6

Managing Active Directory

In this chapter, we cover the following recipes:

* Installing an Active Directory Forest Root Domain
* Configuring CredSSP
* Installing a Replica Domain Controller
* Installing a Child Domain
* Creating and Managing AD Users and Groups
* Managing Active Directory Computers
* Creating a Group Policy Object for user Logon script
* Adding/Removing users using CSV files
* Finding Expired/Unused AD User and Computer Objects

# Introduction

A core component of almost all organizations' IT infrastructure is Active Directory (AD). AD provides access control, user and system customization, and a wealth of directory and other services. Microsoft first introduced AD with Windows 2000 and has improved and expanded the product with each successive release of Windows Server.

Over the years, Microsoft has made “AD” more of a brand than a single feature. At the cores is Active Directory Domain Services (ADDS). There are four additional Windows Server features under the AD brand:

* AD Certificate Services (AD-CS) - this allows you to issue X.509 certificates for your organization. For an overview of AD-CS, see https://docs.microsoft.com/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/hh831740(v=ws.11).
* AD Federation Services (AD-FS) - this feature enablers you to federate identity with other organisations to facilitate interworking. You can find an overview to AD-FS at https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/hh831502(v=ws.11).
* AD Lightweight Directory Services (AD-LDS) - This provides rich directory services for use by applications. You can find an overview of AD-LDS at https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/hh831593(v=ws.11)
* AD Rights Management Services (AD-RMS ) - RMS enables you to control the rights to document access so as to limit information leakage. For an overview of RMS, see https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/hh831364(v=ws.11).

Note that the overview documents referred to above are older documents based on Windows Server 2012. At the time of writing, the documentation teams have not updated these to reflect fully the latest version of Windows server. The overvies and the basic operation of these features remains largely unchanged.

Active Directory’s domain service is complex and there are a lot of moving parts. With AD, you have a logical structure consisting of Forests, Domains, Domain Trees and Organisational Units. You also have the physical structure, including Domain Controllers and global catalogs. There is also a replication mechism to replicate objects across your domain.

A forest is a top-level container that houses domains. A forest is a security boundary although you can set up cross-forest trusts to enable interworking between multiple forests. Forest (and domain) names are based on DNS

A domain are a collection of objects, including users, computers, policies, and much more. You create a forest by installing the forest’s first domain controller. Domains trees are collections of domains that are grouped together in a hirarchcal structure. Most organisations use a single doman (and domain tree) withing a single forest. Multiple domains in one or more domain trees is also supported but is not recommended.

A domain controller (DC) is a Windows Server running AD and holding the objects for a given domain. ALL domains must have at least one DC although best practice is to always have at least two. You install the AD DS service onto your server, then promote the server to be a DC.

The global Catalog (GC) is a partial replica of objects from every domain in an object to enable searching. Exchange, for example, uses the GC heavily. You can have the GC service on some or all DCs in your forest. Generally, you install the GC facility at the same time you promote a Windows Server to be a DC.

Using AD Domain Services (or AD) and PowerShell, you can deploy your domain controllers throughout your organization. Use the “Installing Active Directory and Forest Root Domain” recipe to install a forest root domain controller and establish an AD Forest.

Installing features and services using PowerShell in a domain environment often makes use of remoting which in turn requires authentication. From one machine, you use PowerShell remoting to perform operations on other systems and you need the correct credentials for those operations. Sometimes, you may encounter the well understood Kerberos double hop problem. In “Configuring CredSSP”, you install the Credential Security Subsystem Provider to manage the issue. For a better explanation of this issue see https://docs.microsoft.com/en-us/powershell/scripting/learn/remoting/ps-remoting-second-hop?view=powershell-7.1.

Once you have a first DC in your Reskit.Org forest, you should add replica DC to ensure reliable domain operations. In “Installing a Replica Domain Controller”, you add a second DC to your domain. In “Installing a Child Domain, you extend the forest and add a child domain to your forest.

