Active Directory Troubleshooting

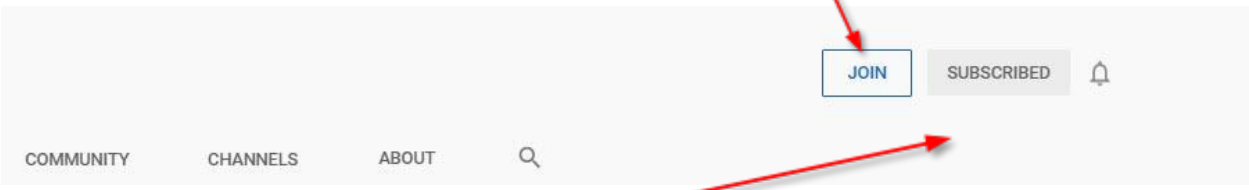
Lowell Vanderpool
TECH SAVVY PRODUCTIONS

CONTACT US:

mrvanderpool@techsavvyproductions.com

SUPPORT US:

If you would like to support the channel, Join our channel membership, it's $2.99/month (less than a Starbucks coffee); see the "Join" button on our channel homepage.



OR

Subscribe to the channel as it helps our channel perform better on YouTube's algorithm.

SOCIAL MEDIA AND WEBSITE:

Check out our **Website**: https://www.techsavvyproductions.com

Mr.V **Linkedin**: https://www.linkedin.com/in/lowell-vanderpool-57970623/

We translate subtitles on our videos into the many languages.

Tech Savvy Productions

- The 10 most common Active Directory problems faced by administrators.
- DCDiag: Quick summary of popular tests and command-line switches
- Executing DCDiag with PowerShell
- Dcdiag: Microsoft's Docs

- My Notes: on various DCdiag tests and details on what components of Active Direcotry are examined by the tests

Tech Savvy Productions

Advertising	Checks whether each domain controller advertises itself in the roles that it should be capable of performing. This test fails if the Netlogon Service has stopped or failed to start.
CheckSDRefDom	Checks that all application directory partitions have appropriate security descriptor reference domains.
CrossRefValidation	Checks the validity of cross-references.
FrsEvent	Checks to see if there are errors in the file replication system. Failing replication of the SysVol share can cause policy problems.
FrsSysVol	Checks that the file replication system and system volume are ready.
FSMOCheck	Checks that the domain controller can contact a Kerberos Key Distribution Center (KDC), a time server, a preferred time server, a primary domain controller (PDC), and a global catalog server. This test doesn't test any of the servers for operations master roles.
Intersite	Checks for failures that would prevent or temporarily hold up intersite replication and predicts how long it would take for the Knowledge Consistency Checker (KCC) to recover.
KccEvent	Checks that the KCC is completing without errors.
KnowsOfRoleHolders	Checks whether the domain controller can contact the servers that hold the five operations master roles (also known as flexible single master operations or FSMO roles).
MachineAccount	Checks whether the machine account has properly registered and that the services are advertised. Use the /RecreateMachineAccount parameter to attempt a repair if the local machine account is missing. Use the /FixMachineAccount parameter if the machine account flags are incorrect.
NCSecDesc	Checks that the security descriptors on the naming context heads have appropriate permissions for replication.
NetLogons	Checks that the appropriate logon privileges exist to allow replication to proceed.
ObjectsReplicated	Checks that the Machine Account and Directory System Agent (DSA) objects have replicated. You can use the /objectdn:dn parameter with the /n:nc parameter to specify an additional object to check.
OutboundSecureChannels	Checks that secure channels exist from all of the domain controllers in the domain to the domains that are specified by the /testdomain parameter. The /nositerestriction parameter prevents dcdiag from limiting the test to the domain controllers in the site.

Tech Savvy Productions

Replications	Checks for timely replication and any replication errors between domain controllers.
RidManager	Checks whether the relative identifier (RID) master is accessible and if it contains the proper information.
Services	Checks whether the appropriate domain controller services are running.
SystemLog	Checks that the system is running without errors.
VerifyEnterpriseReferences	Checks that specified system references are intact for the FRS and replication infrastructure across all objects in the enterprise on each domain controller.
VerifyReferences	Checks that certain system references are intact for the FRS and replication infrastructure.
VerifyReplicas	Checks that all application directory partitions are fully instantiated on all replica servers.

Tech Savvy Productions

Contents

Advertising 3

The 10 most common Active Directory problems faced by administrators. 10

Below are some of the key Event IDs to consider for monitoring Active Directory: 11

DCDiag: Quick summary of popular tests and command-line switches 13

Executing DCDiag with PowerShell 15

Dcdiag: Microsoft's Docs 16
In this article 16
Dcdiag syntax 16
Dcdiag default tests 19
DNS test syntax 21
Examples 22
Connectivity test 22
Output to a log file 26

My Notes: on various DCdiag tests and details on what components of Active Direcotry are examined by the tests 27

"Advertising" test: 27

dcdiag /test:CheckSDRefDom" 31

The CrossRefValidation test 34

The dcdiag /test:OutboundSecureChannels 45

Firewall ports that must be open between AD domain controllers: 53

VerifyReplicas 63
1. **Testing General Health:** 66

Active Directory replication troubleshooting 90

ADReplStatus **92**

https://www.youtube.com/watch?v=w2Pa2ztBwlI&t=1s DCDiag How to check Domain Controller Health 95

Permissions Analyzer for Active Directory Get instant visibility into user and group permissions 98

Use DcDiag with PowerShell to check domain controller health 99
Subscribe to 4sysops newsletter! 102

Dcdiag: How to Check Domain Controller Health 103
What is Dcdiag 104
How to install Dcdiag 104
How to use Dcdiag (examples) 105
Example 1: Use /s to run against a remote server 105
Example 2: Using /v to display more details 106
Example 3: Using /f to save to a log file 106
Example 4: Using /a to run against all domain controllers 106
Example 5: Use /q to only display the errors 106
Example 6: Use multiple switches (My favorite) 106
How to test DNS with DCDiag 107
Example: Dcdiag results from a normal domain controller 107
Example: Failed kdc service and NETLOGON Service 109

What does DCDIAG actually... do? 111

DCDIAG.EXE /E or /A or /C expected errors 127
In this article 127

Tech Savvy Productions

Symptoms 127
Cause 130
Resolution 130
More information 132

Troubleshooting RPC protocol 133

PowerShell Basics: How to Check Active Directory Replication Status 134

Troubleshooting Inconsistent Group Policy Objects (GPOs): 146

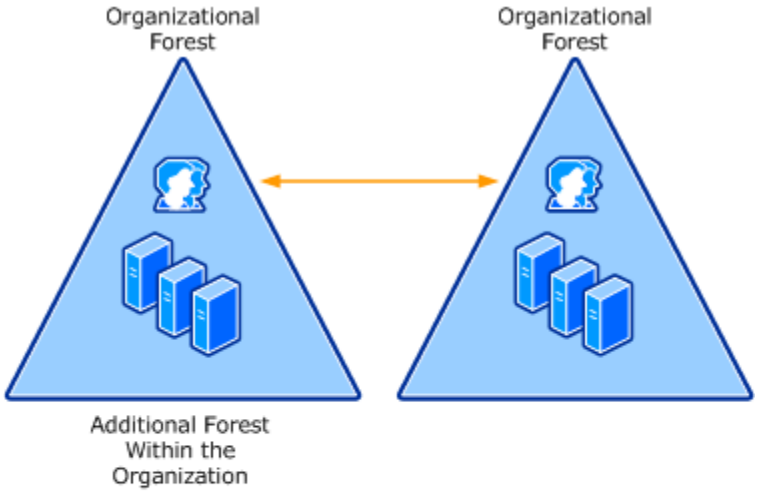
Troubleshooting AD performance issues 147

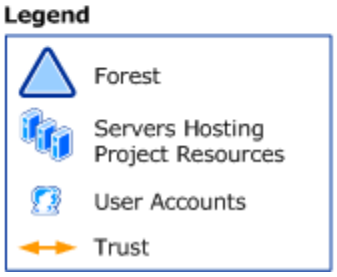
Time Syncing Issues: 149

Troubleshooting Outdated or Incorrect AD Schema 150

Inadequate Security Measures: 151

What is Active Directory? 154





157

I. Introduction to Directory Services Technologies 159

What's new in Active Directory Domain Services for Windows Server 2016 **160**

Privileged access management 160

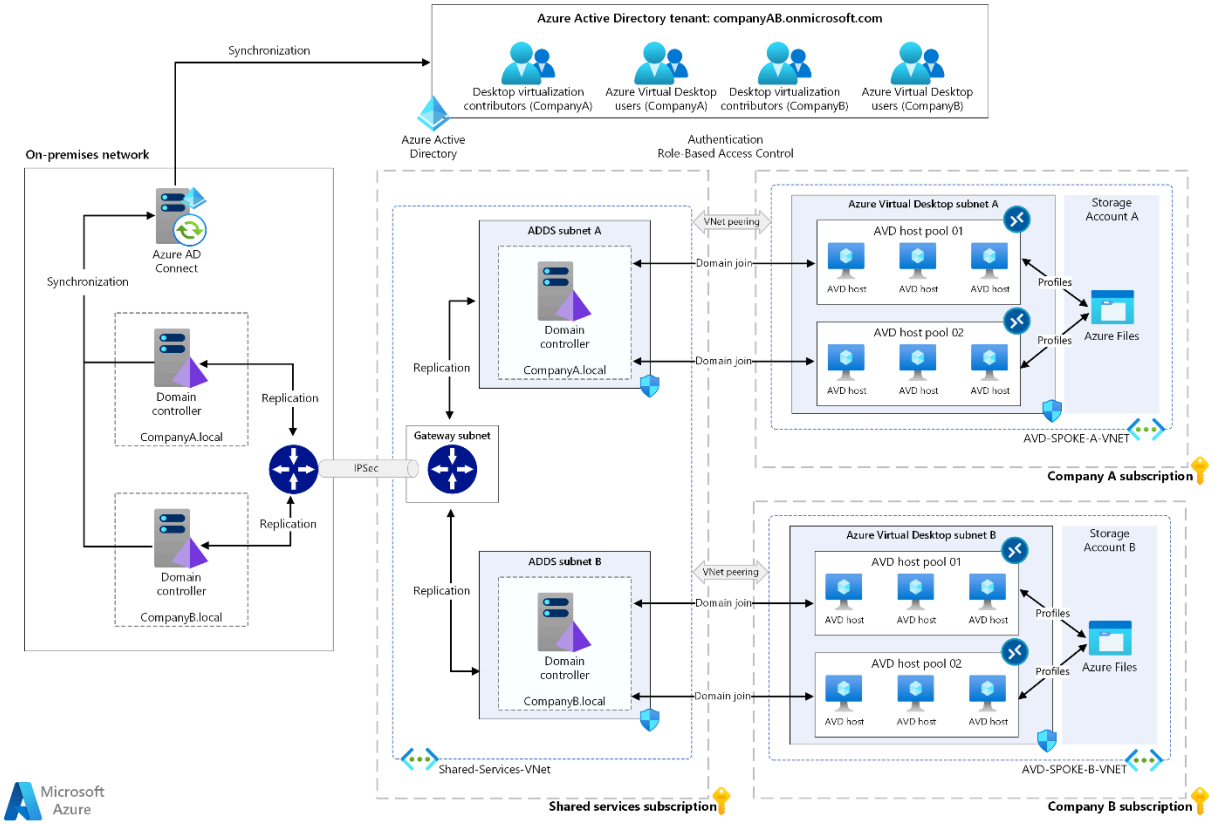
Requirements for Privileged access management 161

Azure AD Join 161

Windows Hello for Business 163

Deprecation of File Replication Service (FRS) and Windows Server 2003 functional levels 164

Tech Savvy Productions



165
In this guide AD design 165

Active Directory Federation Services **166**

1 Introduction 232
2 References 235
3 Normative References 235
4 Informative References 243
5 Overview 246
6 Relationship to Other Protocols 248
7 Prerequisites/Preconditions 248
8 Applicability Statement 248

9 Introduction 249

What is Active Directory: The Ultimate Guide 260
What is Active Directory and why is it used? 260
What is AD Domain Services? 261
Who is Active Directory for? 262
How Active Directory works 262
Understanding the structure of AD Domain Services 264
What about other Active Directory services? 264
How Active Directory helps IT 265
What are the steps in setting up Active Directory? 266
Active Directory vs LDAP or NIS 267
Windows Server Active Directory vs Azure Active Directory 268

Active Directory Glossary 269
Components and Capabilities 296

Tech Savvy Productions

10 Protocol Relationships 297
11 Protocol Summary 299
12 Environment 303

+41 43 508 3472 Fra Lightweight Directory Access Protocol 305
Contents 306
History 307
Protocol overview[edit] 308
Directory structure 308
Operations 310
Add 310
Bind (authenticate) 311
Delete 311
Search and compare 312
Modify[edit] 313
Modify DN[edit] 314
Extended operations[edit] 315
Abandon[edit] 315
Unbind[edit] 315
URI scheme[edit] 316
Schema 317
Variations 318
Other data models 319
Usage 319
See also 320
References 320
Sources 322
Further reading 322
External links 323

Azure Active Directory 324

Active Directory Replication: A Guide for IT Pros 329
Active Directory replication 330
How does Microsoft Active Directory replication work? 330
How often does Active Directory replication occur? 330
Primary replication components 331
Knowledge Consistency Checker (KCC) 331
Directory System Agent (DSA) 332
Extensible Storage Engine (ESE) 332
Remote Procedure Call (RPC) 333
Inter-Site Topology Generator (ISTG) 333
Active Directory replication objects 333
Failover functionality 333
Subnet 334
Site 334
Site link 335
Site link bridge 336
Site link transitivity 336
Global Catalog server 337
Universal group membership caching 337
Replication management 337
Check the replication health 338
Check the inbound replication requests that are queued. 338
Check the replication status 338
Synchronize replication between replication partners 339
Force the Knowledge Consistency Checker to recalculate the topology 340
How to monitor AD replication: Commands and tools 340

How Active Directory Replication Topology Works 343

Tech Savvy Productions

Troubleshooting Active Directory Replication Problems 429

Integrate on-premises AD with Azure 454
Integrate your on-premises domains with Azure AD 454
AD DS in Azure joined to an on-premises forest 455
AD DS in Azure with a separate forest 456
Extend AD FS to Azure 457

Active Directory Troubleshooting

https://activedirectoryfaq.com/

Active Directory

Active Directory Service (ADSI) is a COM-based service providing a mechanism for locating, identifying, and accessing the users and resources available in a distributed computing environment system. In particular, the Active Directory service is used to manage enterprise directory information for distribution by a Windows domain server.

---

https://www.dnsstuff.com/active-directory-best-practices

"The truth is, of course, that the curtness of the Ten Commandments is an evidence, not of the gloom and narrowness of a religion, but, on the contrary, of its liberality and humanity. It is shorter to state the things forbidden than the things permitted: precisely because most things are permitted, and only a few things are forbidden.

GK Cheterson

Tech Savvy Productions

The 10 most common Active Directory problems faced by administrators.

Active Directory (AD) is an essential service used in the business environment, which can cause significant disruption when it encounters issues. Here are ten of the most common Active Directory problems faced by administrators:

1. **Replication Errors:** Active Directory relies on replication between servers to ensure all users have access to the same network resources. If replication fails, some users might not be able to log in or access network resources.
2. **Inconsistent Group Policy Objects (GPOs):** Group Policy is a powerful feature that allows administrators to manage the working environment of users and computers in an AD environment. However, sometimes inconsistent GPOs across different organizational units (OUs) or inheritance issues can create problems, like conflicting settings or denied access.
3. **Time Syncing Issues:** All machines within an Active Directory need to be time-synced. If they aren't, Kerberos authentication can fail and users might be unable to log in.
4. **Account Lockouts:** These are a common issue and can happen for many reasons. It could be due to users entering incorrect passwords multiple times, or it might be a sign of an attempted security breach.
5. **DNS Problems:** Active Directory relies heavily on the Domain Name System (DNS). Issues with DNS, such as misconfiguration or failure, can cause problems with logging in, accessing resources, or even AD server communication.
6. **Performance Issues:** Sometimes, administrators can face problems due to poor performance of Active Directory. This can be due to a lack of server resources, outdated hardware, or even network issues.
7. **Outdated or Incorrect AD Schema:** The AD schema defines objects and their attributes. If it's incorrect or outdated, it can cause various issues, including replication failures, logon failures, and even crashes.
8. **Poorly Designed Structure:** A poorly designed Active Directory structure (poor OU design, inappropriate Group Policy application, inefficient user and group organization) can lead to security vulnerabilities, performance issues, and administrative challenges.

Tech Savvy Productions

9. **Lack of Disaster Recovery Planning:** Active Directory is a critical infrastructure component. If administrators don't have a disaster recovery plan, a server crash or other catastrophic event could cause serious business disruption.
10. **Inadequate Security Measures:** Insufficient security measures in Active Directory can lead to breaches and unauthorized access. Examples include weak password policies, excessive permissions granted to users, lack of regular security audits, and outdated systems.

Below are some of the key Event IDs to consider for monitoring Active Directory:

Directory Services Events:

- **Event ID 4742**: Occurs when a computer object is changed. This can be useful for monitoring changes to domain controllers.
- **Event ID 2887 - 2889**: These events occur when a domain controller is not accepting LDAP binds, this may indicate a problem with network connections or system resources.
- **Event ID 1988**: This event appears when lingering objects are found in a domain. It indicates a replication problem.

Replication Events:

- **Event ID 1083 and 1955**: These events are triggered when AD detects an update conflict.
- **Event ID 1864**: This event occurs when a domain controller has not replicated with its partner for a period of time.

Security Events:

- **Event ID 4624**: This is a security event that occurs when a user successfully logs on.
- **Event ID 4625**: This security event occurs when a logon attempt fails.
- **Event ID 4720**: This event is triggered when a user account is created.

Tech Savvy Productions

- **Event ID 4728**: Occurs when a member is added to a security-enabled global group.
- **Event ID 4732**: Occurs when a member is added to a security-enabled local group.
- **Event ID 4740**: Occurs when a user account is locked out.
- **Event ID 4771**: Occurs when Kerberos pre-authentication fails.
- **Event ID 4767**: A user account was unlocked.
- **Event ID 4724**: An attempt was made to reset an account's password.

Remember, depending on the size and complexity of your network, as well as your security needs, you might want to monitor different sets of event IDs. Be sure to tailor your monitoring approach to your organization's specific needs.

"Darkness cannot drive out darkness; only light can do that. Hate cannot drive out hate; only love can do that." Martin Luther King Jr.

DCDiag: Quick summary of popular tests and command-line switches

is a versatile command-line tool included with Windows Server, which can be used to diagnose and troubleshoot a variety of Active Directory issues. Here are several examples of how DCDiag can be used effectively for troubleshooting:

A few of the various tests DCDiag does on a domain controller:

1. **Connectivity tests** – DCDiag checks if domain controller is connected to the network and can communicate with other domain controllers.
2. **DNS tests** – reviews whether the domain controller resolves and registers the DNS correctly.
3. **Replication tests** – tests whether the domain controller replicates accurately with other domain controllers.
4. **Trust tests** – DCDiag checks whether the domain controller trusts other domains and whether other domains trust the domain controller.
5. **LDAP tests** – checks whether LDAP queries work properly on the domain controller.
6. **Kerberos tests** – DCDiag checks whether Kerberos authentication works correctly on the domain controller.
7. **SYSVOL tests** – DCDiag checks whether SYSVOL replication works correctly on the domain controller.
8. **Global Catalog tests** – DCDiag checks whether the DC is acting as a Global Catalog server and whether Global Catalog queries are working correctly.

"Everybody can be great... because anybody can serve. You don't have to have a college degree to serve. You don't have to make your subject and verb agree to serve. You only need a heart full of grace. A soul generated by love." Martin Luther King Jr.

Tech Savvy Productions

A summary of the popular command switches when using `dcdiag.exe`:

1. **`/s <ServerName>`** – Specifies the server's name to test. If no server name is specified, DCDiag tests the local computer.
2. **`/v`** – Provides verbose output and additional information about the tests performed.
3. **`/test:<TestName>`** – Specifies the name of the test to run. We run multiple tests by specifying multiple /test switches.
4. **`/e`** – Performs a comprehensive set of tests on the specified domain controller.
5. **`/fix`** – Attempts to fix any problems found during the tests.
6. **`/f:<LogFileName>`** – Specifies the log file's name to create.
7. **`/c`** – Specifies that DCDiag should only perform the tests required for advertising the domain controller's services.
8. **`/skip:<TestName>`** – Skips the specified test.
9. **`/l:<LogFileName>`** – Specifies the log file name to append to rather than overwriting.
10. **`/q`** – Specifies quiet mode, which suppresses the display of informational messages.
11. **`/test:DNS`** – Runs DNS tests on the specified domain controller.
12. **`/test:KCC`** – Runs KCC tests on the specified domain controller.
13. **`/test:Replications`** – Runs replication tests on the specified dc.
14. **`/test:Advertising`** – Runs advertising tests on the specified dc.
15. **`/test:Services`** – Runs service tests on the specified dc.

Why do you balk at the doctrine of the Trinity - God the three in One - yet meekly acquiesce when Einstein tells you E=mc2? Dorothy Sayers

---

Tech Savvy Productions

Executing DCDiag with PowerShell

There is no DCDiag counterpart in Windows PowerShell. However, we use DCDiag in PowerShell by running the dcdiag.exe command with the appropriate parameters using the **Invoke-Expression** or **Invoke-Command** cmdlets. Here's an example of how to use DCDiag in PowerShell to run the DNS tests:

```
Invoke-Expression "dcdiag.exe /test:DNS /v"
```

In this example, we used the **Invoke-Expression** cmdlet to run the dcdiag.exe command with the **/test:DNS** parameter to run the DNS tests and the **/v** parameter to provide verbose output. Of course, we can replace **/test:DNS** with any other DCDiag test we want to run. Alternatively, we use the **Invoke-Command** cmdlet to run DCDiag on a remote computer.

Here's another example:

```
Invoke-Command -ComputerName "ServerName" -ScriptBlock
{ dcdiag.exe /test:DNS /v }
```

In the previous example, we used the **-ScriptBlock** parameter with **Invoke-Command** to specify the code we wanted to run on the remote computer. The **-ScriptBlock** parameter takes a script block, essentially a block of code enclosed in braces **{}**.

When we use **Invoke-Command** with the **-ScriptBlock** parameter, we execute the script block on the remote computer specified with the **-ComputerName** parameter, and the script block returns the output to the local computer. Using the **-ScriptBlock** parameter with **Invoke-Command** is a powerful way to execute commands or scripts on remote computers. It helps manage large numbers of computers in a network environment, just like how we used the DCDiag tool.

"Even idiots occasionally speak the truth accidentally."

— Dorothy L. Sayers

Tech Savvy Productions

Dcdiag: Microsoft's Docs

In this article

1. Dcdiag syntax
2. Dcdiag default tests
3. DNS test syntax
4. Examples
5. Additional references

Applies To: Windows Server 2022, Windows Server 2019, Windows Server 2016

Dcdiag.exe analyzes the state of domain controllers in a forest or enterprise and reports any problems to help in troubleshooting.

As an end-user reporting program, dcdiag is a command-line tool that encapsulates detailed knowledge of how to identify abnormal behavior in the system. Dcdiag displays command output at the command prompt.

Dcdiag consists of a framework for executing tests and a series of tests to verify different functional areas of the system. This framework selects which domain controllers are tested according to scope directives from the user, such as enterprise, site, or single server.

To use dcdiag, you must run dcdiag from an elevated command prompt or PowerShell with administrative rights.

Dcdiag syntax

dcdiag [/s:<DomainController>] [/n:<NamingContext>]
[/u:<Domain>\<UserName> /p:{* | <Password> | ""}] [{/a | /e}] [{/q | /v}] [/i]
[/f:<LogFile>] [/c [/skip:<Test>]] [/test:<Test>] [/fix] [{/h | /?}]
[/ReplSource:<SourceDomainController>]

Dcdiag uses the following parameters:

Parameter	Description
/s:<DomainController>	Specifies the name of the server to run the command against. If this parameter

Tech Savvy Productions

Parameter	Description
	isn't specified, the tests are run against the local domain controller. This parameter is ignored for DcPromo and RegisterInDns tests, which can only be run locally.
/n:<NamingContext>	Uses NamingContext as the naming context to test. You can specify domains in NetBIOS, Domain Name System (DNS), or distinguished name format.
/u:<Domain>\<UserName> /p:{<Password> | ""}	Uses Domain\UserName. Dcdiag uses the current credentials of the user (or process) that is logged on. If alternate credentials are needed, use the following options to provide those credentials for binding with Password as the password: Use quotation marks ("") for an empty or null password. Use the wildcard character (*) to prompt for the password.
/a	Tests all the servers on this AD DS site.
/e	Tests all the servers in the enterprise. This overrides /a.
/q	Quiet. Prints only error messages.
/v	Verbose. Prints extended information.
/fix	Affects the MachineAccount test only. This parameter causes the test to fix the Service Principal Names (SPNs) on the Machine Account object of the domain controller.
/f:<LogFile>	Redirects all output to a log file.
/c	Comprehensive. Runs all tests except DCPromo and RegisterInDNS, including non-default tests. Optionally, you can use this parameter with the /skip

Tech Savvy Productions

Parameter	Description
	parameter to skip specified tests. The following tests aren't run by default: ● Topology ● CutoffServers ● OutboundSecureChannels.
/h or /?	Displays help at the command prompt.
/test:<Test>	Runs this test only. The connectivity test, which you can't skip, is also run. This can't be used with /skip parameter.
/ReplSource:<SourceDomainController>	Tests the connection between the domain controller on which you run the command and the source domain controller. (This parameter is used for the CheckSecurityError test.) SourceDomainController is the DNS name, NetBIOS name, or distinguished name of a real or potential server that will be the source domain controller for replication, as represented by a real or potential connection object.

Dcdiag default tests

The following table shows tests that run by default.

Tech Savvy Productions

Test	Description
Advertising	Checks whether each domain controller advertises itself in the roles that it should be capable of performing. This test fails if the Netlogon Service has stopped or failed to start.
CheckSDRefDom	Checks that all application directory partitions have appropriate security descriptor reference domains.
CrossRefValidation	Checks the validity of cross-references.
FrsEvent	Checks to see if there are errors in the file replication system. Failing replication of the SysVol share can cause policy problems.
FrsSysVol	Checks that the file replication system and system volume are ready.
FSMOCheck	Checks that the domain controller can contact a Kerberos Key Distribution Center (KDC), a time server, a preferred time server, a primary domain controller (PDC), and a global catalog server. This test doesn't test any of the servers for operations master roles.
Intersite	Checks for failures that would prevent or temporarily hold up intersite replication and predicts how long it would take for the Knowledge Consistency Checker (KCC) to recover.
KccEvent	Checks that the KCC is completing without errors.
KnowsOfRoleHolders	Checks whether the domain controller can contact the servers that hold the five operations master roles (also known as flexible single master operations or FSMO roles).
MachineAccount	Checks whether the machine account has properly registered and that the services are advertised. Use the /RecreateMachineAccount parameter to attempt a repair if the local machine account is missing. Use the /FixMachineAccount parameter if the machine account flags are incorrect.
NCSecDesc	Checks that the security descriptors on the naming context heads have appropriate permissions for replication.
NetLogons	Checks that the appropriate logon privileges exist to allow replication to proceed.
ObjectsReplicated	Checks that the Machine Account and Directory System Agent (DSA) objects have replicated. You can use the /objectdn:dn parameter with the /n:nc parameter to specify an additional object to check.

Tech Savvy Productions

Test	Description
OutboundSecureChannels	Checks that secure channels exist from all of the domain controllers in the domain to the domains that are specified by the /testdomain parameter. The /nositerestriction parameter prevents dcdiag from limiting the test to the domain controllers in the site.
Replications	Checks for timely replication and any replication errors between domain controllers.
RidManager	Checks whether the relative identifier (RID) master is accessible and if it contains the proper information.
Services	Checks whether the appropriate domain controller services are running.
SystemLog	Checks that the system is running without errors.
VerifyEnterpriseReferences	Checks that specified system references are intact for the FRS and replication infrastructure across all objects in the enterprise on each domain controller.
VerifyReferences	Checks that certain system references are intact for the FRS and replication infrastructure.
VerifyReplicas	Checks that all application directory partitions are fully instantiated on all replica servers.

Note

Connectivity checks on domain controllers that are registered on the network and connect with other resources like DNS, LDAP, and RPC **cannot** be skipped.

DNS test syntax

dcdiag /test:DNS [/DnsBasic | /DnsForwarders | /DnsDelegation | /DnsDynamicUpdate | /DnsRecordRegistration | /DnsResolveExtName [/DnsInternetName:<InternetName>] | /DnsAll] [/f:<LogFile>] [/x:<XMLLog.xml>] [/xsl:<XSLFile.xsl> or <XSLTFile.xslt>] [/s:<DomainController>] [/e] [/v]

Tech Savvy Productions

DNS test uses the following parameters:

Parameter	Description
/test:DNS	Performs the specified DNS test. If no test is specified, defaults to /DnsAll.
/DnsBasic	Performs basic DNS tests, including network connectivity, DNS client configuration, service availability, and zone existence.
/DnsForwarders	Performs the /DnsBasic tests and also checks the configuration of forwarders.
/DnsDelegation	Performs the /DnsBasic tests and also checks for proper delegations.
/DnsDynamicUpdate	Performs /DnsBasic tests and also determines if dynamic update is enabled in the Active Directory zone.
/DnsRecordRegistration	Performs the /DnsBasic tests, and also checks if the address (A), canonical name (CNAME) and well-known service (SRV) resource records are registered. In addition, creates an inventory report based on the test results.
/DnsResolveExtName [/DnsInternetName:\<InternetName>]	Performs the /DnsBasic tests, and also attempts to resolve InternetName. If /DnsInternetName isn't specified, attempts to resolve the name www.microsoft.com. If /DnsInternetName is specified, attempts to resolve the Internet name supplied by the user.
/DnsAll	Performs all tests, except for the /DnsResolveExtName test, and generates a report.
/f:<LogFile>	Redirects all output to a log file.
/s:<DomainController>	Runs the tests against the domain controller. If this parameter isn't specified, the tests are run against the local domain controller.
/e	Runs all tests specified by /test:DNS against all domain controllers in the Active Directory forest.

Tech Savvy Productions

Parameter	Description
/v	Verbose. Presents extended information about successful test results in addition to information about errors and warnings. When the /v parameter isn't used, provides only error and warning information. Use the /v switch when errors or warnings are reported in the summary table.
/x:<XMLLog.xml>	Redirects all output to xmllog.xml. This parameter works only with the /test:DNS option.
/xsl:<XSLFile.xsl> or /xsl:<XSLTFile.xslt>	Adds the processing instructions that reference the specified sheet. This parameter only works with the /test:DNS /x:<XMLLog.xml option.

Note

Run times for DNS tests can be significant in large enterprises when the **/e** parameter is used. Domain controllers and DNS servers that are offline will increase run times due to long time-out periods for RPC and other protocols.

Examples

Connectivity test

To run a series of connectivity tests on the local domain, run the following command:

PowerShell
dcdiag

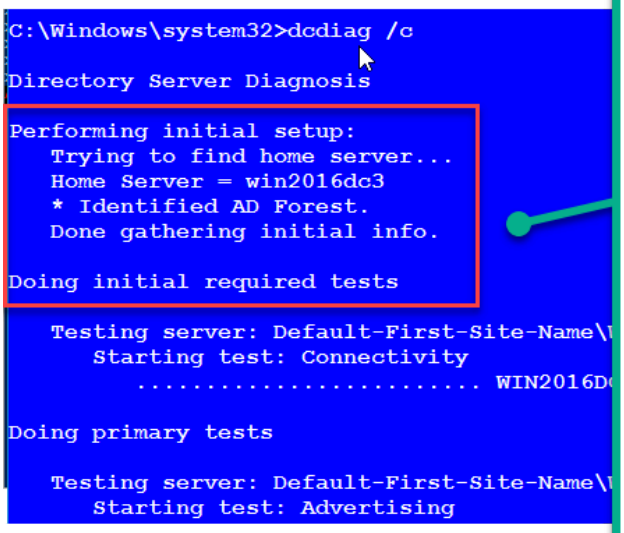
A successful connectivity test output can be seen below:

PowerShell
PS C:\Users\Administrator> dcdiag

```
Directory Server Diagnosis
Performing initial setup:
   Trying to find home server...
   Home Server = MapleWaffle-WS22
```


Tech Savvy Productions

* Identified AD Forest.
Done gathering initial info.
Doing initial required tests
Testing server: Default-First-Site-Name\MAPLEWAFFLE-WS2



- Verifying the DC can be located in DNS <https://technet.microsoft.com/en-us/library/cc772774(v=WS.10).aspx>.
- Verifying the DC responds to ICMP <https://support.microsoft.com/kb/170292> pings.
- Verifying the DC allows LDAP <https://technet.microsoft.com/en-us/library/cc755809(WS.10).aspx> connectivity by binding to the instance.
- Verifying the DC allows binding to the AD RPC interface <https://msdn.microsoft.com/en-us/library/ms675879(v=VS.85).aspx> using the DsBindWithCred <https://msdn.microsoft.com/en-us/library/ms675961(VS.85).aspx> function.

The DNS test can be satisfied out of the client cache so restarting the DNS client service locally is advisable when running DCDIAG to guarantee a full test of name resolution. For example:

Net stop "dns client" & net start "dns client" & dcdiag /test:verifyreplicas /s:DC-01

The initial tests cannot be skipped.

The initial tests use ICMP <https://support.microsoft.com/kb/170292>, LDAP <https://technet.microsoft.com/en-us/library/cc755809(WS.10).aspx>, DNS <https://technet.microsoft.com/en-us/library/cc772774(v=WS.10).aspx>, and RPC <https://technet.microsoft.com/en-us/library/cc738291(WS.10).aspx> on the network.

Editorial note: Blocking ICMP will prevent DCDIAG from working.

Starting test: Connectivity
..... MAPLEWAFFLE-WS2 passed test Connectivity
Doing primary tests
Testing server: Default-First-Site-Name\MAPLEWAFFLE-WS2

Starting test: Advertising
..... MAPLEWAFFLE-WS2 passed test Advertising
Starting test: FrsEvent
..... MAPLEWAFFLE-WS2 passed test FrsEvent

Tech Savvy Productions

Starting test: DFSREvent
..... MAPLEWAFFLE-WS2 passed test DFSREvent

```
   Starting test: SysVolCheck
      ..... MAPLEWAFFLE-WS2 passed test SysVolCheck

   Starting test: KccEvent
      ..... MAPLEWAFFLE-WS2 passed test KccEvent

   Starting test: KnowsOfRoleHolders
      ..... MAPLEWAFFLE-WS2 passed test KnowsOfRoleHolders

   Starting test: MachineAccount
      ..... MAPLEWAFFLE-WS2 passed test MachineAccount

   Starting test: NCSecDesc
      ..... MAPLEWAFFLE-WS2 passed test NCSecDesc

   Starting test: NetLogons
      ..... MAPLEWAFFLE-WS2 passed test NetLogons

   Starting test: ObjectsReplicated
      ..... MAPLEWAFFLE-WS2 passed test ObjectsReplicated

   Starting test: Replications
      ..... MAPLEWAFFLE-WS2 passed test Replications

   Starting test: RidManager
      ..... MAPLEWAFFLE-WS2 passed test RidManager

   Starting test: Services
      ..... MAPLEWAFFLE-WS2 passed test Services

   Starting test: SystemLog
      ..... MAPLEWAFFLE-WS2 passed test SystemLog

   Starting test: VerifyReferences
      ..... MAPLEWAFFLE-WS2 passed test VerifyReferences

Running partition tests on : ForestDnsZones

   Starting test: CheckSDRefDom
      ..... ForestDnsZones passed test CheckSDRefDom

   Starting test: CrossRefValidation
      ..... ForestDnsZones passed test CrossRefValidation

Running partition tests on : DomainDnsZones

   Starting test: CheckSDRefDom
      ..... DomainDnsZones passed test CheckSDRefDom

   Starting test: CrossRefValidation
      ..... DomainDnsZones passed test CrossRefValidation

Running partition tests on : Schema

   Starting test: CheckSDRefDom
      ..... Schema passed test CheckSDRefDom

   Starting test: CrossRefValidation
      ..... Schema passed test CrossRefValidation
```


Tech Savvy Productions

```
Running partition tests on : Configuration

   Starting test: CheckSDRefDom
      ..................... Configuration passed test
      CheckSDRefDom

   Starting test: CrossRefValidation
      ..................... Configuration passed test
      CrossRefValidation

Running partition tests on : corp

   Starting test: CheckSDRefDom
      ..................... corp passed test CheckSDRefDom

   Starting test: CrossRefValidation
      ..................... corp passed test CrossRefValidation

Running enterprise tests on : corp.baconpancake.com

   Starting test: LocatorCheck
      ..................... corp.baconpancake.com passed test
      LocatorCheck

   Starting test: Intersite
      ..................... corp.baconpancake.com passed test
      Intersite
```

PS C:\Users\Administrator>

To run a series of connectivity tests on a specific domain controller, run the following command:

PowerShell
```
dcdiag /s:<DomainControllerName>
```

It should generate similar results as the local test if no issues are encountered.

Output to a log file

Dcdiag is able to save the output results to a text file by running the following:

PowerShell
```
dcdiag /s:<DomainControllerName> /f:<FileName.txt>
```


Tech Savvy Productions

If no <FilePath> is specified, results are saved to `C:\Users\<UserName>\<FileName.txt>` by default.

To save to a specific location, run the following:

PowerShell

```
dcdiag /s:<DomainControllerName>
/f:<DriveLetter>\<FilePath>\<FileName.txt>
```

My Notes: on various DCdiag tests and details on what components of Active Direcotry are examined by the tests

"Advertising" test:

is one of the many checks performed by the DCDiag utility, which is used to analyze the health and functionality of Domain Controllers within an Active Directory forest. This test is designed to ensure that each Domain Controller is properly advertising itself.

Here is a more detailed explanation of what it checks:

Tech Savvy Productions

1. **Advertising Status:** The test checks to ensure that the Domain Controller is currently advertising its availability and services to clients in the network. This includes authenticating users, replicating directory updates, and other domain services.
2. **Netlogon Service:** Advertising heavily relies on the Netlogon service on each Domain Controller, which is responsible for creating the necessary Service (SRV) records in DNS. The test checks the status of this service to make sure it's running as expected.
3. **DNS Service Records:** The test checks the existence of appropriate service records in DNS. These records are essential for clients to locate the Domain Controller and utilize its services. The records include entries like LDAP, Kerberos, Global Catalog, etc.
4. **Event Log Errors:** The Advertising test also examines the System Event Log on the Domain Controller for recent Netlogon or Advertising-related errors. These could be indications of potential problems in service advertising.

If the Advertising test fails, it may indicate issues like incorrect DNS configuration, Netlogon service failures, or potential connectivity problems. As a result, clients might not be able to locate the Domain Controller or utilize its services, which could impact network operations and user authentications.

PowerShell commands and scripts:

- `Get-EventLog`: Use the Get-EventLog cmdlet to retrieve event logs. You can specify the event log source and filter the events based on event IDs related to the advertising test failures. For example:
    - `Get-EventLog -LogName Application -Source "Directory Service" | Where-Object {$_.EventID -eq <EventID>}`

Tech Savvy Productions

- `Get-WinEvent`: Use the Get-WinEvent cmdlet to retrieve events from the Windows Event Log. Similar to Get-EventLog, you can filter events based on specific event IDs and other criteria.
- `Test-ComputerSecureChannel`: This PowerShell cmdlet can help diagnose issues with the secure channel between the domain controller and client computers. It verifies the machine account password and resets the secure channel if necessary.

3. Command Prompt utilities:
   - **EventCombMT:** This utility allows you to search event logs across multiple machines simultaneously. You can specify the event IDs and the target domain controllers to gather relevant event log information. This utility must be downloaded.

Thank you for downloading **Account Lockout and Management Tools**
If your download does not start after 30 seconds,
click here to download manually
ALTools.exe includes:

AcctInfo.dll. Helps isolate and troubleshoot account lockouts and to change a user's password on a domain controller in that user's site. It works by adding new property pages to user objects in the Active Directory Users and Computers Microsoft Management Console (MMC).

ALockout.dll. On the client computer, helps determine a process or application that is sending wrong credentials.

Caution: Do not use this tool on servers that host network applications or services. Also, you should not use ALockout.dll on Exchange servers, because it may prevent the Exchange store from starting.

ALoInfo.exe. Displays all user account names and the age of their passwords.

EnableKerbLog.vbs. Used as a startup script, allows Kerberos to log on to all your clients that run Windows 2000 and later.

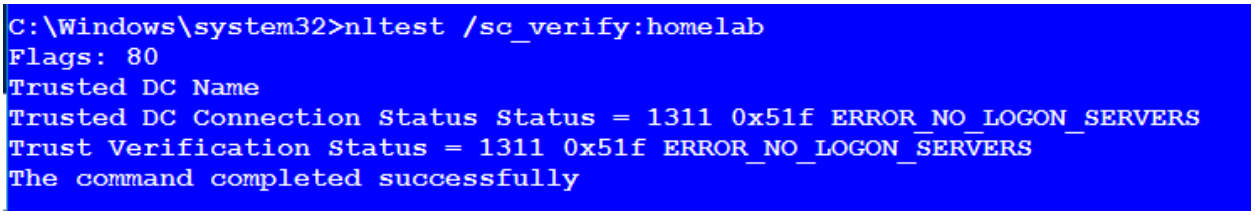
EventCombMT.exe. Gathers specific events from event logs of several different machines to one central location.

LockoutStatus.exe. Determines all the domain controllers that are involved in a lockout of a user in order to assist in gathering the logs. LockoutStatus.exe uses the NLParse.exe tool to parse Netlogon logs for specific Netlogon return status codes. It directs the output to a comma-separated value (.csv) file that you can sort further, if needed.

Tech Savvy Productions

NLParse.exe. Used to extract and display desired entries from the Netlogon log files.

- **Repadmin:** This command-line tool is used for Active Directory replication diagnostics and monitoring. Running "`repadmin /showrepl`" can help identify replication issues between domain controllers.
- **nltest:** The nltest command can be used to check the trust relationship between the domain controller and the client computers. Running "`nltest /sc_verify:<domain_name>`" can verify the secure channel between the domain controller and the client.



Remember to run these PowerShell scripts and CMD utilities with appropriate administrative privileges on the affected domain controller or management workstation.

Tech Savvy Productions

dcdiag /test:CheckSDRefDom" command in dcdiag (Domain Controller Diagnostic) is used to check whether all application directory partitions in Active Directory (AD) have appropriate security descriptor reference domains. To understand this, we need to break down the components:

1. Application Directory Partitions:
    - In AD, directory partitions are logical divisions that store specific types of data. The default partitions include the domain partition, schema partition, and configuration partition.
    - Application directory partitions are additional partitions created to store application-specific data. They allow applications to have their own separate data within the AD infrastructure.
2. Security Descriptor Reference (SDRef):
    - In AD, every object (e.g., user, group, computer) has a security descriptor associated with it. The security descriptor defines the permissions and access control settings for that object.
    - The security descriptor references a security principal (user or group) that is granted specific permissions on the object.

Tech Savvy Productions

3. Reference Domains:
    - A reference domain, in this context, refers to the domain or domains from which security principals are referenced within the security descriptors.
    - It ensures that the security descriptors properly reference security principals from the correct domain(s).

The purpose of the "dcdiag /test:CheckSDRefDom" command is to verify that the application directory partitions within the AD infrastructure have appropriate reference domains in their security descriptors. It ensures that the security descriptors accurately reference the security principals from the correct domains.

If any issues are found during the test, it typically indicates that the security descriptors within the application directory partitions are not correctly referencing the appropriate security principals or reference domains. This may lead to access or permission issues when interacting with the application-specific data stored in those partitions.

1. **Event ID 2886:** This event indicates that the Directory Service has detected a client-side or server-side error related to the replication process. It often points to issues with replication, such as DNS misconfiguration or problems with the secure channel. Check the event description for more details.
2. **Event ID 2087**: This event suggests that the Active Directory Domain Services could not resolve the replication partner's distinguished name (DN) due to a DNS resolution failure. It typically occurs when the domain controller is unable to locate its replication partner.
3. **Event ID 2088:** This event signifies that the Active Directory Domain Services could not resolve the replication partner's domain name because the DNS name is not registered or the DNS server is not reachable. It points to DNS-related issues between the domain controllers.
4. **Event ID 2089:** This event indicates that a replication partner was detected, but the replication topology could not be determined due to DNS lookup failures. It often occurs when the domain controller is unable to locate replication partners or when there are issues with DNS resolution.

Tech Savvy Productions

5. **Event ID 2042:** This event suggests that the Knowledge Consistency Checker (KCC) encountered a problem while attempting to configure the replication topology. It could point to issues with the replication topology generation process or problems with DNS.
6. **Event ID 13508:** This event is logged when the File Replication Service (FRS) or the Distributed File System Replication (DFSR) encounters problems during the replication process. It indicates issues with file replication and can be related to the CheckSDRefDom test failure.

- `Get-EventLog`: Use this cmdlet to retrieve event logs related to Active Directory and replication. You can filter the logs based on event IDs associated with the failures encountered during the "dcdiag /test:CheckSDRefDom" command. For example:
- `Get-EventLog -LogName DirectoryService | Where-Object {$_.EventID -eq <EventID>}`
- Get-WinEvent: Similar to Get-EventLog, the Get-WinEvent cmdlet allows you to retrieve events from the Windows Event Log. You can filter events based on specific event IDs and other criteria. For example:
    - powershell
- `Get-WinEvent -LogName DirectoryService | Where-Object {$_.Id -eq <EventID>}`
- Test-ComputerSecureChannel: This cmdlet can help diagnose problems with the secure channel between the domain controller and client computers. It verifies the machine account password and resets the secure channel if necessary. Run the following command on the affected domain controller:
- `Test-ComputerSecureChannel -Repair`
- `Get-ADReplicationFailure`: Use this cmdlet to identify replication failures between domain controllers. It retrieves information about the replication status and any errors or failures. Run the following command on a domain controller:
- `Get-ADReplicationFailure`
- Get-ADDomainController -Filter: You can use this cmdlet to retrieve information about the domain controllers in your environment. Filter the results based on specific criteria, such as the domain controller's name or site. For example:

Tech Savvy Productions

- `Get-ADDomainController -Filter {Name -like "*<DCName>*"}`
- `Get-ADDomainControllerPasswordReplicationPolicy`: This cmdlet retrieves the password replication policy for a specified domain controller. It can help identify if there are any issues with the password replication process. Run the following command on a domain controller:

  ```
  Get-ADDomainControllerPasswordReplicationPolicy
  -Identity <DCName>
  ```

The CrossRefValidation test specifically checks the consistency and configuration of the "crossRef" objects in the Partitions container in the Configuration container in Active Directory.

CrossRef objects represent all the naming contexts (NCs), including domain NCs, schema NC, configuration NC, and application NCs, held by each domain controller in an Active Directory forest. Each naming context has a corresponding crossRef object.

Here is a more detailed explanation of what it checks:

1. **Existence:** It ensures that every domain and application partition in the forest has a corresponding crossRef object.

Tech Savvy Productions

2. **Configuration:** It checks the attributes of each crossRef object for correctness, specifically the systemFlags attribute, dnsRoot attribute, and the nCName attribute:
    - systemFlags is checked to ensure the value is correct in relation to the type of partition the crossRef object represents (whether it is a domain, application partition, etc.).
    - dnsRoot attribute is checked to ensure that it matches the DNS name of the domain or application partition that the crossRef object represents.
    - nCName attribute is checked to ensure that it correctly refers to the distinguished name (DN) of the naming context.
3. **Connectivity:** It also attempts to bind (establish a connection) to the Naming Context (NC) that the crossRef object represents to ensure the NC is reachable and operational.

If there are any inconsistencies, missing crossRef objects, or incorrect attribute values, then the CrossRefValidation test will fail. This could indicate replication issues, schema problems, DNS misconfigurations, or other serious Active Directory issues.

In Active Directory (AD), the configuration partition is one of the five partitions that make up the directory database. It stores configuration information about the structure and settings of the AD forest. The configuration partition is replicated to all domain controllers within the forest to ensure consistency across the entire forest.

Here are the key aspects and details of the configuration partition in AD:

1. **Purpose:** The configuration partition contains information about the forest-wide settings and objects that define the overall structure and

Tech Savvy Productions

operation of the AD forest. It includes details about domains, sites, trust relationships, schema, and other forest-wide configuration settings.

2. **Replication Scope:** The configuration partition is replicated to all domain controllers within the forest. This ensures that all domain controllers have consistent information about the forest-wide configuration and can apply the appropriate policies and settings.
3. **Forest-Wide Configuration Settings:** The configuration partition stores various forest-wide configuration settings, including:
    - Schema: The schema defines the structure and attributes of objects within the forest. It includes definitions for common objects such as users, groups, computers, and organizational units (OUs). Any changes to the schema are stored in the configuration partition and replicated to all domain controllers.
    - **Sites and Subnets:** Information about the physical network topology of the forest, such as sites and subnets, is stored in the configuration partition. This information helps in the efficient replication of data and the optimization of domain controller and client interactions based on geographic or network boundaries.
    - **Trust Relationships:** Details about trust relationships between different domains within the forest or with external domains or forests are stored in the configuration partition. This includes information about transitive and non-transitive trusts, authentication settings, and security policies related to trust relationships.
    - **Domain Controllers:** The configuration partition maintains information about the domain controllers in the forest, including their names, roles, site assignments, and replication metadata. This information is essential for the proper functioning of the replication process and the availability of domain services.
4. **Replication Process:** The configuration partition follows the same replication mechanisms as other AD partitions. Domain controllers use the Directory Replication Service (DRS) to replicate the configuration data between each other. Replication occurs through a multi-master model, where updates made on one domain controller are propagated to others using a replication topology defined within the forest.
5. **Management and Access:** The configuration partition is typically managed using tools such as Active Directory Sites and Services, Active Directory Domains and Trusts, and the ADSI Edit snap-in. These tools allow

Tech Savvy Productions

administrators to view and modify forest-wide configuration settings stored in the configuration partition.

The configuration partition in Active Directory stores essential information about the forest-wide settings, schema, sites, trust relationships, and other configuration details. It ensures consistent configuration and operation across the entire forest by replicating this information to all domain controllers within the forest. Proper management and understanding of the configuration partition are crucial for maintaining a well-functioning and properly configured AD environment.

The **"dcdiag /test:frssysvol"** command is used to check the status of the File Replication Service (FRS) for the SYSVOL replication on domain controllers in an Active Directory domain. SYSVOL is a shared folder that contains important files such as group policy settings and logon scripts, which need to be consistent across all domain controllers for proper operation of the domain.

When you run the "dcdiag /test:frssysvol" command, it performs a series of checks to ensure that the FRS is functioning correctly and that the SYSVOL replication is working as expected. The results of this command provide information about the replication status and any potential issues that may exist within the SYSVOL replication process.

Here are some possible results you might see:

1. "Starting test: frssysvol" This message indicates that the test is being initiated.
2. "..... DC1 passed test frssysvol" This message indicates that the domain controller named "DC1" has passed the frssysvol test. A similar

Tech Savvy Productions

message will appear for each domain controller in the domain that successfully passes the test.

3. "..... DC1 failed test frssysvol" This message indicates that the domain controller named "DC1" has failed the frssysvol test. A similar message will appear for each domain controller in the domain that fails the test.
4. "Test omitted by user request: frssysvol" This message indicates that the frssysvol test was skipped because it was not included in the command or configuration.

If a domain controller fails the frssysvol test, it suggests that there may be issues with the FRS or SYSVOL replication, which can lead to inconsistencies in group policy settings and logon scripts across the domain. It's important to investigate and resolve any failures to ensure proper functioning of the domain.

To troubleshoot frssysvol test failures, you can review the associated error messages or event logs on the affected domain controller. Common issues may include network connectivity problems, permission issues, or problems with the FRS service itself. Taking corrective actions based on the specific error messages can help resolve the SYSVOL replication issues and maintain a consistent and functional Active Directory environment

FRS or DFRS?

By default, Active Directory (AD) uses FRS (File Replication Service) for the replication of the SYSVOL (System Volume) share. FRS has been the default replication mechanism for SYSVOL since the introduction of Windows 2000 Server.

However, starting with Windows Server 2008, Microsoft introduced a new replication technology called **DFSR** (Distributed File System Replication). DFSR provides enhanced capabilities and improved performance compared to FRS. With Windows Server 2008 and later versions, including Windows Server 2016 and Windows Server 2019, DFSR became the recommended and default replication method for SYSVOL.

The SYSVOL share plays a crucial role in Active Directory (AD) as it is responsible for storing and replicating important system files, such as Group Policy Objects

Tech Savvy Productions

(GPOs), logon scripts, and other shared resources, across all domain controllers within a domain.

Here are the key roles and functions of the SYSVOL share:

1. **Storage of Group Policy Objects (GPOs):** The SYSVOL share is the central location where GPOs are stored. GPOs contain policies and settings that govern the behavior and configuration of user accounts and computer systems within an AD domain. When GPOs are created or modified, they are stored in the SYSVOL share to ensure consistent replication across all domain controllers.
2. **Replication of GPOs:** SYSVOL utilizes the File Replication Service (FRS) or, in more recent versions of Windows Server, the Distributed File System Replication (DFSR) to replicate the GPOs across all domain controllers in a domain. This replication process ensures that GPO changes made on one domain controller are propagated to all other domain controllers, ensuring consistent Group Policy application throughout the domain.
3. **Logon Scripts:** The SYSVOL share is also used to store logon scripts, which are executed when users log on to a domain-joined computer. Logon scripts can contain instructions or commands that are executed automatically during the user logon process, allowing administrators to automate tasks or configure specific settings for users.
4. **Shared System Files:** In addition to GPOs and logon scripts, the SYSVOL share may also store other shared system files, such as system policies, startup scripts, or administrative templates. These files are typically accessed and utilized by domain-joined computers and users as part of the Active Directory functionality.

By utilizing the SYSVOL share and ensuring its proper replication across all domain controllers, Active Directory can provide consistent configuration, policy enforcement, and resource access for users and computers within the domain. It plays a crucial role in maintaining the integrity and functionality of the AD environment.

To repair a local machine account, you can use the "netdom" command as follows:

Tech Savvy Productions

```
netdom resetpwd /server:yourdcname
/userd:domainname\administrator
/passwordd:administratorpassword
```
Here, you replace "yourdcname" with the name of your domain controller, "domainname" with your domain, "administrator" with an account that has the necessary privileges, and "administratorpassword" with the password for that account.

Note: Using these commands can disrupt services and should only be used when necessary and during a maintenance window. Always ensure you have a current backup of important data. If you're unsure, consider seeking assistance from a professional.

The **dcdiag /test:NCSecDesc** command is a built-in tool for testing an aspect of Microsoft's Active Directory (AD). Dcdiag stands for Directory Service Diagnostics, and it's used to check the status and health of domain controllers.

The /test:NCSecDesc test specifically checks that the **security descriptors on the naming context heads** have **appropriate permissions for replication**. The Naming Context (NC) is essentially a partition in the directory's database.

If the permissions are not set correctly, the replication between domain controllers might fail. A replication error could lead to inconsistencies in the Active Directory database across domain controllers, which could cause various types of issues, including logon failures, inconsistencies in group policy object application, and other unpredictable behavior.

Some common reasons for failure could include:

1. **Incorrect Permissions:** The most likely cause of an error with the NCSecDesc test is incorrect permissions. The test will fail if the Enterprise Domain Controllers group does not have the "Replicating Directory Changes" permission.
2. **Network Issues:** If there are problems with the network, such as firewalls blocking necessary traffic, or DNS misconfigurations, then replication might fail.

Tech Savvy Productions

3. **Schema Mismatch:** This error might occur if there's a schema mismatch between domain controllers. This could happen after an unsuccessful or partially successful schema update.
4. **Replication Latency:** This could occur if changes made on one domain controller take too long to replicate to other domain controllers in the network.
5. **Active Directory Database Corruption:** Though rare, physical corruption of the Active Directory database could lead to a failure of this test.

To resolve these issues, an administrator should review the error message, identify the cause of the problem, and take appropriate action, such as fixing permissions, checking network settings, or potentially repairing the AD database.

1. **Event ID 1311 (NTDS KCC)**: This event is logged when the Knowledge Consistency Checker (KCC) detects problems with the partitions in Active Directory. It could signal issues with the naming context.
2. **Event ID 1084 (NTDS Replication)**: This event is logged when the Active Directory Domain Services could not update the following object with changes received from the directory service at the following network address because Active Directory Domain Services was busy processing information.
3. **Event ID 1694 (NTDS Replication)**: This event indicates that the Active Directory Domain Services replicated a security descriptor that was larger than the maximum size.
4. **Event ID 1925 (NTDS KCC)**: This event signifies that the attempt to establish a replication link for the specified partition with the specified domain controller failed.
5. **Event ID 4013 (DNS-Server-Service)**: This event means the DNS server is waiting for Active Directory Domain Services (AD DS) to signal that the initial synchronization of the directory has been completed. It could indicate an underlying issue with AD DS that's affecting replication.

Tech Savvy Productions

The **dcdiag /test:NetLogons** command is part of Microsoft's Directory Service Diagnostics Tool (dcdiag.exe) suite. It's used to check the status and health of various components of the Active Directory system on domain controllers.

The /test:NetLogons command specifically verifies that the Netlogon service on a domain controller is up and running properly. The Netlogon service plays a crucial role in the authentication process for computer and user accounts in the Active Directory domain. It's also responsible for other operations like updating and applying Group Policies.

Probable causes for failures with the dcdiag /test:NetLogons test could include:

1. **Service Interruption:** If the Netlogon service is stopped or not running properly on a domain controller, this could cause the test to fail.
2. **Network Issues:** Problems with the network, such as incorrect DNS settings, can interfere with the Netlogon service and cause the test to fail.
3. **Replication Errors:** If there are errors or delays in the replication of Active Directory data between domain controllers, the Netlogon service may not have the information it needs, causing the test to fail.
4. **SYSVOL/NETLOGON share issues:** If there are problems with the SYSVOL or NETLOGON shares, such as if they are missing or not shared properly, this can cause failures as well.
5. **File System Permissions:** If the permissions on the SYSVOL folder are incorrect, it may cause failures with the Netlogon service and result in test failure.

Tech Savvy Productions

To resolve these issues, an administrator should review the error message, identify the cause of the problem, and take appropriate action. This could involve restarting the Netlogon service, checking network settings, verifying replication health, or correcting file system permissions.

1. **Event ID 1053 (Userenv)**: This event is logged when Windows can't determine the user or computer name.
2. **Event ID 1058 (Userenv)**: This event is logged when Windows can't access the file for Group Policy.
3. **Event ID 5706 (NetLogon)**: This event is logged when Netlogon service couldn't create the shared file.
4. **Event ID 5719 (NetLogon)**: This event is logged when there are no domain controllers available for domain due to network issues or domain controllers are down.
5. **Event ID 5722 (NetLogon)**: This event indicates a secure session with a domain controller failed to be created due to account name collision.
6. **Event ID 5805 (NetLogon)**: This event is logged when an attempt to log on with an unknown account was made from a certain workstation.

The **dcdiag /test:ObjectsReplicated** command is a part of Microsoft's Directory Service Diagnostics Tool (dcdiag.exe) suite. The ObjectsReplicated test specifically checks if the Active Directory objects are being replicated correctly between domain controllers.

Tech Savvy Productions

Replication in Active Directory is a crucial process that ensures that changes made to the directory (like adding user accounts, creating group policies, etc.) on one domain controller are synchronized and updated on all other domain controllers. Probable causes for failures with the dcdiag /test:ObjectsReplicated test could include:

1. **Network Issues:** Problems with the network, such as incorrect DNS settings, firewalls blocking necessary traffic, or network latency, can interfere with replication and cause the test to fail.
2. **Replication Configuration Errors:** Errors in the configuration of replication between domain controllers, such as incorrect site and services settings, can lead to replication failures.
3. **Domain Controller Availability:** If one or more domain controllers are down or unreachable, it could cause replication to fail.
4. **Active Directory Database Corruption:** Although rare, if the AD database is corrupted on one or more domain controllers, replication can fail.
5. **Insufficient Permissions:** If the necessary permissions for replication are not correctly set, it can prevent objects from being replicated.

To resolve these issues, an administrator should review the error message, identify the cause of the problem, and take the appropriate action, such as fixing network settings, checking replication configurations, verifying the availability and health of domain controllers, or correcting permissions.

1. **Event ID 1084 (NTDS Replication)**: This event is logged when the Active Directory Domain Services could not update an object with changes received from the directory service at another network address because Active Directory Domain Services was busy processing information.
2. **Event ID 1388 or 1988 (NTDS Replication)**: These events are logged when a lingering object is detected. Lingering objects can occur if a domain controller does not replicate for an interval of time that is longer than the tombstone lifetime (TSL).
3. **Event ID 1864 (NTDS Replication)**: This event is logged when replication has not occurred with a partner for a period of time equal to the tombstone lifetime.
4. **Event ID 1311 (NTDS KCC)**: This event is logged when the Knowledge Consistency Checker (KCC) detects problems with the partitions in Active Directory. It could signal issues with the naming context.

Tech Savvy Productions

5. **Event ID 1925 (NTDS KCC)**: This event signifies that the attempt to establish a replication link for the specified partition with the specified domain controller failed.

The dcdiag /test:OutboundSecureChannels command in Microsoft's Directory Service Diagnostics Tool (dcdiag.exe) tests the ability of a domain controller to make outbound secure channel connections to other domain controllers.

Tech Savvy Productions

Secure channels are used by a domain controller for communications with other domain controllers that are part of the same domain or a different domain, for tasks like replication and user authentication. This test ensures the domain controller can establish these secure connections as expected.

Probable causes for failures with the dcdiag /test:OutboundSecureChannels test could include:

1. **Network Issues:** If there are problems with the network, such as incorrect DNS settings or firewalls blocking necessary traffic, this can prevent secure channel connections.
2. **Incorrect Time Settings:** If there is a significant time difference between domain controllers, Kerberos authentication, which secure channels depend on, may fail.
3. **Secure Channel Broken or Corrupted:** Occasionally, the secure channel can become broken or corrupted due to various reasons, which would then prevent successful connections.
4. **Active Directory Trust Issues:** If there are issues with trust relationships between domains, this could affect secure channel connections.
5. **Domain Controller Availability:** If one or more domain controllers are down or unreachable, it could cause secure channel connection failures.

To resolve these issues, an administrator should review the error message, identify the cause of the problem, and take the appropriate action. This could involve checking network settings, correcting time settings, resetting the secure channel, verifying trust relationships, or ensuring the availability and health of domain controllers.

1. **Event ID 5 (Kerberos)**: If there's an issue with time synchronization between domain controllers, which is crucial for Kerberos authentication, you might see this "Clock skew too great" event.
2. **Event ID 3210 (NETLOGON)**: This event is logged when a secure channel to a domain controller can't be established. This usually indicates issues with the machine account.
3. **Event ID 5721 (NETLOGON)**: This event signifies that the session setup to the Windows NT or Windows 2000 Domain Controller for the domain is not responsive.

Tech Savvy Productions

4. **Event ID 5722 (NETLOGON)**: This event is logged when a secure session with a domain controller failed to be created due to account name collision.
5. **Event ID 139 (Kerberos-Key-Distribution-Center)**: This event indicates problems with Kerberos Ticket-Granting-Ticket renewals, which can affect secure channel communications.
6. **Event ID 5719 (NETLOGON)**: This event is logged when there are no domain controllers available for domain due to network issues or domain controllers being down.

The **dcdiag /test:Replications** command in Microsoft's Directory Service Diagnostics Tool (dcdiag.exe) is used to check the status of Active Directory replication between the domain controllers.

Replication is a critical process in Active Directory that ensures any changes made to the directory (like adding user accounts, modifying group policies, etc.) on one domain controller are synchronized and updated on all other domain controllers.

Tech Savvy Productions

This test checks whether all domain controllers can communicate and sync properly.

Probable causes for failures with the dcdiag /test:Replications test could include:

- **Improper DNS Configuration:** AD heavily relies on DNS for the functioning of services. Errors in DNS server configuration, such as incorrect DNS entries for domain controllers or lack of SRV records, can cause replication issues.
- **Incorrect Site and Services Configuration:** If the configuration in the Active Directory Sites and Services tool is incorrect, it can lead to replication issues. This could include incorrect subnet definitions, incorrect site link definitions, or incorrect placement of domain controllers in sites.
- **Firewall and Network Issues:** If necessary ports for replication are not open in firewalls or if there are other network issues, replication can fail.
- **Inadequate Replication Permissions:** The account being used for replication needs to have the right permissions. Without them, the replication process will fail.
- **Replication Latency and Tombstone Lifetimes:** If there's a significant latency in replication, or if a domain controller has been offline for a time longer than the tombstone lifetime, replication can fail or create lingering objects, leading to potential inconsistencies.
- **Time Synchronization Issues:** AD uses the Kerberos protocol for authentication, which requires time synchronization. Significant time differences between domain controllers can cause replication issues.
- **Broken Trust Relationships:** If trust relationships between domain controllers are broken, it can lead to replication failure.
- **Incorrectly Seized or Transferred FSMO Roles:** If Flexible Single Master Operation (FSMO) roles are improperly seized or transferred, this can cause replication issues.

Here are some useful Event IDs that can help identify the specific issues related to Active Directory (AD) replication failures:

Tech Savvy Productions

- **Event ID 1311 (NTDS KCC)**: This event is logged when the Knowledge Consistency Checker (KCC) detects problems with the Active Directory partitions, which can point to issues with Sites and Services configurations.
- **Event ID 1053 (Userenv)**: This event is logged when Windows cannot determine the user or computer name. This may indicate DNS issues.
- **Event ID 4015 (DNS-Server-Service)**: This event indicates that the DNS server has encountered a critical error from the Active Directory, which might be caused by misconfigured DNS settings.
- **Event ID 1925 (NTDS KCC)**: This event signifies that the attempt to establish a replication link for a specific partition with a specified domain controller failed. This might be related to network/firewall issues or improper Site and Services configuration.
- **Event ID 2042 (NTDS Replication)**: This event is logged when a domain controller has been offline for too long (longer than the tombstone lifetime), causing it to be outdated and unable to replicate when it comes back online.
- **Event ID 2092 (NTDS Replication)**: This event is logged when a domain controller holding a FSMO role did not advertise as a time server. This could indicate a problem with time synchronization.
- **Event IDs 1988 and 1388 (NTDS Replication)**: These events are logged when lingering objects are detected, which are often due to problems with tombstone lifetime and replication latency.
- **Event ID 5722 (NetLogon)**: This event indicates a secure session with a domain controller failed to be created, possibly due to a broken trust relationship.

Some PowerShell scripts that are helpful in testing the health of your AD replication

Get-ADReplicationFailure - This cmdlet retrieves the data about replication failures that occurred within a specified duration on a specified domain controller.

Tech Savvy Productions

Example usage: Get-ADReplicationFailure -Target DC01 -Scope Site -TargetScope Site

```
PS C:\Windows\system32> Get-ADReplicationFailure -Target Win2016dc1, Win2016dc2


FailureCount     : 24
FailureType      : Link
FirstFailureTime : 6/30/2023 10:59:59 AM
LastError        : 1256
Partner          : CN=NTDS Settings,CN=WIN2016DC1,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=
                   HomeLab,DC=TechSavvyProductions,DC=com
PartnerGuid      : 28029095-585f-4851-bbd8-9a2e0f8ff03e
Server           : Win2016DC2.HomeLab.TechSavvyProductions.com
```

The results above show 2 domain controllers off

Get-ADReplicationConnection - This cmdlet retrieves the AD replication connection data that is used by the Knowledge Consistency Checker (KCC) or administrators to manage replication topology.

Example usage: Get-ADReplicationConnection -Filter *

Get-ADReplicationPartnerMetadata - This cmdlet retrieves the replication metadata for a specified domain controller, such as last successful replication, replication failure count, etc.

Example usage: Get-ADReplicationPartnerMetadata -Target "DC=MyDomain,DC=Com" -Partition *

Repadmin PowerShell Wrapper - Repadmin is a command-line tool that's been used for years to troubleshoot AD replication issues. You can use this PowerShell script to call repadmin commands more conveniently:

```
Function Check-Replication {
  param (
    [Parameter(Mandatory=$true)]
    [string]$dc
  )

  $output = repadmin /showrepl $dc
  $output
}

Check-Replication -dc "DC01"
```

replace "DC01" with the hostname of your Domain Controller in this example

Tech Savvy Productions

here is the results of the above script on my DC3 with 2 DCs off

```
Check-Replication -dc "Win2016dc3"
Default-First-Site-Name\WIN2016DC3

DSA Options: IS_GC

Site Options: (none)

DSA object GUID: a0597b7a-2eed-4a03-9674-d44c6a8ac635

DSA invocationID: 013443f1-ff5b-4a6e-8ea3-1cddbc4f0eb2


==== INBOUND NEIGHBORS ======================================


DC=HomeLab,DC=TechSavvyProductions,DC=com

    Default-First-Site-Name\WIN2016DC1 via RPC

        DSA object GUID: 28029095-585f-4851-bbd8-9a2e0f8ff03e

        Last attempt @ 2023-07-03 05:52:44 failed, result 1722 (0x6ba):

            The RPC server is unavailable.

        67 consecutive failure(s).

        Last success @ 2023-06-30 10:51:56.

    Default-First-Site-Name\WIN2016DC2 via RPC

        DSA object GUID: 60228dba-cbc8-4304-868a-28f53a302f46

        Last attempt @ 2023-07-03 05:54:09 was successful.


CN=Configuration,DC=HomeLab,DC=TechSavvyProductions,DC=com

    Default-First-Site-Name\WIN2016DC1 via RPC

        DSA object GUID: 28029095-585f-4851-bbd8-9a2e0f8ff03e

        Last attempt @ 2023-07-03 05:53:26 failed, result 1722 (0x6ba):

            The RPC server is unavailable.

        67 consecutive failure(s).

        Last success @ 2023-06-30 10:51:55.

    Default-First-Site-Name\WIN2016DC2 via RPC

        DSA object GUID: 60228dba-cbc8-4304-868a-28f53a302f46

        Last attempt @ 2023-07-03 05:53:26 was successful.


CN=Schema,CN=Configuration,DC=HomeLab,DC=TechSavvyProductions,DC=com

    Default-First-Site-Name\WIN2016DC1 via RPC

        DSA object GUID: 28029095-585f-4851-bbd8-9a2e0f8ff03e
```

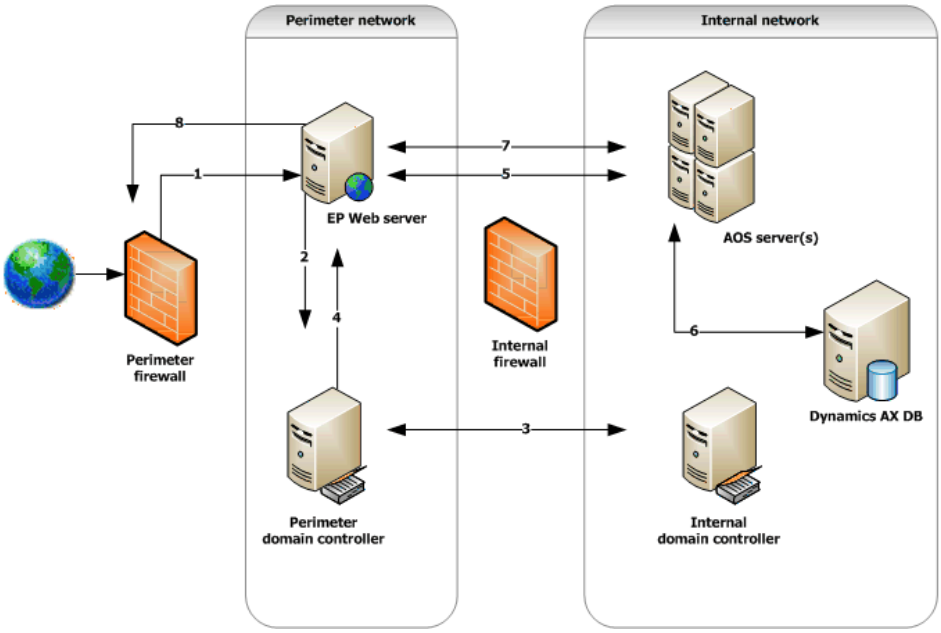

Tech Savvy Productions

```
        Last attempt @ 2023-07-03 05:54:09 failed, result 1722 (0x6ba):
            The RPC server is unavailable.
        67 consecutive failure(s).
        Last success @ 2023-06-30 10:51:55.
    Default-First-Site-Name\WIN2016DC2 via RPC
        DSA object GUID: 60228dba-cbc8-4304-868a-28f53a302f46
        Last attempt @ 2023-07-03 05:54:09 was successful.

DC=ForestDnsZones,DC=HomeLab,DC=TechSavvyProductions,DC=com
    Default-First-Site-Name\WIN2016DC1 via RPC
        DSA object GUID: 28029095-585f-4851-bbd8-9a2e0f8ff03e
        Last attempt @ 2023-07-03 05:52:44 failed, result 1256 (0x4e8):
            The remote system is not available. For information about network troubleshooting,
see Windows Help.
        67 consecutive failure(s).
        Last success @ 2023-06-30 10:51:56.
    Default-First-Site-Name\WIN2016DC2 via RPC
        DSA object GUID: 60228dba-cbc8-4304-868a-28f53a302f46
        Last attempt @ 2023-07-03 05:54:09 was successful.

DC=DomainDnsZones,DC=HomeLab,DC=TechSavvyProductions,DC=com
    Default-First-Site-Name\WIN2016DC1 via RPC
        DSA object GUID: 28029095-585f-4851-bbd8-9a2e0f8ff03e
        Last attempt @ 2023-07-03 05:52:44 failed, result 1256 (0x4e8):
            The remote system is not available. For information about network troubleshooting,
see Windows Help.
        67 consecutive failure(s).
        Last success @ 2023-06-30 10:51:56.
    Default-First-Site-Name\WIN2016DC2 via RPC
        DSA object GUID: 60228dba-cbc8-4304-868a-28f53a302f46
        Last attempt @ 2023-07-03 05:54:09 was successful.

Source: Default-First-Site-Name\WIN2016DC1
******* 67 CONSECUTIVE FAILURES since 2023-06-30 10:51:56
Last error: 1256 (0x4e8):
```


Tech Savvy Productions

```
            The remote system is not available. For information about network troubleshooting,
see Windows Help.
```

Firewall ports that must be open between AD domain controllers:



Active Directory relies on multiple services for its operation, and each of these services uses specific network ports. Here's a list of some of the key ports that need to be open on a firewall to allow Active Directory to function correctly:

LDAP (Lightweight Directory Access Protocol):

- TCP 389: Used for requests to the directory service.
- UDP 389: Used for requests to the directory service.

LDAP over SSL/TLS:

Tech Savvy Productions

- TCP 636: Used for secure requests to the directory service.

LDAP Ping:

- UDP 389: Used for LDAP Ping, which is used to determine the availability of domain controllers.

Kerberos:

- TCP 88: Used for Kerberos authentication.
- UDP 88: Used for Kerberos authentication.

DNS (Domain Name System):

- TCP 53: Used for DNS zone transfers.
- UDP 53: Used for DNS queries.

Global Catalog:

- TCP 3268: Used for requests to the global catalog.
- TCP 3269: Used for secure requests to the global catalog.

SMB (Server Message Block):

- TCP 445: Used for file replication and user and computer authentication.

RPC (Remote Procedure Call):

- TCP 135: Used as the RPC endpoint mapper.
- TCP 1024-65535 (when using RPC over dynamic ports) or 49152-65535 (when using RPC over static ports): Used for RPC endpoint communication.

NTP (Network Time Protocol):

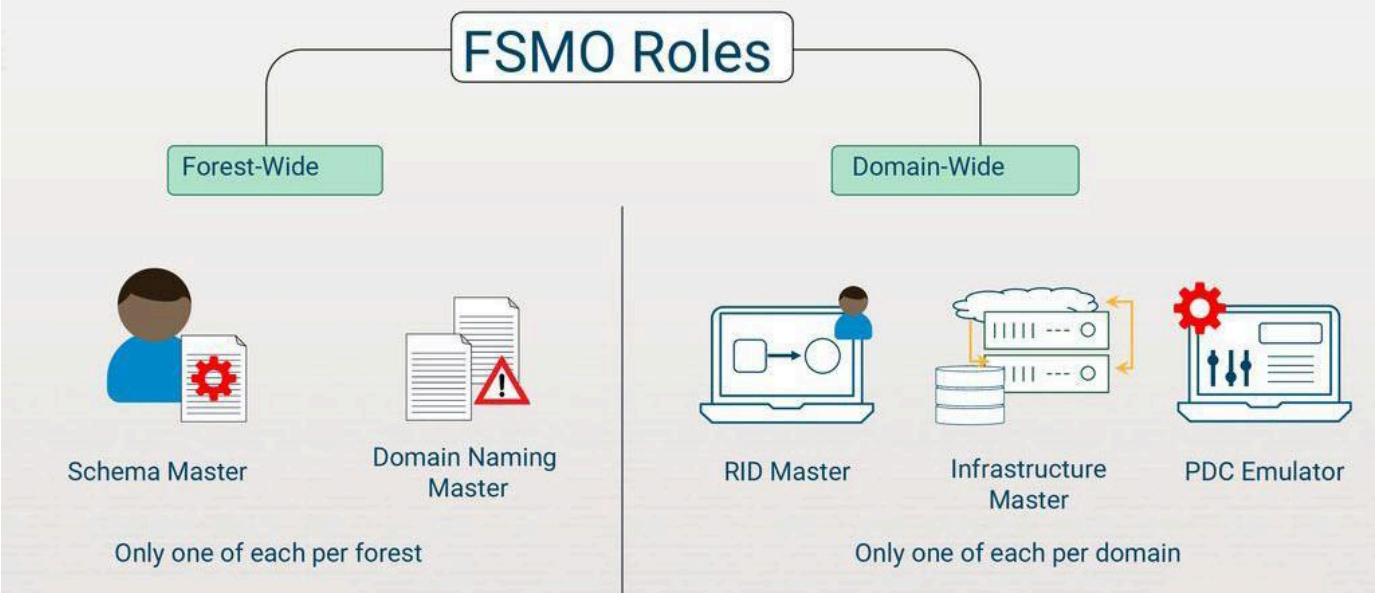
- UDP 123: Used for time synchronization.

Active Directory Web Services:

Tech Savvy Productions

- **TCP 9389**: Used for Active Directory Web Services for tools like Active Directory PowerShell and Active Directory Administrative Center.

The **dcdiag /test:RIDManager** command is a part of Microsoft's Directory Service Diagnostics Tool (dcdiag.exe) suite. This test specifically checks the operations and health of the Relative ID (RID) Master role (also known as the RID pool manager, RID manager, or RID operations master).



In an Active Directory environment, each object requires a unique Security Identifier (SID). A SID consists of a domain SID, which is the same for all SIDs created in a domain, and a relative ID (RID) that is unique for each security principal SID created in a domain.

The RID Master is one of the five Flexible Single Master Operations (FSMO) roles in Active Directory, and it's responsible for issuing a unique RID pool to each domain

Tech Savvy Productions

controller in its domain. When a domain controller creates a new AD object (like a user or group), it assigns the object a unique SID from its RID pool.

Probable causes for failures with the dcdiag /test:RIDManager test could include:

- **FSMO Role Failure:** If the RID Manager (one of the FSMO roles) is not functioning correctly, this can cause the test to fail.
- **Network Issues:** Problems with the network, such as incorrect DNS settings or firewalls blocking necessary traffic, can interfere with the operations of the RID Manager.
- **Domain Controller Availability:** If the domain controller holding the RID Master role is down or unreachable, it could cause the test to fail.
- **Active Directory Database Corruption:** Although rare, if the AD database is corrupted on the domain controller holding the RID Master role, it can cause this test to fail.

This could involve potentially seizing or transferring the RID Master FSMO role to another domain controller.

Event IDs can provide crucial information when troubleshooting issues with the Relative ID (RID) Master. Here are a few important Event IDs that may help:

1. **Event ID 16650 (SAM)**: This event indicates that the Security Accounts Manager (SAM) has determined that the security identifier (SID) for this domain could not be created. This could happen if the RID master is not available or unable to process requests.
2. **Event ID 16651 (SAM)**: This event indicates that there's a problem with the RID master, such as it might be offline or misconfigured.
3. **Event ID 8213 (MS DTC SID)**: This event indicates a failure when MS DTC (Microsoft Distributed Transaction Coordinator) tries to assign a security identifier from the local domain controller, possibly due to RID master issues.
4. **Event ID 1699 (Active Directory Domain Service)**: This event is logged when a request for a new account-identifier pool failed because the RID master is unreachable.

Tech Savvy Productions

5. **Event ID 1555 (NTDS Replication)**: This event is logged when the allocation of the relative identifier block failed because the directory service is paused or started after being paused.
6. This event indicates that the server was not able to renew its identifier pool. This may occur if the RID master for the domain is not available.

Services critical to running Active Directory on DCs

- **Active Directory Domain Services (AD DS)**: This is the core service that implements Active Directory. It stores information about objects on the network and makes this information easy for administrators and users to find and use.
- **DNS Server (DNS)**: DNS is critical for the functioning of Active Directory as it provides name resolution for the domain. AD uses DNS to locate domain controllers, global catalog servers, and other crucial components.
- **Kerberos Key Distribution Center (KDC)**: This service is responsible for handling all Kerberos ticket requests, which are crucial for the authentication of users and computers in an AD domain.
- **Intersite Messaging (IsmServ)**: This service is used for exchanging the replication topology between sites.
- **File Replication Service (FRS) or Distributed File System Replication (DFSR)**: These services are used for replicating the SYSVOL share between domain controllers. DFSR is used in newer versions of Windows Server, replacing FRS.

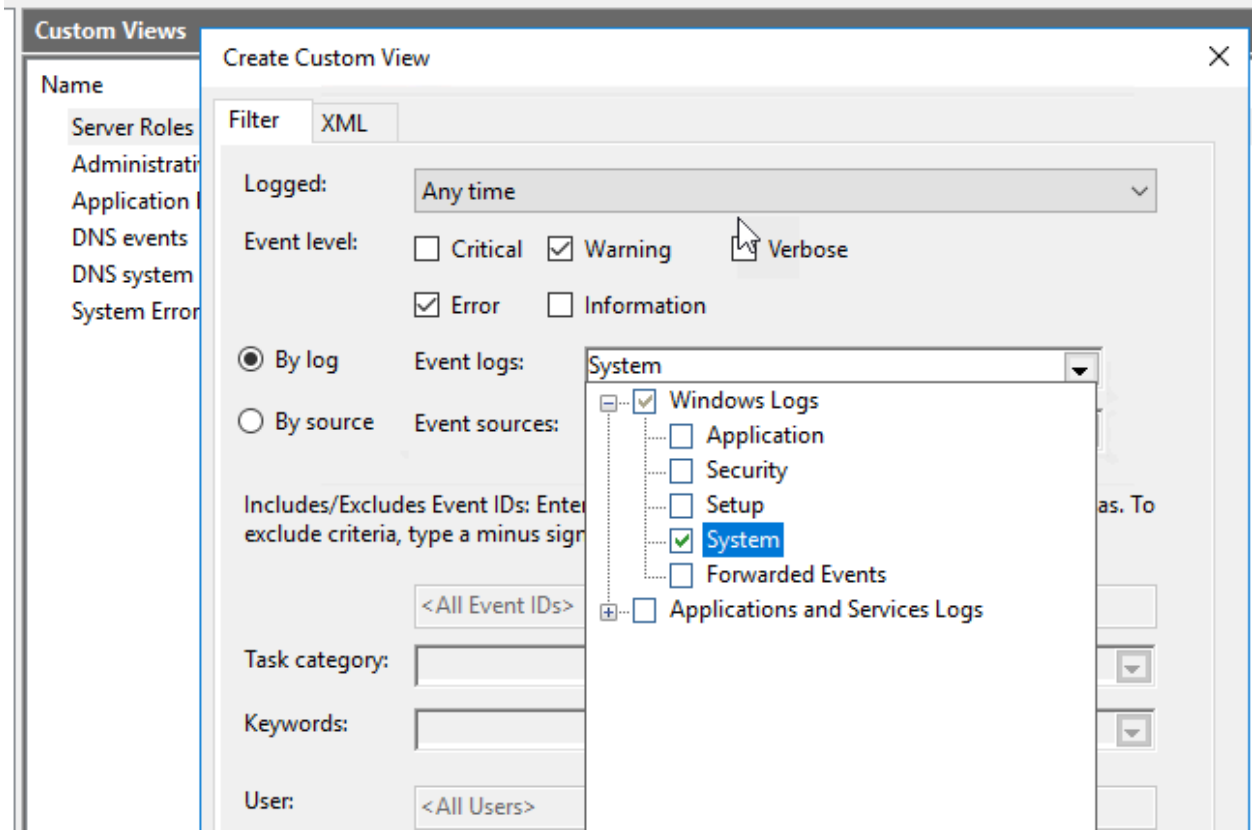
Tech Savvy Productions

- **Netlogon**: This service is responsible for allowing a server to function as a domain controller and authenticates users and other domain controllers in a domain.
- **Windows Time (W32Time)**: This service is critical for synchronizing the time across all devices in an Active Directory domain, which is essential for Kerberos authentication to work correctly.
- **Remote Procedure Call (RPC)**: This service is used for communication between various components of Active Directory, including during replication between domain controllers.

Now all has been heard;
here is the conclusion of the matter:
Fear God and keep his commandments,
for this is the duty of all mankind.
For God will bring every deed into judgment,
including every hidden thing,
whether it is good or evil.

The **dcdiag /test:SystemLog** command is part of Microsoft's Directory Service Diagnostics Tool (dcdiag.exe) suite. This test specifically verifies the System event log for any **Error** **or** **Warning events** logged by the NetLogon service.

Tech Savvy Productions



The System event log contains events logged by Windows system components. For example, the failure of a driver or other system component to load during startup is recorded in the System event log.

