for Azure AD Connect Health while signed in and while in the **Azure Active Directory Connect Health | Quick start** pane:

1. In the **Azure Active Directory Connect Health** navigation pane, click **Sync errors**.
2. In the top ribbon, click **Notification Settings** to navigate to the **Notification** pane.
3. In the field under **Additional email recipients**, enter the email address of a mailbox or distribution list within your organization, outsourcing organization, or partner organization. New fields appear automatically to accommodate your needs.
4. Deselect the **Notify All Global Administrators** option if it is no longer needed.
5. Click **Save** at the top of the **Notification** pane.

Configuring Azure AD Connect Health for AD FS

Azure AD Connect Health can be expanded to include monitoring of AD FS servers and **Web Application Proxy** (**WAP**) servers of your organization's AD FS implementation. This recipe shows how to do this.

Getting ready

To complete this recipe, sign in to Azure AD with an account that has the **Global administrator** role assigned to it. If the organization uses the Azure AD PIM feature, activate the **Global administrator** role in advance. Access to the Azure AD Connect Health dashboard can be delegated through its RBAC IAM settings.

The Azure AD Connect Health functionality requires Azure AD Premium P1 licenses or Microsoft licenses that include the P1 license, such as Azure AD Premium P2, EMS E3, EMS A3, Microsoft 365 E3, or Microsoft 365 Business licenses.

Ensure all AD FS servers and WAP servers run Windows PowerShell `4.0` or above and have **Internet Explorer Enhanced Security Configuration** (**IE ESC**) turned off.

How to do it...

Configuring Azure AD Connect Health for AD FS consists of the following three steps:

- Downloading the agent
- Installing and configuring the agent
- Consuming information in the Azure AD Connect Health dashboard

Downloading the agent

Perform the following steps to download the Azure AD Connect Health Agent for AD FS:

1. Navigate your browser to `https://aad.portal.azure.com`.
2. Sign in with an account in Azure AD that has the **Global administrator** role assigned.
3. Perform MFA when prompted.
4. In the left navigation pane, click **Azure Active Directory**.
5. In the **Azure Active Directory** navigation pane, click **Azure AD Connect** to navigate to the **Azure AD Connect** pane.

6. In the **Azure AD Connect** main pane, under **Health and analytics**, follow the **Azure AD Connect Health** link to go to the **Azure Active Directory Connect Health | Quick start** pane.
7. In the **Azure Active Directory Connect Health | Quick start** pane, click the **Download Azure AD Connect Health Agent for AD FS** link.
8. Download the agent to a location where it is accessible to AD FS servers and WAP servers, or download it and copy it to the hard disks of these servers.

Installing and configuring the agent

Perform these steps to install and configure the Azure AD Connect Health Agent for AD FS on each AD FS server and each WAP server in your environment:

1. Run `AdHealthAdfsAgentSetup.exe`. The **Azure AD Connect Health AD FS Agent** window appears.
2. In the **Azure AD Connect Health AD FS Agent** window, click **Install**.
3. On the **Setup Successful** screen, click **Configure Now**. An elevated Windows PowerShell window appears that executes the `Register-AzureADConnectHealthADFSAgent` cmdlet. Then, the **Sign in to your account** window appears.
4. Sign in with an account in Azure AD that has the **Global administrator** role assigned.
5. Perform MFA when prompted.
6. Close the Windows PowerShell window.

Repeat the preceding steps on all AD FS servers and all WAP servers.

Consuming information in the Azure AD Connect Health dashboard

Perform the following steps to consume information from Azure AD Connect Health Agent for AD FS installations in the Azure AD Connect Health dashboard:

1. Navigate your browser to `https://aad.portal.azure.com`.
2. Sign in with an account in Azure AD that has the **Global administrator** role assigned.
3. Perform MFA when prompted.
4. In the left navigation pane, click **Azure Active Directory**.
5. In the **Azure Active Directory** navigation pane, click **Azure AD Connect** to navigate to the **Azure AD Connect** pane.

6. In the **Azure AD Connect** main pane, under **Health and analytics**, follow the **Azure AD Connect Health** link to go to the **Azure Active Directory Connect Health | Quick start** pane.
7. In the **Azure Active Directory Connect Health** navigation pane, click **AD FS services**.
8. In the **AD FS services** pane, click the service name of the AD FS farm you want to monitor to navigate to its dashboard:

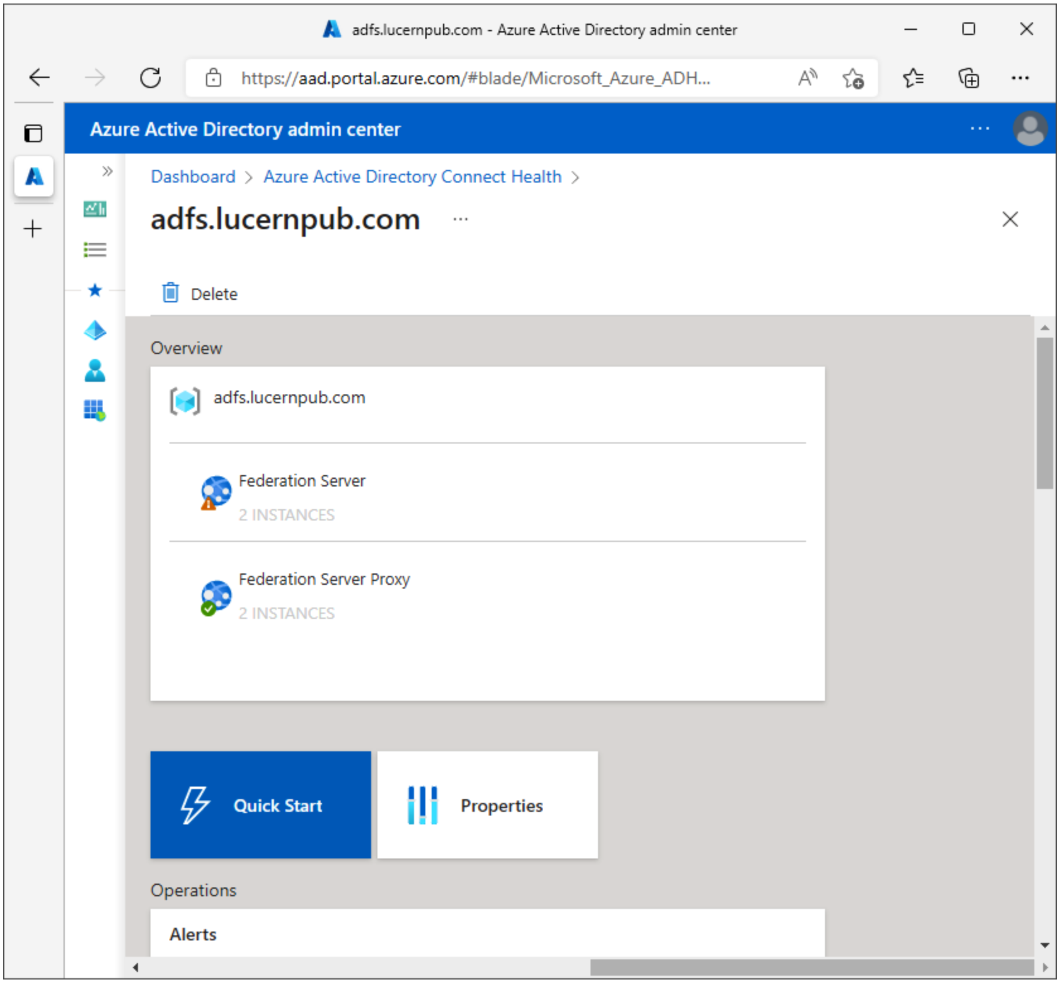

adfs.lucernpub.com - Azure Active Directory admin center
https://aad.portal.azure.com/#blade/Microsoft_Azure_ADH...
Azure Active Directory admin center
Dashboard > Azure Active Directory Connect Health >
adfs.lucernpub.com
Delete
Overview
adfs.lucernpub.com
Federation Server
2 INSTANCES
Federation Server Proxy
2 INSTANCES
Quick Start
Properties
Operations
Alerts

Figure 16.14 – Azure AD Connect Health dashboard for an AD FS implementation

How it works...

The Azure AD Connect Health Agent for AD FS is an additional agent that can be installed on AD FS servers and WAP servers. Microsoft recommends installing the agent on all AD FS servers and WAP servers of an AD FS implementation for complete insight.

Azure AD Connect Health's data is not sent by the WAP servers on behalf of the AD FS servers as outside authentication requests are. Each server sends its health data independently.

Data is available in the Azure AD Connect Health dashboard in the Azure portal but is only accessible when at least one Azure AD Premium P1 license is attached to the Azure AD tenant per each configured Azure AD Connect installation. For correct licensing, every monitored AD FS server and every monitored WAP server require an additional 25 Azure AD Premium P1 licenses attached to the Azure AD tenant.

The Azure AD Connect Health dashboard provides information on the following:

- AD FS servers:
    - Their status
    - Their alerts
    - Their operating systems, domains, time zones, last reboots, machine types, and dimensions in terms of CPU and physical memory
    - The Azure AD Connect Health Agent version
    - Statistical information for the last week, day, or past 6 hours in graphs on the following aspects:
        - Token requests per second
        - AD FS private bytes Extranet account lockouts established **Transmission Control Protocol** (**TCP**) connections
        - Credential authentication failures
        - Credential authentication failures per second used memory (percentage)
        - User processor (percentage)

- WAP servers:
    - Their status
    - Their alerts
    - Their operating systems, domains, time zones, last reboots, machine types, and dimensions in terms of CPU and physical memory
    - The Azure AD Connect Health Agent version
- Statistical information for the last week, day, or past 6 hours in graphs on the following aspects:
    - Token requests per second
    - Outstanding token requests (proxy); rejected token requests per second (proxy); established TCP connections
    - Token request latency
    - Used memory as a percentage of the total memory user processor (percentage)
- Combined statistical information for the last week, day, or past 6 hours in graphs on the information that is available in the preceding individual graphs
- Usage analytics in terms of total requests, total failed requests, and user count per application (relying party trust), server, authentication method, network location, and for workplace-joined devices only
- Reports for the last 30 days on bad password attempts and risky **Internet Protocol** (**IP**) addresses

There's more...

If, for any reason, the configuration of the Azure AD Connect Health Agent for AD FS fails, rerun the following line of PowerShell in an elevated Windows PowerShell window:

```
Register-AzureADConnectHealthADFSAgent
```

Common scenarios where the configuration fails are when IE ESC is still enabled or when the AD FS servers and/or WAP servers require proxy configuration to connect to Azure AD Connect Health endpoints.

Configuring Azure AD Connect Health for AD DS

Azure AD Connect Health can be expanded to include monitoring of the domain controllers of your organization's Active Directory implementation. This recipe shows how to do this.

Getting ready

To complete this recipe, sign in to Azure AD with an account that has the **Global administrator** role assigned to it. If the organization uses the Azure AD PIM feature, activate the **Global administrator** role in advance. Access to the Azure AD Connect Health dashboard can be delegated through its **role-based access-control** (**RBAC**) IAM settings.

The Azure AD Connect Health functionality requires Azure AD Premium P1 licenses or Microsoft licenses that include the P1 license, such as Azure AD Premium P2, EMS E3, EMS A3, Microsoft 365 E3, or Microsoft 365 Business licenses.

Ensure all domain controllers run Windows PowerShell `4.0` or above and have IE ESC turned off.

How to do it...

Configuring Azure AD Connect Health for AD DS consists of three steps:

- Downloading the agent
- Installing and configuring the agent
- Consuming information in the Azure AD Connect Health dashboard

Downloading the agent

Perform the following steps to download the Azure AD Connect Health Agent for AD DS:

1. Navigate your browser to `https://aad.portal.azure.com`.
2. Sign in with an account in Azure AD that has the **Global administrator** role assigned.
3. Perform MFA when prompted.
4. In the left navigation pane, click **Azure Active Directory**.
5. In the **Azure Active Directory** navigation pane, click **Azure AD Connect** to navigate to the **Azure AD Connect** pane.

6. In the **Azure AD Connect** main pane, under **Health and analytics**, follow the **Azure AD Connect Health** link to go to the **Azure Active Directory Connect Health | Quick start** pane.
7. In the **Azure Active Directory Connect Health | Quick start** pane, click the **Download Azure AD Connect Health Agent for AD DS** link.
8. Download the agent to a location where it is accessible to the domain controllers, or download it and copy it to the hard disks of these servers.