AD is based on a database of objects, which include Users, Computers, and Groups. In the “Managing AD Users and Groups” recipe, you create, move, and remove user and group objects as well as create and use Organizational Units (OUs). In “Managing Active Directory Computers” you manage the computers in your Active Directory, including joinoing workgroup systems to the domain. In the “Adding Users to AD via a CSV” recipe, you add users to your AD making use of a comma-separated value file containing details of users.

Group policy is another important feature of Active Directory. With group policy, you can define policies for users and computers that are applied automatically to the user and/or computer. In the “Creating a Group Policy Object for User Logon Script” recipe, you create a simple GPO and observe applying that policy.

In the final recipe, “Reporting on AD users”, you examine the AD to find details on users that have not logged on for a while, computers that have not been used for a while, and users who are members of special security groups (such as enterprise administrators). The final recipe, Finding expired computers and disabled users, finds computer and user objects that have not been used in a while. These final two recipes help to keep your AD free of stale objects or objects that could represent a security risk.

# Installing an Active Directory Forest Root Domain

You create an AD forest by creating your first domain controller. Installing Active Directory and DNS has always been fairly straightforward. You can always use the Server Manager GUI, but using PowerShell is also straightforward.

In order to create a DC, you start with a system running Windows Server. You then add the AD DS services windows feature to the server. Finally, you create your first DC. This creates a single domain controller, DC1.Reskit.Org for the Reskit.Org domain.

## Getting Ready

You run this recipe on DC1, a workgroup server on which you have installed PowerShell 7 and VS Code.

## How to do it...

1. Installing the AD Domain Services feature and management tools

Install-WindowsFeature -Name AD-Domain-Services -IncludeManagementTools

1. Importing ADDSDeployment module

Import-Module -Name ADDSDeployment

1. Examining the commands in the ADDSDeployment module

Get-Command -Module ADDSDeployment

1. Creating a secure password for Administrator

$PSSHT = @{

  String      = 'Pa$$w0rd'

  AsPlainText = $true

  Force       = $true

}

$PSS = ConvertTo-SecureString @PSSHT

1. Testing DC Forest installation starting on DC1

$FOTHT = @{

  DomainName           = 'Reskit.Org'

  InstallDNS           = $true

  NoRebootOnCompletion = $true

  SafeModeAdministratorPassword = $PSS

  ForestMode           = 'WinThreshold'

  DomainMOde           = 'WinThreshold'

}

Test-ADDSForestInstallation @FOTHT -WarningAction SilentlyContinue

1. Creating Forest Root DC on DC1

$ADHT = @{

  DomainName                    = 'Reskit.Org'

  SafeModeAdministratorPassword = $PSS

  InstallDNS                    = $true

  DomainMode                    = 'WinThreshold'

  ForestMode                    = 'WinThreshold'

  Force                         = $true

  NoRebootOnCompletion          = $true

  WarningAction                 = 'SilentlyContinue'

}

Install-ADDSForest @ADHT

1. Checking Key AD and related services

Get-Service -Name DNS, Netlogon

1. Checking DNS Zones

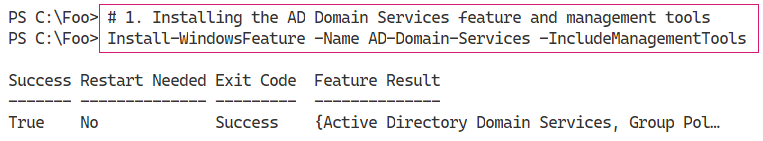
Get-DnsServerZone

1. Restarting DC1 to complete promotion

Restart-Computer -Force

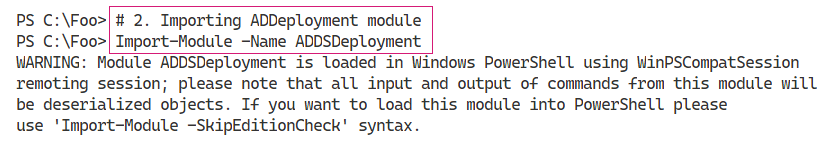
## How it works...