- **Event ID 1054 (GroupPolicy)**: This event indicates that Windows couldn't obtain the domain controller name for your computer network during the Group Policy update. This could signal issues with DNS, network connectivity, or the availability of domain controllers.
- **Event ID 1055 (GroupPolicy)**: This event signifies that the processing of Group Policy failed due to lack of network connectivity to a domain controller. It's indicative of network, DNS, or domain controller availability issues.
- **Event ID 5719 (NETLOGON)**: This event is logged when a domain controller couldn't be found for the specified domain or the Netlogon service failed to establish a secure channel to a domain controller. This could be due to network issues, DNS problems, or domain trust relationship issues.

Tech Savvy Productions

- **Event ID 5722 (NETLOGON)**: This event is logged when a secure session with a domain controller failed to be created, possibly due to a broken trust relationship or outdated computer account password.
- **Event ID 3210 (NETLOGON)**: This event signifies that a secure session with a domain controller could not be established, possibly due to outdated computer account password.
- **Event ID 7023 (Service Control Manager)**: This event is logged when a service unexpectedly stops. If the service is related to AD operation (like Netlogon, DFSR, etc.), it could indicate an issue affecting the health of AD.
- **Event ID 7, 11, 55, 130 (Disk)**: These events are related to hard disk errors. They can indicate physical hardware issues or file system corruption, which could impact the operation of AD if they occur on a domain controller.

He has shown you, O mortal, what is good.

And what does the Lord require of you?

To act justly and to love mercy

and to walk humbly with your God.

Tech Savvy Productions

The **dcdiag /test:VerifyEnterpriseReferences** command is part of Microsoft's Directory Service Diagnostics Tool (dcdiag.exe).

The VerifyEnterpriseReferences test specifically checks for consistency of Enterprise Read-Only Domain Controller (RODC) security principal references in the Active Directory database.

Enterprise RODC security principals are security groups used for allowing Read-Only Domain Controllers to authenticate user and computer accounts in a domain. This includes the "Allowed RODC Password Replication Group" and "Denied RODC Password Replication Group".

Therefore, this test is particularly important in environments where Read-Only Domain Controllers are deployed, ensuring that there is consistency in the security group membership information held within the Active Directory database.

If the dcdiag /test:VerifyEnterpriseReferences test fails, it suggests that there are inconsistencies with the RODC security principals in your Active Directory database, which could potentially lead to authentication and authorization issues.

- **Event ID 1107 (Source: ADWS)**: This event indicates that the Active Directory Web Services (ADWS) could not determine if the domain controller is a global catalog server. This is important as RODCs often interact with global catalog servers to facilitate authentication and directory lookups.
- **Event ID 2023 (Source: REPADMIN)**: This event ID is associated with replication issues, which could be crucial as RODCs rely on unidirectional replication from a writable domain controller.
- **Event ID 1699 (Source: Active Directory Domain Service)**: This event is logged when an RODC is unable to contact the domain naming FSMO to request a new pool of relative identifiers (RIDs).
- **Event ID 2886 (Source: Active Directory Domain Service)**: This event signifies that an LDAP client attempted to connect to the directory service on the RODC with a set of authentication credentials that causes the directory to return a referral.
- **Event IDs 8192 - 8200 (Source: DFSR)**: These events are related to the Distributed File System Replication (DFSR) service, which RODCs use to replicate the SYSVOL share.

Tech Savvy Productions

- **Event ID 1645 (Source: Active Directory Domain Service)**: This event signifies that an attempt was made to replicate an object to an RODC while the object was in the process of being moved to the LostAndFound container.
- **Event ID 2038 (Source: NTDS Replication)**: This event indicates an error during the promotion of an RODC, which can help diagnose issues with the initial setup and promotion of an RODC.

In Windows Server 2008 , you can also take advantage of read-only domain controllers (RODCs). An RODC is a new type of domain controller that hosts read-only partitions of the Active Directory database. Except for account passwords, an RODC holds all the Active Directory objects and attributes that a writable domain controller holds. However, changes can't be made to the database that is stored on the RODC. Changes must be made on a writable domain controller and then replicated back to the RODC.

An RODC is designed primarily to be deployed in remote or branch office environments, which typically have relatively few users, poor physical security, relatively poor network bandwidth to a hub site, and personnel with limited knowledge of information technology (IT). Deploying RODCs results in improved security and more efficient access to network resources. For more information about RODC features, see AD DS: Read-Only Domain Controllers. For information about how to deploy an RODC, see the Read-Only Domain Controllers Step-by-Step Guide

If I speak in the tongues of men or of angels, but do not have love, I am only a resounding gong or a clanging cymbal. If I have the gift of prophecy and can fathom all mysteries and all knowledge, and if I have a faith that can move mountains, but do not have love, I am nothing. If I give all I possess to the poor and give over my body to hardship that I may boast, but do not have love, I gain nothing.

Love is patient, love is kind. It does not envy, it does not boast, it is not proud. It does not dishonor others, it is not self-seeking, it is not easily angered, it keeps no record of wrongs. Love does not delight in evil but rejoices with the truth. It always protects, always trusts, always hopes, always perseveres.

The **dcdiag /test:VerifyReferences** command is part of Microsoft's Directory Service Diagnostics Tool (dcdiag.exe).

The VerifyReferences test checks to make sure that all directory references are correctly resolved in the Active Directory database. More specifically, this test

Tech Savvy Productions

verifies that **all back-links** in the directory database are correctly established and that the **forward-links and back-links are consistent** with each other.

1. **Forward-Link**: A forward-link is a reference from one object to another object it refers to. For example, in a group object, the "member" attribute contains the Distinguished Names (DNs) of all user objects that are members of the group. So, the "member" attribute is a forward-link from a group to its members.

   Forward-links are multi-valued, which means they can contain multiple references. In the group example, the "member" attribute (forward-link) contains references to all the users that are members of that group.

2. **Back-Link**: A back-link is a reference from an object back to another object that refers to it. Taking the group example again, for every user object listed in the "member" attribute of a group, there is a corresponding "memberOf" attribute in the user object that points back to the group. So, the "memberOf" attribute is a back-link.

   Back-links are derived from the values of their corresponding forward-links, so they are effectively read-only. You cannot directly modify a back-link attribute. Instead, changes to the forward-link are automatically reflected in the back-link.

These link relationships are vital for representing associations between objects in AD, such as group memberships, manager and direct report relationships, and so on.

https://neroblanco.co.uk/2015/07/links-and-backlinks-in-active-directory-for-exchange/

A back-link is a reference from one object back to another object that refers to it. For example, if User1 is a member of Group1, then User1 has a back-link to Group1. On the other hand, a forward-link is a reference from one object to another object it refers to. For instance, in the same example, Group1 would have a forward-link to User1.

Inconsistencies in the forward-links and back-links can lead to issues with group membership, access control, and various other aspects of Active Directory operations.

If the dcdiag /test:VerifyReferences test fails, it suggests that there are inconsistencies in the references in your Active Directory database, and further investigation would be required to find and fix the issue.

Tech Savvy Productions

Of *making many books there is no end, and much study wearies the body.*

Now all has been heard;
here is the conclusion of the matter:
Fear God and keep his commandments,
for this is the duty of all mankind.
For God will bring every deed into judgment,
including every hidden thing,
whether it is good or evil.

VerifyReplicas	Checks that all **application directory partitions** are fully **instantiated** on all replica servers.

Active Directory Application Partitions are directory partitions that are replicated only to specified domain controllers. A domain controller that stores a particular application partition in its local copy of the directory hosts a replica of that partition. An application partition is said to be **"instantiated"** on a domain controller when the domain controller has been designated to host a replica of that partition.

https://www.google.com/search?client=firefox-b-1-d&q=instantiated

When an application partition is instantiated on a domain controller, it means that the partition has been created and populated with its initial data, and is now

Tech Savvy Productions

ready to be updated and replicated to other domain controllers that are also hosting that partition.

The process of instantiation can involve the creation of the partition's objects and attributes, the setting of replication boundaries, and the initiation of the first replication event for that partition.

In summary, when you see the term "instantiated" being used with reference to Active Directory application partitions, it typically means that a replica of the partition has been created and is actively being hosted on a particular domain controller.

- **Event ID 1126 (Source: Active Directory Domain Service)**: This event is logged when the Active Directory Domain Services (AD DS) was unable to establish a connection with the global catalog, which could impact the application directory partitions.
- **Event ID 1982 (Source: NTDS Replication)**: This event is logged when the domain controller hosting the DNS server was unable to replicate the application directory partition. This could occur due to a variety of reasons, including network connectivity, security, or replication latency.
- **Event ID 1983 (Source: NTDS Replication)**: This event is logged when a domain controller attempts to replicate an application directory partition it already hosts.
- **Event ID 2103 (Source: Microsoft-Windows-ActiveDirectory_DomainService)**: The Replication Engine failed to initialize the inbound replication thread for a directory partition. It happens when there is a failure in initializing the replication of a specific application directory partition.

Tech Savvy Productions

Whoever of you loves life

and desires to see many good days,

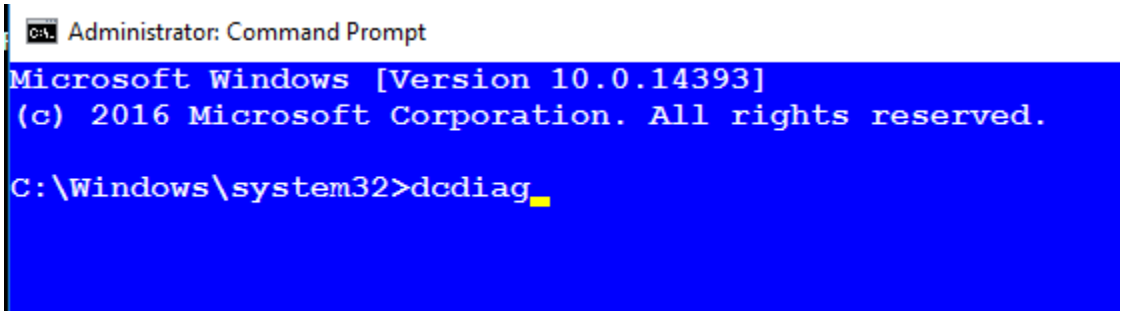
keep your tongue from evil

and your lips from telling lies.

Turn from evil and do good;

seek peace and pursue it.

1. **Testing General Health:** Run `dcdiag` without any switches to perform a series of default tests on the local domain controller. This can provide a quick snapshot of your Active Directory's health.



Tech Savvy Productions

2. **Testing Specific Domain Controllers:** To run DCDiag tests on a specific domain controller, you can use the `/s:<servername>` switch. For example: `dcdiag /s:DC01`.

```
C:\Windows\system32>dcdiag /s:Win2016dc2

Directory Server Diagnosis

Performing initial setup:
   * Identified AD Forest.
   Done gathering initial info.
Doing initial required tests
   Testing server: Default-First-Site-Name\WIN2016DC2
      Starting test: Connectivity
         ..... WIN2016DC2 passed test Connectivity
Doing primary tests
   Testing server: Default-First-Site-Name\WIN2016DC2
      Starting test: Advertising
         ..... WIN2016DC2 passed test Advertising
      Starting test: FrsEvent
```


Tech Savvy Productions

..... WIN2016DC2 passed test FrsEvent
Starting test: DFSREvent
There are warning or error events within the last 24 hours after the SYSVOL has been shared.
Failing SYSVOL
replication problems may cause Group Policy problems.
..... WIN2016DC2 failed test DFSREvent
Starting test: SysVolCheck
..... WIN2016DC2 passed test SysVolCheck
Starting test: KccEvent
..... WIN2016DC2 passed test KccEvent
Starting test: KnowsOfRoleHolders
..... WIN2016DC2 passed test KnowsOfRoleHolders
Starting test: MachineAccount
..... WIN2016DC2 passed test MachineAccount
Starting test: NCSecDesc
..... WIN2016DC2 passed test NCSecDesc
Starting test: NetLogons
..... WIN2016DC2 passed test NetLogons
Starting test: ObjectsReplicated
..... WIN2016DC2 passed test ObjectsReplicated
Starting test: Replications
..... WIN2016DC2 passed test Replications
Starting test: RidManager
..... WIN2016DC2 passed test RidManager
Starting test: Services
..... WIN2016DC2 passed test Services
Starting test: SystemLog
..... WIN2016DC2 passed test SystemLog
Starting test: VerifyReferences
..... WIN2016DC2 passed test VerifyReferences
Running partition tests on : ForestDnsZones
Starting test: CheckSDRefDom
..... ForestDnsZones passed test CheckSDRefDom
Starting test: CrossRefValidation
..... ForestDnsZones passed test CrossRefValidation
Running partition tests on : DomainDnsZones
Starting test: CheckSDRefDom
..... DomainDnsZones passed test CheckSDRefDom
Starting test: CrossRefValidation
..... DomainDnsZones passed test CrossRefValidation
Running partition tests on : Schema
Starting test: CheckSDRefDom
..... Schema passed test CheckSDRefDom
Starting test: CrossRefValidation
..... Schema passed test CrossRefValidation
Running partition tests on : Configuration
Starting test: CheckSDRefDom
..... Configuration passed test CheckSDRefDom

Tech Savvy Productions

```
    Starting test: CrossRefValidation
      ..... Configuration passed test CrossRefValidation
  Running partition tests on : HomeLab
    Starting test: CheckSDRefDom
      ..... HomeLab passed test CheckSDRefDom
    Starting test: CrossRefValidation
      ..... HomeLab passed test CrossRefValidation
  Running enterprise tests on : HomeLab.TechSavvyProductions.com
    Starting test: LocatorCheck
      ..... HomeLab.TechSavvyProductions.com passed test LocatorCheck
    Starting test: Intersite
      ..... HomeLab.TechSavvyProductions.com passed test Intersite
```

3. **Testing DNS Registration and Resolution:** The /test:DNS switch can be used to verify that the domain controllers' DNS records are properly registered and resolvable, which is critical for AD functionality.

```
C:\Windows\system32>dcdiag /test:DNS
Directory Server Diagnosis
Performing initial setup:
  Trying to find home server...
  Home Server = Win2016DC2
  * Identified AD Forest.
  Done gathering initial info.
Doing initial required tests
  Testing server: Default-First-Site-Name\WIN2016DC2
    Starting test: Connectivity
      ..... WIN2016DC2 passed test Connectivity
Doing primary tests
  Testing server: Default-First-Site-Name\WIN2016DC2
    Starting test: DNS
      DNS Tests are running and not hung. Please wait a few minutes...
      ..... WIN2016DC2 passed test DNS
  Running partition tests on : ForestDnsZones
  Running partition tests on : DomainDnsZones
  Running partition tests on : Schema
```


Tech Savvy Productions

```
Running partition tests on : Configuration
Running partition tests on : HomeLab
Running enterprise tests on : HomeLab.TechSavvyProductions.com
  Starting test: DNS
    ..... HomeLab.TechSavvyProductions.com passed test DNS
```

4. **Testing Replication:** The /test:Replications switch tests the replication between domain controllers. For example: dcdiag /test:replications.

```
C:\Windows\system32>dcdiag /test:replications

Directory Server Diagnosis
Performing initial setup:
  Trying to find home server...
  Home Server = Win2016DC2
  * Identified AD Forest.
  Done gathering initial info.
Doing initial required tests
  Testing server: Default-First-Site-Name\WIN2016DC2
    Starting test: Connectivity
      ..... WIN2016DC2 passed test Connectivity
Doing primary tests
  Testing server: Default-First-Site-Name\WIN2016DC2
    Starting test: Replications
      ..... WIN2016DC2 passed test Replications
  Running partition tests on : ForestDnsZones
  Running partition tests on : DomainDnsZones
  Running partition tests on : Schema
  Running partition tests on : Configuration
```


Tech Savvy Productions

Running partition tests on : HomeLab
Running enterprise tests on : HomeLab.TechSavvyProductions.com

When you run this test, it checks for any replication errors between the Domain Controllers. Here's a brief breakdown of what happens:

- The test begins **by locating all the Domain Controllers** within the domain.
- It then **attempts to perform a replication with each of these DCs**.
- The test checks **if there are any failures or errors** during these replication attempts.

Here are some potential results you might get:

- **Passed replication test**: If all Domain Controllers are able to successfully replicate with each other, then the test will indicate a pass. This suggests that AD replication is functioning correctly.
- **Failed replication test**: If there is an error or failure during replication, the test will fail. The output will typically include the error message returned by the replication system, which can help you troubleshoot the issue. Common causes for failures include network connectivity issues, DNS problems, or issues with the AD services on the Domain Controllers.

It's important to remember that if replication isn't working properly, changes made on one Domain Controller might not propagate to others. This can cause inconsistency in your Active Directory environment and can lead to a variety of problems, such as authentication failures, Group Policy inconsistencies, and more.

Therefore, running this test can be a vital part of your Active Directory health check process. If any errors are detected, they should be addressed as soon as possible to ensure the smooth operation of your network.

5. **Testing Specific AD Partitions:** You can use the /test:CrossRefValidation switch to verify the integrity of specific AD partitions.

```
C:\Windows\system32>dcdiag /test:CrossRefValidation
Directory Server Diagnosis
Performing initial setup:
   Trying to find home server...
   Home Server = Win2016DC1
   * Identified AD Forest.
   Done gathering initial info.
Doing initial required tests
   Testing server: Default-First-Site-Name\WIN2016DC1
      Starting test: Connectivity
         ..... WIN2016DC1 passed test Connectivity
Doing primary tests
   Testing server: Default-First-Site-Name\WIN2016DC1
   Running partition tests on : ForestDnsZones
      Starting test: CrossRefValidation
         ..... ForestDnsZones passed test CrossRefValidation
   Running partition tests on : DomainDnsZones
      Starting test: CrossRefValidation
         ..... DomainDnsZones passed test CrossRefValidation
   Running partition tests on : Schema
      Starting test: CrossRefValidation
```


Tech Savvy Productions

```
      ..... Schema passed test CrossRefValidation
   Running partition tests on : Configuration
     Starting test: CrossRefValidation
      ..... Configuration passed test CrossRefValidation
   Running partition tests on : HomeLab
     Starting test: CrossRefValidation
      ..... HomeLab passed test CrossRefValidation
   Running enterprise tests on : HomeLab.TechSavvyProductions.com
```

6. **Testing the RID Master:** The /test:RidManager switch can be used to verify the functionality of the Relative Identifier (RID) master, which is a critical AD FSMO role.

```
C:\Windows\system32>dcdiag /test:RidManager
Directory Server Diagnosis
Performing initial setup:
   Trying to find home server...
   Home Server = Win2016DC1
   * Identified AD Forest.
   Done gathering initial info.
Doing initial required tests

   Testing server: Default-First-Site-Name\WIN2016DC1
     Starting test: Connectivity
      ..... WIN2016DC1 passed test Connectivity
Doing primary tests
   Testing server: Default-First-Site-Name\WIN2016DC1
     Starting test: RidManager
      ..... WIN2016DC1 passed test RidManager
   Running partition tests on : ForestDnsZones
   Running partition tests on : DomainDnsZones
   Running partition tests on : Schema
   Running partition tests on : Configuration
   Running partition tests on : HomeLab
   Running enterprise tests on : HomeLab.TechSavvyProductions.com
```


Tech Savvy Productions

dcdiag is a command-line tool provided by Microsoft that allows administrators to check the state and functionality of various components of an Active Directory (AD) Domain Controller. The /test:RidManager option specifically tests the operation of the Relative ID (RID) master role within the domain.

The **RID master role is responsible for processing RID pool requests from all domain controllers in a particular domain**. Every object in an AD domain has a unique Security Identifier (SID). The SID consists of a domain SID, which is the same for all SIDs created in a domain, and a RID. **The RID is unique for each security principal SID created in a domain.**

The test dcdiag /test:RidManager checks that the RID master is available and able to service requests. This is crucial for the proper operation of the AD domain, as without available RIDs, no new AD objects (like users or computers) can be created.

Running partition tests on the following components refers to the different partitions in Active Directory, each of which stores a different type of information:

- **ForestDNSZones**: This application partition contains forest-wide DNS data which is stored in the forest DNS zone.
- **DomainDNSZones**: This application partition contains domain-wide DNS data stored in the domain DNS zone.
- **Schema**: This partition contains the schema for the forest. The schema is the set of definitions for all object classes and their attributes within the AD forest.
- **Configuration**: This partition contains configuration information for the entire forest. This includes information about the physical structure of the forest, including what sites and servers exist in the forest.
- **Domain**: This partition contains all objects in the domain including users, groups, computers, and organizational units.

A successful test would indicate that the AD Domain Controller, and all the related services and roles, are functioning correctly. If there were errors or failures, it would give you a clue as to where the problem might be - whether it's a network issue, a DNS issue, a problem with the RID Master, and so on. Each error should be addressed individually based on the specifics of the error message.

Tech Savvy Productions

7. **Testing the Trust Relationships:** The /test:CheckSecurityError switch can be used to identify any security-related issues in trust relationships between domains.

```
C:\Windows\system32>dcdiag /test:CheckSecurityError
Directory Server Diagnosis
Performing initial setup:
   Trying to find home server...
   Home Server = Win2016DC1
   * Identified AD Forest.
   Done gathering initial info.
Doing initial required tests
  Testing server: Default-First-Site-Name\WIN2016DC1
     Starting test: Connectivity
        ..... WIN2016DC1 passed test Connectivity
Doing primary tests
   Testing server: Default-First-Site-Name\WIN2016DC1
     Starting test: CheckSecurityError
        [WIN2016DC1] No security related replication errors were found on this DC!  To target the
connection to a
        specific source DC use /ReplSource:<DC>.
        ..... WIN2016DC1 passed test CheckSecurityError
   Running partition tests on : ForestDnsZones
   Running partition tests on : DomainDnsZones
   Running partition tests on : Schema
   Running partition tests on : Configuration
```


Tech Savvy Productions

```
Running partition tests on : HomeLab
Running enterprise tests on : HomeLab.TechSavvyProductions.com
```

Here's a brief breakdown of the test:

- It will **first identify all the domain controllers** in the domain.
- Then, it checks for potential security-related errors by **examining the security context of the replication process between the DCs**.
- The test focuses on critical security components such as the Security Accounts Manager (SAM), **security descriptors**, and the **sysvol share**.
- It ensures that there are **no security inconsistencies or errors** that could be preventing normal Active Directory replication and operation.

The possible results could be:

- **Passed CheckSecurityError test**: If the security context for all of these components is consistent and there are no detected errors, then the test will return a success message. This means that, from a security perspective, replication appears to be functioning normally.
- **Failed CheckSecurityError test**: If there is an error detected, the test will fail and return an error message. This error message will typically provide details about what kind of security error has occurred. These could include things like access denied errors, indicating that a DC doesn't have the proper permissions to replicate a certain piece of data. Other issues could include misconfigured security descriptors or issues with the sysvol share.

Understanding these results can help you identify and troubleshoot any security-related issues that may be interfering with normal Active Directory operations. Any issues detected should be addressed promptly to maintain the integrity and functionality of your domain.

8. **Checking Event Logs:** Use the /test:Topology or /test:SystemLog to check for errors and warnings in the Event Log that relate to Active Directory.

```
C:\Windows\system32>dcdiag /test:SystemLog
Directory Server Diagnosis
Performing initial setup:
   Trying to find home server...
   Home Server = Win2016DC1
   * Identified AD Forest.
   Done gathering initial info.
Doing initial required tests
   Testing server: Default-First-Site-Name\WIN2016DC1
      Starting test: Connectivity
         ..... WIN2016DC1 passed test Connectivity
Doing primary tests
   Testing server: Default-First-Site-Name\WIN2016DC1
      Starting test: SystemLog
         An error event occurred.  EventID: 0x00002720
            Time Generated: 06/24/2023   08:19:31
            Event String:
            The application-specific permission settings do not grant Local Activation permission for the COM
Server application with CLSID
         ..... WIN2016DC1 failed test SystemLog
   Running partition tests on : ForestDnsZones
   Running partition tests on : DomainDnsZones
   Running partition tests on : Schema
```


Tech Savvy Productions

```
  Running partition tests on : Configuration
  Running partition tests on : HomeLab
  Running enterprise tests on : HomeLab.TechSavvyProductions.com
```

1. **ForestDnsZones**: This partition contains DNS data applicable to the entire forest.
2. **DomainDnsZones**: This partition contains DNS data applicable to the specific domain.
3. **Schema**: This partition contains the schema of the Active Directory. The schema defines object classes and their attributes that are used within the AD forest.
4. **Configuration**: This partition contains forest-wide configuration information, such as information about available sites or services.

```
C:\Windows\system32>dcdiag /test:Topology
Directory Server Diagnosis
Performing initial setup:
   Trying to find home server...
   Home Server = Win2016DC1
   * Identified AD Forest.
   Done gathering initial info.
Doing initial required tests
   Testing server: Default-First-Site-Name\WIN2016DC1
      Starting test: Connectivity
         ..... WIN2016DC1 passed test Connectivity
Doing primary tests

   Testing server: Default-First-Site-Name\WIN2016DC1
      Starting test: Topology
         ..... WIN2016DC1 passed test Topology
   Running partition tests on : ForestDnsZones
   Running partition tests on : DomainDnsZones
   Running partition tests on : Schema
   Running partition tests on : Configuration
   Running partition tests on : HomeLab
   Running enterprise tests on : HomeLab.TechSavvyProductions.com

C:\Windows\system32>dcdiag /test:Topology /v /a

Directory Server Diagnosis

Performing initial setup:
   Trying to find home server...
   * Verifying that the local machine Win2016DC1, is a Directory Server.
   Home Server = Win2016DC1
   * Connecting to directory service on server Win2016DC1.
   * Identified AD Forest.
   Collecting AD specific global data
   * Collecting site info.
```

```
   Calling
ldap_search_init_page(hld,CN=Sites,CN=Configuration,DC=HomeLab,DC=TechSavvyProductions,DC=com
,LDAP_SCOPE_SUBTREE,(objectCategory=ntDSSiteSettings),.......
   The previous call succeeded
   Iterating through the sites
   Looking at base site object: CN=NTDS Site
Settings,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=HomeLab,DC=TechSavvyProductions
,DC=com
   Getting ISTG and options for the site
   * Identifying all servers.
   Calling
ldap_search_init_page(hld,CN=Sites,CN=Configuration,DC=HomeLab,DC=TechSavvyProductions,DC=com
,LDAP_SCOPE_SUBTREE,(objectClass=ntDSDsa),.......
   The previous call succeeded....
   The previous call succeeded
   Iterating through the list of servers
   Getting information for the server CN=NTDS
Settings,CN=WIN2016DC1,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=Home
Lab,DC=TechSavvyProductions,DC=com
   objectGuid obtained
   InvocationID obtained
   dnsHostname obtained
   site info obtained
   All the info for the server collected
   Getting information for the server CN=NTDS
Settings,CN=WIN2016DC2,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=Home
Lab,DC=TechSavvyProductions,DC=com
   objectGuid obtained
   InvocationID obtained
   dnsHostname obtained
   site info obtained
   All the info for the server collected
   Getting information for the server CN=NTDS
Settings,CN=WIN2016DC3,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=Home
Lab,DC=TechSavvyProductions,DC=com
   objectGuid obtained
   InvocationID obtained
   dnsHostname obtained
   site info obtained
   All the info for the server collected
   * Identifying all NC cross-refs.
   * Found 3 DC(s). Testing 3 of them.
   Done gathering initial info.

Doing initial required tests

   Testing server: Default-First-Site-Name\WIN2016DC1
      Starting test: Connectivity
```


Tech Savvy Productions

```
        * Active Directory LDAP Services Check
        Determining IP4 connectivity
        * Active Directory RPC Services Check
        ..... WIN2016DC1 passed test Connectivity

    Testing server: Default-First-Site-Name\WIN2016DC2
      Starting test: Connectivity
        * Active Directory LDAP Services Check
        Determining IP4 connectivity
        * Active Directory RPC Services Check
        ..... WIN2016DC2 passed test Connectivity

    Testing server: Default-First-Site-Name\WIN2016DC3
      Starting test: Connectivity
        * Active Directory LDAP Services Check
        Determining IP4 connectivity
        * Active Directory RPC Services Check
        ..... WIN2016DC3 passed test Connectivity

Doing primary tests

      Starting test: Topology
        * Configuration Topology Integrity Check
        * Analyzing the connection topology for
DC=ForestDnsZones,DC=HomeLab,DC=TechSavvyProductions,DC=com.
        * Performing upstream (of target) analysis.
        * Performing downstream (of target) analysis.
        * Analyzing the connection topology for
DC=DomainDnsZones,DC=HomeLab,DC=TechSavvyProductions,DC=com.
        * Performing upstream (of target) analysis.
        * Performing downstream (of target) analysis.
        * Analyzing the connection topology for
CN=Schema,CN=Configuration,DC=HomeLab,DC=TechSavvyProductions,DC=com.
        * Performing upstream (of target) analysis.
        * Performing downstream (of target) analysis.
        * Analyzing the connection topology for
CN=Configuration,DC=HomeLab,DC=TechSavvyProductions,DC=com.
        * Performing upstream (of target) analysis.
        * Performing downstream (of target) analysis.
        * Analyzing the connection topology for DC=HomeLab,DC=TechSavvyProductions,DC=com.
        * Performing upstream (of target) analysis.
        * Performing downstream (of target) analysis.
        ..... WIN2016DC1 passed test Topology


    Testing server: Default-First-Site-Name\WIN2016DC2
      Starting test: Topology
        * Configuration Topology Integrity Check
```


Tech Savvy Productions

```
      * Analyzing the connection topology for
DC=ForestDnsZones,DC=HomeLab,DC=TechSavvyProductions,DC=com.
      * Performing upstream (of target) analysis.
      * Performing downstream (of target) analysis.
      * Analyzing the connection topology for
DC=DomainDnsZones,DC=HomeLab,DC=TechSavvyProductions,DC=com.
      * Performing upstream (of target) analysis.
      * Performing downstream (of target) analysis.
      * Analyzing the connection topology for
CN=Schema,CN=Configuration,DC=HomeLab,DC=TechSavvyProductions,DC=com.
      * Performing upstream (of target) analysis.
      * Performing downstream (of target) analysis.
      * Analyzing the connection topology for
CN=Configuration,DC=HomeLab,DC=TechSavvyProductions,DC=com.
      * Performing upstream (of target) analysis.
      * Performing downstream (of target) analysis.
      * Analyzing the connection topology for DC=HomeLab,DC=TechSavvyProductions,DC=com.
      * Performing upstream (of target) analysis.
      * Performing downstream (of target) analysis.
      ..... WIN2016DC2 passed test Topology
   Starting test: Topology
      * Configuration Topology Integrity Check
      * Analyzing the connection topology for
DC=ForestDnsZones,DC=HomeLab,DC=TechSavvyProductions,DC=com.
      * Performing upstream (of target) analysis.
      * Performing downstream (of target) analysis.
      * Analyzing the connection topology for
DC=DomainDnsZones,DC=HomeLab,DC=TechSavvyProductions,DC=com.
      * Performing upstream (of target) analysis.
      * Performing downstream (of target) analysis.
      * Analyzing the connection topology for
CN=Schema,CN=Configuration,DC=HomeLab,DC=TechSavvyProductions,DC=com.
      * Performing upstream (of target) analysis.
      * Performing downstream (of target) analysis.
      * Analyzing the connection topology for
CN=Configuration,DC=HomeLab,DC=TechSavvyProductions,DC=com.
      * Performing upstream (of target) analysis.
      * Performing downstream (of target) analysis.
      * Analyzing the connection topology for DC=HomeLab,DC=TechSavvyProductions,DC=com.
      * Performing upstream (of target) analysis.
      * Performing downstream (of target) analysis.
      ..... WIN2016DC3 passed test Topology
```

The `dcdiag /test:Topology` command is used to test the integrity of the topology of your Active Directory environment. It checks that the **intersite**

Tech Savvy Productions

topology generator (ISTG) has created a suitable and effective topology for Active Directory replication.
Here's what to look for in the results:

- **Passed test Topology**: If the test passes, that means that the ISTG has created a suitable topology and all Domain Controllers are connected in a way that allows for effective replication.
- **Failed test Topology**: If the test fails, that means there may be an issue with the topology of your Active Directory environment. The specific error message will provide more details about the nature of the problem. Common issues could be isolated domain controllers, issues with site links, or issues with the KCC (Knowledge Consistency Checker) which helps create the replication topology.

In case of any failures, you should investigate further based on the error details and take necessary actions to resolve the problem.

This might involve modifying.

- site links,
- checking network connectivity between DCs
- looking into issues with the ISTG or KCC.

Tech Savvy Productions

9. **Verifying FSMO Role Holders:** The /test:KnowsOfRoleHolders switch can confirm whether all domain controllers are aware of the correct Flexible Single Master Operations (FSMO) role holders.

```
C:\Windows\system32>dcdiag /test:KnowsOfRoleHolders
Directory Server Diagnosis
Performing initial setup:
   Trying to find home server...
   Home Server = Win2016DC1
   * Identified AD Forest.
   Done gathering initial info.
Doing initial required tests
   Testing server: Default-First-Site-Name\WIN2016DC1
      Starting test: Connectivity
         ..... WIN2016DC1 passed test Connectivity
Doing primary tests
   Testing server: Default-First-Site-Name\WIN2016DC1
      Starting test: KnowsOfRoleHolders
         ..... WIN2016DC1 passed test KnowsOfRoleHolders
   Running partition tests on : ForestDnsZones
   Running partition tests on : DomainDnsZones
   Running partition tests on : Schema
   Running partition tests on : Configuration
   Running partition tests on : HomeLab
   Running enterprise tests on : HomeLab.TechSavvyProductions.com
```


Tech Savvy Productions

```
C:\Windows\system32> netdom query fsmo
Schema master               Win2016DC1.HomeLab.TechSavvyProductions.com
Domain naming master        Win2016DC1.HomeLab.TechSavvyProductions.com
PDC                         Win2016DC1.HomeLab.TechSavvyProductions.com
RID pool manager            Win2016DC1.HomeLab.TechSavvyProductions.com
Infrastructure master       Win2016DC1.HomeLab.TechSavvyProductions.com
The command completed successfully.
```

FSMO roles are special roles in an Active Directory environment that are held by one or more Domain Controllers.

There are five of these roles:

- Schema Master,
- Domain Naming Master,
- Relative Identifier (RID) Master
- Primary Domain Controller (PDC) Emulator
- Infrastructure Master.

Each of these roles must be held by a DC, and each role (except for the PDC Emulator and RID Master roles which are unique to each domain) must be unique within the entire forest.

Therefore, it's important that every DC knows which DC holds each role, to ensure proper functioning of the AD environment.

Here's how you interpret the results of the dcdiag /test:KnowsOfRoleHolders command:

- **Passed test KnowsOfRoleHolders**: This result means that **all the Domain Controllers are aware of the current FSMO role holders**. Each DC has the correct information about which DC holds each FSMO role. This indicates that your Active Directory environment is functioning correctly with respect to FSMO role assignment.
- **Failed test KnowsOfRoleHolders**: This result means that one or more Domain Controllers do not have the correct information about the FSMO role holders. The output should include details about which DCs have incorrect information, and what the incorrect information is. This could be

Tech Savvy Productions

due to a variety of issues, including network connectivity problems, replication issues, or problems with the FSMO role holders themselves.

If you encounter a failure, you should investigate based on the details provided in the test output. For example, if a DC believes that a FSMO role is held by a DC that no longer exists, you might need to seize the FSMO role using a tool like `ntdsutil`. If the issue is due to network or replication problems, you would need to address those issues first.

10. **Testing Services:** The /test:Services switch can be used to ensure that all required services are running on the domain controllers.

```
C:\Windows\system32>dcdiag /test:Services
Directory Server Diagnosis
Performing initial setup:
   Trying to find home server...
   Home Server = Win2016DC1
   * Identified AD Forest.
   Done gathering initial info.
Doing initial required tests
   Testing server: Default-First-Site-Name\WIN2016DC1
      Starting test: Connectivity
         ..... WIN2016DC1 passed test Connectivity
Doing primary tests

   Testing server: Default-First-Site-Name\WIN2016DC1
      Starting test: Services
         ..... WIN2016DC1 passed test Services
   Running partition tests on : ForestDnsZones
   Running partition tests on : DomainDnsZones
   Running partition tests on : Schema
```


Tech Savvy Productions

```
Running partition tests on : Configuration
Running partition tests on : HomeLab
Running enterprise tests on : HomeLab.TechSavvyProductions.com
```

When you run dcdiag /test:Services, it checks if all the required services are running on the DC. **The specific services checked** may vary slightly depending on the version of Windows Server you're using, but generally include:

- Active Directory Domain Services (NTDS)
- DNS Server (DNS)
- File Replication Service (NTFRS) or Distributed File System Replication (DFSR), depending on the server version
- Kerberos Key Distribution Center (KDC)
- Intersite Messaging (IsmServ)
- Net Logon (NetLogon)
- Windows Time Service (W32Time)
- And possibly others, depending on server version and role configuration

Here's how you interpret the results of the dcdiag /test:Services command:

- **Passed test Services**: If you see this result, it means **all the necessary services for the operation of the DC are running as expected**. This indicates that, from a services perspective, your DC is functioning correctly.
- **Failed test Services**: If you see this result, **it means one or more services that are vital for the operation of the DC are not running**. The test output should give you more details about which services are not running correctly. This issue could cause problems with your DC's functionality and should be addressed immediately. The solution will depend on the service that is not running, but often involves troubleshooting why the service isn't starting, which can involve checking service dependencies, permissions, or issues with the service itself.

Please note that DCDiag should be run from an elevated command prompt (run as administrator), and results should be carefully reviewed to understand the potential impact on your AD environment.

Tech Savvy Productions

DcDiag allows you to perform a quick general health test of Active Directory and domain controllers. To check all DCs in the domain, use the /e parameter.

The following command will only list errors that require the AD administrator's attention:

```
dcdiag /e /v /q
```

You can perform a specific AD test only by specifying its name, for example:

```
dcdiag /s:DC01 /a /test:NetLogons
```

or test the health of the RID master FSMO owner in the domain:

Dcdiag.exe /TEST:RidManager /v

Or you can exclude a specific test from the checklist:

```
dcdiag /s:DC01 /a /skip:Replication
```

When launching the DcDiag tool remotely, you need to specify the credentials with the domain admin privileges:

```
dcdiag /s:DC01 /u:contosoadmin /p:P@SSwoord
```

In order to display the extended information and save the test results to files, use the command:

Tech Savvy Productions

```
dcdiag /s:DC01 /v /f:c:\ps\dcdiag_report.log
```

To test all domain controllers in the current Active Directory site, run the command:

```
dcdiag /s:DC01 /a
```

If you want to remove the extra information from the test results to display only the errors found, use the /q parameter (if no errors were found, the command will return nothing):

```
dcdiag /s:DC01 /q
```

Some trivial errors can be fixed with DcDiag by itself. To do this, use the /fix switch:

```
dcdiag /s:DC01 /fix
```

You can check the health of your name resolution service in AD using the DNS tests. For example, to run all DNS tests for a specific domain controller and export the result to a text file:

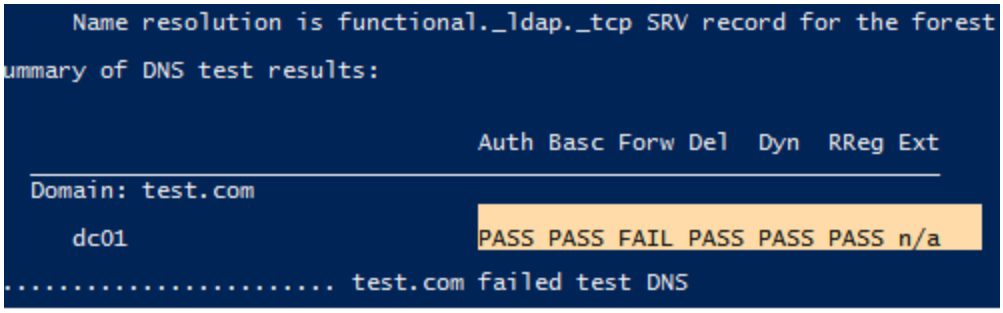
```
DCDiag /Test:DNS /e /v /s:dc01.theitbros.com
>c:\logs\DcdiagDNSCheck.txt
```

Open the resulting DNS test log file:

```
Get-Content c:\logs\DcdiagDNSCheck.txt
```

The result of each DNS test is listed in a column under the “Summary of DNS test results” section. In this example, all DNS tests passed successfully (PASS), except for the DNS forwarding test (FAIL):

Tech Savvy Productions



You can only run certain DNS tests:

dcdiag /test:DNS [/DnsBasic | /DnsForwarders | /DnsDelegation | /DnsDynamicUpdate | /DnsRecordRegistration | /DnsResolveExtName [/DnsInternetName:<InternetName>] | /DnsAll] [/f:<LogFile>] [/x:<XMLLog.xml>] [/xsl:<XSLFile.xsl> or <XSLTFile.xslt>] [/s:<DomainController>] [/e] [/v]

/DnsBasic	Basic DNS tests, connectivity, DNS client configuration, service availability, presence of a domain zone
/DnsForwarders	DnsBasic Tests and DNS Forwarding
/DnsDelegation	DnsBasic Tests and delegation verification
/DnsDynamicUpdate	Tests DnsBasic and checks if dynamic update is enabled for an Active Directory zone
/DnsRecordRegistration	DnsBasic tests and also checks if A records, CNAMEs, and SRV services

Tech Savvy Productions

	are registered. In addition, an inventory report is generated based on the test results.
/DnsResolveExtName **[/DnsInternetName:<**InternetName>]	DnsBasic tests and resolves the InternetName. If DnsInternetName is not specified, the tool attempts to resolve www.microsoft.com address. If DnsInternetName is specified, it resolves the specified InternetName.

For example, you need to automatically fix some common DNS errors. Use the following command to fix any errors found in the DNS service on the specified domain controller:

```
DCDiag /Test:DNS /e /v /s:dc01.theitbros.com /fix
```

Active Directory replication troubleshooting

Active Directory replication errors can be challenging to troubleshoot, but several utilities can help identify and resolve these issues. Here are a few examples:

1. **RepAdmin:** This is a command-line tool included with Windows Server. It's extremely useful for diagnosing replication issues. For example, repadmin /showrepl will display the replication status of each domain controller.

```
C:\Windows\system32>repadmin /replsummary
Replication Summary Start Time: 2023-06-24 09:53:25
Beginning data collection for replication summary, this may take awhile:
  ......
Source DSA          largest delta    fails/total %%   error
 WIN2016DC1             40m:04s    0 /  10    0
 WIN2016DC2             02m:59s    0 /  10    0
 WIN2016DC3             39m:46s    0 /  10    0
Destination DSA     largest delta    fails/total %%   error
 WIN2016DC1             02m:59s    0 /  10    0
 WIN2016DC2             40m:04s    0 /  10    0
```


Tech Savvy Productions

WIN2016DC3 01m:38s 0 / 10 0

```
C:\Windows\system32>repadmin /replsummary
Replication Summary Start Time: 2023-06-24 09:53:25

Beginning data collection for replication summary, this may take awhile:
  ......


Source DSA          largest delta    fails/total %%   error
 WIN2016DC1                 40m:04s    0 /  10    0
 WIN2016DC2                 02m:59s    0 /  10    0
 WIN2016DC3                 39m:46s    0 /  10    0


Destination DSA     largest delta    fails/total %%   error
 WIN2016DC1                 02m:59s    0 /  10    0
 WIN2016DC2                 40m:04s    0 /  10    0
 WIN2016DC3                 01m:38s    0 /  10    0
```

Here's how you interpret the results of the `repadmin /replsummary` command:

1. **Source DSA**: The Source Domain System Agent (DSA), **essentially the Domain Controller, from which replication is being attempted.**
2. **Largest Delta**: This is **the longest amount of time the DC has gone without being able to successfully replicate from a specific replication partner**. In a healthy AD environment, this should ideally be a small number.
3. **Failures**: This shows the number of failed replication attempts from each replication partner. **Ideally, this should be 0 in a healthy AD** environment.
4. **Error**: This column indicates the last error (if any) that was encountered during a replication attempt. If replication is working as expected, this column should say "0" or "None".

If the "Largest Delta" and "Failures" columns are both showing reasonable numbers (low delta and zero failures), and the "Error" column shows "0" or "None", then the AD environment is healthy from a replication perspective.

However, if the "Largest Delta" is a large number, there are multiple failures, and/or there are non-zero error codes listed, these are indications that there are issues with replication between Domain Controllers in your AD environment.

Tech Savvy Productions

If the repadmin /replsummary command indicates that there are replication issues, you should investigate these further. This could involve examining the event logs on the affected DCs, checking network connectivity between DCs, examining the replication topology, or investigating the specific error codes returned by repadmin.

2. **Active Directory Replication Status Tool:** This is a GUI tool from Microsoft that provides a visual overview of the replication status of domain controllers in a forest. It can be easier to use than command-line tools for some administrators. (**no longer available** but the GitHub project below is hopefully replacing it)

https://github.com/ryanries/ADReplStatus

ADReplStatus

Active Directory Replication Status Tool, aka ADREPLSTATUS

Joseph Ryan Ries - 2023

This project is NOT supported or endorsed by Microsoft. I work for Microsoft, but this is a personal project.

This tool is still very early in development. Right now I need beta testers to report bugs, submit feature requests, and send me screenshots of ADREPLSTATUS that I can post here in this README to spruce up the page.

I am rebuilding the AD Replication Status Tool. Here are my reasons:

Tech Savvy Productions

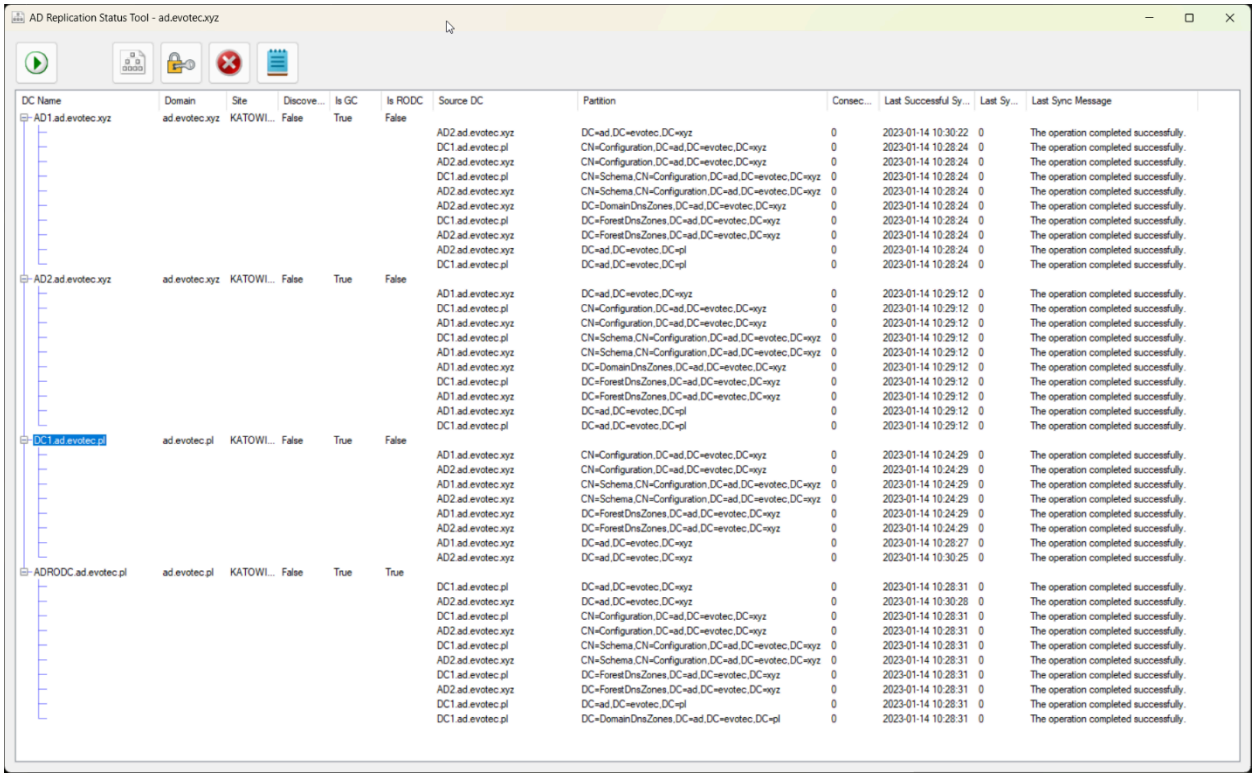
- The old version of the tool had a time bomb in it – an expiring SSL certificate – that rendered the app unusable sometime around September 2022. Only through great effort internally were we able to periodically renew this certificate and republish the app, and the app was architected in such a way that excising the signing certificate check was more work than just rewriting the whole thing. (I know about the Sirona DLL hack, but still.)

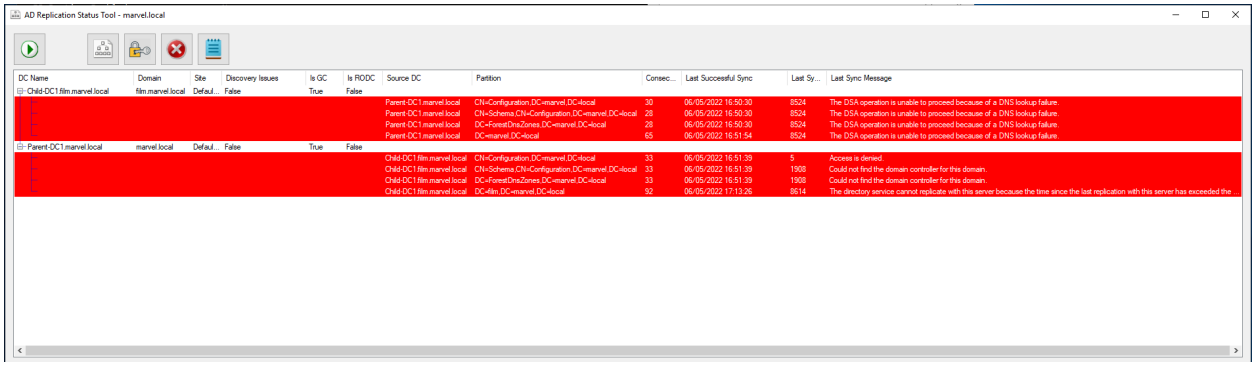
- The old tool had no active maintainers and no one who was still around was familiar with the app's internals or source code.

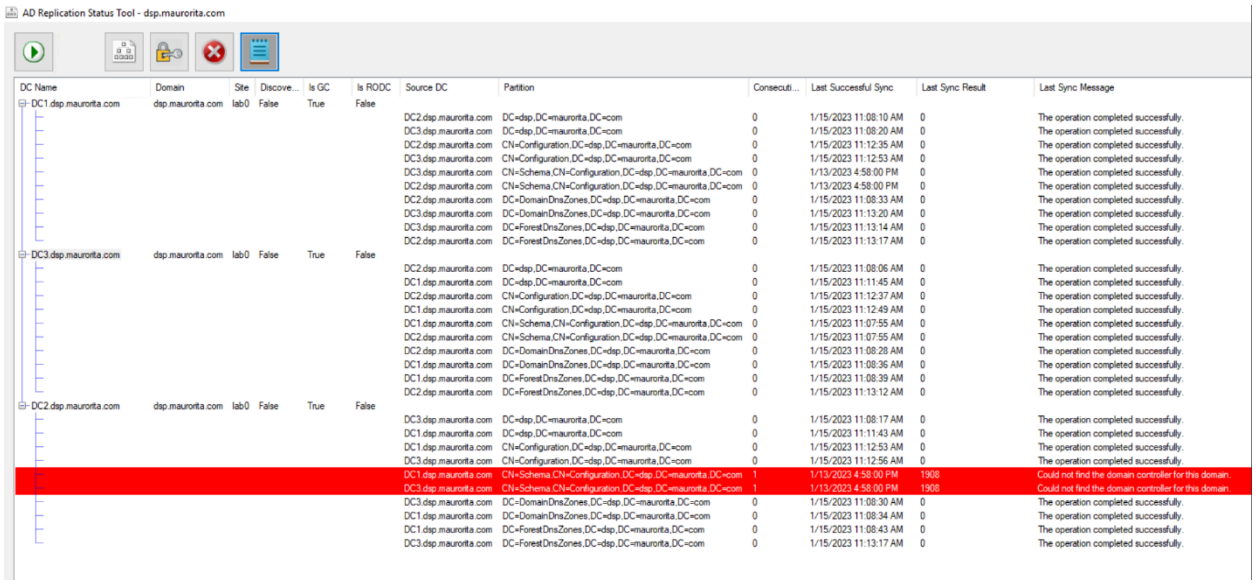
- The old tool was showing its age… I mean, that Office-style ribbon was straight out of the Windows Vista era, was it not? Plus it contained a bunch of outdated TechNet links. You might argue that WinForms is similarly old-fashioned, but I just so happen to think WinForms is and always was PEAK Windows UI.

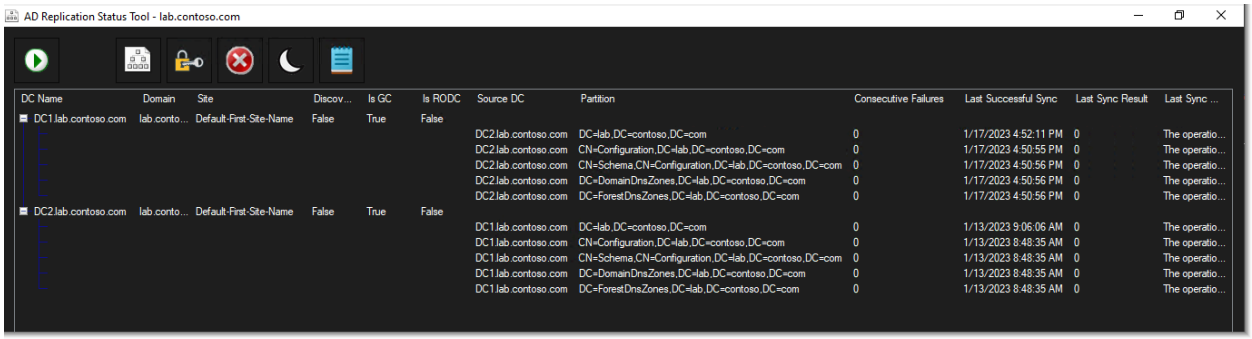
- Now we can maintain and improve this tool together. The Active Directory community.

Here are some screenshots of what the tool looks like currently:

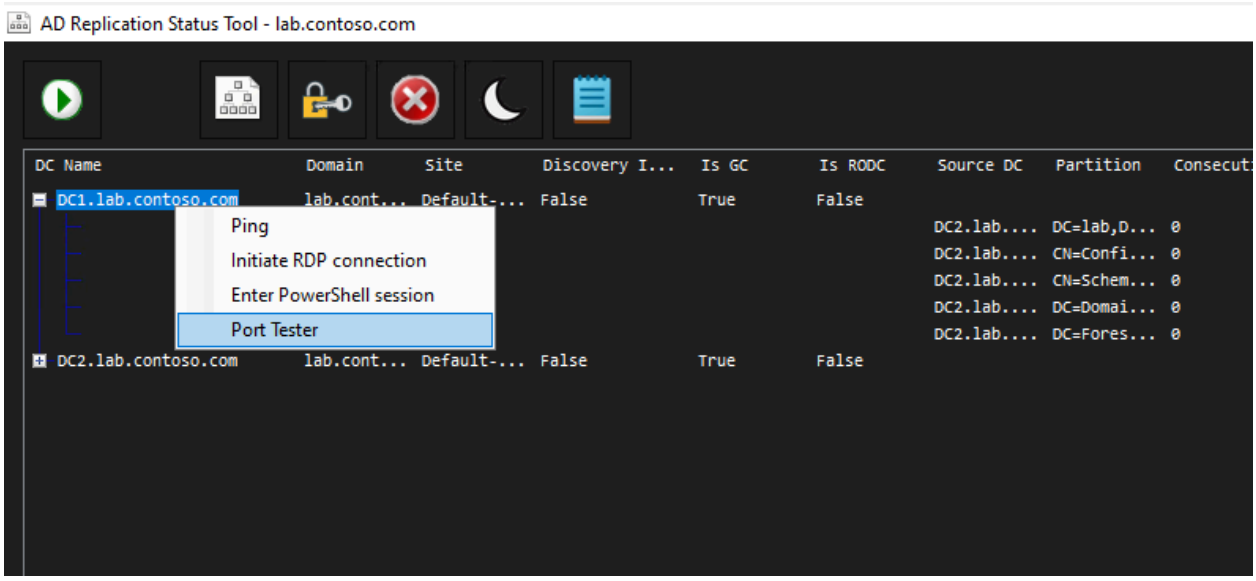


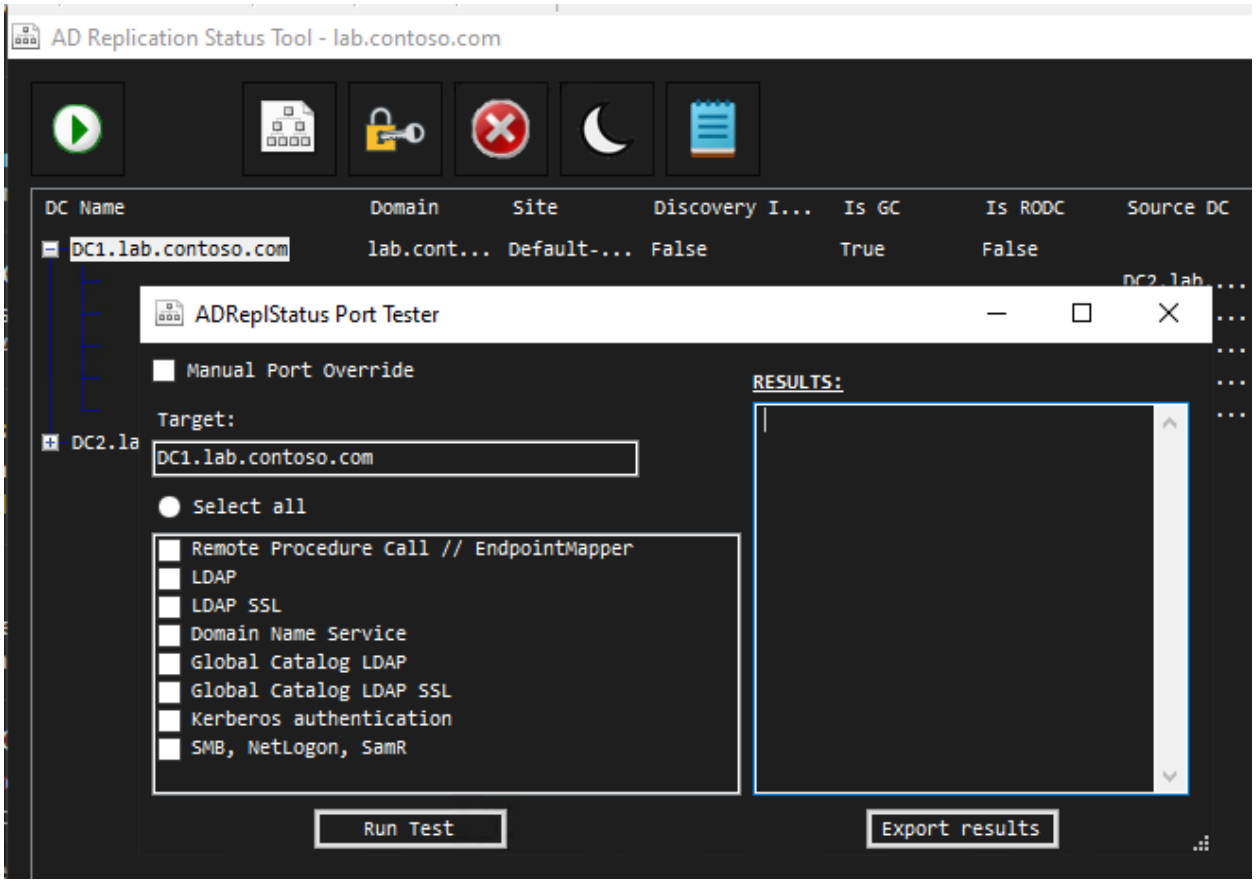






Tech Savvy Productions





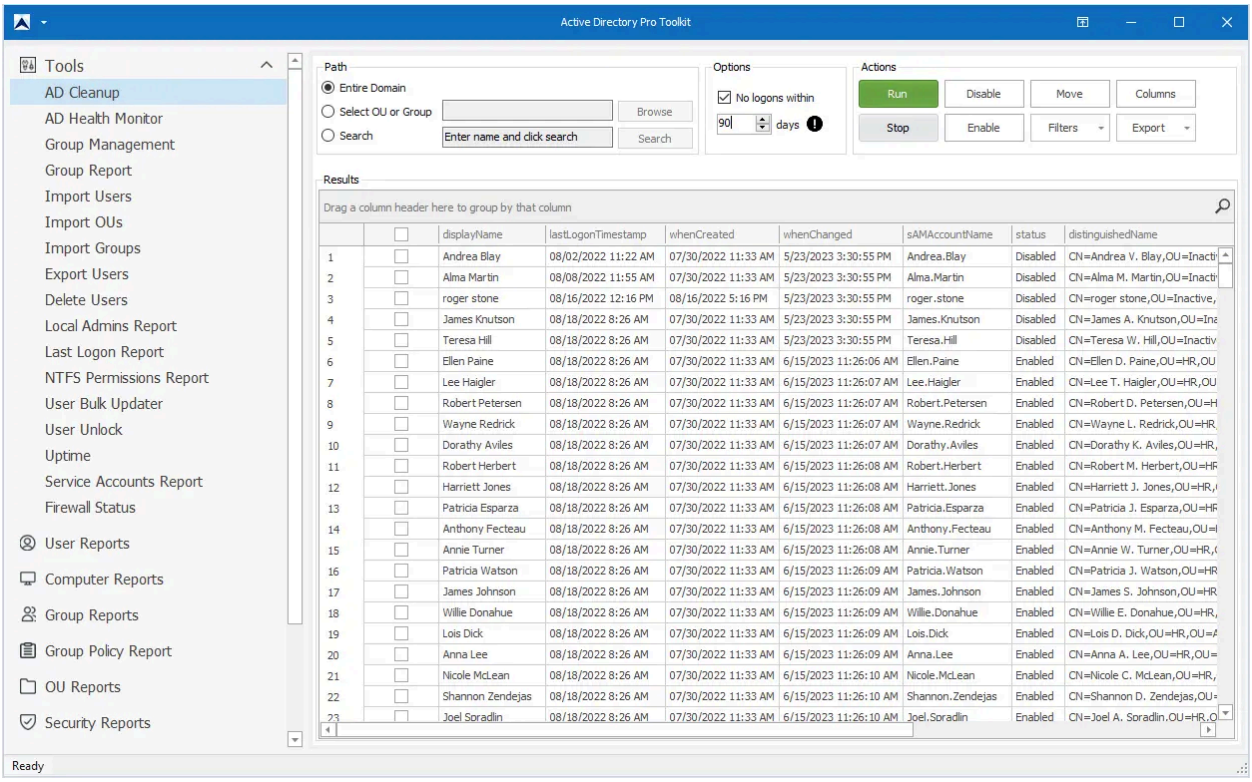
This project currently targets .NET Framework 4.8.

Tech Savvy Productions

3. **DCDiag:** Another command-line tool included with Windows Server, DCDiag runs a variety of tests to verify different aspects of AD health, including replication. For instance, dcdiag /test:replications checks the replication efficiency.

https://www.youtube.com/watch?v=w2Pa2ztBwlI&t=1s DCDiag How to check Domain Controller Health

https://activedirectorypro.com/

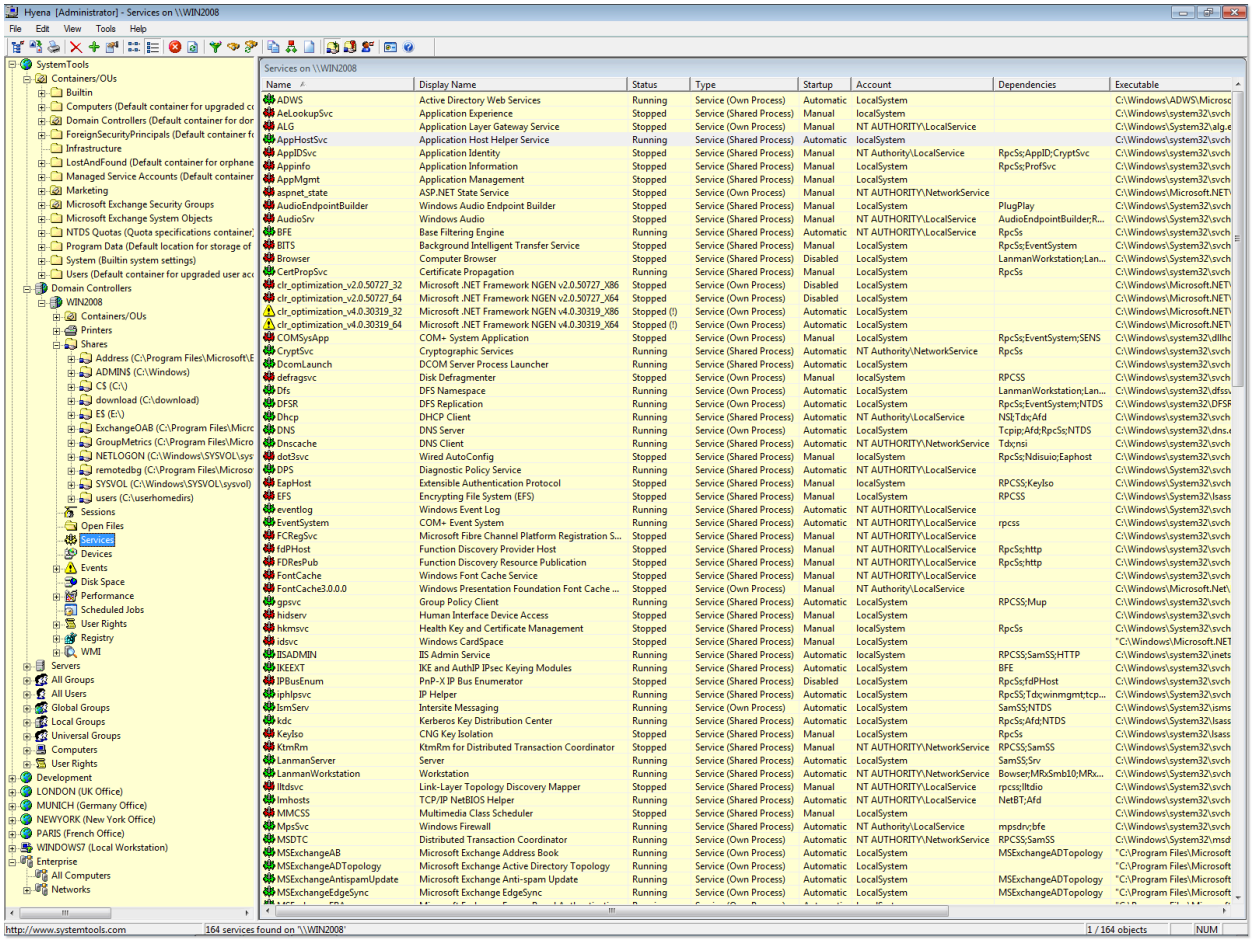


Tech Savvy Productions

https://www.systemtools.com/hyena/ Tools that have been around for many years.

Licenses	Enterprise Edition Cost per License (USD)
1-2	$ 329

Tech Savvy Productions



Permissions Analyzer for Active Directory Get instant visibility into user and group permissions

Unravel your tangled mess of permissions for Active Directory, network shares, folders, and files for users and groups with this free tool.

Key Features

- Quickly identify how a user's permissions are inherited
- Browse permissions by group or individual user
- Analyze user permissions based on group membership and permissions

https://www.solarwinds.com/free-tools/permissions-analyzer-for-active-directory?irclickid=xqQ3DVxWCxyPWJWUS5TJ82YBUkFxYkQFUX%3ALUQ0&irgwc=1&CMP=BIZ-TAD-ADP-SW_WW_X_X_PPD_FT_EN_0_0-PAFT-0_0_X_X_X_X-X

4. **ReplMon:** Though this tool was officially retired after Windows Server 2003, many administrators still find it useful for older systems. It provides a GUI for monitoring replication and diagnosing issues.
5. **AD Replication PowerShell cmdlets:** Windows Server includes several PowerShell cmdlets that can help with replication issues, such as `Get-ADReplicationFailure` and `Get-ADReplicationConnection`.
6. **Active Directory Sites and Services:** This is a Microsoft Management Console (MMC) snap-in that you can use to check replication topology and manually force replication between domain controllers.
7. **Event Viewer:** Windows logs many events, including replication errors, in the Event Viewer. Understanding how to filter and interpret these logs can be extremely useful in diagnosing replication issues.

Remember to use these utilities in your test environment first to understand their results and potential impact on your AD infrastructure before using them in a production environment.

Use DcDiag with PowerShell to check domain controller health

In the earlier 2000s, a command-line utility that returned dumb strings would have been OK, but today we have smart objects with PowerShell! Let's build a wrapper around the dcdiag tool with PowerShell to bring this handy tool into the current times.

Tech Savvy Productions

If you run dcdiag by itself, you can see that it returns some useful information, but this is just loose text. Dcdiag does not return objects we can parse and manipulate as we can in PowerShell.

```
PS C:\windows\system32> dcdiag

Directory Server Diagnosis

Performing initial setup:
   Trying to find home server...
   Home Server = DC
   * Identified AD Forest.
   Done gathering initial info.

Doing initial required tests

   Testing server: Default-First-Site-Name\DC
      Starting test: Connectivity
         ......................... DC passed test Connectivity

Doing primary tests

   Testing server: Default-First-Site-Name\DC
      Starting test: Advertising
         ......................... DC passed test Advertising
```

Use DcDiag with PowerShell to check domain controller health

To wrap this tool in PowerShell, we'll need to build a function. I'll call my function Invoke-DcDiag. We can get crazy with this, but to keep it simple, I want to pass a single domain controller (DC) to dcdiag, so I'll create this function with a parameter called DomainController.

```
function Invoke-DcDiag {
param(
[Parameter(Mandatory)]
[ValidateNotNullOrEmpty()]
[string]$DomainController
)
## Code goes here
}
```

Once I've built the skeleton, I can then add in the code. To run dcdiag against a domain controller, I need to use the /s switch, so I'll add that to my function. I

Tech Savvy Productions

need to capture the output, so I'll assign it to a variable. To mimic the code that will go into the function, I'll assign $DomainController a value of DC representing the name of the DC in my environment.

```
$DomainController = 'DC'
$result = dcdiag /s:$DomainController
```

I've now captured the output to a variable. That's great and all, but we now need to parse the output. Let's get our regex on! Before we get that far though, I need to define what the output should look like. Eyeballing our sample output, it seems like for every test that happens, dcdiag always returns one of two strings:

```
<DomainControllerName> passed test <TestName>
<DomainControllerName> failed test <TestName>
```

I want PowerShell to return an object for each test result with a TestName, TestResult, and an EntityName to represent a domain controller, NT Directory Service (NTDS) partition, or whatever else the test is running against.

We'll need to parse each line in the string. One way to do this is to use Select-String. Select-String has a parameter called Pattern that allows us to try to match a string against a regular expression. Without teaching you regex, I'll give you the answer here. The regex you need for this is '\. (.*) \b(passed|failed)\b test (.*)'.

When you run it as is, you may think that either I'm crazy or it doesn't work. One of these things is right.

```
PS> $dcDiagOutput | select-string -pattern '\. (.*) \b(passed|failed)\b test (.*)'
..... DC passed test Connectivity
..... DC passed test Advertising
..... DC passed test FrsEvent
..... DC passed test DFSREvent
..... DC passed test SysVolCheck
..... DC passed test KccEvent
..... DC passed test KnowsOfRoleHolders
```

However, PowerShell is hiding regex groups from you. Each regex group contains each of the properties we'd like the resulting output to have. We reference these

Tech Savvy Productions

groups by looking at the various indexes in the Groups array that the Matches property contains that Select-String returns.

You can see below that I'm looping over each match, assigning each group value to an object property we'd like, and then returning the object.

```
$result | select-string -pattern '\. (.*) \b(passed|failed)\b test (.*)' | foreach {
$obj = @{
TestName = $_.Matches.Groups[3].Value
TestResult = $_.Matches.Groups[2].Value
Entity = $_.Matches.Groups[1].Value
}
[pscustomobject]$obj
}
```

Bringing this code all back into our function makes the function look like this:

```
function Invoke-DcDiag {
param(
[Parameter(Mandatory)]
[ValidateNotNullOrEmpty()]
[string]$DomainController
)
$result = dcdiag /s:$DomainController
$result | select-string -pattern '\. (.*) \b(passed|failed)\b test (.*)' | foreach {
$obj = @{
TestName = $_.Matches.Groups[3].Value
TestResult = $_.Matches.Groups[2].Value
Entity = $_.Matches.Groups[1].Value
}
[pscustomobject]$obj
}
}
```

At this time, we bring this function into our session, run it via Invoke-DcDiag -DomainController DC, and voila, dcdiag for the modern age!

Subscribe to 4sysops newsletter!

Tech Savvy Productions

```
PS> Invoke-DcDiag -DomainController DC
TestResult Entity TestName
---------- ------ --------
passed DC Connectivity
passed DC Advertising
passed DC FrsEvent
passed DC DFSREvent
passed DC SysVolCheck
passed DC KccEvent
passed DC KnowsOfRoleHolders
passed DC MachineAccount
passed DC NCSecDesc
passed DC NetLogons
passed DC ObjectsReplicated
passed DC Replications
passed DC RidManager
passed DC Services
passed DC SystemLog
passed DC VerifyReferences
passed ForestDnsZones CheckSDRefDom
passed DomainDnsZones CheckSDRefDom
passed Schema CheckSDRefDom
passed Schema CrossRefValidation
passed Configuration CheckSDRefDom
passed Configuration CrossRefValidation
passed techsnips CheckSDRefDom
passed techsnips CrossRefValidation
passed techsnips.local LocatorCheck
passed techsnips.local Intersite
```

Helpful tests to run against AD

- dcdiag /v /c /d /e /s: > c:\dcdiag.txt
- ipconfig /all (from all DCs and DNS servers)

Tech Savvy Productions

- repadmin /showrepl (from each DC)
- repadmin /replsum
- dcdiag /test:dns /s: /dnsbasic
- repadmin /syncall /aped
- Ping each DC by name and verify that the name resolves to the correct IP address.
- Use nslookup to test DNS across different DCs.
- Use tracert to test the routes between servers.
- repadmin /bind servername - Can the DCs bind to each other?
- You can run command below to check AD Topology.
  repadmin /showrepl * /csv >c:\repsum.csv

Dcdiag: How to Check Domain Controller Health

Last Updated: December 30, 2022 by Robert Allen
https://activedirectorypro.com/dcdiag-check-domain-controller-health/

In this tutorial, I'll show you how to use the Dcdiag command line utility to perform a domain controller health Check.

I'll also show you how to use Dcdiag to test DNS.

Be sure to check the examples section below and I'll show you the output of a normal domain controller and one that has issues.

Let's dive right in.

Tech Savvy Productions

What is Dcdiag

Dcdiag is a Microsoft Windows command line utility that can analyze the state of domain controllers in a forest or enterprise. You can choose to analyze a single domain controller or all DC's in a forest.

You might be thinking, how well does a command line utility really do at testing and finding issues with domain controllers?

Well, it actually does a pretty good job! I was surprised to find out how many different tests this command actually did.

I counted a total of 30 different tests, and some tests had multiple tests.

So what does Dcdiag actually do? What kinds of tests does it perform?

You can view a list of tests in the command help menu, run dcdiag /? to see the help menu.

I'll have to admit the descriptions from the help command are not that helpful. I came across a good blog post by Ned Pyle called What does DCDiag actually do that explains each test in more details. If you are interested in learning exactly what the Dcdiag command does then I recommend you read that post.

How to install Dcdiag

There is a really good chance that you already have the Dcdiag tool installed.

If you have the AD DS role installed then Dcdiag is already installed. If you have the Remote Server Administration Tools (RSAT) tools installed then you already have Dcdiag installed.

If you want to install it on a system that doesn't have either of those then you can download the tool from here and install it a Windows XP or later system (hopefully your still not running XP).

How to use Dcdiag (examples)

DCDiag is very easy to use.

Tech Savvy Productions

There are several different command line switches that can be used with Dcdiag, to view them all just use this command dcdiag /?

Here is an overview of the switches I'll be using in the examples below.

/s: dcname switch is used to run Dcdiag against a remote server

/v: switch prints more detailed information about each test

/c: switch means comprehensive, this will run all tests including the dns test.

/q: switch will only print errors. This is useful as dcdiag can display a lot of information, if you want to see just the errors then use this switch.

/f: switch is used to redirect the results to a file.

TIP: When running dcdiag it will probably report some errors but this doesn't necessarily mean you have issues with your domain controllers. For example, the command will query the system logs on the DC and display errors logs, but they could be errors from a computer or another server. Again this may not be a DC issue. You will just have to review and determine if it's related or not.

.

Example 1: Use /s to run against a remote server

This will run all the DC tests against the remote server DC1.

dcdiag /s:DC1

To run this on a local server just leave off the /s:servername

Example 2: Using /v to display more details

The verbose switch will display additional details. It does the same tests as the first example it just displays more details about each test.

dcdiag /s:DC1 /v

Example 3: Using /f to save to a log file

Tech Savvy Productions

By using the /f you can save the results to a text file. This is great for logging the results and reviewing them at a later time.

dcdiag /s:DC1 /f:c:\it\dcdiag_test.txt

Example 4: Using /a to run against all domain controllers

If you have multiple domain controllers and want to test them all at once, then use this command. If you have many domain controllers this will be a lot of information displayed, this is where using the /f option would come in handy.

dcdiag /s:DC1 /a

Example 5: Use /q to only display the errors

DCDiag can display a lot of information, to remove the noise and only display the errors use this command.

dcdiag /s:DC1 /q

Example 6: Use multiple switches (My favorite)

Here are the commands I like to run. It will run all tests, displays all the details, and outputs its to a file.

dcdiag /s:DC1 /c /v /f:c:\it\dcdiag_test.txt

Now that you know how to test your domain controllers I'll move on to testing DNS. By the way, if you use the last example that includes the /c that switch will run all tests including DNS.

How to test DNS with DCDiag

The DNS diagnostics also includes several tests.

Here are the tests that are performed, by default all tests except external name resolution run.

/DnsBasic (basic tests, can't be skipped)
/DnsForwarders (forwarders and root hints tests)

Tech Savvy Productions

/DnsDelegation (delegations tests)
/DnsDynamicUpdate (dynamic update tests)
/DnsRecordRegistration (records registration tests)
/DnsResolveExtName (external name resolution test)
/DnsAll (includes all tests above)
/DnsInternetName: <internet name> (for test /DnsResolveExtName)

Use the following command to run a DNS test

dcdiag /s:dc1 /test:dns

Example: Dcdiag results from a normal domain controller

Directory Server Diagnosis
Performing initial setup:
** Identified AD Forest.*
Done gathering initial info.

Doing initial required tests

Testing server: Default-First-Site-Name\DC1
Starting test: Connectivity
..... DC1 passed test Connectivity

Doing primary tests

Testing server: Default-First-Site-Name\DC1
Starting test: Advertising
..... DC1 passed test Advertising
Starting test: FrsEvent
..... DC1 passed test FrsEvent
Starting test: DFSREvent
..... DC1 passed test DFSREvent
Starting test: SysVolCheck
..... DC1 passed test SysVolCheck
Starting test: KccEvent
..... DC1 passed test KccEvent
Starting test: KnowsOfRoleHolders
..... DC1 passed test KnowsOfRoleHolders
Starting test: MachineAccount
..... DC1 passed test MachineAccount
Starting test: NCSecDesc
..... DC1 passed test NCSecDesc
Starting test: NetLogons
..... DC1 passed test NetLogons

Tech Savvy Productions

Starting test: ObjectsReplicated
..... DC1 passed test ObjectsReplicated
Starting test: Replications
..... DC1 passed test Replications
Starting test: RidManager
..... DC1 passed test RidManager
Starting test: Services
..... DC1 passed test Services
Starting test: SystemLog ..... DC1 passed test SystemLog Starting test: VerifyReferences
..... DC1 passed test VerifyReferences

Running partition tests on : ForestDnsZones
Starting test: CheckSDRefDom
..... ForestDnsZones passed test CheckSDRefDom
Starting test: CrossRefValidation
..... ForestDnsZones passed test CrossRefValidation

Running partition tests on : DomainDnsZones
Starting test: CheckSDRefDom
..... DomainDnsZones passed test CheckSDRefDom
Starting test: CrossRefValidation
..... DomainDnsZones passed test CrossRefValidation

Running partition tests on : Schema
Starting test: CheckSDRefDom
..... Schema passed test CheckSDRefDom
Starting test: CrossRefValidation
..... Schema passed test CrossRefValidation

Running partition tests on : Configuration
Starting test: CheckSDRefDom
..... Configuration passed test CheckSDRefDom
Starting test: CrossRefValidation
..... Configuration passed test CrossRefValidation

Running partition tests on : ad
Starting test: CheckSDRefDom
..... ad passed test CheckSDRefDom
Starting test: CrossRefValidation
..... ad passed test CrossRefValidation

Running enterprise tests on : ad.activedirectorypro.com
Starting test: LocatorCheck
..... ad.activedirectorypro.com passed test LocatorCheck
Starting test: Intersite
..... ad.activedirectorypro.com passed test Intersite

Tech Savvy Productions

Example: Failed kdc service and NETLOGON Service

Here is the dcdiag output when the kdc and NETLOGON service is stopped. I used the /q switch to only display the errors

C:\Users\rallen>dcdiag /s:dc2 /q
Fatal Error:DsGetDcName (DC2) call failed, error 1717
The Locator could not find the server.
..... DC2 failed test Advertising
kdc Service is stopped on [DC2]
NETLOGON Service is stopped on [DC2]
..... DC2 failed test Services
Warning: DcGetDcName(GC_SERVER_REQUIRED) call failed, error 1717
A Global Catalog Server could not be located - All GC's are down.
Warning: DcGetDcName(PDC_REQUIRED) call failed, error 1717
A Primary Domain Controller could not be located.
The server holding the PDC role is down.
Warning: DcGetDcName(TIME_SERVER) call failed, error 1717
A Time Server could not be located.
The server holding the PDC role is down.
Warning: DcGetDcName(GOOD_TIME_SERVER_PREFERRED) call failed, error 1717
A Good Time Server could not be located.
Warning: DcGetDcName(KDC_REQUIRED) call failed, error 1717
A KDC could not be located - All the KDCs are down.
..... ad.activedirectorypro.com failed test LocatorCheck

DCDiag is a simple yet very powerful tool to check and diagnose domain controllers. I highly recommend that you become familiar with this tool and run it in your environment from time to time.

Tech Savvy Productions

What does DCDIAG actually… do?

https://learn.microsoft.com/en-us/archive/blogs/askds/what-does-dcdiag-actually-do

Hi folks, Ned here again. I recently wrote a KB article about some expected DCDIAG.EXE behaviors. This required reviewing **DCDIAG.EXE** as I wasn't finding anything deep in TechNet about the "Services" test that had my interest. By the time I was done, I had found a dozen other test behaviors I had never known existed.

While we have documented the version of DCDIAG that shipped with Windows Server 2008 – sometimes with excellent specificity, like Justin Hall's article about

Tech Savvy Productions

the DNS tests – mostly it's a black box and you only find out what it tests when the test fails. Oh, we have help of course: just run **DCDIAG /?** to see it. But it's help written by developers. Meaning you get wording like this:

Advertising
Checks whether each DSA is advertising itself, and whether it is advertising itself as having the capabilities of a DSA.

So, it checks each DSA (whatever that is) to see if it's advertising (whatever that means). The use of an undefined acronym is an especially nice touch, as even within Microsoft, DSA could mean:

- Digital Signature Algorithm
- Dynamic Structure Array
- Distributed Systems Architecture
- Directory System Agent <-- *it's this one*
- Directory Service Agent <-- *except when it's this one*

Naturally, this brings out my particular brand of OCD. What follows is the result of my compulsion to understand. I'm not documenting every last switch in DCDIAG, just the tests. I am only documenting Windows Server 2008 R2 SP1 behavior – I have no idea where the source code is for the ancient Support Tools version of DCDIAG and you aren't paying me enough here to find it :-). The Windows Server 2008 RTM through Windows Server 2008 R2 SP1 versions are nearly identical except for bug fixes:

KB2401600 The Dcdiag.exe VerifyReferences test fails on an RODC that is running Windows Server 2008 R2
https://support.microsoft.com/default.aspx?scid=kb;en-US;2401600

KB979294 The Dcdiag.exe tool takes a long time to run in Windows Server 2008 R2 and in Windows 7
https://support.microsoft.com/default.aspx?scid=kb;EN-US;979294

KB978387 FIX: The connectivity test that is run by the Dcdiag.exe tool fails together with error code 0x621
https://support.microsoft.com/default.aspx?scid=kb;EN-US;978387

Everything I describe below you can discover and confirm yourself with careful examination of network captures and logging, to include the public functions being used – but why walk when you can ride? Using **/v** can also provide considerable

Tech Savvy Productions

details on some tests. No internal source code is described nor do I show any special hidden functionality.

For info on all the network protocols I list out – or if you run into network errors when using DCDIAG – see Service overview and network port requirements for the Windows Server system. I went pretty link-happy in general in this post to help people using it as a reference; that way if you just look at your one little test it has all the info you need. I don't always call out name resolution being tested because it is implicit; it's also testing TCP, UDP, and IP.

Finally: this post is more of a reference than my usual lighthearted fare. Do not operate heavy machinery while reading.

Initial Required Tests

This tests general connectivity and responsiveness of a DC, to include:

- Verifying the DC can be located in DNS.
- Verifying the DC responds to ICMP pings.
- Verifying the DC allows LDAP connectivity by binding to the instance.
- Verifying the DC allows binding to the AD RPC interface using the DsBindWithCred function.

The DNS test can be satisfied out of the client cache so restarting the DNS client service locally is advisable when running DCDIAG to guarantee a full test of name resolution. For example:

Net stop "dns client" & net start "dns client" & dcdiag /test:verifyreplicas /s:DC-01

The initial tests cannot be skipped.

The initial tests use ICMP, LDAP, DNS, and RPC on the network.