Installing and configuring the agent

Perform the following steps to install and configure the Azure AD Connect Health Agent for AD DS on each domain controller in your environment:

1. Run `AdHealthAddsAgentSetup.exe`. The **Azure AD Connect Health AD DS Agent** window appears.
2. In the **Azure AD Connect Health AD FS Agent** window, click **Install**.
3. On the **Setup Successful** screen, click **Configure Now**. An elevated Windows PowerShell window appears that executes the `Register-AzureADConnectHealthADDSAgent` cmdlet. Then, the **Sign in to your account** window appears.
4. Sign in with an account in Azure AD that has the **Global administrator** role assigned.
5. Perform MFA when prompted.
6. Close the Windows PowerShell window.

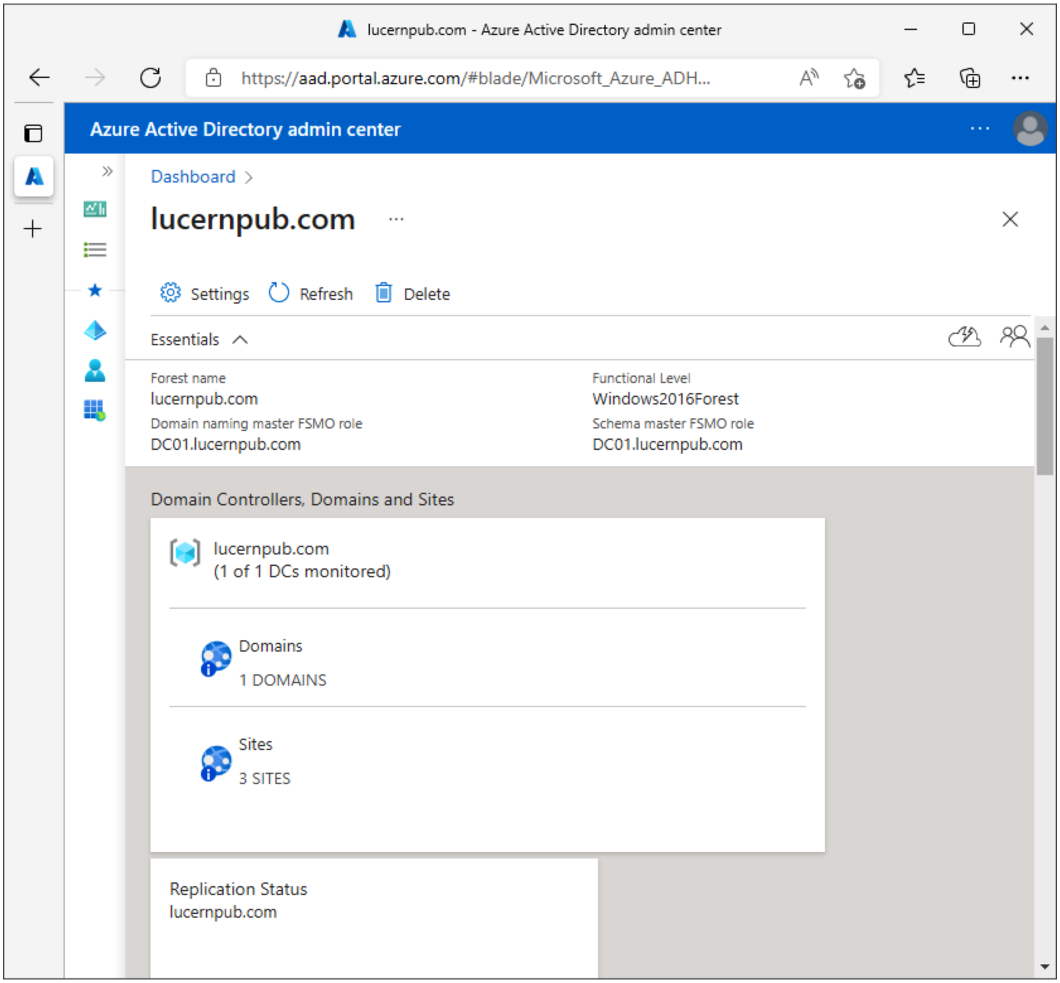
Repeat the preceding steps on all domain controllers.

Consuming information in the Azure AD Connect Health dashboard

Perform these steps to consume information from the Azure AD Connect Health Agent for AD DS installations in the Azure AD Connect Health dashboard:

1. Navigate your browser to `https://aad.portal.azure.com`.
2. Sign in with an account in Azure AD that has the **Global administrator** role assigned.
3. Perform MFA when prompted.
4. In the left navigation pane, click **Azure Active Directory**.
5. In the **Azure Active Directory** navigation pane, click **Azure AD Connect** to navigate to the **Azure AD Connect** pane.

6. In the **Azure AD Connect** main pane, under **Health and analytics**, follow the **Azure AD Connect Health** link to go to the **Azure Active Directory Connect Health | Quick start** pane.
7. In the **Azure Active Directory Connect Health** navigation pane, click **AD DS services**.
8. In the **AD DS services** pane, click the DNS domain name of the Active Directory forest you want to monitor to navigate to its dashboard:


lucernpub.com - Azure Active Directory admin center
https://aad.portal.azure.com/#blade/Microsoft_Azure_ADH...
Azure Active Directory admin center
Dashboard >
lucernpub.com
Settings
Refresh
Delete
Essentials

Forest name
lucernpub.com

Domain naming master FSMO role
DC01.lucernpub.com

Functional Level
Windows2016Forest

Schema master FSMO role
DC01.lucernpub.com

Domain Controllers, Domains and Sites

lucernpub.com
(1 of 1 DCs monitored)

Domains
1 DOMAINS

Sites
3 SITES

Replication Status
lucernpub.com

Figure 16.15 – Azure AD Connect Health dashboard for an AD forest

How it works...

The Azure AD Connect Health Agent for AD DS is an additional agent that can be installed on domain controllers. Microsoft recommends installing the agent on all domain controllers for complete insight.

Data is available in the Azure AD Connect Health dashboard in the Azure portal but is only accessible when at least one Azure AD Premium P1 license is attached to the Azure AD tenant per configured Azure AD Connect installation. For correct licensing, every monitored domain controller requires an additional 25 Azure AD Premium P1 licenses attached to the Azure AD tenant.