In step 1, you install the AD-Domain-Services feature. This feature enables you to deploy a server as a domain controller. The output of this command looks like this:



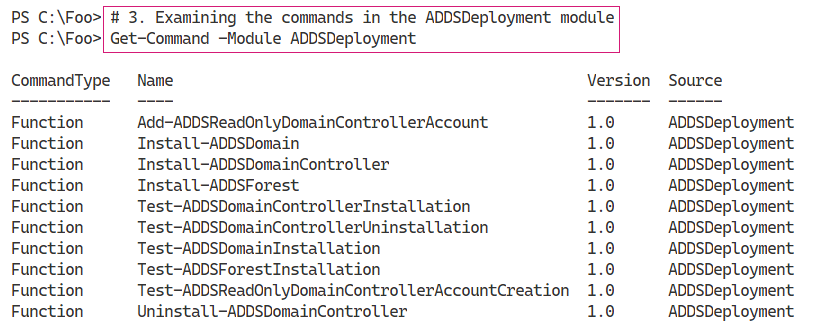
1. Insert image B42024\_06\_01.png

In step 2, you manually import the ADDSDeployment module. Since this module is not supported natively by PowerShell 7, this step loads the module using the Windows PowerShell compatibility feature. The output of this command looks like this:



1. Insert image B42024\_06\_02.png

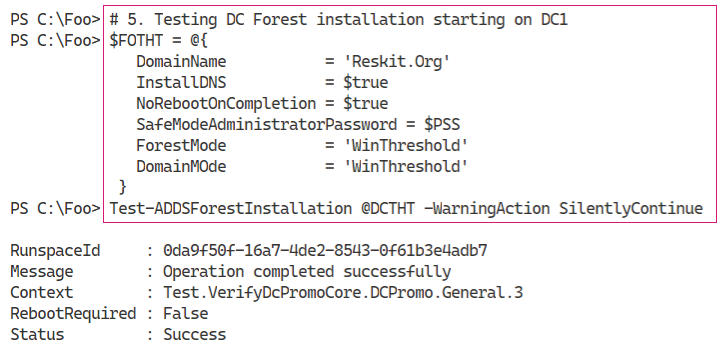
In step 3, you use the Get-Command cmdlet to discover the commands contained in the ADDSDeployment module, which looks like this:



1. Insert image B42024\_06\_03.png

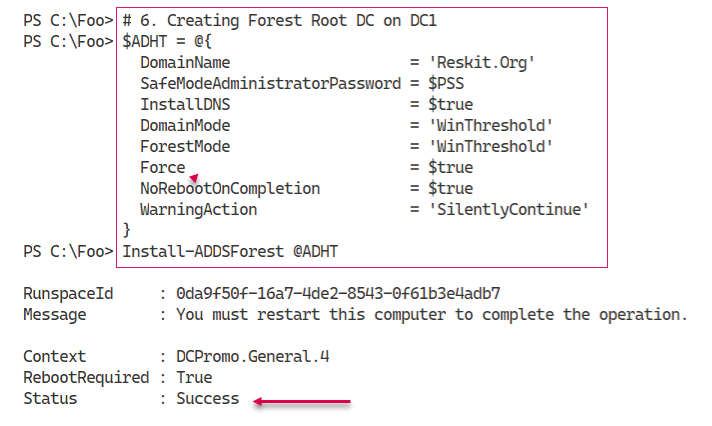
With Step 4, you create a secure string password to use as the Administrator password in the domain you are creating. This step produces output.

Prior to promoting a server to be a DC, it’s useful to test to ensure that as far as possible, a promotion would be successful. In step 5, you use the Test-ADDSForestInstallation command to check to see whether you can promote DC1 to be a DC in the Reskit.Org domain. The output of this command looks like this:



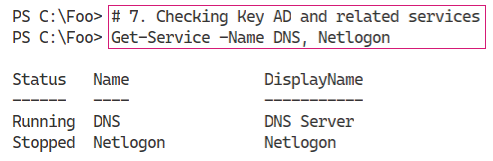
1. Insert image B42024\_06\_04.png

In step 6, you promote DC1 to be the first domain controller in a new domain, Reskit.Org. The output looks like this:



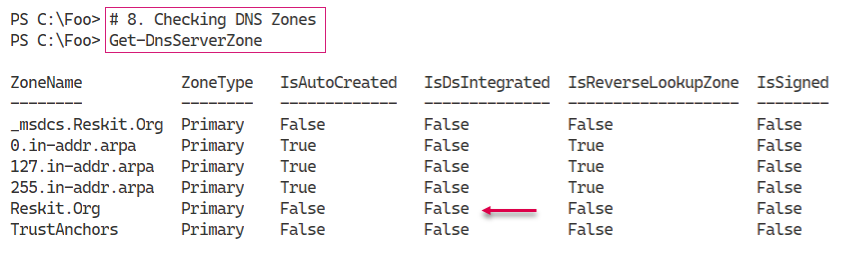
1. Insert image B42024\_06\_05.png

After the promotion is complete, you can check key services which are required for Active Directory. Checking the Netlogon and DNS Services, which you do in step 7, should look like this:



1. Insert image B42024\_06\_06.png

When you promoted DC1, you also specified that DNS is installed on DC1. In step 8, you check on the zones created by the DC promotion process, which looks like this:



1. Insert image B42024\_06\_07.png

To complete the promotion process, you need to reboot, which you do in step 9, generating no actual output.

## There's more...

The cmdlets that enable you to promote a server to be a DC are not installed on a server system by default. Adding the Active Directory Domain Services Windows feature, in step 1, adds the necessary cmdlets to the system.

In step 6, you install AD and direct that a DNS Server should also be installed - and you check for its presence in step 7. In step 8, you view the DNS zones created automatically by the promotion process. A key DNS domain is Reskit.Org as specified in the DomainName paremeter to Install‑ADDSForest . This DNS domain created but is, at this point, still a NON-AD integrated zone. Once you reboot the service, this zone should become AD-integrated (and set for secure updates only).

Once you complete the verification of a successful AD installation, you reboot the server. After the restart, there are further tests that you should run, as we show in the next recipe, “Testing an AD installation”

# Testing an AD Installation

In “Installing an Active Directory Forest Root Domain”, you installed AD on SRV1. In that recipe, you installed AD (without rebooting) and tested certain services. After the required reboot (which you completed at the end of the previous recipe), it is useful to check to ensure that your domain is fully up, running, and working correctly. In this recipe, you examine core aspects of the AD infrastructure on your first DC.

## Getting Ready

You run this recipe on DC1, the first domain controller in the Reskit.Org domain after you have promoted it to be a DC. You should logon as Reskit\Administrator.

## How to do it...

1. Examining AD Root Directory Service Entry (DSE)

Get-ADRootDSE -Server DC1.Reskit.Org

1. Viewing AD forest details

Get-ADForest

1. Viewing AD domain details

Get-ADDomain

1. Checking Netlogon and DNS services

Get-Service NetLogon, DNS

1. Getting Initial AD Users

Get-ADUser -Filter \* |

  Sort-Object -Property Name |

    Format-Table -Property Name, DistinguishedName

1. Getting initial AD groups

Get-ADGroup -Filter \*  |

  Sort-Object -Property Groupscope,Name |

    Format-Table -Property Name, GroupScope

1. Examining Enterprise Admins group membership

Get-ADGroupMember -Identity 'Enterprise Admins'

1. Checking DNS Zones on DC1

Get-DnsServerZone -ComputerName DC1

1. Testing domain name DNS resolution

Resolve-DnsName -Name Reskit.Org

## How it works...

After you completed the installation of AD, and rebooted, in step 1, you examind the AD Directory Service Entry DSE), which looks like this:

1. Insert image B42024\_06\_08.png

## There's more

In step 1, you viewed the DSE for your domain. The DSE, a component of LDAP directories is information about your directory. The DSE is available without requiring authentication. For a more detailed look a the Root DSE, see https://docs.microsoft.com/windows/win32/adschema/rootdse.

# Installing a Replica Directory Controller

In “Installing an Active Directory Forest Root Domain”, you installed AD on SRV1.

## Getting Ready

You run this recipe on SRV1, a workgroup server on which you have installed PowerShell 7 and VS Code.

## How to do it...

1. Step 1

## How it works...

1. Insert image B42024\_06\_01.png

## There's more...

r other .NET Language) code.