Editorial note: Blocking ICMP will prevent DCDIAG from working. While blocking ICMP is highly recommended at the Internet-edge of your network, internally

Tech Savvy Productions

blocking ICMP traffic mainly just leads to administrative headaches like breaking legacy group policy, breaking black hole router detection (or leading to highly inefficient MTU sizes due to lack of a discovery option), and breaking troubleshooting tools like **ping.exe** or **tracert.exe**. It creates an illusion of security; there are a great many other easy ways for a malicious internal user to locate computers.

Advertising

This test validates that the public DsGetDcName function used by computers to locate domain controllers will correctly locate any DCs specified with in the command line with the **/s**, **/a**, or **/e** parameter. It checks that the server successfully reports itself with DS_Flags for:

- DC
- LDAP server
- Writable or Read-Only DC
- KDC
- Time Server
- GC or not (and if claiming to be a GC, if the is GC ready to respond to requests)

Note that "advertising" is not the same as "working". For instance, if the KDC service is stopped the **Advertising** test will fail since the flag returned from DsGetDcName will not include KDC. But if port 88 over TCP and UDP are blocked on a firewall, the **Advertising** test will pass – even though the KDC is not going to be able to answer requests for Kerberos tickets.

This test is done using RPC over SMB (using a Netlogon named pipe) to the DC plus LDAP to locate the DCs site information.

CheckSDRefDom

This test validates that your application partition cross reference objects (located in "*cn=partitions,cn=configuration,dc=<forest root domain>* ") contain the correct domain names in their msDS-SDReferenceDomain attributes. The test uses LDAP.

I find no history of anyone ever seeing the error message that can be displayed here.

The test uses LDAP.

Tech Savvy Productions

CheckSecurityError

This test does a variety of checks around the security components of a DC like Kerberos. For it to be more specifically useful you should provide **/replsource: *<some partner DC>*** as the default checks are not as comprehensive.

This test:

- Validates that at least one KDC is online for each domain and they are reachable (first in the same site, then anywhere in the domain if that fails)
- Checks if packet fragmentation of Kerberos over UDP might be an issue based on current MTU size by sending non-fragmenting ICMP packets
- Checks if the DC's computer account exists in AD, if it's within the default "Domain Controllers" OU, if it has the correct UserAccountControl flags for DCs, that the correct ServerReference attributes are set, and if the minimum Service Principal Names are set
- Validates that the DCs computer object has replicated to other DCs
- Validates that there are no replication or KCC connection issues for connected partners by querying the function DsReplicaGetInfo to get any security-related errors

When the **/replsource** is added, a few more tests happen. The partner is checked for all of the above also, then:

- Time skew is calculated between the servers to verify it is less than 300 seconds for Kerberos. It does *not* check the Kerberos policy to see if allowed skew has been modified
- Permissions are checked on all the naming contexts (such as Schema, Configuration, etc.) on the source DC to validate that replication and connectivity will work between DCs
- Connectivity is checked to validate that the user running DCDIAG (and therefore in theory, all other users) can connect to and read the SYSVOL and NETLOGON shares without any security errors. It also checks IPC$, but inability to connect there would have broken many earlier tests
- The "Access this computer from the network" privilege on the DC is checked to verify it is held by Administrators, Authenticated Users, and Everyone groups
- The DC's computer object is checked to ensure it is the latest version on the DCs. This is done to prove replication convergence since a very stale DC

Tech Savvy Productions

might lead to security issues for users, problems with the DCs own computer account password, or secure channels to other servers. It checks versions, USNs, originating servers, and timestamps

These tests are performed using LDAP, RPC, RPC over SMB, and ICMP.

Connectivity

No matter what you specify for tests, this always runs as part of **Initial Required Tests**.

CrossRefValidation

This test retrieves a list of naming contexts (located in "*cn=partitions,cn=configuration,dc=<forest root domain>* ") with their cross references and then validates them, similar to the CheckSDRefDom test above. It is looking at the nCName , dnsRoot, nETBIOSName, and systemFlags attributes to:

- Make sure the names or DNs are not invalid or null
- Confirm DNs are not otherwise mangled with CNF or 0ADEL (which happens during Conflict or Deletion operations)
- Ensure the systemFlags are correct for that object
- Call out any empty (orphaned) replica sets

The test uses LDAP.

CutoffServers

Tests the AD replication topology to ensure there are no DCs without working connection objects between partners. Any servers that cannot replicate inbound or outbound from any DCs are considered "cut off". It uses the function DsReplicaSyncAll to do this which means this "test" actually triggers replication on the DCs so use with caution if you are the owner of crud WAN links that you keep clean with schedules, and certainly consider this before using **/e**.

This test is rather misleading in its help description; if it cannot contact a server that is actually unavailable to LDAP on the network then it gives no error or test results, even if the **/v** parameter is specified. You have to notice that there is no series of "analyzing the alive system replication topology" or "performing upstream (of target) analysis" messages being printed for a cutoff server. However, the Connectivity test will fail if the server is unreachable so it's a wash.

Tech Savvy Productions

The test uses RPC.

DcPromo

The DCpromo test is one of the two oddballs in DCDIAG (the other is 'DNS'). It is designed to test how well a DCPROMO *would* proceed if you were to run it on the server where DCDIAG is launched. It also has a number of required switches for each kind of promotion operation. All of the tests are against the server specified first in the client DNS settings. It tests:

- If at least one network adapter has a primary DNS server set
- If you would have a disjoint namespace based on the DNS suffix
- That the proposed authoritative DNS zone can be contacted
- If dynamic DNS updates are possible for the server's A record. It checks both the setting on the authoritative DNS zone as well as the client registry configuration of DnsUpdateOnAllAdapters and DisableDynamicUpdate
- If an LDAP DClocator record (i.e. "*_ldap ._tcp.dc._msdcs.<domain>*") is returned when querying for existing forests

The test uses DNS on the network.

DNS

This series of enterprise-wide DNS tests are already well documented here:

https://technet.microsoft.com/en-us/library/cc731968(WS.10).aspx

The tests use DNS, RPC, and WMI protocols.

FrsEvent

This test validates the File Replication Service's health by reading (and printing, if using **/v**) FRS event log warning and error entries from the past 24 hours. It's possible this service won't be running or installed on Windows Server 2008 or later if SYSVOL has been migrated to DFSR. On Windows Server 2008, some events may be misleading as they may refer to custom replica sets and not necessarily SYSVOL; on Windows Server 2008 R2, however, FRS can be used for SYSVOL only.

By default, remote connections to the event log are disabled by the Windows Server 2008/R2 firewall rules so this test will fail. KB2512643 covers enabling those rules to allow the test to succeed.

The test uses RPC, specifically with the EventLog Remoting Protocol.

Tech Savvy Productions

DFSREvent

This test validates the Distributed File System Replication service's health by reading (and printing, if using **/v**) DFSR event log warning and error entries from the past 24 hours. It's possible this service won't be running or installed on Windows Server 2008 if SYSVOL is still using FRS; on Windows Server 2008 R2 the service is always present on DCs. While this ostensibly tests DFSR-enabled SYSVOL, any errors within custom DFSR replication groups would also appear here, naturally.

By default, remote connections to the event log are disabled by the Windows Server 2008/R2 firewall rules so this test will fail. KB2512643 covers enabling those rules to allow the test to succeed.

The test uses RPC, specifically with the EventLog Remoting Protocol.

SysVolCheck

This test reads the DCs Netlogon SysvolReady registry key to validate that SYSVOL is being advertised:

HKEY_Local_Machine\System\CurrentControlSet\Services\Netlogon\Parameters
SysvolReady=1

The value name has to exist with a value of 1 to pass the test. This test will work with either FRS or DFSR-replicated SYSVOLs. It doesn't check if the SYSVOL and NELOGON shares are actually accessible, though (**CheckSecurityError** does that).

The test uses RPC over SMB (through a named pipe to WinReg).

LocatorCheck

This test validates that DCLocator queries return the five "capabilities" that any DC must know of to operate correctly.

- GC
- PDCE
- Time Server
- Preferred Time Server
- KDC

If not hosting one, the DC will refer to another DC that can satisfy the request; this means that you must carefully examine this under **/v** to make sure a server you

Tech Savvy Productions

thought was supposed to be holding a capability actually is correctly returned. If no DC answers or if the queries return errors then the test will fail.

The tests use RPC over SMB with the standard DsGetDcName DCLocator queries.

Intersite

This test uses Directory Replication Service (DRS) functions to check for conditions that would prevent inter-site AD replication within a specific site or all sites:

- Locates and connect to the Intersite Topology Generators (ISTG)
- Locates and connect to the bridgehead servers
- Reports back any replication failures after triggering a replication
- Validates that all DCs within sites with inbound connections to this site are available
- Checks the KCC values for "IntersiteFailuresAllowed" and "MaxFailureTimeForIntersiteLink" overrides within the registry key:

KEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NTDS\Parameters

You must be careful with this test's command-line arguments and always provide **/a** or **/e**. Not providing a site means that the test runs but skips actually testing anything (you can see this under **/v**).

All tests use RPC over the network to test the replication aspects and will make registry connections (RPC over SMB to WinReg) to check for those NTDS settings override entries. LDAP is also used to locate connection info.

KccEvent

This test queries the Knowledge Consistency Checker on a DC for KCC errors and warnings generated in the Directory Services event log during the last 15 minutes. This 15 minute threshold is irrespective of the Repl topology update period (secs) registry value on the DC.

By default, remote connections to the event log are disabled by the Windows Server 2008/R2 firewall rules so this test will fail. KB2512643 covers enabling those rules to allow the test to succeed.

The test uses RPC, specifically with the EventLog Remoting Protocol.

Tech Savvy Productions

KnowsOfRoleHolders

This test returns the DC's knowledge of the five Flexible Single Master Operation (FSMO) roles. The test does not inherently check all DCs knowledge for consistency, but using the **/e** parameter would provide data sufficient to allow comparison.

The test uses RPC to return DSListRoles within the Directory Replication Service (DRS) functions.

MachineAccount

This test checks if:

- The DC's computer account exists in AD
- It's within the Domain Controllers OU
- It has the correct UserAccountControl flags for DCs
- The correct ServerReference attributes are set
- The minimum Service Principal Names are set. For those paying close attention, this is identical to one test aspect of **CheckSecurityError**; this is because they use the same internal test

This test also mentions two repair options:

- **/RecreateMachineAccount** will recreate a missing DC computer object. This is ***not a recommended fix*** as it does not recreate any child objects of a DC, such as FRS and DFSR subscriptions. The best practice is to use a valid SystemState backup to authoritatively restore the DC's deleted object and child objects. If you do use this **/RecreateMachineAccount** option then the DC should then be gracefully demoted and promoted to repair all the missing relationships
- **/FixMachineAccount** will add the UserAccountControl flags to a DCs computer object for "TRUSTED_FOR_DELEGATION" and "SERVER_TRUST_ACCOUNT". It's safe to use as a DC missing those bit flags will not function and it does not remove other bit flags present. Using this repair option is preferred over trying to set these flags yourself through **ADSIEDIT** or other LDAP editors

This test uses LDAP and RPC over SMB.

Tech Savvy Productions

NCSecDesc

This test checks permissions on all the naming contexts (such as Schema, Configuration, etc.) on the source DC to validate that replication and connectivity will work between DCs. It makes sure that "Enterprise Domain Controllers" and "Administrators" groups have the correct minimum permissions. This is the same performed test within **CheckSecurityError**.

The test uses LDAP.

NetLogons

This test is designed to:

- Validate that the user running DCDIAG (and therefore in theory, all other users) can connect to and read the SYSVOL and NETLOGON shares without any security errors. It also checks IPC$, but inability to connect there would have broken many earlier tests
- Verify that the Administrators, Authenticated Users, and Everyone group have the "access this computer from the network" privilege on the DC. If not, you'd see a ton of other errors here though, naturally

Both of these tests are also performed by **CheckSecurityError**.

The tests use SMB and RPC over SMB (through named pipes).

ObjectsReplicated

This test verifies that replication of a few key objects and attributes has occurred and displays up-to-dateness info if replication is stale. By default the two objects validated are:

- The "*CN=NTDS Settings*" object of each DC exists up to date on all other DCs.
- The "*CN=<DC name>* " object of each DC exists up to date on all other DCs.

This test is not valuable unless run with **/e** or **/a** as it just asks the DC about itself when those are not specified. Using /v will give more details on objects thought to be stale based on version.

You can also specify arbitrary objects to test with **/objectdn /n**, which can be useful after creating a "canary" object to validate replication.

Tech Savvy Productions

The tests are done using RPC with Directory Replication Service (DRS) functions.

OutboundSecureChannels

This test is designed to check external trusts. It will not run by default and *will fail even when provided correct **/testdomain** parameters, validating the secure channel with **NLTEST.EXE**, and using a working external trust*. It does state that the secure channel is valid but then mistakenly reports that there are no working trust objects. I'll update this post when I find out more. ***This test should not be used.***

RegisterLocatorDnsCheck

Validates many of the same aspects as the **Dcpromo** test. It requires the **/dnsdomain** switch to specify a domain that would be the target of registration; this can be a different domain than the current primary one. It specifically verifies:

- If at least one network adapter has a primary DNS server set.
- If you would have a disjoint namespace based on the DNS suffix
- That the proposed authoritative DNS zone can be contacted
- If dynamic DNS updates are possible for the server's A record. It checks both the setting on the authoritative DNS zone as well as the client registry configuration of DnsUpdateOnAllAdapters and DisableDynamicUpdate
- If an LDAP DClocator record (i.e. "*_ldap ._tcp.dc._msdcs.<domain>*") is returned when querying for existing forests
- That the authoritative DNS zone can be contacted

The test uses DNS on the network.

Replications

This test checks all AD replication connection objects for all naming contexts on specified DC(s) to see:

- If the last replication attempted was successful or returned an error
- If replication is disabled
- If replication latency is more than 12 hours

The tests are done with LDAP and RPC using DsReplicaGetInfo.

RidManager

This test validates that the RID Master FSMO role holder:

Tech Savvy Productions

- Can be located and contacted through a DsBind
- Has valid RID pool values

This role must be online and accessible for DCs to be able to create security principals (users, computers, and groups) as well as for further DCs to be promoted within a domain.

The test uses LDAP and RPC.

Services

This test validates that various AD-dependent services are running, accessible, and set to specific start types:

- RPCSS - Start Automatically – Runs in Shared Process
- EVENTSYSTEM - Start Automatically - Runs in Shared Process
- DNSCACHE - Start Automatically - Runs in Shared Process
- NTFRS - Start Automatically - Runs in Own Process *(if domain functional level is less than Windows Server 2008. Does not trigger on SYSVOL being replicated by FRS)*
- ISMSERV - Start Automatically - Runs in Shared Process
- KDC - Start Automatically - Runs in Shared Process
- SAMSS - Start Automatically - Runs in Shared Process
- SERVER - Start Automatically - Runs in Shared Process
- WORKSTATION - Start Automatically - Runs in Shared Process
- W32TIME - Start Manually or Automatically - Runs in Shared Process
- NETLOGON - Start Automatically - Runs in Shared Process

(If target is Windows Server 2008 or later)

- NTDS - Start Automatically - Runs in Shared Process
- DFSR - Start Automatically - Runs in Own Process *(if domain functional level is Windows Server 2008 or greater. Does not trigger on SYSVOL being replicated by DFSR)*

(If using SMTP-based AD replication)

- IISADMIN - Start Automatically - Runs in Shared Process
- SMTPSVC - Start Automatically - Runs in Shared Process

Tech Savvy Productions

These are the “real” service names listed in HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services. If this test is specified when targeting Windows Server 2003 DCs it is expected to fail on RpcSs. See KB2512643.

The test uses RPC and the Service Control Manager remote protocol.

SystemLog

This test validates the System Event Log’s health by reading and printing entries from the past 60 minutes (stopping at computer startup timestamp if less than 60 minutes). Errors and warnings will be printed, with no evaluation done of them being expected or not – this is left to the DCDIAG user.

By default, remote connections to the event log are disabled by the Windows Server 2008/R2 firewall rules so this test will fail. KB2512643 covers enabling those rules to allow the test to succeed.

The test uses RPC, specifically with the EventLog Remoting Protocol.

Topology

This test checks that a server has a fully-connected AD replication topology. This test must be explicitly run. It checks:

- If automatic intra-site topology generation is disabled
- If automatic inter-site topology generation is disabled
- For disconnected topologies (i.e. missing connection objects), both upstream and downstream from each reference DC

The test uses DsReplicaSyncAll with the flag of DS_REPSYNCALL_DO_NOT_SYNC. Meaning that the test analyzes and validates replication topology without actually replicating changes. The test does not validate the availability of replication partners – having a partner offline will not cause failures in this test. This does not test if the schedule is completely closed, preventing replication; to see those active replication results, use tests **Replications** or **CutoffServers**.

The test uses RPC and LDAP.

VerifyEnterpriseReferences

This test verifies computer reference attributes for all DCs, including:

Tech Savvy Productions

- **ServerReference** attribute correct for a DC on *cn=<DC name>,cn=<site>,cn=sites,cn=configuration,dc=<domain>*
- **ServerReferenceBL** attribute correct for a DC site object on a DC on *cn=<DC Name>,ou=domain controllers,dc=<domain>*
- **frsComputerReference** attribute correct for a DC site object on *cn=domain system volume (sysvol share),cn=ntfrs subscriptions,cn=<DC Name>,ou=domain controllers,DC=<domain>*
- **frsComputerReferenceBL** attribute correct for a DC object on *cn=<DC Name>,cn=domain system volume (sysvol share),cn=file replication service,cn=system,dc=<domain>*
- **hasMasterNCs** attribute correct for a DC on *cn=ntds settings,cn=<DC Name>,cn=<site>,cn=sites,cn=configuration,dc=<domain>*
- **nCName** attribute correct for a partition at *cn=<partition name>,cn=partitions,cn=configuration,dc=<domain>*
- **msDFSR-ComputerReference** attribute correct for a DC DFSR replication object on *cn=<DC Name>,cn=topology,cn=domain system volume,cn=dfsr-blobalsettings,cn=system,dc=<domain>*
- **msDFSR-ComputerReferenceBL** attribute correct for a DC site object on a DC on *cn=<DC Name>,ou=domain controllers,dc=<domain>*

Note that the two DFSR tests are only performed if domain functional level is Windows Server 2008 or higher. This means there will be an *expected* failure if DFSR has not been migrated to SYSVOL as the test does not actually care if FRS is still in use.

The test uses LDAP. The DCS are not all individually contacted, only the specified DCs are contacted.

VerifyReferences

This test verifies computer reference attributes for a single DC, including:

- **ServerReference** attribute correct for a DC on *cn=<DC name>,cn=<site>,cn=sites,cn=configuration,dc=<domain>*
- **ServerReferenceBL** attribute correct for a DC site object on a DC on *cn=<DC Name>,ou=domain controllers,dc=<domain>*
- **frsComputerReference** attribute correct for a DC site object on *cn=domain system volume (sysvol share),cn=ntfrs subscriptions,cn=<DC Name>,ou=domain controllers,DC=<domain>*

Tech Savvy Productions

- **frsComputerReferenceBL** attribute correct for a DC object on *cn=<DC Name>,cn=domain system volume (sysvol share),cn=file replication service,cn=system,dc=<domain>*
- **msDFSR-ComputerReference** attribute correct for a DC DFSR replication object on *cn=<DC Name>,cn=topology,cn=domain system volume,cn=dfsr-blobalsettings,cn=system,dc=<domain>*
- **msDFSR-ComputerReferenceBL** attribute correct for a DC site object on a DC on *cn=<DC Name>,ou=domain controllers,dc=<domain>*

This is similar to the **VerifyEnterpriseRefrences** test except that it does not check partition cross references or all other DC objects.

The test uses LDAP.

DCDIAG.EXE /E or /A or /C expected errors

https://learn.microsoft.com/en-US/troubleshoot/windows-server/identity/dcdiag-commands-running-errors

In this article

1. Symptoms
2. Cause
3. Resolution
4. More information

This article helps fix errors that occur when you run DCDIAG.EXE /E or /A or /C commands.

Original KB number: 2512643

Symptoms

Consider the following scenario:

You administer an AD environment. There may be a mixture of Windows Server 2003, Windows Server 2003 R2, Windows Server 2008, and Windows Server 2008 R2 DCs.

When running DCDIAG.EXE /E (or /A or /C) on Windows Server 2008 or Windows Server 2008 R2 (included with the operating systems), you see the following errors against all Windows Server 2003 DCs:

Tech Savvy Productions

Starting test: Services
Invalid service type: RpcSs on DCDIAG-2003, current value
WIN32_OWN_PROCESS, expected value WIN32_SHARE_PROCESS
..... DCDIAG-2003 failed test Services

When running DCDIAG.EXE /E (or /A or /C) on Windows Server 2008 or Windows Server 2008 R2 (included with the operating systems), you see the following errors against all Win2008 and Win2008 R2 DCs:

Starting test: FrsEvent
The event log File Replication Service on server
dcdiag-2008.margiestravel.com could not be queried, error 0x6ba
"The RPC server is unavailable."
..... dcdiag-2008 failed test FrsEvent

Starting test: DFSREvent
The event log DFS Replication on server
dcdiag-2008r2.margiestravel.com could not be queried, error 0x6ba
"The RPC server is unavailable."
..... dcdiag-2008r2 failed test DFSREvent

Starting test: KccEvent
The event log Directory Service on server
dcdiag-2008.margiestravel.com could not be queried, error 0x6ba
"The RPC server is unavailable."
..... dcdiag-2008 failed test KccEvent

Starting test: SystemLog
The event log System on server
dcdiag-2008.margiestravel.com could not be queried, error 0x6ba
"The RPC server is unavailable."
..... dcdiag-2008 failed test SystemLog

When running DCDIAG.EXE /E (or /A or /C) on Windows Server 2003 (included with the Support Tools out-of-band install):

- No errors are logged for any of the DCs, regardless of OS.

When running DCDIAG.EXE /C (which includes /test:verifyenterprisereferences) on Windows Server 2008 or Windows Server 2008 R2, and the Domain Functional

Tech Savvy Productions

Mode is Windows Server 2008 or higher, and FRS is still being used to replicate SYSVOL, you see the following errors:

Starting test: VerifyEnterpriseReferences
The following problems were found while verifying various important DN references. Note, that these problems
can be reported because of latency in replication. So follow up to resolve the following problems, only if
the same problem is reported on all DCs for a given domain or if the problem persists after replication has
had reasonable time to replicate changes.
[1] Problem: Missing Expected Value
Base Object: CN=SRV-01,OU=Domain Controllers,DC=margiestravel,DC=com
Base Object Description: "DC Account Object"
Value Object Attribute Name: msDFSR-ComputerReferenceBL
Value Object Description: "SYSVOL FRS Member Object"
Recommended Action: See Knowledge Base Article: Q312862
[2] Problem: Missing Expected Value
Base Object: CN=SRV-02,OU=Domain Controllers,DC=margiestravel,DC=com
Base Object Description: "DC Account Object"
Value Object Attribute Name: msDFSR-ComputerReferenceBL
Value Object Description: "SYSVOL FRS Member Object"
Recommended Action: See Knowledge Base Article: Q312862
LDAP Error 0x20 (32) - No Such Object.
..... SRV-01 failed test VerifyEnterpriseReferences

When running DCDIAG.EXE /C (which includes /test:outboundsecurechannels) and you specify /testdomain:<your trusted domain> on Windows Server 2008, or on Windows Server 2008 R2 you see the following errors:

Testing server: Default-First-Site-Name\SRV-01
Starting test: OutboundSecureChannels
[SRV-01] Does not have downlevel trust object for [contoso.com]
[SRV-01] Does not have uplevel trust object for [contoso.com]
[SRV-02] Does not have downlevel trust object for [contoso.com]
[SRV-02] Does not have uplevel trust object for [contoso.com]
..... SRV-01 failed test OutboundSecureChannels

Tech Savvy Productions

Cause

All of these behaviors are expected.

- The Windows Server 2008/200R2 versions of DCDIAG are designed to test RPCSS for the Windows Server 2008 shared process setting -not the previous isolated process setting used in Windows Server 2003 and older operating systems. The tool doesn't distinguish between OSs for this service.
- The Windows Server 2008/200R2 versions of DCDIAG assume that a Windows Server 2008 domain functional level means the DCs are replicating SYSVOL with DFSR.
- The Windows Server 2008/200R2 versions of DCDIAG don't correctly test trust health
- Windows Server 2008/2008 R2 doesn't allow remote connectivity to the event log based on default firewall rules.
- The Windows Server 2003 version of DCDIAG doesn't report back an error if it can't connect to the event log; it only reports if it connects and finds errors.
- The Windows Server 2003 version of DCDIAG doesn't test the RPCSS service configuration.

Resolution

There are multiple workarounds to these issues:

- Ignore all these errors when running DCDIAG.
- To stop the event log-related errors, enable the built-in incoming firewall rules on DCs so that the event logs can be accessed remotely:

  Remote Event Log Management (NP-In)
  Remote Event Log Management (RPC)
  Remote Event Log Management (RPC-EPMAP)

  This can be done through the "Windows Firewall with Advanced Security" snap-in (**WF.MSC**), using the firewall group policy (**Computer Configuration\ Policies\ Windows Settings\ Security Settings\ Windows Firewall with Advanced Security**), or by using NETSH.EXE ADVFIREWALL.

Tech Savvy Productions

- To stop the RPCSS service error, you can opt out of the test with /SKIP:SERVICES. There are caveats to this, see More Information. It's better to ignore this specific error altogether when it's returned from Win2003 DCs.

  It's expected and normal for this service's behavior type to be 0x10 (isolated) on Windows Server 2003 and 0x20 (shared) on Windows Server 2008 and later. Don't change it based on what DCDIAG says unless you're running the version of DCDIAG that goes with that OS. Between Windows Server 2003 and Windows Server 2008, the behavior changed for the RPC service, but there was nothing yet to share in that svchost.exe process. In Windows Server 2008 R2, the new RPCEptMapper service was added to that shared svchost process. You can see who would launch in that same process by looking for this value in the service registry keys: %systemroot%\system32\svchost.exe -k RPCSS.

  Do not change the service type on Windows Server 2003 to stop the error. There may be unexpected issues long-term as this isn't a tested configuration. These issues would be difficult to identify or trace back to this root cause. Long-term solution to the RPCSS service issue: replace the remaining Windows Server 2003 servers with a later operating system.

- To stop the OutboundSecureChannels errors, use /skip:outboundsecurechannels. The tests aren't valid and can be instead tested with NETDOM.EXE and NLTEST.EXE.

- To stop the VerifyEnterpriseReferences errors, migrate your SYSVOL from FRS to DFSR. Since you are using a native Windows Server 2008 or later domain, FRS is no longer recommended for SYSVOL replication and nothing is preventing you from replacing the deprecated FRS system. See https://technet.microsoft.com/library/dd640019.aspx.

More information

The Windows Server 2008/2008 R2 DCDIAG **Services** test checks for the following services to be running, configured with default start options, and configured with default process types:

Note

Tech Savvy Productions

These are the true service names in the registry under HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services

RPCSS - Start Automatically - Shared Process
EVENTSYSTEM - Start Automatically - Shared Process
DNSCACHE - Start Automatically - Shared Process
NTFRS - Start Automatically - Own Process
ISMSERV - Start Automatically - Shared Process
KDC - Start Automatically - Shared Process
SAMSS - Start Automatically - Shared Process
SERVER - Start Automatically - Shared Process
WORKSTATION - Start Automatically - Shared Process
W32TIME - Start Manually or Automatically - Shared Process
NETLOGON - Start Automatically - Shared Process
(If domain functional level is Windows Server 2008 or later)
DFSR - Start Automatically - Own Process
(If target is Windows Server 2008 or later)
NTDS - Start Automatically - Shared Process
(If using SMTP-based AD replication)
IISADMIN - Start Automatically - Shared Process
SMTPSVC - Start Automatically - Shared Process

Troubleshooting RPC protocol

Troubleshooting the Windows Remote Procedure Call (RPC) service involves checking several different aspects of the system. Here's a general list of steps that could be used to troubleshoot RPC problems:

1. **Check RPC Services**: The first thing to check is whether the necessary services are running. You should ensure that the RPC related services, which include "Remote Procedure Call (RPC)", "RPC Endpoint Mapper" and

Tech Savvy Productions

"DCOM Server Process Launcher", are running and their startup type is set to automatic. You can check these in the Services console (services.msc).

2. **Check Network Connectivity**: Ensure that the client machine can communicate with the server machine. You can test basic network connectivity using a tool like ping.
3. **Check Firewall Settings**: Firewalls can block RPC communications. You should verify that the firewall on both the client and server machines are configured to allow RPC communication. Windows Firewall should automatically allow RPC traffic, but if you're using a third-party firewall, you may need to manually configure it.
4. **Check DNS Resolution**: If the client machine is unable to resolve the server's hostname to its IP address, this could cause RPC errors. You can use a tool like nslookup to verify that DNS resolution is working correctly.
5. **Check RPC Server Application**: The problem might be with the specific RPC server application you're trying to connect to. Try to connect to the server application from another client, or try to connect to a different RPC server application from the same client, to help narrow down the source of the problem.
6. **Check Port Availability**: RPC typically uses port 135, among others. Check if the port is available and not blocked by any network rules or firewall settings. You can use a tool like netstat to check the port usage on a system.
7. **Check Event Viewer**: The Windows Event Viewer (eventvwr.msc) logs system events, including errors related to RPC. Checking these logs may provide more information about what's causing the RPC problem.
8. **Use Network Monitoring Tools**: Tools like Wireshark can be used to monitor the network traffic between the client and server. This could help identify any issues with the network that might be causing the RPC problem.
9. **Check System Files**: If none of the above steps resolve the issue, the problem might be due to corrupted system files. You can use a tool like sfc /scannow to check and repair system files.

Tech Savvy Productions

PowerShell Basics: How to Check Active Directory Replication Status

Download article

by Dishan Francis | Jan 30, 2019

https://argonsys.com/microsoft-cloud/library/powershell-basics-how-to-check-active-directory-replication-status/

Data Replication is crucial for healthy Active Directory Environment. There are different ways to check status of replication. In this article I am going to explain how you can check status of domain replication using PowerShell.

For a given domain controller we can find its inbound replication partners using,

Tech Savvy Productions

Get-ADReplicationPartnerMetadata -Target REBEL-SRV01.rebeladmin.com

Above command provide detail description for the given domain controller including last successful replication, replication partition, server etc.

We can list down all the inbound replication partners for given domain using,

```
PS C:\Windows\system32> Get-ADReplicationPartnerMetadata -Target win2016dc2.homelab.techsavvyproductions.com


CompressChanges                : False
ConsecutiveReplicationFailures : 0
DisableScheduledSync           : False
IgnoreChangeNotifications      : False
IntersiteTransport             :
IntersiteTransportGuid         :
IntersiteTransportType         : IP
LastChangeUsn                  : 369693
LastReplicationAttempt         : 6/23/2023 4:20:45 PM
LastReplicationResult          : 0
LastReplicationSuccess         : 6/23/2023 4:20:45 PM
Partition                      : DC=HomeLab,DC=TechSavvyProductions,DC=com
PartitionGuid                  : f78e894e-5fdb-4819-8919-f0da63fd69d9
Partner                        : CN=NTDS Settings,CN=WIN2016DC3,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Con
                                 figuration,DC=HomeLab,DC=TechSavvyProductions,DC=com
PartnerAddress                 : a0597b7a-2eed-4a03-9674-d44c6a8ac635._msdcs.HomeLab.TechSavvyProductions.com
PartnerGuid                    : a0597b7a-2eed-4a03-9674-d44c6a8ac635
PartnerInvocationId            : 013443f1-ff5b-4a6e-8ea3-1cddbc4f0eb2
PartnerType                    : Inbound
ScheduledSync                  : True
Server                         : Win2016DC2.HomeLab.TechSavvyProductions.com
SyncOnStartup                  : True
TwoWaySync                     : False
UsnFilter                      : 369693
Writable                       : True

CompressChanges                : False
ConsecutiveReplicationFailures : 0
DisableScheduledSync           : False
IgnoreChangeNotifications      : False
IntersiteTransport             :
IntersiteTransportGuid         :
IntersiteTransportType         : IP
LastChangeUsn                  : 770703
LastReplicationAttempt         : 6/23/2023 4:20:45 PM
LastReplicationResult          : 0
LastReplicationSuccess         : 6/23/2023 4:20:45 PM
Partition                      : DC=HomeLab,DC=TechSavvyProductions,DC=com
PartitionGuid                  : f78e894e-5fdb-4819-8919-f0da63fd69d9
Partner                        : CN=NTDS Settings,CN=WIN2016DC1,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Con
                                 figuration,DC=HomeLab,DC=TechSavvyProductions,DC=com
PartnerAddress                 : 28029095-585f-4851-bbd8-9a2e0f8ff03e._msdcs.HomeLab.TechSavvyProductions.com
PartnerGuid                    : 28029095-585f-4851-bbd8-9a2e0f8ff03e
PartnerInvocationId            : 28029095-585f-4851-bbd8-9a2e0f8ff03e
PartnerType                    : Inbound
ScheduledSync                  : True
Server                         : Win2016DC2.HomeLab.TechSavvyProductions.com
SyncOnStartup                  : True
TwoWaySync                     : False
UsnFilter                      : 770703
Writable                       : True



PS C:\Windows\system32>
```

Get-ADReplicationPartnerMetadata -Target "rebeladmin.com" -Scope Domain

In above command the scope is defined as the domain. this can change to forest and get list of inbound partners in the forest. The output is for default partition. If needed the partition can change using – Partition to Configuration or Schema partition. It will list down the relevant inbound partners for given partition.

Tech Savvy Productions

Associated replication failures for a site, forest, domain, domain controller can find using Get-ADReplicationFailure cmdlet.

```
Get-ADReplicationPartnerMetadata -Target "Homelab.Techsavvyproductions.com" -Scope
Domain
```

```
CompressChanges                : False
ConsecutiveReplicationFailures : 0
DisableScheduledSync           : False
IgnoreChangeNotifications      : False
IntersiteTransport             :
IntersiteTransportGuid         :
IntersiteTransportType         : IP
LastChangeUsn                  : 807327
LastReplicationAttempt         : 6/23/2023 4:20:57 PM
LastReplicationResult          : 0
LastReplicationSuccess         : 6/23/2023 4:20:57 PM
Partition                      : DC=HomeLab,DC=TechSavvyProductions,DC=com
PartitionGuid                  : f78e894e-5fdb-4819-8919-f0da63fd69d9
Partner                        : CN=NTDS
Settings,CN=WIN2016DC2,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Con
                                 figuration,DC=HomeLab,DC=TechSavvyProductions,DC=com
PartnerAddress                 :
60228dba-cbc8-4304-868a-28f53a302f46._msdcs.HomeLab.TechSavvyProductions.com
PartnerGuid                    : 60228dba-cbc8-4304-868a-28f53a302f46
PartnerInvocationId            : fccac31e-5b70-4665-88cf-0c3a08db6485
PartnerType                    : Inbound
ScheduledSync                  : True
Server                         : Win2016DC1.HomeLab.TechSavvyProductions.com
SyncOnStartup                  : True
TwoWaySync                     : False
UsnFilter                      : 807327
Writable                       : True

CompressChanges                : False
ConsecutiveReplicationFailures : 0
DisableScheduledSync           : False
IgnoreChangeNotifications      : False
IntersiteTransport             :
IntersiteTransportGuid         :
IntersiteTransportType         : IP
LastChangeUsn                  : 369708
LastReplicationAttempt         : 6/23/2023 4:20:54 PM
LastReplicationResult          : 0
LastReplicationSuccess         : 6/23/2023 4:20:54 PM
Partition                      : DC=HomeLab,DC=TechSavvyProductions,DC=com
PartitionGuid                  : f78e894e-5fdb-4819-8919-f0da63fd69d9
Partner                        : CN=NTDS
Settings,CN=WIN2016DC3,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Con
                                 figuration,DC=HomeLab,DC=TechSavvyProductions,DC=com
```


Tech Savvy Productions

```
PartnerAddress                  :
a0597b7a-2eed-4a03-9674-d44c6a8ac635._msdcs.HomeLab.TechSavvyProductions.com
PartnerGuid                     : a0597b7a-2eed-4a03-9674-d44c6a8ac635
PartnerInvocationId             : 013443f1-ff5b-4a6e-8ea3-1cddbc4f0eb2
PartnerType                     : Inbound
ScheduledSync                   : True
Server                          : Win2016DC1.HomeLab.TechSavvyProductions.com
SyncOnStartup                   : True
TwoWaySync                      : False
UsnFilter                       : 369708
Writable                        : True

CompressChanges                 : False
ConsecutiveReplicationFailures  : 0
DisableScheduledSync            : False
IgnoreChangeNotifications       : False
IntersiteTransport              :
IntersiteTransportGuid          :
IntersiteTransportType          : IP
LastChangeUsn                   : 369693
LastReplicationAttempt          : 6/23/2023 4:20:45 PM
LastReplicationResult           : 0
LastReplicationSuccess          : 6/23/2023 4:20:45 PM
Partition                       : DC=HomeLab,DC=TechSavvyProductions,DC=com
PartitionGuid                   : f78e894e-5fdb-4819-8919-f0da63fd69d9
Partner                         : CN=NTDS
Settings,CN=WIN2016DC3,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Con
                                  figuration,DC=HomeLab,DC=TechSavvyProductions,DC=com
PartnerAddress                  :
a0597b7a-2eed-4a03-9674-d44c6a8ac635._msdcs.HomeLab.TechSavvyProductions.com
PartnerGuid                     : a0597b7a-2eed-4a03-9674-d44c6a8ac635
PartnerInvocationId             : 013443f1-ff5b-4a6e-8ea3-1cddbc4f0eb2
PartnerType                     : Inbound
ScheduledSync                   : True
Server                          : Win2016DC2.HomeLab.TechSavvyProductions.com
SyncOnStartup                   : True
TwoWaySync                      : False
UsnFilter                       : 369693
Writable                        : True

CompressChanges                 : False
ConsecutiveReplicationFailures  : 0
DisableScheduledSync            : False
IgnoreChangeNotifications       : False
IntersiteTransport              :
IntersiteTransportGuid          :
IntersiteTransportType          : IP
LastChangeUsn                   : 770703
LastReplicationAttempt          : 6/23/2023 4:20:45 PM
LastReplicationResult           : 0
LastReplicationSuccess          : 6/23/2023 4:20:45 PM
Partition                       : DC=HomeLab,DC=TechSavvyProductions,DC=com
PartitionGuid                   : f78e894e-5fdb-4819-8919-f0da63fd69d9
```

```
Partner                         : CN=NTDS
Settings,CN=WIN2016DC1,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Con
                                  figuration,DC=HomeLab,DC=TechSavvyProductions,DC=com
PartnerAddress                  :
28029095-585f-4851-bbd8-9a2e0f8ff03e._msdcs.HomeLab.TechSavvyProductions.com
PartnerGuid                     : 28029095-585f-4851-bbd8-9a2e0f8ff03e
PartnerInvocationId             : 28029095-585f-4851-bbd8-9a2e0f8ff03e
PartnerType                     : Inbound
ScheduledSync                   : True
Server                          : Win2016DC2.HomeLab.TechSavvyProductions.com
SyncOnStartup                   : True
TwoWaySync                      : False
UsnFilter                       : 770703
Writable                        : True

CompressChanges                 : False
ConsecutiveReplicationFailures  : 0
DisableScheduledSync            : False
IgnoreChangeNotifications       : False
IntersiteTransport              :
IntersiteTransportGuid          :
IntersiteTransportType          : IP
LastChangeUsn                   : 807312
LastReplicationAttempt          : 6/23/2023 4:21:03 PM
LastReplicationResult           : 0
LastReplicationSuccess          : 6/23/2023 4:21:03 PM
Partition                       : DC=HomeLab,DC=TechSavvyProductions,DC=com
PartitionGuid                   : f78e894e-5fdb-4819-8919-f0da63fd69d9
Partner                         : CN=NTDS
Settings,CN=WIN2016DC2,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Con
                                  figuration,DC=HomeLab,DC=TechSavvyProductions,DC=com
PartnerAddress                  :
60228dba-cbc8-4304-868a-28f53a302f46._msdcs.HomeLab.TechSavvyProductions.com
PartnerGuid                     : 60228dba-cbc8-4304-868a-28f53a302f46
PartnerInvocationId             : fccac31e-5b70-4665-88cf-0c3a08db6485
PartnerType                     : Inbound
ScheduledSync                   : True
Server                          : win2016dc3.HomeLab.TechSavvyProductions.com
SyncOnStartup                   : True
TwoWaySync                      : False
UsnFilter                       : 807312
Writable                        : True

CompressChanges                 : False
ConsecutiveReplicationFailures  : 0
DisableScheduledSync            : False
IgnoreChangeNotifications       : False
IntersiteTransport              :
IntersiteTransportGuid          :
IntersiteTransportType          : IP
LastChangeUsn                   : 770703
LastReplicationAttempt          : 6/23/2023 4:20:36 PM
LastReplicationResult           : 0
LastReplicationSuccess          : 6/23/2023 4:20:36 PM
```


Tech Savvy Productions

```
Partition                     : DC=HomeLab,DC=TechSavvyProductions,DC=com
PartitionGuid                 : f78e894e-5fdb-4819-8919-f0da63fd69d9
Partner                       : CN=NTDS
Settings,CN=WIN2016DC1,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Con
                                figuration,DC=HomeLab,DC=TechSavvyProductions,DC=com
PartnerAddress                :
28029095-585f-4851-bbd8-9a2e0f8ff03e._msdcs.HomeLab.TechSavvyProductions.com
PartnerGuid                   : 28029095-585f-4851-bbd8-9a2e0f8ff03e
PartnerInvocationId           : 28029095-585f-4851-bbd8-9a2e0f8ff03e
PartnerType                   : Inbound
ScheduledSync                 : True
Server                        : win2016dc3.HomeLab.TechSavvyProductions.com
SyncOnStartup                 : True
TwoWaySync                    : False
UsnFilter                     : 770703
Writable                      : True
```

Get-ADReplicationFailure -Target REBEL-SRV01.rebeladmin.com

Above command will list down the replication failures for the given domain controller.

Replication failures for domain can find out using,

Get-ADReplicationFailure -Target rebeladmin.com -Scope Domain

Replication failures for forest can find out using,

```
Get-ADReplicationFailure -Target homelab.techsavvyproductions.com -scope domain


FailureCount     : 0
FailureType      : Link
FirstFailureTime : 6/11/2023 6:41:46 PM
LastError        : 1256
Partner          : CN=NTDS
Settings,CN=WIN2016DC1,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration
,DC=
                   HomeLab,DC=TechSavvyProductions,DC=com
PartnerGuid      : 28029095-585f-4851-bbd8-9a2e0f8ff03e
Server           : win2016dc3.HomeLab.TechSavvyProductions.com

FailureCount     : 0
FailureType      : Link
```

```
FirstFailureTime : 6/22/2023 8:18:08 PM
LastError        : 1256
Partner          : CN=NTDS
Settings,CN=WIN2016DC2,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration
,DC=
                   HomeLab,DC=TechSavvyProductions,DC=com
PartnerGuid      : 60228dba-cbc8-4304-868a-28f53a302f46
Server           : win2016dc3.HomeLab.TechSavvyProductions.com
```

Get-ADReplicationFailure -Target rebeladmin.com -Scope Forest

Replication failures for site can find out using,

Get-ADReplicationFailure -Target LondonSite -Scope Site

In command, LondonSite can replace using relevant site name.

Using both Get-ADReplicationPartnerMetadata and Get-ADReplicationFailure, following PowerShell script can provide report against specified domain controller.

```
## Active Directory Domain Controller Replication Status##
$domaincontroller = Read-Host 'What is your Domain Controller?'

## Define Objects ##
$report = New-Object PSObject -Property @{

ReplicationPartners = $null
LastReplication = $null
FailureCount = $null
FailureType = $null
FirstFailure = $null
}

## Replication Partners ##
$report.ReplicationPartners = (Get-ADReplicationPartnerMetadata -Target $domaincontroller).Partner
$report.LastReplication = (Get-ADReplicationPartnerMetadata -Target
$domaincontroller).LastReplicationSuccess

## Replication Failures ##
$report.FailureCount  = (Get-ADReplicationFailure -Target $domaincontroller).FailureCount
$report.FailureType = (Get-ADReplicationFailure -Target $domaincontroller).FailureType
```


Tech Savvy Productions

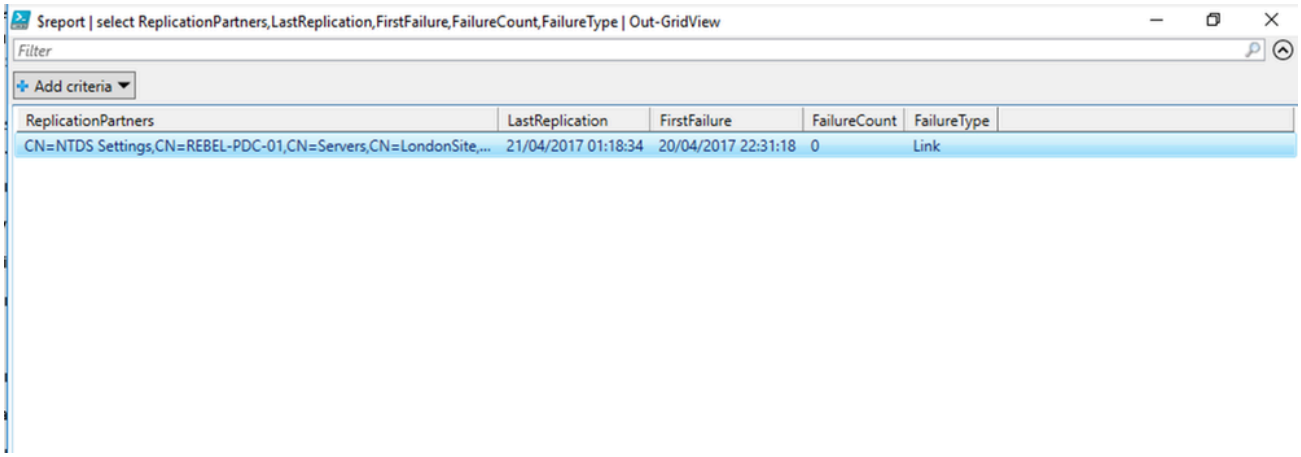
```
$report.FirstFailure = (Get-ADReplicationFailure -Target $domaincontroller).FirstFailureTime

## Format Output ##
$report | select ReplicationPartners,LastReplication,FirstFailure,FailureCount,FailureType | Out-GridView
```

In this command, it will give option for engineer to specify the Domain Controller name.

```
$domaincontroller = Read-Host 'What is your Domain Controller?'
```

Then its creates some object and map those to result of the PowerShell command outputs. Last but not least it provides a report to display a report including,

- Replication Partner (ReplicationPartners)
- Last Successful Replication (LastReplication)
- AD Replication Failure Count (FailureCount)
- AD Replication Failure Type (FailureType)
- AD Replication Failure First Recorded Time (FirstFailure)



Further to Active Directory replication topologies, there are two types of replications.

- **Intra-Site –** Replications between domain controllers in same Active Directory Site
- **Inter-Site –** Replication between domain controllers in different Active Directory Site

Tech Savvy Productions

We can review AD replication site objects using Get-ADReplicationSite cmdlet.

Get-ADReplicationSite -Filter *

Above command returns all the AD replication sites in the AD forest.

```
PS C:\Users\Administrator> Get-ADReplicationSite -Filter *

Description                     : UK AD Site
DistinguishedName               : CN=LondonSite,CN=Sites,CN=Configuration,DC=rebeladmin,DC=com
InterSiteTopologyGenerator      : CN=NTDS Settings,CN=REBEL-PDC-01,CN=Servers,CN=LondonSite,CN=Sites,CN=Configuration,DC=rebeladmin,DC=com
ManagedBy                       :
Name                            : LondonSite
ObjectClass                     : site
ObjectGUID                      : fbef3a2c-2de8-44d2-bde9-c37403c9f3a9
ReplicationSchedule             : System.DirectoryServices.ActiveDirectory.ActiveDirectorySchedule
UniversalGroupCachingRefreshSite :

Description                     : Canada AD Site
DistinguishedName               : CN=CanadaSite,CN=Sites,CN=Configuration,DC=rebeladmin,DC=com
InterSiteTopologyGenerator      : CN=NTDS Settings\0ADEL:816b7b06-7427-4ccb-8303-ab229caa9931,CN=REBEL-SDC-02\0ADEL:1ed99178-bfb2-4717-80b2-c
                                  48d3c5f80ad,CN=Servers,CN=LondonSite,CN=Sites,CN=Configuration,DC=rebeladmin,DC=com
ManagedBy                       :
Name                            : CanadaSite
ObjectClass                     : site
ObjectGUID                      : 1bc04b4a-0f69-4ef5-8083-98d9bb0e88ca
ReplicationSchedule             :
UniversalGroupCachingRefreshSite :
```

```
PS C:\Windows\system32> Get-ADReplicationSite -Filter *



Description                     :
DistinguishedName               :
CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=HomeLab,DC=TechSavvyProduc
                                  tions,DC=com
InterSiteTopologyGenerator      : CN=NTDS
Settings,CN=WIN2016DC3,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=C

onfiguration,DC=HomeLab,DC=TechSavvyProductions,DC=com
ManagedBy                       :
Name                            : Default-First-Site-Name
ObjectClass                     : site
ObjectGUID                      : ad9a654c-7079-43c0-bcbc-587c66f73b22
ReplicationSchedule             :
System.DirectoryServices.ActiveDirectory.ActiveDirectorySchedule
UniversalGroupCachingRefreshSite :
```


Tech Savvy Productions

We can review AD replication site links on the AD forest using,

Get-ADReplicationSiteLink -Filter *

In site links, most important information is to know the site cost and replication schedule. It allows to understand the replication topology and expected delays on replications.

```
PS C:\Windows\system32> Get-ADReplicationSiteLink -Filter *


Cost                          : 100
DistinguishedName             : CN=DEFAULTIPSITELINK,CN=IP,CN=Inter-Site

Transports,CN=Sites,CN=Configuration,DC=HomeLab,DC=TechSavvyProductions,DC=com
Name                          : DEFAULTIPSITELINK
ObjectClass                   : siteLink
ObjectGUID                    : 2cb21e65-16a9-4b7d-aaeb-b49df87e2416
ReplicationFrequencyInMinutes : 180
SitesIncluded                 :
{CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=HomeLab,DC=TechSavvyProducti
                                ons,DC=com}
```

Get-ADReplicationSiteLink -Filter {SitesIncluded -eq "CanadaSite"} | Format-Table Name,Cost,ReplicationFrequencyInMinutes -A

Above command list all the replication sites link included CanadaSite AD site along with the site link name, link cost, replication frequency.

Tech Savvy Productions

A site link bridge can use to bundle two or more site links and enables transitivity between site links.

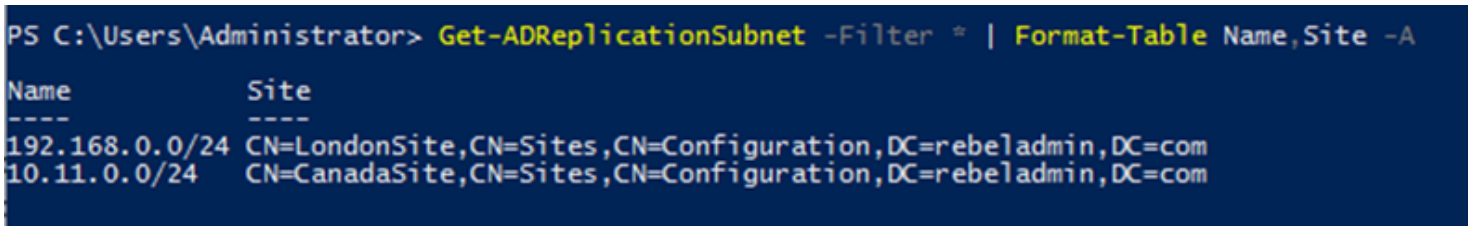
Site link bridge information can retrieve using,

```
Get-ADReplicationSiteLinkBridge -Filter *
```

Active Directory sites may use multiple IP address segments for its operations. It is important to associate those with the AD site configuration so domain controllers know which computer related to which site.

```
Get-ADReplicationSubnet -Filter * | Format-Table Name,Site -A
```

Above command will list down all the Subnets in the forest in a table with subnet name and AD site.



Bridgehead servers are operating as the primary communication point to handle replication data which comes in and go out from AD site.

We can list down all the preferred bridgehead servers in a domain using,

```
$BHservers = ([adsi]"LDAP://CN=IP,CN=Inter-Site Transports,CN=Sites,CN=Configuration,DC=rebeladmin,DC=com").bridgeheadServerListBL

$BHservers | Out-GridView
```

In above command the attribute value bridgeheadServerListBL retrieve via ADSI connection.

Tech Savvy Productions

We can list down all of these findings using on script.

```
## Script to gather information about Replication Topology ##

## Define Objects ##
$replreport = New-Object PSObject -Property @{
Domain = $null
}

## Find Domain Information ##
$replreport.Domain = (Get-ADDomain).DNSroot

## List down the AD sites in the Domain ##
$a = (Get-ADReplicationSite -Filter *)
Write-Host "########" $replreport.Domain "Domain AD Sites" "########"
$a | Format-Table Description,Name -AutoSize

## List down Replication Site link Information ##
$b = (Get-ADReplicationSiteLink -Filter *)
Write-Host "########" $replreport.Domain "Domain AD Replication SiteLink Information" "########"
$b | Format-Table Name,Cost,ReplicationFrequencyInMinutes -AutoSize

## List down SiteLink Bridge Information ##
$c = (Get-ADReplicationSiteLinkBridge -Filter *)
Write-Host "########" $replreport.Domain "Domain AD SiteLink Bridge Information" "########"
$c | select Name,SiteLinksIncluded | Format-List

## List down Subnet Information ##
$d = (Get-ADReplicationSubnet -Filter * | select Name,Site)
Write-Host "########" $replreport.Domain "Domain Subnet Information" "########"
$d | Format-Table Name,Site -AutoSize

## List down Prefered BridgeHead Servers ##
$e = ([adsi]"LDAP://CN=IP,CN=Inter-Site
Transports,CN=Sites,CN=Configuration,DC=rebeladmin,DC=com").bridgeheadServerListBL
Write-Host "########" $replreport.Domain "Domain Prefered BridgeHead Servers" "########"
$e

## End of the Script ##
```

The only thing we need to change is the ADSI connection with relevant domain DN.

Tech Savvy Productions

```
$e = ([adsi]"LDAP://CN=IP,CN=Inter-Site
Transports,CN=Sites,CN=Configuration,DC=rebeladmin,DC=com")
```

Troubleshooting Inconsistent Group Policy Objects (GPOs):

Group Policy is a complex feature of Active Directory, and inconsistent Group Policy Objects (GPOs) can cause various issues. Here are some utilities and tools that can help troubleshoot these problems:

1. **Group Policy Management Console (GPMC):** This is an in-built tool in Windows Server that allows you to manage Group Policy. It includes features for modeling and simulating policy interactions, which can be very helpful in troubleshooting.
2. **Group Policy Results Wizard:** This is a feature within the GPMC that allows you to select a computer and a user account and see what policies would be applied to that combination. It can help identify conflicts and other issues.
3. **GPResult:** This is a command-line tool that displays the Resultant Set of Policy (RSoP) information for a remote user and computer. The gpresult /h report.html command can generate a detailed report in HTML format.
4. **Group Policy Preferences Logging and Tracing:** This feature helps diagnose issues with Group Policy Preferences. It provides detailed error information and is especially useful when trying to troubleshoot why a particular setting isn't being applied.
5. **Event Viewer:** Windows logs Group Policy events in the System log and the Application log in Event Viewer. It can be useful for identifying errors that occurred during the processing of specific policies.

Tech Savvy Productions

6. **ADSI Edit:** This tool can help you navigate and manage your Active Directory objects. While not specifically for Group Policies, it can be useful if you need to troubleshoot at a lower level.
7. **PowerShell cmdlets for Group Policy:** There are several PowerShell cmdlets like Get-GPOReport, Get-GPResultantSetOfPolicy, and Get-GPInheritance which can help troubleshoot GPO problems. PowerShell is particularly handy if you need to script or automate your troubleshooting.
8. **Microsoft's Advanced Group Policy Management (AGPM):** Part of the Microsoft Desktop Optimization Pack (MDOP), AGPM extends the capabilities of the Group Policy Management Console (GPMC) to provide comprehensive change control and improved management.

Remember, troubleshooting GPOs can be complex due to their hierarchical nature and the fact that they can be applied at different levels (site, domain, and OU). It's often helpful to start troubleshooting at the highest level and work your way down.

Tech Savvy Productions

Troubleshooting AD performance issues

Performance issues with Active Directory (AD) can stem from a wide range of causes, such as network issues, high CPU or memory usage, disk space limitations, or inefficient database indexing. Here are some tools that can assist in diagnosing these problems:

1. **Performance Monitor (perfmon):** This built-in Windows tool can monitor various aspects of system performance, such as CPU usage, memory usage, disk activity, and network activity. The data collected can help identify bottlenecks or other issues that might be impacting Active Directory performance.
2. **Windows Event Viewer:** The Event Viewer can provide information about errors or warnings that may relate to performance issues. For instance, long logon times could be logged as warnings here.
3. **AD Replication Status Tool:** This tool can help detect performance issues related to AD replication, such as slow replication or replication failures.
4. **Resource Monitor:** This tool gives you an in-depth look at CPU, memory, disk, and network usage on a per-process basis. It can help you identify which processes are consuming the most resources.
5. **Windows Reliability and Performance Monitor:** This tool combines the functionality of Performance Monitor, Reliability Monitor, and other tools into a single interface. It can provide detailed system statistics and identify potential bottlenecks.

Tech Savvy Productions

6. **Task Manager:** Although simple, Task Manager is an excellent tool for quickly checking CPU, memory, disk, and network usage.
7. **Network Monitor or Wireshark:** These tools can capture network traffic and provide detailed analysis, which can be useful if network issues are suspected.
8. **DCDiag:** This command-line tool can check various aspects of AD health, and some of its tests can help identify performance issues.
9. **PAL (Performance Analysis of Logs) Tool:** This open-source tool reads in a performance monitor counter log and analyzes it using known thresholds.
10. **Database Performance Analyzer:** It can monitor and analyze the performance of AD databases (NTDS.dit), helping you find expensive queries or other issues impacting database performance.

Remember that AD performance can be impacted by a wide range of factors, so it's important to look at the whole picture rather than just a single metric or tool.

Time Syncing Issues:

Time synchronization in an Active Directory (AD) environment is crucial as it affects Kerberos authentication, replication, logging, and other services. Here are some tools and utilities that can help diagnose and fix time synchronization issues:

1. **W32tm:** This is the primary command-line tool for configuring, monitoring, or troubleshooting the Windows Time service. Examples of its usage include w32tm /resync to force a computer to synchronize its time with the domain controller, and w32tm /query /status to check the status of the Windows Time service on a computer.
2. **Net Time:** This is another command-line tool that can be used to set the date and time for a computer or display a list of all domain controllers. For instance, net time /set can force a computer to sync its time with the domain controller.
3. **PowerShell:** Several PowerShell cmdlets can be used for time-related issues in AD, like Get-ADComputer to get information about computer objects in AD and their last logon time.
4. **Event Viewer:** Windows logs many events related to time synchronization. Examining these logs can help identify why a computer is having trouble synchronizing its time.

Tech Savvy Productions

5. **Group Policy:** Administrators can use Group Policy to configure the Windows Time service settings on all computers in a domain. The relevant settings are found under Computer Configuration > Administrative Templates > System > Windows Time Service.
6. **Registry Editor:** The settings for the Windows Time service are stored in the Windows Registry and can be manually edited if necessary. However, this should be done with caution, as incorrect changes can cause serious issues.
7. **System Configuration Checker (sconfig):** A command-line tool available on Windows Server Core installations that allows administrators to configure and manage various aspects of Server Core installations, including setting the current date and time.

Remember, before making any changes in a production environment, it's essential to understand the impact and ensure that changes are tested in a controlled environment. Time synchronization problems can lead to significant issues in an Active Directory environment, so they should be addressed promptly and carefully.

Troubleshooting Outdated or Incorrect AD Schema

Active Directory (AD) Schema contains the definitions of all object classes and their attributes that can be created in an AD environment. It's very critical and rarely modified, but when problems arise due to an outdated or incorrect schema, it can be challenging to troubleshoot. Below are utilities and examples to troubleshoot these issues:

1. **ADSI Edit:** This is a Microsoft Management Console (MMC) snap-in tool used to view and manage low-level details about objects in Active Directory, including schema objects. It's a powerful tool and should be used with care.
2. **Schema Master Operations MMC Snap-In:** This tool is a part of the Active Directory Domain Services role. It allows the administrator to add new attributes and classes to the schema and perform other schema-related tasks.
3. **LDIFDE:** This is a command-line tool that can import or export AD data using files that conform to the LDAP Data Interchange Format (LDIF). It can be used to apply schema updates or changes.

Tech Savvy Productions

4. **DCDiag:** This command-line utility can be used to check the health of the AD environment, including schema-related issues. It provides several tests, including a test specifically for the schema (dcdiag /test:schema).
5. **RepAdmin:** The repadmin /showobjmeta command can be used to see the replication metadata for a specific object, which can help identify issues with schema replication.
6. **Active Directory Schema MMC Snap-In:** This is an MMC tool that can be used to view and modify the AD schema. You need to register the DLL (schmmgmt.dll) first to use this tool.
7. **PowerShell:** You can use PowerShell cmdlets to query schema information and check for potential discrepancies. For example, Get-ADObject -SearchBase (Get-ADRootDSE).schemaNamingContext -Filter * returns all schema objects.

Remember, changes to the AD schema are not reversible and should be made very carefully. It's generally recommended to make changes in a test environment first, and only apply them to the production environment once they've been thoroughly tested.

Also, schema changes should be part of a change control process due to their far-reaching impact.

Tech Savvy Productions

Inadequate Security Measures:

Active Directory (AD) is a critical component of many organizations' IT infrastructure, so it's essential to maintain robust security measures. Here are ten best practices for setting up and evaluating adequate security measures in Active Directory:

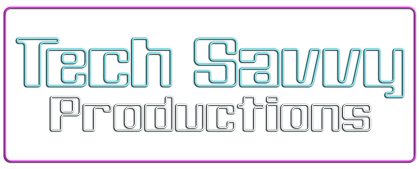
1. **Least Privilege Principle:** Ensure users have the minimum level of access required to perform their roles. Regularly review and update permissions to prevent unauthorized access.
2. **Strong Password Policies:** Implement strong password policies. This includes minimum password lengths, complexity requirements, and password expiration policies. Use fine-grained password policies to enforce stronger rules for privileged accounts.
3. **Multi-Factor Authentication (MFA):** Implement MFA for all users, especially for those with administrative privileges, to add an extra layer of security.
4. **Protect Privileged Accounts:** Ensure that domain admin accounts are only used where necessary, and consider implementing Privileged Access Management (PAM) strategies. Implement the concept of a Secure Admin Workstation (SAW) for administrators.
5. **Regular Auditing and Monitoring:** Regularly audit and monitor Active Directory logs to detect any suspicious activity. Use tools like Windows Event Viewer and third-party SIEM systems for comprehensive monitoring.
6. **Patch Management:** Ensure all domain controllers and AD-connected devices are up-to-date with the latest security patches.
7. **Use Security Groups:** Use security groups to manage permissions rather than assigning permissions to individual users. This makes managing and auditing permissions simpler and more reliable.

Tech Savvy Productions

8. **Implement Role-Based Access Control (RBAC):** Similar to the least privilege principle, RBAC involves assigning roles to users and then assigning permissions to those roles.
9. **Secure LDAP:** Use secure LDAP (LDAPS or startTLS) when integrating applications with Active Directory to prevent password hashes from being transmitted in plain text over the network.
10. **Disaster Recovery Plan:** Ensure a solid disaster recovery plan is in place. This includes regular backups and testing restore procedures to minimize the impact of potential attacks or disasters.

Finally, consider having regular third-party security audits of your AD environment. These audits can identify potential security issues that might be missed internally, helping to maintain a high level of security.

Tech Savvy Productions

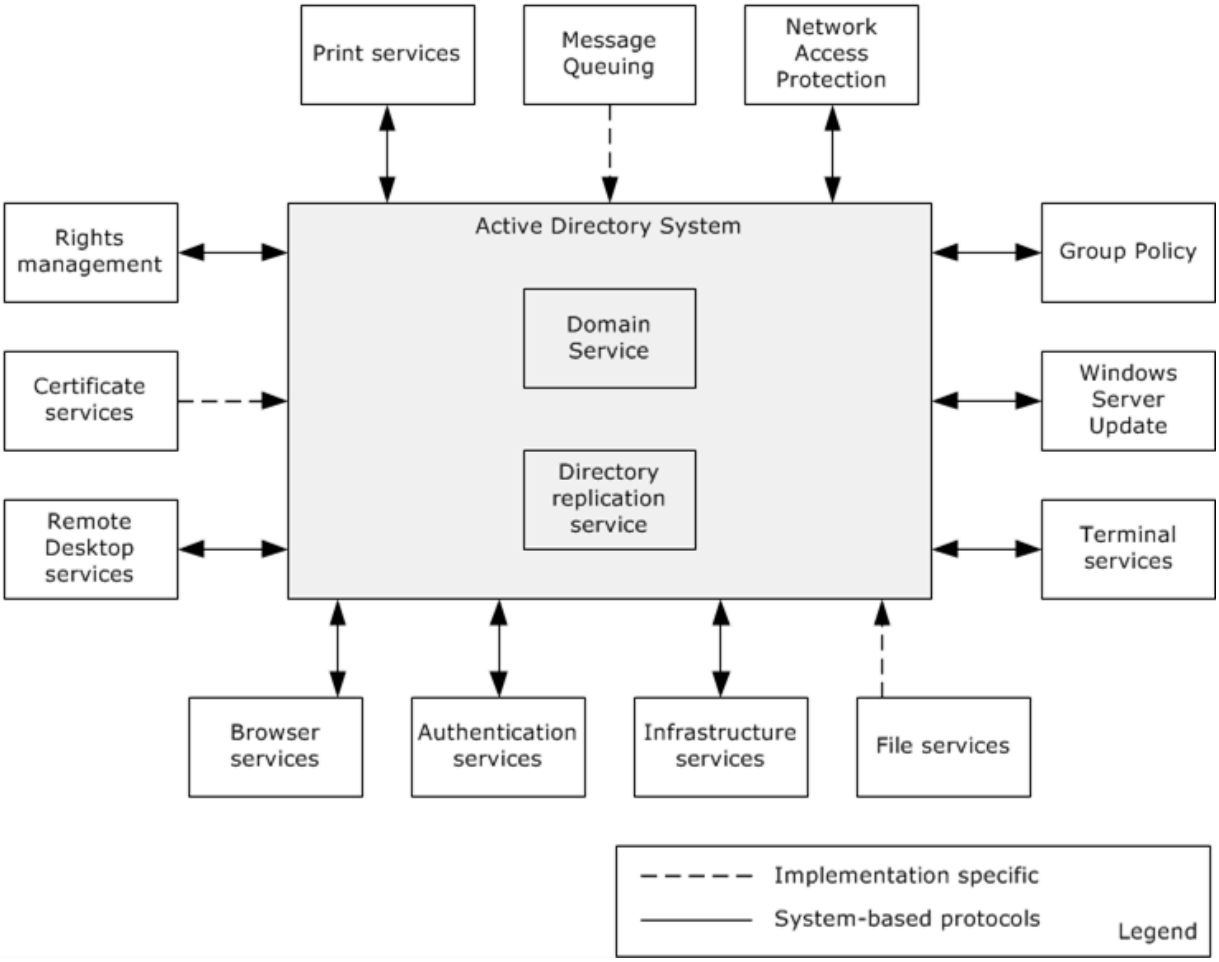

Tech Savvy
Productions

What is Active Directory?

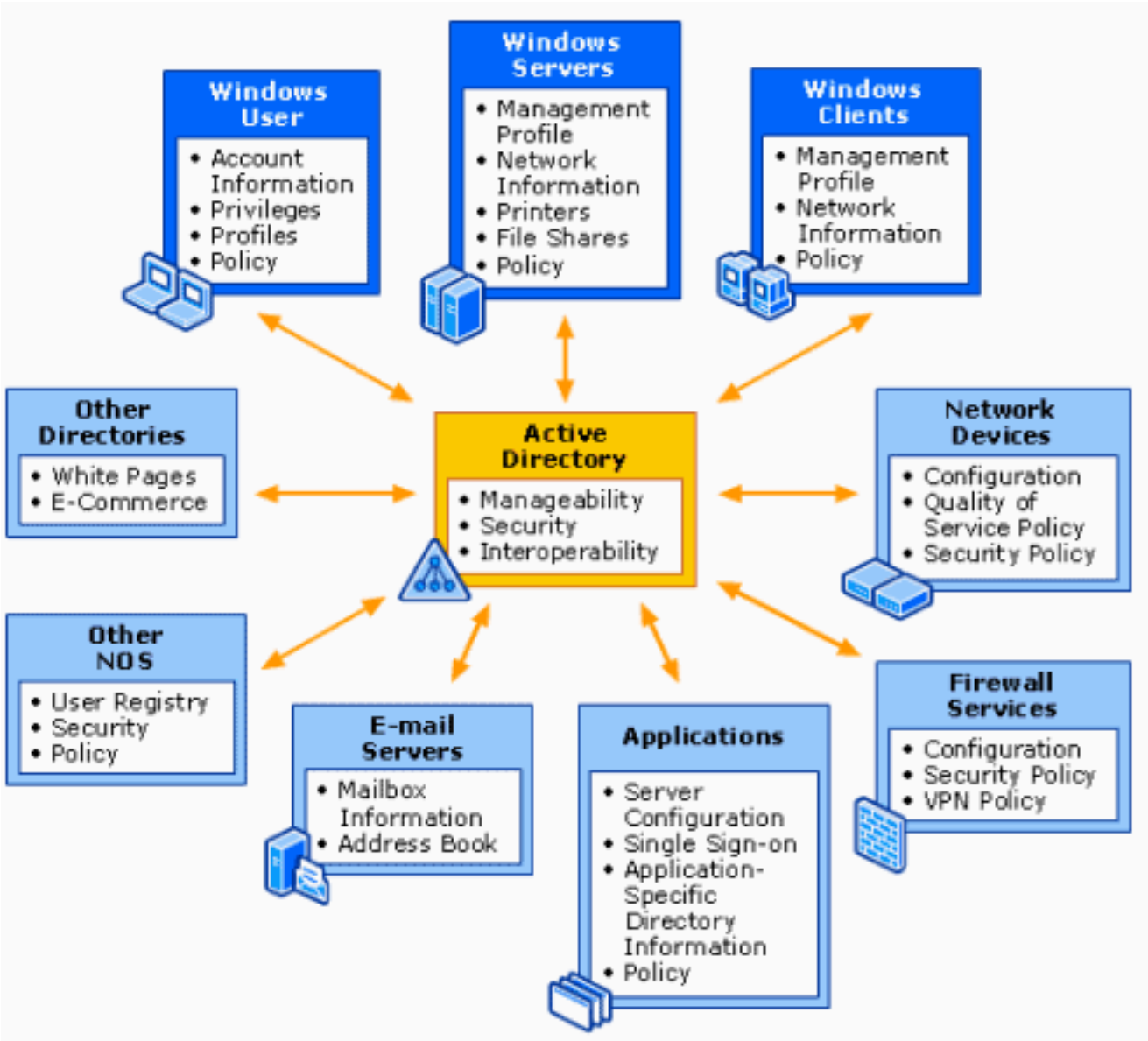
For more information, visit www.netwrix.com

Brian Svidergol

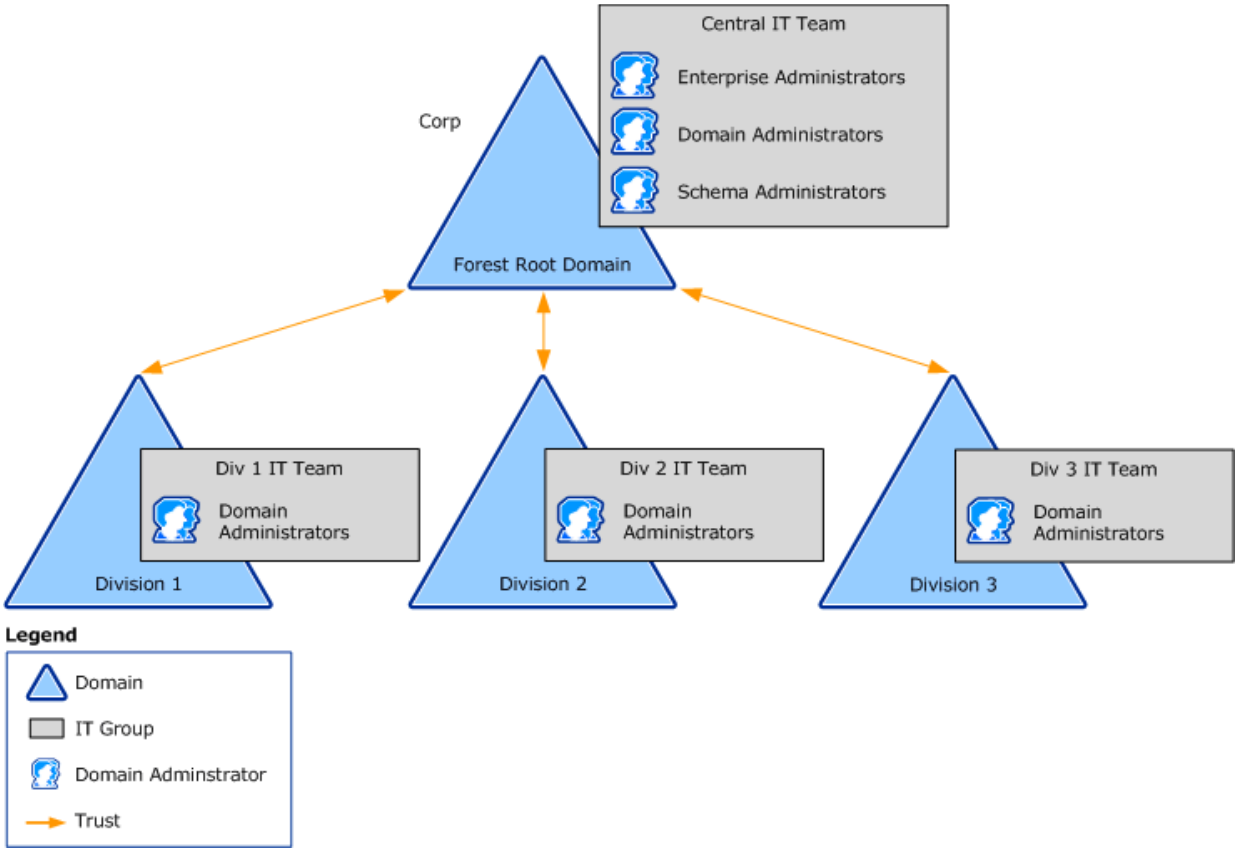
Introduction



Tech Savvy Productions



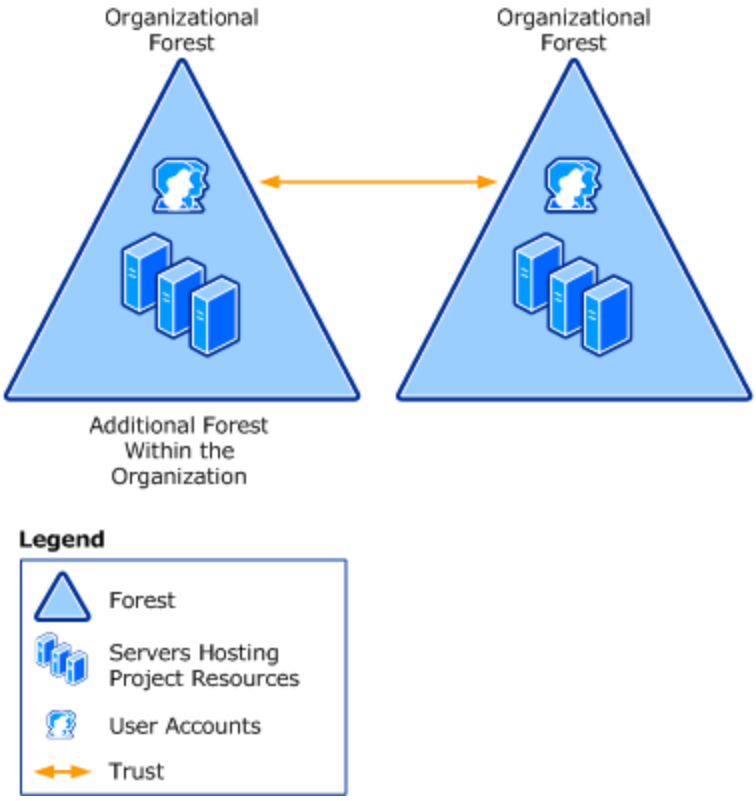
Tech Savvy Productions



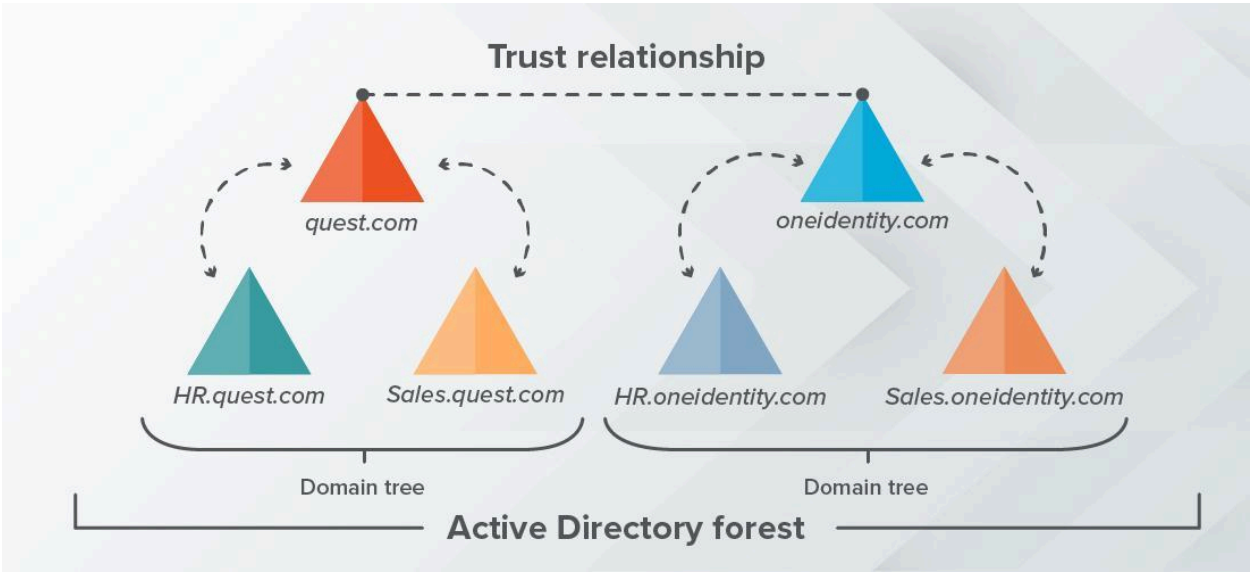
IT administrators have been working with and around Active Directory since the introduction of the technology in Windows 2000 Server. Windows 2000 Server was released on February 17, 2000 but many administrators began working with Active Directory in late 1999 when it was released to manufacturing (RTM) on December 15, 1999.

This e-book is intended not only for beginner SysAdmins who want learn about Active Directory structure, key terminology and configurations. More experienced administrators will find a few best practices of AD management and discover the areas that are often lesser known to IT pros.

Tech Savvy Productions



Every Active Directory design includes at least one organizational forest.

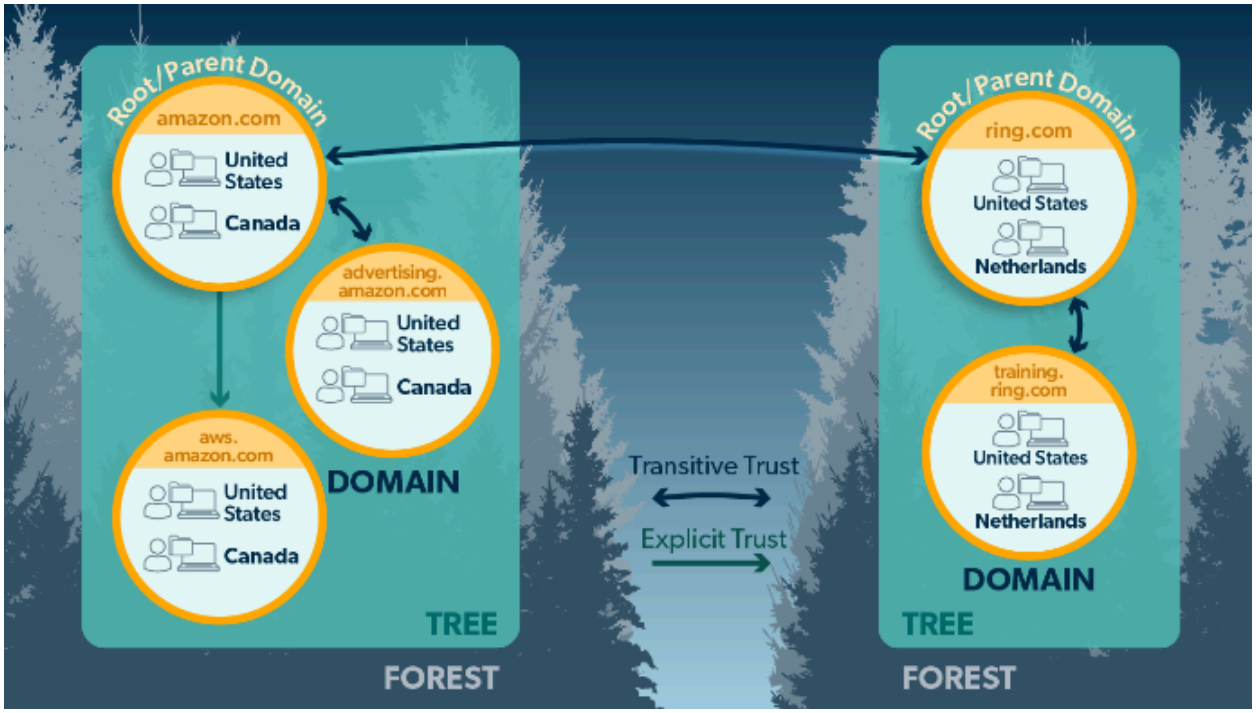


Tech Savvy Productions

https://youtu.be/6yonrl3LU58

I. Introduction to Directory Services Technologies

Like many other areas of IT, directory services has rapidly expanded with new features and functionality along with additional complexity. Instead of a single directory product such as AD DS, there are quite a few other services that make up the directory services category. In addition to the Microsoft solutions, many third-party vendors are creating products that standalone on their own or enhance and expand the Microsoft offerings.



Today, directory services technologies from Microsoft includes the following products:

Active Directory Domain Services (AD DS).

Tech Savvy Productions

- AD DS is the core focus of this e-book so it doesn't require an introduction. But, how about an interesting fact instead?

- According to Microsoft Corporate Vice President Takeshi Numoto, Active Directory is used by 93% of the Fortune 1000.

What's new in Active Directory Domain Services for Windows Server 2016

- 08/15/2022

Applies to: Windows Server 2022, Windows Server 2019, Windows Server 2016

The following new features in Active Directory Domain Services (AD DS) improve the ability for organizations to secure Active Directory environments and help them migrate to cloud-only deployments and hybrid deployments, where some applications and services are hosted in the cloud and others are hosted on premises. The improvements include:

- Privileged access management
- Extending cloud capabilities to Windows 10 devices through Azure Active Directory Join
- Connecting domain-joined devices to Azure AD for Windows 10 experiences
- Enable Windows Hello for Business in your organization
- Deprecation of File Replication Service (FRS) and Windows Server 2003 functional levels

Privileged access management

Privileged access management (PAM) helps mitigate security concerns for Active Directory environments that are caused by credential theft techniques such pass-the-hash, spear phishing, and similar types of attacks. It provides a new administrative access solution that is configured by using Microsoft Identity Manager (MIM). PAM introduces:

- A new bastion Active Directory forest, which is provisioned by MIM. The bastion forest has a special PAM trust with an existing forest. It provides a new

Tech Savvy Productions

Active Directory environment that is known to be free of any malicious activity, and isolation from an existing forest for the use of privileged accounts.

- New processes in MIM to request administrative privileges, along with new workflows based on the approval of requests.
- New shadow security principals (groups) that are provisioned in the bastion forest by MIM in response to administrative privilege requests. The shadow security principals have an attribute that references the SID of an administrative group in an existing forest. This allows the shadow group to access resources in an existing forest without changing any access control lists (ACLs).
- An expiring links feature, which enables time-bound membership in a shadow group. A user can be added to the group for just enough time required to perform an administrative task. The time-bound membership is expressed by a time-to-live (TTL) value that is propagated to a Kerberos ticket lifetime.

Note

Expiring links are available on all linked attributes. But the member/memberOf linked attribute relationship between a group and a user is the only example where a complete solution such as PAM is preconfigured to use the expiring links feature.

- KDC enhancements are built in to Active Directory domain controllers to restrict Kerberos ticket lifetime to the lowest possible time-to-live (TTL) value in cases where a user has multiple time-bound memberships in administrative groups. For example, if you are added to a time-bound group A, then when you log on, the Kerberos ticket-granting ticket (TGT) lifetime is equal to the time you have remaining in group A. If you are also a member of another time-bound group B, which has a lower TTL than group A, then the TGT lifetime is equal to the time you have remaining in group B.
- New monitoring capabilities to help you easily identify who requested access, what access was granted, and what activities were performed.

Requirements for Privileged access management

- Microsoft Identity Manager
- Active Directory forest functional level of Windows Server 2012 R2 or higher.