The Azure AD Connect Health dashboard provides information on the following:

- The Active Directory forest
    - The domain controller that holds the Domain Naming Master **Flexible Single-Master Operation** (**FSMO**) role
    - The domain controller that holds the Schema Naming Master FSMO role
    - Its **forest functional level** (**FFL**)
- Active Directory domains
    - Its replication status
    - The domain controllers and their sites, FSMO roles, status, active alerts, and whether they are global catalogs
- Alerts
- Combined statistical information for the last week, day, or past 6 hours in graphs on the following aspects:
    - Successful **Lightweight Directory Access Protocol** (**LDAP**) binds per second
    - Kerberos authentications per second
    - **New Technology LAN Manager** (**NTLM**) authentications per second
    - Replication queue length

Configuring Azure AD PIM

This recipe shows how to get the most out of Azure AD **Privileged Identity Management** (**PIM**).

Getting ready

To complete this recipe, sign in to Azure AD with an account that has the **Global administrator** role assigned to it.

The PIM functionality requires Azure AD Premium P2 licenses or Microsoft licenses that include the P2 license, such as EMS E5, EMS A5, or Microsoft 365 E5.

People whose Azure AD accounts are assigned privilege roles in PIM and are required to perform MFA to request the role should already have registered at least one MFA method.

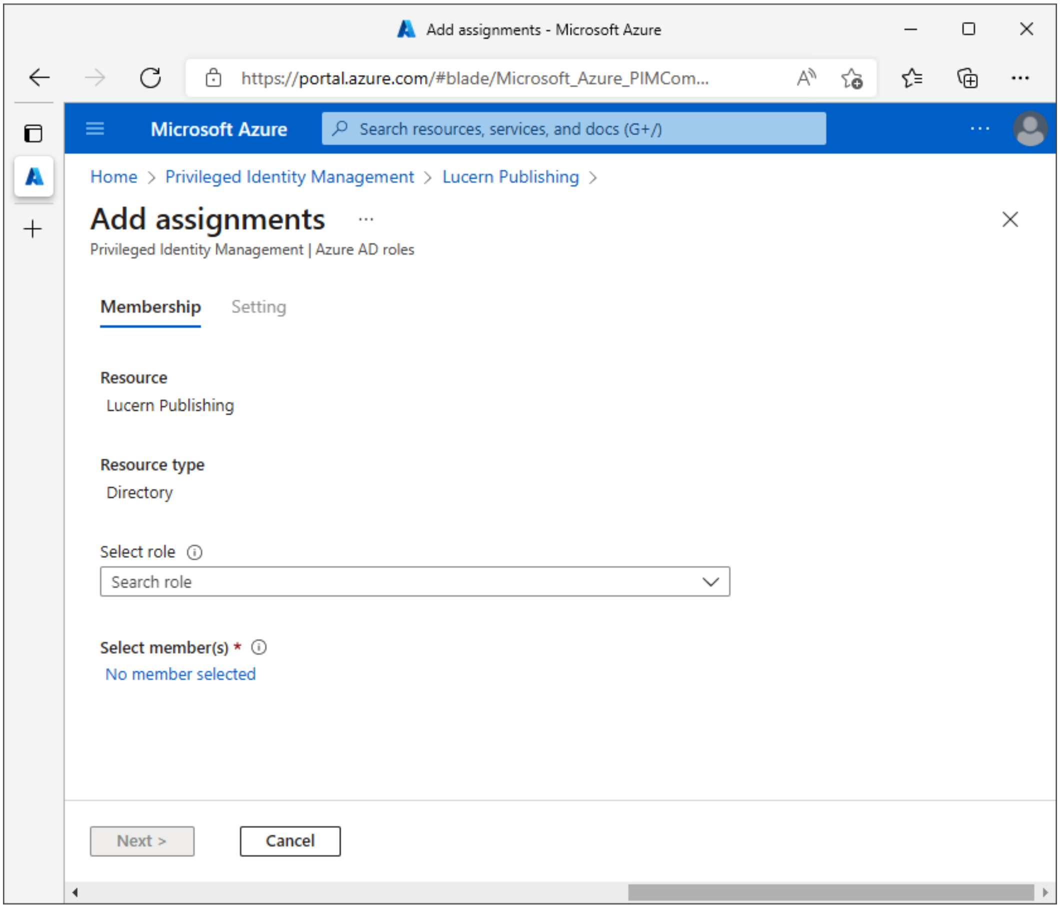
> **Tip**
>
> Microsoft recommends configuring at least two MFA methods that are not tied to the same mobile number or mobile device.

How to do it...

Perform these steps to set up a person with the **Conditional Access administrator** privileged role in PIM that requires MFA and a justification to request it:

1. Navigate your browser to `https://portal.azure.com`.
2. Sign in with an account in Azure AD that has the **Global administrator** role assigned.
3. Perform MFA when prompted.
4. In the top bar in the search field, search for **Azure AD Privileged Identity Management**. Click its search result to navigate to the **Privileged Identity Management | Quick start** pane.
5. In the **Privileged Identity Management** navigation pane, click **Azure AD roles**.
6. In the **Roles** navigation pane, click **Assignments** to navigate to the **Assignments** pane.

7. In the **Assignments** pane, click + **Add assignments**. An **Add assignments** pane will appear:


Add assignments - Microsoft Azure
https://portal.azure.com/#blade/Microsoft_Azure_PIMCom...
Microsoft Azure
Search resources, services, and docs (G+/)
Home > Privileged Identity Management > Lucern Publishing >