# Installing a Child Domain

Stuff

## Getting Ready

You run this recipe on SRV1, a workgroup server on which you have installed PowerShell 7 and VS Code.

## How to do it...

1. Step 1

## How it works...

1. Insert image B42024\_06\_01.png

## There's more...

r other .NET Language) code.

# Creating and Managing AD Users and Groups

Stuff

## Getting Ready

You run this recipe on SRV1, a workgroup server on which you have installed PowerShell 7 and VS Code.

## How to do it...

1. Step 1

## How it works...

1. Insert image B42024\_06\_01.png

## There's more...

r other .NET Language) code.

# Managing Active Directory Computers

Stuff

## Getting Ready

You run this recipe on SRV1, a workgroup server on which you have installed PowerShell 7 and VS Code.

## How to do it...

1. Step 1

## How it works...

1. Insert image B42024\_06\_01.png

## There's more...

r other .NET Language) code.

# Adding/Removing users using CSV files

Stuff

## Getting Ready

You run this recipe on SRV1, a workgroup server on which you have installed PowerShell 7 and VS Code.

## How to do it...

1. Step 1

## How it works...

1. Insert image B42024\_06\_01.png

## There's more...

r other .NET Language) code.

# Creating Group Policy Objects

Stuff

## Getting Ready

You run this recipe on DC1, a domain controller in the Reskit.Org domain. You created this DC in “Installing an Active Directory Forest Root Domain”.

## How to do it...

1. Creating a Group Policy object

$Pol = New-GPO -Name ITPolicy -Comment "IT GPO" -Domain Reskit.Org

1. Ensuring just computer settings are enabled

$Pol.GpoStatus = 'UserSettingsDisabled'

1. Configuring the policy with two registry based settings

$EPHT1= @{

  Name   = 'ITPolicy'

  Key    = 'HKLM\Software\Policies\Microsoft\Windows\PowerShell'

  ValueName = 'ExecutionPolicy'

  Value  = 'Unrestricted'

  Type   = 'String'

}

Set-GPRegistryValue @EPHT1 | Out-Null

$EPHT2= @{

  Name   = 'ITPolicy'

  Key    = 'HKLM\Software\Policies\Microsoft\Windows\PowerShell'

  ValueName = 'EnableScripts'

  Type   = 'DWord'

  Value  = 1

}

Set-GPRegistryValue @EPHT2 | Out-Null

1. Creating a screen saver GPO

$Pol2 = New-GPO -Name 'Screen Saver Time Out'

$Pol2.GpoStatus   = 'ComputerSettingsDisabled'

$Pol2.Description = '15 minute timeout'

1. Seting a Group Policy enforced registry value

$EPHT3= @{

  Name   = 'Screen Saver Time Out'

  Key    = 'HKCU\Software\Policies\Microsoft\Windows\'+

              'Control Panel\Desktop'

  ValueName = 'ScreenSaveTimeOut'

  Value  = 900

  Type   = 'DWord'

}

Set-GPRegistryValue @EPHT3 | Out-Null

1. Linking both GPOs to the IT OU

$GPLHT1 = @{

  Name     = 'ITPolicy'

  Target   = 'OU=IT,DC=Reskit,DC=org'

}

New-GPLink @GPLHT1 | Out-Null

$GPLHT2 = @{

  Name     = 'Screen Saver Time Out'

  Target   = 'OU=IT,DC=Reskit,DC=org'

}

New-GPLink @GPLHT2 | Out-Null

1. Displaying the GPOs in the domain

Get-GPO -All -Domain Reskit.Org |

  Sort-Object -Property DisplayName |

    Format-Table -Property Displayname, Description, GpoStatus

1. Creating and view a GPO Report

$RPath = 'C:\Foo\GPOReport1.HTML'

Get-GPOReport -All -ReportType Html -Path $RPath

Invoke-Item -Path $RPath

1. Getting Report in XML format

Get-GPOReport -All -ReportType XML -Path $RPath2

$XML = [xml] (Get-Content -Path $RPath2)

1. Creating simple GPO report

$RPath2 = 'C:\Foo\GPOReport2.XML'

$FMTS = "{0,-33}  {1,-30} {2,-10} {3}"