Azure AD Join

Tech Savvy Productions

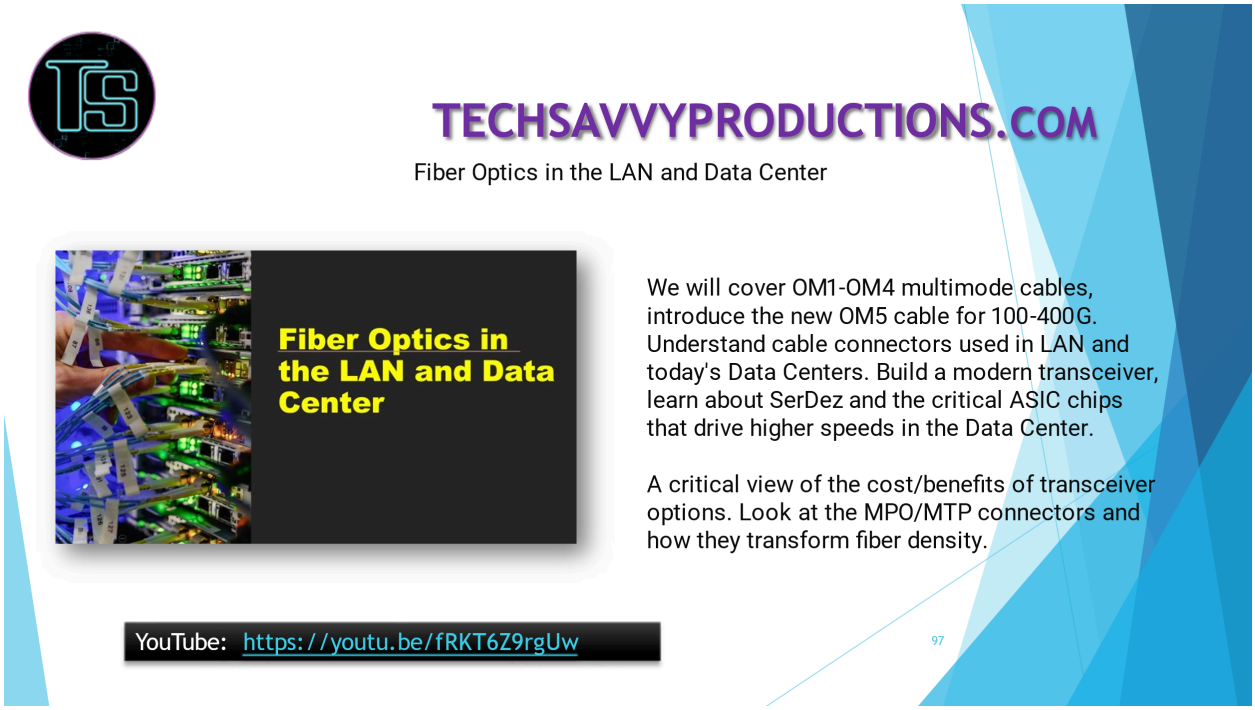
Azure Active Directory Join enhances identity experiences for enterprise, business and EDU customers- with improved capabilities for corporate and personal devices.

Benefits:

- **Availability of Modern Settings** on corp-owned Windows devices. Oxygen Services no longer require a personal Microsoft account: they now run off users' existing work accounts to ensure compliance. Oxygen Services will work on PCs that are joined to an on-premises Windows domain, and PCs and devices that are "joined" to your Azure AD tenant ("cloud domain"). These settings include:
    - Roaming or personalization, accessibility settings and credentials
    - Backup and Restore
    - Access to Microsoft Store with work account
    - Live tiles and notifications
- **Access organizational resources** on mobile devices (phones, tablets) that can't be joined to a Windows Domain, whether they are corp-owned or BYOD.
- **Single-Sign On** to Office 365 and other organizational apps, websites, and resources.
- **On BYOD devices**, add a work account (from an on-premises domain or Azure AD) to a personally owned device and enjoy SSO to work resources, via apps and on the web, in a way that helps ensure compliance with new capabilities such as Conditional Account Control and Device Health attestation.
- **MDM integration** lets you auto-enroll devices to your MDM (Intune or third-party).
    - **Set up "kiosk" mode and shared devices** for multiple users in your organization.
    - **Developer experience** lets you build apps that cater to both enterprise and personal contexts with a shared programing stack.
    - **Imaging** option lets you choose between imaging and allowing your users to configure corp-owned devices directly during the first-run experience.

For more information, see, Introduction to device management in Azure Active Directory.

Tech Savvy Productions



https://youtu.be/fRKT6Z9rgUw

He has shown you, O mortal, what is good.

And what does the Lord require of you?

To act justly and to love mercy

and to walk humbly with your God.

Windows Hello for Business

Windows Hello for Business is a key-based authentication approach for organizations and consumers that goes beyond passwords. This form of authentication relies on breach, theft, and phish-resistant credentials.

The user logs on to the device with a biometric or PIN logon information that is linked to a certificate or an asymmetrical key pair. The Identity Providers (IDPs) validate the user by mapping the public key of the user to IDLocker and provides log on information through One Time Password (OTP), Phone or a different notification mechanism.

For more information, see, Windows Hello for Business

Deprecation of File Replication Service (FRS) and Windows Server 2003 functional levels

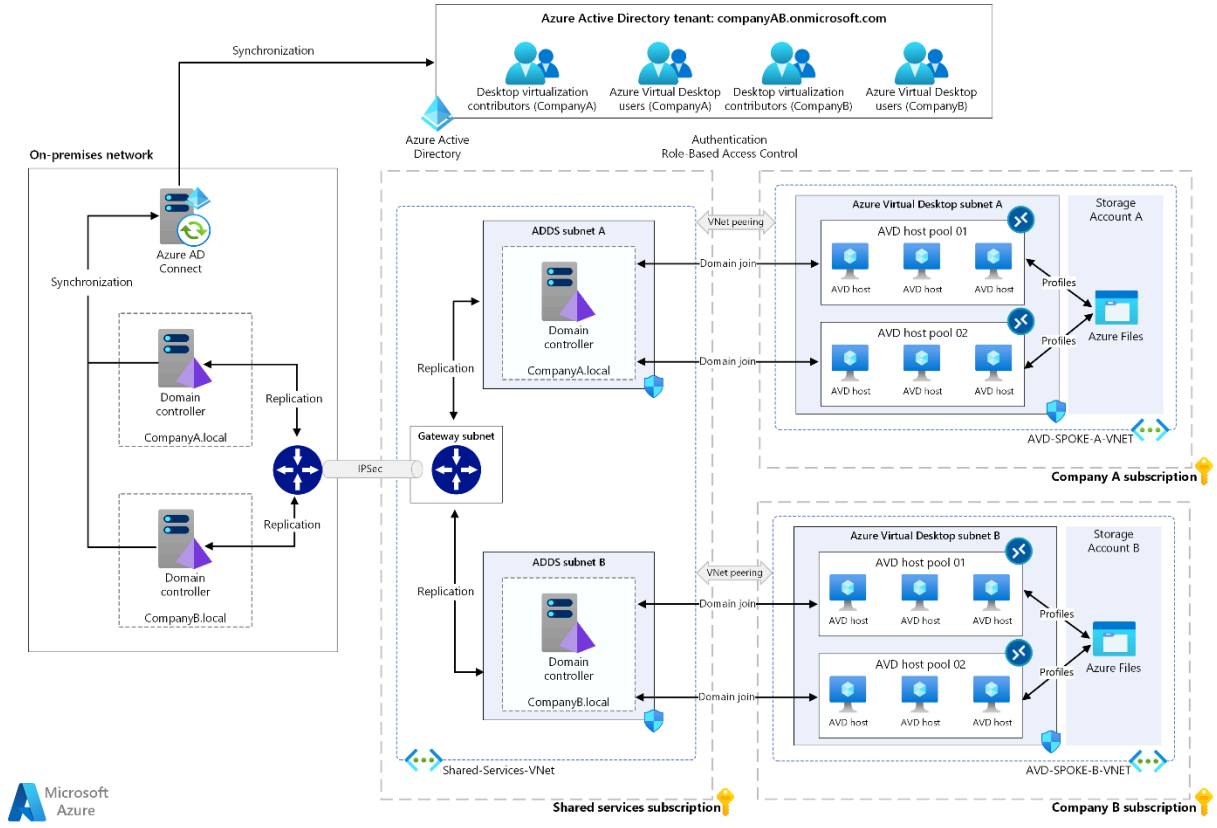
Although File Replication Service (FRS) and the Windows Server 2003 functional levels were deprecated in previous versions of Windows Server, it bears repeating that the Windows Server 2003 operating system is no longer supported. As a result, any domain controller that runs Windows Server 2003 should be removed from the domain. The domain and forest functional level should be raised to at least Windows Server 2008 to prevent a domain controller that runs an earlier version of Windows Server from being added to the environment.

At the Windows Server 2008 and higher domain functional levels, Distributed File Service (DFS) Replication is used to replicate SYSVOL folder contents between domain controllers. If you create a new domain at the Windows Server 2008 domain functional level or higher, DFS Replication is automatically used to replicate the SYSVOL folder. If you created the domain at a lower functional level, you will need to migrate from using FRS to DFS replication for the SYSVOL folder. For migration steps, you can either follow these steps or you can refer to the streamlined set of steps on the Storage Team File Cabinet blog.

The Windows Server 2003 domain and forest functional levels continue to be supported, but organizations should raise the functional level to Windows Server 2008 (or higher if possible) to ensure SYSVOL replication compatibility and support in the future. In addition, there are many other benefits and features available at the higher functional levels higher. For more information, see the following resources:

- Understanding Active Directory Domain Services (AD DS) Functional Levels
- Raise the Domain Functional Level
- Raise the Forest Functional Level

Tech Savvy Productions



In this guide AD design

Understanding AD DS Design

Identifying Your AD DS Design and Deployment Requirements

Mapping Your Requirements to an AD DS Deployment Strategy

Designing the Logical Structure for Windows Server 2008 AD DS

Designing the Site Topology for Windows Server 2008 AD DS

Enabling Advanced Features for AD DS

Evaluating AD DS Deployment Strategy Examples

Appendix A: Reviewing Key AD DS Terms

Tech Savvy Productions

If I speak in the tongues of men or of angels, but do not have love, I am only a resounding gong or a clanging cymbal. If I have the gift of prophecy and can fathom all mysteries and all knowledge, and if I have a faith that can move mountains, but do not have love, I am nothing. If I give all I possess to the poor and give over my body to hardship that I may boast, but do not have love, I gain nothing.

Love is patient, love is kind. It does not envy, it does not boast, it is not proud. It does not dishonor others, it is not self-seeking, it is not easily angered, it keeps no record of wrongs. Love does not delight in evil but rejoices with the truth. It always protects, always trusts, always hopes, always perseveres.

Active Directory Lightweight Directory Services (AD LDS).

AD LDS is the lightweight, developer- friendly, directory that can be deployed on a client computer and client operating system as well as on a server. It isn't as full featured as AD DS (for example, Group Policy isn't part of it) but it can be useful as a decentralized directory for developers and testers.

Active Directory Federation Services

- 07/29/2021

Applies to: Windows Server 2022, Windows Server 2019, Windows Server 2016, Windows Server 2012 R2, Windows Server 2012

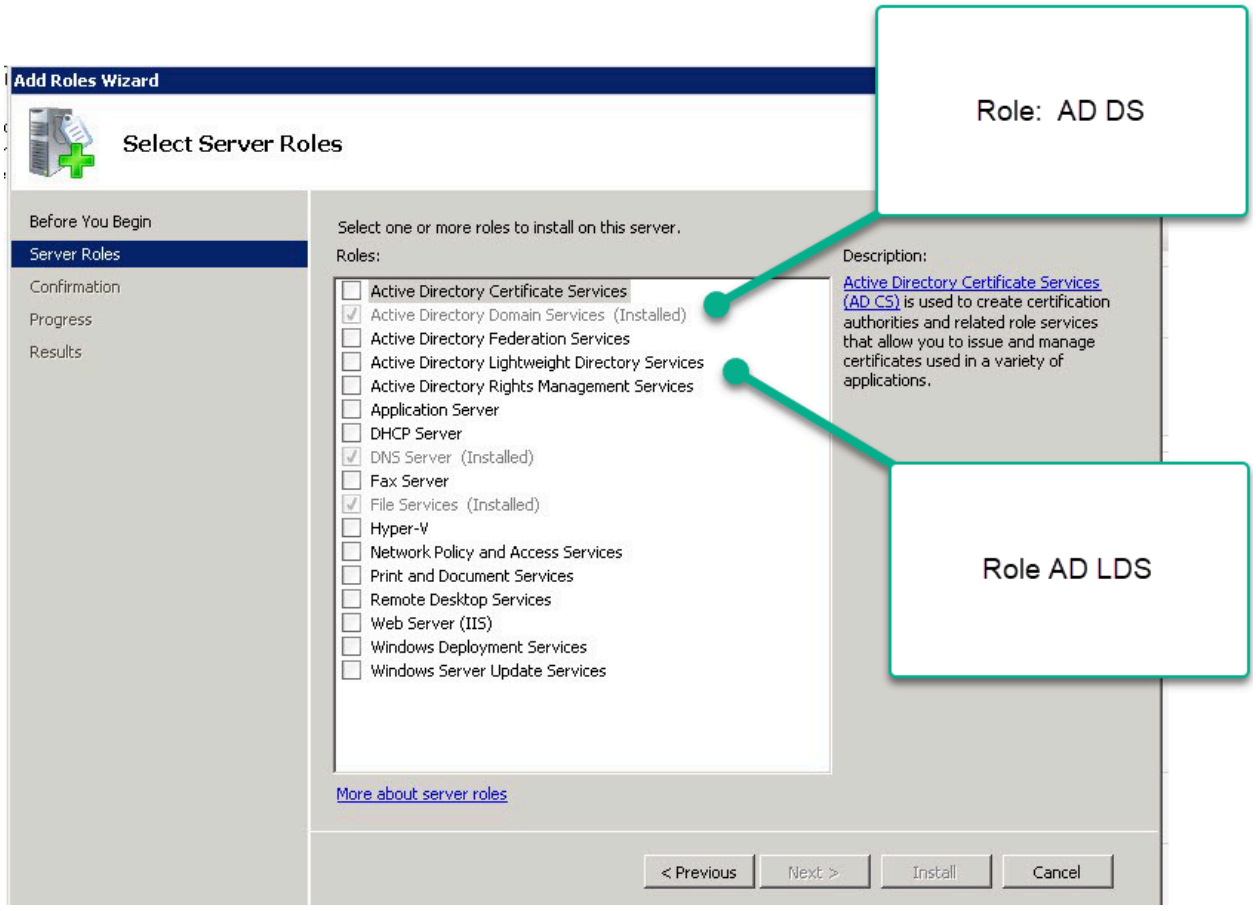
This document contains a list of all of the documentation areas for AD FS for Windows Server 2016, 2012 R2, and 2012. This includes the following:

- AD FS Overview
- AD FS Design
- AD FS Deployment
- AD FS Development
- AD FS Operations
- AD FS Technical Reference

Active Directory Federation Service (AD FS) enables Federated Identity and Access Management by securely sharing digital identity and entitlements rights across security and enterprise boundaries. AD FS extends the ability to use single sign-on functionality that is available within a single security or enterprise boundary to Internet-facing applications to enable customers, partners, and suppliers a streamlined user experience while accessing the web-based applications of an organization.

Tech Savvy Productions

https://youtu.be/_jOY-EIDTxk



Of making many books there is no end, and much study wearies the body.

Now all has been heard;

here is the conclusion of the matter:

Fear God and keep his commandments,

for this is the duty of all mankind.

For God will bring every deed into judgment,

including every hidden thing,

whether it is good or evil.

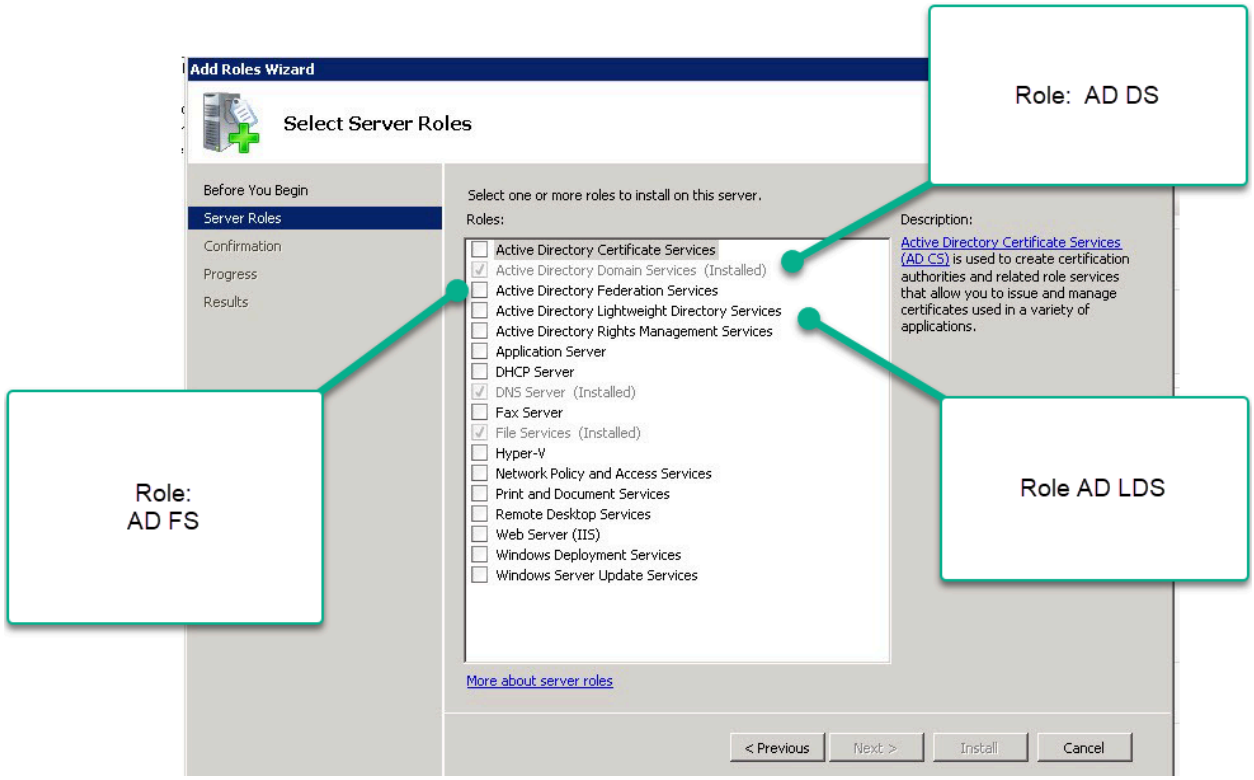
Active Directory Federation Services (AD FS)

AD FS is a claims-based identity solution that helps independent organizations connect their directory services technologies together to facilitate single sign-on and cross-organizational resource access.

Today, it has become a fairly common solution because it helps organizations connect to cloud services such as Microsoft Azure.

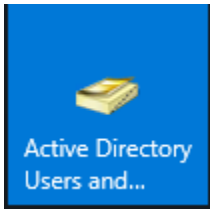
Additionally, there are two other roles that you may be wondering about. Active Directory Certificate Services (AD CS) and Active Directory Rights Management Services (AD RMS) are often grouped in with the other technologies listed above to form the suite of technologies offered by Microsoft for on-premise Active Directory related deployments.

Tech Savvy Productions



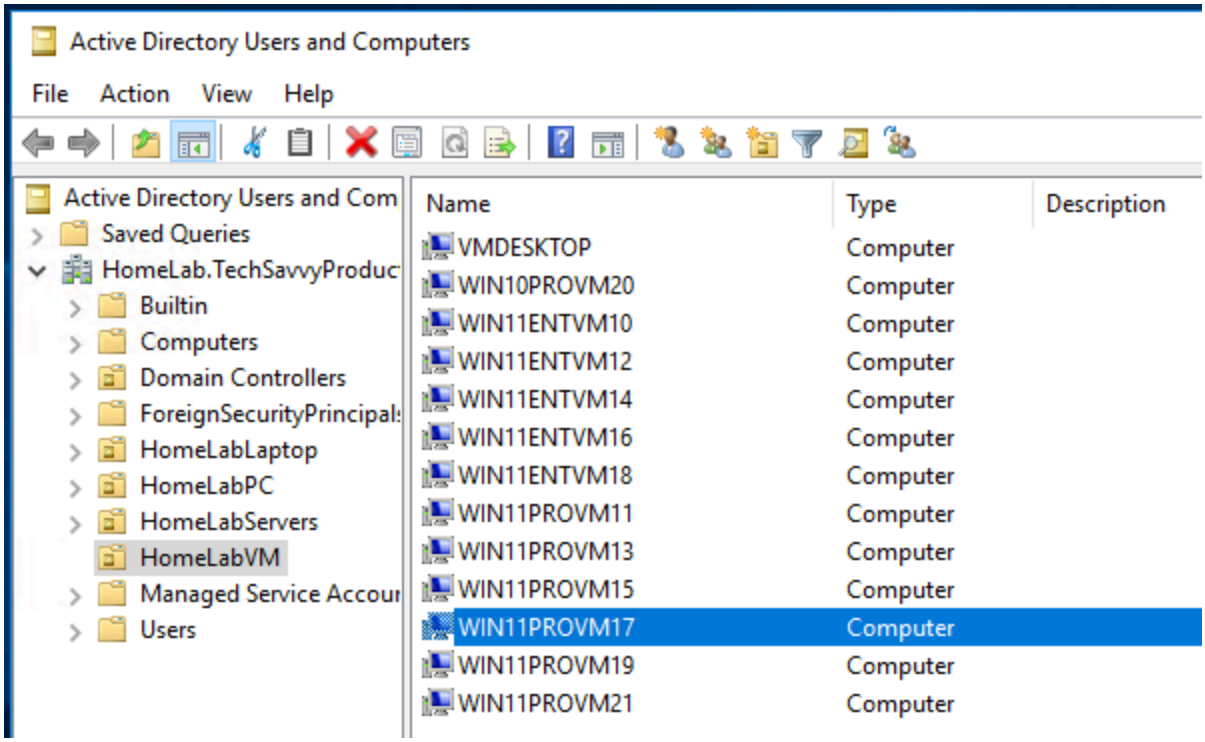
Additionally, there are products outside of the immediate Active Directory family such as Microsoft Forefront Identity Manager (FIM). Beyond the on-premise technologies, there are also several cloud-based solutions that offer services in the cloud such as Azure Active Directory and Azure Multi-Factor Authentication. For this white paper, we are focusing on AD DS deployed on-premise.

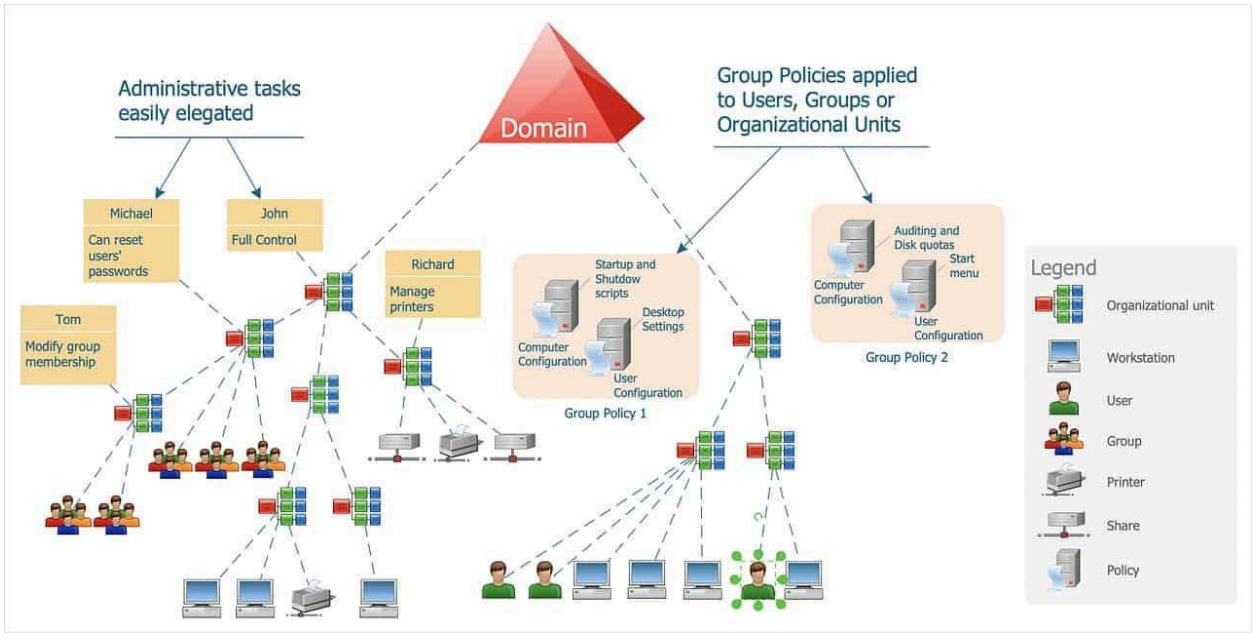
II. Administration Tools



There are numerous tools for Active Directory. The tool that we will cover today is Active Directory Users and Computers (ADUC), which was released with Windows 2000 Server. ADUC is an MMC snap-in that enables administrators to manage Active Directory objects, including users, computers, groups, organizational units (OUs), and attributes.

Tech Savvy Productions

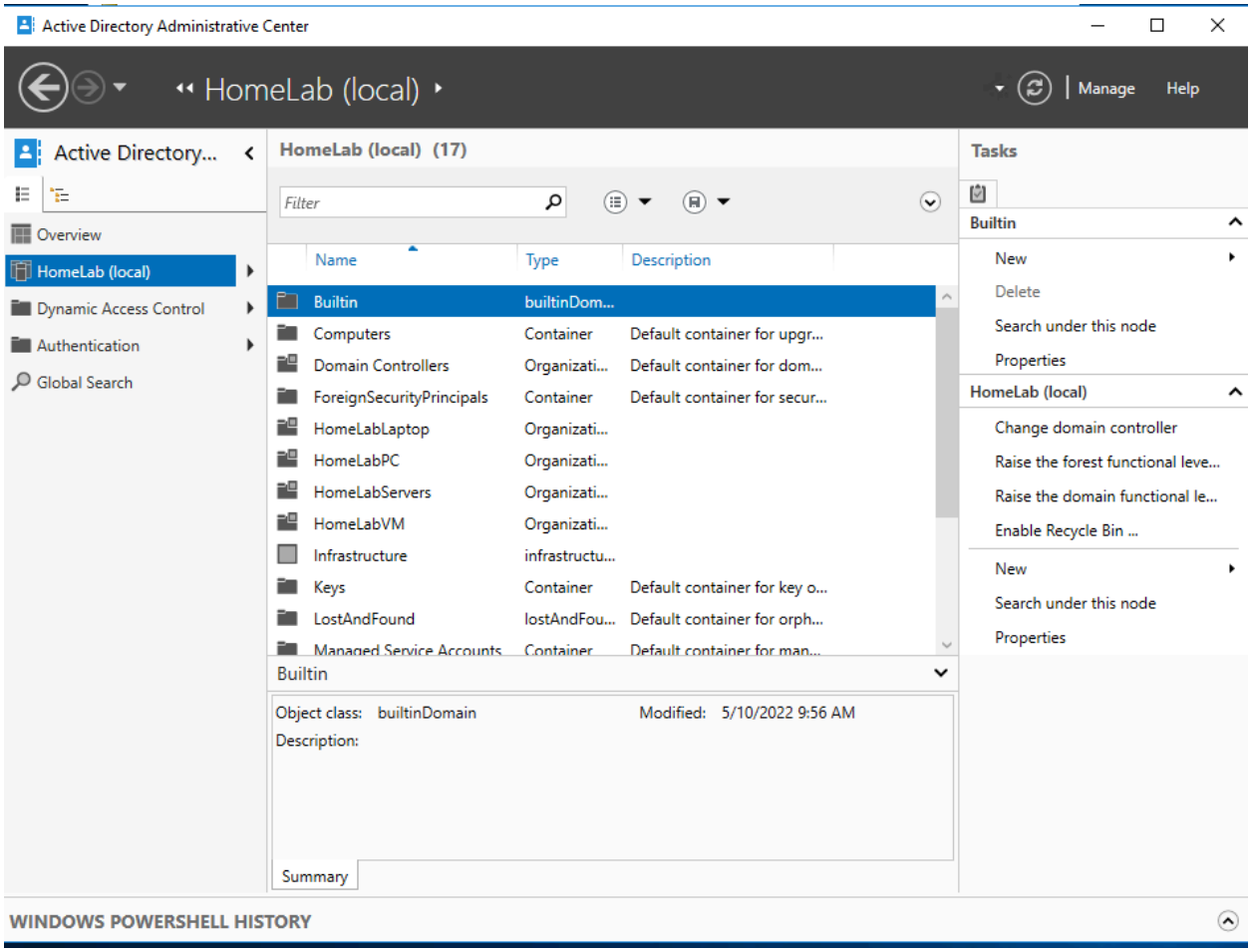




While the features of ADUC and many other features have been added to a new tool named Active Directory Administrative Center, ADUC remains a popular

Tech Savvy Productions

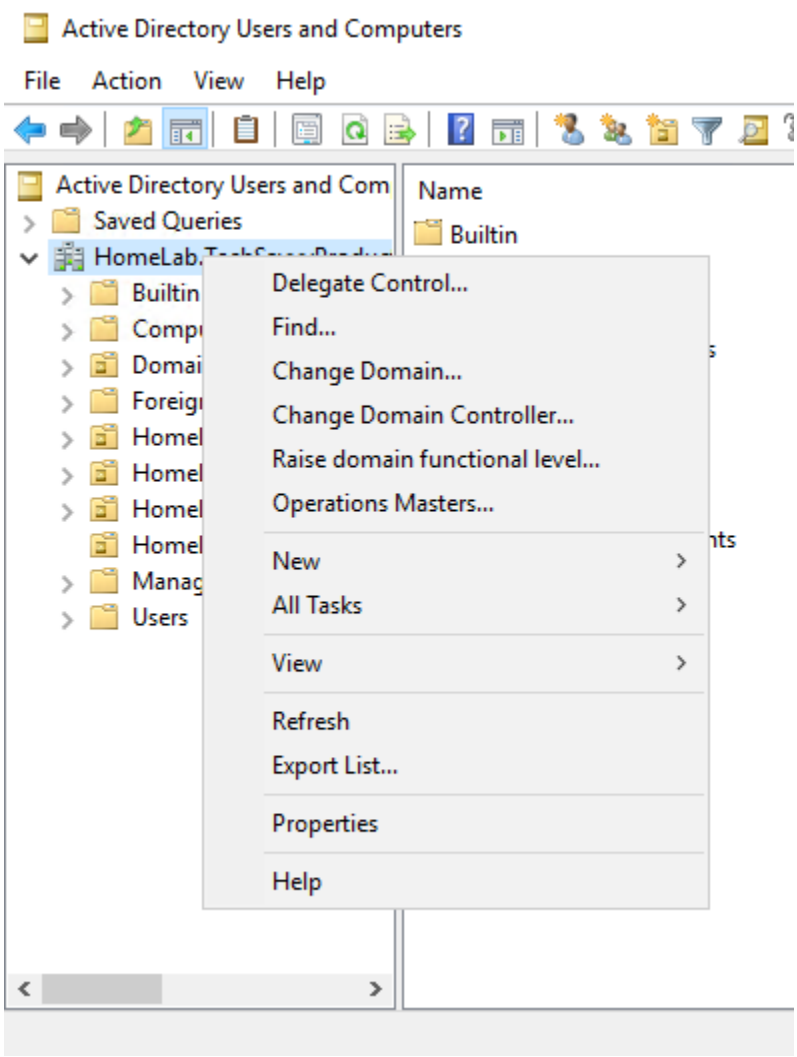
tool that administrator's use to manage their environment.



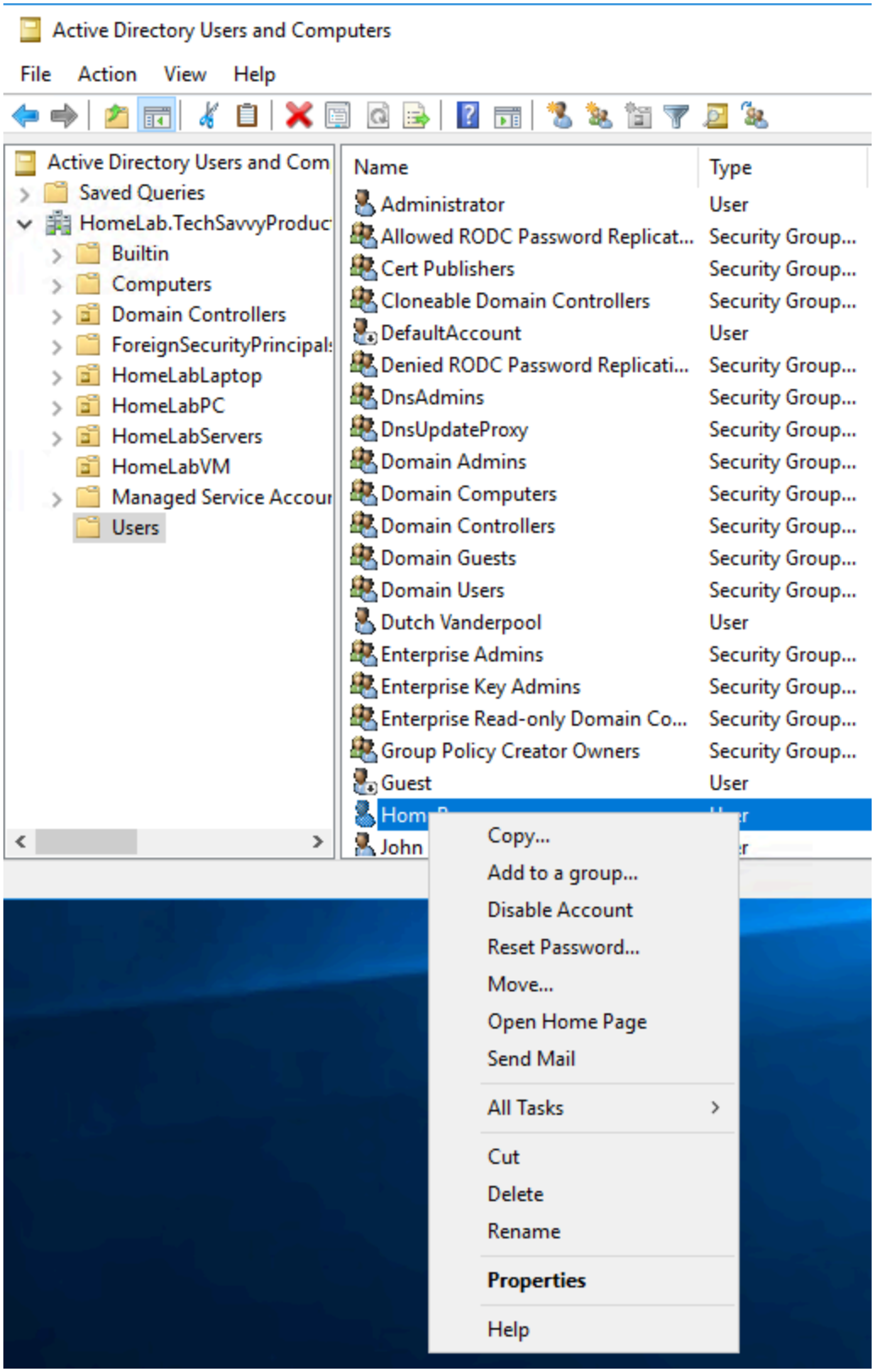
In addition to managing objects, ADUC can also manage domain operations. For example, you can raise the domain functional level from ADUC. You can also transfer the RID, PDC Emulator, and Infrastructure FSMO roles to a different

Tech Savvy Productions

domain controller by using ADUC.



Tech Savvy Productions



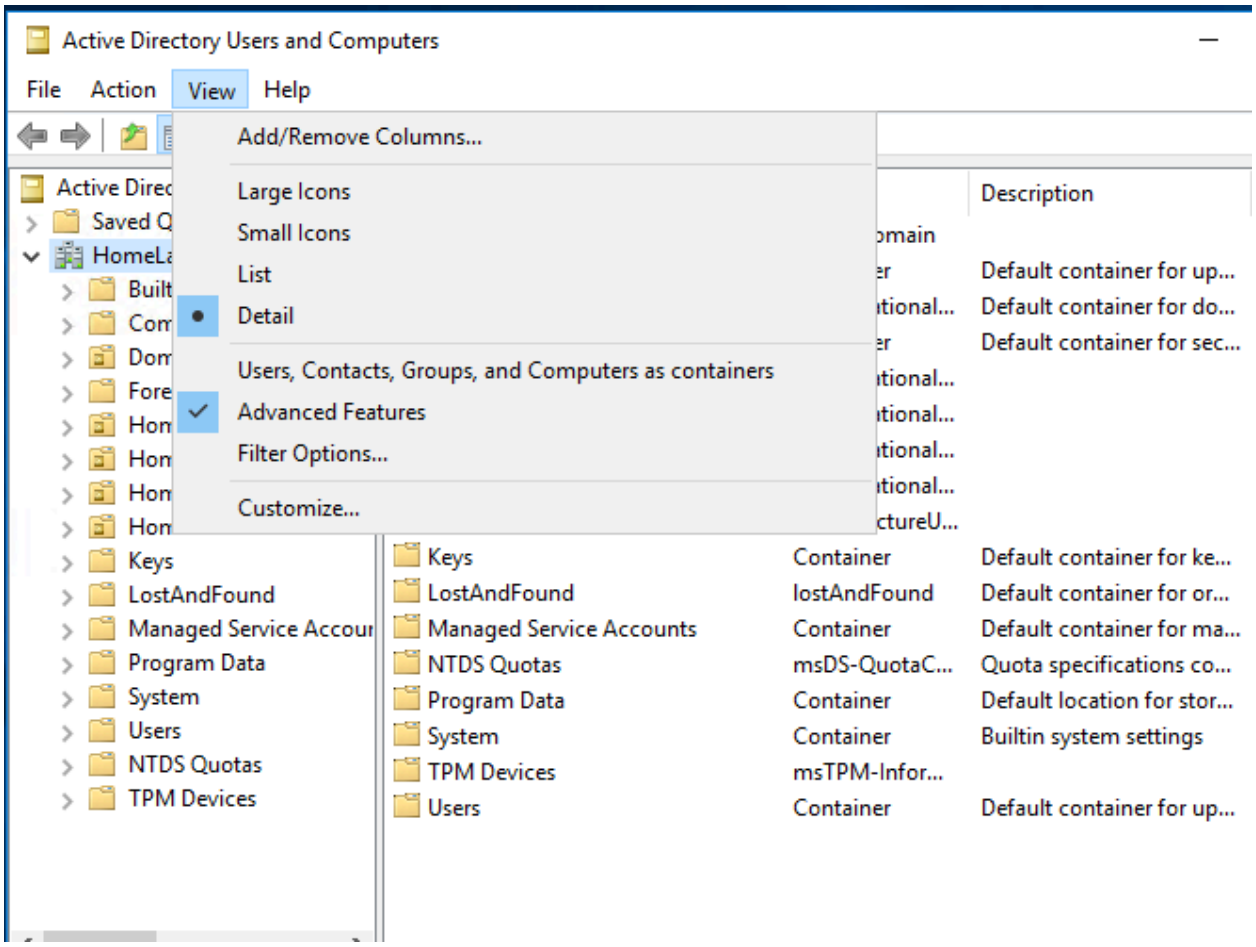
Managing an object consists of some of the more obvious tasks such as

- Resetting a user's password,

Tech Savvy Productions

- adding users to security groups,
- and moving computer objects.

However, the Advanced Features setting within ADUC can also allow you to manage the LostAndFound container, NTDS Quotas, Program Data, and System information.



This view is not enabled by default but you can enable through the View menu.

The Advanced Features option adds many tabs to the properties page of an object, including Published Certificates, Attribute Editor, Password Replication, and others.

The View menu also allows you to filter the view based on the object type, such as user, computer, printer and more. Individual columns can also be added or

Tech Savvy Productions

removed, to customize the view to include other attributes that have been assigned to the object, for example the last modify date, city, country, email address, and more.
Finally, ADUC also enables you to delegate control of objects through the Delegation of Control wizard or by manually modifying permissions on an object.

Inside AD

There are many aspects to an AD DS deployment. In this white paper, we will look at domain controllers, the AD DS database, and SYSVOL. While we only focus on parts of these three components in this e-book, other parts and components are also important.

Domain Controllers

The domain controller is the backbone of Active Directory. Without a domain controller, you can't have a directory!

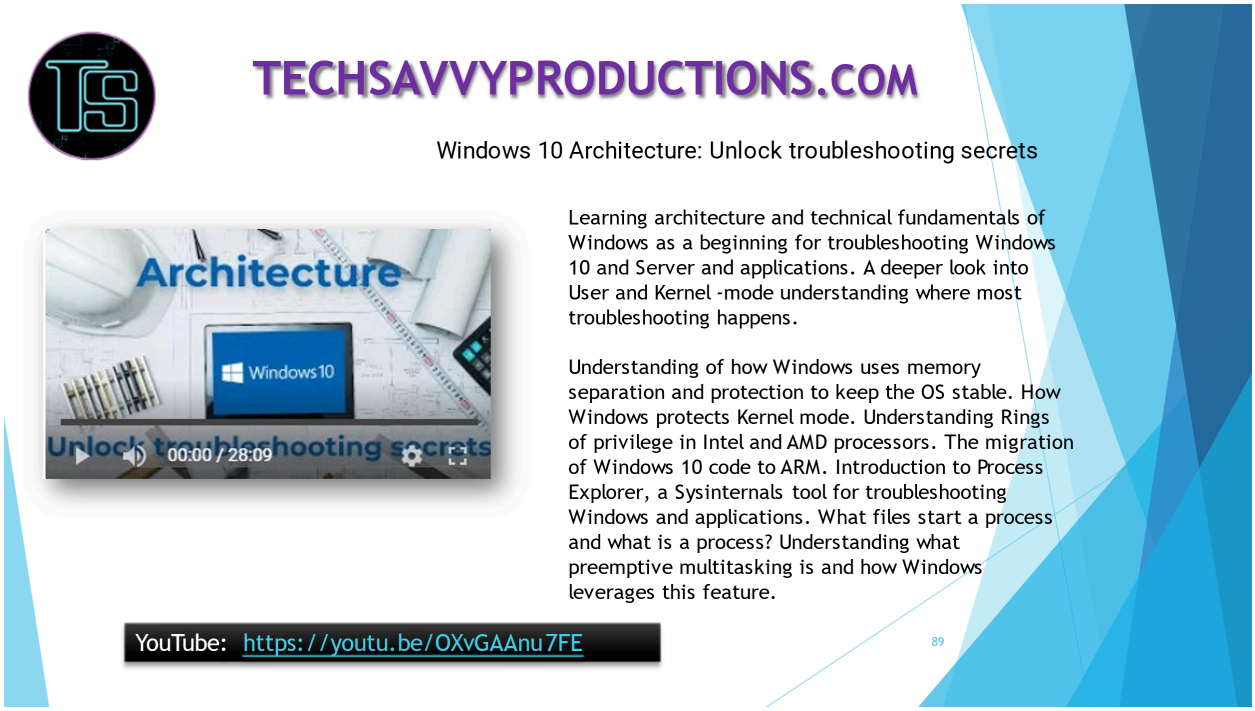
You can use up to 1,200 domain controllers in a single domain.

But, don't judge another administrator's environment by the size or scale of it!
Let's look at the evolution of the domain controller:
Windows NT 3.1 introduced the original Microsoft domain. Windows NT 3.1 (subsequently 3.5 and then 3.51) should not be confused with Windows 3.1 which was a 16-bit client operating system. The domain functionality included with Windows NT was not a multi-master model like AD DS. Thus, there was a primary domain controller (PDC) and backup domain controllers (BDCs).

All changes were handled by the PDC. A BDC could be promoted to a PDC in a disaster recovery situation.

Tech Savvy Productions

Today, we have the PDC Emulator FSMO role which is directly related to the original PDC.



https://youtu.be/OXvGAAnu7FE

Windows 2000 Server introduced Active Directory.

With the release of Windows
2000 Server, Microsoft revamped a large amount of the traditional domain and marketed the service as Active Directory. A key feature of Active Directory was the multi-master model which allowed most of the Active Directory functionality, including changes, to take place on any DC in the domain.

Windows Server 2003 introduced new features.

With Windows Server 2003, Active Directory was updated with some administrative enhancements (such as multiselecting objects in ADUC), added the ability to create forest trusts, and added the universal group membership caching feature. Other features were added or expanded too, especially around command-line administration.

Windows Server 2003 R2 introduced AD FS and Active Directory Application Mode (ADAM).

AD FS and ADAM were big enhancements, especially looking at them today in 2015. Back then, they weren't used much though. ADAM later became AD LDS while AD FS was updated along the way for cloud integration.

7

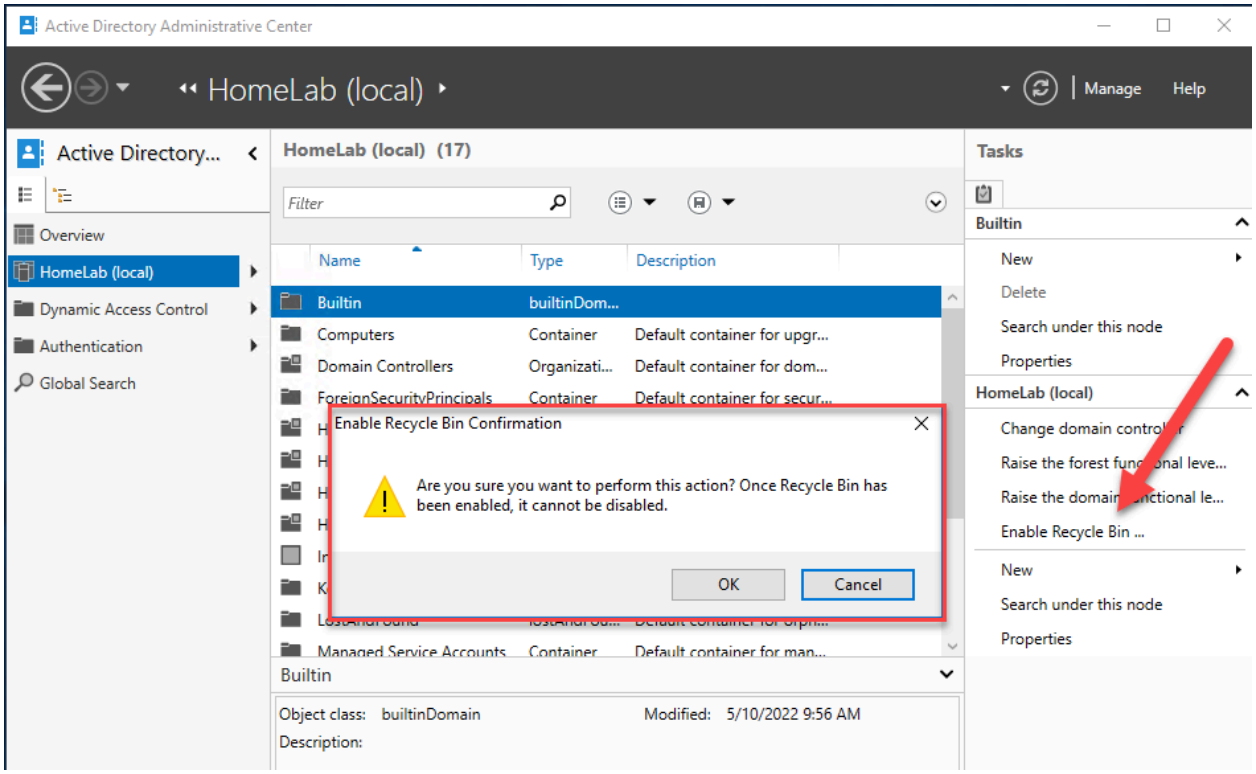
Windows Server 2008 introduced read-only domain controllers (RODCs) and fine-grained password policies.

With Windows Server 2008, RODCs became an option which allowed administrators to deploy DCs in insecure computer closets at branch offices, among other uses. In addition, fine-grained password policies were introduced, albeit with some administrative challenges such as not having a graphical user interface to manage the policies.

Windows Server 2008 R2 introduced the recycle bin and the PowerShell module.

Windows Server 2008 R2 continued refining some of the features introduced in Windows Server 2008 and offered the Recycle Bin and a PowerShell module which was paramount for administrators to be able to effectively manage AD DS from PowerShell.

Tech Savvy Productions



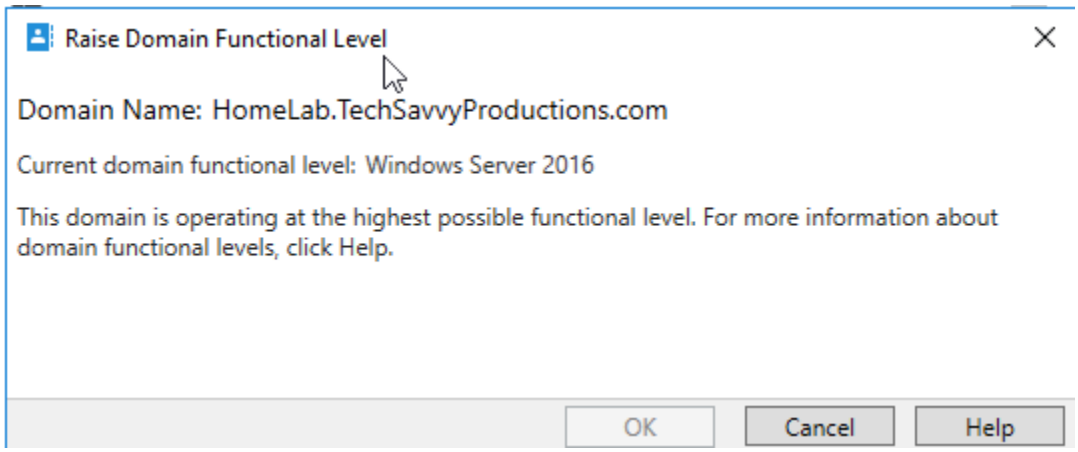
https://youtu.be/-bsLmDfvF1Y

Windows Server 2012 introduced simplified management and enhanced virtualization support.

The long awaited graphical user interface tools to manage the Recycle Bin and fine-grained password policies were introduced. Additionally, virtualization was enhanced and support for virtualizing DCs became mainstream. See https://technet.microsoft.com/en-us/library/hh831477.aspx for a complete guide on the changes.

Windows Server 2012 R2 focused on security enhancements.

New features included multi-factor authentication, single sign-on from connected devices, and multi-factor access control. See https://technet.microsoft.com/en-us/library/dn268294.aspx for a complete guide on the changes.



Inside AD

Tech Savvy Productions

There are some good practices to adhere to when deploying DCs. Many of these practices are documented. But not many organizations are implementing these practices. We will skip over the well-known good practices such as maintaining the **Active Directory database on one set of disk spindles**, the **log files on separate disk spindle**s, and the operating system on its own set of disk spindles.

Some of the lesser implemented good practices for domain controllers are: Many administrators avoid change, especially for systems such as AD DS that are incredibly stable. So when a new administrator proposes switching over to the Server Core installation, he is often met with icy stares.

But the reality is that most administrators administrate AD DS remotely by launching ADUC or PowerShell on their client or administrative computer.

All of the core management tools including the Active Directory Administrative Center (ADAC) and Windows PowerShell work almost identically when used locally on a DC or remotely from a client computer or an administrative computer.

Thus, by moving to the Server Core installation, the administrative experience isn't degraded. And, you gain security enhancements and some small performance enhancements.

Run the Server Core installation of the operating system

Back in the old days, like 10 years ago, most organizations used physical servers because virtualization was in its infancy. So, when it was time to provision a new file server, DHCP server, or print server, administrators often just tapped an existing server.

A DC was often used too. Fast forward to 2015 when virtualization is the de facto standard and automated provisioning helps deliver a new VM in minutes and the old way of doing things isn't nearly as compelling.

Do not run other software or services on a DC

Now, when you need a place for a file server, DHCP server, print server, or some other application server, you can provision a new VM. Or, better yet, you can

Tech Savvy Productions

provision a new VM as a utility server. A utility server is a server that hosts all of the applications and services that are too small to warrant a dedicated server.

This allows your DCs to stick with a dedicated service which brings more stability.

While all of your read-write DCs should be in a secure data center, there are plenty of IT and non-IT people that have access to the data center. For example, the contracted electricians that works on the cooling system have data center access. In addition, there are likely network guys, cabling guys, and IT management with data center access. Anybody that has physical access to a DC can gain access to a physical DC in only a couple of minutes at a console in the data center. There are specialized freeware boot images available that you can use to boot into and reset passwords, install malware, or gain access to the disk data, assuming that the disk isn't encrypted. To avoid this, perform the following configurations:

Security for your domain Controller:

- Adjust the startup order and set a BIOS password
- Ensure that all removable media is not part of the BIOS boot order
- TPM
- Secure Boot
- Locked hardware enclosure for unauthorized removal

Instead, only the hard disk where the operating system installed should be part of the boot order. This is true for your virtualization host servers too, if you have virtual DCs.

Set a strong BIOS password.

If you don't set a BIOS password, somebody can
update the boot order, boot to the Windows Server installation media or many freeware toolkits, perform a repair to get to a command prompt. Once at the command prompt, they can wreak some havoc and quickly reset passwords for domain accounts.

Keep the DCs in a locked cabinet.

Tech Savvy Productions

While a BIOS password is one layer of security, if
the attacker is semi-capable, he or she will likely know how to reset the BIOS so that the configuration resets and password is removed. Often, this requires gaining access to the motherboard. You can reduce the risk of such an attack by keeping DCs in a locked cabinet. Some servers also allow for chassis locks. In high security environments, you should opt for both.

You should try to match the configuration settings for each DC. You can accomplish some of this by using build automation through deployment tools such as System Center Configuration Manager. Items of interest for DCs are the event log size settings to ensure that you have large sizes to capture auditing and security related information, boot settings such as the timeout waiting for OS selection on physical servers, firmware and BIOS versions and settings, and hardware configuration. Of course, there are many other configuration items to standardize by using Group Policy.

The primary goal is to configure the DCs identically.

Standardize the configuration of all domain controllers.

SYSVOL

The system volume (SYSVOL) is a special directory on each DC. It is made up of several folders with one being shared and referred to as the SYSVOL share. The default location is %SYSTEMROOT%\SYSVOL\sysvol for the shared folder, although you can change that during the DC promotion process or anytime thereafter.

SYSVOL is made up of:

The folders are used to store:

- **Group Policy templates (GPTs**), which are replicated via SYSVOL container (GPC) is replicated via Active Directory replication

Tech Savvy Productions

- **Scripts, such as startup scripts that are referenced in a GPO.**
  Junction points. Junction points work like a shortcut. One directory can point to a different directory. In File Explorer, a junction point and a directory look and feel the same. You can view junction points by running the *dir /AL /S* command.

SYSVOL replication occurs over DFSR. Initially with Windows 2000 Server, Windows Server 2003, and Windows Server 2003 R2, replication was handled by File Replication Service (FRS).

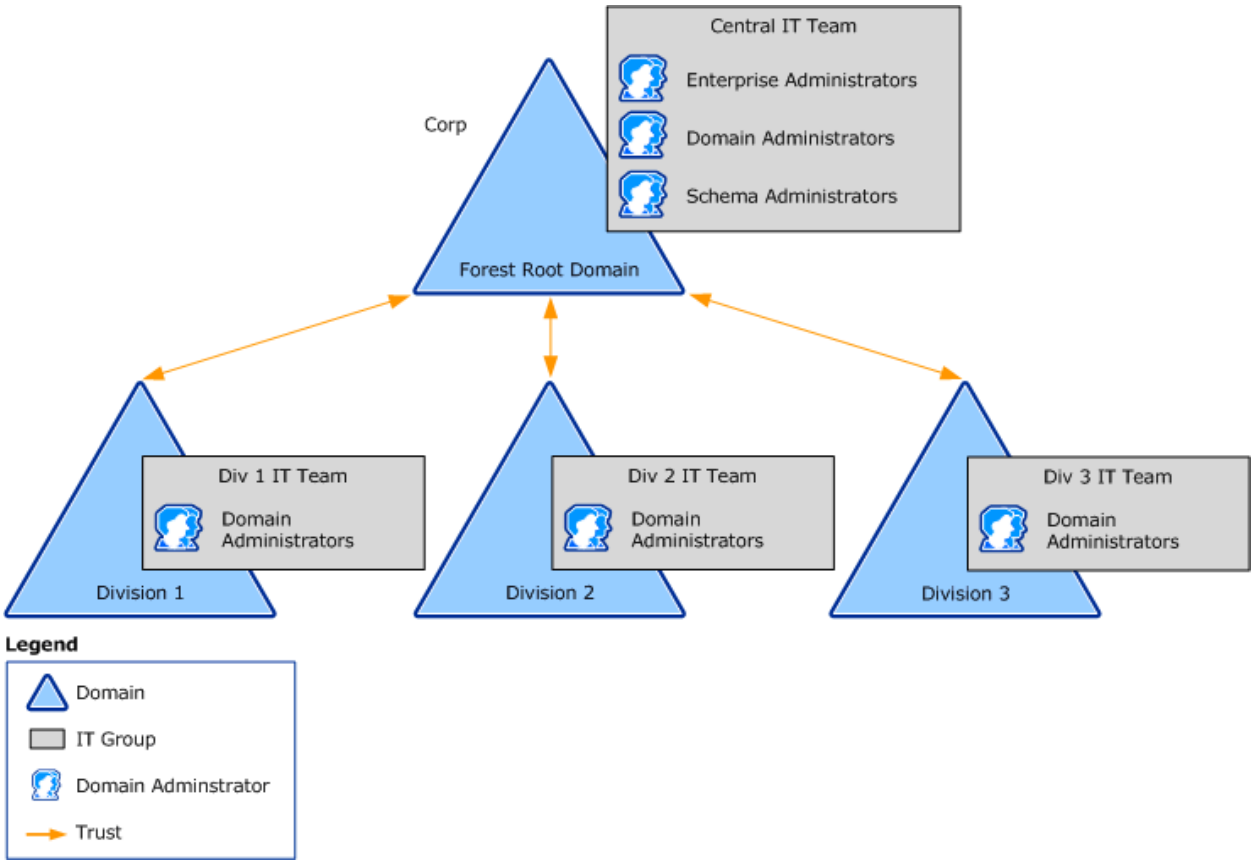
Starting with domains created in Windows Server 2008, DFSR is the default SYSVOL replication method. FRS wasn't very efficient. Any time that a file in SYSVOL changed, FRS replicated the entire file to all domain controllers. With DFSR, only the changed part of the file is replicated, although only for files over 64KB. DFSR uses Remote Differential Compression (RDC). RDC is what enables the replication of only changed data.

Some admins may remember migrating from FRS to DFSR when Windows Server 2008 was released. Without reliable and timely replication, one side effect that users may experience is inconsistent GPO application since the SYSVOL data may not be in sync across all of the DCs.

IV. Forests, Domains, Trusts

Forests and domains are fairly well understood by administrators. Trusts aren't though. In this section, we'll take a look at some portions of forests, domains, and trusts and discuss some good practices around them.

Tech Savvy Productions

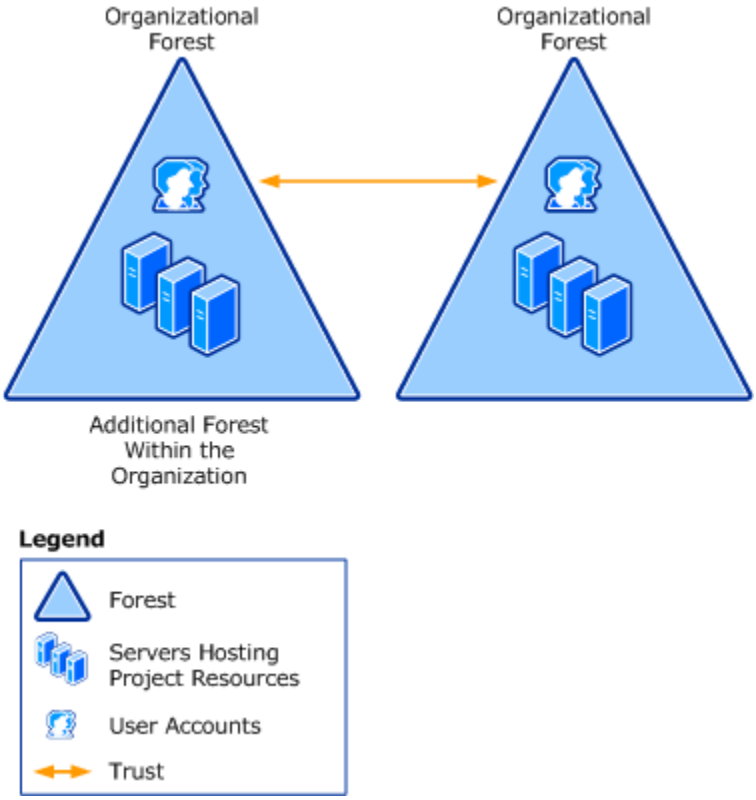


Forests

A forest is the top most logical container in an AD DS environment. It was first introduced with Active Directory in Windows Server 2000. A forest is made up of one or more domains and all of the objects in the domains. In the database, a forest is a just a container, similar to many of the objects below it such as domains and OUs. Importantly, the forest is the defined security boundary for an AD DS environment. In the early days of Active Directory, the domain was originally defined as the security boundary. Unlikely many of the other components that we discuss in this white paper, there aren't any direct limitations on the number of forests that you can deploy. Since they are the top most object, you can create as many as you want, assuming that you have enough physical servers or VMs (don't

Tech Savvy Productions

take this as a recommendation though!).



There are three forest-wide directory partitions in a forest:

- Schema.

The schema partition defines all of the classes, objects, and attributes that can be used. The schema is shared among all of the domains in the forest. Objects such as users, groups, and OUs are defined in the schema.

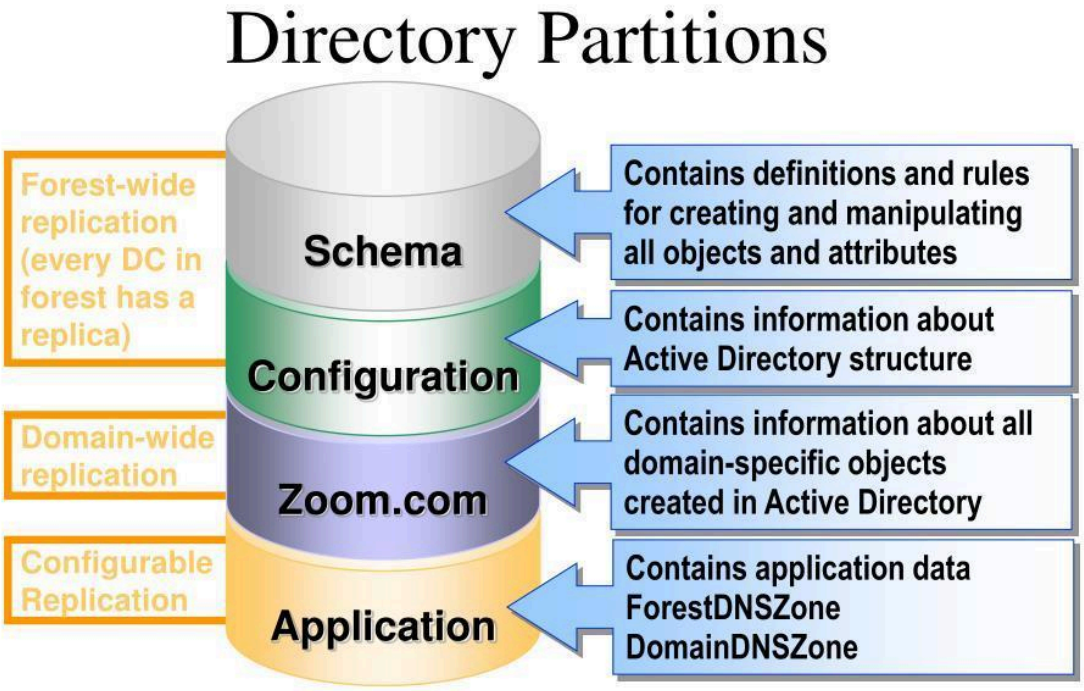
- Configuration

The configuration partition is responsible for managing the forest topology, forest settings, and domain settings. You can find a list of all of the domains, DCs, and GCs in the configuration partition. You can view the configuration partition in a domain named contoso.com by viewing *cn=configuration,dc=contoso,dc=com in ADSIEdit*.

- Application

Tech Savvy Productions

The application partition is used to store application data. A common example of data in the application partition is DNS.



Of the 5 FSMO roles, 2 of the roles are specific to the forest:

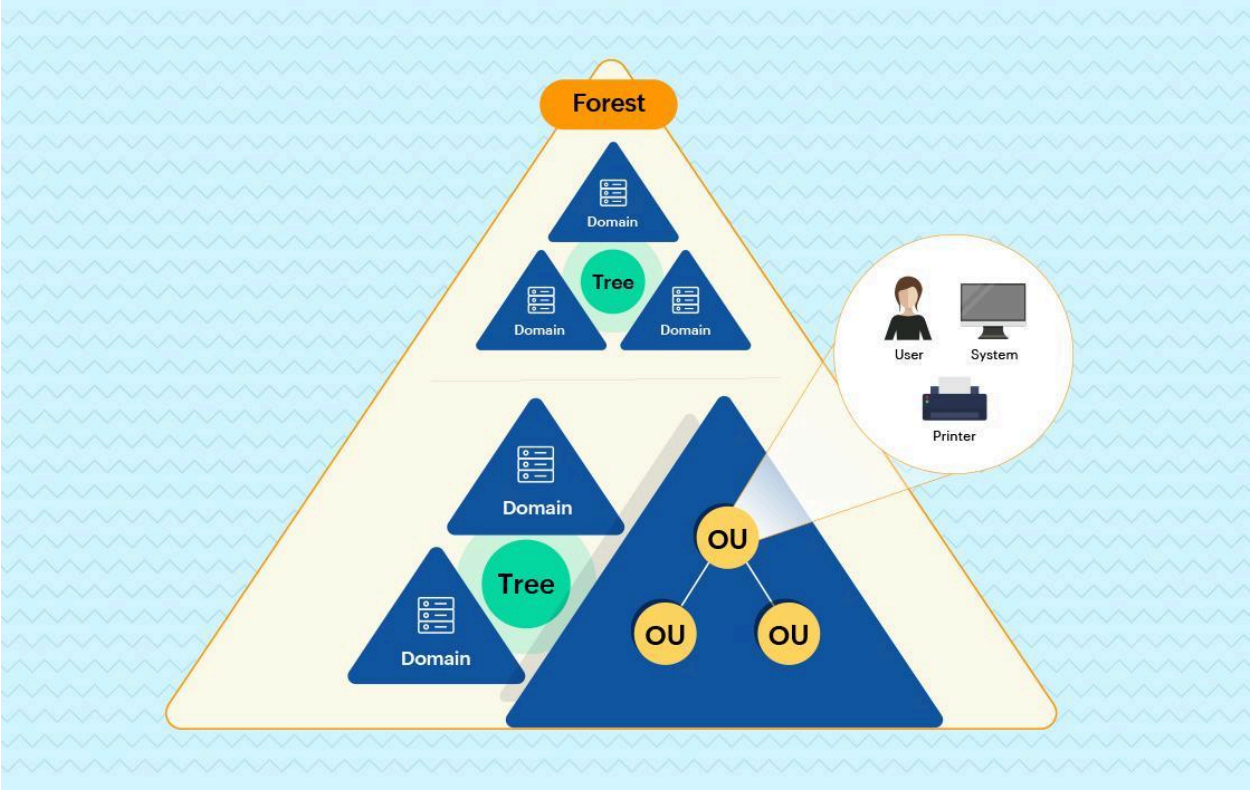
Schema Master.
This role is used for schema updates. As such, the role holder must be online and available to perform a schema update.

Domain Naming Master.
This role is used to add and remove domains for the forest. As such, the role holder must be online and available to perform domain additions and removals.

Forests, Domains, Trusts

There is a good amount of guidance around Active Directory forests published on the internet.

Tech Savvy Productions



Below are some of the recommended practices surrounding forests:

Always start with a single forest.

Then, if you have requirements that cannot be
met with a single forest implementation, begin adding forests as necessary. Better yet, go back and validate the requirements first.

Using multiple forests in a production environment is often unnecessary and adds management overheard and unneeded complexity.

With a backend technology that everybody expects to be always running, you should opt for a simple implementation that is implemented and maintained based on good practices, as opposed to a multi-forest implementation with a large number of domain controllers.
For many environments, a single production forest will meet or exceed requirements.

Additionally, it is a good idea to have a second non-production forest to use for development, testing, and quality assurance.

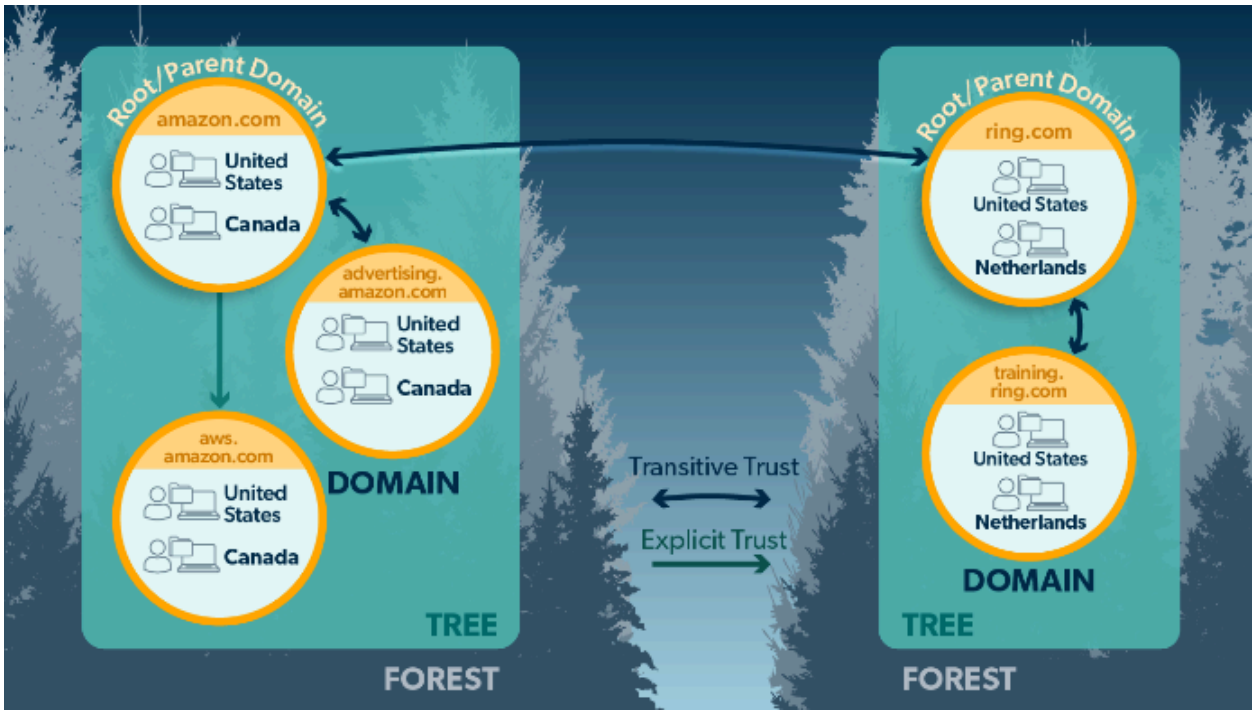
Tech Savvy Productions

Avoid the empty forest root domain.

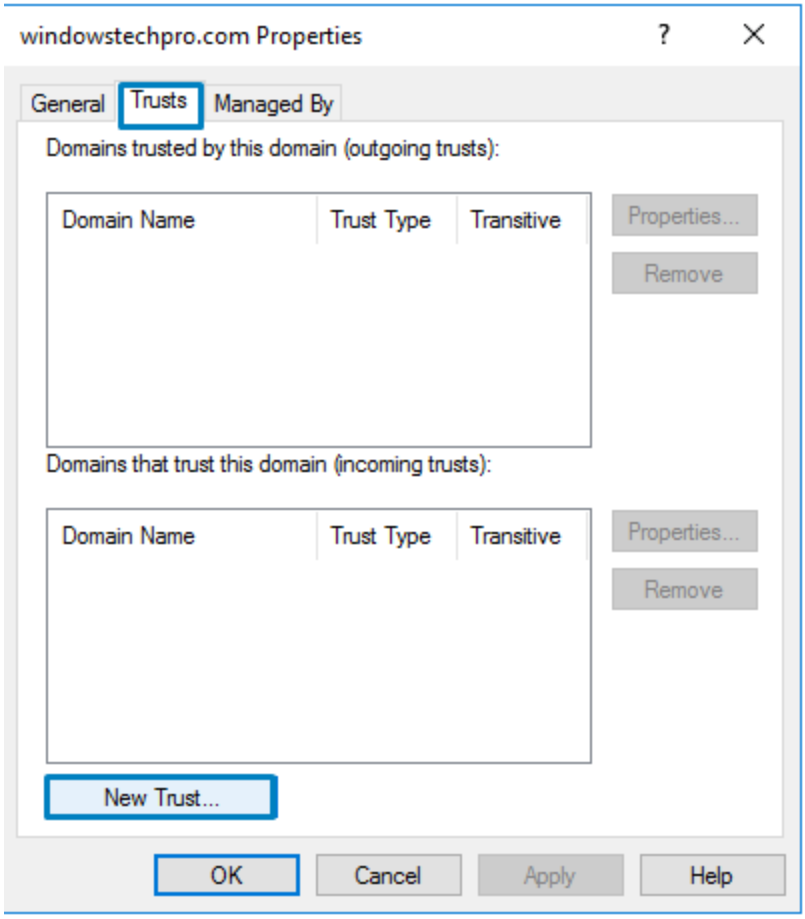
Upon initial release of Active Directory, Microsoft recommended using an empty forest root domain which would form a security
boundary for enterprise objects stored in the root domain such as the Enterprise Admins group. However, not long thereafter, the guidance changed and the empty forest root was no longer recommended by default. Administrators found that maintaining the empty forest root domain added to the administrative overhead of their environment without returning much value.

Today, the latest thinking is forest reduction. Minimize the total number of forests.

If using two-way forests trusts, consolidate forests.



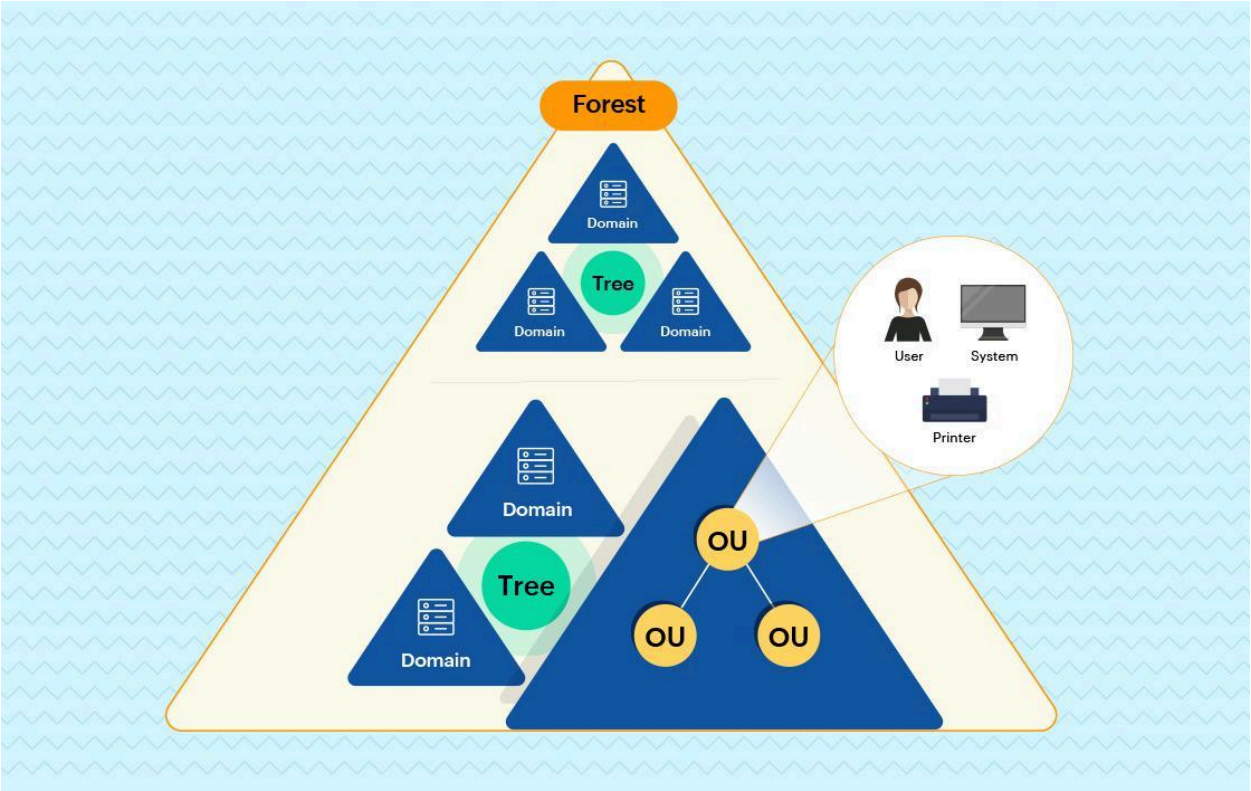
Tech Savvy Productions



Each forest that you maintain requires administrative overhead. In addition, each forest increases the complexity of your environment which also makes it harder to secure, maintain, and recover.

If you are using two-way trusts between forests, you should strongly consider consolidating forests because a two-way trust between forests is effectively a single forest with extra overhead.

Tech Savvy Productions



Active Directory Forests Best Practices

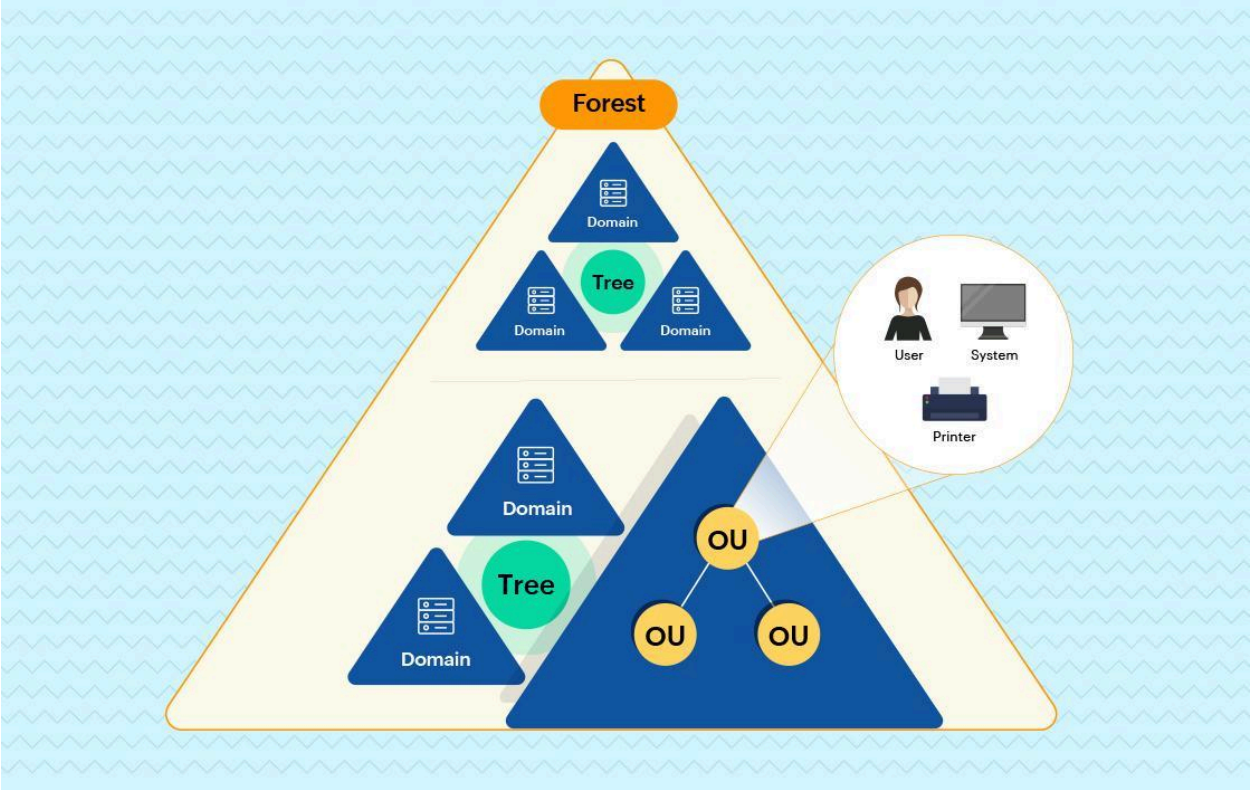
AD forests have been around since 2000, so there are many different theories about the best way to configure Active Directory and forests. Current best practices include:

- When possible, consolidate to a single forest
- Secure resources and data via GPO and apply a least privileged model
- Use GPOs to further limit users ability to create new folders without following a set process. The least privileged permissions model.
- Give your domain admins a 2nd admin account they use only when required per the change management process.
- If you have multiple AD forests with trust relationships, consider consolidation.
- If you need to create a restricted access forest, make sure it is truly restricted. As secure as we want the primary forest to be, a restricted access forest should be Castle Black. Put a 700' wall around it and keep it there.

Tech Savvy Productions

Domains

A domain is the logical container that sits directly below the forest container. Previous to Active Directory, there was a Windows domain that was part of previous Windows Server versions. It had similar core functionality as an Active Directory domain but without a forest above it. Historically, the beginning of the domain as we know it goes back to X.400 which is a telecommunications standard first recommended in 1984!



Each domain is contained in a single forest container. A domain houses other containers and objects below it. In the early days of Active Directory, the domain was originally defined as the security boundary. However, that definition has been updated and now the forest is defined as the security boundary. That was a key change that went unnoticed by some administrators.

From a scalability perspective, you can have a very large number of domains in a single forest, as follows:

Tech Savvy Productions

Once you use the Windows Server 2003 forest functional level or a higher level, a single forest can support up to 1,200 domains.

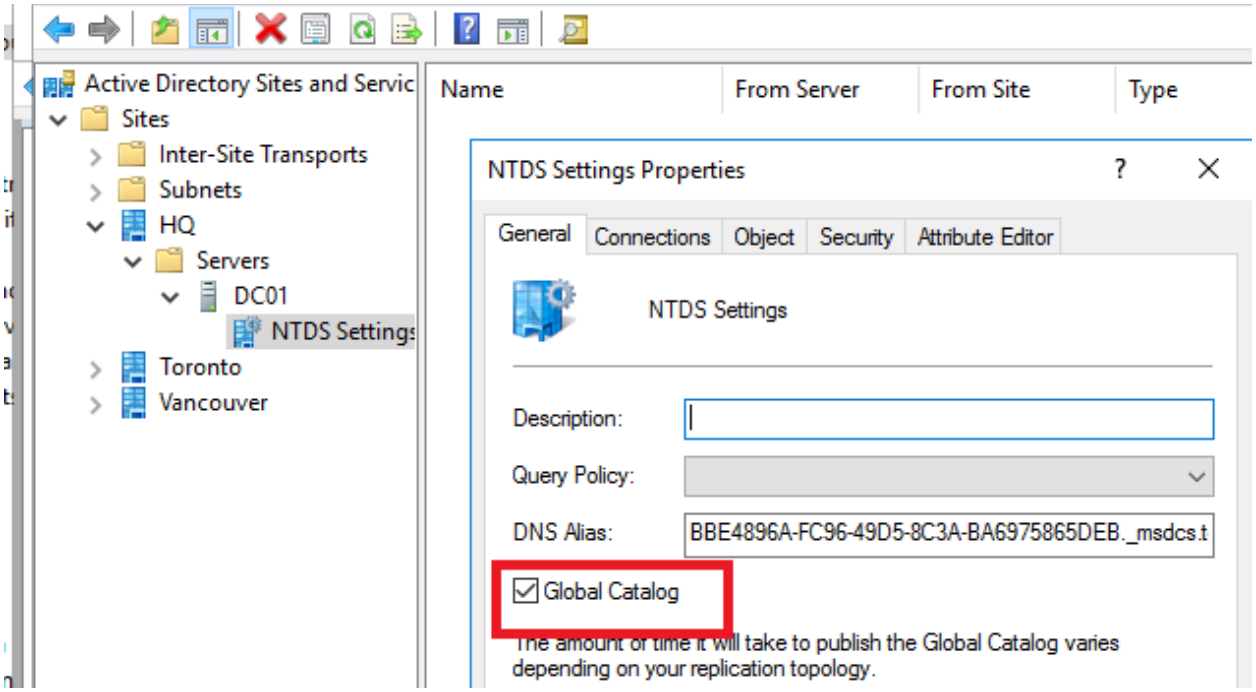
Several components work together in a domain. A domain includes the following components:

- Schema
- Global catalog
- Replication service
- Operations master roles

The schema, defined earlier in the Forest section, defines objects that are used in a domain. These can be both physical and logical objects. For example, a physical computer is represented by a computer account object, while a subnet is represented by a subnet object. Objects have many attributes. Object attributes define the properties, limits, and format of the objects. Attributes can be multi-valued, strings, integers, Boolean (true or false), or many other types. The specific attributes that an object has is defined by the schema.

A global catalog server stores information about every object within a domain. Administrators and users query a global catalog server to find information about objects. For example, if an administrator needs to look up information about a user account, including address, phone number, and office location, he would query the global catalog server to retrieve the information.