Add assignments

Privileged Identity Management | Azure AD roles

Resource
Lucern Publishing

Resource type
Directory

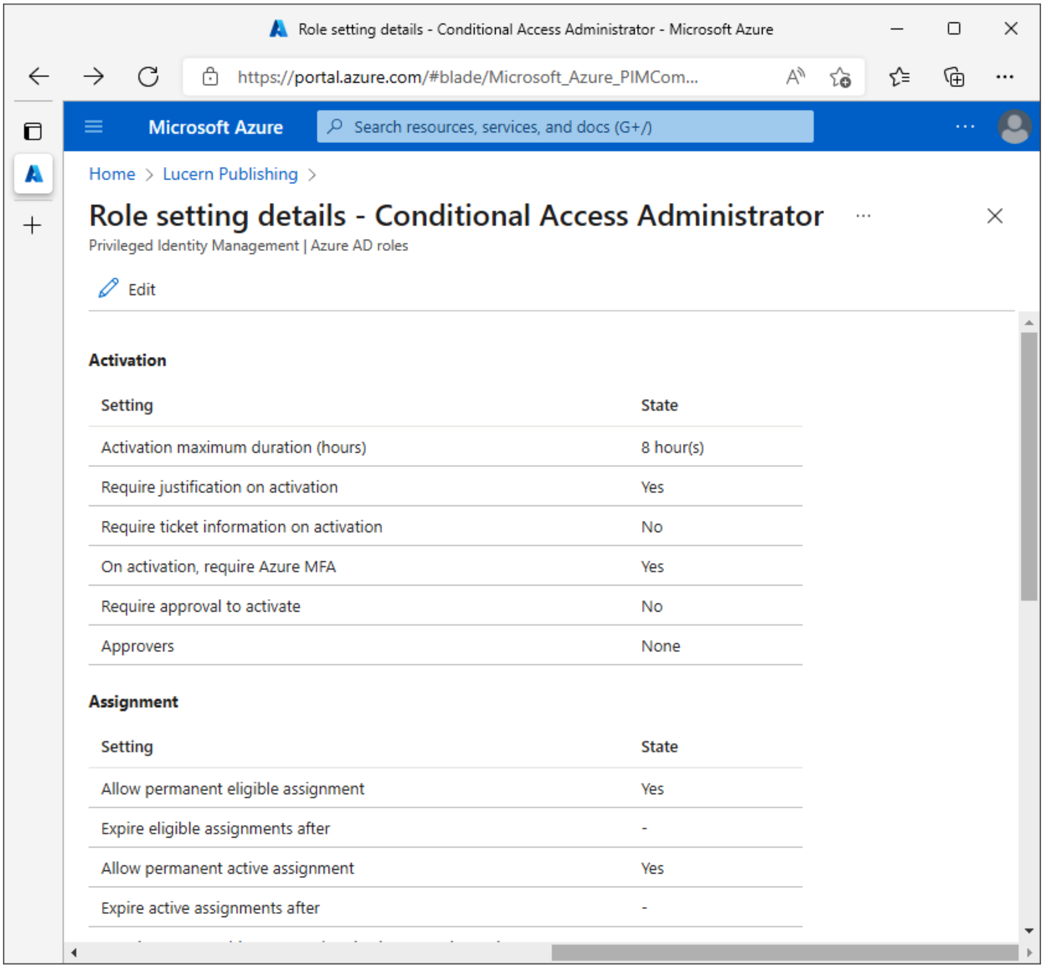
Select role
Search role

Select member(s) *
No member selected

Figure 16.16 – Add assignments pane

8. From the **Select role** drop-down list, select the **Conditional Access administrator** role.
9. Under **Select members**, click **No member selected**. A **Select a member** blade will appear. Select a user account to which you want to assign the role. Click **Select** at the bottom of the **Select a member** blade.
10. Click **Next >**.
11. For the **Assignment type** setting, ensure the **Eligible** option is selected.
12. Click **Assign**.

13. In the **Roles** navigation pane, click **Settings**.
14. In the **Settings** pane, in the list with roles, click the **Conditional Access administrator** role to go to its role setting details pane.
15. In the **Role setting details - Conditional Access Administrator** pane, review the settings:


Role setting details - Conditional Access Administrator - Microsoft Azure
https://portal.azure.com/#blade/Microsoft_Azure_PIMCom...
Microsoft Azure
Search resources, services, and docs (G+/)
Home > Lucern Publishing >
Edit

Role setting details - Conditional Access Administrator

Privileged Identity Management | Azure AD roles

Activation

Setting	State
Activation maximum duration (hours)	8 hour(s)
Require justification on activation	Yes
Require ticket information on activation	No
On activation, require Azure MFA	Yes
Require approval to activate	No
Approvers	None

Assignment

Setting	State
Allow permanent eligible assignment	Yes
Expire eligible assignments after	-
Allow permanent active assignment	Yes
Expire active assignments after	-

Figure 16.17 – Role settings details - Conditional Access Administrator pane

16. To change any of these settings, click **Edit** at the top of the pane to go to the **Edit role setting** pane. After making the required changes, click **Update** at the bottom of the pane.

How it works...

Azure AD PIM offers management, control, and monitoring of privileged access. By minimizing the number of administrators and the time these people are equipped as administrators with their privileges, Azure AD tenants can be hardened.

Always-on administrative privileges might seem like a good idea but are not. Even the most dedicated administrator would only need these privileges 10 out of 24 hours and only on some weekends. Azure AD PIM offers a solution to this situation: automatically expiring privileges that administrators need to specifically request.

With optimal settings, up to four people are assigned permanent **Global administrator** privileges. One additional account is configured as a break-glass account and also has these privileges. The roles for these accounts are out of the scope of PIM. All other people who need administrative privileges are provided with **just-in-time** (**JIT**), time-bound privileges in restricted administrator roles, such as the **SharePoint administrator**, **Exchange administrator**, and **Intune administrator** roles. Their roles are in scope for PIM.

When a role is in scope for PIM, requesting the privileges of this role can be configured to require MFA, a justification, and/or approval. Notifications can be configured, as well as access reviews.

Every request is logged.

The activation duration of privileges for a role can be configured between 0.5 hours and 24 hours. The default duration (8 hours) might not be sufficient time for certain scripts for certain people in certain roles to complete.

Configuring Azure AD Identity Protection

Azure Identity Protection offers additional protection to organizations that worry about password breaches. This recipe shows how to configure an MFA registration policy.