$FMTS -f 'Name','Linked To', 'Enabled', 'No Override'

$FMTS -f '----','---------', '-------', '-----------'

$XML.report.GPO |

  Sort-Object -Property Name |

    ForEach-Object {

     $Gname = $\_.Name

     $SOM = $\_.linksto.SomPath

     $ENA = $\_.linksto.enabled

     $NOO = $\_.linksto.nooverride

     $FMTS -f $Gname, $SOM, $ENA, $NOO

   }

## How it works...

1. In step 1, you create a new GPO in the Reskit.Org domain. This step creates an empty GPO that is not linked to any OU.
2. In step 2, you disable user settings. This allows the GPO client to ignore any user settings that might be in this GPO thus making the client GPO processing a bit faster.
3. In step 3, you set this GPO to have two specific registry-based values. When a computer starts up, the GPO processing on that client computer ensures that these two registry values are set on the client. During Group Policy refresh (which happens approximately ever 2 hours) the value in the policy is enforced.
4. In step 4 and step 5, you create a new GPO and set a screen saver time out of 900 seconds.
5. In step 6, you link the two GPOs to the IT Organizational unit. Until you link the GPOs to an OU (or to the domain or to a domain site), GPO processing ignores the GPO. In this recipe, step 1 through step 6 produce no output

In step step 7, you use Get-GPO to return information about all the GPOs in the domain, which looks like this:

In step 8, you generate and display a GPO report by using the Get-GPOReport command. The command produces no output, but by using Invoke-Command, you view the report in your default browser, which looks like this:

In step 9, you use Get-GPOReport to return a report of all the GPOs in the domain in an XML format, which produces no output. In step 10, you iterate through the XML and produces a simple report on GPOs and where they are linked.

## There's more...

In step 9 and step 10, you create your own mini-report. In step 9, you use the Get-GPOReport command to obtain a report of all GPOs in the domain returned as XML. Then in step 10, you report on the GPOs in the Reskit Domain using .NET Composite string formatting using the -f operator.

Using .NET composite formatting enables you to use when the objects returned by a cmdlet are not in a form to be used directly with Format-Table. In step 9, for example, the XML returned contains details of the GPO links as a property which is actual an object with sub-properties. Using the tequnice, you first create a format string you use to display each row of your report. You use this format string to write two report header lines. Then for each GPO in the returned list, you obtain the GPO’s name, which OU the GPO is linked to, whether it is enabled, and whether the GPO is non-overrideable. Finally, you use the format string to output a single report line.

The Format-Table and Format-List cmdlets are of most use when each object has simple properties. When an object has a property that itself is an object with properties, the format commands do not surface these (sub) properties. In this case, you must obtain the details manually of the GPO links for each GPO and generate your report lines based on those results. Note too, the report layout works well for this specific set of GPOs - should you have GPOs with longer names, or linked to deeper Organizational units, you may need to adjust the format string you set in Step 10.

# Managing AD replication

Stuff

## Getting Ready

You run this recipe on SRV1, a workgroup server on which you have installed PowerShell 7 and VS Code.

## How to do it...

1. Step 1

## How it works...

1. Insert image B42024\_06\_01.png

## There's more...

r other .NET Language) code.

# Configuring CredSSP

Stuff

## Getting Ready

You run this recipe on SRV1, a workgroup server on which you have installed PowerShell 7 and VS Code.

## How to do it...

1. Step 1

## How it works...

1. Insert image B42024\_06\_01.png

## There's more...

r other .NET Language) code.

# Reporting on AD Computers

Stuff

## Getting Ready

You run this recipe on SRV1, a workgroup server on which you have installed PowerShell 7 and VS Code.

## How to do it...

1. Step 1

## How it works...

1. Insert image B42024\_06\_01.png

## There's more...

r other .NET Language) code.

# Reporting on AD Users

Stuff

## Getting Ready

You run this recipe on SRV1, a workgroup server on which you have installed PowerShell 7 and VS Code.

## How to do it...

1. Step 1

## How it works...

1. Insert image B42024\_06\_01.png

## There's more...

r other .NET Language) code.