Tech Savvy Productions



The operations master roles, also known as **flexible single master operations** (FSMO) roles, perform specific tasks within a domain. The five FSMO roles are:

- Schema Master
- Domain naming Master
- Infrastructure Master
- Relative ID (RID) Master
- PDC Emulator

In every forest, there is

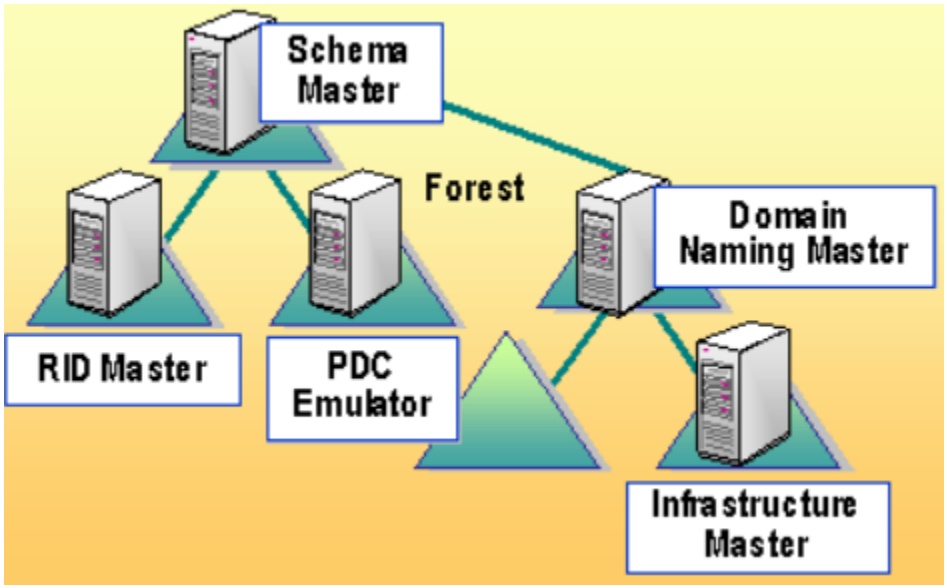
Tech Savvy Productions

a single Schema and
Domain naming Master which are discussed in the Forest section of this e-book.

In each domain,

- 1 Infrastructure Master
- 1 RID Master
- 1 PDC Emulator

At any given time, there can only be one DC performing the functions of each role. Therefore, a single DC could be running all five FSMO roles, however, **there can be no more than five servers in a single-domain environment that run the roles**.



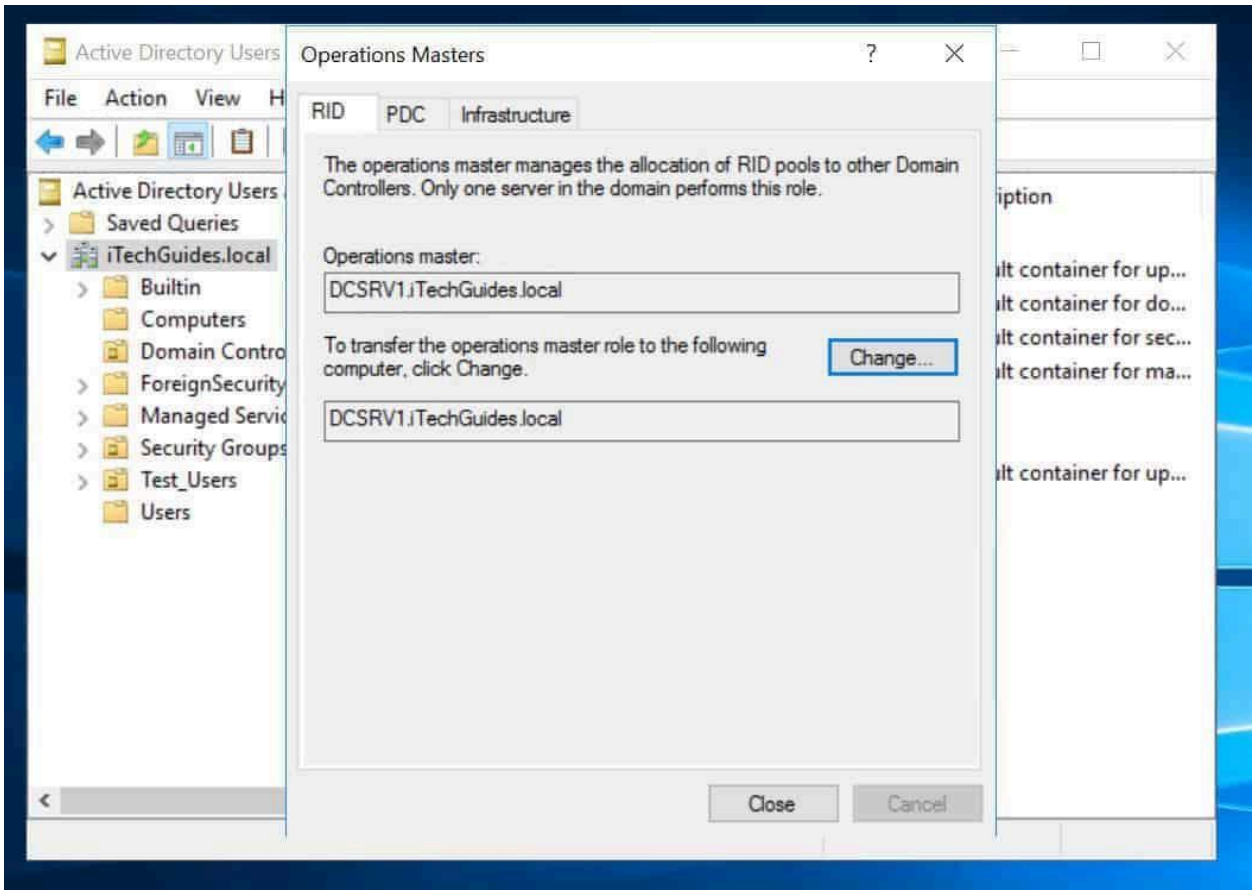
For additional domains, each domain will contain its

- own Infrastructure Master
- RID Master

Tech Savvy Productions

- PDC Emulator

Windows 2000 Server. Upon initial release, Active Directory supported up to 800 domains in a single forest.



The RID Master provisions RIDs to each DC in a domain. New objects in a domain, such as a user or computer object, receive a unique security identifier (SID). The SID includes a domain identifier, which is unique to each domain, and a specific RID for each object. Combining the two ensures that every object in the domain has a unique identifier, but contains both the domain SID and the RID.

The PDC Emulator controls authentication within a domain, whether Kerberos v5 or NTLM. When a user changes their password, the change is processed by the PDC Emulator.

Tech Savvy Productions

Finally, the Infrastructure Master synchronizes objects with the global catalog servers. The infrastructure Master will compare its data to a global catalog server's data and receive the data not found in its database from the global catalog server.

If all DCs in a domain are also global catalog servers, then all DCs will have up-to-date information, assuming that replication is functional. In such a scenario, the location of the Infrastructure Master role is irrelevant since it doesn't have any real work to do.

Trusts

A trust is a relationship between forest and/or domains. In a forest, all of the domains trust each because a two-way transitive trust is created when each domain is added.

This allows authentication to pass through from one domain to any other domain in the same forest.

Tech Savvy Productions

You can create trusts outside of the forest too with other AD DS forests and domains or Kerberos v5 realms. Back in the days of Windows NT 4.0, there wasn't a forest or a hierarchical structure. If you had multiple domains, you had to manually create trusts between them.

With Active Directory, you automatically have two-way transitive trusts between domains in the same forest. Back with Windows NT 4.0, you had to use NetBIOS to establish trusts too! Luckily, things have come a long way and now we've got additional trust functionality, especially around securing trusts with selective authentication and SID filtering.

Each trust in a domain is stored as a *trustedDomain* object (TDO) in the System container. Thus, to find and list all of the trusts and trust types in a domain named contoso.com, run the Windows PowerShell command:

Get-ADObject –SearchBase "cn=system,dc=contoso,dc=com" –Filter * -Properties trustType | where {$_.objectClass –eq "trustedDomain"} | select Name,trustType

There are 4 valid values for the *trustType* attribute. However, only the value 1 (indicating a trust with an NT domain) and the value 2 (indicating a trust with an Active Directory domain) are common. There is a lot of other good information about trusts stored in the *trustedDomain* object. In a domain named contoso.com, run the Windows PowerShell command to look at all of the trust properties:

Get-ADObject –SearchBase "cn=system,dc=contoso,dc=com" –Filter * -Properties * | where {$_.objectClass –eq "trustedDomain"} | FL

You can also view many of the core properties of a trust by running the *Get-ADTrust –Filter ** command. The table below shows the trust properties and a description of each property.

DisallowTransivity I think this is a Microsoft typo as it really should be *"DisallowTransitivity"*. This can be set to True or False based on whether the trust disallows tranitivity.

Direction

Valid values are bidirectional, inbound, or outbound. Note that the direction is

Tech Savvy Productions

relative to the
domain in which you are running the query.

Property description

DistinguishedName The DN of the trusted domain object.
ForestTransitive This is set to True when a forest trust is transitive and False when a forest trust is
non-transitive.
IntraForest This is set to True when a trust is between domains in the same forest or set to False when
a trust is between domains in different forests.
IsTreeParent Valid values are True and False.
IsTreeRoot Valid values are True and False.
Name The name of the domain that is part of the trust, not the domain where the query is run.
ObjectClass This is set to trustedDomain for trusts.
ObjectGUID
Globally unique identifier for the trust.
An example is *de207451-51ed-44cd-4248- 85ad9fcb2d50*.
SelectiveAuthentication Set to True if the trust is configured for selective authentication or False if it isn't.
SIDFilteringForestAware Set to True if a forest trust is configured for selective authentication.
SIDFilteringQuarantined Set to True when SID filtering with quarantining is used for a trust. Used for external trusts
only.

Target Set to the domain name of the other side of the trust.
Source Set to the DN of the trust root. In a forest trust, the DN of the root domain of the forest is the
source.
TGTDelegation Set to True if Kerberos full delegation is enabled on outbound forest trusts. Default is False.
TrustAttributes Set to a numerical value indicating the trust configuration.
TrustedPolicy Undocumented.
TrustingPolicy Undocumented.
TrustType
Set to Uplevel for trusts with Active Directory forests and domains, DownLevel for

Tech Savvy Productions

trusts
pre-Active Directory domains such as NT 4 domains, Kerberos realm for trusts with Unix/Linux realms.
UplevelOnly Set to True if only Windows 2000 and later operating systems can use the trust link.
UsesAESKeys Set to True for realm trusts that use AES encryption keys.
UsesRC4Encryption Set to True for realm trusts that use RC4 encryption keys.

From a scalability perspective, there are a couple of things about trusts that you should be aware of:
Maximum number of trusts for Kerberos authentication.

If a client in a trusted domain attempts to access a resource in a trusting domain, the client can't authenticate if the truth path has more than 10 trust links. In environments with a large number of trusts and long trust paths, you should implement shortcut trusts to improve performance and ensure Kerberos authentication functionality.
Performance deteriorates after 2,400 trusts.

In really large and complex environments, you may have an enormous number of trusts. After you reach 2,400 trusts, any additional trusts added to your environment could significantly impact performance over the trusts, especially related to authentication.

Group Policy

Group Policy provides a method of centralizing configuration settings and management of operating systems, computer settings, and user settings in an environment. While these settings are managed by using Group Policy Objects (GPOs), GPOs cannot be applied directly to user or computer objects. A GPO must be applied to a domain, site, or organizational unit.

By default, all objects within the container that the GPO has been applied to will receive the GPO settings. Child objects and containers will also receive the configured settings through inheritance, unless inheritance blocking is

Tech Savvy Productions

configured.

Blocking inheritance complicates configurations and can cause unexpected results. Additionally, GPOs may be set to be enforced which ensures that GPOs will always be applied, regardless of inheritance settings. The enforced setting should also be used with caution.

- How to Search Group Policy for a Specific Setting in Windows 10 and Windows 11: https://techdirectarchive.com/2022/09/30/how-to-search-group-policy-for-a-specific-setting-in-windows-10-and-windows-11/

Group Policy processes policy settings in the following order:

1. Local Group Policy
2. Site-linked policies
3. Domain-linked policies
4. OU-linked policies

Any policy that is configured by two or more GPOs will be overwritten or modified by the last GPO that is processed. For example, if a site policy is applied that modifies system settings, and an OU policy modifies the same system settings, then the OU policy will take precedent because it is processed last. Additionally, if a policy is enforced, the settings that are defined by that specific policy cannot be overwritten by a subsequent GPO, even if the other GPO is processed last.
GPOs that modify computer settings are applied at computer startup. Settings that are for users are applied at the time that the user logs on. By default, GPOs are processed synchronously. Synchronous processing ensures that all settings are applied before the user completes the log on process. Alternatively, asynchronous processing can
be configured, to allow multiple operations to be performed. However, asynchronous processing can cause undesired effects. For example, if a policy is configured to remove Start Menu options for a user, the user could log on (and have access to the start menu) before the policy is applied. By default, GPOs are also reapplied every 90 minutes, with a randomized offset of up to 30 minutes. For domain controllers, the policies are refreshed every 5 minutes.

Tech Savvy Productions

Both of these refresh settings can be configured by using Group Policy.

If you do not wish to wait for a policy to refresh automatically, the ***gpupdate.exe*** command can be used locally, or with switches, remotely. The gpupdate.exe command processes the policies and applies only the settings that have been changed since the last refresh. Additionally, other command switches can be used to force applying all settings, specifying only user or computer settings, logging off, or restarting a computer after applying the settings.

To troubleshoot or view the applied policies, the ***gpresult.exe*** command can be used. The *gpresult.exe* command also has switches, allowing you to view the final applied policy in HTML format. The gpresult.exe command can also be ran remotely against target computers by name or by IP address. You can also specify specific user accounts to see the settings that would be applied if that user were to log on.

Tech Savvy Productions

```
C:\Users\Dutch>gpresult.exe /r

Microsoft (R) Windows (R) Operating System Group Policy Result tool v2.0
© Microsoft Corporation. All rights reserved.

Created on  10/ 4/ 2022 at 10:31:37 AM


RSOP data for VMDESKTOP\Dutch on VMDESKTOP : Logging Mode
-----------------------------------------------------

OS Configuration:            Member Workstation
OS Version:                  10.0.22000
Site Name:                   Default-First-Site-Name
Roaming Profile:             N/A
Local Profile:               C:\Users\Dutch
Connected over a slow link?: No


COMPUTER SETTINGS
------------------

    Last time Group Policy was applied: 10/4/2022 at 8:46:07 AM
    Group Policy was applied from:      Win2016DC1.HomeLab.TechSavvyProductions.com
    Group Policy slow link threshold:   500 kbps
    Domain Name:                        HOMELAB
    Domain Type:                        Windows 2008 or later

    Applied Group Policy Objects
    -----------------------------
        Enable WinRM
        Default Domain Policy
        Local Group Policy

    The computer is a part of the following security groups
    -------------------------------------------------------
        BUILTIN\Administrators
        Everyone
        BUILTIN\Users
        NT AUTHORITY\NETWORK
        NT AUTHORITY\Authenticated Users
        This Organization
        VMDESKTOP$
        Domain Computers
        Authentication authority asserted identity
```

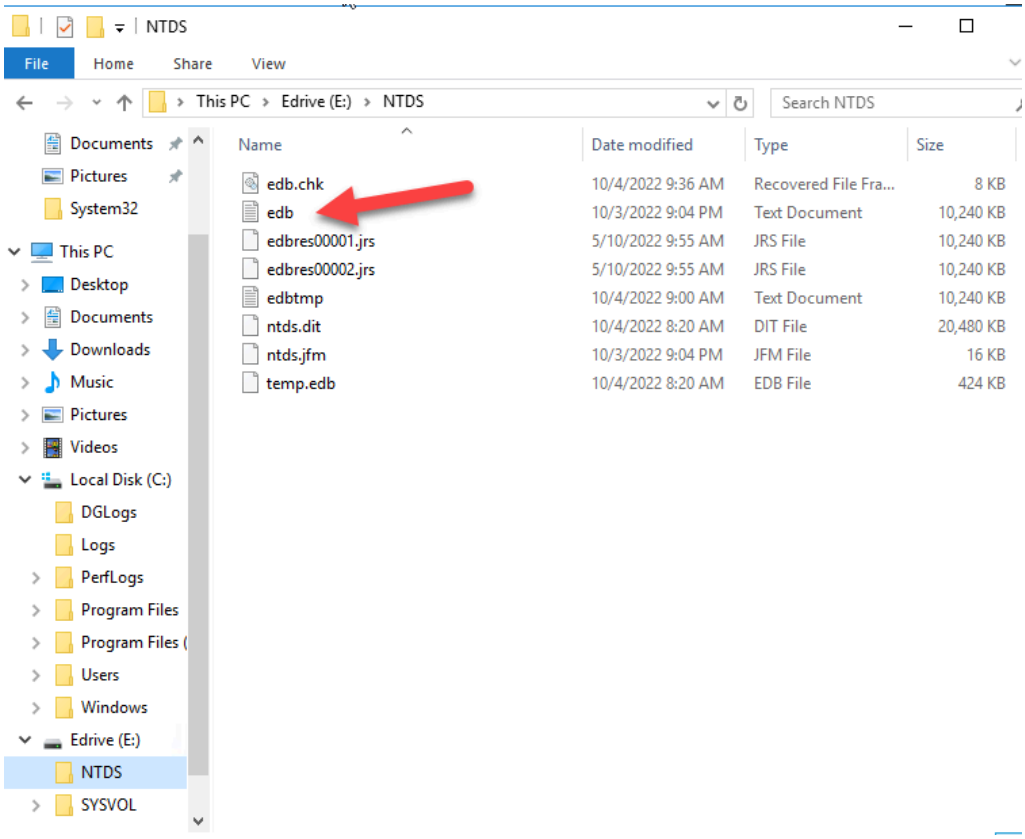

Tech Savvy Productions

Inside the AD DS Database

The Active Directory database is made up of a single file named *ntds.dit*. By default, it is stored in the *%SYSTEMROOT%\NTDS* folder. The folder also contains the following related files:

Edb.log. There are typically multiple log files starting with *"edb"* such as *edb0013A.*

log and *edb0013B.log*. Additionally, there is the *edb.log* file which is the active log file. These logs are the transaction logs used to record changes made in AD DS. All changes are first written to a transaction log and eventually make their way into the
database a short time later.



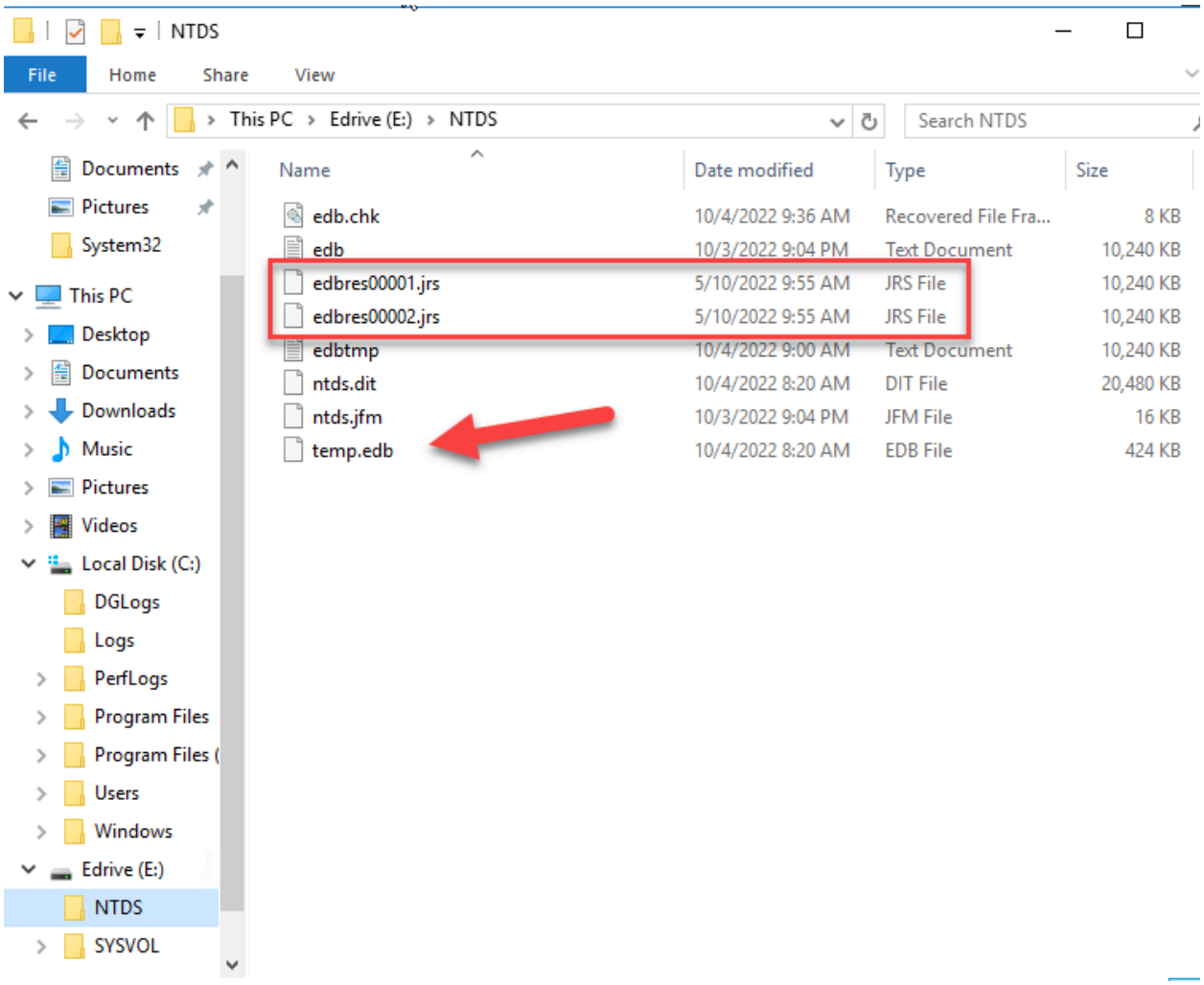
Temp.edb. As the name implies, this file is a temporary file used to track

Tech Savvy Productions

transactions that are taking place. It is also used when you run a database compaction job.

Res1.log and res2.log or edbres00001.jrs and edbres00002.jrs.

These log files are
each 10MB in space and used in a situation where you are critically low on disk space on the system volume. In older versions of Windows Server, the *res1.log* and *res2.log* files are used. Since Windows Server 2008, the *"edbres"* naming is used, along with a new file extension of *.jrs*.



The **Active Directory database** is based on Microsoft's Joint Engine Technology (JET) which is a database engine that was developed in 1992. Microsoft Access is also based on the JET technology. Over the years, there have been rumors that

Tech Savvy Productions

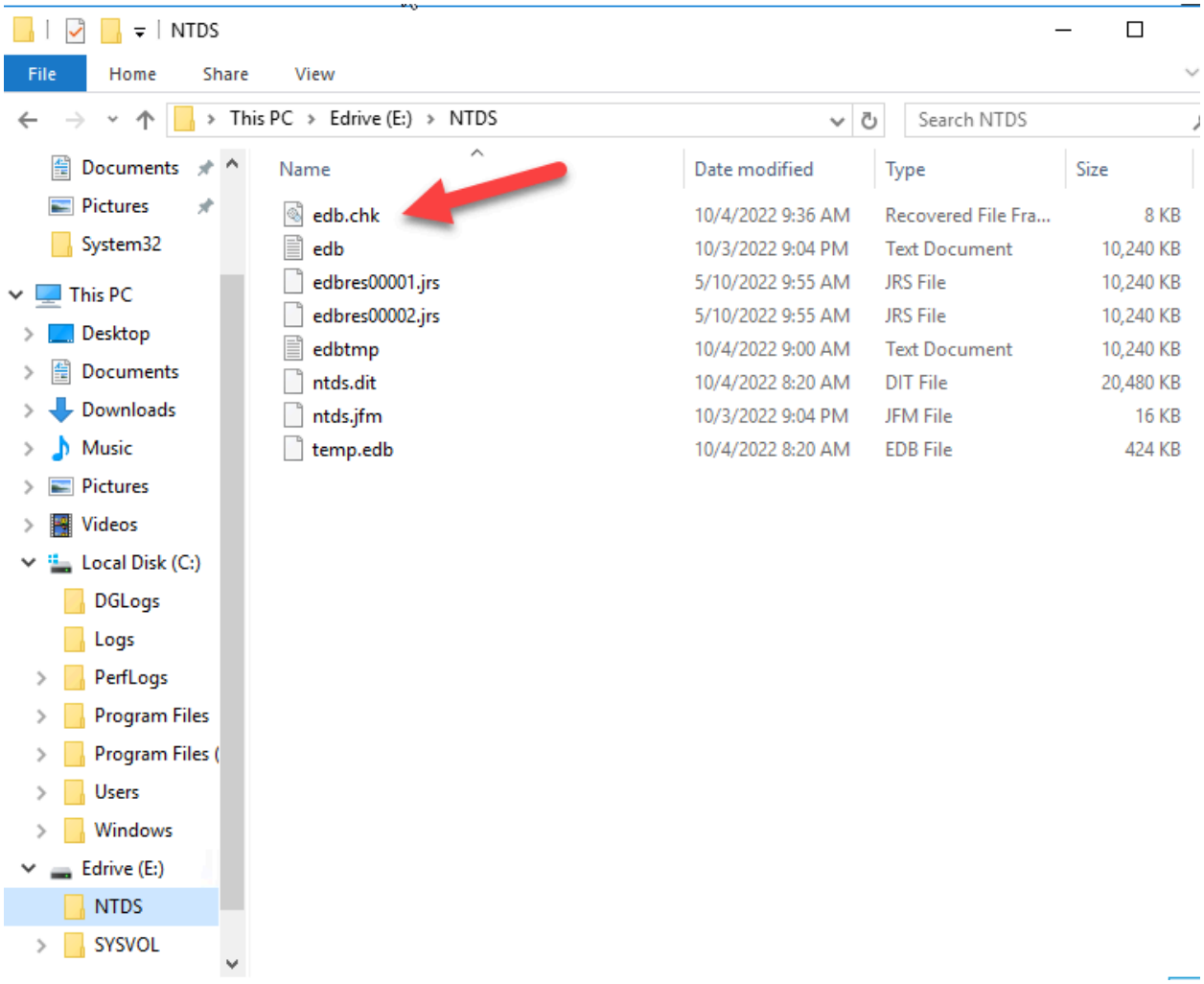
Active Directory's database would be moved over to SQL Server (similar to rumors for Microsoft Exchange) but so far, that doesn't seem likely. I've heard third-hand that SQL was tested as the AD DS database engine but that performance issues prevented it from becoming the database standard. Because AD DS is a single use database, it can effectively run on JET technology (whereas JET technology may not be a good fit for the majority of transactional database needs which often have multiple uses).

Microsoft chose to use the Indexed Sequential Access Method (ISAM) model for indexing data in the AD DS database. To work with the data, including transferring data in and out of the database, the Extensible Storage Engine (ESE) is used. ESE helps to maintain a consistent, and therefore optimal, database, especially in the event of a system crash. ESE is sometimes called JET Blue and is used by other technologies besides Active Directory including Microsoft Exchange, Windows Server's BranchCache, and Microsoft's Desktop Search.

The database technologies for Active Directory have been around a long time. Each technology, by itself, could account for several pages of text to dive into how they work.

Edb.chk. This file is a checkpoint file. Checkpoint files are commonly used in a transactional database system to keep track of which log file entries have been committed to the database. This is useful during a system crash to avoid data loss.

Tech Savvy Productions



Extensible Storage Engine Architecture at https://technet.microsoft.com/en- us/library/aa998171(v=exchg.65).aspx

How the Active Directory data store really works (inside NTDS.dit) – a blog by Christoffer Andersson at https://technet.microsoft.com/en-us/library/aa998171(v=exchg.65).aspx

Replication

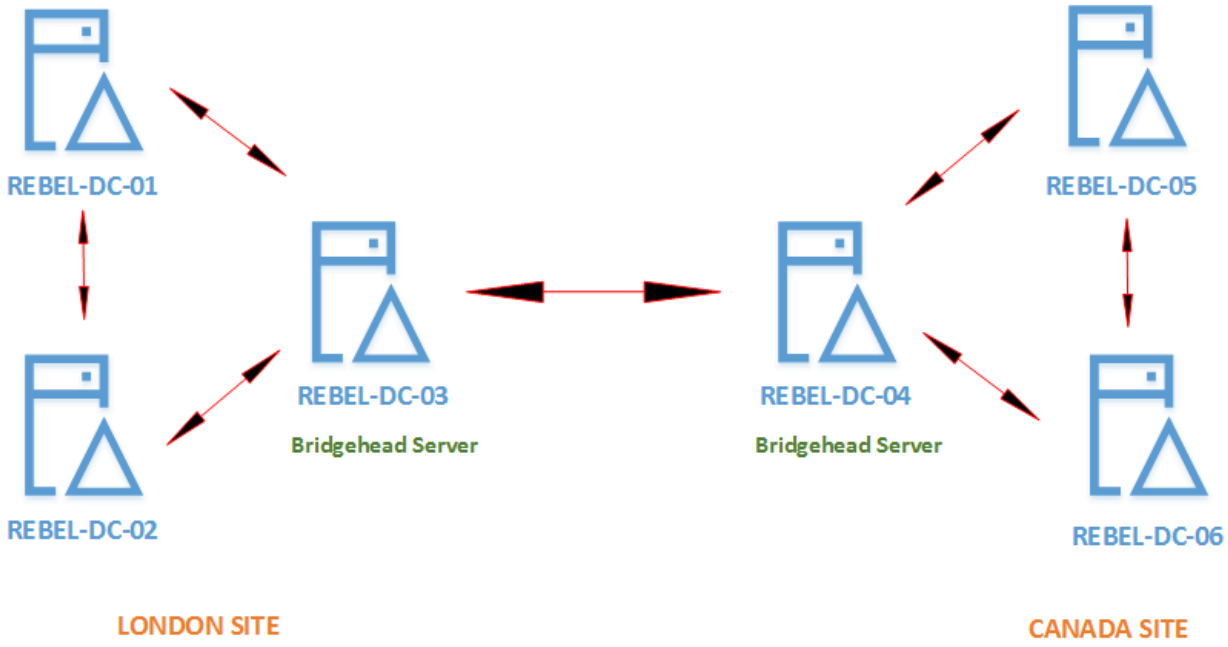
Active Directory replication is the method of transferring and updating Active Directory objects from one DC to another DC. The connections between DCs are built based on their locations within a forest and site. Each site in Active Directory contains one or more subnets, which identify the range of IP addresses associated with the site. By mapping the IP address of a DC to a subnet, Active Directory

Tech Savvy Productions

knows which DCs are in which site. Connections are configured between sites to ensure that Active Directory objects are replicated between sites.

Active Directory replication relies on the following technologies to operate successfully:

DNS
Remote procedure call (RPC)
SMTP (optional)
Kerberos
LDAP



Multimaster replication. Multimaster replication, compared to single-master replication as used in Windows NT 4.0, ensures that each domain controller can receive updates for objects for which it is authoritative. This provides fault tolerance within an Active Directory environment.

Pull replication. Pull replication ensures that DCs request object changes instead of changes being pushed (especially unnecessarily). Pulling slightly reduces replication traffic between DCs.

Tech Savvy Productions

Store-and-forward replication. Store-and-forward replication ensures that every DC communicates with a subset of DCs to transfer the object changes that have occurred. With store- and-forward, every DC would communicate with every other DC, which is inefficient. Store-and- forward replication balances the replication load among the DCs within an Active Directory environment.

There are four main components of replication in Active Directory:

- State-based replication. State-based replication ensures that each DC tracks the
  state of replication updates which eliminates conflicts and unnecessary replication.

Replication is managed by the Knowledge Consistency Checker (KCC). The KCC manages replication between DCs in a single site by using automatically created connections. The KCC reads configuration data and reads and writes connection objects for DCs. The KCC only uses RPC to communicate with the directory service.

Intrasite replication does not use compression and changes are sent to DCs immediately. However, intersite replication relies on user-defined links that must be created. The KCC uses these links to create a topology so that replication is managed across the site-to-site links. Site connections can be controlled on a schedule and the replication data is compressed to minimize bandwidth usage.

The default replication schedule for site-to-site connections is 180 minutes which is usually way too long for the vast majority of organization. This can be configured to as low as 15 minutes in the GUI, and even faster by modifying the registry. A replication packet size is calculated based on the amount of RAM in the DC. By default, the packet size limits are 1/100th the size of RAM, with a minimum of 1 MB and a maximum of 10 MB. Additionally, the maximum number of objects in a packet is 1/1,000,000th the size of the system RAM, with a minimum of 100 objects, and a maximum of 1,000 objects. Therefore, in modern servers that have more than 1 GB or RAM, replication packet sizes will either

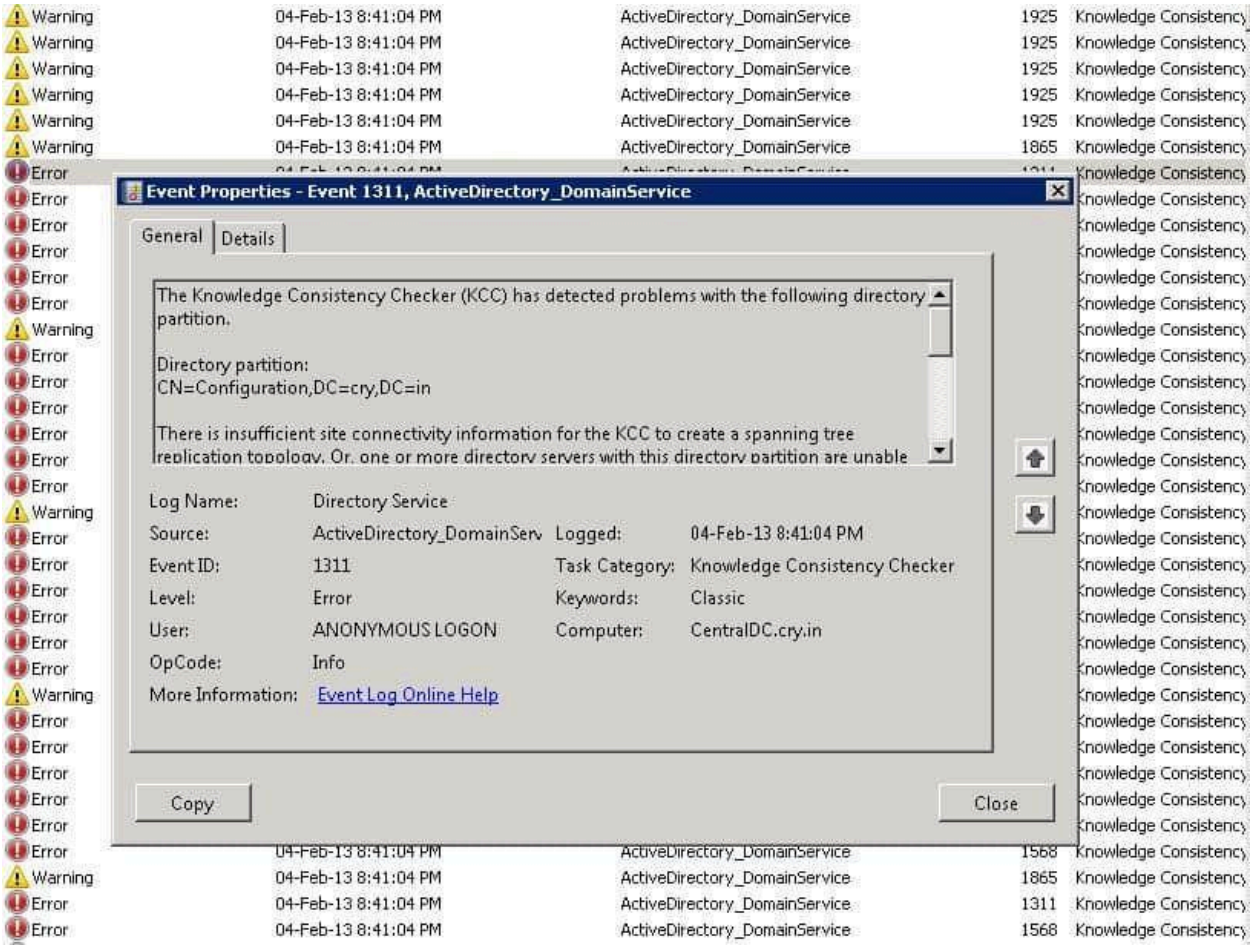
Tech Savvy Productions

contain up to 10 MB of data or up to 1,000 objects. .The maximum packet size and object limit can be configured by modifying the registry in the following location: HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\NTDS\Parameters

The following are components the primary replication components:

Knowledge Consistency Checker (KCC).

The KCC is a process that runs on each DC
and communicates directly with Ntdsa.dll to read and write replication objects.



Directory System Agent (DSA).

The DSA is a directory service component that runs as Ntdsa.dll on each DC. It provides an interface for services and processes to read the directory database.

Extensible Storage Engine (ESE).

The ESE manages directory database records, which may contain one or more columns.

Remote Procedure Call (RPC).

Directory replication is communicated by using the RPC protocol.

RPC is a communication protocol that allows developers to execute code on a local or remote system without having to develop specific code for remote execution. The KCC also uses RPC to communicate with DCs to request information when building a replication topology.

Intersite Topology Generator (ISTG).

The ISTG manages the intersite inbound replication connection objects for a specific site. There is one ISTG server in each site.

By default, the first DC in each site is the ISTG. To find the ISTG in a site named HQ in a domain named tailspintoys.com, you can run the Windows PowerShell command:

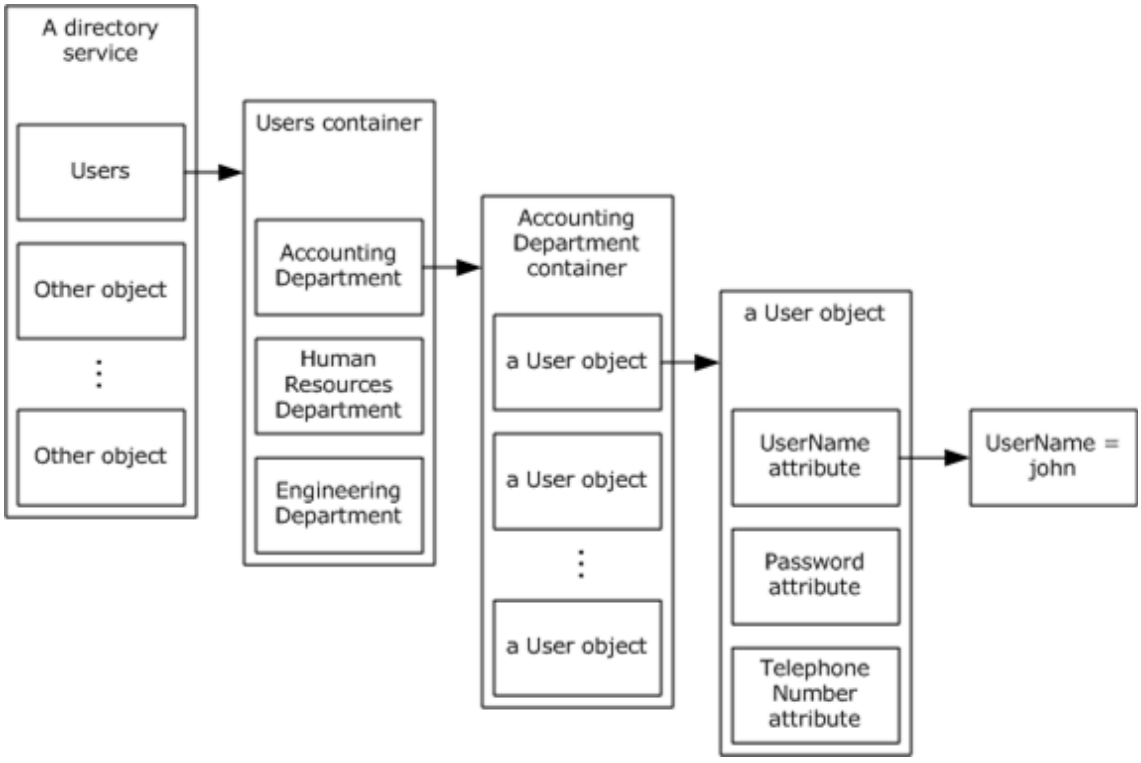
Get- ADObject –Identity "cn=NTDS Site Settings, cn=HQ,cn=sites,cn=configuration,dc=tailspintoys,dc=com" -Properties interSiteTopologyGenerator | Select interSiteTopologyGenerator

The Active Directory objects that are used by the KCC and its components include:
The diagram below shows a typical two-site Active Directory environment with

Tech Savvy Productions

some of the replication components.

A typical two-site Active Directory environment Beginning with Windows PowerShell in Windows Server 2012, there are 25 cmdlets to specifically manage Active Directory replication. These cmdlets offer functionality such as viewing replication information, configuring sites, managing site links, and forcing replication to occur.



Tech Savvy Productions

The **RepAdmin.exe command line tool** is also available to provide information and configure Active Directory replication.

```
C:\Users\Administrator>repadmin /showrepl

Repadmin: running command /showrepl against full DC localhost
Default-First-Site-Name\WIN2016DC1
DSA Options: IS_GC
Site Options: (none)
DSA object GUID: 28029095-585f-4851-bbd8-9a2e0f8ff03e
DSA invocationID: 28029095-585f-4851-bbd8-9a2e0f8ff03e

==== INBOUND NEIGHBORS ======================================

DC=HomeLab,DC=TechSavvyProductions,DC=com
    Default-First-Site-Name\WIN2016DC2 via RPC
        DSA object GUID: 60228dba-cbc8-4304-868a-28f53a302f46
        Last attempt @ 2022-10-04 10:50:55 was successful.

CN=Configuration,DC=HomeLab,DC=TechSavvyProductions,DC=com
    Default-First-Site-Name\WIN2016DC2 via RPC
        DSA object GUID: 60228dba-cbc8-4304-868a-28f53a302f46
        Last attempt @ 2022-10-04 10:50:55 was successful.

CN=Schema,CN=Configuration,DC=HomeLab,DC=TechSavvyProductions,DC=com
    Default-First-Site-Name\WIN2016DC2 via RPC
        DSA object GUID: 60228dba-cbc8-4304-868a-28f53a302f46
        Last attempt @ 2022-10-04 10:50:55 was successful.

DC=DomainDnsZones,DC=HomeLab,DC=TechSavvyProductions,DC=com
    Default-First-Site-Name\WIN2016DC2 via RPC
        DSA object GUID: 60228dba-cbc8-4304-868a-28f53a302f46
        Last attempt @ 2022-10-04 11:10:02 was successful.

DC=ForestDnsZones,DC=HomeLab,DC=TechSavvyProductions,DC=com
    Default-First-Site-Name\WIN2016DC2 via RPC
        DSA object GUID: 60228dba-cbc8-4304-868a-28f53a302f46
        Last attempt @ 2022-10-04 10:59:33 was successful.
```


Tech Savvy Productions

Another replication tool is the
Active Directory Replication Status Tool. It is available at
http://www.microsoft.com/en-us/download/details.aspx?id=30005 . You can use it to analyze and troubleshoot

Active Directory replication issues.

Sites.

Sites are Active Directory objects in the site class, which correspond to the subnets in a given site.

Subnets.

Subnet objects are in the subnet class, and define the network IP subnet that is corresponded with a site.

Servers.

A server object, in the server class, represents server computers, including

DCs.

Server objects are treated as security principals which are stored in a separate directory partition and have separate globally unique identifiers (GUIDs).

NTDS Settings.

NTDS Setting objects are in the nTDSDSA class, and represent an instance of Active Directory on a specific DC.

Tech Savvy Productions

Connections.

Connection objects are in the nTDSConnection class, and define a one-way, inbound route from a source DC to the DC that is storing the connection object.

Site Links.

Site Link objects are in the siteLink class, and identify the protocol and schedule to replicate data between two or more sites.

NTDS Site Settings.

NTDS Site Setting objects are in the nTDSSiteSettings class, and identify site-wide settings for Active Directory. There is only one NTDS Site Settings object per site in the Sites container.

Cross-reference.

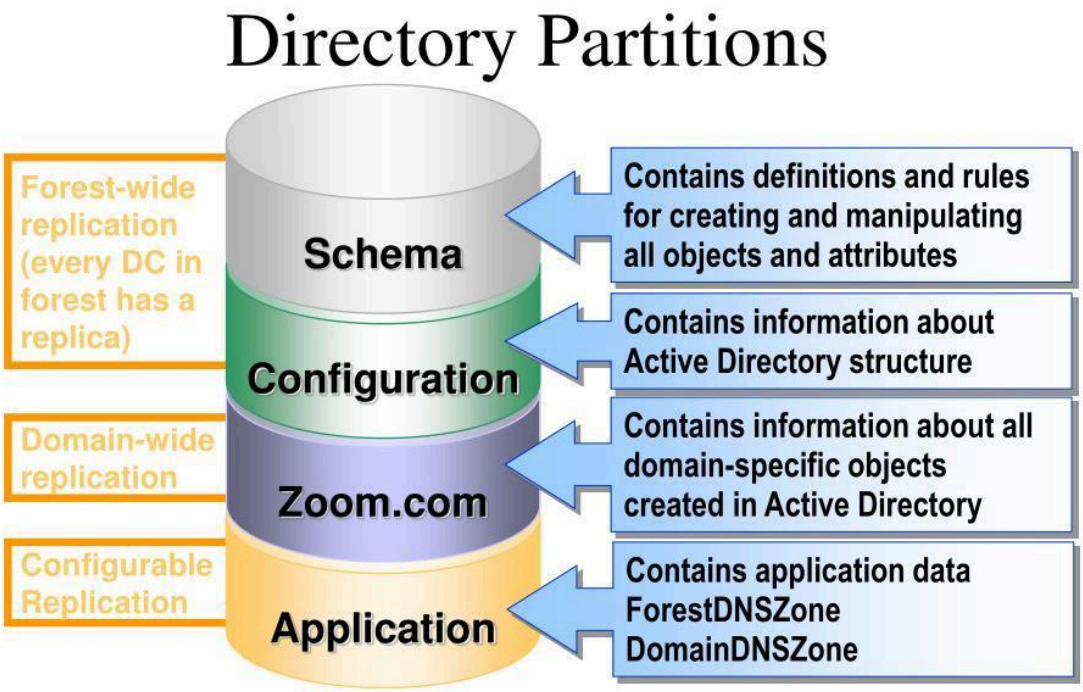
Cross-reference objects are in the crossRef class, and store the location of Active Directory partitions in the Partitions container.

Tech Savvy Productions

DNS and DHCP

AD DS provides a built-in method of storing and replicating DNS records by using Active Directory-integrated DNS zones. All of the records and zone data stored within the zone are replicated to other DNS servers by using the native AD DS replication service. Each DC stores a writable copy of the DNS zone data for namespaces for which they are authoritative. Active Directory-integrated zones also provide the ability to use secure dynamic updates, which supports controlling which computers may make updates and prevents unauthorized changes from being made.

DNS zone data is stored in an application directory partition. A forest-wide partition named ForestDnsZones is used for the zone data. For each AD DS domain, a domain partition is created named DomainDnsZones.



Tech Savvy Productions

Typically, DNS implementations are used with a contiguous namespace. For example, the Fully Qualified Domain Name (FQDN) of an AD DS domain might be corp.contoso.com, and the FQDN of a client in that domain would be client.corp.contoso.com.

However, AD DS and Active Directory-integrated DNS zones support disjoint namespaces. In such a scenario, the FQDN of the AD DS domain might be na.corp.contoso.com, while a client FQDN could be client.corp.contoso.com.

Notice that the "na" portion of the FQDN is not present in the client FQDN. There are several requirements and considerations when using a disjoint namespace.

For more information, see https://technet.microsoft.com/en-us/library/cc731125%28v=ws.10%29.aspx

AD DS requires DNS to function, and uses three specific components for the AD DS infrastructure:

Domain controller locator.

The Locator is implemented in the Net Logon service and provides the names of DCs in an AD DS environment. The Locator uses address (A) and service (SRV) DNS resource records to identify DCs in an AD DS environment.

Active Directory domain names in DNS.

The AD DS domain names in DNS are the FQDN that we discussed earlier.

Active Directory DNS objects.

While DNS domains and AD DS domains typically have the same name, they are two separate objects with different roles. DNS stores zones and zone data required by AD DS and responds to DNS queries from clients.

AD DS stores object names and object records and uses LDAP queries to retrieve or modify data. DNS zones that are stored in AD DS have a container object that is

Tech Savvy Productions

in the dnsZone class. The dnsZone object has a DNS node, which uses the dnsNode class. Each unique name in a DNS zone has a unique dnsNode object. For AD DS, this also includes individual functions.

Therefore, one DC may have multiple roles, such as being a global catalog server, which is indicated in the dnsNode object.

As mentioned earlier, DCs are identified by the SRV records in a DNS zone. Components of AD DS are stored in DNS using the following format in the _msdcs subdomain: _Service.Protocol.DcType._msdsc.DnsDomainName. For example, the Lightweight Directory Access Protocol (LDAP) service of the Primary Domain Controller (PDC) in the contoso.com AD DS domain would be ldap._tcp.pdc.contoso.com.

The service and protocol strings use underscores (_) as a prefix to avoid potential collisions with existing resources or records in the namespace. The Net Logon service requires 17 different SRV records to perform lookups. A full list of SRV records can be found at https://technet.microsoft.com/en-us/library/cc759550%28v=ws.10%29.aspx

In addition to the SRV records, the Net Logon service also requires two A records for clients that may not be SRV- aware. This includes a record for the DnsDomainName, and a record for gc._msdsc.DnsForestName. This enables non-SRV-aware clients to look up a domain controller or global catalog server by using an A record.

DNS is susceptible to security threats, such as foot printing, denial-of-service attacks, data modification, and redirection. To mitigate these threats, DNS zones can be secured by using secure dynamic updates, restricting zone transfers, plus implementing zone delegation and DNS Security Extensions (DNSSEC). By using secure dynamic updates, computers will be authenticated through Active Directory, and security settings will be applied when performing a zone transfer.

Tech Savvy Productions

Additionally, zone transfers can also be restricted to specific IP addresses within the network. Zone delegation can be approached by using two methods. First, is to limit DNS changes to a single team or entity, with all changes tracked and approved. This method limits the amount of people making changes, but allows for a single point of failure.

Secondly, zones can be delegated to individuals who will be managing each component of a network or domain. While changes may still need to be approved and tracked, this spreads out risk among multiple people, and may limit damage if only one component becomes compromised. DNSSEC validates DNS responses by providing origin authority, data integrity, and authenticated denial of existence. Windows Server 2012's implementation of DNSSEC meets the standards for RFC 4033, 4034, and 4035.

There are six resource record types that are used specifically with DNSSEC:
Resource record signature (RRSIG)
Next Secure (NSEC)
Next Secure 3 (NSEC3)
Next Secure 3 Parameter (NSEC3PARAM)
DNS Key (DNSKEY)
Delegation Signer (DS)

For more information on each of the record types and their use, see https://technet.microsoft.com/en- us/library/jj200221.aspx

Tech Savvy Productions

DHCP is another network service that is used by Windows Server. In an AD DS environment, DHCP servers must be authorized before they can lease IP addresses to clients on a network. DHCP servers are authorized by their IP addresses, and will be checked against AD DS to verify that it is authorized to lease IP addresses.

If an unauthorized DHCP server detects an authorized DHCP server, the unauthorized DHCP server will stop leasing addresses to clients. In an AD DS environment, the DHCP service must be installed on a server that is a member of the domain, or it cannot be authorized. Installing and running the DHCP service on a stand-alone server is supported, but must be on a separate network or VLAN than any authorized DHCP server.

To authorize a DHCP
server, the administrator must be a member of the Enterprise Admins built-in security group. However, the right to authorize DHCP server may be delegated to other administrators within the domain. To authorize a DHCP by using it's FQDN, the FQDN must not exceed 64 characters. If the FQDN is more than 64 characters, it must be authorized by using an IP address.

DHCP can be integrated with DNS to provide dynamic updates to pointer (PTR) and A records in a DNS zone. This ability enables a DHCP server to be a proxy for any DHCP client running an operating system that does not automatically update their DNS registration.

In Windows Server 2012, DHCP can be configured with DHCP failover. DHCP failover enables the DHCP server to be configured in hot standby mode, which provides redundancy, or load balance mode, which allocates client leases across two DHCP servers.

Tech Savvy Productions

The mode can be changed at any time, but a DHCP scope only supports using one mode at a time. IPv4 addresses that have been leased or reserved, including the options and settings for each scope, are shared by two DHCP servers. A single DHCP server supports up to 31 failover relationships. Failover relationships can be reused for additional scopes to avoid exceeding the limit.

When using DHCP hot standby mode, two servers operate the DHCP service, however one server provides and responds to all DHCP requests. The secondary server will only provide leases if the primary server is unreachable. To provide leases, a percentage of the IP address pool must be reserved for use by the secondary server.

By default, this is set to 5%. If the secondary server leases all of the IP addresses in the reserved space, it will not issue additional IP addresses from the primary server's scope. Existing leases will be renewed if requested by a DHCP client. Additionally, when the secondary server leases an IP address, the lease time is the maximum client lead time (MCLT) duration, not the full scope lease time. After the MCLT time has expired, the secondary server will
use the entire address pool in the scope, assuming that the primary server has resumed.

Using DHCP in load balancing mode is the default method of deployment. In this method, two servers provide the DHCP services simultaneously for a DHCP scope. The load balancing method is defined by a percentage of IP addresses on each server, and by default is split 50:50. This ratio or percentage can be configured to any amount between the two servers. The DHCP servers load balance based on a hash of the requesting client's MAC address. The MAC address thus determines which DHCP server will respond to a client's DHCP request. Similar to hot standby mode, if the partner server is unavailable, the remaining server will lease and renew IP addresses for the MCLT duration. After the MCLT time has expired, if the partner server is not online, the remaining server will lease addresses from the entire IP address pool for the scope.

Tech Savvy Productions

Security

Security is a huge topic because it encompasses so many areas. Even in a specific technology like AD DS, security is a huge topic. For this e-book, we will focus on 3 specific areas of security in AD DS:

- Securing LDAP traffic with SSL/TLS
- Modifying the access control list (ACL) on administrative accounts
- Enabling strong authentication in a domain

The first step to securing LDAP traffic with SSL/TLS is to install AD CS. It is a good practice is to install the role on a member server. Before you begin implementing a PKI, you should spend ample time gathering information, designing the PKI, and planning.

Once you have an operational PKI, you can issue a certificate to each DC. It is also possible to use a third-party Certification Authority (CA) to secure LDAP communication. The certificate must not have strong private key protection enabled.

Additionally, the DC FQDN must appear in either the Common Name of the Subject field, or as a DNS entry in the Subject Alternative Name extension. They Enhanced Key Usage must also include Server Authentication as an option, object identifier (OID) 1.3.6.1.5.5.7.3.1 By enabling LDAP over SSL, LDAP communication can occur on both ports 389 (LDAP) and 636 (LDAP/SSL). Global catalog requests performed with SSL used port 3269.

Tech Savvy Productions

The certreq.exe command can be used to create a new certificate request. To create a request, run the following command: `certreq -new request.inf request.req`

The request.inf file must contain the DC FQDN, and also provides information on the key length and request time. For a full example of the request.inf file, see http://support.microsoft.com/kb/321051

The request.req file will be created, and must be submitted to the CA. After receiving the certificate, process the request with the certreq.exe command.

```
certreq -accept certnew.cer
```

To verify that SSL has been enabled successful, the Active Directory Administration Tool (ldp.exe) can be used to manually connect to the AD DS environment.

Launch the ldp.exe tool, and perform a new connection. In the port field, specify 636. If the RootDSE information is displayed, LDAP over SSL has been configured successfully.

Chapter IX | Security 27

Another way to secure AD DS is to modify the ACL of user accounts that have administrative privileges. User accounts in administrative groups have their ACLs replaced every hour by a process on the PDC Emulator. This process compares the ACLs on each account in the administrative groups and the group itself to the AdminSDHolder object.

Tech Savvy Productions

In a domain named contoso.com, the object is located at CN=AdminSDHolder,CN=System, DC=Contoso,DC=Com. Any difference between the account and the AdminSDHolder object is then replaced on the account. The following groups and members of these groups are checked by the AdminSDHolder process:

- Administrators
- Account Operators
- Cert Publishers
- Backup Operators
- Domain Admins
- Enterprise Admins
- Print Operators
- Schema Admins
- Server Operators

Additionally, the Administrator and krbtgt user accounts are also checked independently regardless of group membership. Accounts that are protected by the AdminSDHolder process can be identified by the adminCount attribute of the account. If the account is being protected, this attribute will be set to 1.

To modify the ACL of an account being protected by the AdminSDHolder process, three steps must be taken to verify that the ACL is not replaced.

1. Remove the account object from all protected groups.
2. Set the adminCount attribute to 0.
3. Enable inheritance on the account object.

Tech Savvy Productions

Note that simply changing the adminCount to 0 without removing the object from the protected groups will not stop the process from replacing the ACLs. Removing the object from the groups without modifying the adminCount attribute doesn't fix the situation.

Another solution is to modify the AdminSDHolder object to include the ACLs that you want to apply to the user accounts. By including the additional access control entries in the AdminSDHolder object, they will also be included when the ACLs of each account object is compared. This is useful for modifying the permissions of all administrative user account objects.

Another area of AD DS security is to enable strong domain authentication in your environment. This will ensure that users can authenticate to Active Directory by using only strong authentication protocols. By default, Windows Server 2012 accepts authentication requests that use NTLM and NTLMv2, although it responds by replying with NTLMv2 only. This setting can be restricted by modifying the "Network Security:

LAM Manager Authentication Level" setting in group policy.

This setting is applied to DCs, and can be set to "Send NTLMv2 response only. Refuse LM and NTLM." By configuring this setting, DCs will not respond unless they receive an NTLMv2 request.

Chapter X | Auditing 28

Prior to Windows Server 2008 R2 and Windows 7, auditing in Windows was a fairly simple topic. You navigated to the auditing policies in a GPO and enabled auditing and chose Success, Failure, or both. There were a number of articles on the internet describing each of the auditing policies and many administrators quickly avoided anything that didn't provide much value.

Tech Savvy Productions

Below is a screen capture showing the available audit policy settings.
Figure 10.1 Audit policy settings In Windows Server 2008 R2, a new feature was introduced to allow for advanced auditing policies in Group Policy. Officially, 53 new settings were made available to replace the original 9 policy settings shown above.

A little known fact is that these 53 new settings were actually available in Windows Server 2008. However, you had to use logon scripts and auditpol.exe to take advantage of the new settings. Thus, most administrators didn't.
One common area of confusion is the apparent overlap of the original 9 policy settings (hereafter called the basic audit policy settings) and the advanced audit policy settings.

There actually isn't any overlap though. Let's examine why by looking at the account management auditing. With basic audit policy settings, you can enable the "Audit account management" policy for Success and Failure. With advanced audit policy, you can enable auditing for application group management, computer account management, distribution group management, account management events, security group management, and user account management.

Enabling the "Audit account management" basic audit policy setting is the same as enabling auditing in the 6 subcategories available in an advanced audit policy. Neither provide more data. But, as many administrators have realized, generating too much audit data can be worse
than not generating any audit data because of the massive volume of audit data that can be generated. Administrators have been struggling with audit data for a long time.

Some of the common struggles are:

- Chapter X | Auditing 29

- Windows event logs fill up. Windows event logs can be configured in a number of different ways. You can set a maximum log size and delete old

Tech Savvy Productions

event as needed. You can archive a log when it fills up and then start a new log.

- Or, you can configure the logs not to overwrite events and require manual intervention. You can even shut the server down if you can't write to the Security event log.
- Administrators often can't afford for new events not to be written or for servers to shut down when a log fills up. Thus, overwriting events or archiving are the most common settings.
- But this leads to administrative overhead: monitor event log sizes, monitor disk space, move archived logs off of server, manage archived logs, and figuring out a way to search through all of the data.
  Disk volumes run out of space. I still find it amusing that in 2015, disk space is still the major source of server downtime in many companies.
- Log files are a common problem whether from auditing or from applications such as IIS. I have heard from several organizations that experienced an outage and the root cause was the system volume running out of space due to Windows event log archiving.

- Inability to locate specific audit data. When you generate a massive amount of data, every data management task, even normally simple tasks, become complex and time consuming. Tasks such as compressing the files, copying the files to another location over the network, or searching through the files for a specific key term become problematic and incredibly time consuming.
- Administrators are turning toward third-party solutions to help.
  Inability to use audit data in a timely fashion. Imagine a call from the security team
  about an employee that has potentially viewed confidential HR data. They ask you to pull up audit data for the user over the last few weeks. Not a big deal if you have 1GB worth of audit data. But when you have 500GB worth of audit data, it suddenly becomes your full time job for a few weeks.

- The advanced audit policy settings can help. By offering more granular auditing options, you can greatly reduce the amount of data gathered.

Tech Savvy Productions

This minimizes the struggles mentioned above. But, there is a big investment in time to switch over to the advanced audit policy settings. For some organizations, that investment will pay for itself and more.

- Let's take a quick look at how this impacts the number of events captured. In this first example, in a Windows Server 2003 R2 domain named adatum.com, I set basic audit settings so that only account management success events are being audited, as shown below.
- There isn't any significance to the operating system version because the basic audit settings shown below are available in every version of Windows Server since Windows 2000 Server.

-
- Figure 10.2
  Basic audit settings
  Chapter X | Auditing 30
  Then, I created a new computer object and refreshed the Security event log. Below are the entries related to the new computer object creation.
  Figure 10.3 Entries related to the new computer object creation
  There are 5 events.
- Next, in a Windows Server 2012 R2 domain named contoso.com, I created an advanced audit policy based on wanting to audit only successful user account management events, as shown below.
  Figure 10.4
- An advanced audit policy
  Then, I created a new computer object and refreshed the Security event log. Below are the entries related to the new computer object creation.
  Figure 10.5

- Entries related to the new computer object creation With the advanced auditing policy, only 2 events were logged. That's a big difference, especially when you think about how that would extrapolate once all auditing categories were configured. Note that not only would we save some time with the granular approach, but we will also save quite a bit more time later after the other auditing categories are configured. Now,

Tech Savvy Productions

you may be wondering why creating a computer account is showing up in user account auditing? That's because computer objects are part of the AD DS user class along with user objects! You may also be wondering why the event log IDs are different for each environment. This is unrelated to the auditing configuration and instead is because of a change in the event log IDs used in Windows Server 2008 and newer. Many of the event IDs changed between versions of Windows Server.

Chapter XI | User Unfriendly Log Data 31

XI. User Unfriendly Log Data

As a side effect of auditing and logging, administrators have to deal with a ton of data and information. Often, not only is there an overload of information, but often the information isn't readily usable. Here are some of the pain points that administrators are dealing with when it comes to audit and log data:

- The information is unformatted. In this situation, information is captured but it is unformatted. In other words, everything looks the same and it can be a struggle to extract the key pieces of information that you need. In these situations, administrators struggle to work with the data – filtering, sorting, manipulating, and extracting are difficult without hours of administrative effort.

- The informatio is verbose. In this situation, there is too much information. For example, take an example of a user signing in to their computer in the morning.
  Instead of a single event log entry, you end up with several entries, each covering a part of the overall sign in process. Another example is a packet capture. You may be troubleshooting an authentication issue on a DC, but your packet capture contains tons of other irrelevant data. In some cases, administrators can turn to tools to create filters to minimize data capture. But that isn't nearly as effective for capturing Windows security related data on an ongoing basis.

Tech Savvy Productions

- The information is uncorrelated. For example, imagine a scenario where you are troubleshooting authentication from a federated partner. You have information in the Windows event logs on the DC, Windows event logs on the AD FS servers, Windows event logs on the target web server, Windows event logs on the client computer, AD FS proxy logs, AD FS specific logs, and IIS logs. What's worse is that there isn't any correlation to the logs beyond a date and time stamp. The built-in tools in Windows don't provide a good event correlation solution for this problem so administrators are often using third-party tools to help.

- The information is ignored. After extended periods of time dealing with unformatted, verbose, and uncorrelated information, administrators often start ignoring logs, ignoring alerts, and ignoring important information that may be buried "in there somewhere". This same phenomenon is often seen in monitoring projects when so many alerts are being generated that they go unanswered.

- Part of the solution to this problem is to reduce the amount of information you capture. But that sometimes goes against what a security team wants a company to capture – everything. So there is a balancing act. You need to capture enough information to make it useful but without making it a burden. Administrators often turn to third- party tools to help come up with the right solution.
  About Netwrix

  For more information, visit www.netwrix.com.

1 Introduction

This is the primary specification for **Active Directory**, both **Active Directory Domain Services (AD DS)** and **Active Directory Lightweight Directory Services (AD LDS)**. When the specification does not refer specifically to AD DS or AD LDS, it applies to both. The state model for this specification is prerequisite to the other specifications for Active Directory: [MS-DRSR] and [MS-SRPL].

When no operating system version information is specified, information in this document applies to all relevant versions of Windows. Similarly, when no **DC functional level** is specified, information in this document applies to all DC functional levels.

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs.

Note: The terms "earlier" and "later", when used with a product version, refer to either all preceding versions or all subsequent versions, respectively. The term "through" refers to the inclusive range of versions. Applicable Microsoft products are listed chronologically in this section.

- Windows 2000 Server operating system
- Windows Server 2003 operating system
- Windows Server 2003 R2 operating system
- Windows Server 2008 operating system
- Windows Server 2008 R2 operating system
- Windows Server 2012 operating system
- Windows Server 2012 R2 operating system
- Windows Server 2016 operating system
- Windows Server v1709 operating system
- Windows Server v1803 operating system

Tech Savvy Productions

- Windows Server v1809 operating system
- Windows Server 2019 operating system
- Windows Server v1903 operating system
- Windows Server 2022 operating system

AD DS:

First became available as part of Microsoft Windows 2000 operating system and is available as part of Windows 2000 Server, Windows Server 2003, and Windows Server 2003 R2 products; in these products it is called "Active Directory".

It is also available as part of Windows Server 2008 and later. AD DS is not present in Windows NT 3.1 operating system, Windows NT 3.51 operating system, Windows NT 4.0 operating system, or Windows XP operating system.

AD LDS: first became available during the release of Windows Server 2003.

In Windows XP, Windows Server 2003, and Windows Server 2003 R2, it is a standalone application called "Active Directory Application Mode (ADAM)". AD LDS is also available as part of Windows Server 2008 and later. Unless otherwise specified, information in this specification is also applicable to AD LDS. There are two versions of ADAM, ADAM RTW (introduced in the same timeframe as Windows Server 2003 operating system with Service Pack 1 (SP1)) and ADAM SP1 (introduced in the same timeframe as Windows Server 2003 operating system with Service Pack 2 (SP2)); unless otherwise specified, where ADAM is discussed in this document it refers to both versions of ADAM.

Tech Savvy Productions

AD LDS for a particular Windows client is a standalone application that provides AD LDS capabilities for that Windows client. Information that is applicable to AD LDS on applicable Windows Server releases is generally also applicable to AD LDS on Windows clients, including Windows 11 operating system and later, except where it is explicitly specified that such information is not applicable to that product.

Information that is applicable to AD LDS on Windows Server v1903 is also applicable to AD LDS for Windows 10 v1903 operating system.

- Information that is applicable to AD LDS on Windows Server 2022 is also applicable to AD LDS for Windows 10 v21H1 operating system.

State is included in the state model for this specification only as necessitated by the requirement that a licensee implementation of the protocols of applicable Windows Server releases has to be capable of receiving messages and responding in the same manner as applicable Windows Server releases. Behavior is specified in terms of request message received, processing based on current state, resulting state transformation, and response message sent. Unless otherwise specified in the sections that follow, all of the behaviors are required for interoperability.

The following typographical convention is used to indicate the special meaning of certain names:

- Underline, as in instanceType: the name of an **attribute** or **object class** whose interpretation is specified in the following documents:
  - [MS-ADA1] Attribute names for AD DS whose initial letter is A through L.
  - [MS-ADA2] Attribute names for AD DS whose initial letter is M.
  - [MS-ADA3] Attribute names for AD DS whose initial letter is N through Z.
  - [MS-ADSC] **Object class names** for AD DS.
  - [MS-ADLS] Object class names and attribute names for AD LDS.

For clarity, bit flags are sometimes shown as bit field diagrams. In the case of bit flags for **Lightweight Directory Access Protocol (LDAP)** attributes, these diagrams take on **big-endian** characteristics but do not reflect the actual byte ordering of

Tech Savvy Productions

integers over the wire, because LDAP transfers an integer as the **UTF-8** string of the decimal representation of that integer, as specified in [RFC2252].

2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the Errata.

3 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information.

[C706] The Open Group, "DCE 1.1: Remote Procedure Call", C706, August 1997, https://publications.opengroup.org/c706

Note Registration is required to download the document.

[GRAY] Gray, J., and Reuter, A., "Transaction Processing: Concepts and Techniques", The Morgan Kaufmann Series in Data Management Systems, San Francisco: Morgan Kaufmann Publishers, 1992, Hardcover ISBN: 9781558601901.

[IEEE1003.1] The Open Group, "IEEE Std 1003.1, 2004 Edition", 2004, http://www.unix.org/version3/ieee_std.html

[ISO-8601] International Organization for Standardization, "Data Elements and Interchange Formats - Information Interchange - Representation of Dates and Times", ISO/IEC 8601:2004, December 2004, http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=40874&ICS1=1&ICS2=140&ICS3=30

Note There is a charge to download the specification.

[ISO/IEC-14977] International Organization for Standardization, "Information technology -- Syntactic metalanguage -- Extended BNF", ISO/IEC 14977:1996,

Tech Savvy Productions

http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=26153

[ISO/IEC-9899] International Organization for Standardization, "Programming Languages - C", ISO/IEC 9899:TC2, May 2005, http://www.open-std.org/jtc1/sc22/wg14/www/docs/n1124.pdf

[ITUX680] ITU-T, "Abstract Syntax Notation One (ASN.1): Specification of Basic Notation", Recommendation X.680, July 2002, http://www.itu.int/ITU-T/studygroups/com17/languages/X.680-0207.pdf

[ITUX690] ITU-T, "ASN.1 Encoding Rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)", Recommendation X.690, July 2002, http://www.itu.int/ITU-T/studygroups/com17/languages/X.690-0207.pdf

[KNUTH1] Knuth, D., "The Art of Computer Programming: Volume 1/Fundamental Algorithms (Second Edition)", Reading, MA: Addison-Wesley, 1973, ASIN: B000NV8YOA.

[MS-ADA1] Microsoft Corporation, "Active Directory Schema Attributes A-L".

[MS-ADA2] Microsoft Corporation, "Active Directory Schema Attributes M".

[MS-ADA3] Microsoft Corporation, "Active Directory Schema Attributes N-Z".

[MS-ADLS] Microsoft Corporation, "Active Directory Lightweight Directory Services Schema".

[MS-ADSC] Microsoft Corporation, "Active Directory Schema Classes".

[MS-APDS] Microsoft Corporation, "Authentication Protocol Domain Support".

[MS-CTA] Microsoft Corporation, "Claims Transformation Algorithm".

[MS-DRSR] Microsoft Corporation, "Directory Replication Service (DRS) Remote Protocol".

[MS-DTYP] Microsoft Corporation, "Windows Data Types".

[MS-ERREF] Microsoft Corporation, "Windows Error Codes".

Tech Savvy Productions

[MS-FRS1] Microsoft Corporation, "File Replication Service Protocol".

[MS-GKDI] Microsoft Corporation, "Group Key Distribution Protocol".

[MS-GPSB] Microsoft Corporation, "Group Policy: Security Protocol Extension".

[MS-KILE] Microsoft Corporation, "Kerberos Protocol Extensions".

[MS-LSAD] Microsoft Corporation, "Local Security Authority (Domain Policy) Remote Protocol".

[MS-MAIL] Microsoft Corporation, "Remote Mailslot Protocol".

[MS-NLMP] Microsoft Corporation, "NT LAN Manager (NTLM) Authentication Protocol".

[MS-NRPC] Microsoft Corporation, "Netlogon Remote Protocol".

[MS-PAC] Microsoft Corporation, "Privilege Attribute Certificate Data Structure".

[MS-RPCE] Microsoft Corporation, "Remote Procedure Call Protocol Extensions".

[MS-SAMR] Microsoft Corporation, "Security Account Manager (SAM) Remote Protocol (Client-to-Server)".

[MS-SFU] Microsoft Corporation, "Kerberos Protocol Extensions: Service for User and Constrained Delegation Protocol".

[MS-SPNG] Microsoft Corporation, "Simple and Protected GSS-API Negotiation Mechanism (SPNEGO) Extension".

[MS-SRPL] Microsoft Corporation, "Directory Replication Service (DRS) Protocol Extensions for SMTP".

[MS-UCODEREF] Microsoft Corporation, "Windows Protocols Unicode Reference".

[MS-W32T] Microsoft Corporation, "W32Time Remote Protocol".

[MSKB-3106637] Microsoft Corporation, "Incorrect results in LDAP query, domain controller restarts, or user logons are denied in Windows Server 2012 R2", https://support.microsoft.com/en-us/kb/3106637

Tech Savvy Productions

[MSKB-3155495] Microsoft Corporation, "You can't use the Active Directory shadow principal groups feature for groups that are always filtered out in Windows", revision 2.0, May 2016, https://support.microsoft.com/en-us/kb/3155495

[MSKB-3192404] Microsoft Corporation, "October 2016 Preview of Monthly Quality Rollup for Windows 8.1 and Windows Server 2012 R2", https://support.microsoft.com/en-us/kb/3192404

[MSKB-4019217] Microsoft Corporation, "May 16, 2017 - KB4019217 (Preview of Monthly Rollup)", https://support.microsoft.com/en-us/kb/4019217

[MSKB-4025334] Microsoft Corporation, "July 18, 2017—KB4025334 (OS Build 14393.1532)", https://support.microsoft.com/en-us/kb/4025334

[MSKB-4038801] Microsoft Corporation, "September 28, 2017—KB4038801 (OS Build 14393.1737)", https://support.microsoft.com/help/4038801

[MSKB-4490425] Microsoft Corporation, "Updates to TGT delegation across incoming trusts in Windows Server", https://support.microsoft.com/en-us/help/4490425/updates-to-tgt-delegation-across-incoming-trusts-in-windows-server

[MSKB-4505903] Microsoft Corporation, "July 26, 2019--KB4505903 (OS Build 18362.267)", July 2019, https://support.microsoft.com/en-us/help/4505903/windows-10-update-kb4505903

[RFC1001] Network Working Group, "Protocol Standard for a NetBIOS Service on a TCP/UDP Transport: Concepts and Methods", RFC 1001, March 1987, http://www.ietf.org/rfc/rfc1001.txt

[RFC1002] Network Working Group, "Protocol Standard for a NetBIOS Service on a TCP/UDP Transport: Detailed Specifications", STD 19, RFC 1002, March 1987, http://www.rfc-editor.org/rfc/rfc1002.txt

[RFC1034] Mockapetris, P., "Domain Names - Concepts and Facilities", STD 13, RFC 1034, November 1987, http://www.ietf.org/rfc/rfc1034.txt

Tech Savvy Productions

[RFC1035] Mockapetris, P., "Domain Names - Implementation and Specification", STD 13, RFC 1035, November 1987, http://www.ietf.org/rfc/rfc1035.txt

[RFC1088] McLaughlin III, L., "A Standard for the Transmission of IP Datagrams over NetBIOS Networks", RFC 1088, February 1989, http://www.ietf.org/rfc/rfc1088.txt

[RFC1166] Kirkpatrick, S., Stahl, M., Recker, M., "Internet Numbers", RFC 1166, July 1990, http://www.ietf.org/rfc/rfc1166.txt

[RFC1274] Barker, P. and Kille, S., "The COSINE and Internet X.500 Schema", RFC 1274, November 1991, http://www.ietf.org/rfc/rfc1274.txt

[RFC1278] Hardcastle-Kille, S. E., "A string encoding of Presentation Address", RFC 1278, November 1991, http://www.ietf.org/rfc/rfc1278.txt

[RFC1777] Yeong, W., Howes, T., and Kille, S., "Lightweight Directory Access Protocol", RFC 1777, March 1995, http://www.ietf.org/rfc/rfc1777.txt

[RFC1798] Young, A., "Connection-less Lightweight X.500 Directory Access Protocol", RFC 1798, June 1995, http://www.ietf.org/rfc/rfc1798.txt

[RFC1964] Linn, J., "The Kerberos Version 5 GSS-API Mechanism", RFC 1964, June 1996, http://www.rfc-editor.org/rfc/rfc1964.txt

[RFC2078] Linn, J., "Generic Security Service Application Program Interface, Version 2", RFC 2078, January 1997, http://www.ietf.org/rfc/rfc2078.txt

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, https://www.rfc-editor.org/rfc/rfc2119.html

[RFC2136] Thomson, S., Rekhter Y. and Bound, J., "Dynamic Updates in the Domain Name System (DNS UPDATE)", RFC 2136, April 1997, http://www.ietf.org/rfc/rfc2136.txt

[RFC2222] Myers, J., "Simple Authentication and Security Layer (SASL)", RFC 2222, October 1997, http://www.ietf.org/rfc/rfc2222.txt

[RFC2246] Dierks, T., and Allen, C., "The TLS Protocol Version 1.0", RFC 2246, January 1999, https://www.rfc-editor.org/info/rfc2246

Tech Savvy Productions

[RFC2247] Kille, S., Wahl, M., Grimstad, A., et al., "Using Domains in LDAP/X.500 Distinguished Names", RFC 2247, January 1998, http://www.ietf.org/rfc/rfc2247.txt

[RFC2251] Wahl, M., Howes, T., and Kille, S., "Lightweight Directory Access Protocol (v3)", RFC 2251, December 1997, http://www.ietf.org/rfc/rfc2251.txt

[RFC2252] Wahl, M., Coulbeck, A., Howes, T., and Kille, S., "Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions", RFC 2252, December 1997, http://www.ietf.org/rfc/rfc2252.txt

[RFC2253] Wahl, M., Kille, S., and Howe, T., "Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names", RFC 2253, December 1997, http://www.ietf.org/rfc/rfc2253.txt

[RFC2255] Howes, T. and Smith, M., "The LDAP URL Format", RFC 2255, December 1997, http://www.ietf.org/rfc/rfc2255.txt

[RFC2256] Wahl, M., "A Summary of the X.500(96) User Schema for use with LDAPv3", RFC 2256, December 1997, http://www.ietf.org/rfc/rfc2256.txt

[RFC2307] Howard, L., "An Approach for Using LDAP as a Network Information Service", RFC 2307, March 1998, http://www.ietf.org/rfc/rfc2307.txt

[RFC2589] Yaacovi, Y., Wahl, M., and Genovese, T., "Lightweight Directory Access Protocol (v3): Extensions for Dynamic Directory Services", RFC 2589, May 1999, http://www.ietf.org/rfc/rfc2589.txt

[RFC2696] Weider, C., Herron, A., Anantha, A., and Howes, T., "LDAP Control Extension for Simple Paged Results Manipulation", RFC 2696, September 1999, http://www.ietf.org/rfc/rfc2696.txt

[RFC2782] Gulbrandsen, A., Vixie, P., and Esibov, L., "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000, http://www.ietf.org/rfc/rfc2782.txt

[RFC2798] Smith, M., "Definition of the inetOrgPerson LDAP Object Class", RFC 2798, April 2000, http://www.ietf.org/rfc/rfc2798.txt

Tech Savvy Productions

[RFC2829] Wahl, M., Alvestrand, H., Hodges, J., and Morgan, R., "Authentication Methods for LDAP", RFC 2829, May 2000, http://www.ietf.org/rfc/rfc2829.txt

[RFC2830] Hodges, J., Morgan, R., and Wahl, M., "Lightweight Directory Access Protocol (v3): Extension for Transport Layer Security", RFC 2830, May 2000, http://www.ietf.org/rfc/rfc2830.txt

[RFC2831] Leach, P. and Newman, C., "Using Digest Authentication as a SASL Mechanism", RFC 2831, May 2000, http://www.ietf.org/rfc/rfc2831.txt

[RFC2849] Good, G., "The LDAP Data Interchange Format (LDIF) - Technical Specification", RFC 2849, June 2000, http://www.ietf.org/rfc/rfc2849.txt

[RFC2891] Howes, T., Wahl, M., and Anantha, A., "LDAP Control Extension for Server Side Sorting of Search Results", RFC 2891, August 2000, http://www.ietf.org/rfc/rfc2891.txt

[RFC3377] Hodges, J. and Morgan, R., "Lightweight Directory Access Protocol (v3): Technical Specification", RFC 3377, September 2002, http://www.ietf.org/rfc/rfc3377.txt

[RFC3961] Raeburn, K., "Encryption and Checksum Specifications for Kerberos 5", RFC 3961, February 2005, http://www.ietf.org/rfc/rfc3961.txt

[RFC4120] Neuman, C., Yu, T., Hartman, S., and Raeburn, K., "The Kerberos Network Authentication Service (V5)", RFC 4120, July 2005, https://www.rfc-editor.org/rfc/rfc4120.txt

[RFC4121] Zhu, L., Jaganathan, K., and Hartman, S., "The Kerberos Version 5 Generic Security Service Application Program Interface (GSS-API) Mechanism: Version 2", RFC 4121, July 2005, http://www.ietf.org/rfc/rfc4121.txt

[RFC4122] Leach, P., Mealling, M., and Salz, R., "A Universally Unique Identifier (UUID) URN Namespace", RFC 4122, July 2005, http://www.rfc-editor.org/rfc/rfc4122.txt

[RFC4178] Zhu, L., Leach, P., Jaganathan, K., and Ingersoll, W., "The Simple and Protected Generic Security Service Application Program Interface (GSS-API)

Tech Savvy Productions

Negotiation Mechanism", RFC 4178, October 2005, https://www.rfc-editor.org/rfc/rfc4178.txt

[RFC4291] Hinden, R. and Deering, S., "IP Version 6 Addressing Architecture", RFC 4291, February 2006, http://www.ietf.org/rfc/rfc4291.txt

[RFC4370] Weltman, R., "Lightweight Directory Access Protocol (LDAP) Proxied Authorization Control", RFC 4370, February 2006, http://www.ietf.org/rfc/rfc4370.txt

[RFC4532] Zeilenga, K., "Lightweight Directory Access Protocol (LDAP)", Who Am I?" Operation", RFC 4532, June 2006, http://www.ietf.org/rfc/rfc4532.txt

[RFC4757] Jaganathan, K., Zhu, L., and Brezak, J., "The RC4-HMAC Kerberos Encryption Types Used by Microsoft Windows", RFC 4757, December 2006, http://www.ietf.org/rfc/rfc4757.txt

[RFC5056] Williams, N., "On the Use of Channel Bindings to Secure Channels", RFC 5056, November 2007, https://www.rfc-editor.org/rfc/rfc5056.txt

[RFC5929] Altman, J., Williams, N., and Zhu, L., "Channel Bindings for TLS", RFC 5929, July 2010, https://www.rfc-editor.org/rfc/rfc5929.txt

[RFC5952] Kawamura, S., Kawashima, M., "A Recommendation for IPv6 Address Text Representation", RFC 5952, August 2010, https://tools.ietf.org/html/rfc5952

[RFC7049] Bormann, C., and Hoffman, P., "Concise Binary Object Representation (CBOR)", RFC 7049, October 2013, https://www.rfc-editor.org/info/rfc7049

[RFC791] Postel, J., Ed., "Internet Protocol: DARPA Internet Program Protocol Specification", RFC 791, September 1981, http://www.rfc-editor.org/rfc/rfc791.txt

[RFC8017] Moriarty, K., Ed., Kaliski, B., Jonsson, J., and Rusch, A., "PKCS #1: RSA Cryptography Specifications Version 2.2", November 2016, https://www.rfc-editor.org/rfc/rfc8017.txt

[RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 8259, December 2017, https://www.rfc-editor.org/rfc/rfc8259.txt

Tech Savvy Productions

[W3C-WebAuthPKC1] Balfanz, D., Czeskis, A., Hodges, J., et al., Eds., "Web Authentication: An API for accessing Public Key Credentials Level 1", W3C Recommendation, March 2019, https://www.w3.org/TR/webauthn-1/

[X501] ITU-T, "Information Technology - Open Systems Interconnection - The Directory: The Models", Recommendation X.501, August 2005, http://www.itu.int/rec/T-REC-X.501-200508-S/en

[XMLSCHEMA1] Thompson, H., Beech, D., Maloney, M., and Mendelsohn, N., Eds., "XML Schema Part 1: Structures", W3C Recommendation, May 2001, https://www.w3.org/TR/2001/REC-xmlschema-1-20010502/

[XMLSCHEMA2/2] Biron, P., and Malhotra, A., Eds., "XML Schema Part 2: Datatypes Second Edition", W3C Recommendation, October 2004, https://www.w3.org/TR/2004/REC-xmlschema-2-20041028/

[XPATH] Clark, J. and DeRose, S., "XML Path Language (XPath), Version 1.0", W3C Recommendation, November 1999, http://www.w3.org/TR/1999/REC-xpath-19991116/

4 Informative References

[ADDLG] Microsoft Corporation, "Security Briefs: Credentials and Delegation", September 2005, http://msdn.microsoft.com/en-us/magazine/cc163740.aspx

[LISP15] McCarthy, J., Abrahams, P., Edwards, D., Hart, T., and Levin, M., "LISP 1.5 Programmers Manual", Cambridge, MA: The M.I.T. Press, 1965, ISBN-10: 0262130114.

[MS-ADDM] Microsoft Corporation, "Active Directory Web Services: Data Model and Common Elements".

[MS-AUTHSOD] Microsoft Corporation, "Authentication Services Protocols Overview".

[MS-DSSP] Microsoft Corporation, "Directory Services Setup Remote Protocol".

[MS-GPOD] Microsoft Corporation, "Group Policy Protocols Overview".

Tech Savvy Productions

[MS-SYS-ARCHIVE] Microsoft Corporation, "Windows System Overview", https://docs.microsoft.com/en-us/openspecs/windows_protocols/MS-WINPROTLP/df36f95e-6a6b-48d6-a3ae-35a17674f546

[MS-XCA] Microsoft Corporation, "Xpress Compression Algorithm".