Getting ready

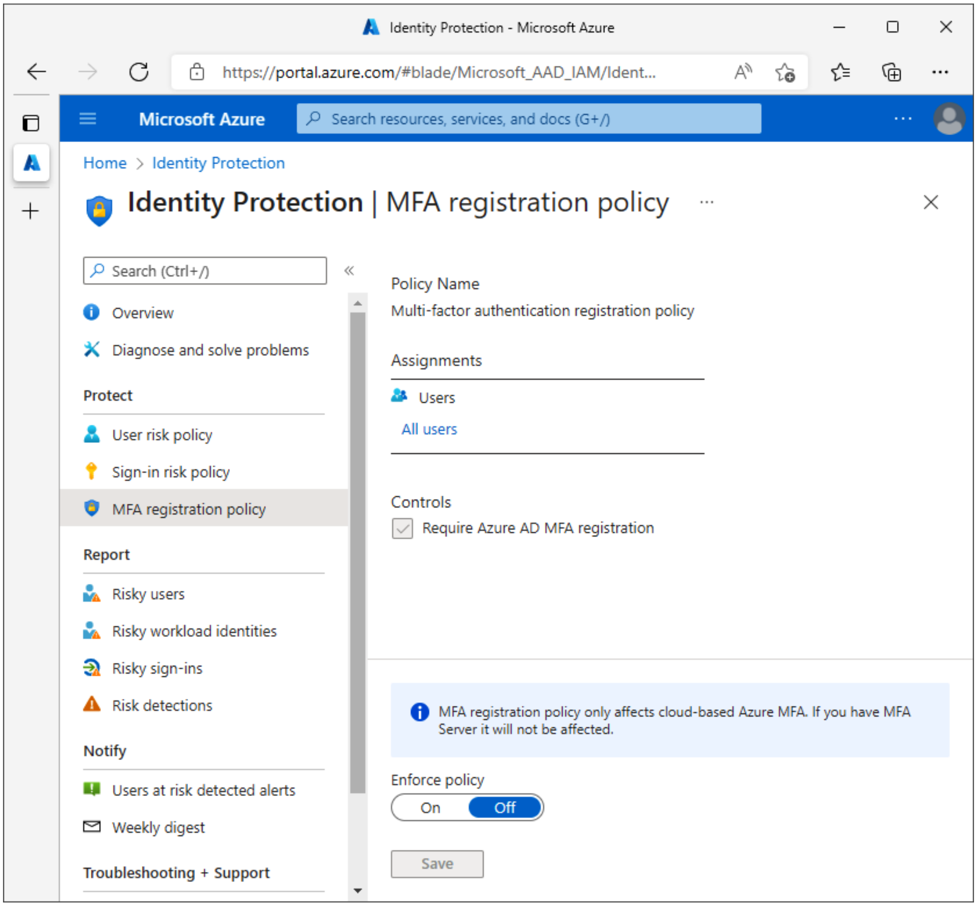
To complete this recipe, sign in to Azure AD with an account that has the **Global administrator** role assigned to it. If the organization uses the Azure AD PIM feature, activate the **Global administrator** role in advance.

The **Azure AD Identity Protection** functionality requires Azure AD Premium P2 licenses or Microsoft licenses that include the P2 license, such as EMS E5, EMS A5, Microsoft 365 E5, Microsoft 365 Information Protection and Compliance, or Microsoft 365 Business Premium licenses.

How to do it...

Perform the following steps:

1. Navigate your browser to `https://portal.azure.com`.
2. Sign in with an account in Azure AD that has the **Global administrator** role assigned.
3. Perform MFA when prompted.
4. In the top bar in the search field, search for **Azure AD Identity Protection**. Click its search result to navigate to the **Privileged Identity Management | Overview** pane.
5. In the **Identity Protection** navigation pane, click **MFA registration policy** to navigate to the **Identity Protection | MFA registration policy** pane:


Identity Protection - Microsoft Azure
https://portal.azure.com/#blade/Microsoft_AAD_IAM/Ident...
Microsoft Azure
Search resources, services, and docs (G+/)
Home > Identity Protection
Identity Protection | MFA registration policy
Search (Ctrl+/)
Overview
Diagnose and solve problems
Protect
User risk policy
Sign-in risk policy
MFA registration policy
Report
Risky users
Risky workload identities
Risky sign-ins
Risk detections
Notify
Users at risk detected alerts
Weekly digest
Troubleshooting + Support
Save

Policy Name
Multi-factor authentication registration policy

Assignments
Users
All users

Controls
☑ Require Azure AD MFA registration

MFA registration policy only affects cloud-based Azure MFA. If you have MFA Server it will not be affected.

Enforce policy
On [✓ Off]

Figure 16.18 – Identity Protection | MFA registration policy pane

6. Change the **Enforce policy** setting to **On**.
7. Click **Save** at the bottom of the pane.

How it works...

Azure AD Identity Protection is enabled for every Azure AD account and every Microsoft account, formerly known as a Windows Live ID. Every authentication goes through the Identity Protection funnel, where Microsoft's **machine learning** (**ML**) model tries to distinguish risky and bad sign-ins from valid ones. Accounts that see a lot of risky sign-ins, or have their password breached, are marked as risky users.

Azure Identity Protection offers additional investigation and configuration mechanisms for risky sign-ins and risky users. In the **Azure AD Identity Protection** dashboard, administrators can do the following:

- Investigate users flagged for risk, risk events, and vulnerabilities.
- Configure MFA registration, user risk policies, and sign-in risk policies.

The investigation options provide administrators with a way to find out why people are required to perform MFA or change their password more often than they (think they) should. In many tenants, **Sign-ins from unfamiliar locations** events might trigger often for traveling colleagues and these risk events trigger MFA for them. It is recommended to review and dismiss false positives for **Users flagged for risk** and **risk events** regularly.

The **All Users** assignment in Azure AD Identity Protection contains all users, including guest users.

MFA registration policy

Registering all users for MFA provides organizations with a second layer of security, beyond just usernames and passwords.

Administrators can configure the MFA registration policy. The policy requires selected users to register for Azure MFA. Simply switch the state of the policy to **On**. When enabled, people who interactively sign in to their Azure AD accounts for the first time are required to register for MFA. They are allowed a 14-day window for this registration.

User risk policy

With the user risk policy turned on, Azure AD Identity Protection calculates the probability that a user account has been compromised. As an administrator, you can configure a Conditional Access policy with a specific user risk level as a condition—for example, you can block access to sensitive resources or require a password change to get a user account back to a clean state. Only when the account is returned to a clean state is the user allowed to access resources normally.

Sign-in risk policy

Turning on the sign-in risk policy ensures that suspicious sign-ins are challenged for MFA.

In Azure AD Identity Protection, administrators can configure the sign-in risk remediation policy. For the users of this policy, administrators set conditions (risk level) that trigger the policy.

Switch the state of the policy to **On**.

When people who haven't registered MFA trigger the user risk policy and/or sign-in risk policy that requires MFA, they are blocked from accessing resources.

Ensure the MFA registration policy is on for all users who are a part of the user risk policy and/or sign-in risk policy.

Implementing Defender for Identity

Microsoft Defender for Identity offers additional alerts, reports, and hunting capabilities for Active Directory forests. This recipe shows how to deploy the Defender for Identity sensor on your domain controllers.