[MSDN-CAR] Microsoft Corporation, "Control Access Rights", https://msdn.microsoft.com/en-us/library/ms680945(v=vs.85).aspx

[MSDN-gethostbyname] Microsoft Corporation, "gethostbyname function", http://msdn.microsoft.com/en-us/library/windows/desktop/ms738524(v=vs.85).aspx

[MSDOCS-SchUpd] Microsoft Corporation, "Schema Updates", https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/deploy/schema-updates

[MSFT-CVE-2021-42278] Microsoft Corporation, "Active Directory Domain Services Elevation of Privilege Vulnerability", CVE-2021-42278, November 9, 2021, https://msrc.microsoft.com/update-guide/vulnerability/CVE-2021-42278

[MSFT-CVE-2021-42282] Microsoft Corporation, "Active Directory Domain Services Elevation of Privilege Vulnerability", CVE-2021-42282, November 9, 2021, https://msrc.microsoft.com/update-guide/vulnerability/CVE-2021-42282

[MSFT-CVE-2021-42291] Microsoft Corporation, "Active Directory Domain Services Elevation of Privilege Vulnerability", CVE-2021-42291, November 9, 2021, https://msrc.microsoft.com/update-guide/vulnerability/CVE-2021-42291

[MSFT-CVE-2022-21857] Microsoft Corporation, "Active Directory Domain Services Elevation of Privilege Vulnerability", CVE-2022-21857, January 11, 2022, https://msrc.microsoft.com/update-guide/vulnerability/CVE-2022-21857

[MSKB-3070083] Microsoft Corporation, "Domain join against a Windows Server 2012 R2-based domain controller fails if SPN is not unique in the forest", https://support.microsoft.com/en-us/kb/3070083

[MSKB-5011543] Microsoft Corporation, "March 22, 2022—KB5011543 (OS Builds 19042.1620, 19043.1620, and 19044.1620) Preview", March 2022,

Tech Savvy Productions

https://support.microsoft.com/en-us/topic/march-22-2022-kb5011543-os-builds-19042-1620-19043-1620-and-19044-1620-preview-4fe2d1c0-720f-47fe-9523-75339bc107a1

[MSKB-5011551] Microsoft Corporation, "March 22, 2022—KB5011551 (OS Build 17763.2746) Preview", March 2022, https://support.microsoft.com/en-us/topic/march-22-2022-kb5011551-os-build-17763-2746-preview-690a59cd-059e-40f4-87e8-e9139cc65de4

[MSKB-5011558] Microsoft Corporation, "March 22, 2022—KB5011558 (OS Build 20348.617) Preview", March 2022, https://support.microsoft.com/en-us/topic/march-22-2022-kb5011558-os-build-20348-617-preview-8bb6ded6-7515-44eb-9fa0-e214eb6d7a75

[MSKB-5011563] Microsoft Corporation, "March 28, 2022—KB5011563 (OS Build 22000.593) Preview", March 2022, https://support.microsoft.com/en-us/topic/march-28-2022-kb5011563-os-build-22000-593-preview-40df54c9-b5a9-42e5-ae1c-9a33ff91ca91

[RFC1122] Braden, R., Ed., "Requirements for Internet Hosts -- Communication Layers", STD 3, RFC 1122, October 1989, http://www.rfc-editor.org/rfc/rfc1122.txt

[RFC768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980, http://www.rfc-editor.org/rfc/rfc768.txt

[RFC792] Postel, J., "Internet Control Message Protocol", RFC 792, September 1981, http://www.ietf.org/rfc/rfc792.txt

[RFC793] Postel, J., Ed., "Transmission Control Protocol: DARPA Internet Program Protocol Specification", RFC 793, September 1981, http://www.rfc-editor.org/rfc/rfc793.txt

[SPNNAMES] Microsoft Corporation, "Name Formats for Unique SPNs", http://msdn.microsoft.com/en-us/library/ms677601.aspx

[VLVDRAFT] Boreham, D., Sermersheim, J., and Kashi, A., "LDAP Extensions for Scrolling View Browsing of Search Results", draft-ietf-ldapext-ldapv3-vlv-09, November 2002, http://tools.ietf.org/html/draft-ietf-ldapext-ldapv3-vlv-09

Tech Savvy Productions

5 Overview

This is the primary specification for **Active Directory**. The state model for this specification is prerequisite to the other specifications for Active Directory: [MS-DRSR] and [MS-SRPL].

Active Directory is either deployed as **AD DS** or as **AD LDS**. This document describes both forms. When the specification does not refer specifically to AD DS or AD LDS, it applies to both.

The remainder of this section describes the structure of this document.

The basic state model is specified in section 3.1.1.1. The basic state model is prerequisite to the remainder of the document. Section 3.1.1.1 also includes descriptive content to introduce key concepts and refer to places in the document where the full specification is given.

The **schema** completes the state model and is specified in section 3.1.1.2. The schema is prerequisite to the remainder of the document.

Active Directory is a server for **LDAP**. Section 3.1.1.3 specifies the extensions and variations of LDAP that are supported by Active Directory.

LDAP is an access protocol that determines very little about the behavior of the data being accessed. Section 3.1.1.4 specifies read (LDAP Search) behaviors, and section 3.1.1.5 specifies **update** (LDAP Add, Modify, Modify DN, Delete) behaviors.

Tech Savvy Productions

Section 3.1.1.6 specifies background tasks required due to write operations, to the extent that those tasks are exposed by protocols.

One of the update behaviors is the maintenance of the change log for use by Windows NT 4.0 **backup domain controller (BDC) replication** [MS-NRPC] section 3.6. The maintenance of this change log is specified in section 3.1.1.7.

The security services that Active Directory offers clients of LDAP are specified in section 5.1.

Active Directory contains a number of **objects**, visible through LDAP, that have special significance to the system. Section 6.1 specifies these objects.

A server running Active Directory is part of a distributed system that performs replication. The **Knowledge Consistency Checker (KCC)** is a component that is used to create spanning trees for **DC**-to-DC replication, and is specified in section 6.2.

A server running Active Directory is responsible for publishing the services that it offers, in order to eliminate the administrative burden of configuring clients to use particular servers running Active Directory. A server running Active Directory also implements the server side of the **LDAP ping** and **mailslot ping** protocols to aid clients in selecting among all the servers offering the same service. Section 6.3 specifies how a server running Active Directory publishes its services, and how a client needing some service can use this publication plus the LDAP ping or mailslot ping to locate a suitable server.

Computers in a network with Active Directory can be put into a state called "domain joined"; when in this state, the computer can authenticate itself. Section 6.4 specifies both the state in Active Directory and the state on a computer required for the **domain joined** state.

Each type of data stored in Active Directory has an associated function that compares two values to determine if they are equal and, if not, which is greater. Section 3.1.1.2 specifies all but one of these functions; the methodology for comparing two **Unicode** strings is specified in section 6.5.

Tech Savvy Productions

6 Relationship to Other Protocols

This is the primary specification for **Active Directory**. The state model for this specification is prerequisite to the specification for Active Directory described in [MS-DRSR]. This Active Directory Technical Specification depends on the following protocols:

- Lightweight Directory Access Protocol (LDAP)
- Remote Procedure Call (RPC)
- Domain Name System (DNS)

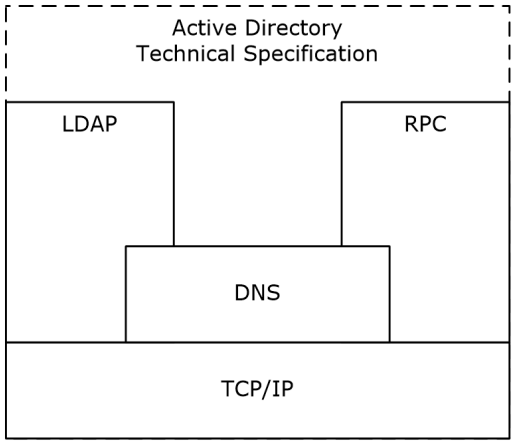


Figure 1: Protocol and technical specification relationships

Other protocols make use of implementations of the Active Directory Technical Specification as a data store.

7 Prerequisites/Preconditions

Active Directory requires an IP network and a **DNS** infrastructure.

8 Applicability Statement

Active Directory is not suitable for storing very large **attribute** values because, for instance, there is no provision for check-pointing a large data transfer to allow restart after a failure. The bandwidth and latency of typical networks makes Active Directory unsuitable for storing volatile data in **replicated attributes**. Active Directory is especially suitable for storing security account data, including passwords, and email address book data.

Tech Savvy Productions

Active Directory Protocols Overview

Provides an overview of the functionality and relationship of the protocols that make up the client-server and server-to-server behavior of Active Directory.

The Active Directory protocols provide directory services for the:

- centralized storage of identity
- account information
- storage for other forms of data such as group policies
- printer location information
- foundation for authentication services in a domain environment
- domain services
- directory replication services in Windows

The Active Directory protocols are specified in [LDAP], [MS-ADTS], [MS-SRPL], [MS-DRSR], [MS-SNTP], [MS-LSAD], [MS-LSAT], [MS-DSSP], [MS-SAMR], [MS-SAMS], [MS-WSDS], [WFXR], [WSENUM], [MS-WSTIM], [MS-ADDM], [MS-WSPELD], and [MS-ADCAP].

9 Introduction

Active Directory® is a **directory service (DS)**. Directory services can be used to provide a central store for **identity** and **account** information as well as storage of information for other systems and applications. Use of the **Active Directory** system is appropriate when there is a requirement for a DS. It is also appropriate when building another system that has a dependency on the Active Directory protocols. An example of such a system is the Group Policy system, which is described in the Group Policy Protocols Overview document [MS-GPOD].

This document describes the member protocols that comprise the Active Directory system. It also describes the abstract state that is shared between the system's protocols. This document is intended for anyone who plans to implement the Active Directory system because it provides a high-level introduction to the

Tech Savvy Productions

functionality of the system and also describes the protocols that an implementation of the Active Directory system supports.

This document does not duplicate or replace the content of the protocol specifications that describe the individual protocols in the Active Directory system. An implementer has to refer to those Technical Documents (TDs) for information about each protocol. Additionally, the Active Directory Technical Specification [MS-ADTS] contains vital information about the behavior of the DS, such as the state model and processing rules, that is essential to the correct functioning of the system.

A DS is a service that stores and organizes **directory objects** in a centralized, hierarchical data store. This hierarchical organization of objects is called the **directory**. A directory object is an object that contains one or more **attributes**. Each attribute can have one or more values. Directory objects are identified by a name that is unique among all directory objects in the DS. The directory objects are organized in a hierarchical manner with regards to other directory objects. For example, a DS might have a container directory object named Users, the contents of which (referred to as child directory objects) are containers named for each logical division of users; for example, Accounting Department, Human Resources Department, Engineering Department, and so on. The contents of each of these containers, in turn, could be user objects, each of which represents one individual user and contains attributes that store information about that user, such as the user name, password, or telephone number. The following diagram shows this example.

Tech Savvy Productions

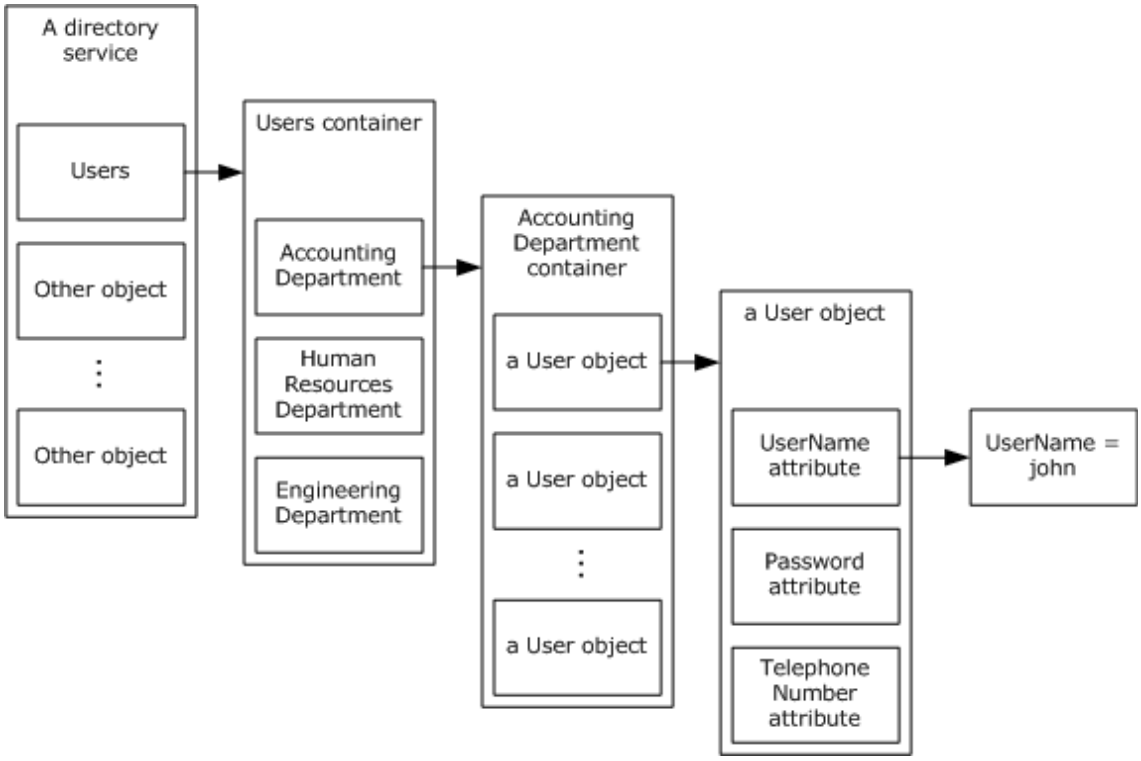


Figure 2: Example of directory organization

The Active Directory system can operate in two distinct modes:

- as **Active Directory Lightweight Directory Services (AD LDS)**
- and as **Active Directory Domain Services (AD DS)**.

AD LDS consists of a directory service that is accessible via the **Lightweight Directory Access Protocol (LDAP)** versions 2 and 3. AD LDS is primarily intended for use by application software as a storage mechanism. Note that information that is applicable to AD LDS on applicable Windows Server releases is also generally applicable to AD LDS on Windows clients. For more information, see [MS-ADTS] section 1.

AD DS is also accessible via LDAP versions 2 and 3, but it extends the basic DS to include additional capabilities, such as the ability to host **domain naming contexts (domain NCs)**, and additional protocols. This permits AD DS to store the account information for the users of a computer network. The collection of accounts that is stored in AD DS is referred to as a **domain**. Such account storage is a vital function of the Active Directory system, and in particular of AD DS. However, the Active Directory system is not limited to storing such information. Any information

Tech Savvy Productions

that can be represented as a collection of attribute/value pairs, including the possibility of multivalued attributes, can be modeled as a directory object and can be stored in the Active Directory system.

Except where noted otherwise, information in this document applies to both AD DS and AD LDS. The Active Directory system encompasses both AD DS and AD LDS.

Physically, the Active Directory system consists of one or more computer **servers** that run a directory service. In the case of both AD DS and AD LDS, these computers are referred to as **domain controllers (DCs)**. Even though the directory service can be running on multiple computers, these computers replicate the contents of the directory so that a **client** sees a consistent view of the directory no matter which directory server or DC it communicates with. The network protocols that perform this replication are described in [MS-DRSR], [MS-NRPC] section 3.6, and [MS-SRPL].<1>

Note This document, like [MS-ADTS] and [MS-DRSR], uses the term "domain controller" to refer to a DS that runs as either AD DS or AD LDS. Both AD DS and AD LDS are considered to be directory services.

Directory objects can be created and deleted in the directory. Subsequent to its creation, the contents of a directory object can be modified by adding or removing attributes and their values, or by changing the values of existing attributes. Clients can retrieve the contents of directory objects either by reading the attributes of a specific object or by querying for any objects that match client-specified criteria.

The core protocol that clients use to perform operations is LDAP version 3 as described in [MS-ADTS] which is preferred over version 2. Clients can also communicate with the Active Directory system by using the network protocols described in [MS-DRSR], [MS-SRPL], [MS-SAMR], [MS-LSAD], [MS-LSAT], and [MS-DSSP], as well as the Web Service protocols described in [MS-ADDM], [MS-WSTIM], [MS-ADCAP], [MS-WSDS], and [MS-WSPELD]. The set of protocols that the Active Directory system supports depends on whether the system is running as AD DS or AD LDS (see section 2.8).

This document provides a description of the client-server functionality of the Active Directory system and additional specific benefits that a client realizes from

Tech Savvy Productions

being associated with an AD DS directory service (that is, "joined to a domain") that are described later in this document. This document does not describe identity and security concepts that are defined as part of the Windows Protocols Overview Document [MS-WPO].

A directory service is a service that stores and organizes directory objects in a centralized, hierarchical data store. A central concept in the directory service is the **directory tree**. A directory tree is an arrangement of directory objects into a tree structure. Each directory object has exactly one parent directory object, except for the object that serves as the root of the tree and has no parent. Each directory object can have zero or more child objects. Each directory object is assigned a name (the **relative distinguished name (RDN)**) that is unique among its sibling objects. Each directory object can be uniquely identified from among all the other objects in the directory service by its **distinguished name (DN)**, which is formed by concatenating the RDNs of the directory objects along the path from the root of the tree to the specific object. For more information, see [MSDN-DomainTrees].

Domain Interaction

AD DS implements one or more domains within a **forest**. A domain provides a number of services to its clients, primarily related to security and management. The **security principals** of the domain are all available from the AD DS domain controller; therefore, the domain serves as the primary source of identity for the clients of the domain. The domain, through the relevant security protocols, provides the basis for authentication within the domain, allowing **principals** within the domain to establish authenticated connections with each other. During the authentication process, the domain provides authorization information in the form of additional identities that represent groups, enabling authorization decisions to be made.

AD DS stores directory data and manages communication between users and domains. This includes user logon processes, authentication, and directory searches. AD DS provides a distributed database that stores and manages information about network resources and application-specific data from directory-enabled applications. Administrators can use AD DS to organize elements of a network, such as users, computers, and other devices, into a hierarchical containment structure. The hierarchical containment structure

Tech Savvy Productions

includes the Active Directory forest, domains in the forest, and organizational units (OUs) in each domain.

The Active Directory system that runs as the AD DS service on an AD DS domain controller provides certain services and benefits to domain-joined clients. These benefits include support for the **Kerberos** authentication protocol, which domain-joined clients can use to authenticate to the AD DS domain controller and to each other. The benefits also include **certificate** autoenrollment, which automatically deploys certificates to domain-joined computers and to users whose accounts are stored in the directory service, and automatic deployment of administrator-configured **policy** settings.

Many network-related operations depend on domains in order to complete various tasks. This document describes some of these tasks, including:

- Locating a domain controller using **DNS** and **NetBIOS**.
- Joining a domain by creating an account via the Security Account Manager **remote procedure call (RPC)** protocol [MS-SAMR].
- Joining a domain by creating an account via LDAP.
- Removing a **domain member**.

This document includes protocols that are used to communicate with a domain controller and maintain state. It also includes protocols that are used to augment authentication and authorization actions, and protocols that are used to interact with domain controllers.

The domain controller serves a central **role** in an enterprise network by functioning as the root of authority for sets of users and computers. A domain controller aggregates functionality that relates to identity management, authentication, authorization, and other management policy. Clients of the domain functionality in turn rely on the domain controller to establish secure communication, authorize requests, and apply policy. A client of the domain can itself be a server of some other role, for example, a file server that is handling the file storage requirements of other client workstations.

Directory Replication

Tech Savvy Productions

Active Directory is a distributed directory service that stores objects that represent real-world entities, such as users, computers, services, and network resources. Objects in the directory are distributed among all domain controllers in a forest, and all domain controllers can be updated directly. Active Directory replication is the process by which the changes that originate on one domain controller are automatically transferred to other domain controllers that store the same data.

The Active Directory replication model ensures that Active Directory data on one domain controller will eventually converge with the **replica** of the same data on other domain controllers in the same domain. The Active Directory replication model determines how changes to Active Directory data are propagated and tracked automatically between domain controllers. The replication model allows directory data on each domain controller to be updated directly. Each domain controller maintains replication metadata that indicates the update status both for itself and relative to other domain controllers. In addition, the replication model allows each domain controller to request (or pull) only the changes that need to be replicated and to forward changes to other domain controllers that require them.

When a change is made to an object in a directory partition, the value of the changed attribute or attributes is updated on all domain controllers that store a replica of the same directory partition. Domain controllers communicate data updates automatically through Active Directory replication. Their communication about updates is always specific to a single directory partition at a time.

A domain that is run by AD DS can consist of many partitions or **naming contexts (NCs)**. The DN of an object includes enough information to locate a replica of the partition that holds the object. The **global catalog (GC)** contains a partial replica of every NC in the directory. It also contains the **schema** and **configuration naming contexts (config NCs)**. This means that the GC holds a replica of every object in the directory but with only a small number of their attributes. The attributes in the GC are those that are most frequently used in search operations, such as a user's first and last names or logon names, and those required to locate the full replica of the object.

Tech Savvy Productions

Active Directory objects are instances of schema-defined classes, which consist of named sets of attributes. Schema definitions determine whether an attribute can be administratively changed. Attributes that cannot be changed are never updated and therefore never replicated. However, most Active Directory objects have attribute values that can be updated.

Replication within a **site** occurs as a response to changes. On its **NTDS** Settings object, the source domain controller stores a repsTo attribute that lists all servers in the same site that pull replication from it. The Knowledge Consistency Checker (KCC) updates these attributes, as described later in this section.

When a change occurs on a source domain controller, it notifies its destination replication partner, prompting the destination domain controller to request the changes from the source domain controller. The source domain controller either responds to the change request with a replication operation or places the request in a queue if requests are already pending. Each domain controller has a queue of pending replication operations that are processed one at a time.

Directory Replication Service (DRS) Remote Protocol [MS-DRSR] is an RPC protocol for replication and management of data in Active Directory. Domain controllers use the Security Account Manager (SAM) Remote Protocol (Server-to-Server) [MS-SAMS] to forward time-critical database changes to the **primary domain controller (PDC)**, and to forward time-critical database changes from a **read-only domain controller (RODC)** to a **writable NC replica** within the same domain outside the normal replication protocol. This protocol is used only between Active Directory servers in the same domain. Beginning with Windows Server 2008 operating system, this protocol was extended to forward certain write operations that are not time critical from an RODC to a writable NC replica. The SAMS protocol addresses the requirement to propagate a subset of database changes to the PDC more quickly than the Directory Replication Service (DRS) Remote Protocol. This rapid propagation is used for sensitive information when the delay imposed by standard Active Directory replication creates an unwelcome burden on the user or creates a risk to the enterprise. An example of the former is a password change operation; if the password is not made available rapidly, a user can experience unpredictable authentication failures when the new password is tried against domain controllers that have not yet replicated it. An example of the

Tech Savvy Productions

latter is when an account is locked out due to multiple password failures; the lockout condition, and, equally important, the lockout-cleared condition, has to be propagated rapidly throughout the domain.

Replication can be event-driven or schedule-driven as explained in [MS-ADTS] section 3.1.1.1.14.

Knowledge Consistency Checker (KCC)

A domain controller that runs Active Directory is part of a distributed system that performs replication. The KCC is a component that reduces the administrative burden of maintaining a functioning replication topology. The KCC ensures that a replication path exists between the same NCs that are present in different DCs. The KCC is explained in [MS-ADTS] sections 3.1.1.1.13 and 6.2. The KCC helps administrators build a replication topology that incurs minimal cost. This cost is defined by the administrator as explained in [MS-ADTS] section 3.1.1.1.13.

FSMO Roles

Each DC accepts **originating updates** for most attributes of most objects within its writable NC replicas. However, certain updates are accepted only if the DC is the single designated "master" DC for the update. This mechanism is called **flexible single master operation (FSMO)**.

If some or all of the updates to an object are single-mastered, that object belongs to a defined set of objects. [MS-DRSR] section 4.1.10.5.3 (GetReplScope) specifies these sets, which are called **FSMO roles**. Each FSMO role is applicable to a certain scope: either domain-wide, or forest-wide. The domain-wide FSMO roles include the *Infrastructure Master FSMO*, the *Rid Master FSMO*, and the *PDC Emulator FSMO*. The forest-wide FSMO roles include the *Domain Naming FSMO* and the *Schema Master FSMO*. There are no FSMO roles that apply strictly to **application NCs**.

Because a server that is operating as AD LDS does not host domain NCs, it cannot own any of the three domain-specific FSMO roles. It can own the Schema Master FSMO and Domain Naming FSMO roles.

Tech Savvy Productions

In a given NC, each FSMO role is represented by an object. [MS-DRSR] section 4.1.10.5.3 (GetReplScope) specifies these objects, which are called **FSMO role objects**.

The fSMORoleOwner attribute of each FSMO role object is an object reference to the nTDSDSA object of the DC that owns the role; that is, the DC that performs updates to objects in the role. Information about nTDSDSA objects and how they represent DCs are specified in [MS-ADTS] section 6.1.

An originating update to an object within a FSMO role generates an LDAP referral if the DC that receives the request cannot perform the update; the referral is to the DC represented by the nTDSDSA object referenced by the FSMO role object's fSMORoleOwner attribute on the DC that received the request.

The processing of updates affected by FSMO roles is fully specified in [MS-ADTS] section 3.1.1.5.

The IDL_DRSGetNCChanges method ([MS-DRSR] section 4.1.10) makes an originating update to the fSMORoleOwner attribute of a FSMO role object while preserving single-mastering of updates to the FSMO role. The ability to update the fSMORoleOwner attribute in this way is exposed through LDAP as the root DSE updates becomeDomainMaster, becomeInfrastructureMaster, becomePdc, becomePdcWithCheckPoint, becomeRidMaster, and becomeSchemaMaster, as specified in [MS-ADTS] section 3.1.1.3.

Reading the rootDSE attribute valid FSMOs on a DC returns the set of all FSMO roles (represented as FSMO role objects) that the DC will update, as specified in [MS-ADTS] section 3.1.1.3.

Active Directory Trust Management

Active Directory domains rarely exist in isolation. Many Active Directory deployments in customer sites consist of two or more domains that represent boundaries between different geographical, managerial, organizational, or administrative layouts. For example, when company "A" acquires company "B", it quickly becomes necessary for preexisting domains to start trusting each other. Alternatively, in some deployments, servers that have a specific role (such as a mail server) can be members of a "resource domain", easing the management burden by combining like roles under one administrative domain.

Tech Savvy Productions

Communication between disparate domains, especially secure communication that involves authentication and authorization, requires that some stateful knowledge is shared between the peer domains in order for them to trust one another. Some of this knowledge is sensitive, forming the cryptographic basis of trust mechanisms used in protocols such as Kerberos and **Netlogon** RPC. Other state is public knowledge, such as the NetBIOS name of a peer domain, or which **security identifiers** are owned by the peer domain. Information like this plays a crucial role when performing name lookups, which are essential for authorization, locating user accounts, or simply displaying information in some type of user interface.

Active Directory stores trust information in **trusted domain objects (TDOs)** ([MS-ADTS] section 6.1.6.2) and, depending on the kind of trust established, in associated user accounts, **interdomain trust accounts**, for the **trusted domain**. There are different types of trusts that exist between Active Directory domains, as described in [MS-ADTS] section 6.1.6.2. TDOs can be managed through the LSAD protocol ([MS-LSAD] section 3.1.4.7).

Tech Savvy Productions

What is Active Directory: The Ultimate Guide

Michael Reinders
|

JAN 6, 2022

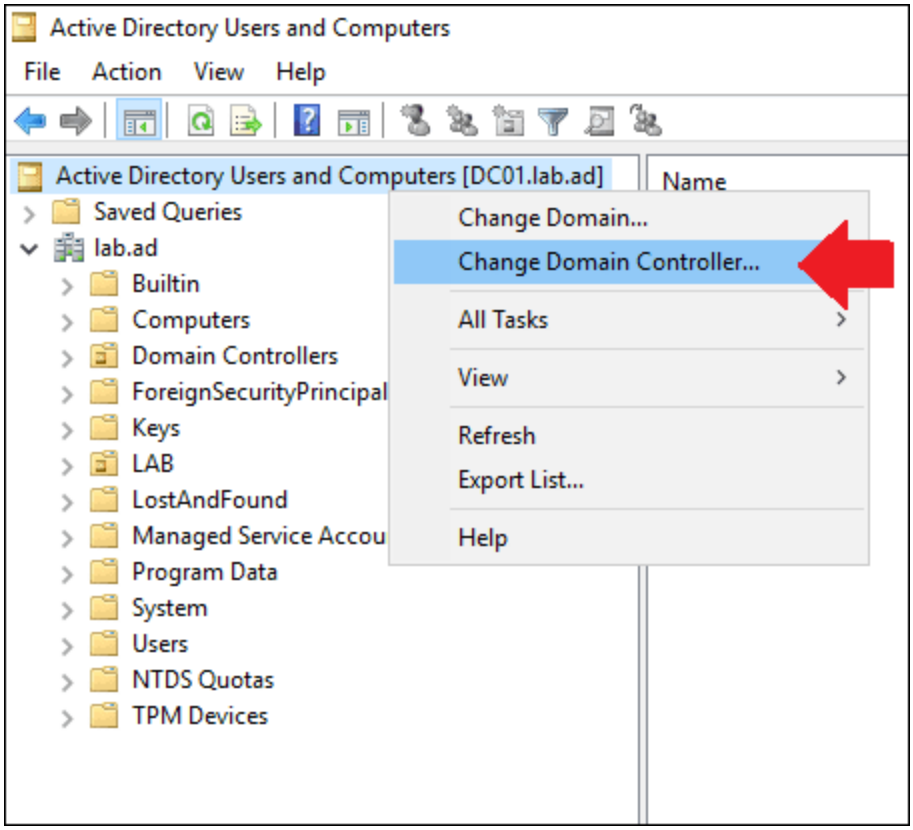
Learn what Active Directory (AD) is and how AD makes it easier for IT to manage their organization's computer resources. Active Directory is especially useful for companies that have to manage lots of endpoints and servers. Read more here.

Table of Contents[hide]

- What is Active Directory and why is it used?
    - What is AD Domain Services?
- Who is Active Directory for?
- How Active Directory works
    - Understanding the structure of AD Domain Services
    - What about other Active Directory services?
- How Active Directory helps IT
- What are the steps in setting up Active Directory?
- Active Directory vs LDAP or NIS
- Windows Server Active Directory vs Azure Active Directory

What is Active Directory and why is it used?
Active Directory (AD) is a directory service from Microsoft that stores information about objects on the network and makes this information easy for administrators and users to find and use. Active Directory uses a structured data store as the basis for a logical, hierarchical organization of directory information. Think of it as a telephone directory but for objects and devices on a computer network.

Tech Savvy Productions



Active Directory Users and Computers

AD controls much of the activity that goes on in your IT environment. In particular, they make sure each person is who they claim to be (authentication), usually by checking the user ID and password they enter and allowing them to access only the data they're allowed to use (authorization).

What is AD Domain Services?

Active Directory Domain Services (AD DS)

A directory is a hierarchical structure that stores information about objects on the network. A directory service, such as Active Directory Domain Services (AD DS), provides methods for storing directory data and making this data available to network users and administrators.

When you install AD DS in your environment, servers install the AD DS Server Role and they become the keys to the kingdom for this directory service, other known as domain controllers (DC). For example, AD DS stores information about user accounts, such as names, passwords, phone numbers, computers, and other devices like printers and other network resources.

Tech Savvy Productions

This allows only authorized users, which are each assigned a unique security identifier (SID), to access protected information on the same network to. The individual attributes of a user: name, address, location, manager, and so on, are stored in the directory database.

What determines what users can do with the various resources in your network? Permissions. Tokens, are granted every time a user logs on to a computer. Those access tokens contain the keys that the user needs to be able to open the G: drive and open that Excel file in the Sales folder.

Who is Active Directory for?
Well, honestly, I would have to say AD is for almost any company or organization out there. I have yet to come across an environment that didn't utilize Active Directory Domain Services. AD simplifies life for administrators and end users while enhancing security for organizations.

Administrators and end users share the centralized user and rights management, as well as centralized control over computer and user configurations through the AD Group Policy feature. Users can authenticate once and then seamlessly access any resources in the domain for which they're authorized (single sign-on).

Plus, files are stored in a central repository where they can be shared with other users to ease collaboration and backed up properly by IT teams to ensure business continuity.

How Active Directory works
So, what's at the heart of how this service introduced in Windows 2000 works? Well, Active Directory Domain Services (AD DS) is a part of the Windows Server operating system. The servers that run AD DS are called domain controllers (DCs).

Back in the days of Windows NT Server, there was a specific role, the Primary Domain Controller (PDC), that was specified to hold a lot of the key roles across the domain amongst the other DCs.

Organizations normally have multiple DCs for redundancy, performance, and business continuity. Each one has a copy of the directory for the entire domain. Changes made to the directory on one domain controller — such as password update or the deletion of a user account — are replicated using a replication service to the other DCs so they all stay up to date.

Tech Savvy Productions

In addition, the schema of the database central to AD DS includes the various attributes each person, group, or computer can hold.

A Global Catalog (GC) server is a domain controller that stores a complete copy of all objects in the directory of its domain and a partial copy of all objects of all other domains in the forest; this enables users and applications to find objects in any domain of their forest.

Desktops, laptops, and other devices running Windows (rather than Windows Server) can be part of an AD environment, but they do not run AD DS.

AD DS relies on several established protocols and standards, including LDAP (Lightweight Directory Access Protocol), Kerberos, and DNS (Domain Name System).

IT admins can use various tools to access AD information, including the Active Directory Users and Computers (ADUC), the AD Administrative Center, as well as AD Sites and Services. I invite you to check my separate post about how to access Active Directory.

It's important to understand that Active Directory is only for on-premises Microsoft environments. Microsoft environments in the cloud use Azure Active Directory, which serves the same purposes as its on-prem namesake. AD and Azure AD are separate but can work together to some degree if your organization has both on-premises and cloud IT environments (a hybrid deployment). I'll give you more on what Azure Active Directory (AAD) is at the end of this post.

Understanding the structure of AD Domain Services

There are three core logical structures to AD DS. They are domains, trees, and forests. A domain is a group of related users, computers, and other AD objects, such as all the AD objects for your company's various headquarters and branch offices. Multiple domains can be combined into a tree, and multiple trees can be grouped into a forest.

Keep in mind that a domain is a management boundary. The objects for a given domain are stored in a single database and can be managed together. A forest is a security boundary. Objects in different forests are not able to interact with each other unless the administrators of each forest create a trust between them.

Tech Savvy Productions

For instance, if you have multiple disjointed business units, it's likely to assist you with managing them if you create multiple forests.

What about other Active Directory services?

So, is Active Directory simply AD DS? Nope. There are a few other software features that make up the whole of AD. The most common one after AD DS is Active Directory Federation Services (AD FS).

Active Directory Federation Service (AD FS) enables Federated Identity and Access Management (IAM) by securely sharing digital identity and entitlements rights across security and enterprise boundaries.

> *AD FS extends the ability to use single sign-on functionality that is available within a single security or enterprise boundary to Internet-facing applications to enable customers, partners, and suppliers a streamlined user experience while accessing the web-based applications of an organization.*

Starting in Windows Server 2012 R2, AD FS included a federation service role service that acts as an identity provider (authenticates users to provide security tokens to applications that trust AD FS) or as a federation provider (consumes tokens from other identity providers and then provides security tokens to applications that trust AD FS).

The other main services include:

- Active Directory Lightweight Directory Services (LDAP) – a light-weight instance of Active Directory that can be used for apps that need a directory service.
- Active Directory Certificate Services (AD CS) – manages and issues digital certificates to users and computers.
- Active Directory Rights Management Services (AD RMS) – lets organizations granularly control how documents are used by entities that have access.
- Active Directory Federation Services (AD FS) – used for enabling access and management between enterprises.

How Active Directory helps IT

There are several benefits for IT folk in using AD DS in your organization:

- You can choose how to organize your data across the various teams of users and roles in your organization.

Tech Savvy Productions

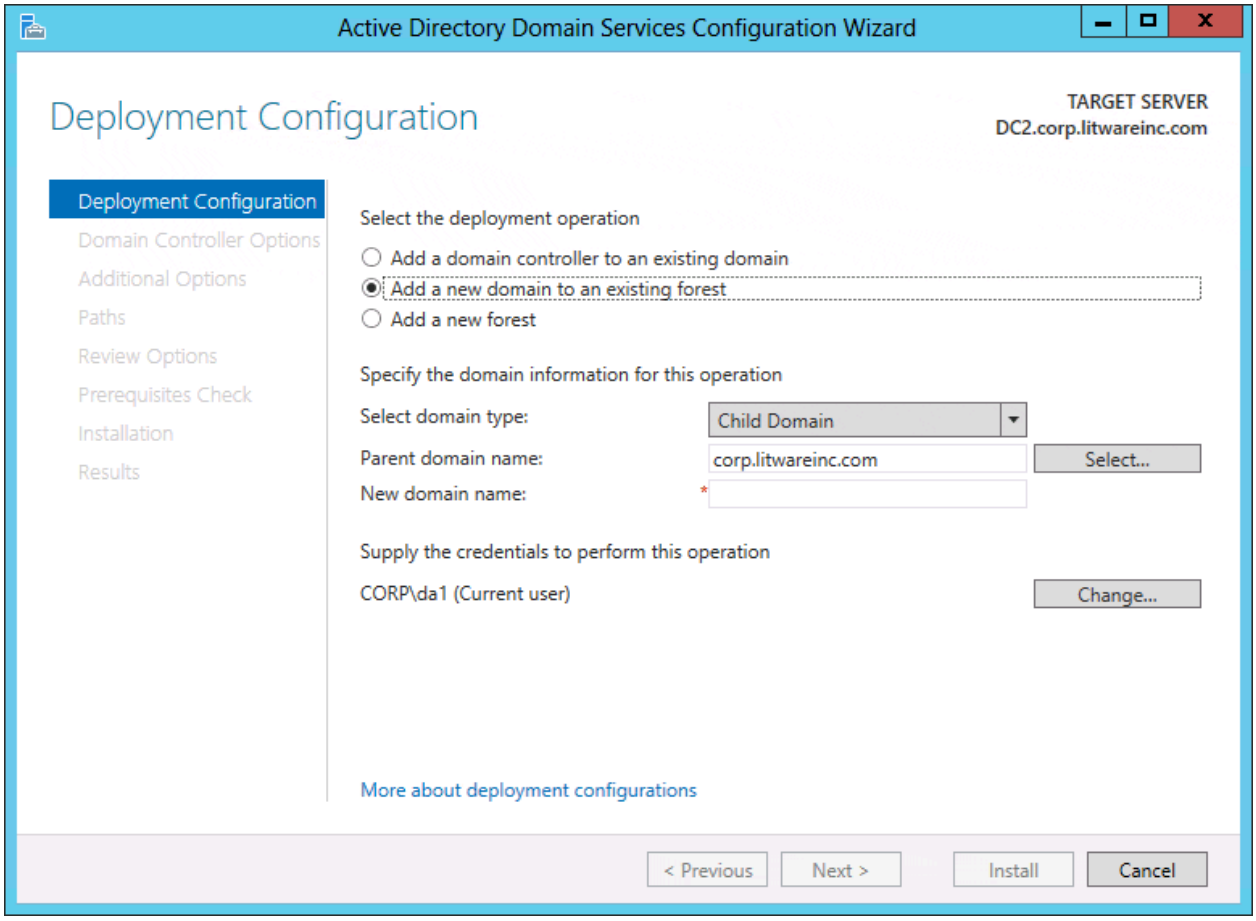
- You can manage AD DS from any computer on the network, if necessary.
- AD DS provides built-in replication and redundancy: if one domain controller (DC) fails, another DC picks up the load.
- All access to network resources goes through AD DS, which keeps network access rights management centralized.

The largest benefit of implementing Active Directory for your users is a centralized token to log in to their computer. Instead of needing this user account to access the resources on this server, and another for server two, and so on, you can grant tokens to a single account that a user can use to seamlessly access resources and printers across the enterprise.

What are the steps in setting up Active Directory?

There are several articles on Petri that contain step-by-step tutorials on how to install AD in your environment. There is also a considerable amount of planning that is required in laying out the design of your infrastructure. Number of servers, physical or virtual, on-premises or in Azure, etc. I'll give you the high-level outline of the steps required.

Tech Savvy Productions



Adding a new AD domain to an existing forest

1. Install the AD DS Role in Server Manager or PowerShell.
2. Promote the server to a domain controller.
3. Reboot your server and start adding additional domain controllers, and then user account objects.
4. Start joining organization devices and computers to the domain.

Once you've built your Active Directory environment, you'll need to learn how to back it up and restore it to prevent any possible interruptions to your environment. You can check out our guides on how to back up Active Directory and how to restore Active Directory on Petri.

Tech Savvy Productions

Active Directory vs LDAP or NIS

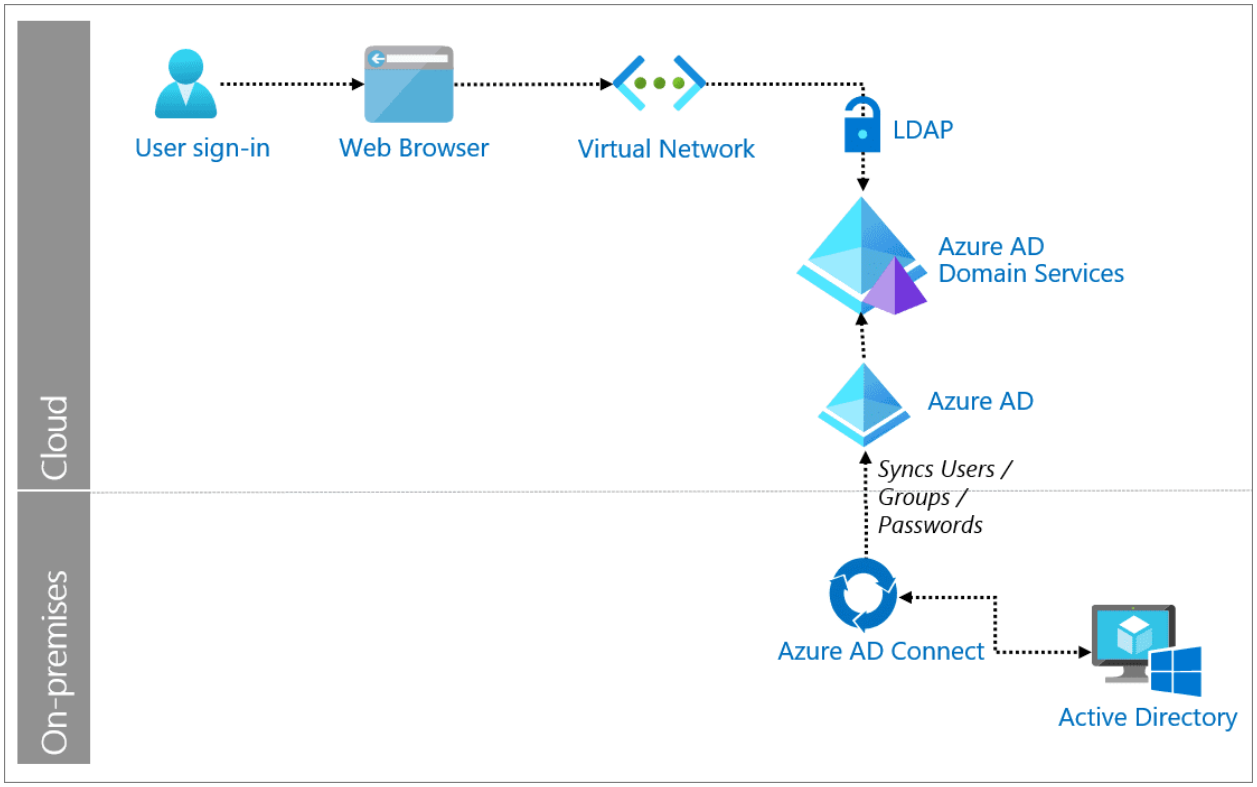
We already explained what Active Directory is above. But, what about LDAP and NIS? What are they, how are they different?

The Network Information Service or NIS (originally called Yellow Pages or YP) is Sun Microsystems client-server Directory Service protocol for distributed system configuration data such as user and host names between computers on a computer network. NIS is a discovery mechanism.

What is LDAP? Companies store usernames, passwords, email addresses, printer connections, and other static data within directories. LDAP is an open, vendor-neutral application protocol for accessing and maintaining that data.

LDAP can also utilize authentication, so users can sign on just once and access many different files on the server (SSO).

Windows Server Active Directory vs Azure Active Directory



Syncing your on-premises AD environment with Azure Active Directory

Tech Savvy Productions

Azure Active Directory (Azure AD, or AAD) is the next evolution of identity and access management solutions for the cloud. As I stated at the top of this post, Microsoft introduced Active Directory Domain Services in Windows 2000 to give organizations the ability to manage multiple on-premises infrastructure components and systems using a single identity per user.

Azure AD takes this approach to the next level by providing organizations with an Identity as a Service (IDaaS) solution for all their apps across cloud and on-premises.

Feel free to read our separate post about Active Directory, Azure Active Directory, and other identity providers to see all the core differences between AD and AAD and how managing them differs.

Active Directory Glossary

This document uses the following terms:

access control entry (ACE): An entry in an **access control list (ACL)** that contains a set of user rights and a **security identifier (SID)** that identifies a principal for whom the rights are allowed, denied, or audited.

access control list (ACL): A list of **access control entries (ACEs)** that collectively describe the security rules for authorizing access to some resource; for example, an object or set of objects.

account: A user (including machine account), **group**, or alias object. Also a synonym for security principal or principal.

account database: The portion of the directory that maintains the accounts for the **principals** of the domain. In Windows NT-4 style domains, the account database includes all information in the domain; in Active Directory–style domains, the account database contains a subset of the entire LDAP-accessible directory that the Active Directory–style domain hosts.

Tech Savvy Productions

Active Directory: The Windows implementation of a general-purpose **directory service**, which uses **LDAP** as its primary access protocol. **Active Directory** stores information about a variety of objects in the network such as user accounts, computer accounts, groups, and all related credential information used by **Kerberos** [MS-KILE]. **Active Directory** is either deployed as **Active Directory Domain Services (AD DS)** or **Active Directory Lightweight Directory Services (AD LDS)**, which are both described in [MS-ADOD]: Active Directory Protocols Overview.

Active Directory Domain Services (AD DS): A **directory service (DS)** implemented by a **domain controller (DC)**. The **DS** provides a data store for objects that is distributed across multiple **DCs**. The **DCs** interoperate as peers to ensure that a local change to an object replicates correctly across **DCs**. AD DS is a deployment of **Active Directory** [MS-ADTS].

Active Directory Lightweight Directory Services (AD LDS): A **directory service (DS)** implemented by a **domain controller (DC)**. AD LDS is a deployment of **Active Directory** [MS-ADTS]. The most significant difference between **AD LDS** and **Active Directory Domain Services (AD DS)** is that **AD LDS** does not host **domain naming contexts (domain NCs)**. A server can host multiple **AD LDS DCs**. Each **DC** is an independent **AD LDS** instance, with its own independent state. **AD LDS** can be run as an operating system **DS** or as a directory service provided by a standalone application (Active Directory Application Mode (ADAM)).

Active Directory-style domain: A domain that is created as described in [MS-ADTS] section 1. Active Directory-style domains implement Active Directory, LDAP, Kerberos authentication, and advanced configurations and features that are not supported in Windows NT 4.0-style domains.

ADCAP: The Active Directory Web Services Custom Action Protocol [MS-ADCAP].

application naming context (application NC): A specific type of **naming context (NC)**, or an instance of that type, that supports only full replicas (no partial replicas). An **application NC** cannot contain security principal objects in Active Directory Domain Services (AD DS), but can contain security principal objects in Active Lightweight Directory Services (AD LDS). A **forest** can have

Tech Savvy Productions

zero or more **application NCs** in either AD DS or AD LDS. An application NC can contain dynamic objects. **Application NCs** do not appear in the **global catalog (GC)**. The root of an **application NC** is an object of class domainDNS.

attribute: An identifier for a single or multivalued data element that is associated with a directory object. An object consists of its **attributes** and their values. For example, cn (common name), street (street address), and mail (email addresses) can all be **attributes** of a user object. An **attribute's** schema, including the syntax of its values, is defined in an attributeSchema object.

backup domain controller (BDC): A **domain controller (DC)** that receives a copy of the **domain** directory database from the **primary domain controller (PDC)**. This copy is synchronized periodically and automatically with the **primary domain controller (PDC)**. BDCs also authenticate user logons and can be promoted to function as the **PDC**. There is only one **PDC** or **PDC** emulator in a **domain**, and the rest are **backup domain controllers**.

binary large object (BLOB): A collection of binary data stored as a single entity in a database.

binding: The string representation of the protocol sequence, NetworkAddress, and optionally the **endpoint**. Also referred to as "string binding". For more information, see [C706] section "String Bindings".

certificate: A certificate is a collection of attributes and extensions that can be stored persistently. The set of attributes in a certificate can vary depending on the intended usage of the certificate. A certificate securely binds a public key to the entity that holds the corresponding private key. A certificate is commonly used for authentication and secure exchange of information on open networks, such as the Internet, extranets, and intranets. Certificates are digitally signed by the issuing certification authority (CA) and can be issued for a user, a computer, or a service. The most widely accepted format for certificates is defined by the ITU-T X.509 version 3 international standards. For more information about attributes and extensions, see [RFC3280] and [X509] sections 7 and 8.

client: Synonym for **client computer**.

Tech Savvy Productions

client computer: The client machine in the **domain** or network topology of clients, servers, and **domain controllers**. Alternatively, a computer that is not a **domain controller server**; the computer may or may not be joined to a domain.

configuration naming context (config NC): A specific type of **naming context (NC)**, or an instance of that type, that contains configuration information. In **Active Directory**, a single **config NC** is shared among all **domain controllers (DCs)** in the forest. A **config NC** cannot contain security principal objects.

credential: Previously established, authentication data that is used by a security principal to establish its own identity. When used in reference to the Netlogon Protocol, it is the data that is stored in the NETLOGON_CREDENTIAL structure.

deleted-object: An object that has been deleted, but remains in storage until a configured amount of time (the deleted-object lifetime) has passed, after which the object is transformed to a recycled-object. Unlike a recycled-object or a **tombstone**, a **deleted-object** maintains virtually all the state of the object before deletion, and can be undeleted without loss of information. **Deleted-objects** exist only when the Recycle Bin optional feature is enabled.

directory: The database that stores information about objects such as users, groups, computers, printers, and the **directory service** that makes this information available to users and applications.

directory object: A **Lightweight Directory Access Protocol (LDAP)** object, as specified in [RFC2251], that is a specialization of an object.

directory service (DS): A service that stores and organizes information about a computer network's users and network shares, and that allows network administrators to manage users' access to the shares. See also **Active Directory**.

directory tree: An **LDAP directory service** is organized into a hierarchical tree structure in which each **directory object** has exactly one parent **directory object** (except for one object that serves as the root of the tree) and zero or more child **directory objects**.

Tech Savvy Productions

distinguished name (DN): In the **Active Directory** directory service, the unique identifier of an object in **Active Directory**, as described in [MS-ADTS] and [RFC2251].

domain: A set of users and computers sharing a common namespace and management infrastructure. At least one computer member of the set must act as a **domain controller (DC)** and host a member list that identifies all members of the domain, as well as optionally hosting the **Active Directory** service. The domain controller provides authentication of members, creating a unit of trust for its members. Each domain has an identifier that is shared among its members. For more information, see [MS-AUTHSOD] section 1.1.1.5 and [MS-ADTS].

domain account: A stored set of **attributes** representing a principal used to authenticate a user or machine to an **Active Directory** domain.

domain client: A client computer that is joined to a domain. The domain client can be a client or a server that offers other services to its clients. When the domain client acts as a supplicant to another domain client, the supplicant is referred to as a domain client in a workstation role and the latter as a domain client in a server role.

domain controller (DC): The service, running on a server, that implements **Active Directory**, or the server hosting this service. The service hosts the data store for objects and interoperates with other **DCs** to ensure that a local change to an object replicates correctly across all **DCs**. When **Active Directory** is operating as **Active Directory Domain Services (AD DS)**, the **DC** contains full NC replicas of the **configuration naming context (config NC)**, schema naming context (schema NC), and one of the **domain NCs** in its **forest**. If the **AD DS DC** is a global catalog server (GC server), it contains partial NC replicas of the remaining **domain NCs** in its **forest**. For more information, see [MS-AUTHSOD] section 1.1.1.5.2 and [MS-ADTS]. When **Active Directory** is operating as **Active Directory Lightweight Directory Services (AD LDS)**, several **AD LDS DCs** can run on one server. When **Active Directory** is operating as **AD DS**, only one **AD DS DC** can run on one server. However, several **AD LDS DCs** can coexist with one **AD DS DC** on one server. The **AD LDS DC** contains full NC replicas of the **config NC** and the schema NC in its **forest**.

Tech Savvy Productions

The domain controller is the server side of Authentication Protocol Domain Support [MS-APDS].

domain controller server: A domain member, which can be a client or a server that offers other services to its clients. When the domain client acts as a supplicant to another domain client, the supplicant is referred to as a domain client in a workstation role and the latter as a domain client in a server role.

domain functional level: A specification of functionality available in a **domain**. Must be less than or equal to the DC functional level of every **domain controller (DC)** that hosts a **replica** of the **domain's naming context (NC)**. For information on defined levels, corresponding features, information on how the **domain functional level** is determined, and supported **domain controllers**, see [MS-ADTS] sections 6.1.4.2 and 6.1.4.3. When **Active Directory** is operating as **Active Directory Lightweight Directory Services (AD LDS)**, **domain functional level** does not exist.

domain member (member machine): A machine that is joined to a domain by sharing a secret between the machine and the domain.

Domain Name System (DNS): A hierarchical, distributed database that contains mappings of domain names to various types of data, such as IP addresses. DNS enables the location of computers and services by user-friendly names, and it also enables the discovery of other information stored in the database.

domain naming context (domain NC): A specific type of **naming context (NC)**, or an instance of that type, that represents a **domain**. A **domain NC** can contain security principal objects; no other type of **NC** can contain security principal objects. **Domain NCs** appear in the **global catalog (GC)**. A **domain NC** is hosted by one or more **domain controllers (DCs)** operating as **AD DS**. In **AD DS**, a **forest** has one or more **domain NCs**. A domain NC cannot exist in AD LDS. The root of a **domain NC** is an object of class domainDNS; for directory replication [MS-DRSR], see domainDNS.

domain object: A unit of data storage in a **domain** that is maintained and made available to **domain** members by a **domain controller (DC)**.

endpoint: A network-specific address of a remote procedure call (RPC) server process for remote procedure calls. The actual name and type of the

Tech Savvy Productions

endpoint depends on the **RPC** protocol sequence that is being used. For example, for RPC over TCP (RPC Protocol Sequence ncacn_ip_tcp), an endpoint might be TCP port 1025. For RPC over Server Message Block (RPC Protocol Sequence ncacn_np), an endpoint might be the name of a named pipe. For more information, see [C706].

enumeration context: A session context that represents a specific traversal through a logical sequence of XML element information items using the Pull operation defined in WS-Enumeration specification. See [WSENUM].

extended control: A mechanism that is used to specify extension information in a **Lightweight Directory Access Protocol (LDAP)** version 3 operation. It is documented in [RFC2251] section 4.1.12, Controls, where it is referred to as a "control".

flexible single master operation (FSMO): A read or update operation on a **naming context (NC)**, such that the operation must be performed on the single designated master **replica** of that **NC**. The master **replica** designation is "flexible" because it can be changed without losing the consistency gained from having a single master. This term, pronounced "fizmo", is never used alone; see also **FSMO role**, FSMO role owner, and FSMO object.

forest: For **Active Directory Domain Services (AD DS)**, a set of **naming contexts (NCs)** consisting of one schema naming context (schema NC), one **configuration naming context (config NC)**, one or more **domain naming contexts (domain NCs)**, and zero or more **application naming contexts (application NCs)**. Because a set of **NCs** can be arranged into a tree structure, a **forest** is also a set containing one or several trees of **NCs**. For **AD LDS**, a set of **NCs** consisting of one schema NC, one **config NC**, and zero or more **application NCs**. (In Microsoft documentation, an **AD LDS forest** is called a "configuration set".)

FSMO role: A set of objects that can be updated in only one **naming context (NC)** replica (the FSMO role owner's replica) at any given time. For more information, see [MS-ADTS] section 3.1.1.1.11. See also FSMO role owner.

Tech Savvy Productions

FSMO role object: An object in a **directory** that represents a specific **FSMO role**. This object is an element of the **FSMO role** and contains the fSMORoleOwner attribute.

fully qualified domain name (FQDN): (1) An unambiguous domain name that gives an absolute location in the **Domain Name System's (DNS)** hierarchy tree, as defined in [RFC1035] section 3.1 and [RFC2181] section 11.

(2) In **Active Directory**, a **fully qualified domain name (FQDN) (1)** that identifies a **domain**.

global catalog (GC): A unified partial view of multiple **naming contexts (NCs)** in a distributed partitioned directory. The **Active Directory** directory service **GC** is implemented by GC servers. The definition of **global catalog** is specified in [MS-ADTS] section 3.1.1.1.8.

group: A collection of objects that can be treated as a whole.

identity: An account that represents a person (user account), an application (service account), and computers that participate in the domain (machine accounts). A password is used by the system as proof of an identity.

interdomain trust account: An account that stores information associated with a **domain** trust in the **domain controllers (DCs)** of the **domain** that is trusted to perform authentication.

Interface Definition Language (IDL): The International Standards Organization (ISO) standard language for specifying the interface for remote procedure calls. For more information, see [C706] section 4.

Kerberos: An authentication system that enables two parties to exchange private information across an otherwise open network by assigning a unique key (called a ticket) to each user that logs on to the network and then embedding these tickets into messages sent by the users. For more information, see [MS-KILE].

Key Distribution Center (KDC): The **Kerberos** service that implements the authentication and ticket granting services specified in the **Kerberos** protocol. The service runs on computers selected by the administrator of the realm or domain; it is not present on every machine on the network. It must have

Tech Savvy Productions

access to an **account** database for the realm that it serves. **KDCs** are integrated into the **domain controller** role. It is a network service that supplies tickets to **clients** for use in authenticating to services.

Lightweight Directory Access Protocol (LDAP): The primary access protocol for **Active Directory**. Lightweight Directory Access Protocol (LDAP) is an industry-standard protocol, established by the Internet Engineering Task Force (IETF), which allows users to query and update information in a **directory service (DS)**, as described in [MS-ADTS]. The Lightweight Directory Access Protocol can be either version 2 [RFC1777] or version 3 [RFC3377].

mailslot: A mechanism for one-way interprocess communications (IPC). For more information, see [MSLOT] and [MS-MAIL].

member server: A server that is joined to a domain and is not acting as an **Active Directory domain controller (DC)**. Member servers typically function as file servers, application servers, and so on and defer user authentication to the domain controller.

mutual authentication: A mode in which each party verifies the identity of the other party, as described in [RFC3748] section 7.2.1.

naming context (NC): An **NC** is a set of objects organized as a tree. It is referenced by a DSName. The **DN** of the DSName is the distinguishedName **attribute** of the tree root. The GUID of the DSName is the objectGUID **attribute** of the tree root. The **security identifier (SID)** of the DSName, if present, is the objectSid **attribute** of the tree root; for **Active Directory Domain Services (AD DS)**, the **SID** is present if and only if the **NC** is a **domain naming context (domain NC)**. **Active Directory** supports organizing several **NCs** into a tree structure.

NetBIOS: A particular network transport that is part of the LAN Manager protocol suite. **NetBIOS** uses a broadcast communication style that was applicable to early segmented local area networks. A protocol family including name resolution, datagram, and connection services. For more information, see [RFC1001] and [RFC1002].

Netlogon: The Netlogon Remote Protocol, as specified in [MS-NRPC].

Tech Savvy Productions

Network Data Representation (NDR): A specification that defines a mapping from **Interface Definition Language (IDL)** data types onto octet streams. **NDR** also refers to the runtime environment that implements the mapping facilities (for example, data provided to **NDR**). For more information, see [MS-RPCE] and [C706] section 14.

NT Directory Service (NTDS): A previous name for **Active Directory**.

object class: A set of restrictions on the construction and update of objects. An **object class** can specify a set of must-have attributes (every object of the class must have at least one value of each) and may-have attributes (every object of the class may have a value of each). An **object class** can also specify the allowable classes for the parent object of an object in the class. An **object class** can be defined by single inheritance; an object whose class is defined in this way is a member of all **object classes** used to derive its most specific class. An **object class** is defined in a classSchema object. See section 1 of [MS-ADTS] and section 1 of [MS-DRSR].

originating update: An update that is performed to an NC replica via any protocol except replication. An **originating update** to an attribute or link value generates a new stamp for the attribute or link value.

policy: A collection of settings that contains global settings, profile settings, firewall rules, and connection security rules. Together these settings specify how the host firewall and Internet Protocol security (IPsec) behave on the client computer.

primary domain controller (PDC): A **domain controller (DC)** designated to track changes made to the accounts of all computers on a **domain**. It is the only computer to receive these changes directly, and is specialized so as to ensure consistency and to eliminate the potential for conflicting entries in the **Active Directory** database. A **domain** has only one **PDC**.

principal: A unique entity identifiable by a **security identifier (SID)** that is typically the requester of access to securable objects or resources. It often corresponds to a human user but can also be a computer or service. It is sometimes referred to as a **security principal**.

Tech Savvy Productions

read-only domain controller (RODC): A **domain controller (DC)** that does not accept **originating updates**. Additionally, an **RODC** does not perform outbound replication. An RODC cannot be the primary domain controller (PDC) for its domain.

relative distinguished name (RDN): In the **Active Directory** directory service, the unique name of a child element relative to its parent in Active Directory. The RDN of a child element combined with the **fully qualified domain name (FQDN) (2)** of the parent forms the FQDN of the child.

relative identifier (RID): The last item in the series of SubAuthority values in a **security identifier (SID)** [SIDD]. It distinguishes one account or group from all other accounts and groups in the domain. No two accounts or groups in any domain share the same RID.

remote procedure call (RPC): A communication protocol used primarily between client and server. The term has three definitions that are often used interchangeably: a runtime environment providing for communication facilities between computers (the RPC runtime); a set of request-and-response message exchanges between computers (the RPC exchange); and the single message from an RPC exchange (the RPC message). For more information, see [C706].

replica: A variable containing a set of objects.

replica domain controller / replica directory server: A server that contains a replicated copy of the directory and is able to answer client requests over any protocol that the directory service supports.

role: The domain role quantifies the relationship between a computer and a domain. Domain roles include the following: Joined: Linked to a domain for purposes of policy and security. Standalone: Not associated with any domain. Domain controller: Linked to a domain, and hosting that domain.

SASL: The Simple Authentication and Security Layer, as described in [RFC2222]. This is an authentication mechanism used by the **Lightweight Directory Access Protocol (LDAP)**.

Tech Savvy Productions

schema: The set of **attributes** and **object classes** that govern the creation and update of objects.

Secure Sockets Layer (SSL): A security protocol that supports confidentiality and integrity of messages in client and server applications that communicate over open networks. SSL supports server and, optionally, client authentication using X.509 certificates [X509] and [RFC5280]. SSL is superseded by **Transport Layer Security (TLS)**. TLS version 1.0 is based on SSL version 3.0 [SSL3].

security descriptor: A data structure containing the security information associated with a securable object. A **security descriptor** identifies an object's owner by its **security identifier (SID)**. If access control is configured for the object, its **security descriptor** contains a discretionary access control list (DACL) with **SIDs** for the **security principals** who are allowed or denied access. Applications use this structure to set and query an object's security status. The **security descriptor** is used to guard access to an object as well as to control which type of auditing takes place when the object is accessed. The **security descriptor** format is specified in [MS-DTYP] section 2.4.6; a string representation of **security descriptors**, called SDDL, is specified in [MS-DTYP] section 2.5.1.

security identifier (SID): An identifier for **security principals** that is used to identify an account or a group. Conceptually, the **SID** is composed of an account authority portion (typically a **domain**) and a smaller integer representing an identity relative to the account authority, termed the **relative identifier (RID)**. The **SID** format is specified in [MS-DTYP] section 2.4.2; a string representation of **SIDs** is specified in [MS-DTYP] section 2.4.2 and [MS-AZOD] section 1.1.1.2.

security principal: An entity that is associated with a human user or a program that can be authenticated. At a minimum, it has two basic attributes, a name and an identifier, that uniquely identifies it and makes it meaningful to the system, administrators, and users. A security principal is also known as a principal or an account.

server: A domain controller. Used as a synonym for domain controller. See [MS-ADOD]

Tech Savvy Productions

Server Message Block (SMB): A protocol that is used to request file and print services from server systems over a network. The SMB protocol extends the CIFS protocol with additional security, file, and disk management support. For more information, see [CIFS] and [MS-SMB].

service (SRV) resource record: A **Domain Name System (DNS)** resource record used to identify computers that host specific services, as specified in [RFC2782]. SRV resource records are used to locate **domain controllers (DCs)** for **Active Directory**.

service principal name (SPN): The name a client uses to identify a service for mutual authentication. (For more information, see [RFC1964] section 2.1.1.) An **SPN** consists of either two parts or three parts, each separated by a forward slash ('/'). The first part is the service class, the second part is the host name, and the third part (if present) is the service name. For example, "ldap/dc-01.fabrikam.com/fabrikam.com" is a three-part **SPN** where "ldap" is the service class name, "dc-01.fabrikam.com" is the host name, and "fabrikam.com" is the service name. See [SPNNAMES] for more information about **SPN** format and composing a unique **SPN**.

service ticket: A ticket for any service other than the **ticket-granting service (TGS)**. A **service ticket** serves only to classify a ticket as not a **ticket-granting ticket (TGT)** or cross-realm TGT, as specified in [RFC4120].

site: A collection of one or more well-connected (reliable and fast) TCP/IP subnets. By defining **sites** (represented by site objects) an administrator can optimize both **Active Directory** access and **Active Directory** replication with respect to the physical network. When users log in, **Active Directory** clients find **domain controllers (DCs)** that are in the same **site** as the user, or near the same **site** if there is no **DC** in the **site**. See also Knowledge Consistency Checker (KCC). For more information, see [MS-ADTS].

SOAP: A lightweight protocol for exchanging structured information in a decentralized, distributed environment. **SOAP** uses **XML** technologies to define an extensible messaging framework, which provides a message construct that can be exchanged over a variety of underlying protocols. The framework has been designed to be independent of any particular

Tech Savvy Productions

programming model and other implementation-specific semantics. SOAP 1.2 supersedes SOAP 1.1. See [SOAP1.2-1/2003].

ticket-granting service (TGS): A service that issues tickets for admission to other services in its own domain or for admission to the ticket-granting service in another domain.

ticket-granting ticket (TGT): A special type of ticket that can be used to obtain other tickets. The TGT is obtained after the initial authentication in the Authentication Service (AS) exchange; thereafter, users do not need to present their credentials, but can use the TGT to obtain subsequent tickets.

tombstone: An object that has been deleted, but remains in storage until a configured amount of time (the tombstone lifetime) has passed, after which the object is permanently removed from storage. By keeping the **tombstone** in existence for the tombstone lifetime, the deleted state of the object is able to replicate. **Tombstones** exist only when the Recycle Bin optional feature is not enabled.

Transmission Control Protocol (TCP): A protocol used with the Internet Protocol (IP) to send data in the form of message units between computers over the Internet. TCP handles keeping track of the individual units of data (called packets) that a message is divided into for efficient routing through the Internet.

Transport Layer Security (TLS): A security protocol that supports confidentiality and integrity of messages in client and server applications communicating over open networks. TLS supports server and, optionally, client authentication by using X.509 certificates (as specified in [X509]). TLS is standardized in the IETF TLS working group.

trusted domain: A domain that is trusted to make authentication decisions for security principals in that domain.

trusted domain object (TDO): A collection of properties that define a trust relationship with another domain, such as direction (outbound, inbound, or both), trust attributes, name, and security identifier of the other domain. For more information, see [MS-ADTS].

Tech Savvy Productions

User Datagram Protocol (UDP): The connectionless protocol within TCP/IP that corresponds to the transport layer in the ISO/OSI reference model.

user principal name (UPN): A user account name (sometimes referred to as the user logon name) and a domain name that identifies the domain in which the user account is located. This is the standard usage for logging on to a Windows domain. The format is: someone@example.com (in the form of an email address). In **Active Directory**, the userPrincipalName **attribute** of the account object, as described in [MS-ADTS].

Windows NT 4.0-style domain: A domain that is created from Windows NT 4.0 operating system servers with an account database that includes all the information in the domain. Windows NT 4.0-style domains do not implement Active Directory, LDAP directories, or Kerberos authentication.

writable naming context (NC) replica: A **naming context (NC)** replica that accepts originating updates. A **writable NC replica** is always full, but a full NC replica is not always writable. Partial replicas are not writable. See also read-only full NC replica.

XML: The Extensible Markup Language, as described in [XML1.0].

challenge: A 64-bit **nonce** generated on the client side.

computer name: The **DNS** or NetBIOS name.

computer object: An object of class computer. A **computer object** is a **security principal** object; the principal is the operating system running on the computer. The shared secret allows the operating system running on the computer to authenticate itself independently of any user running on the system. See **security principal**.

credential: Previously established, authentication data that is used by a security principal to establish its own identity. When used in reference to the Netlogon Protocol, it is the data that is stored in the NETLOGON_CREDENTIAL structure.

Tech Savvy Productions

database: For the purposes of the Netlogon RPC, a database is a collection of user accounts, machine accounts, aliases, groups, and policies, managed by a component. The database, or the component managing the database, must expose a mechanism to enable Netlogon to gather changes from and apply changes to the database. Additionally, it must export a database serial number in order to track changes for efficient replication.

database serial number: A numeric value that is incremented each time a database transaction is applied to the database.

decryption: In cryptography, the process of transforming encrypted information to its original clear text form.

delta: One of a set of possible changes that can be made to a **database**.

direct trust: A type of authentication functionality in which one **domain** accepts another **domain** as an authoritative source to provide object authentication and other **Active Directory** services for that other **domain**. For example, if a **direct trust** is established from **domain**, DOMAIN-A, to **domain**, DOMAIN-B, DOMAIN-A trusts DOMAIN-B. If a **domain**, DOMAIN-A, must authenticate an object, such as a user account, from a **domain**, DOMAIN-B, DOMAIN-A requests that DOMAIN-B authenticate the user account, and DOMAIN-A will treat the response from DOMAIN-B as reliable.

directory service (DS): A service that stores and organizes information about a computer network's users and network shares, and that allows network administrators to manage users' access to the shares. See also **Active Directory**.

DNS name: A **fully qualified domain name (FQDN)**.

domain: A set of users and computers sharing a common namespace and management infrastructure. At least one computer member of the set must act as a **domain controller (DC)** and host a member list that identifies all members of the domain, as well as optionally hosting the **Active Directory** service. The domain controller provides authentication of members, creating a unit of trust for its members. Each domain has an identifier that is shared

Tech Savvy Productions

among its members. For more information, see [MS-AUTHSOD] section 1.1.1.5 and [MS-ADTS].

domain account: A stored set of attributes representing a principal used to authenticate a user or machine to an **Active Directory** domain.

domain controller (DC): The service, running on a server, that implements **Active Directory**, or the server hosting this service. The service hosts the data store for objects and interoperates with other **DCs** to ensure that a local change to an object replicates correctly across all **DCs**.

When **Active Directory** is operating as Active Directory Domain Services (AD DS), the **DC** contains full NC replicas of the configuration naming context (config NC), schema naming context (schema NC), and one of the domain NCs in its **forest**. If the AD DS **DC** is a global catalog server (GC server), it contains partial NC replicas of the remaining domain NCs in its **forest**. For more information, see [MS-AUTHSOD] section 1.1.1.5.2 and [MS-ADTS].

When **Active Directory** is operating as Active Directory Lightweight Directory Services (AD LDS), several AD LDS **DCs** can run on one server. When **Active Directory** is operating as AD DS, only one AD DS **DC** can run on one server. However, several AD LDS **DCs** can coexist with one AD DS **DC** on one server. The AD LDS **DC** contains full NC replicas of the config NC and the schema NC in its **forest**. The domain controller is the server side of Authentication Protocol Domain Support [MS-APDS].

domain local group: An **Active Directory** group that allows user objects, global groups, and universal groups from any **domain** as members. It can additionally include, and be a member of, other **domain local groups** from within its **domain**. A group object g is a **domain local group** if and only if GROUP_TYPE_RESOURCE_GROUP is present in g!groupType; see [MS-ADTS] section 2.2.12, "Group Type Flags". A security-enabled **domain local group** is valid for inclusion within access control lists (ACLs) from its own **domain**. If a **domain** is in **mixed mode**, then a security-enabled **domain local group** in that **domain** allows only user objects as members.

domain member (member machine): A machine that is joined to a domain by sharing a secret between the machine and the domain.

Tech Savvy Productions

domain name: A domain name or a **NetBIOS name** that identifies a **domain**.

Domain Name System (DNS): A hierarchical, distributed database that contains mappings of domain names to various types of data, such as IP addresses. DNS enables the location of computers and services by user-friendly names, and it also enables the discovery of other information stored in the database.

domain tree: A set of **domains** that are arranged hierarchically, typically following an accompanying DNS hierarchy, with trusts between parents and children. An example **domain tree** might be a.example.com, b.example.com, and example.com; domain A and domain B each trust example.com but do not trust each other directly. They will have a transitive trust relationship through example.com.

dynamic endpoint: A network-specific server address that is requested and assigned at run time. For more information, see [C706].

encryption key: One of the input parameters to an encryption algorithm. Generally speaking, an encryption algorithm takes as input a clear-text message and a key, and results in a cipher-text message. The corresponding **decryption** algorithm takes a cipher-text message, and the key, and results in the original clear-text message.

endpoint: A network-specific address of a remote procedure call (RPC) server process for remote procedure calls. The actual name and type of the endpoint depends on the **RPC** protocol sequence that is being used. For example, for RPC over TCP (RPC Protocol Sequence ncacn_ip_tcp), an endpoint might be TCP port 1025. For RPC over Server Message Block (RPC Protocol Sequence ncacn_np), an endpoint might be the name of a named pipe. For more information, see [C706].

enterprise network: The network of computer systems in an organization, such as a corporation. An enterprise can span geographical locations and often includes a variety of computer types, operating systems, protocols, and network architectures.

forest: One or more **domains** that share a common schema and trust each other transitively. An organization can have multiple **forests**. A **forest**

Tech Savvy Productions

establishes the security and administrative boundary for all the objects that reside within the **domains** that belong to the **forest**. In contrast, a **domain** establishes the administrative boundary for managing objects, such as users, groups, and computers. In addition, each **domain** has individual security policies and trust relationships with other **domains**.

forest trust: A type of **trust** where the trusted party is a **forest**, which means that all **domains** in that **forest** are trusted.

forest trust information: Information about namespaces, **domain names**, and **security identifiers (SIDs)** owned by a trusted **forest**.

full database synchronization: A mechanism for synchronizing an entire database record set on a particular replication partner.

fully qualified domain name (FQDN): In **Active Directory**, a fully qualified domain name (FQDN) that identifies a **domain**.

global catalog (GC): A unified partial view of multiple **naming contexts (NCs)** in a distributed partitioned directory. The **Active Directory** directory service **GC** is implemented by GC servers. The definition of **global catalog** is specified in [MS-ADTS] section 3.1.1.1.8.

globally unique identifier (GUID): A term used interchangeably with **universally unique identifier (UUID)** in Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the value. Specifically, the use of this term does not imply or require that the algorithms described in [RFC4122] or [C706] must be used for generating the **GUID**. See also **universally unique identifier (UUID)**.

group: A collection of objects that can be treated as a whole.

Hash-based Message Authentication Code (HMAC): A mechanism for message authentication using cryptographic hash functions. HMAC can be used with any iterative cryptographic hash function (for example, MD5 and SHA-1) in combination with a secret shared key. The cryptographic strength of HMAC depends on the properties of the underlying hash function.

Tech Savvy Productions

interactive logon: A software method in which the account information and credentials input by the user interactively are authenticated by a **server** or **domain controller (DC)**.

Interface Definition Language (IDL): The International Standards Organization (ISO) standard language for specifying the interface for remote procedure calls. For more information, see [C706] section 4.

Key Distribution Center (KDC): The Kerberos service that implements the authentication and ticket granting services specified in the Kerberos protocol. The service runs on computers selected by the administrator of the realm or domain; it is not present on every machine on the network. It must have access to an account database for the realm that it serves. **KDCs** are integrated into the **domain controller** role. It is a network service that supplies tickets to clients for use in authenticating to services.

key list request: A Kerberos protocol message used to request a list of key types the KDC can supply to the client to support single sign-on capabilities in legacy protocols.

Lightweight Directory Access Protocol (LDAP): The primary access protocol for **Active Directory**. Lightweight Directory Access Protocol (LDAP) is an industry-standard protocol, established by the Internet Engineering Task Force (IETF), which allows users to query and update information in a **directory service (DS)**, as described in [MS-ADTS]. The Lightweight Directory Access Protocol can be either version 2 [RFC1777] or version 3 [RFC3377].

Local Security Authority (LSA): A protected subsystem that authenticates and logs users onto the local system. **LSA** also maintains information about all aspects of local security on a system, collectively known as the local security policy of the system.

Local Security Authority (LSA) database: A Microsoft-specific terminology for the part of the user account database containing account privilege information (such as specific account rights) and domain security policy information.

Tech Savvy Productions

mailslot: A mechanism for one-way interprocess communications (IPC). For more information, see [MSLOT] and [MS-MAIL].

mixed mode: A state of an **Active Directory** **domain** that supports **domain controllers (DCs)** running Windows NT Server 4.0 operating system. Mixed mode does not allow organizations to take advantage of new **Active Directory** features such as universal groups, nested group membership, and interdomain group membership. See also native mode.

naming context (NC): An **NC** is a set of objects organized as a tree. It is referenced by a DSName. The DN of the DSName is the distinguishedName attribute of the tree root. The **GUID** of the DSName is the objectGUID attribute of the tree root. The **security identifier (SID)** of the DSName, if present, is the objectSid attribute of the tree root; for Active Directory Domain Services (AD DS), the **SID** is present if and only if the **NC** is a domain naming context (domain NC). **Active Directory** supports organizing several **NCs** into a tree structure.

NetBIOS name: A 16-byte address that is used to identify a NetBIOS resource on the network. For more information, see [RFC1001] and [RFC1002].

Netlogon: In a Windows NT operating system-compatible network security environment, the component responsible for synchronization and maintenance functions between a **primary domain controller (PDC)** and backup domain controllers (BDC). **Netlogon** is a precursor to the directory replication server (DRS) protocol.

network logon: A software method in which the account information and credentials previously supplied by the user as part of an interactive logon are used again to log the user onto another network resource.

nonce: A number that is used only once. This is typically implemented as a random number large enough that the probability of number reuse is extremely small. A nonce is used in authentication protocols to prevent replay attacks. For more information, see [RFC2617].

NT LAN Manager (NTLM): An **authentication** protocol that is based on a challenge-response sequence for **authentication**.

Tech Savvy Productions

one-way function (OWF): The calculation of a hash of the password using the Rivest-Shamir-Adleman (RSA) MD4 function. **OWF** is used to refer to the resulting value of the hash operation.

opnum: An operation number or numeric identifier that is used to identify a specific **remote procedure call (RPC)** method or a method in an interface. For more information, see [C706] section 12.5.2.12 or [MS-RPCE].

original equipment manufacturer (OEM) character set: A character encoding used where the mappings between characters is dependent upon the code page configured on the machine, typically by the manufacturer.

partial database synchronization: A mechanism for synchronizing a set of database records on a particular replication partner.

primary domain: A **domain** (identified by a **security identifier (SID)**) that the server is joined to. For a **domain controller (DC)**, the **primary domain** is that of the **domain** itself.

primary domain controller (PDC): A **domain controller (DC)** designated to track changes made to the accounts of all computers on a **domain**. It is the only computer to receive these changes directly, and is specialized so as to ensure consistency and to eliminate the potential for conflicting entries in the **Active Directory** database. A **domain** has only one **PDC**.

principal: An authenticated entity that initiates a message or channel in a distributed system.

privilege: The right of a user to perform system-related operations, such as debugging the system. A user's authorization context specifies what privileges are held by that user.

RC4: A variable key-length symmetric encryption algorithm. For more information, see [SCHNEIER] section 17.1.

read-only domain controller (RODC): A **domain controller (DC)** that does not accept originating updates. Additionally, an **RODC** does not perform outbound replication. An RODC cannot be the primary domain controller (PDC) for its domain.

Tech Savvy Productions

relative identifier (RID): The last item in the series of SubAuthority values in a **security identifier (SID)** [SIDD]. It distinguishes one account or group from all other accounts and groups in the domain. No two accounts or groups in any domain share the same RID.

remote procedure call (RPC): A communication protocol used primarily between client and server. The term has three definitions that are often used interchangeably: a runtime environment providing for communication facilities between computers (the RPC runtime); a set of request-and-response message exchanges between computers (the RPC exchange); and the single message from an RPC exchange (the RPC message). For more information, see [C706].

RPC protocol sequence: A character string that represents a valid combination of a **remote procedure call (RPC)** protocol, a network layer protocol, and a transport layer protocol, as described in [C706] and [MS-RPCE].

RPC transport: The underlying network services used by the remote procedure call (RPC) runtime for communications between network nodes. For more information, see [C706] section 2.

secret key: A symmetric encryption key shared by two entities, such as between a user and the **domain controller (DC)**, with a long lifetime. A password is a common example of a secret key. When used in a context that implies Kerberos only, a principal's secret key.

secure channel: An authenticated **remote procedure call (RPC)** connection between two machines in a **domain** with an established **security context** used for signing and encrypting **RPC** packets.

Security Account Manager (SAM): A centrally managed service, such as Active Directory Domain Services (AD DS), that enables a server to establish a trust relationship with other authorized servers. The SAM also maintains information about domains and security principals, and provides client-to-server information by using several available standards for access control lists (ACLs).

Tech Savvy Productions

security account manager (SAM) built-in database: The part of the user account database that contains account information (such as account names and passwords) for accounts and groups that are pre-created at the database installation.

security context: An abstract data structure that contains authorization information for a particular **security principal** in the form of a Token/Authorization Context (see [MS-DTYP] section 2.5.2). A server uses the authorization information in a **security context** to check access to requested resources. A **security context** also contains a key identifier that associates mutually established cryptographic keys, along with other information needed to perform secure communication with another security principal.

Security Descriptor Definition Language (SDDL): The format used to specify a security descriptor as a text string, specified in [MS-DTYP] section 2.5.1.

security identifier (SID): An identifier for **security principals** that is used to identify an account or a group. Conceptually, the **SID** is composed of an account authority portion (typically a **domain**) and a smaller integer representing an identity relative to the account authority, termed the **relative identifier (RID)**. The **SID** format is specified in [MS-DTYP] section 2.4.2; a string representation of **SIDs** is specified in [MS-DTYP] section 2.4.2 and [MS-AZOD] section 1.1.1.2.

security principal: A unique entity, also referred to as a principal, that can be authenticated by **Active Directory**. It frequently corresponds to a human user, but also can be a service that offers a resource to other security principals. Other security principals might be a group, which is a set of principals. Groups are supported by **Active Directory**.

security provider: A pluggable security module that is specified by the protocol layer above the **remote procedure call (RPC)** layer, and will cause the **RPC** layer to use this module to secure messages in a communication session with the server. The security provider is sometimes referred to as an **authentication** service. For more information, see [C706] and [MS-RPCE].

security support provider (SSP): A dynamic-link library (DLL) that implements the **Security Support Provider Interface (SSPI)** by making one or more

Tech Savvy Productions

security packages available to applications. Each security package provides mappings between an application's **SSPI** function calls and an actual security model's functions. Security packages support security protocols such as Kerberos authentication and NTLM.

Security Support Provider Interface (SSPI): An API that allows connected applications to call one of several security providers to establish authenticated connections and to exchange data securely over those connections. It is equivalent to Generic Security Services (GSS)-API, and the two are on-the-wire compatible.

server: A computer on which the **remote procedure call (RPC)** server is executing.

server challenge (SC): A 64-bit nonce generated on the server side.

service principal name (SPN): The name a client uses to identify a service for mutual authentication. For more information, see [MS-ADTS] section 2.2.21 (Service Principal Name) and [RFC1964] section 2.1.1.

session key: A relatively short-lived symmetric key (a cryptographic key negotiated by the client and the server based on a shared secret). A **session key's** lifespan is bounded by the session to which it is associated. A **session key** has to be strong enough to withstand cryptanalysis for the lifespan of the session.

shared secret: A piece of data that is known only to the **security principal** and an authenticating authority; for example, a user and a domain controller. It is used to prove the **principal's** identity. A password is a common example of a shared secret. Also called a "secret key".

site: A collection of one or more well-connected (reliable and fast) TCP/IP subnets. By defining **sites** (represented by site objects) an administrator can optimize both **Active Directory** access and **Active Directory** replication with respect to the physical network. When users log in, **Active Directory** clients find **domain controllers (DCs)** that are in the same **site** as the user, or near the same **site** if there is no **DC** in the **site**. See also Knowledge Consistency Checker (KCC). For more information, see [MS-ADTS].

Tech Savvy Productions

sub-authentication: Optional and additional **authentication** functionality, usually provided by extending an **authentication** algorithm.

sub-authentication package: An optional component that provides additional **authentication** functionality. If a **sub-authentication package** is installed, the **authentication** package calls the **sub-authentication package** before returning its **authentication** result. The request to verify by a **sub-authentication package** is indicated by the ParameterControl field of the LogonInformation parameter (see [MS-APDS] section 3.1.5.2.1, Verifying Responses with Sub-Authentication Packages).

Time-To-Live (TTL): The time duration for which a Server Object is available.

transitive trust: The state of two domains establishing trust through an intermediary domain. For example, if domain A trusts domain B, and domain B trusts domain C, then domain A can be configured to trust domain C through transitive trust.

trust: To accept another authority's statements for the purposes of **authentication** and authorization, especially in the case of a relationship between two domains. If **domain** A trusts **domain** B, **domain** A accepts **domain** B's **authentication** and authorization statements for **principals** represented by security principal objects in **domain** B; for example, the list of groups to which a particular user belongs. As a noun, a **trust** is the relationship between two **domains** described in the previous sentence.

trusted domain: A domain that is trusted to make authentication decisions for security principals in that domain.

trusted domain object (TDO): A collection of properties that define a trust relationship with another domain, such as direction (outbound, inbound, or both), trust attributes, name, and security identifier of the other domain. For more information, see [MS-ADTS].

Unicode: A character encoding standard developed by the Unicode Consortium that represents almost all of the written languages of the world. The **Unicode** standard [UNICODE5.0.0/2007] provides three forms (UTF-8, UTF-16, and

Tech Savvy Productions

UTF-32) and seven schemes (UTF-8, UTF-16, UTF-16 BE, UTF-16 LE, UTF-32, UTF-32 LE, and UTF-32 BE).

Unicode string: A **Unicode** 8-bit string is an ordered sequence of 8-bit units, a **Unicode** 16-bit string is an ordered sequence of 16-bit code units, and a **Unicode** 32-bit string is an ordered sequence of 32-bit code units. In some cases, it could be acceptable not to terminate with a terminating null character. Unless otherwise specified, all **Unicode strings** follow the UTF-16LE encoding scheme with no Byte Order Mark (BOM).

universally unique identifier (UUID): A 128-bit value. UUIDs can be used for multiple purposes, from tagging objects with an extremely short lifetime, to reliably identifying very persistent objects in cross-process communication such as client and server interfaces, manager entry-point vectors, and **RPC** objects. UUIDs are highly likely to be unique. UUIDs are also known as **globally unique identifiers (GUIDs)** and these terms are used interchangeably in the Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the UUID. Specifically, the use of this term does not imply or require that the algorithms described in [RFC4122] or [C706] must be used for generating the UUID.

user principal name (UPN): A user account name (sometimes referred to as the user logon name) and a domain name that identifies the domain in which the user account is located. This is the standard usage for logging on to a Windows domain. The format is: someone@example.com (in the form of an email address). In **Active Directory**, the userPrincipalName attribute of the account object, as described in [MS-ADTS].

Windows Time Service (W32Time): A service that supports time synchronization against network and hardware time sources. For more information, see [WTSREF] and [MS-SNTP].

writability: The abstract feature capability representing the ability of a **domain controller (DC)** to accept modifications and issue originating updates, with respect to a given **naming context (NC)** replica.

Tech Savvy Productions

Components and Capabilities

From an abstract point of view, the functionality provided by the **Active Directory** protocols can be represented by the components shown in the following diagram.

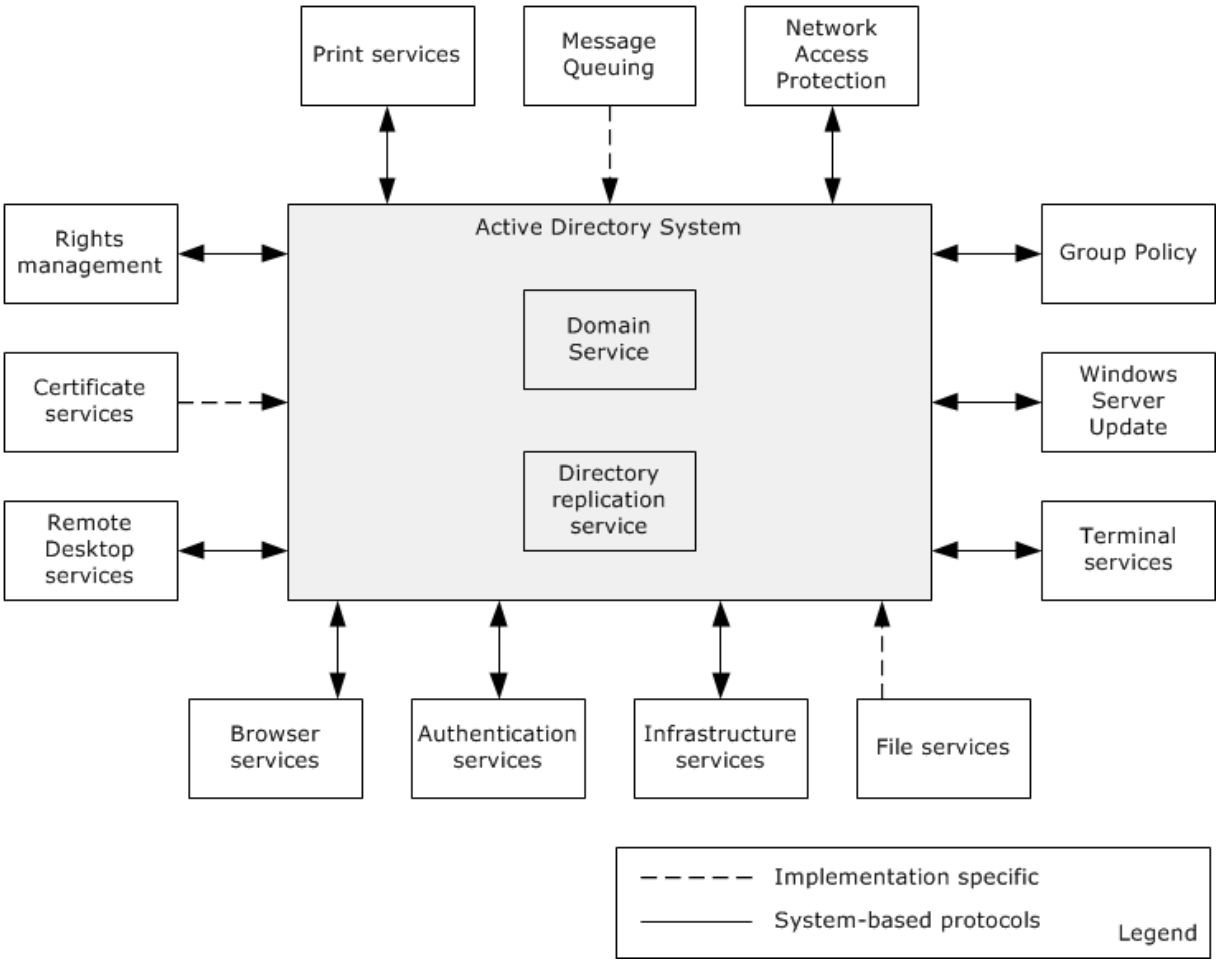


Figure 3: Active Directory system components

Tech Savvy Productions

Protocol Standard for a NetBIOS Service on a TCP/UDP Transport: Concepts and Methods, as specified in [RFC1001]. This standard is used for **NetBIOS** services.

Protocol Standard for a NetBIOS Service on a TCP/UDP Transport: Detailed Specifications, as specified in [RFC1002]. This standard is used for NetBIOS services.

10 Protocol Relationships

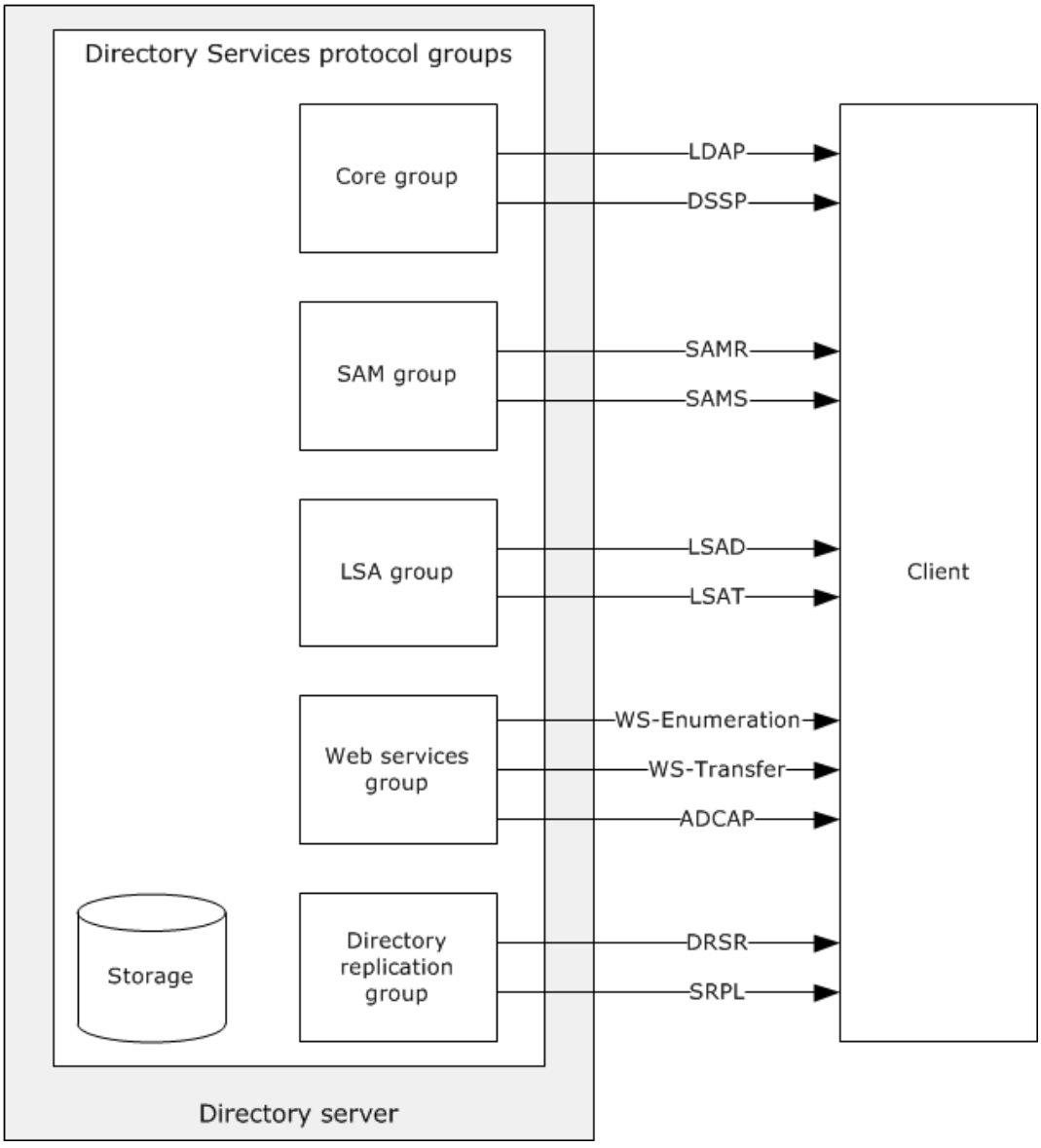


Figure 4: Active Directory protocol grouping

s shown in the preceding diagram, the member protocols that make up the Active Directory Services Protocol Groups can be divided into five functional groups. Each group accomplishes an interrelated set of tasks. A client typically uses protocols in the same group in conjunction with each other. The groups are as follows:

- The core group contains protocols that are supported by all **directory servers** in the Active Directory system, whether they run in **AD DS** or **AD LDS** mode. This group includes **LDAP**, which is the primary protocol that is used to read and write objects in the **directory tree**. The [MS-DSSP] (DSSP) protocol does not perform operations against the directory tree, but is included in this group because it is also present on all directory servers in the Active Directory Services Protocols.
- The SAM group includes SAMR and SAMS. SAMR is used to perform **account** maintenance and operates on the same directory tree as the core group of protocols, but it provides access to only a subset of the objects in that tree, and further, provides access to only a subset of the **attributes** on those objects. SAMR is supported only when operating in AD DS mode. SAMS is used to perform account maintenance and time-critical database changes between Active Directory servers that are in the same **domain**.
- The LSA group contains the LSAD and LSAT protocols. Both protocols are serviced by the same **RPC** interface and **endpoint** ([MS-LSAD] section 1.8, Vendor-Extensible Fields, and [MS-LSAT] section 1.8, Vendor-Extensible Fields).
- The Web Services group consists of the WS-Transfer, WS-Enumeration, and **ADCAP** protocols along with the [MS-WSDS] (WSDS), and [MS-WSPELD] (WSPELD) protocol extensions. This protocol group is only supported on some versions of the Active Directory Services. Much like the core group, these protocols permit clients to read and write **directory objects** in the directory tree, and to perform selected tasks against the tree. Unlike the core group, the protocols in this group are based on **SOAP** rather than remote procedure call (RPC) or block-structured transports.
- The Directory Replication group contains the DRSR and SRPL protocols. DRSR is an RPC protocol that is used for management of replication and management of data in Active Directory. SRPL is the extension to the DRS protocol for transport over the Simple Mail Transfer Protocol (SMTP).

Tech Savvy Productions

The following diagram shows the relationship among the protocols of the Active Directory system.

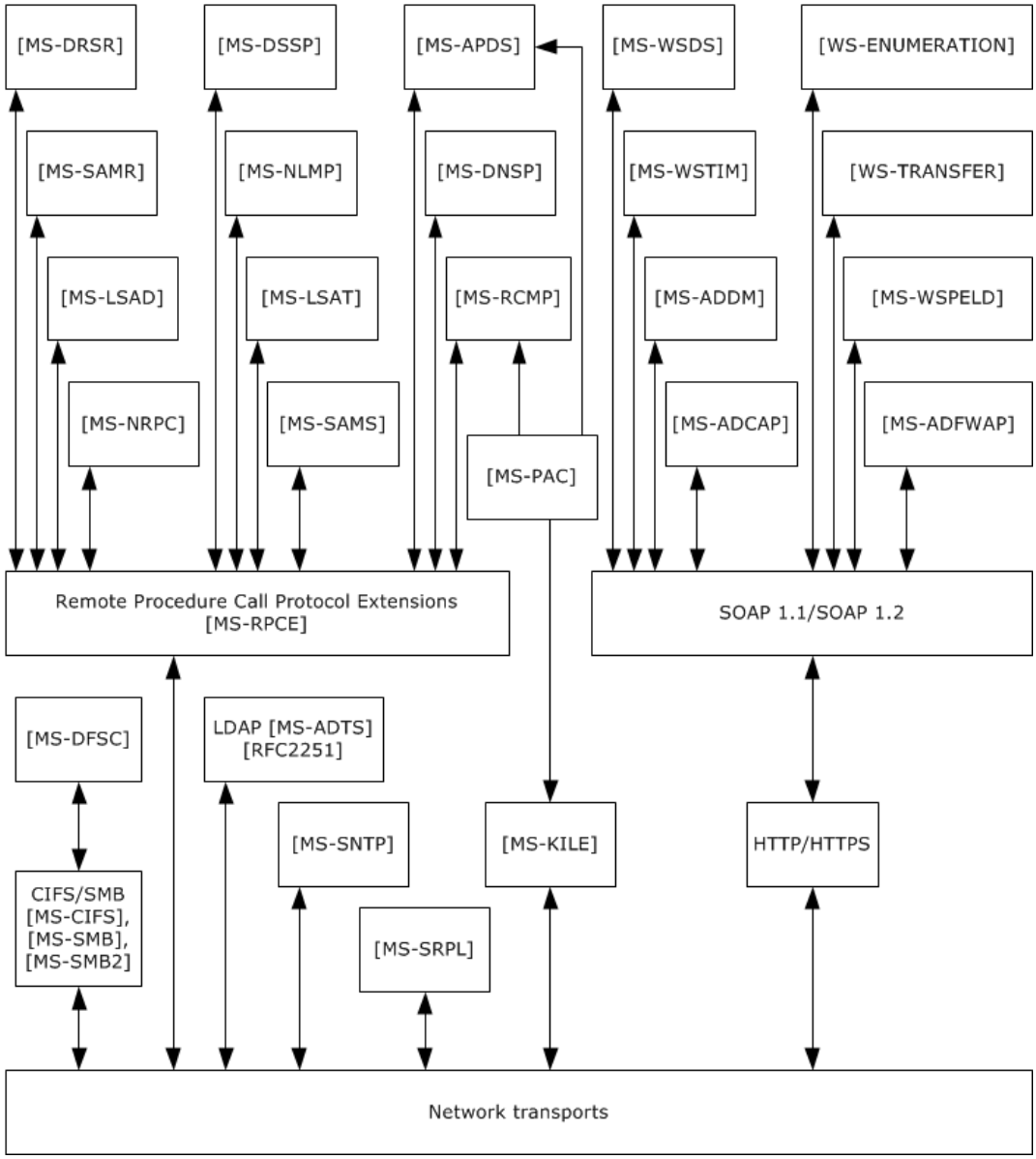


Figure 5: Protocol relationships

11 Protocol Summary

The following tables provide a comprehensive list of the member protocols of the Active Directory system. Section 2.8 provides details about which protocols or protocol subsets are supported in the different modes of operation.

Tech Savvy Productions

The protocols in the following table enable the core functionality of the Active Directory system, including access to the directory tree, replication, name translation, determination of **group** membership, and **domain controller** status. These protocols are supported by all **directory servers** in the Active Directory system, whether running in **Active Directory Domain Services (AD DS)** or **Active Directory Lightweight Directory Services (AD LDS)** mode.

Protocol name	Description	Short name
Active Directory extensions for **Lightweight Directory Access Protocol (LDAP)**, versions 2 and 3	Active Directory is a server for LDAP. [MS-ADTS] section 3.1.1.3 specifies the extensions and variations of LDAP that are supported by Active Directory. **Note** In a reference to LDAP without a version number, LDAP refers to both versions 2 and 3.	[MS-ADTS] section 3.1.1.3.1
Directory Replication Service Remote Protocol (drsuapi) - Replication	The Directory Replication Service (DRS) Remote Protocol. This protocol includes the drsuapi and dsaop **RPC** interfaces. Methods on these interfaces provide replication of directory information among the domain controllers of an AD DS **domain**. Methods on these interfaces also provide a variety of functionality to clients, such as converting names between formats and retrieving information about AD DS domain controllers. This protocol also supports DC cloning operations.<2>	[MS-DRSR]
SMTP Replication Protocol Extensions	The Directory Replication Service (DRS) Protocol Extensions for SMTP. This protocol provides Simple Mail Transfer Protocol (SMTP) transport of replication information as an alternative to RPC.	[MS-SRPL]

Tech Savvy Productions

Protocol name	Description	Short name
Directory Services Setup Remote Protocol	The Directory Services Setup Remote Protocol, as defined in [MS-DSSP]. This protocol can be used to retrieve information about the state of a computer in a domain or a non-domain workgroup.	[MS-DSSP]

The protocols in the following table enable account maintenance when the Active Directory system is operating in AD DS mode. This includes the creation, modification, retrieval, and deletion of users and groups.

Protocol name	Description	Short name
Security Account Manager (SAM) Remote Protocol (Client-to-Server)	The Security Account Manager (SAM) Remote Protocol. Clients can use this protocol to perform account maintenance, for example, to create and delete accounts. The capabilities of this protocol are a subset of the capabilities of LDAP.	[MS-SAMR]
Security Account Manager (SAM) Remote Protocol (Server-to-Server)	The Security Account Manager (SAM) Remote Protocol. Domain controllers (DCs) use this protocol to forward time-critical database changes to the **primary domain controller (PDC)**, and to forward time-critical database changes from a **read-only domain controller (RODC)** to a **writable NC replica** within the same domain outside the normal replication protocol. This protocol is used only between Active Directory servers in the same domain.	[MS-SAMS]

The protocols in the following table allow **clients** to retrieve security **policy** information and translate **security identifiers (SIDs)** that **identity security principals**, such as users, to human-readable names.

Protocol name	Description	Short name
Local Security Authority (Domain Policy) Remote Protocol	The Local Security Authority (Domain Policy) Remote Protocol. Clients can use this protocol to retrieve security policy information.	[MS-LSAD]

Tech Savvy Productions

Protocol name	Description	Short name
Local Security Authority (Translation Methods) Remote Protocol	The Local Security Authority (Translation Methods) Remote Protocol. Clients can use this protocol to translate security identifiers (SIDs) of security principals to human-readable names, and vice versa.	[MS-LSAT]

The protocols in the following table enable Web services for the Active Directory system that allow access to the directory tree and the management of Active Directory account information and topologies.

Protocol name	Description	Short name
Active Directory Web Services Custom Action Protocol	The Active Directory Web Services Custom Action Protocol. It is a **SOAP**-based Web Services protocol for managing account and topology information.	[MS-ADCAP]
WS-Transfer: Identity Management Operations for Directory Access Extensions	WS-Transfer: Identity Management Operations for Directory Access Extensions. This is a set of protocol extensions to WS-Transfer that allows **directory objects** to be manipulated at a finer level of granularity than unextended WS-Transfer.	[MS-WSTIM]
WS-Enumeration: Web Services Enumeration	The WS-Enumeration protocol. This protocol allows directory objects to be queried by using a SOAP-based Web Services protocol.	[WSENUM]
WS-Transfer: Web Services Transfer	The WS-Transfer protocol. This protocol allows directory objects to be created, removed, modified, and read by using a SOAP-based Web Services protocol.	[WXFR]

Tech Savvy Productions

Protocol name	Description	Short name
WS-Enumeration: Directory Services Protocol Extensions	The WS-Enumeration Directory Services Protocol Extensions. This is a set of protocol extensions to WS-Enumeration that, among other things, allows a client to request that query results be sorted. It also specifies a query language that is used by clients to specify which directory objects are to be returned from the query.	[MS-WSDS]
WS-Transfer and WS-Enumeration Protocol Extension for Lightweight Directory Access Protocol v3 Controls	WS-Transfer and WS-Enumeration Protocol Extension for Lightweight Directory Access Protocol v3 Controls. This is a protocol extension to WS-Transfer and WS-Enumeration. It permits LDAP **extended controls** to be attached to operations in the protocols that it extends.	[MS-WSPELD]
Active Directory Web Services: Data Model and Common Elements	The Active Directory Web Services: Data Model and Common Elements. Although not a protocol itself, this defines an **XML** data model that is shared by the other Web Service protocols and protocol extensions, as well as common protocol elements referenced by the other documents.	[MS-ADDM]

12 Environment

Because **domain** interactions are distributed among many computers for different, but related purposes, enumerating the dependencies of the **Active Directory** protocols is complex. The following rough diagram serves as a very high-level illustration of how the dependencies among components can be visualized.

Tech Savvy Productions

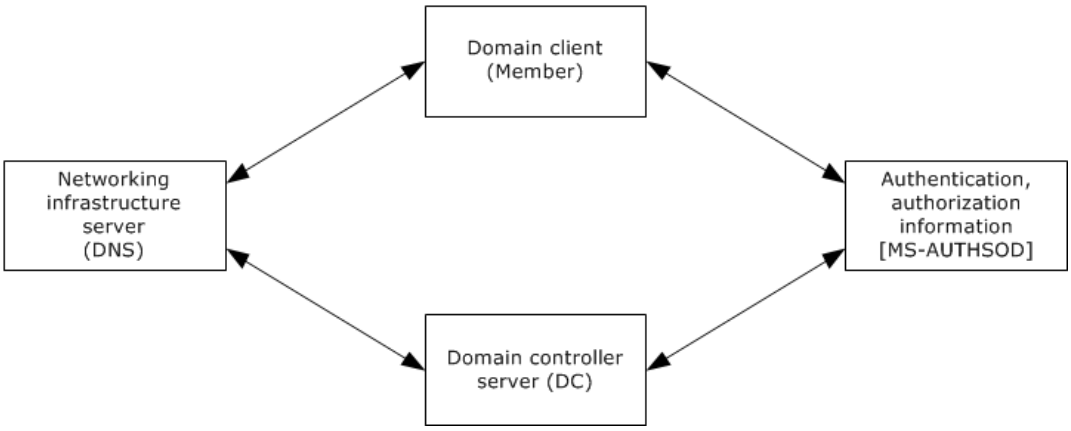


Figure 6: Dependencies among domain components

In this diagram, the dependencies of the system are symmetrical between the **domain client** and the **domain controller server**. Both the domain client and the domain controller server rely upon infrastructure **servers**, such as **DNS**, and leverage those servers for locating each other (rendezvous). During this rendezvous process, the domain controller server publishes its name and the domain client locates the domain controller server through DNS. The details of this rendezvous process are described in section 2.7.7.3.1.

In addition to service location, the rendezvous process between a domain client and a domain controller server relies upon authentication and authorization information. The domain controller server, for example, leverages the authorization information that it contains for controlling access to its resources. For more detailed information about authentication and authorization, see [MS-AUTHSOD] and related documents.

Tech Savvy Productions

Lightweight Directory Access Protocol

From Wikipedia, the free encyclopedia

Lightweight Directory Access Protocol	
Communication protocol	
Purpose	Directory service
Based on	X.500
OSI layer	Application layer
Port(s)	389 (ldap), 636 (ldaps)
RFC(s)	RFC 4510, RFC 4511

The **Lightweight Directory Access Protocol** (**LDAP** /ˈɛldæp/) is an open, vendor-neutral, industry standard application protocol for accessing and maintaining distributed directory information services over an Internet Protocol (IP) network.[1] Directory services play an important role in developing intranet and Internet applications by allowing the sharing of information about users, systems, networks, services, and applications throughout the network.[2] As examples, directory services may provide any organized set of records, often with a hierarchical structure, such as a corporate email directory. Similarly, a telephone directory is a list of subscribers with an address and a phone number.

LDAP is specified in a series of Internet Engineering Task Force (IETF) Standard Track publications called Request for Comments (RFCs), using the description language ASN.1. The latest specification is Version 3, published as RFC 4511[3] (a road map to the technical specifications is provided by RFC4510).

Tech Savvy Productions

A common use of LDAP is to provide a central place to store usernames and passwords. This allows many different applications and services to connect to the LDAP server to validate users.[4]

LDAP is based on a simpler subset of the standards contained within the X.500 standard. Because of this relationship, LDAP is sometimes called X.500-lite.[5]

Contents

- 1History
- 2Protocol overview
- 3Directory structure
- 4Operations
    - 4.1Add
    - 4.2Bind (authenticate)
    - 4.3Delete
    - 4.4Search and compare
    - 4.5Modify
    - 4.6Modify DN
    - 4.7Extended operations
        - 4.7.1StartTLS
    - 4.8Abandon
    - 4.9Unbind
- 5URI scheme
- 6Schema
- 7Variations
- 8Other data models
- 9Usage
- 10See also
- 11References
    - 11.1Sources
- 12Further reading
- 13External links

Tech Savvy Productions

History

Telecommunication companies' understanding of directory requirements were well developed after some 70 years of producing and managing telephone directories. These companies introduced the concept of directory services to information technology and computer networking, their input culminating in the comprehensive X.500 specification,[6] a suite of protocols produced by the International Telecommunication Union (ITU) in the 1980s.

X.500 directory services were traditionally accessed via the X.500 Directory Access Protocol (DAP), which required the Open Systems Interconnection (OSI) protocol stack. LDAP was originally intended to be a lightweight alternative protocol for accessing X.500 directory services through the simpler (and now widespread) TCP/IP protocol stack. This model of directory access was borrowed from the DIXIE and Directory Assistance Service protocols.

The protocol was originally created[7] by Tim Howes of the University of Michigan, Steve Kille of Isode Limited, Colin Robbins of Nexor and Wengyik Yeong of Performance Systems International, circa 1993, as a successor[8] to DIXIE and DAS. Mark Wahl of Critical Angle Inc., Tim Howes, and Steve Kille started work in 1996 on a new version of LDAP, LDAPv3, under the aegis of the Internet Engineering Task Force (IETF). LDAPv3, first published in 1997, superseded LDAPv2 and added support for extensibility, integrated the Simple Authentication and Security Layer, and better aligned the protocol to the 1993 edition of X.500. Further development of the LDAPv3 specifications themselves and of numerous extensions adding features to LDAPv3 has come through the IETF.

In the early engineering stages of LDAP, it was known as *Lightweight Directory Browsing Protocol*, or *LDBP*. It was renamed with the expansion of the scope of the protocol beyond directory browsing and searching, to include directory update functions. It was given its *Lightweight* name because it was not as network intensive as its DAP predecessor and thus was more easily implemented over the Internet due to its relatively modest bandwidth usage.

LDAP has influenced subsequent Internet protocols, including later versions of X.500, XML Enabled Directory (XED), Directory Service Markup Language (DSML), Service Provisioning Markup Language (SPML), and the Service Location Protocol (SLP). It is also used as the basis for Microsoft's Active Directory.

Tech Savvy Productions

Protocol overview[edit]

A client starts an LDAP session by connecting to an LDAP server, called a Directory System Agent (DSA), by default on TCP and UDP port 389, or on port 636 for LDAPS (LDAP over TLS/SSL, see below).[9] The client then sends an operation request to the server, and a server sends responses in return. With some exceptions, the client does not need to wait for a response before sending the next request, and the server may send the responses in any order. All information is transmitted using Basic Encoding Rules (BER).

The client may request the following operations:

- StartTLS – use the LDAPv3 Transport Layer Security (TLS) extension for a secure connection
- Bind – authenticate and specify LDAP protocol version
- Search – search for and/or retrieve directory entries
- Compare – test if a named entry contains a given attribute value
- Add a new entry
- Delete an entry
- Modify an entry
- Modify Distinguished Name (DN) – move or rename an entry
- Abandon – abort a previous request
- Extended Operation – generic operation used to define other operations
- Unbind – close the connection (not the inverse of Bind)

In addition the server may send "Unsolicited Notifications" that are not responses to any request, e.g. before the connection is timed out.

A common alternative method of securing LDAP communication is using an SSL tunnel. The default port for LDAP over SSL is 636. The use of LDAP over SSL was common in LDAP Version 2 (LDAPv2) but it was never standardized in any formal specification. This usage has been deprecated along with LDAPv2, which was officially retired in 2003.[10]

Directory structure

The protocol provides an interface with directories that follow the 1993 edition of the X.500 model:

- An entry consists of a set of attributes.

Tech Savvy Productions

- An attribute has a name (an *attribute type* or *attribute description*) and one or more values. The attributes are defined in a *schema* (see below).
- Each entry has a unique identifier: its *Distinguished Name* (DN). This consists of its *Relative Distinguished Name* (RDN), constructed from some attribute(s) in the entry, followed by the parent entry's DN. Think of the DN as the full file path and the RDN as its relative filename in its parent folder (e.g. if `/foo/bar/myfile.txt` were the DN, then `myfile.txt` would be the RDN).

A DN may change over the lifetime of the entry, for instance, when entries are moved within a tree. To reliably and unambiguously identify entries, a UUID might be provided in the set of the entry's *operational attributes*.

An entry can look like this when represented in LDAP Data Interchange Format (LDIF), a plain text format (as opposed a binary protocol such as LDAP itself):

```
dn: cn=John Doe,dc=example,dc=com
cn: John Doe
givenName: John
sn: Doe
telephoneNumber: +1 888 555 6789
telephoneNumber: +1 888 555 1232
mail: john@example.com
manager: cn=Barbara Doe,dc=example,dc=com
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: top
```

"`dn`" is the distinguished name of the entry; it is neither an attribute nor a part of the entry. "`cn=John Doe`" is the entry's RDN (Relative Distinguished Name), and "`dc=example,dc=com`" is the DN of the parent entry, where "`dc`" denotes 'Domain Component'. The other lines show the attributes in the entry. Attribute names are typically mnemonic strings, like "`cn`" for common name, "`dc`" for domain component, "`mail`" for e-mail address, and "`sn`" for surname.

A server holds a subtree starting from a specific entry, e.g. "`dc=example,dc=com`" and its children. Servers may also hold references to other servers, so an attempt

Tech Savvy Productions

to access "ou=department,dc=example,dc=com" could return a *referral* or *continuation reference* to a server that holds that part of the directory tree. The client can then contact the other server. Some servers also support *chaining*, which means the server contacts the other server and returns the results to the client.

LDAP rarely defines any ordering: The server may return the values of an attribute, the attributes in an entry, and the entries found by a search operation in any order. This follows from the formal definitions - an entry is defined as a set of attributes, and an attribute is a set of values, and sets need not be ordered.

Operations

Add

The ADD operation inserts a new entry into the directory-server database.[11] If the distinguished name in the add request already exists in the directory, then the server will not add a duplicate entry but will set the result code in the add result to decimal 68, "entryAlreadyExists".[12]

- LDAP-compliant servers will never dereference the distinguished name transmitted in the add request when attempting to locate the entry, that is, distinguished names are never de-aliased.
- LDAP-compliant servers will ensure that the distinguished name and all attributes conform to naming standards.
- The entry to be added must not exist, and the immediate superior must exist.

```
dn: uid=user,ou=people,dc=example,dc=com
changetype: add
objectClass:top
objectClass:person
uid: user
sn: last-name
cn: common-name
userPassword: password
```

In the above example, uid=user,ou=people,dc=example,dc=com must not exist, and ou=people,dc=example,dc=com must exist.

Tech Savvy Productions

Bind (authenticate)

When an LDAP session is created, that is, when an LDAP client connects to the server, the **authentication state** of the session is set to anonymous. The BIND operation establishes the authentication state for a session.

Simple BIND and SASL PLAIN can send the user's DN and password in plaintext, so the connections utilizing either Simple or SASL PLAIN should be encrypted using Transport Layer Security (TLS). The server typically checks the password against the userPassword attribute in the named entry. Anonymous BIND (with empty DN and password) resets the connection to anonymous state.

SASL (Simple Authentication and Security Layer) BIND provides authentication services through a wide range of mechanisms, e.g. Kerberos or the client certificate sent with TLS.[13]

BIND also sets the LDAP protocol version by sending a version number in the form of an integer. If the client requests a version that the server does not support, the server must set the result code in the BIND response to the code for a protocol error. Normally clients should use LDAPv3, which is the default in the protocol but not always in LDAP libraries.

BIND had to be the first operation in a session in LDAPv2, but is not required as of LDAPv3. In LDAPv3, each successful BIND request changes the authentication state of the session and each unsuccessful BIND request resets the authentication state of the session.

Delete

To delete an entry, an LDAP client transmits a properly formed delete request to the server.[14]

- A delete request must contain the distinguished name of the entry to be deleted
- Request controls may also be attached to the delete request
- Servers do not dereference aliases when processing a delete request
- Only leaf entries (entries with no subordinates) may be deleted by a delete request. Some servers support an operational attribute hasSubordinates whose value indicates whether an entry has any subordinate entries, and some servers support an operational attribute numSubordinates[15] indicating the number of entries subordinate to the entry containing the numSubordinates attribute.

Tech Savvy Productions

- Some servers support the subtree delete request control permitting deletion of the DN and all objects subordinate to the DN, subject to access controls. Delete requests are subject to access controls, that is, whether a connection with a given authentication state will be permitted to delete a given entry is governed by server-specific access control mechanisms.

Search and compare

The Search operation is used to both search for and read entries. Its parameters are:

baseObject

The name of the base object entry (or possibly the root) relative to which the search is to be performed.

scope

What elements below the baseObject to search. This can be `BaseObject` (search just the named entry, typically used to read one entry), `singleLevel` (entries immediately below the base DN), or `wholeSubtree` (the entire subtree starting at the base DN).

filter

Criteria to use in selecting elements within scope. For example, the filter `(&(objectClass=person)(|(givenName=John)(mail=john*)))` will select "persons" (elements of objectClass `person`) where the matching rules for `givenName` and `mail` determine whether the values for those attributes match the filter assertion. Note that a common misconception is that LDAP data is case-sensitive, whereas in fact matching rules and ordering rules determine matching, comparisons, and relative value relationships. If the example filters were required to match the case of the attribute value, an *extensible match filter* must be used, for example, `(&(objectClass=person)(|(givenName:caseExactMatch:=John)(mail:caseExactSubstringsMatch:=john*)))`

derefAliases

Whether and how to follow alias entries (entries that refer to other entries),

attributes

Which attributes to return in result entries.

sizeLimit, timeLimit

Tech Savvy Productions

Maximum number of entries to return, and maximum time to allow search to run. These values, however, cannot override any restrictions the server places on size limit and time limit.

typesOnly

Return attribute types only, not attribute values.

The server returns the matching entries and potentially continuation references. These may be returned in any order. The final result will include the result code.

The Compare operation takes a DN, an attribute name and an attribute value, and checks if the named entry contains that attribute with that value.

Modify[edit]

The MODIFY operation is used by LDAP clients to request that the LDAP server make changes to existing entries.[16] Attempts to modify entries that do not exist will fail. MODIFY requests are subject to access controls as implemented by the server.

The MODIFY operation requires that the distinguished name (DN) of the entry be specified, and a sequence of changes. Each change in the sequence must be one of:

- add (add a new value, which must not already exist in the attribute)
- delete (delete an existing value)
- replace (replace an existing value with a new value)

LDIF example of adding a value to an attribute:

```
dn: dc=example,dc=com
changetype: modify
add: cn
cn: the-new-cn-value-to-be-added
-
```

To replace the value of an existing attribute, use the `replace` keyword. If the attribute is multi-valued, the client must specify the value of the attribute to update.

Tech Savvy Productions

To delete an attribute from an entry, use the keyword `delete` and the changetype designator `modify`. If the attribute is multi-valued, the client must specify the value of the attribute to delete.

There is also a Modify-Increment extension[17] which allows an incrementable attribute value to be incremented by a specified amount. The following example using LDIF increments `employeeNumber` by `5`:

```
dn: uid=user.0,ou=people,dc=example,dc=com
changetype: modify
increment: employeeNumber
employeeNumber: 5
-
```

When LDAP servers are in a replicated topology, LDAP clients should consider using the post-read control to verify updates instead of a search after an update.[18] The post-read control is designed so that applications need not issue a search request after an update – it is bad form to retrieve an entry for the sole purpose of checking that an update worked because of the replication eventual consistency model. An LDAP client should not assume that it connects to the same directory server for each request because architects may have placed load-balancers or LDAP proxies or both between LDAP clients and servers.

Modify DN[edit]

Modify DN (move/rename entry) takes the new RDN (Relative Distinguished Name), optionally the new parent's DN, and a flag that indicates whether to delete the value(s) in the entry that match the old RDN. The server may support renaming of entire directory subtrees.

An update operation is atomic: Other operations will see either the new entry or the old one. On the other hand, LDAP does not define transactions of multiple operations: If you read an entry and then modify it, another client may have updated the entry in the meantime. Servers may implement extensions[19] that support this, though.

Extended operations[edit]

The Extended Operation is a generic LDAP operation that can define new operations that were not part of the original protocol specification. StartTLS is one

Tech Savvy Productions

of the most significant extensions. Other examples include Cancel and Password Modify.

StartTLS[edit]

The StartTLS operation establishes Transport Layer Security (the descendant of SSL) on the connection. It can provide data confidentiality (to protect data from being observed by third parties) and/or data integrity protection (which protects the data from tampering). During TLS negotiation the server sends its X.509 certificate to prove its identity. The client may also send a certificate to prove its identity. After doing so, the client may then use SASL/EXTERNAL. By using the SASL/EXTERNAL, the client requests the server derive its identity from credentials provided at a lower level (such as TLS). Though technically the server may use any identity information established at any lower level, typically the server will use the identity information established by TLS.

Servers also often support the non-standard "LDAPS" ("Secure LDAP", commonly known as "LDAP over SSL") protocol on a separate port, by default 636. LDAPS differs from LDAP in two ways: 1) upon connect, the client and server establish TLS before any LDAP messages are transferred (without a StartTLS operation) and 2) the LDAPS connection must be closed upon TLS closure.

Some "LDAPS" client libraries only encrypt communication; they do not check the host name against the name in the supplied certificate.[20]

Abandon[edit]

The Abandon operation requests that the server abort an operation named by a message ID. The server need not honor the request. Neither Abandon nor a successfully abandoned operation send a response. A similar Cancel extended operation does send responses, but not all implementations support this.

Unbind[edit]

The Unbind operation abandons any outstanding operations and closes the connection. It has no response. The name is of historical origin, and is *not* the opposite of the Bind operation.[21]

Clients can abort a session by simply closing the connection, but they should use Unbind.[22] Unbind allows the server to gracefully close the connection and free resources that it would otherwise keep for some time until discovering the client had abandoned the connection. It also instructs the server to cancel operations

Tech Savvy Productions

that can be canceled, and to not send responses for operations that cannot be canceled.[23]

URI scheme[edit]

An LDAP uniform resource identifier (URI) scheme exists, which clients support in varying degrees, and servers return in referrals and continuation references (see RFC 4516):

```
ldap://host:port/DN?attributes?scope?filter?extensions
```

Most of the components described below are optional.

- *host* is the FQDN or IP address of the LDAP server to search.
- *port* is the network port (default port 389) of the LDAP server.
- *DN* is the distinguished name to use as the search base.
- *attributes* is a comma-separated list of attributes to retrieve.
- *scope* specifies the search scope and can be "base" (the default), "one" or "sub".
- *filter* is a search filter. For example, `(objectClass=*)` as defined in RFC 4515.
- *extensions* are extensions to the LDAP URL format.

For example, "`ldap://ldap.example.com/cn=John%20Doe,dc=example,dc=com`" refers to all user attributes in John Doe's entry in `ldap.example.com`, while "`ldap:///dc=example,dc=com??sub?(givenName=John)`" searches for the entry in the default server (note the triple slash, omitting the host, and the double question mark, omitting the attributes). As in other URLs, special characters must be percent-encoded.

There is a similar non-standard `ldaps` URI scheme for LDAP over SSL. This should not be confused with LDAP with TLS, which is achieved using the StartTLS operation using the standard `ldap` scheme.

Schema

The contents of the entries in a subtree are governed by a directory schema, a set of definitions and constraints concerning the structure of the directory information tree (DIT).

Tech Savvy Productions

The schema of a Directory Server defines a set of rules that govern the kinds of information that the server can hold. It has a number of elements, including:

- Attribute Syntaxes—Provide information about the kind of information that can be stored in an attribute.
- Matching Rules—Provide information about how to make comparisons against attribute values.
- Matching Rule Uses—Indicate which attribute types may be used in conjunction with a particular matching rule.
- Attribute Types—Define an object identifier (OID) and a set of names that may refer to a given attribute, and associates that attribute with a syntax and set of matching rules.
- Object Classes—Define named collections of attributes and classify them into sets of required and optional attributes.
- Name Forms—Define rules for the set of attributes that should be included in the RDN for an entry.
- Content Rules—Define additional constraints about the object classes and attributes that may be used in conjunction with an entry.
- Structure Rule—Define rules that govern the kinds of subordinate entries that a given entry may have.

Attributes are the elements responsible for storing information in a directory, and the schema defines the rules for which attributes may be used in an entry, the kinds of values that those attributes may have, and how clients may interact with those values.

Clients may learn about the schema elements that the server supports by retrieving an appropriate subschema subentry.

The schema defines *object classes*. Each entry must have an objectClass attribute, containing named classes defined in the schema. The schema definition of the classes of an entry defines what kind of object the entry may represent - e.g. a person, organization or domain. The object class definitions also define the list of attributes that must contain values and the list of attributes which may contain values.

For example, an entry representing a person might belong to the classes "top" and "person". Membership in the "person" class would require the entry to contain the "sn" and "cn" attributes, and allow the entry also to contain "userPassword", "telephoneNumber", and other attributes. Since entries may have multiple

Tech Savvy Productions

ObjectClasses values, each entry has a complex of optional and mandatory attribute sets formed from the union of the object classes it represents. ObjectClasses can be inherited, and a single entry can have multiple ObjectClasses values that define the available and required attributes of the entry itself. A parallel to the schema of an objectClass is a class definition and an instance in Object-oriented programming, representing LDAP objectClass and LDAP entry, respectively.

Directory servers may publish the directory schema controlling an entry at a base DN given by the entry's subschemaSubentry operational attribute. (An *operational attribute* describes operation of the directory rather than user information and is only returned from a search when it is explicitly requested.)

Server administrators can add additional schema entries in addition to the provided schema elements. A schema for representing individual people within organizations is termed a white pages schema.

Variations

A lot of the server operation is left to the implementor or administrator to decide. Accordingly, servers may be set up to support a wide variety of scenarios.

For example, data storage in the server is not specified - the server may use flat files, databases, or just be a gateway to some other server. Access control is not standardized, though there has been work on it and there are commonly used models. Users' passwords may be stored in their entries or elsewhere. The server may refuse to perform operations when it wishes, and impose various limits.

Most parts of LDAP are extensible. Examples: One can define new operations. *Controls* may modify requests and responses, e.g. to request sorted search results. New search scopes and Bind methods can be defined. Attributes can have *options* that may modify their semantics.

Other data models

As LDAP has gained momentum, vendors have provided it as an access protocol to other services. The implementation then recasts the data to mimic the LDAP/X.500 model, but how closely this model is followed varies. For example, there is software to access SQL databases through LDAP, even though LDAP does not readily lend itself to this.[24] X.500 servers may support LDAP as well.

Tech Savvy Productions

Similarly, data previously held in other types of data stores are sometimes moved to LDAP directories. For example, Unix user and group information can be stored in LDAP and accessed via PAM and NSS modules. LDAP is often used by other services for authentication and/or authorization (what actions a given already-authenticated user can do on what service). For example in Active Directory Kerberos is used in the authentication step, while LDAP is used in the authorization step.

An example of such data model is the GLUE Schema,[25] which is used in a distributed information system based on LDAP that enable users, applications and services to discover which services exist in a Grid infrastructure and further information about their structure and state.

Usage

An LDAP server may return referrals to other servers for requests that it cannot fulfill itself. This requires a naming structure for LDAP entries so one can find a server holding a given distinguished name (DN), a concept defined in the X.500 Directory and also used in LDAP. Another way of locating LDAP servers for an organization is a DNS server record (SRV).

An organization with the domain example.org may use the top level LDAP DN `dc=example, dc=org` (where *dc* means domain component). If the LDAP server is also named ldap.example.org, the organization's top level LDAP URL becomes `ldap://ldap.example.org/dc=example,dc=org`.

Primarily two common styles of naming are used in both X.500 [2008] and LDAPv3. These are documented in the ITU specifications and IETF RFCs. The original form takes the top level object as the country object, such as `c=US`, `c=FR`. The domain component model uses the model described above. An example of country based naming could be `l=Locality, ou=Some Organizational Unit, o=Some Organization, c=FR`, or in the US: `cn=Common Name, l=Locality, ou=Some Organizational Unit, o=Some Organization, st=CA, c=US`.

See also

- Ambiguous name resolution
- CCSO Nameserver
- Federated Naming Service
- Hesiod (name service)

Tech Savvy Productions

- Hierarchical database model
- Key server (cryptographic)
- LDAP Application Program Interface
- List of LDAP software
- Simple Authentication and Security Layer (SASL)

References

1. ^ *"Network Working Group RFC 4511"*

   . IETF.org. 2006-06-01. Retrieved 2014-04-04.

2. ^ *"Directory Services LDAP"*

   . Oracle.com. Retrieved 2014-04-04.

3. ^ What is LDAP?

   . Gracion.com. Retrieved on 2013-07-17.

4. ^ *"Introduction to OpenLDAP Directory Services"*

   . OpenLDAP. Retrieved 1 February 2016.

5. ^ *"LDAP - Lightweight Directory Access Protocol"*

   . Webopedia.com. 4 December 1996. Retrieved 2014-04-05.

6. ^ The X.500 series - ITU-T Rec. X.500 to X.521
7. ^ *Howes, Tim. "The Lightweight Directory Access Protocol: X.500 Lite"*

   (PDF). Retrieved 26 December 2012.

8. ^ *"Pre-History of LDAP"*

   . Cyber Matters. 2013-04-09. Retrieved 5 October 2014.

9. ^ *"Service Name and Transport Protocol Port Number Registry"*

   . IANA. Retrieved 24 March 2021.

Tech Savvy Productions

10. ^ RFC3494
11. ^ Add section of RFC4511
12. ^ LDAP result codes
13. ^ SASL Mechanisms at IANA
14. ^ RFC4511: delete request
15. ^ Boreham Draft (numSubordinates)
16. ^ Modify Section of RFC4511
17. ^ *Zeilenga, K. LDAP Modify-Increment Extension*

    . *doi:10.17487/RFC4525*

    . *RFC 4525*

    .

18. ^ *Zeilenga, K. Lightweight Directory Access Protocol (LDAP) Read Entry Controls*

    . *IETF. doi:10.17487/RFC4527*

    . *RFC 4527*

    .

19. ^ INTERNET-DRAFT LDAP Transactions draft-zeilenga-ldap-txn-15.txt
20. ^ Shibboleth Security alert 20120227
21. ^ Tools.ietf.org
22. ^ Tools.ietf.org
23. ^ Tools.ietf.org
24. ^ Openldap.org
25. ^ Open Grid Forum : Project Home

Sources

- ITU-T Rec. X.680, "Abstract Syntax Notation One (ASN.1) - Specification of Basic Notation", 1994

Tech Savvy Productions

- Basic encoding rules (BER) - ITU-T Rec. X.690, "Specification of ASN.1 encoding rules: Basic, Canonical, and Distinguished Encoding Rules", 1994
- RFC 3641

  - Generic String Encoding Rules (GSER) for ASN.1 Types

- RFC 4346

  - The TLS Protocol Version 1.1

- RFC 4422

  - Simple Authentication and Security Layer (SASL)

- SASL mechanisms registered at IANA

Further reading

- *Arkills, B (2003). LDAP Directories Explained: An Introduction and Analysis*

  . Addison-Wesley Professional. ISBN 978-0-201-78792-4.

- *Carter, G (2003). LDAP System Administration*

  . O'Reilly Media. ISBN 978-1-56592-491-8.

- *Donley, C (2002). LDAP Programming, Management, and Integration. Manning Publications. ISBN 978-1-930110-40-3.*
- *Howes, T; Smith, M; Good, G (2003). Understanding and Deploying LDAP Directory Services*

  . Addison-Wesley Professional. ISBN 978-0-672-32316-4.

- *Rhoton, J (1999). Programmer's Guide to Internet Mail: SMTP, POP, IMAP, and LDAP. Elsevier. ISBN 978-1-55558-212-8.*

Tech Savvy Productions

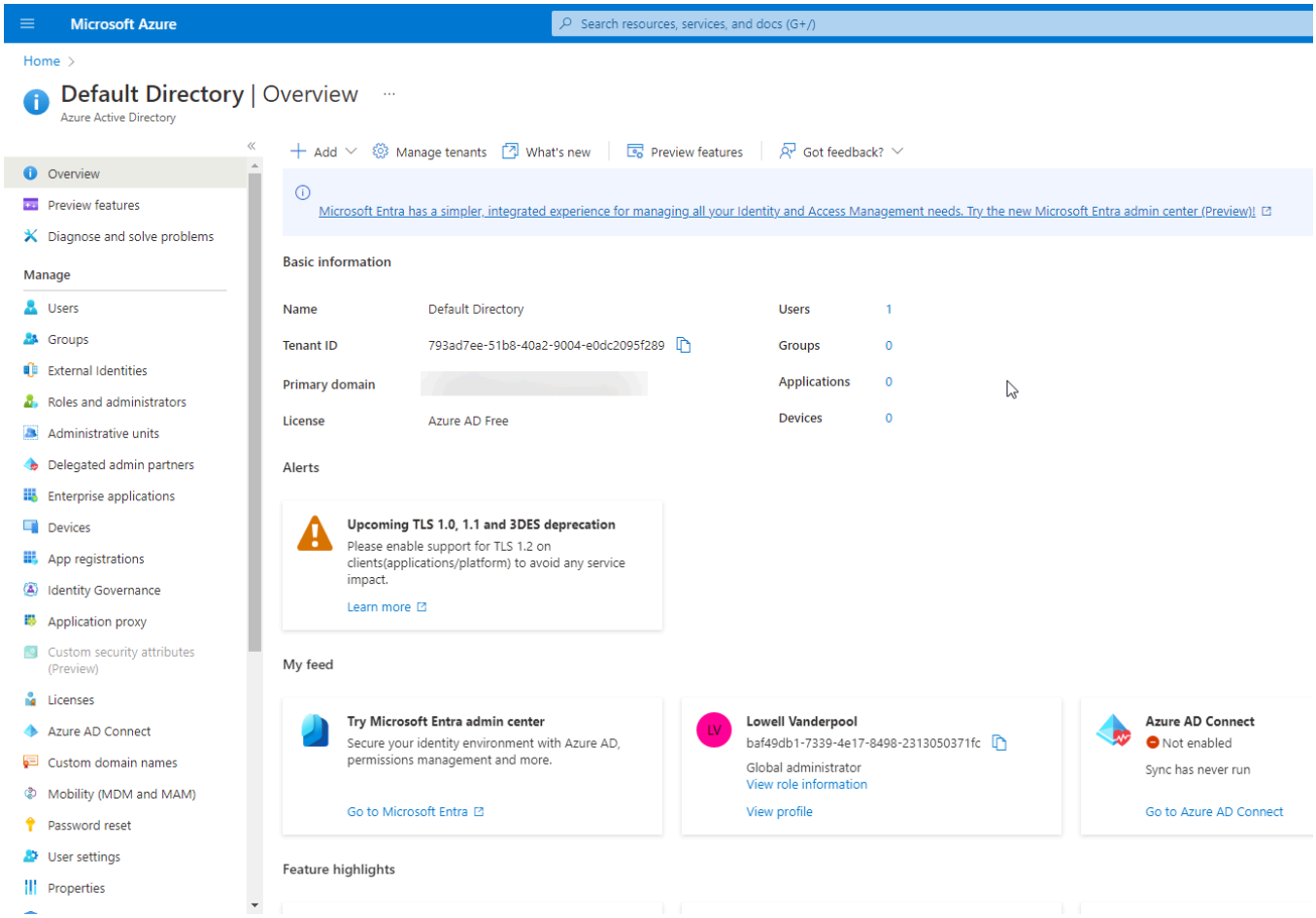
- *Voglmaier, R (2003). The ABCs of LDAP: How to Install, Run, and Administer LDAP Services. Auerbach Publications. ISBN 978-0-8493-1346-2.*

External links

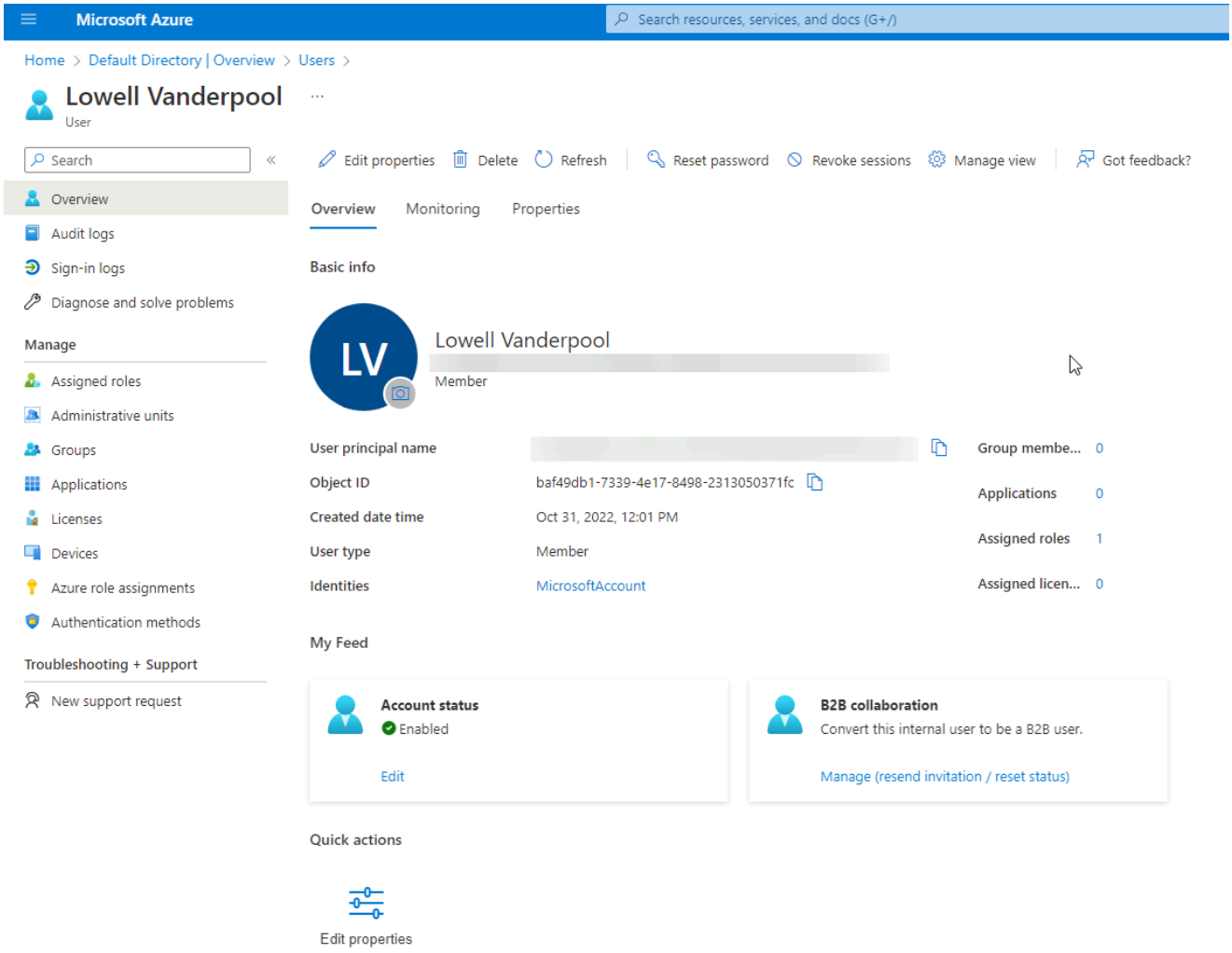
- List of public LDAP Servers (2013): *"Ldapwiki: Public LDAP Servers"*

Tech Savvy Productions

Azure Active Directory

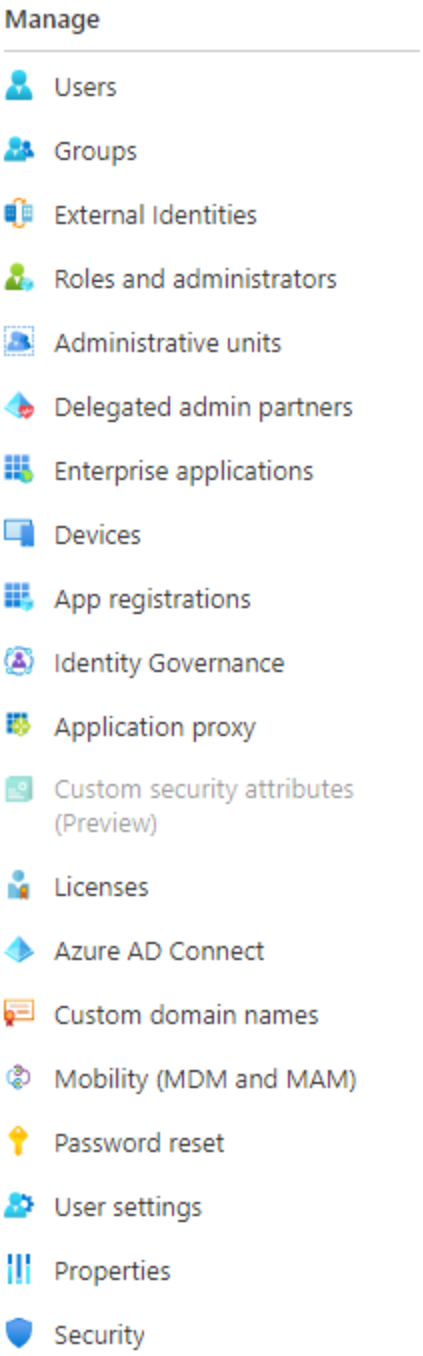


Tech Savvy Productions

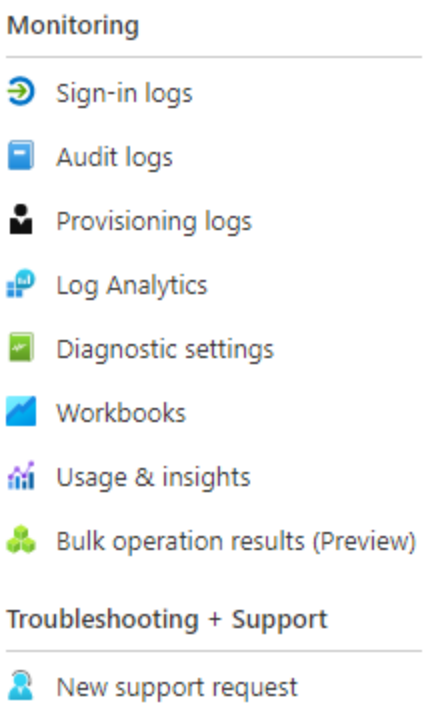


Azure Active Directory Main Menu options

Tech Savvy Productions



Tech Savvy Productions


Monitoring
Sign-in logs
Audit logs
Provisioning logs
Log Analytics
Diagnostic settings
Workbooks
Usage & insights
Bulk operation results (Preview)
Troubleshooting + Support
New support request

Michael Reinders

Feb 14, 2022

Learn all there is to know about how Active Directory (AD) replication works. This guide covers the basics of how domain controllers (DCs) replicate all of your user accounts, passwords, computers, and other objects in your environment. Learn about how sites define the logical layout of your network and how the tools and features in Active Directory Domain Services work together to keep your directory running smoothly.

Table of Contents [hide]

- Active Directory replication
  - How does Microsoft Active Directory replication work?
  - How often does Active Directory replication occur?
- Primary replication components
  - Knowledge Consistency Checker (KCC)
  - Directory System Agent (DSA)
  - Extensible Storage Engine (ESE)
  - Remote Procedure Call (RPC)
  - Inter-Site Topology Generator (ISTG)
- Active Directory replication objects
  - Failover functionality
  - Subnet
  - Site
  - Site link
  - Site link bridge
  - Site link transitivity
  - Global Catalog server
  - Universal group membership caching
- Replication management
  - Check the replication health
  - Check the inbound replication requests that are queued.
  - Check the replication status
  - Synchronize replication between replication partners
  - Force the Knowledge Consistency Checker to recalculate the topology

Tech Savvy Productions

- How to monitor AD replication: Commands and tools

Active Directory replication

Have you ever wondered what happens behind the scenes when you add a new user to Active Directory? Or change their password? Or join a new computer in your domain? If you add a new user on 'DC01', how and when does 'DC05' know about it? Active Directory replication, that's how. This is the engine that pushes changes from domain controller to domain controller in your environment.

How does Microsoft Active Directory replication work?

The replication of updates between Active Directory objects means that data is sent between multiple domain controllers to keep replicas of directory partitions synchronized. Multiple domains are common in large organizations, as are multiple sites in disparate locations. Also, DCs for the same domain are generally placed in more than one site to accommodate compliance, mergers and acquisitions, etc. Each time a new domain controller is added to an existing domain or forest, new components are created, connection objects, site links, etc. And when you add a new DC on a new subnet, MAGIC! Yes, a new 'subnet' is created in Active Directory sites and services.

How often does Active Directory replication occur?

Well, that depends. By default, for domain controllers that are in the same site (intra-site replication), replication occurs **every 15 seconds.** As soon as you change an attribute of an AD object, for instance the job title of your newly promoted systems engineer, the DC will send out the update to its replication partner.

Advertisement

Inter-site replication is set to 180 minutes (three hours). There are a lot of variables, topologies, hardware, subnets, and overall network design that goes into the planning of inter-site replication. With separate offices in New York and Los Angeles, it may be acceptable to wait up to three hours for changes to take effect across the country. Again, there are many deployment methods you could use. You may have a root forest and child domains in each site (city). At that point, waiting three hours for someone's password to update from New York to Los Angeles may work fine for your business needs.

Tech Savvy Productions

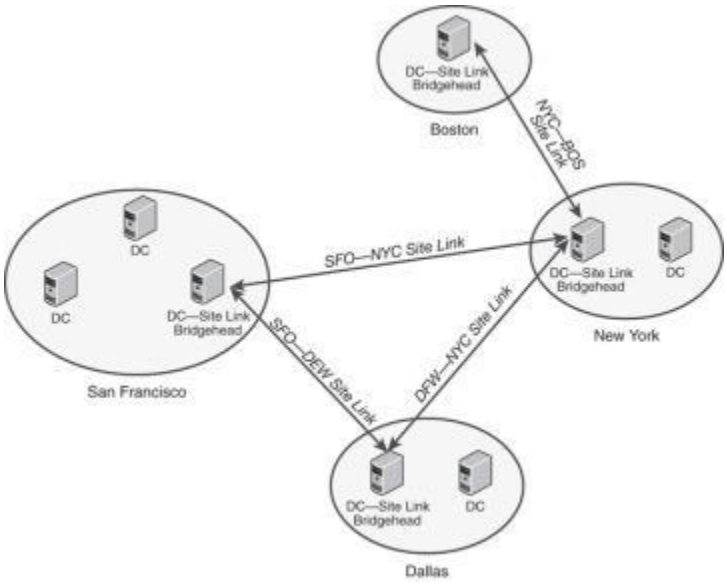
Primary replication components

Let's briefly go over the basic replication components in Active Directory replication and explain their purpose.

Knowledge Consistency Checker (KCC)

All domain controllers have a built-in process called the Knowledge Consistency Checker (KCC). This generates replication topology for the Active Directory forest. The KCC will automatically create separate replication topologies based on your intra-site and inter-site topologies. To accommodate the addition of new DCs, the removal of DCs, or the act of physically moving DCs from one geographic location to another, the KCC also dynamically adjusts the topology on a periodic schedule.

Within a site, the connections between writable domain controllers are always arranged in a bi-directional ring, with additional shortcut connections to reduce latency in large sites. On the other hand, the inter-site topology is a layering of spanning trees, which means one inter-site connection exists between any two sites for each directory partition and generally does not contain shortcut connections.



Understanding AD replication

Directory System Agent (DSA)

In Active Directory Domain Services, the Directory System Agent (DSA) is part of the Local System Authority (LSA) subsystem. The DSA is a collection of services and processes that run on each domain controller and provides access to the data

Tech Savvy Productions

store. The data store is the physical store of directory data location on the server's hard disk.

Extensible Storage Engine (ESE)

The Extensible Storage Engine (ESE) is a database engine utilized by Active Directory Domain Services that stores information in a logical sequence. Information can be retrieved by various methods, for example, sequentially or by accessing defined indices. Database updates are implemented with a transaction in order to ensure secure operations.

The ESE is a relatively scalable design, scalable to large or small applications. The ESE also provides consistent data access in the event of a system crash. This provides a lightweight, highly responsive engine to store the 'raw contents' of Active Directory.

Remote Procedure Call (RPC)

Within a site, Active Directory replication uses Remote Procedure Call (RPC) over IP for replication. RPC is an industry-standard protocol for client/server communications. It is highly compatible with a wide variety of network types.

For replication within a site, RPC provides uniform, high-speed connectivity. This efficiency provides the perfect backdrop for even complex layouts of network and multiple site Active Directory topologies.

Inter-Site Topology Generator (ISTG)

The Inter-Site Topology Generator is an Active Directory process that defines the replication between Active Directory sites on a network. A single domain controller in each site is automatically designated to be the Inter-Site Topology Generator. Because this action is automatic, you don't need to complete any explicit steps during the installation of new DCs.

The domain controller that holds the Inter-Site Topology Generator role performs two functions:

- It automatically selects one or more domain controllers to become bridgehead servers, which are DCs that are mainly used for intersite replication. If a bridgehead server becomes unavailable, the Inter-Site Topology Generator automatically selects another bridgehead server.

Tech Savvy Productions

- It runs the Knowledge Consistency Checker to determine the replication topology and resultant connection objects that the bridgehead servers can use to communicate with bridgehead servers of other sites.

Active Directory replication objects

Next, we'll go over and explain the various types of objects you'll come across in the design of your Active Directory topology.

Failover functionality

What happens when one of your domain controllers is offline or down for maintenance? Will its replication partners simply be content with potentially stale information? Hardly.

Sites (and services) ensure that replication is routed around network failures and domain controllers unavailable due to maintenance. Remember, the Knowledge Consistency Checker runs automatically at specified intervals to keep track of the logical topology of your network and DCs. The KCC will periodically review the replication status of existing connections to find out if any connections are inoperable. If a connection is not working due to a failed domain controller, the KCC will automatically build temporary connections to other replication partners (if they're available) to ensure that replication occurs. If all the DCs in a site are unavailable, the KCC will create replication connections between domain controllers from another site.

Subnet

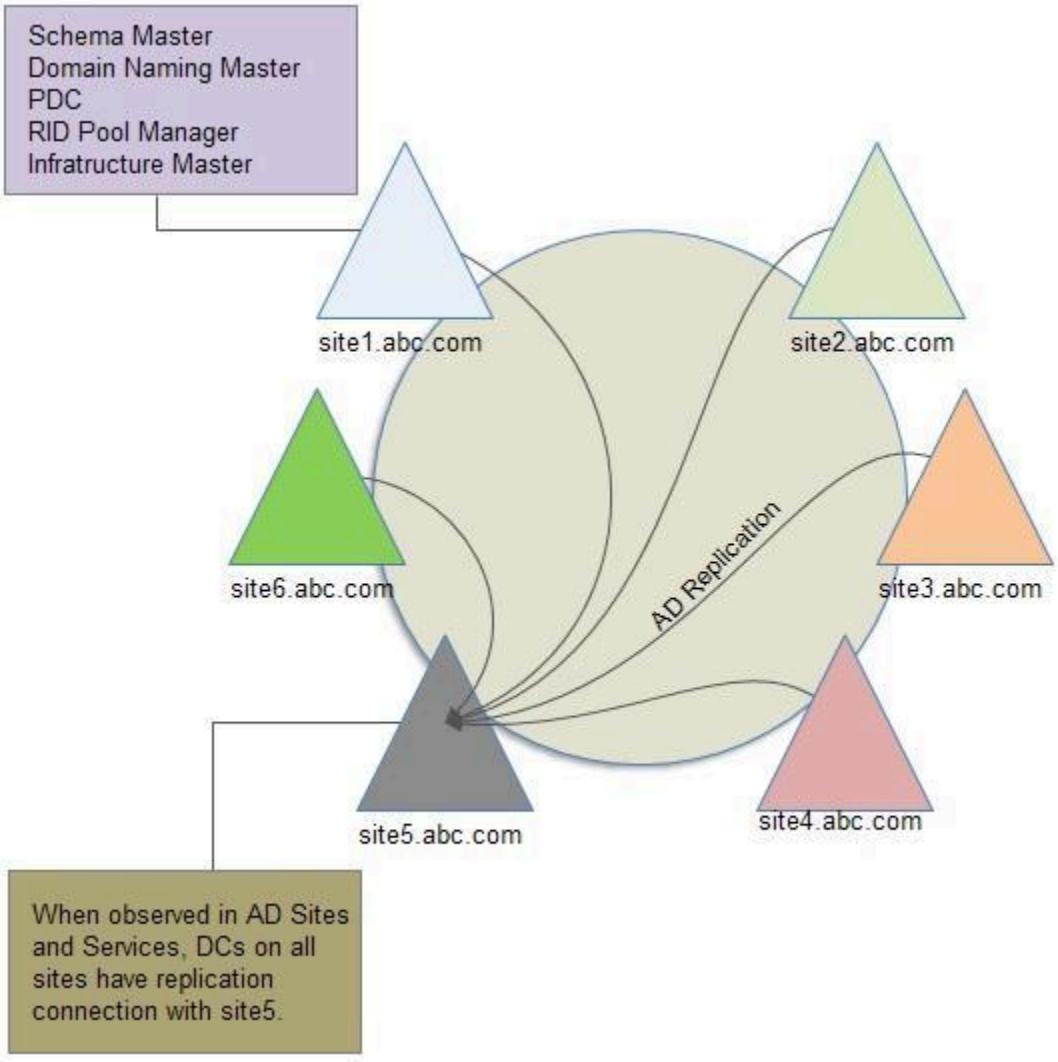
Now, I would imagine most IT Pros know what a Subnet is in TCP/IP arenas, but let's be complete: A Subnet is a segment of a TCP/IP network that defines a set of logical IP addresses. Subnets group computers in a way that identifies their relative physical location on your network. Subnet objects in Active Directory Domain Services help to identify the network addresses that are used to map computers to sites.

Site

If you do a search on this post for the word 'site', you'll see quite a few hits (up to this point). Sites are a very large topic when it comes to the logical layout of your Active Directory topology. A site is an Active Directory object that represents one or more TCP/IP subnets with highly reliable and 'fast' network connections. Site information allows admins to configure AD access and replication to optimize

Tech Savvy Productions

usage of their physical network. Site objects are associated with a collection of subnets, and each domain controller in a forest is associated with an AD site according to its IP address.



AD sites and services

Site link

When the Knowledge Consistency Checker creates a connection object for domain controllers between sites (setting up inter-site replication), site links are created. They are Active Directory objects that represent logical paths for Active Directory replication. A site link object represents a set of sites that can communicate at a uniform cost through a specified inter-site transport.

Tech Savvy Productions

All sites contained within the site link are connected by means of the same (logical) network type. Sites must be manually linked to other sites by using site links so that domain controllers in one site can replicate directory changes from DCs in another site.

Do you remember when we previously learned about the Inter-Site Topology Generator and bridge servers? Well, that knowledge will help you understand this: When two sites are connected by a site link, the replication system automatically creates connections between specific domain controllers in each site that are called bridgehead servers. When the KCC creates replication connections, they are randomly distributed among all candidate bridgehead servers in a site to share the replication workload.

Site link bridge

A site link bridge is an AD object that represents a set of site links. These 'connected' sites can communicate by using a common transport. Site link bridges enable domain controllers that are not directly connected via a communication link to replicate with each other. Typically, a site link bridge corresponds to a router (or a set of routers) on an IP network.

The Knowledge Consistency Checker forms a transitive route through all site links that have some sites in common. Each site link represents its own distinct and isolated network if this behavior is disabled. Containers of site links that can be treated as a single route are expressed through a site link bridge. Each bridge represents an isolated segment of the communication environment for network traffic.

Site link bridges are used to represent transitive physical connectivity accurately and logically between sites. A site link bridge offers the Knowledge Consistency Checker the ability to use any combination of the included site links to determine the least expensive route to interconnect directory partitions held in those sites. The site link bridge doesn't actually provide connectivity to the domain controllers. If the site link bridge is removed, replication will continue until the Knowledge Consistency Checker removes the links.

Site link transitivity

By default, all site links are transitive, or "bridged." When you bridge site links and the schedules overlap, the Knowledge Consistency Checker will create replication connections that determine domain controller replication partners between sites,

Tech Savvy Productions

where the sites are not connected directly by site links but are connected transitively through a set of common sites. This means that you can connect any site to any other site through a combination of site links.

Global Catalog server

A Global Catalog (GC) server is a domain controller that stores information about all objects in the forest so that users, applications, and other pieces of software can search Active Directory Domain Services without referring to specific DCs that store the requested data. Like all DCs, a Global Catalog server stores complete, updateable replicas of the configuration and schema directory partitions and a full, writable replica of the domain directory partition for the domain that it is hosting. In addition, a Global Catalog server stores a partial, read-only replica of every other domain in the forest. Partial, read-only domain replicas contain every object in the domain but only a subset of the attributes (those attributes that are most commonly used for searching the object).

Universal group membership caching

Universal group membership caching allows the domain controllers to cache the membership of universal groups info for users. You can enable domain controllers that are running Windows Server 2008 or higher to cache universal group memberships by using the Active Directory Sites and Services snap-in.

You can avoid installing a Global Catalog server at every site in a domain by enabling universal group membership caching. This will minimize network bandwidth usage because a domain controller does not need to replicate all of the objects located in the forest. It also reduces logon times because the authenticating domain controllers do not always need to access a global catalog to obtain universal group membership information.

Replication management

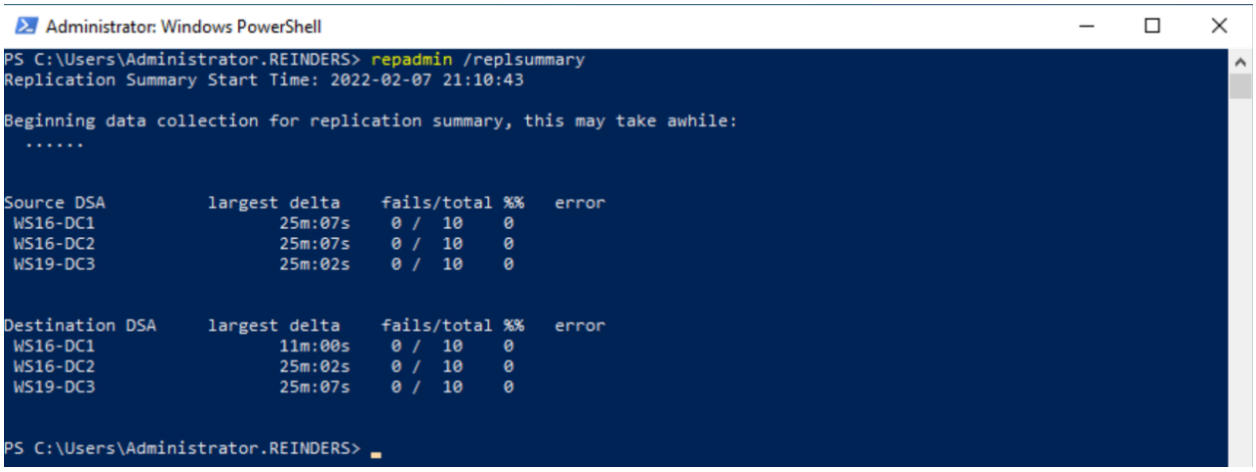
There are various tools you can use to monitor and manage the Active Directory replication status in your environment: Active Directory Sites and Services, PowerShell, the trusty Command Line, and the Active Directory Replication Status Tool (I'll show you this at the end of this post…). For now, let me show you how to use the 'Repadmin' command to discover the status and any potential replication errors present in your environment.

Tech Savvy Productions

Check the replication health

To check the overall replication health of your domain controllers, run this command:

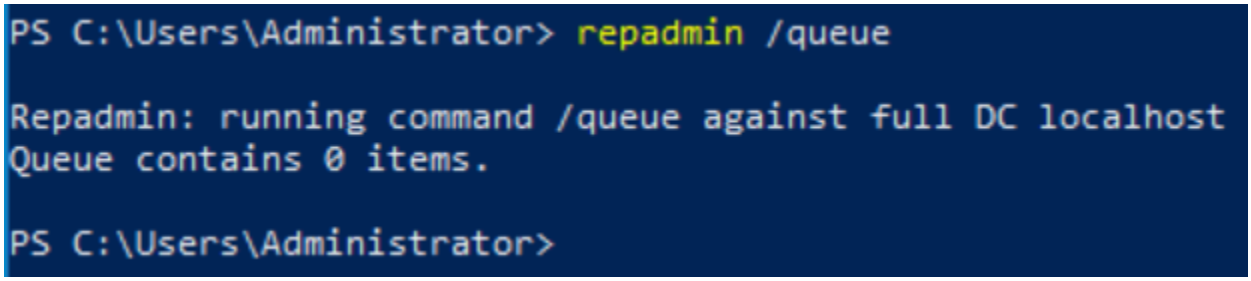
repadmin /replsummary



Checking your AD Replication Health

Check the inbound replication requests that are queued.

Repadmin /Queue

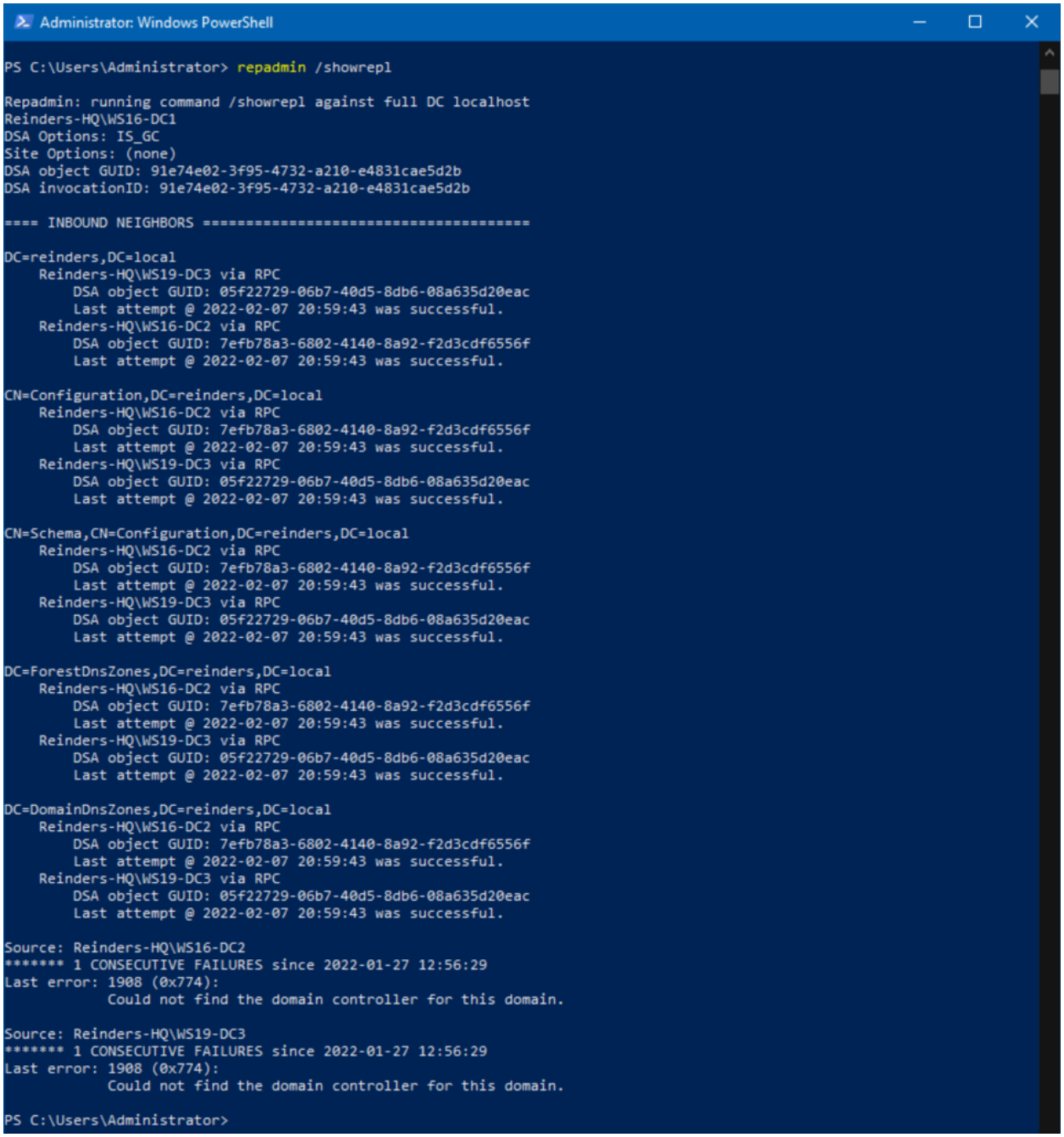


Checking the queue during replication

Check the replication status

Repadmin /Showrepl

Tech Savvy Productions



Showing full replication status

Synchronize replication between replication partners

Repadmin /syncall

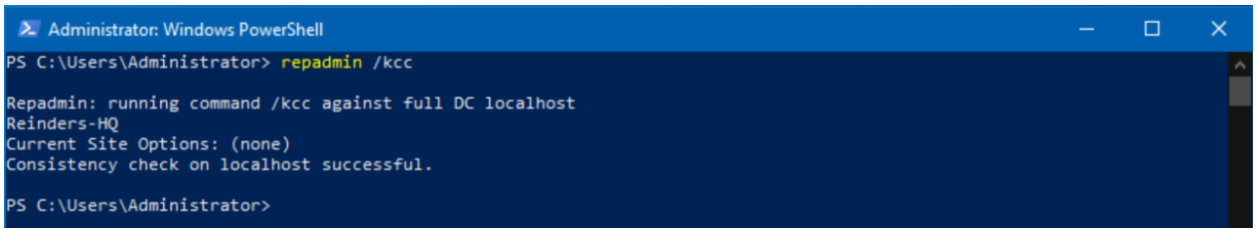
Tech Savvy Productions

```
PS C:\Users\Administrator> repadmin /syncall
CALLBACK MESSAGE: The following replication is in progress:
    From: 7efb78a3-6802-4140-8a92-f2d3cdf6556f._msdcs.reinders.local
    To  : 91e74e02-3f95-4732-a210-e4831cae5d2b._msdcs.reinders.local
CALLBACK MESSAGE: The following replication completed successfully:
    From: 7efb78a3-6802-4140-8a92-f2d3cdf6556f._msdcs.reinders.local
    To  : 91e74e02-3f95-4732-a210-e4831cae5d2b._msdcs.reinders.local
CALLBACK MESSAGE: The following replication is in progress:
    From: 05f22729-06b7-40d5-8db6-08a635d20eac._msdcs.reinders.local
    To  : 91e74e02-3f95-4732-a210-e4831cae5d2b._msdcs.reinders.local
CALLBACK MESSAGE: The following replication completed successfully:
    From: 05f22729-06b7-40d5-8db6-08a635d20eac._msdcs.reinders.local
    To  : 91e74e02-3f95-4732-a210-e4831cae5d2b._msdcs.reinders.local
CALLBACK MESSAGE: SyncAll Finished.
SyncAll terminated with no errors.

PS C:\Users\Administrator>
```

Running the /Syncall switch to force replication

Force the Knowledge Consistency Checker to recalculate the topology

Repadmin /KCC



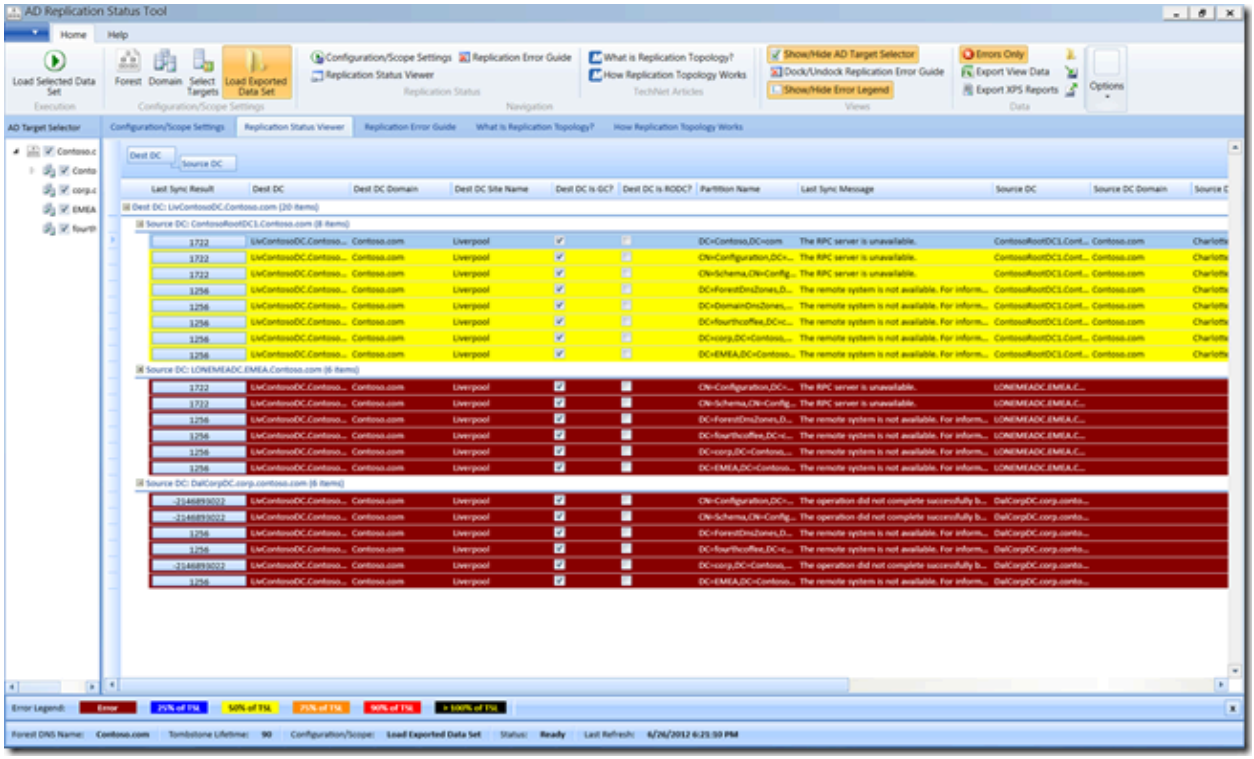
Using the /KCC switch to tell the KCC to manually re-check all replication partners are valid

How to monitor AD replication: Commands and tools

Besides the RepAdmin.exe command-line tool, there's also a Graphical User Interface (GUI) front-end called Active Directory Replication Status Tool.