Getting ready

To complete this recipe, sign in to the Microsoft 365 Defender portal with an account that has the **Global administrator** or **Security administrator** role assigned to it. If the organization uses the Azure AD PIM feature, activate the **Global administrator** or **Security administrator** role in advance.

Microsoft Defender for Identity requires at least one EMS E5 or Microsoft 365 license.

To install the Defender for Identity sensor on your domain controllers, sign in with an account that has local administrator privileges on the domain controllers. By default, members of the **Administrators**, **Domain Admins**, and **Enterprise Admins** security groups in Active Directory have these privileges.

How to do it…

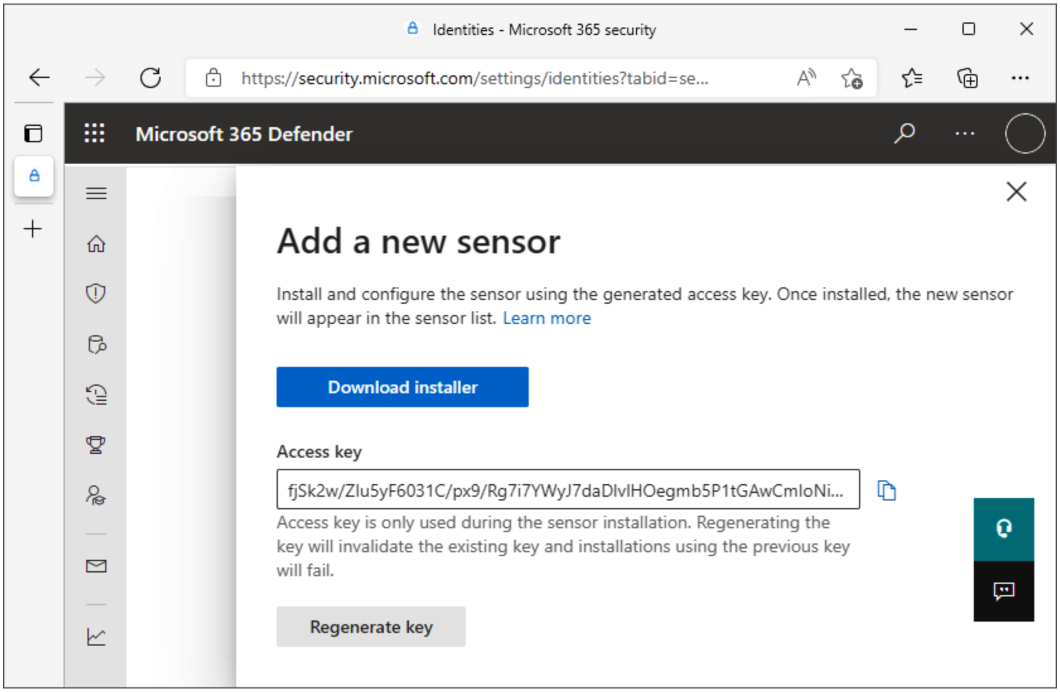
Implementing Defender for Identity consists of the following three steps:

- Downloading the sensor
- Installing and configuring the sensor
- Consuming information in the **Microsoft 365 Defender** portal

Downloading the sensor

Perform these steps to download the Defender for Identity sensor from the **Microsoft 365 Defender** portal:

1. Navigate your browser to `https://security.microsoft.com`.
2. Sign in with an account in Azure AD that has the **Global administrator** or **Security administrator** role assigned.
3. Perform MFA when prompted.
4. In the left navigation pane, click **Settings** to navigate to the **Settings** pane.
5. In the **Settings** pane, click **Identities**.
6. In the left navigation pane, under **General**, click **Sensors** to navigate to the **Sensors** pane.
7. At the bottom of the **Sensors** pane, click **Add sensor**. The **Add a new sensor** blade appears:


Identities - Microsoft 365 security
https://security.microsoft.com/settings/identities?tabid=se...
Microsoft 365 Defender
Download installer
Regenerate key

Add a new sensor

Install and configure the sensor using the generated access key. Once installed, the new sensor will appear in the sensor list. Learn more

Access key

fjSk2w/Zlu5yF6031C/px9/Rg7i7YWyJ7daDlvlHOegmb5P1tGAwCmloNi...

Access key is only used during the sensor installation. Regenerating the key will invalidate the existing key and installations using the previous key will fail.

Figure 16.19 – Add a new sensor blade

8. Click **Download installer**. Download the sensor package to a location where it is accessible to domain controllers, or download it and copy it to the hard disks of these servers.
9. Copy the **Access key** value.

Installing and configuring the sensor

Perform the following steps to install and configure the sensor on a domain controller:

1. Right-click `Azure ATP Sensor Setup.zip` and select **Properties** from the context menu. The **Azure ATP Sensor Setup Properties** window appears.
2. At the bottom of the **General** tab of the **Azure ATP Sensor Setup Properties** window, select the **Unblock** option.
3. Click **OK** to close the **Azure ATP Sensor Setup Properties** window.
4. Right-click `Azure ATP Sensor Setup.zip` again. This time, select **Extract all…** from the context menu. The **Extract Compressed (Zipped) Folders** window appears.
5. Click **Extract**.

6. Double-click `Azure AD Sensor Setup.exe`. The **Azure Advanced Threat Protection Sensor** window appears.
7. On the **Install Microsoft Defender for Identity Sensor** screen, click **Next**.
8. On the **Sensor deployment type** screen, click **Next**.
9. On the **Configure the Sensor** screen, enter the access key in the **Access key** field.
10. Click **Install**.
11. On the **Install completed successfully** screen, click **Finish** to close the **Azure Advanced Threat Protection Sensor** window:

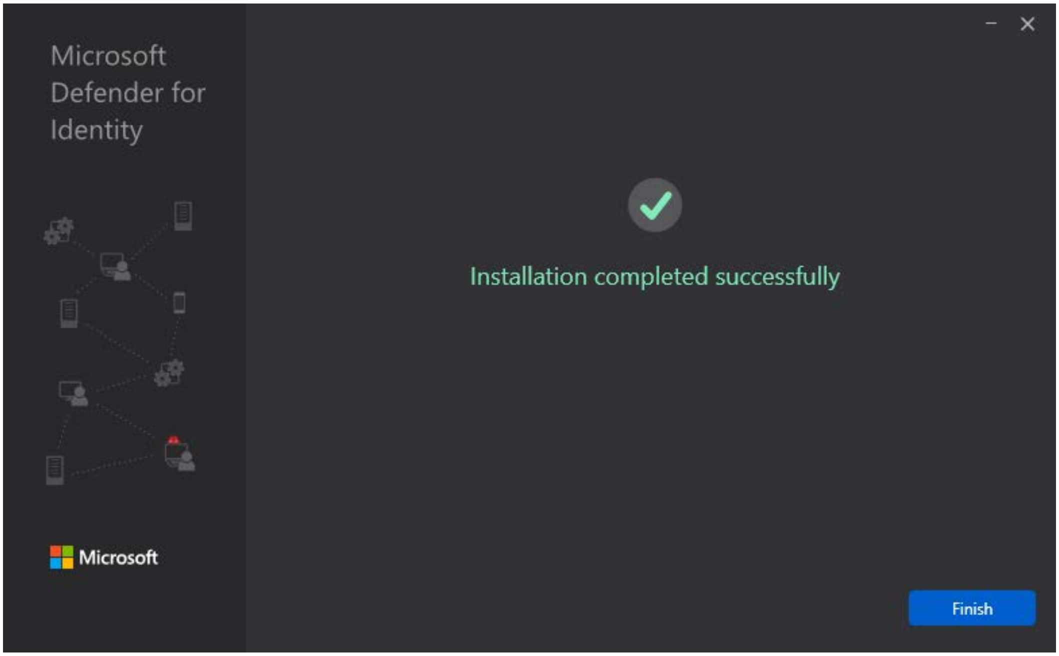

Microsoft Defender for Identity
Installation completed successfully
Microsoft
Finish

Figure 16.20 – Installation completed successfully screen

12. Remove the `Azure ATP Sensor Setup.zip` file and its extracted contents from the domain controller.

Repeat the preceding steps on all domain controllers.

Consuming information in the Microsoft 365 Defender portal

With the sensor installed and transmitting data, information can now be consumed in the **Microsoft 365 Defender** portal. Perform these steps to do so:

1. Navigate your browser to `https://security.microsoft.com`.
2. Sign in with an account in Azure AD that has the **Global administrator** or **Security administrator** role assigned.
3. Perform MFA when prompted.
4. In the left navigation pane, expand **Incidents and alerts** and select **Incidents** to navigate to the **Incidents** pane and view incidents found by Defender for Identity.
5. In the left navigation pane, select **Alerts** to navigate to the **Alerts** pane and view alerts generated by Defender for Identity.

How it works…

Deploying Defender for Identity sensors enables administrators to monitor the Active Directory environment for suspicious activities and risky configurations. The signals sent by these sensors are used by Microsoft's ML model to identify, detect, and investigate advanced threats, compromised identities, and malicious insider actions directed at the organization's Active Directory environment.

Some popular Defender for Identity detections include:

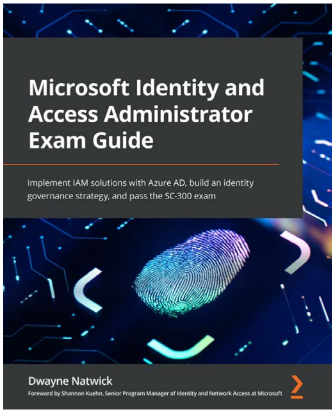
- Suspected overpass-the-hash attack
- Suspected brute-force attack
- Suspected DCSync attack
- Suspected Golden Ticket usage
- Suspected Skeleton Key attack
- Suspected NTLM relay attack
- Suspected rogue Kerberos certificate usage
- Suspicious additions to sensitive groups
- Malicious request of **Data Protection API** (**DPAPI**) master key

Microsoft Defender for Identity was formerly known as **Azure Advanced Threat Protection** (**Azure ATP**).

Index

Other Books You May Enjoy

If you enjoyed this book, you may be interested in these other books by Packt:

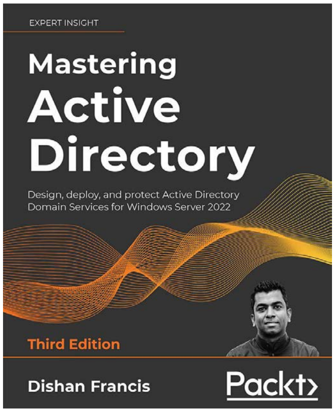

Microsoft Identity and Access Administrator Exam Guide
Implement IAM solutions with Azure AD, build an identity governance strategy, and pass the SC-300 exam
Dwayne Natwick
Foreword by Shannon Kuehn, Senior Program Manager of Identity and Network Access at Microsoft

Microsoft Identity and Access Administrator Exam Guide

Dwayne Natwick

ISBN: 9781801818049

- Understand core exam objectives to pass the SC-300 exam
- Implement an identity management solution with MS Azure AD
- Manage identity with multi-factor authentication (MFA), conditional access, and identity protection
- Design, implement, and monitor the integration of enterprise apps for Single Sign-On (SSO)
- Add apps to your identity and access solution with app registration
- Design and implement identity governance for your identity solution


EXPERT INSIGHT
Mastering Active Directory
Design, deploy, and protect Active Directory Domain Services for Windows Server 2022
Third Edition
Dishan Francis
Packt

Mastering Active Directory - Third Edition

Dishan Francis

ISBN: 9781801070393

- Install, protect, and manage Active Directory Domain Services (Windows Server 2022)
- Design your hybrid identity by evaluating business and technology requirements
- Automate administrative tasks in Active Directory using Windows PowerShell 7.x
- Protect sensitive data in a hybrid environment using Azure Information Protection
- Learn about Flexible Single Master Operation (FSMO) roles and their placement
- Manage directory objects effectively using administrative tools and PowerShell
- Centrally maintain the state of user and computer configuration by using Group Policies
- Harden your Active Directory using security best practices

Packt is searching for authors like you

If you're interested in becoming an author for Packt, please visit `authors.packtpub.com` and apply today. We have worked with thousands of developers and tech professionals, just like you, to help them share their insight with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Share Your Thoughts

Now you've finished *Active Directory Administration Cookbook, Second Edition*, we'd love to hear your thoughts! If you purchased the book from Amazon, please click here to go straight to the Amazon review page for this book and share your feedback or leave a review on the site that you purchased it from.

Your review is important to us and the tech community and will help us make sure we're delivering excellent quality content.