The Active Directory Replication Status Tool (ADREPLSTATUS) does some analyzing of the replication status for domain controllers in an Active Directory domain or forest. ADREPLSTATUS is essentially a helpful, front-end GUI for the commands I showed you above. This is similar to REPADMIN /SHOWREPL * /CSV imported into Excel, but with significant enhancements.

Tech Savvy Productions



The Active Directory Replication Status Tool

Specific capabilities for this tool include:

- Expose Active Directory replication errors occurring in a domain or forest
- Prioritize errors that need to be resolved in order to avoid the creation of lingering objects in Active Directory forests
- Help administrators and support professionals resolve replication errors by linking to Active Directory replication troubleshooting content on Microsoft TechNet
- Allow replication data to be exported to source or destination domain administrators or support professionals for offline analysis

To download the Active Directory Replication Status Tool, click here.

Active Directory replication is quite a technical topic, but we hope this guide helped you to understand everything you need to know about primary replication components, Active Directory replication objects, and replication management.

Tech Savvy Productions

[How to Restore Active Directory]

How Active Directory Replication Topology Works

In this article

1. How Active Directory Replication Topology Works
2. Active Directory KCC Architecture and Processes
3. Replication Topology Physical Structure
4. Performance Limits for Replication Topology Generation

Applies To: Windows Server 2003, Windows Server 2003 R2, Windows Server 2003 with SP1, Windows Server 2003 with SP2, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2

How Active Directory Replication Topology Works

In this section

- Active Directory KCC Architecture and Processes
- Replication Topology Physical Structure
- Performance Limits for Replication Topology Generation
- Goals of Replication Topology
- Topology-Related Objects in Active Directory
- Replication Transports
- Replication Between Sites
- KCC and Topology Generation
- Network Ports Used by Replication Topology
- Related Information

Active Directory implements a replication topology that takes advantage of the network speeds within sites, which are ideally configured to be equivalent to local area network (LAN) connectivity (network speed of 10 megabits per second [Mbps] or higher). The replication topology also minimizes the use of potentially slow or expensive wide area network (WAN) links between sites.

Tech Savvy Productions

Note

In Windows 2000 Server and Windows Server 2003, the directory service is named Active Directory. In Windows Server 2008 and Windows Server 2008 R2, the directory service is named Active Directory Domain Services (AD DS). The rest of this topic refers to Active Directory, but the information is also applicable to AD DS.

When you create a site object in Active Directory, you associate one or more Internet Protocol (IP) subnets with the site. Each domain controller in a forest is associated with an Active Directory site. A client workstation is associated with a site according to its IP address; that is, each IP address maps to one subnet, which in turn maps to one site.

Active Directory uses sites to:

- Optimize replication for speed and bandwidth consumption between domain controllers.
- Locate the closest domain controller for client logon, services, and directory searches.
- Direct a Distributed File System (DFS) client to the server that is hosting the requested data within the site.
- Replicate the system volume (SYSVOL), a collection of folders in the file system that exists on each domain controller in a domain and is required for implementation of Group Policy.

The ideal environment for replication topology generation is a forest that has a forest functional level of at least Windows Server 2003. In this case, replication topology generation is faster and can accommodate more sites and domains than occurs when the forest has a forest functional level of Windows 2000. When at least one domain controller in each site is running Windows Server 2003, more domain controllers in each site can be used to replicate changes between sites than when all domain controllers are running Windows 2000 Server.

In addition, replication topology generation requires the following conditions:

Tech Savvy Productions

- A Domain Name System (DNS) infrastructure that manages the name resolution for domain controllers in the forest. Active Directory–integrated DNS is assumed, wherein DNS zone data is stored in Active Directory and is replicated to all domain controllers that are DNS servers.
- All physical locations that are represented as site objects in Active Directory have LAN connectivity.
- IP connectivity is available between each site and all sites in the same forest that host operations master roles.
- Domain controllers meet the hardware requirements for Windows Server 2008 R2, Windows Server 2008, Windows Server 2003, Standard Edition; Windows Server 2003, Enterprise Edition; and Windows Server 2003, Datacenter Edition.
- The appropriate number of domain controllers is deployed for each domain that is represented in each site.

This section covers the replication components that create the replication topology and how they work together, plus the mechanisms and rationale for routing replication traffic between domain controllers in the same site and in different sites.

Active Directory KCC Architecture and Processes

The replication topology is generated by the Knowledge Consistency Checker (KCC), a replication component that runs as an application on every domain controller and communicates through the distributed Active Directory database. The KCC functions locally by reading, creating, and deleting Active Directory data. Specifically, the KCC reads configuration data and reads and writes connection objects. The KCC also writes local, nonreplicated attribute values that indicate the replication partners from which to request replication.

For most of its operation, the KCC that runs on one domain controller does not communicate directly with the KCC on any other domain controller. Rather, all KCCs use the knowledge of the common, global data that is stored in the configuration directory partition as input to the topology generation algorithm to converge on the same view of the replication topology.

Tech Savvy Productions

Each KCC uses its in-memory view of the topology to create inbound connections locally, manifesting only those results that apply to itself. The KCC communicates with other KCCs only to make a remote procedure call (RPC) request for replication error information. The KCC uses the error information to identify gaps in the replication topology. A request for replication error information occurs only between domain controllers in the same site.

Note

- The KCC uses only RPC to communicate with the directory service. The KCC does not use Lightweight Directory Access Protocol (LDAP).

One domain controller in each site is selected as the Intersite Topology Generator (ISTG). To enable replication across site links, the ISTG automatically designates one or more servers to perform site-to-site replication. These servers are called bridgehead servers. A bridgehead is a point where a connection leaves or enters a site.

The ISTG creates a view of the replication topology for all sites, including existing connection objects between all domain controllers that are acting as bridgehead servers. The ISTG then creates inbound connection objects for servers in its site that it determines will act as bridgehead servers and for which connection objects do not already exist. Thus, the scope of operation for the KCC is the local server only, and the scope of operation for the ISTG is a single site.

Each KCC has the following global knowledge about objects in the forest, which it gets by reading objects in the Sites container of the configuration directory partition and which it uses to generate a view of the replication topology:

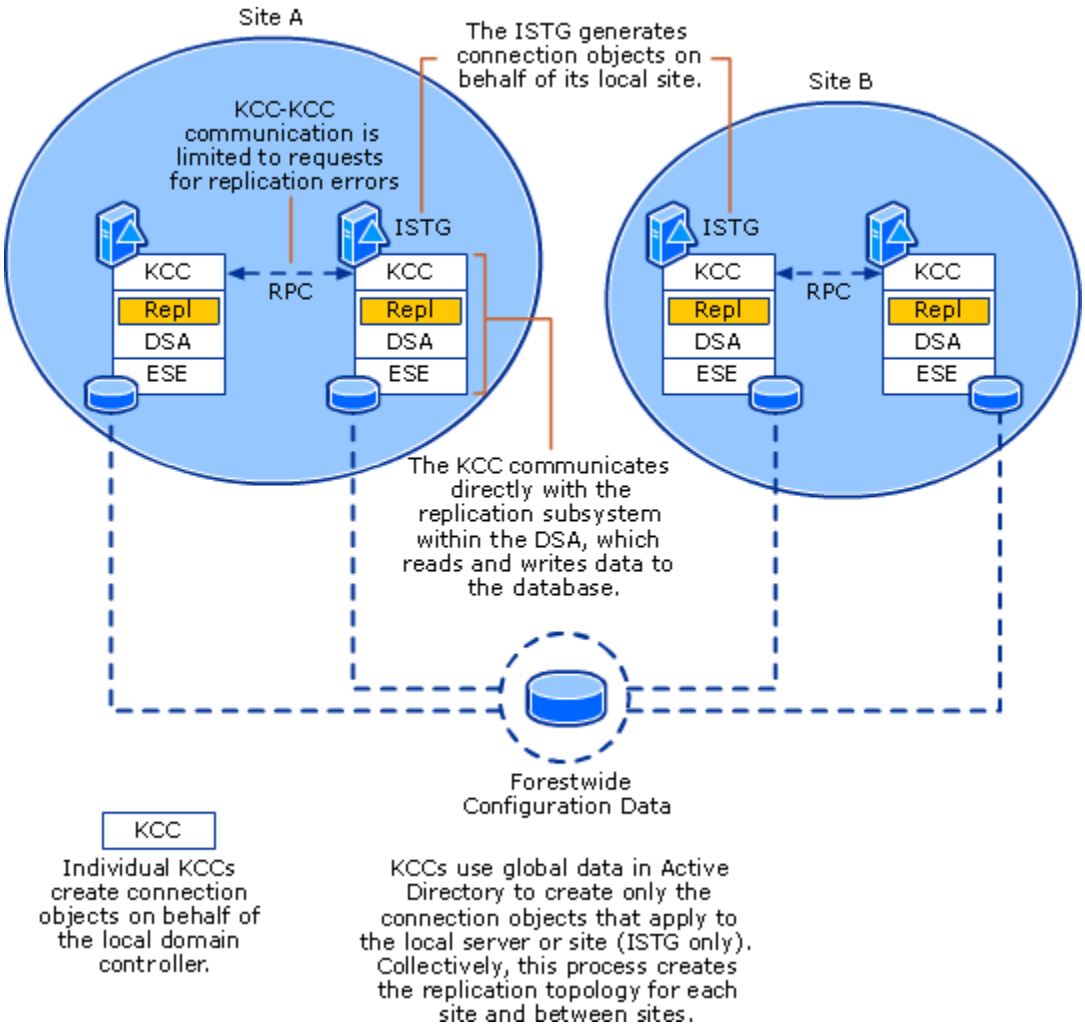
- Sites
- Servers
- Site affiliation of each server
- Global catalog servers
- Directory partitions stored by each server
- Site links
- Site link bridges

Tech Savvy Productions

Detailed information about these configuration components and their functionality is provided later in this section.

The following diagram shows the KCC architecture on servers in the same forest in two sites.

KCC Architecture and Processes



The architecture and process components in the preceding diagram are described in the following table.

KCC Architecture and Process Components

Tech Savvy Productions

Component	Description
Knowledge Consistency Checker (KCC)	The application running on each domain controller that communicates directly with the Ntdsa.dll to read and write replication objects.
Directory System Agent (DSA)	The directory service component that runs as Ntdsa.dll on each domain controller, providing the interfaces through which services and processes such as the KCC gain access to the directory database.
Extensible Storage Engine (ESE)	The directory service component that runs as Esent.dll. ESE manages the tables of records, each with one or more columns. The tables of records comprise the directory database.
Remote procedure call (RPC)	The Directory Replication Service (Drsuapi) RPC protocol, used to communicate replication status and topology to a domain controller. The KCC also uses this protocol to communicate with other KCCs to request error information when building the replication topology.
Intersite Topology Generator (ISTG)	The single KCC in a site that manages intersite connection objects for the site.

The four servers in the preceding diagram create identical views of the servers in their site and generate connection objects on the basis of the current state of Active Directory data in the configuration directory partition. In addition to creating its view of the servers in its respective site, the KCC that operates as the ISTG in each site also creates a view of all servers in all sites in the forest. From this view, the ISTG determines the connections to create on the bridgehead servers in its own site.

Note

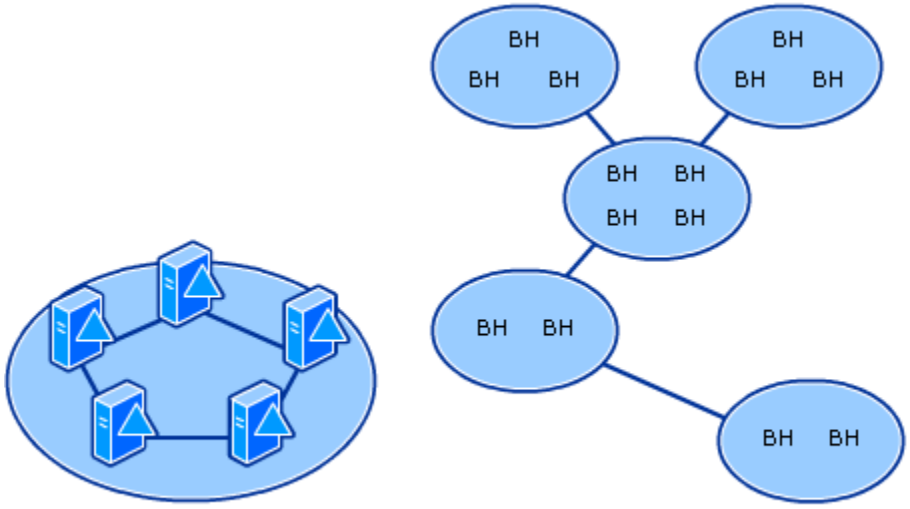
- A connection requires two endpoints: one for the destination domain controller and one for the source domain controller. Domain controllers creating an intrasite topology always use themselves as the destination end point and must consider only the endpoint for the source domain controller.

Tech Savvy Productions

The ISTG, however, must identify both endpoints in order to create connection objects between two other servers.

Thus, the KCC creates two types of topologies: intrasite and intersite. Within a site, the KCC creates a ring topology by using all servers in the site. To create the intersite topology, the ISTG in each site uses a view of all bridgehead servers in all sites in the forest. The following diagram shows a high-level generalization of the view that the KCC sees of an intrasite ring topology and the view that the ISTG sees of the intersite topology. Lines between domain controllers within a site represent inbound and outbound connections between the servers. The lines between sites represent configured site links. Bridgehead servers are represented as BH.

KCC and ISTG Views of Intrasite and Intersite Topology



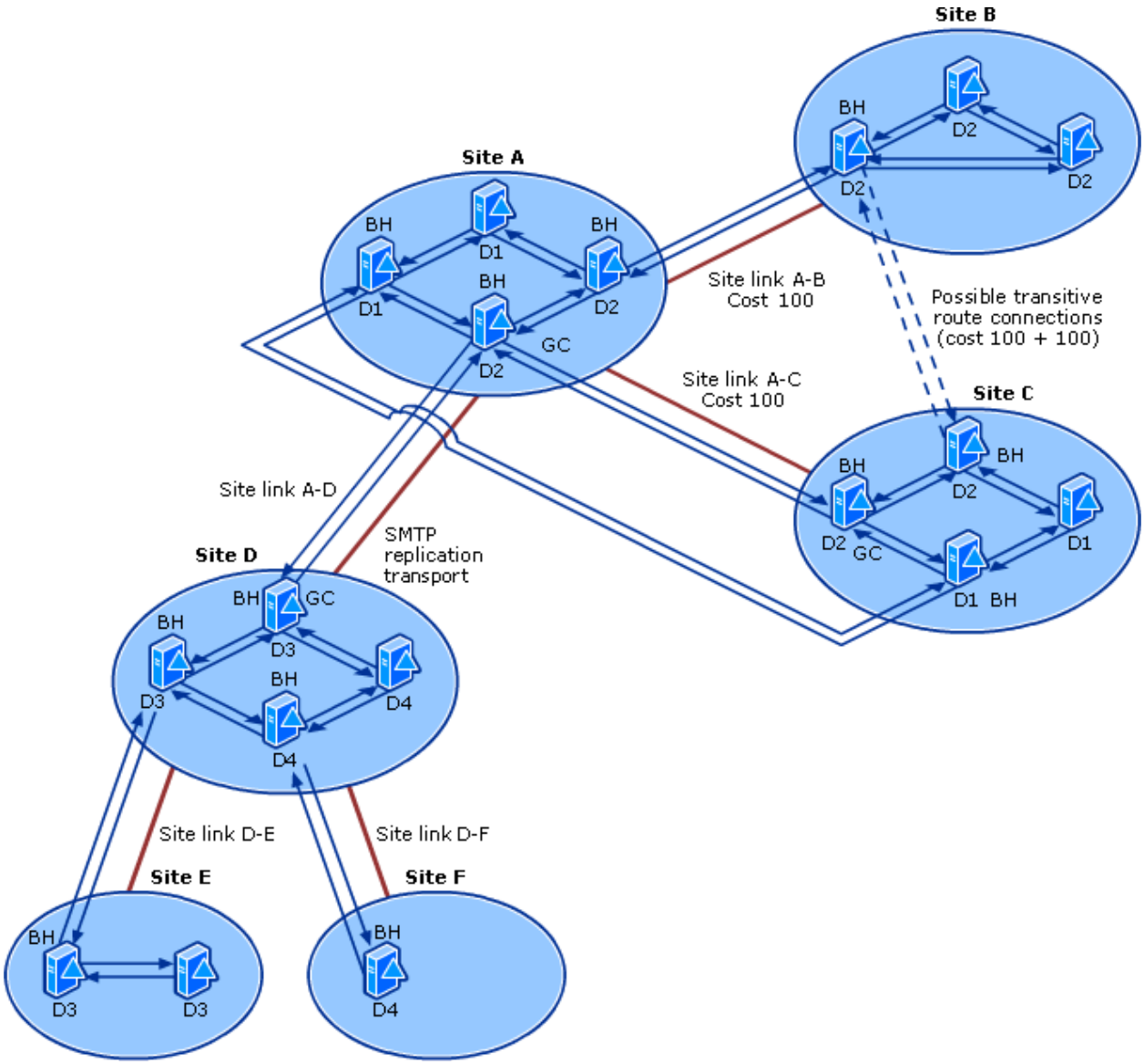
Replication Topology Physical Structure

The Active Directory replication topology can use many different components. Some components are required and others are not required but are available for optimization. The following diagram illustrates most replication topology components and their place in a sample Active Directory multisite and multidomain forest. The depiction of the intersite topology that uses multiple bridgehead servers for each domain assumes that at least one domain controller

Tech Savvy Productions

in each site is running at least Windows Server 2003. All components of this diagram and their interactions are explained in detail later in this section.

Replication Topology Physical Structure



In the preceding diagram, all servers are domain controllers. They independently use global knowledge of configuration data to generate one-way, inbound connection objects. The KCCs in a site collectively create an intrasite topology for all domain controllers in the site. The ISTGs from all sites collectively create an intersite topology. Within sites, one-way arrows indicate the inbound connections by which each domain controller replicates changes from its partner in the ring. For intersite replication, one-way arrows represent inbound connections that are

Tech Savvy Productions

created by the ISTG of each site from bridgehead servers (BH) for the same domain (or from a global catalog server [GC] acting as a bridgehead if the domain is not present in the site) in other sites that share a site link. Domains are indicated as D1, D2, D3, and D4.

Each site in the diagram represents a physical LAN in the network, and each LAN is represented as a site object in Active Directory. Heavy solid lines between sites indicate WAN links over which two-way replication can occur, and each WAN link is represented in Active Directory as a site link object. Site link objects allow connections to be created between bridgehead servers in each site that is connected by the site link.

Not shown in the diagram is that where TCP/IP WAN links are available, replication between sites uses the RPC replication transport. RPC is always used within sites. The site link between Site A and Site D uses the SMTP protocol for the replication transport to replicate the configuration and schema directory partitions and global catalog partial, read-only directory partitions. Although the SMTP transport cannot be used to replicate writable domain directory partitions, this transport is required because a TCP/IP connection is not available between Site A and Site D. This configuration is acceptable for replication because Site D does not host domain controllers for any domains that must be replicated over the site link A-D.

By default, site links A-B and A-C are transitive (bridged), which means that replication of domain D2 is possible between Site B and Site C, although no site link connects the two sites. The cost values on site links A-B and A-C are site link settings that determine the routing preference for replication, which is based on the aggregated cost of available site links. The cost of a direct connection between Site C and Site B is the sum of costs on site links A-B and A-C. For this reason, replication between Site B and Site C is automatically routed through Site A to avoid the more expensive, transitive route. Connections are created between Site B and Site C only if replication through Site A becomes impossible due to network or bridgehead server conditions.

Performance Limits for Replication Topology Generation

Active Directory topology generation performance is limited primarily by the memory on the domain controller. KCC performance degrades at the physical

Tech Savvy Productions

memory limit. In most deployments, topology size will be limited by the amount of domain controller memory rather than CPU utilization required by the KCC.

Scaling of sites and domains is improved in Windows Server 2003 by improving the algorithm that the KCC uses to generate the intersite replication topology. Because all domain controllers must use the same algorithm to arrive at a consistent view of the replication topology, the improved algorithm has a forest functional level requirement of Windows Server 2003 or Windows Server 2003 interim.

KCC scalability was tested on domain controllers with 1.8 GHz processor speed, 512 megabytes (MB) RAM, and small computer system interface (SCSI) disks. KCC performance results at forest functional levels that are at least Windows Server 2003 are described in the following table. The times shown are for the KCC to run where all new connections are needed (maximum) and where no new connections are needed (minimum). Because most organizations add domain controllers in increments, the minimum generation times shown are closest to the actual runtimes that can be expected in deployments of comparable sizes. The CPU and memory usage values for the Local Security Authority (LSA) process (Lsass.exe) indicate the more significant impact of memory versus percent of CPU usage when the KCC runs.

Note

- Active Directory runs as part of the LSA, which manages authentication packages and authenticates users and services.

Minimum and Maximum KCC Generation Times for Domain-Site Combinations

Domains	Sites	Connections	KCC Generation Time (seconds)	Lsass.exe Memory Usage (MB)	Lsass.exe CPU Usage (%)
1	500	Maximum	43	100	39
		Minimum	1	100	29
	1,000	Maximum	49	149	43

Tech Savvy Productions

Domains	Sites	Connections	KCC Generation Time (seconds)	Lsass.exe Memory Usage (MB)	Lsass.exe CPU Usage (%)
		Minimum	2	149	28
	3,000	Maximum	69	236	46
		Minimum	2	236	63
5	500	Maximum	70	125	29
		Minimum	2	126	71
	1,000	Maximum	77	237	28
		Minimum	3	237	78
	2,000	Maximum	78	325	43
		Minimum	5	325	77
	3,000	Maximum	85	449	52
		Minimum	6	449	75
	4,000	Maximum	555	624	46
		Minimum	34	624	69
20	1,000	Maximum	48	423	65
		Minimum	5	423	81
40	1,000	Maximum	93	799	56
		Minimum	12	799	96
	2,000	Minimum	38	874	71

These numbers cannot be used as the sole guidelines for forest and domain design. Other limitations might affect performance and scalability. A limitation of note is that when FRS is deployed, a limit of 1,200 domain controllers per domain is recommended to ensure reliable recovery of SYSVOL.

Tech Savvy Productions

For more information about FRS limitations, see "FRS Technical Reference." For more information about the functional level requirements for the intersite topology generation algorithm, see "Automated Intersite Topology Generation" later in this section.

Goals of Replication Topology

The KCC generates a replication topology that achieves the following goals:

- Connect every directory partition replica that must be replicated.
- Control replication latency and cost.
- Route replication between sites.
- Effect client affinity.

By default, the replication topology is managed automatically and optimizes existing connections. However, manual connections created by an administrator are not modified or optimized.

Connect Directory Partition Replicas

The total replication topology is actually composed of several underlying topologies, one for each directory partition. In the case of the schema and configuration directory partitions, a single topology is created. The underlying topologies are merged to form the minimum number of connections that are required to replicate each directory partition between all domain controllers that store replicas. Where the connections for directory partitions are identical between domain controllers — for example, two domain controllers store the same domain directory partition — a single connection can be used for replication of updates to the domain, schema, and configuration directory partitions.

A separate replication topology is also created for application directory partitions. However, in the same manner as schema and configuration directory partitions, application directory partitions can use the same topology as domain directory partitions. When application and domain directory partitions are common to the source and destination domain controllers, the KCC does not create a separate connection for the application directory partition.

Tech Savvy Productions

A separate topology is not created for the partial replicas that are stored on global catalog servers. The connections that are needed by a global catalog server to replicate each partial replica of a domain are part of the topology that is created for each domain.

The routes for the following directory partitions or combinations of directory partitions are aggregated to arrive at the overall topology:

- Configuration and schema within a site.
- Each writable domain directory partition within a site.
- Each application directory partition within a site.
- Global catalog read-only, partial domain directory partitions within a site.
- Configuration and schema between sites.
- Each writable domain directory partition between sites.
- Each application directory partition between sites.
- Global catalog read-only, partial domain directory partitions between sites.

Replication transport protocols determine the manner in which replication data is transferred over the network media. Your network environment and server configuration dictates the transports that you can use. For more information about transports, see “Replication Transports” later in this section.

Control Replication Latency and Cost

Replication latency is inherent in a multimaster directory service. A period of replication latency begins when a directory update occurs on an originating domain controller and ends when replication of the change is received on the last domain controller in the forest that requires the change. Generally, the latency that is inherent in a WAN link is relative to a combination of the speed of the connection and the available bandwidth. Replication cost is an administrative value that can be used to indicate the latency that is associated with different replication routes between sites. A lower-cost route is preferred by the ISTG when generating the replication topology.

Tech Savvy Productions

Site topology is the topology as represented by the physical network: the LANs and WANs that connect domain controllers in a forest. The replication topology is built to use the site topology. The site topology is represented in Active Directory by site objects and site link objects. These objects influence Active Directory replication to achieve the best balance between replication speed and the cost of bandwidth utilization by distinguishing between replication that occurs within a site and replication that must span sites. When the KCC creates replication connections between domain controllers to generate the replication topology, it creates more connections between domain controllers in the same site than between domain controllers in different sites. The results are lower replication latency within a site and less replication bandwidth utilization between sites.

Within sites, replication is optimized for speed as follows:

- Connections between domain controllers in the same site are always arranged in a ring, with possible additional connections to reduce latency.
- Replication within a site is triggered by a change notification mechanism when an update occurs, moderated by a short, configurable delay (because groups of updates frequently occur together).
- Data is sent uncompressed, and thus without the processing overhead of data compression.

Between sites, replication is optimized for minimal bandwidth usage (cost) as follows:

- Replication data is compressed to minimize bandwidth consumption over WAN links.
- Store-and-forward replication makes efficient use of WAN links — each update crosses an expensive link only once.
- Replication occurs at intervals that you can schedule so that use of expensive WAN links is managed.
- The intersite topology is a layering of spanning trees (one intersite connection between any two sites for each directory partition) and generally does not contain redundant connections.

Tech Savvy Productions

Route Replication Between Sites

The KCC uses the information in Active Directory to identify the least-cost routes for replication between sites. If a domain controller is unavailable at the time the replication topology is created, making replication through that site impossible, the next least-cost route is used. This rerouting is automatic when site links are bridged (transitive), which is the default setting.

Replication is automatically routed around network failures and offline domain controllers.

Effect Client Affinity

Active Directory clients locate domain controllers according to their site affiliation. Domain controllers register SRV resource records in the DNS database that map the domain controller to a site. When a client requests a connection to a domain controller (for example, when logging on to a domain computer), the domain controller Locator uses the site SRV resource record to locate a domain controller with good connectivity whenever possible. In this way, a client locates a domain controller within the same site, thereby avoiding communications over WAN links.

Sites can also be used by certain applications, such as DFS, to ensure that clients locate servers that are within the site or, if none is available, a server in the next closest site. If the ISTG is running Windows Server 2003 or later server operating systems, you can specify an alternate site based on connection cost if no same-site servers are available. This DFS feature, called "site costing," is new in Windows Server 2003.

For more information about the domain controller Locator, see "DNS Support for Active Directory Technical Reference." For more information about DFS site costing, see "DFS Technical Reference."

Topology-Related Objects in Active Directory

Active Directory stores replication topology information in the configuration directory partition. Several configuration objects define the components that are required by the KCC to establish and implement the replication topology.

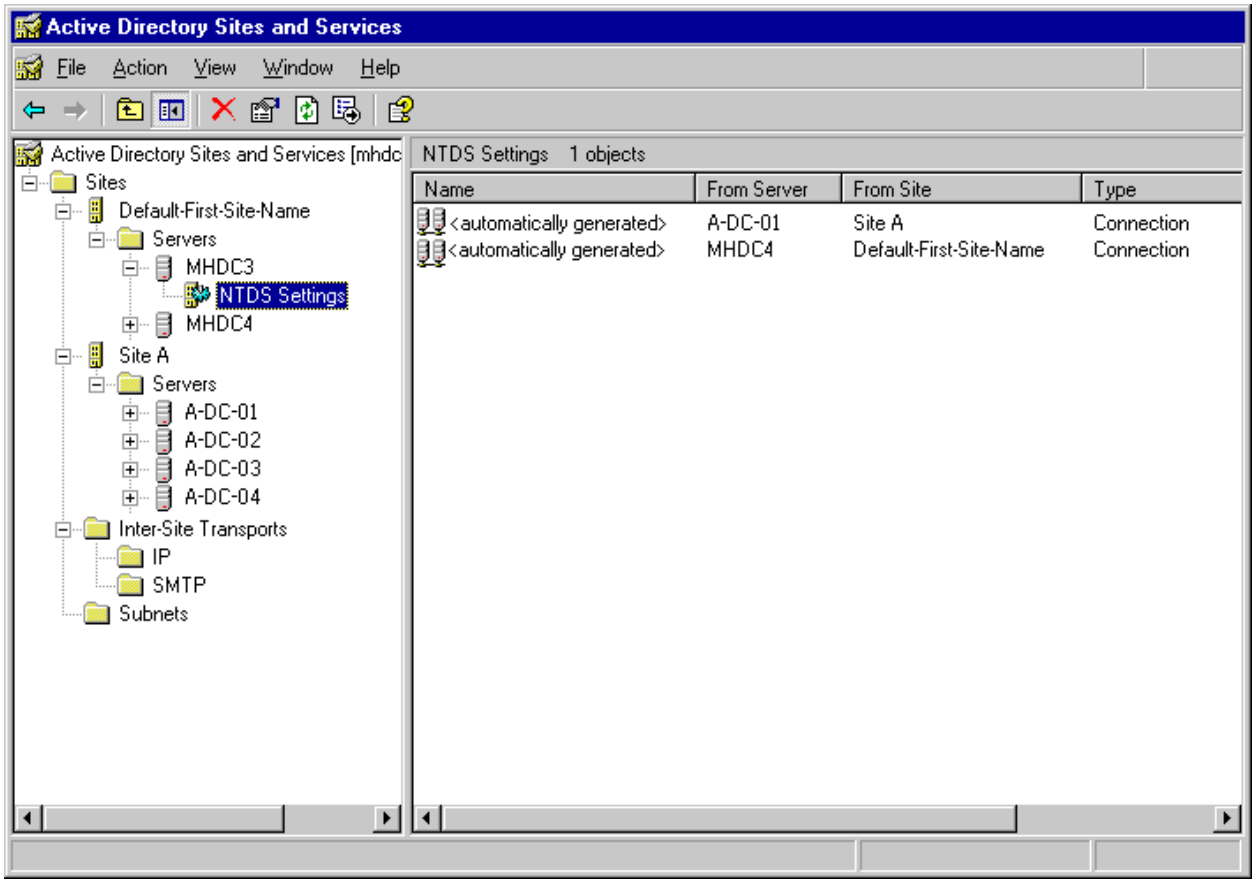
Active Directory Sites and Services is the Microsoft Management Console (MMC) snap-in that you can use to view and manage the hierarchy of objects that are

Tech Savvy Productions

used by the KCC to construct the replication topology. The hierarchy is displayed as the contents of the Sites container, which is a child object of the Configuration container. The Configuration container is not identified in the Active Directory Sites and Services UI. The Sites container contains an object for each site in the forest. In addition, Sites contains the Subnets container, which contains subnet definitions in the form of subnet objects.

The following figure shows a sample hierarchy, including two sites: Default-First-Site-Name and Site A. The selected NTDS Settings object of the server MHDC3 in the site Default-First-Site-Name displays the inbound connections from MHDC4 in the same site and from A-DC-01 in Site A. In addition to showing that MHDC3 and MHDC4 perform intrasite replication, this configuration indicates that MHDC3 and A-DC-01 are bridgehead servers that are replicating the same domain between Site A and Default-First-Site-Name.

Sites Container Hierarchy



Site and Subnet Objects

Tech Savvy Productions

Sites are effective because they map to specific ranges of subnet addresses, as identified in Active Directory by subnet objects. The relationship between sites and subnets is integral to Active Directory replication.

Site Objects

A site object (class **site**) corresponds to a set of one or more IP subnets that have LAN connectivity. Thus, by virtue of their subnet associations, domain controllers that are in the same site are well connected in terms of speed. Each site object has a child NTDS Site Settings object and a Servers container. The distinguished name of the Sites container is CN=Sites,CN=Configuration,DC=*ForestRootDomainName*. The Configuration container is the topmost object in the configuration directory partition and the Sites container is the topmost object in the hierarchy of objects that are used to manage and implement Active Directory replication.

When you install Active Directory on the first domain controller in the forest, a site object named Default-First-Site-Name is created in the Sites container in Active Directory.

Subnet Objects

Subnet objects (class **subnet**) define network subnets in Active Directory. A network subnet is a segment of a TCP/IP network to which a set of logical IP addresses is assigned. Subnets group computers in a way that identifies their physical proximity on the network. Subnet objects in Active Directory are used to map computers to sites. Each subnet object has a **siteObject** attribute that links it to a site object.

Subnet-to-Site Mapping

You associate a set of IP subnets with a site if they have high-bandwidth LAN connectivity, possibly involving hops through high-performance routers.

Note

- LAN connectivity assumes high-speed, inexpensive bandwidth that allows similar and reliable network performance, regardless of which two computers in the site are communicating. This quality of connectivity does not indicate that all servers in the site must be on the same network

Tech Savvy Productions

segment or that hop counts between all servers must be identical. Rather, it is the measure by which you know that if a large amount of data needs to be copied from one server to another, it does not matter which servers are involved. If you find that you are concerned about such situations, consider creating another site.

When you create subnet objects in Active Directory, you associate them with site objects so that IP addresses can be localized according to sites. During the process of domain controller location, subnet information is used to find a domain controller in the same site as, or the site closest to, the client computer. The Net Logon service on a domain controller is able to identify the site of a client by mapping the client's IP address to a subnet object in Active Directory. Likewise, when a domain controller is installed, its server object is created in the site that contains the subnet that maps to its IP address.

You can use Active Directory Sites and Services to define subnets, and then create a site and associate the subnets with the site. By default, only members of the Enterprise Admins group have the right to create new sites, although this right can be delegated.

In a default Active Directory installation, there is no default subnet object, so potentially a computer can be in the forest but have an IP subnet that is not contained in any site. For private networks, you can specify the network addresses that are provided by the Internet Assigned Numbers Authority (IANA). By definition, that range covers all of the subnets for the organization. However, where several class B or class C addresses are assigned, there would necessarily be multiple subnet objects that all mapped to the same default site.

To accommodate this situation, use the following subnets:

- For class B addresses, subnet 128.0.0.0/2 covers all class B addresses.
- For class C addresses, subnet 192.0.0.0/3 covers all class C addresses.

Note

- The Active Directory Sites and Services MMC snap-in neither checks nor enforces IP address mapping when you move a server object to a different

Tech Savvy Productions

site. You must manually change the IP address on the domain controller to ensure proper mapping of the IP address to a subnet in the appropriate site.

Server Objects

Server objects (class **server**) represent server computers, including domain controllers, in the configuration directory partition. When you install Active Directory, the installation process creates a server object in the Servers container within the site to which the IP address of the domain controller maps. There is one server object for each domain controller in the site.

A server object is distinct from the computer object that represents the computer as a security principal. These objects are in separate directory partitions and have separate globally unique identifiers (GUIDs). The computer object represents the domain controller in the domain directory partition; the server object represents the domain controller in the configuration directory partition. The server object contains a reference to the associated computer object.

The server object for the first domain controller in the forest is created in the Default-First-Site-Name site. When you install Active Directory on subsequent servers, if no other sites are defined, server objects are created in Default-First-Site-Name. If other sites have been defined and subnet objects have been associated with these sites, server objects are created as follows:

- If additional sites have been defined in Active Directory and the IP address of the installation computer matches an existing subnet in a defined site, the domain controller is added to that site.
- If additional sites have been defined in Active Directory and the new domain controller's IP address does not match an existing subnet in one of the defined sites, the new domain controller's server object is created in the site of the source domain controller from which the new domain controller receives its initial replication.

When Active Directory is removed from a server, its NTDS Settings object is deleted from Active Directory, but its server object remains because the server object might contain objects other than NTDS Settings. For example, when Microsoft Operations Manager or Message Queuing is running on a domain controller, these applications create child objects beneath the server object.

Tech Savvy Productions

NTDS Settings Objects

The NTDS Settings object (class **nTDSDSA**) represents an instantiation of Active Directory on that server and distinguishes a domain controller from other types of servers in the site or from decommissioned domain controllers. For a specific server object, the NTDS Settings object contains the individual connection objects that represent the inbound connections from other domain controllers in the forest that are currently available to send changes to this domain controller.

Note

- The NTDS Settings object should not be manually deleted.

The **hasMasterNCs** multivalued attribute (where "NC" stands for "naming context," a synonym for "directory partition") of an NTDS Settings object contains the distinguished names for the set of writable (non-global-catalog) directory partitions that are located on that domain controller, as follows:

- DC=Configuration,DC=*ForestRootDomainName*
- DC=Schema,DC=Configuration,DC=*ForestRootDomainName*
- DC=*DomainName*,DC=*ForestRootDomainName*

The **msDSHasMasterNCs** attribute is new attribute introduced in Windows Server 2003, and this attribute of the NTDS Settings object contains values for the above-named directory partitions as well as any application directory partitions that are stored by the domain controller. Therefore, on domain controllers that are DNS servers and use Active Directory–integrated DNS zones, the following values appear in addition to the default directory partitions:

- DC=ForestDNSZones,DC=*ForestRootDomainName* (domain controllers in the forest root domain only)
- DC=DomainDNSZones,DC=*DomainName*,DC=*ForestRootDomainName* (all domain controllers)

Applications that need to retrieve the list of all directory partitions that are hosted by a domain controller can be updated or written to use the **msDSHasMasterNCs** attribute. Applications that need to retrieve only domain directory partitions can continue to use the **hasMasterNCs** attribute.

Tech Savvy Productions

For more information about these attributes, see Active Directory in the Microsoft Platform SDK on MSDN.

Connection Objects

A connection object (class **nTDSConnection**) defines a one-way, inbound route from one domain controller (the source) to the domain controller that stores the connection object (the destination). The KCC uses information in cross-reference objects to create the appropriate connection objects, which enable domain controllers that store the same directory partitions to replicate with each other. The KCC creates connections for every server object in the Sites container that has an NTDS Settings object.

The connection object is a child of the replication destination's NTDS Settings object, and the connection object references the replication source domain controller in the **fromServer** attribute on the connection object — that is, it represents the inbound half of a connection. The connection object contains a replication schedule and specifies a replication transport. The connection object schedule is derived from the site link schedule for intersite connections. For more information about intersite connection schedules, see "Connection Object Schedule" later in this section.

A connection is unidirectional; a bidirectional replication connection is represented as two inbound connection objects. The KCC creates one connection object under the NTDS Settings object of each server that is used as an endpoint for the connection.

Connection objects are created in two ways:

- Automatically by the KCC.
- Manually by a directory administrator by using Active Directory Sites and Services, ADSI Edit, or scripts.

Intersite connection objects are created by the KCC that has the role of intersite topology generator (ISTG) in the site. One domain controller in each site has this role, and the ISTG role owners in all sites use the same algorithm to collectively generate the intersite replication topology.

Ownership of Connection Objects

Tech Savvy Productions

Connections that are created automatically by the KCC are “owned” by the KCC. If you create a new connection manually, the connection is not owned by the KCC. If a connection object is not owned by the KCC, the KCC does not modify it or delete it.

Note

- One exception to this modification rule is that the KCC automatically changes the transport type of an administrator-owned connection if the **transportType** attribute is set incorrectly (see “Transport Type” later in this section).

However, if you modify a connection object that is owned by the KCC (for example, you change the connection object schedule), the ownership of the connection depends on the application that you use to make the change:

- If you use an LDAP editor such as Ldp.exe or Adsiedit.msc to change a connection object property, the KCC reverses the change the next time it runs.
- If you use Active Directory Sites and Services to change a connection object property, the object is changed from automatic to manual and the KCC no longer owns it. The UI indicates the ownership status of each connection object.

In most Active Directory deployments, manual connection objects are not needed.

If you create a connection object, it remains until you delete it, but the KCC will automatically delete duplicate KCC-owned objects if they exist and will continue to create needed connections. Ownership of a connection object does not affect security access to the object; it determines only whether the KCC can modify or delete the object.

Note

- If you create a new connection object that duplicates one that the KCC has already created, your duplicate object is created and the KCC-created object is deleted by the KCC the next time it runs.

ISTG and Modified Connections

Tech Savvy Productions

Because connection objects are stored in the configuration directory partition, it is possible for an intersite connection object to be modified by an administrator on one domain controller and, prior to replication of the initial change being received, to be modified by the KCC on another domain controller. Overwriting such a change can occur within the local site or when a connection object changes in a remote site.

By default, the KCC runs every 15 minutes. If the administrative connection object change is not received by the destination domain controller before the ISTG in the destination site runs, the ISTG in the destination site might modify the same connection object. In this case, ownership of the connection object belongs to the KCC because the latest write to the connection object is the write that is applied.

Manual Connection Objects

The KCC is designed to produce a replication topology that provides low replication latency, that adapts to failures, and that does not need modification. It is usually not necessary to create connection objects when the KCC is being used to generate automatic connections. The KCC automatically reconfigures connections as conditions change. Adding manual connections when the KCC is employed potentially increases replication traffic by adding redundant connections to the optimal set chosen by the KCC. When manually generated connections exist, the KCC uses them wherever possible.

Adding extra connections does not necessarily reduce replication latency. Within a site, latency issues are usually related to factors other than the replication topology that is generated by the KCC. Factors that affect latency include the following:

- Interruption of the service of key domain controllers, such as the primary domain controller (PDC) emulator, global catalog servers, or bridgehead servers.
- Domain controllers that are too busy to replicate in a timely manner (too few domain controllers).
- Network connectivity issues.
- DNS server problems.

Tech Savvy Productions

- Inordinate amounts of directory updates.

For problems such as these, creating a manual connection does not improve replication latency. Adjusting the scheduling and costs that are assigned to the site link is the best way to influence intersite topology.

Site Link Objects

For a connection object to be created on a destination domain controller in one site that specifies a source domain controller in another site, you must manually create a site link object (class **siteLink**) that connects the two sites. Site link objects identify the transport protocol and scheduling required to replicate between two or more sites. You can use Active Directory Sites and Services to create the site links. The KCC uses the information stored in the properties of these site links to create the intersite topology connections.

A site link is associated with a network transport by creating the site link object in the appropriate transport container (either IP or SMTP). All intersite domain replication must use IP site links. The Simple Mail Transfer Protocol (SMTP) transport can be used for replication between sites that contain domain controllers that do not host any common domain directory partition replicas.

Site Link Properties

A site link specifies the following:

- Two or more sites that are permitted to replicate with each other.
- An administrator-defined cost value associated with that replication path. The cost value controls the route that replication takes, and thus the remote sites that are used as sources of replication information.
- A schedule during which replication is permitted to occur.
- An interval that determines how frequently replication occurs over this site link during the times when the schedule allows replication.

For more information about site link properties, see “Site Link Settings and Their Effects on Intersite Replication” later in this section.

Default Site Link

Tech Savvy Productions

When you install Active Directory on the first domain controller in the forest, an object named DEFAULTIPSITELINK is created in the Sites container (in the IP container within the Inter-Site Transports container). This site link contains only one site, Default-First-Site-Name.

Site Link Bridging

By default, site links for the same IP transport that have sites in common are bridged, which enables the KCC to treat the set of associated site links as a single route. If you categorically do not want the KCC to consider some routes, or if your network is not fully routed, you can disable automatic bridging of all site links. When this bridging is disabled, you can create site link bridge objects and manually add site links to a bridge. For more information about using site link bridges, see “Bridging Site Links Manually” later in this section.

NTDS Site Settings Object

NTDS Site Settings objects (class **nTDSSiteSettings**) identify site-wide settings in Active Directory. There is one NTDS Site Settings object per site in the Sites container. NTDS Site Settings attributes control the following features and conditions:

- The identity of the ISTG role owner for the site. The KCC on this domain controller is responsible for identifying bridgehead servers. For more information about this role, see “Automated Intersite Topology Generation” later in this section.
- Whether domain controllers in the site cache membership of universal groups and the site in which to find a global catalog server for creating the cache.
- The default schedule that applies to connection objects. For more information about this schedule, see “Connection Object Schedule” later in this section.

Note

- To allow for the possibility of network failure, which might cause one or more notifications to be missed, a default schedule of once per

Tech Savvy Productions

hour is applied to replication within a site. You do not need to manage this schedule.

Cross-Reference Objects

Cross-reference objects (class **crossRef**) store the location of directory partitions in the Partitions container (CN=Partitions,CN=Configuration,DC=*ForestRootDomainName*). The contents of the Partitions container are not visible by using Active Directory Sites and Services, but can be viewed by using Adsiedit.msc to view the Configuration directory partition.

Active Directory replication uses cross-reference objects to locate the domain controllers that store each directory partition. A cross-reference object is created during Active Directory installation to identify each new directory partition that is added to the forest. Cross-reference objects store the identity (**nCName**, the distinguished name of the directory partition where "NC" stands for "naming context," a synonym for "directory partition") and location (**dNSRoot**, the DNS domain where servers that store the particular directory partition can be reached) of each directory partition.

Note

- Starting in Windows Server 2003 Active Directory, a special attribute of the cross-reference object, **msDS-NC-Replica-Locations**, identifies application directory partitions to the replication system. For more information about how application directory partitions are replicated, see "Topology Generation Phases" later in this section.

Replication Transports

Replication transports provide the wire protocols that are required for data transfer. There are three levels of connectivity for replication of Active Directory information:

- Uniform high-speed, synchronous RPC over IP within a site.
- Point-to-point, synchronous, low-speed RPC over IP between sites.
- Low-speed, asynchronous SMTP between sites.

Tech Savvy Productions

The following rules apply to the replication transports:

- Replication within a site always uses RPC over IP.
- Replication between sites can use either RPC over IP or SMTP over IP.
- Replication between sites over SMTP is supported for only domain controllers of different domains. Domain controllers of the same domain must replicate by using the RPC over IP transport. Therefore, replication between sites over SMTP is supported for only schema, configuration, and global catalog replication, which means that domains can span sites only when point-to-point, synchronous RPC is available between sites.

The Inter-Site Transports container provides the means for mapping site links to the transport that the link uses. When you create a site link object, you create it in either the IP container (which associates the site link with the RPC over IP transport) or the SMTP container (which associates the site link with the SMTP transport).

For the IP transport, a typical site link connects only two sites and corresponds to an actual WAN link. An IP site link connecting more than two sites might correspond to an asynchronous transfer mode (ATM) backbone that connects, for example, more than two clusters of buildings on a large campus or connects several offices in a large metropolitan area that are connected by leased lines and IP routers.

Synchronous and Asynchronous Communication

The RPC intersite and intrasite transport (RCP over IP within sites and between sites) and the SMTP intersite transport (SMTP over IP between sites only) correspond to synchronous and asynchronous communication methods, respectively. Synchronous communication favors fast, available connections, while asynchronous communication is better suited for slow or intermittent connections.

Synchronous Replication Over IP

The IP transport (RPC over IP) provides synchronous inbound replication. In the context of Active Directory replication, synchronous communication implies that after the destination domain controller sends the request for data, it waits for the

Tech Savvy Productions

source domain controller to receive the request, construct the reply, and send the reply before it requests changes from any other domain controllers; that is, inbound replication is sequential. Thus in synchronous transmission, the reply is received within a short time. The IP transport is appropriate for linking sites in fully routed networks.

Asynchronous Replication Over SMTP

The SMTP transport (SMTP over IP) provides asynchronous replication. In asynchronous replication, the destination domain controller does not wait for the reply and it can have multiple asynchronous requests outstanding at any particular time. Thus in asynchronous transmission, the reply is not necessarily received within a short time. Asynchronous transport is appropriate for linking sites in networks that are not fully routed and have particularly slow WAN links.

Note

- Although asynchronous replication can send multiple replication requests in parallel, the received replication packets are queued on the destination domain controller and the changes applied for only one partner and directory partition at a time.

Replication Queue

Suppose a domain controller has five inbound replication connections. As the domain controller formulates change requests, either by a schedule being reached or from a notification, it adds a work item for each request to the end of the queue of pending synchronization requests. Each pending synchronization request represents one <source domain controller, directory partition> pair, such as “synchronize the schema directory partition from DC1,” or “delete the ApplicationX directory partition.”

When a work item has been received into the queue, notification and polling intervals do not apply — the domain controller processes the item (begins synchronizing from that source) as soon as the item reaches the front of the queue, and continues until either the destination is fully synchronized with the source domain controller, an error occurs, or the synchronization is pre-empted by a higher-priority operation.

Tech Savvy Productions

SMTP Intersite Replication

When sites are on opposite ends of a WAN link (or the Internet), it is not always desirable — or even possible — to perform synchronous, RPC-based directory replication. In some cases, the only method of communication between two sites is e-mail. When connectivity is intermittent or when end-to-end IP connectivity is not available (an intermediate site does not support RPC/IP replication), replication must be possible across asynchronous, store-and-forward transports such as SMTP.

In addition, where bandwidth is limited, it can be disadvantageous to force an entire replication cycle of request for changes and transfer of changes between two domain controllers to complete before another can begin (that is, to use synchronous replication). With SMTP, several cycles can be processing simultaneously so that each cycle is being processed to some degree most of the time, as opposed to receiving no attention for prolonged periods, which can result in RPC time-outs.

For intersite replication, SMTP replication substitutes mail messaging for the RPC transport. The message syntax is the same as for RPC-based replication. There is no change notification for SMTP–based replication, and scheduling information for the site link object is used as follows:

- By default, SMTP replication ignores the **Replication Available** and **Replication Not Available** settings on the site link schedule in Active Directory Sites and Services (the information that indicates when these sites are connected). Replication occurs according to the messaging system schedule.
- Within the scope of the messaging system schedule, SMTP replication uses the replication interval that is set on the SMTP site link to indicate how often the server requests changes. The interval (**Replicate every ____ minutes**) is set in 15-minute increments on the **General** tab in site link **Properties** in Active Directory Sites and Services.

The underlying SMTP messaging system is responsible for message routing between SMTP servers.

SMTP Replication and Intersite Messaging

Tech Savvy Productions

Intersite Messaging is a component that is enabled when Active Directory is installed. Intersite Messaging allows for multiple transports to be used as add-ins to the Intersite Messaging architecture. Intersite Messaging enables messaging communication that can use SMTP servers other than those that are dedicated to processing e-mail applications.

When the forest has a functional level of at least Windows 2000, Intersite Messaging also provides services to the KCC in the form of querying the available replication paths. In addition, Net Logon queries the connectivity data in Intersite Messaging when calculating site coverage. By default, Intersite Messaging rebuilds its database once a day, or when required by a site link change.

When the forest has a functional level of at least Windows Server 2003, the KCC does not use Intersite Messaging for calculating the topology. However, regardless of forest functional level, Intersite Messaging is still required for SMTP replication, DFS, universal group membership caching, and Net Logon automatic site coverage calculations. Therefore, if any of these features are in use, do not stop Intersite Messaging.

For more information about site coverage and how automatic site coverage is calculated, see "How DNS Support for Active Directory Works." For more information about DFS, see "DFS Technical Reference."

Requirements for SMTP Replication

The KCC does not create connections that use SMTP until the following requirements are met:

- Internet Information Services (IIS) is installed on both bridgehead servers.
- An enterprise certification authority (CA) is installed and configured on your network. The CA signs and encrypts SMTP messages that are exchanged between domain controllers, ensuring the authenticity of directory updates. Specifically, a domain controller certificate must be present on the replicating domain controllers. The replication request message, which contains no directory data, is not encrypted. The replication reply message, which does contain directory data, is encrypted using a key length of 128 bits.

Tech Savvy Productions

- The sites are connected by SMTP site links.
- The site link path between the sites has a lower cost than any IP/RPC site link that can reach the SMTP site.
- You are not attempting to replicate writable replicas of the same domain (although replication of global catalog partial replicas is supported).
- Each domain controller is configured to receive mail.

You must also determine if mail routing is necessary. If the two replicating domain controllers have direct IP connectivity and can send mail to each other, no further configuration is required. However, if the two domain controllers must go through mail gateways to deliver mail to each other, you must configure the domain controller to use the mail gateway.

Note

- RPC is required for replicating the domain to a new domain controller and for installing certificates. If RPC is not available to the remote site, the domain must be replicated and certificates must be installed over RPC in a hub site and the domain controller then shipped to the remote site.

Comparison of SMTP and RPC Replication

The following characteristics apply to both SMTP and RPC with respect to Active Directory replication:

- For replication between sites, data that is replicated through either transport is compressed.
- Active Directory can respond with only a fixed (maximum) number of changes per change request, based on the size of the replication packet. The size of the replication packet is configurable. For information about configuring the replication packet size, see “Replication Packet Size” later in this section.
- Active Directory can apply a single set of changes at a time for a specific directory partition and replication partner.
- The response data (changes) are transported in one or many frames, based on the total number of changed or new values.

Tech Savvy Productions

- TCP transports the data portion by using the same algorithm for both SMTP and RPC.
- If transmission of the data portion fails, complete retransmission is necessary.

Point-to-point synchronous RPC replication is available between sites to allow the flexibility of having domains that span multiple sites. RPC is best used between sites that are connected by WAN links because it involves lower latency. SMTP is best used between sites where RPC over IP is not possible. For example, SMTP can be used by companies that have a network backbone that is not based on TCP/IP, such as companies that use an X.400 backbone.

Active Directory replication uses both transports to implement a request-response mechanism. Active Directory issues requests for changes and replies to requests for changes. RPC maps these requests into RPC requests and RPC replies. SMTP, on the other hand, actually uses long-lived TCP connections (or X.400-based message transfer agents in non-TCP/IP networks) to deliver streams of mail in each direction. Thus, RPC transport expects a response to any request immediately and can have a maximum of one active inbound RPC connection to a directory partition replica at a time. The SMTP transport expects much longer delays between a request and a response. As a result, multiple inbound SMTP connections to a directory partition replica can be active at the same time, provided the requests are all for a different source domain controller or, for the same source domain controller, a different directory partition. For more information, see “Synchronous and Asynchronous Communication” earlier in this section.

Replication Packet Size

Replication packet sizes are computed on the basis of memory size unless you have more than 1 gigabyte (GB). By default, the system limits the packet size as follows:

- The packet size in bytes is 1/100th the size of RAM, with a minimum of 1 MB and a maximum of 10 MB.
- The packet size in objects is 1/1,000,000th the size of RAM, with a minimum of 100 objects and a maximum of 1,000 objects. For general estimates

Tech Savvy Productions

when this entry is not set, assume an approximate packet size of 100 objects.

There is one exception: the value of the **Replicator async inter site packet size (bytes)** registry entry is always 1 MB if it is not set (that is, when the default value is in effect). Many mail systems limit the amount of data that can be sent in a mail message (2 MB to 4 MB is common), although most Windows-based mail systems can handle large 10-MB mail messages.

Overriding these memory-based values might be beneficial in advanced bandwidth management scenarios. You can edit the registry to set the maximum packet size.

Note

- If you must edit the registry, use extreme caution. Registry information is provided here as a reference for use by only highly skilled directory service administrators. It is recommended that you do not directly edit the registry unless, as in this case, there is no Group Policy or other Windows tools to accomplish the task. Modifications to the registry are not validated by the registry editor or by Windows before they are applied, and as a result, incorrect values can be stored. Storage of incorrect values can result in unrecoverable errors in the system.

Setting the maximum packet size requires adding or modifying entries in the following registry path with the REG_DWORD data type: HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\NTDS\Parameters. These entries can be used to determine the maximum number of objects per packet and maximum size of the packets. The minimum values are indicated as the lowest value in the range.

- For RPC replication within a site:
    - **Replicator intra site packet size (objects)**

Range: >=2

    - **Replicator intra site packet size (bytes)**

Range: >=10 KB

Tech Savvy Productions

- For RPC replication between sites:
    - **Replicator inter site packet size (objects)**

Range: >=2

    - **Replicator inter site packet size (bytes)**

Range: >=10 KB

- For SMTP replication between sites:
    - **Replicator async inter site packet size (objects)**

Range: >=2

    - **Replicator async inter site packet size (bytes)**

Range: >=10 KB

Transport Type

The **transportType** attribute of a connection object specifies which network transport is used when the connection is used for replication. The transport type receives its value from the distinguished name of the container in the configuration directory partition that contains the site link over which the connection occurs, as follows:

- Connection objects that use TCP/IP have the **transportType** value of CN=IP,CN=Inter-Site Transports,CN=IP,DC=Configuration,DC=*ForestRootDomainName*.
- Connection objects that use SMTP/IP have the **transportType** value of CN=SMTP,CN=Inter-Site Transports,CN=IP,DC=Configuration,DC=*ForestRootDomainName*.
- For intrasite connections, **transportType** has no value; Active Directory Sites and Services shows the transport of “RPC” for connections that are from servers in the same site.

If you move a domain controller to a different site, the connection objects from servers in the site from which it was moved remain, but the transport type is blank because it was an intrasite connection. Because the connection has an

Tech Savvy Productions

endpoint outside of the site, the local KCC in the server's new site does not manage the connection. When the ISTG runs, if a blank transport type is found for a connection that is from a server in a different site, the **transportType** value is automatically changed to IP. The ISTG in the site determines whether to delete the connection object or to retain it, in which case the server becomes a bridgehead server in its new site.

Replication Between Sites

Replication between sites transfers domain updates when domain controllers for a domain are located in more than one site. Intersite replication of configuration and schema changes is always required when more than one site is configured in a forest. Replication between sites is accomplished by bridgehead servers, which replicate changes according to site link settings.

Bridgehead Servers

When domain controllers for the same domain are located in different sites, at least one bridgehead server per directory partition and per transport (IP or SMTP) replicates changes from one site to a bridgehead server in another site. A single bridgehead server can serve multiple partitions per transport and multiple transports. Replication within the site allows updates to flow between the bridgehead servers and the other domain controllers in the site. Bridgehead servers help to ensure that the data replicated across WAN links is not stale or redundant.

Any server that has a connection object with a "from" server in another site is acting as a destination bridgehead. Any server that is acting as a source for a connection to another site acts as a source bridgehead.

Note

- You can identify a KCC-selected bridgehead server in Active Directory Sites and Services by viewing connection objects for the server (select the NTDS Settings object below the server object); if there are connections from servers in a different site or sites, the server represented by the selected NTDS Settings object is a bridgehead server. If you have Windows Support Tools installed, you can see all bridgehead servers by using the command **repadmin /bridgeheads**.

Tech Savvy Productions

KCC selection of bridgehead servers guarantees bridgehead servers that are capable of replicating all directory partitions that are needed in the site, including partial global catalog partitions. By default, bridgehead servers are selected automatically by the KCC on the domain controller that holds the ISTG role in each site. If you want to identify the domain controllers that can act as bridgehead servers, you can designate preferred bridgehead servers, from which the ISTG selects all bridgehead servers. Alternatively, if the ISTG is not used to generate the intersite topology, you can create manual intersite connection objects on domain controllers to designate bridgehead servers.

In sites that have at least one domain controller that is running Windows Server 2003, the ISTG can select bridgehead servers from all eligible domain controllers for each directory partition that is represented in the site. For example, if three domain controllers in a site store replicas of the same domain and domain controllers for this domain are also located in three or more other sites, the ISTG can spread the inbound connection objects from those sites among all three domain controllers, including those that are running Windows 2000 Server.

In Windows 2000 forests, a single bridgehead server per directory partition and per transport is designated as the bridgehead server that is responsible for intersite replication of that directory partition. Therefore, for the preceding example, only one of the three domain controllers would be designated by the ISTG as a bridgehead server for the domain, and all four connection objects from the four other sites would be created on the single bridgehead server. In large hub sites, a single domain controller might not be able to adequately respond to the volume of replication requests from perhaps thousands of branch sites.

For more information about how the KCC selects bridgehead servers, see “Bridgehead Server Selection” later in this section.

Compression of Replication Data

Intersite replication is compressed by default. Compressing replication data allows the data to be transferred over WAN links more quickly, thereby conserving network bandwidth. The cost of this benefit is an increase in CPU utilization on bridgehead servers.

By default, replication data is compressed under the following conditions:

Tech Savvy Productions

- Replication of updates between domain controllers in different sites.
- Replication of Active Directory to a newly created domain controller.

A new compression algorithm is employed by bridgehead servers that are running at least Windows Server 2003. The new algorithm improves replication speed by operating between two and ten times faster than the Windows 2000 Server algorithm.

Windows 2000 Server Compression

The compression algorithm that is used by domain controllers that are running Windows 2000 Server achieves a compression ratio of approximately 75% to 85%. The cost of this compression in terms of CPU utilization can be as high as 50% for intersite Active Directory replication. In some cases, the CPUs on bridgehead servers that are running Windows 2000 Server can become overwhelmed with compression requests, compounded by the need to service outbound replication partners. In a worst case scenario, the bridgehead server becomes so overloaded that it cannot keep up with outbound replication. This scenario is usually coupled with a replication topology issue where a domain controller has more outbound partners than necessary or the replication schedule was overly aggressive for the number of direct replication partners.

Note

- If a bridgehead server has too many replication partners, the KCC logs event ID 1870 in the Directory Service log, indicating the current number of partners and the recommended number of partners for the domain controller.

Windows Server 2003 Compression

On domain controllers that are running Windows Server 2003, compression quality is comparable to Windows 2000 but the processing burden is greatly decreased. The Windows Server 2003 algorithm produces a compression ratio of approximately 60%, which is slightly less compression than is achieved by the Windows 2000 Server ratio, but which significantly reduces the processing load on bridgehead servers. The new compression algorithm provides a good compromise by significantly reducing the CPU load on bridgehead servers, while only slightly

Tech Savvy Productions

increasing the WAN traffic. The new algorithm reduces the time taken by compression from approximately 60% of replication time to 20%.

The Windows Server 2003 compression algorithm is used only when both bridgehead servers are running Windows Server 2003. If a bridgehead server that is running Windows Server 2003 replicates with a bridgehead server that is running Windows 2000 Server, then the Windows 2000 compression algorithm is used.

Reverting to Windows 2000 Compression

For slow WAN links (for example, 64 KB or less), if more compression is preferable to a decrease in computation time, you can change the compression algorithm to the Windows 2000 algorithm. The compression algorithm is controlled by the REG_DWORD registry entry HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NTDS\Parameters**Replicator compression algorithm**. By editing this registry entry, you can change the algorithm that is used for compression to the Windows 2000 algorithm.

Note

- If you must edit the registry, use extreme caution. Registry information is provided here as a reference for use by only highly skilled directory service administrators. It is recommended that you do not directly edit the registry unless, as in this case, there is no Group Policy or other Windows tools to accomplish the task. Modifications to the registry are not validated by the registry editor or by Windows before they are applied, and as a result, incorrect values can be stored. Storage of incorrect values can result in unrecoverable errors in the system.

The default value is 3, which indicates that the Windows Server 2003 algorithm is in effect. By changing the value to 2, the Windows 2000 algorithm is used for compression. However, switching to the Windows 2000 algorithm is not recommended unless both bridgehead domain controllers serve relatively few branches and have ample CPU (for example, > dual processor 850 megahertz [MHz]).

Site Link Settings and Their Effects on Intersite Replication

Tech Savvy Productions

In Active Directory Sites and Services, the **General** tab of the site link **Properties** contains the following options for configuring site links to control the replication topology:

- A list of two or more sites to be connected.
- A single numeric cost that is associated with communication over the link. The default cost is 100, but you can assign higher cost values to represent more expensive transmission. For example, sites that are connected by low-speed or dial-up connections would have high-cost site links between them. Sites that are well connected through backbone lines would have low-cost site links. Where multiple routes or transports exist between two sites, the least expensive route and transport combination is used.
- A schedule that determines days and hours during which replication can occur over the link (the link is available). For example, you might use the default (100 percent available) schedule on most links, but block replication traffic during peak business hours on links to certain branches. By blocking replication, you give priority to other traffic, but you also increase replication latency.

Note

  - Scheduling information is ignored by site links that use SMTP transports; the mail is stockpiled and then exchanged at the times that are configured for your mail infrastructure.

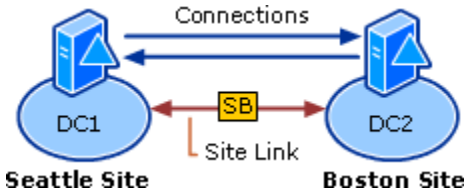
- An interval in minutes that determines how often replication can occur (default is every 180 minutes, or 3 hours). The minimum interval is 15 minutes. If the interval exceeds the time allowed by the schedule, replication occurs once at the scheduled time.

A site can be connected to other sites by any number of site links. For example, a hub site has site links to each of its branch sites. Each site that contains a domain controller in a multisite directory must be connected to at least one other site by at least one site link; otherwise, it cannot replicate with domain controllers in any other site.

Tech Savvy Productions

The following diagram shows two sites that are connected by a site link. Domain controllers DC1 and DC2 belong to the same domain and are acting as partner bridgehead servers. When topology generation occurs, the ISTG in each site creates an inbound connection object on the bridgehead server in its site from the bridgehead server in the opposite site. With these objects in place, replication can occur according to the settings on the SB site link.

Connections Between Domain Controllers in Two Sites that Are Connected by a Site Link



Site Link Cost

The ISTG uses the cost settings on site links to determine the route of replication between three or more sites that replicate the same directory partition. The default cost value on a site link object is 100. You can assign lower or higher cost values to site links to favor inexpensive connections over expensive connections, respectively. Certain applications and services, such as domain controller Locator and DFS, also use site link cost information to locate nearest resources. For example, site link cost can be used to determine which domain controller is contacted by clients located in a site that does not include a domain controller for the specified domain. The client contacts the domain controller in a different site according to the site link that has the lowest cost assigned to it.

Cost is usually assigned not only on the basis of the total bandwidth of the link, but also on the availability, latency, and monetary cost of the link. For example, a 128-kilobits per second (Kbps) permanent link might be assigned a lower cost than a dial-up 128-Kbps dual ISDN link because the dial-up ISDN link has replication latency-producing delay that occurs as the links are being established or removed. Furthermore, in this example, the permanent link might have a fixed monthly cost, whereas the ISDN line is charged according to actual usage. Because the company is paying up-front for the permanent link, the administrator might assign a lower

Tech Savvy Productions

cost to the permanent link to avoid the extra monetary cost of the ISDN connections.

The method used by the ISTG to determine the least-cost path from each site to every other site for each directory partition is more efficient when the forest has a functional level of at least Windows Server 2003 than it is at other levels. For more information about how the KCC computes replication routes, see "Automated Intersite Topology Generation" later in this section. For more information about domain controller location, see "How DNS Support for Active Directory Works."

Transitivity and Automatic Site Link Bridging

By default, site links are transitive, or "bridged." If site A has a common site link with site B, site B also has a common site link with Site C, and the two site links are bridged, domain controllers in site A can replicate directly with domain controllers in site C under certain conditions, even though there is no site link between site A and site C. In other words, the effect of bridged site links is that replication between sites in the bridge is transitive.

The setting that implements automatic site link bridges is **Bridge all site links**, which is found in Active Directory Sites and Services in the properties of the IP or SMTP intersite transport containers. The default bridging of site links occurs automatically and no directory object represents the default bridge. Therefore, in the common case of a fully routed IP network, you do not need to create any site link bridge objects.

Transitivity and Rerouting

For a set of bridged site links, where replication schedules in the respective site links overlap (replication is available on the site links during the same time period), connection objects can be automatically created, if needed, between sites that do not have site links that connect them directly. All site links for a specific transport implicitly belong to a single site link bridge for that transport.

Site link transitivity enables the KCC to re-route replication when necessary. In the next diagram, a domain controller that can replicate the domain is not available in Seattle. In this case, because the site links are transitive (bridged) and the schedules on the two site links allow replication at the same time, the KCC can

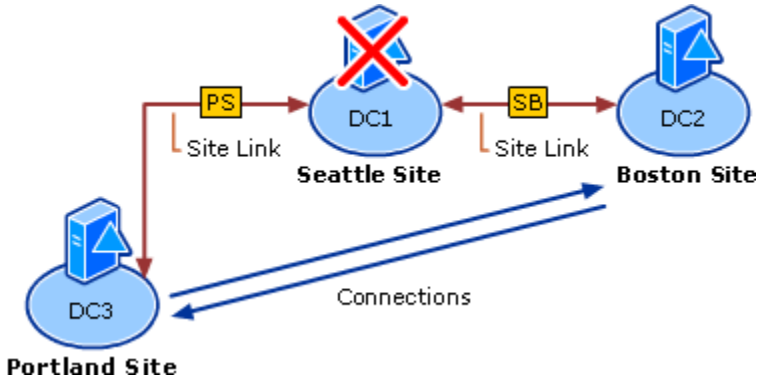
Tech Savvy Productions

re-route replication by creating connections between DC3 in Portland and DC2 in Boston. Connections between domain controllers in Portland and Boston might also be created when a domain controller in Portland is a global catalog server, but no global catalog server exists in the Seattle site and the Boston site hosts a domain that is not present in the Seattle site. In this case, connections can be created between Portland and Boston to replicate the global catalog partial, read-only replica.

Note

- Overlapping schedules are required for site link transitivity, even when **Bridge all site links** is enabled. In the example, if the site link schedules for SB and PS do not overlap, no connections are possible between Boston and Portland.

Transitive Replication when Site Links Are Bridged, Schedules Overlap, and Replication Must Be Rerouted



In the preceding diagram, creating a third site link to connect the Boston and Portland sites is unnecessary and counterproductive because of the way that the KCC uses cost to route replication. In the configuration that is shown, the KCC uses cost to choose either the route between Portland and Seattle or the route between Portland and Boston. If you wanted the KCC to use the route between Portland and Boston, you would create a site link between Portland and Boston instead of the site link between Portland and Seattle.

Aggregated Site Link Cost and Routing

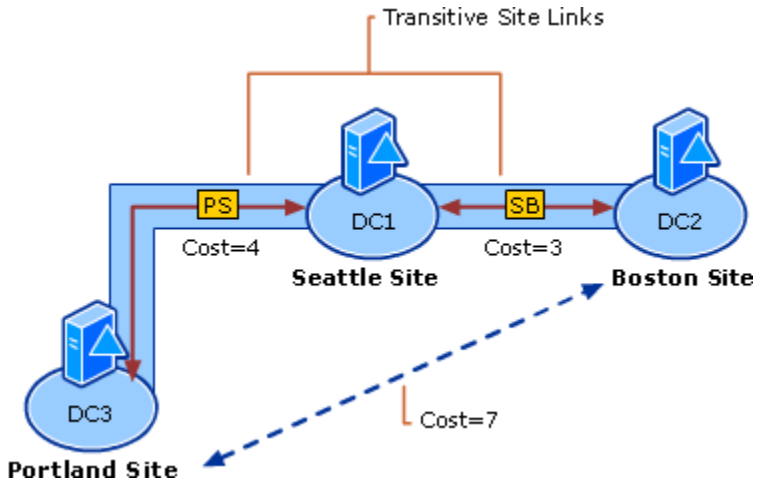
When site links are bridged, the cost of replication from a domain controller at one end of the bridge to a domain controller at the other end is the sum of the

Tech Savvy Productions

costs on each of the intervening site links. For this reason, if a domain controller in an interim site stores the directory partition that is being replicated, the KCC will route replication to the domain controller in the interim site rather than to the more distant site. The domain controller in the more distant site in turn receives replication from the interim site (store-and-forward replication). If the schedules of the two site links overlap, this replication occurs in the same period of replication latency.

The following diagram illustrates an example where two site links connecting three sites that host the same domain are bridged automatically (**Bridge all site links** is enabled). The aggregated cost of directly replicating between Portland and Boston illustrates why the KCC routes replication from Portland to Seattle and from Seattle to Boston in a store-and-forward manner. Given the choice between replicating at a cost of 4 from Seattle or a cost of 7 from Boston, the ISTG in Portland chooses the lower cost and creates the connection object on DC3 from DC1 in Seattle.

Bridged Site Links Routing Replication Between Three Sites According to Cost



In the preceding diagram, if DC3 in Portland needs to replicate a directory partition that is hosted on DC2 in Boston but not by any domain controller in Seattle, or if the directory partition is hosted in Seattle but the Seattle site cannot be reached, the ISTG creates the connection object from DC2 to DC3.

Significance of Overlapping Schedules

Tech Savvy Productions

In the preceding diagram, to replicate the same domain that is hosted in all three sites, the Portland site replicates directly with Seattle and Seattle replicates directly with Boston, transferring Portland's changes to Boston, and vice versa, through store-and-forward replication. Whether the schedules overlap has the following effects:

- PS and SB site link schedules have replication available during at least one common hour of the schedule:
    - Replication between these two sites occurs in the same period of replication latency, being routed through Seattle.
    - If Seattle is unavailable, connections can be created between Portland and Boston.
- PS and SB site link schedules have no common time:
    - Replication of changes between Portland and Boston reach their destination in the next period of replication latency after reaching Seattle.
    - If Seattle is unavailable, no connections are possible between Portland and Boston.

Note

If **Bridge all site links** is disabled, a connection is never created between Boston and Portland, regardless of schedule overlap, unless you manually create a site link bridge.

Site Link Changes and Replication Path

The path that replication takes between sites is computed from the information that is stored in the properties of the site link objects. When a change is made to a site link setting, the following events must occur before the change takes effect:

- The site link change must replicate to the ISTG of each site by using the previous replication topology.
- The KCC must run on each ISTG.

Tech Savvy Productions

As the path of connections is transitively figured through a set of site links, the attributes (settings) of the site link objects are combined along the path as follows:

- Costs are added together.
- The replication interval is the maximum of the intervals that are set for the site links along the path.
- The options, if any are set, are computed by using the AND operation.

Note

  - Options are the values of the **options** attribute on the site link object. The value of this attribute determines special behavior of the site link, such as reciprocal replication and intersite change notification.

Thus the site link schedule is the overlap of all of the schedules of the subpaths. If none of the schedules overlap, the path is not used.

Bridging Site Links Manually

If your IP network is composed of IP segments that are not fully routed, you can disable **Bridge all site links** for the IP transport. In this case, all IP site links are considered nontransitive, and you can create and configure site link bridge objects to model the actual routing behavior of your network. A site link bridge has the effect of providing routing for a disjoint network (networks that are separate and unaware of each other). When you add site links to a site link bridge, all site links within the bridge can route transitively.

A site link bridge object represents a set of site links, all of whose sites can communicate through some transport. Site link bridges are necessary if both of the following conditions are true:

- A site contains a domain controller that hosts a domain directory partition that is not hosted by a domain controller in an adjacent site (a site that is in the same site link).
- That domain directory partition is hosted on a domain controller in at least one other site in the forest.

Tech Savvy Productions

Note

- Site link bridge objects are used by the KCC only when the **Bridge all site links** setting is disabled. Otherwise, site link bridge objects are ignored.

Site link bridges can also be used to diminish potentially high CPU overhead of generating a large transitive replication topology. In very large networks, transitive site links can be an issue because the KCC considers every possible connection in the bridged network, and selects only one. Therefore, in a Windows 2000 forest that has a very large network or a Windows Server 2003 or higher forest that consists of an extremely large hub-and-spoke topology, you can reduce KCC-related CPU utilization and run time by turning off **Bridge all site links** and creating manual site link bridges only where they are required.

Note

- Turning off **Bridge all site links** affects the ability of DFS clients to locate DFS servers in the closest site. If the ISTG is running at least Windows Server 2003, the **Bridge all site links** must be enabled to generate the intersite cost matrix that DFS requires for its site-costing functionality. If the ISTG is running at least Windows Server 2003 with Service Pack 1 (SP1), you can enable **Bridge all site links** and then run the **repadmin /siteoptions W2K3_BRIDGES_REQUIRED** command on each site where you need to accommodate the DFS site-costing functionality. This command disables automatic site link bridging for the KCC but allows default Intersite Messaging options to enable the site-costing calculation to occur for DFS. For more information about turning off this functionality while accommodating DFS, see "DFS Site Costing and Windows Server 2003 SP1 Site Options" later in this section. For more information about site link cost and DFS, see “DFS Technical Reference.”

You create a site link bridge object for a specific transport by specifying two or more site links for the specified transport.

Requirements for manual site link bridges

Each site link in a manual site link bridge must have at least one site in common with another site link in the bridge. Otherwise, the bridge cannot compute the cost from sites in one link to the sites in other links of the bridge. If bridgehead

Tech Savvy Productions

servers that are capable of the transport that is used by the site link bridge are not available in two linked sites, a route is not available.

Manual site link bridge behavior

Separate site link bridges, even for the same transport, are independent. To illustrate this independence, consider the following conditions:

- Four sites have domain controllers for the same domain: Portland, Seattle, Detroit, and Boston.
- Three site links are configured: Portland-Seattle (PS), Seattle-Detroit (SD), and Detroit-Boston (DB).
- Two separate manual site link bridges link the outer site links PS and DB with the inner site link SD.

The presence of the PS-SD site link bridge means that an IP message can be sent transitively from the Portland site to the Detroit site with cost 4 + 3 = 7. The presence of the SD-DB site link bridge means that an IP message can be sent transitively from Seattle to Boston at a cost of 3 + 2 = 5. However, because there is no transitivity between the PS-SB and SB-DB site link bridges, an IP message cannot be sent between Portland and Boston with cost 4 + 3 + 2 = 9, or at any cost.

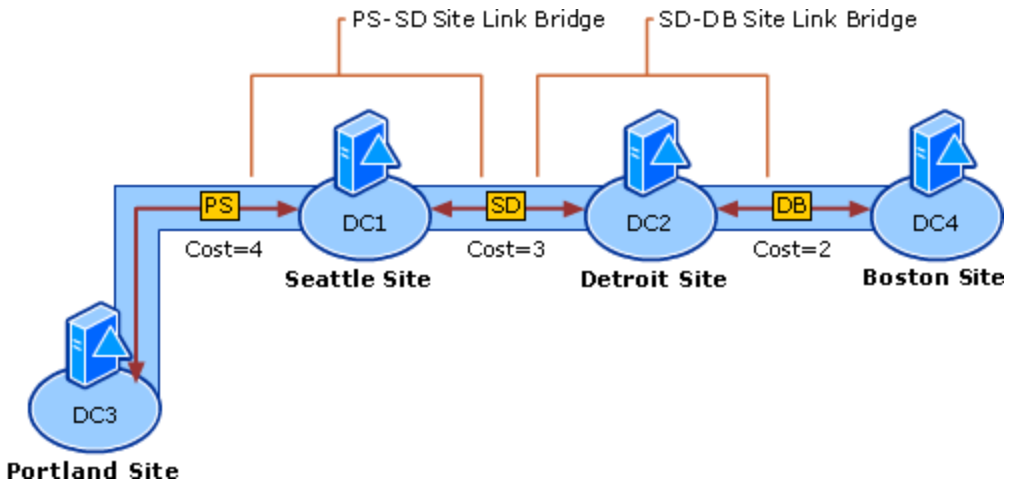
In the following diagram, the two manual site link bridges means that Boston is able to replicate directly only with Detroit and Seattle, and Portland is able to replicate directly only with Seattle and Detroit.

Note

- If you need direct replication between Portland and Detroit, you can create the PS-SB-DB site link bridge. By excluding the PS site link, you ensure that connections are neither created nor considered by the KCC between Portland and Detroit.

Two Site Link Bridges that Are Not Transitive

Tech Savvy Productions



In the diagram, connection objects are not possible between DC4 in Detroit and DC3 in Portland because two site link bridges are not transitive. For connection objects to be possible between DC3 and DC4, the site link DB must be added to the PS-SB site link bridge. In this case, the cost of replication between DC3 and DC4 is 9.

Note

- Cost is applied differently to a site link bridge than to a site link that contains more than two sites. To use the preceding example, if Seattle, Boston, and Portland are all in the same site link, the cost of replication between any of the two sites is the same.

Bridging site links manually is generally recommended for only large branch office deployments. For more information about using manual site link bridging, see the "Windows Server 2003 Active Directory Branch Office Deployment Guide."

Site Link Schedule

Replication using the RPC transport between sites is scheduled. The schedule specifies one or many time periods during which replication can occur. For example, you might schedule a site link for a dial-up line to be available during off-peak hours (when telephone rates are low) and unavailable during high-cost regular business hours. The **schedule** attribute of the site link object specifies the availability of the site link. The default setting is that replication is always available.

Note

Tech Savvy Productions

- The **Ignore schedules** setting on the IP container is equivalent to replication being always available. If **Ignore schedules** is selected, replication occurs at the designated intervals but ignores any schedule.

If replication goes through multiple site links, there must be at least one common time period (overlap) during which replication is available; otherwise, the connection is treated as not available. For example, if site link AB has a schedule of 18:00 hours to 24:00 hours and site link BC has a schedule of 17:00 hours to 20:00 hours, the resulting overlap is 18:00 hours through 20:00 hours, which is the intersection of the schedules for site link AB and site link BC. During the time in which the schedules overlap, replication can occur from site A to site C even if a domain controller in the intermediate site B is not available. If the schedules do not overlap, replication from the intermediate site to the distant site continues when the next replication schedule opens on the respective site link.

Note

- Cost considerations also affect whether connections are created. However, if the site link schedules do not overlap, the cost is irrelevant.

Scheduling across time zones

When scheduling replication across time zones, consider the time difference to ensure that replication does not interfere with peak production times in the destination site.

Domain controllers store time in Coordinated Universal Time (UTC). When viewed through the Active Directory Sites and Services snap-in, time settings in site link object schedules are displayed according to the local time of the computer on which the snap-in is being run. However, replication occurs according to UTC.

For example, suppose Seattle adheres to Pacific Standard Time (PST) and Japan adheres to Japan Standard Time (JST), which is 17 hours later. If a schedule is set on a domain controller in Seattle and the site link on which the schedule is set connects Seattle and Tokyo, the actual time of replication in Tokyo is 17 hours later.

If the schedule is set to begin replication at 10:00 PM PST in Seattle, the conversion can be computed as follows:

Tech Savvy Productions

- Convert 10:00 PM PST to 22:00 PST military time.
- Add 8 hours to arrive at 06:00 UTC, the following day.
- Add 9 hours to arrive at 15:00 JST.
- 15:00 JST converts to 3:00 PM.

Thus, when replication begins at 10:00 o'clock at night in Seattle, it is occurring in Tokyo at 3:00 o'clock in the afternoon the following day. By scheduling replication a few hours later in Seattle, you can avoid replication occurring during working hours in Japan.

Schedule implementation

The times that you can set in the **Schedule** setting on the site link are in one-hour increments. For example, you can schedule replication to occur between 00:00 hours and 01:00 hours, between 01:00 hours and 02:00 hours, and so forth. However, each block in the actual connection schedule is 15 minutes. For this reason, when you set a schedule of 01:00 hours to 02:00 hours, you can assume that replication is queued at some point between 01:00 hours and 01:14:59 hours.

Note

- RPC synchronous inbound replication is serialized so that if the server is busy replicating this directory partition from another source, replication from a different source does not begin until the first synchronization is complete. SMTP asynchronous replication is processed serially by order of arrival, with multiple replication requests queued simultaneously.

Specifically, a replication event is queued at time *t *+ *n*, where *t* is the replication interval that is applied across the schedule and *n* is a pseudo-random number from 1 minute through 15 minutes. For example, if the site link indicates that replication can occur from 02:00 hours through 07:00 hours, and the replication interval is 2 hours (120 minutes), *t* is 02:00 hours, 04:00 hours, and 06:00 hours. A replication event is queued on the destination domain controller between 02:00 hours and 02:14:59 hours, and another replication event is queued between 04:00 hours and 04:14:59 hours. Assuming that the first replication event that was queued is complete, another replication event is queued between 06:00 hours

Tech Savvy Productions

and 06:14:59 hours. If the synchronization took longer than two hours, the second synchronization would be ignored because an event is already in the queue.

Replication can extend beyond the end of the schedule. A period of replication latency that starts before the end of the schedule runs until completion, even if the period is still running when the schedule no longer allows replication to be available.

Note

- The replication queue is shared with other events, and the time at which replication takes place is approximate. Duplicate replication events are not queued for the same directory partition and transport.

Connection object schedule

Each connection object has a schedule that controls when (during what hours) and how frequently (how many times per hour) replication can occur:

- None (no replication)
- Once per hour (the default setting)
- Twice per hour
- Four times per hour

The connection object schedule and interval are derived from one of two locations, depending on whether it is an intrasite or intersite connection:

- Intrasite connections inherit a default schedule from the **schedule** attribute of the NTDS Site Settings object. By default, this schedule is always available and has an interval of one hour.
- Intersite connections inherit the schedule and interval from the site link.

Although intrasite replication is prompted by changes, intrasite connection objects inherit a default schedule so that replication occurs periodically, regardless of whether change notification has been received. The connection object schedule ensures that intrasite replication occurs if a notification message is lost, or if notification does not take place, due to network problems or a domain controller becomes unavailable. The NTDS Site Settings schedule has a minimum replication

Tech Savvy Productions

interval of 15 minutes. This minimum replication interval is not configurable and determines the smallest interval that is possible for both intrasite and intersite replication (on a connection object or a site link, respectively).

For intersite replication, the schedule is configured on the site link object, but the connection object schedule actually determines replication; that is, the connection object schedule for an intersite connection is derived from the site link schedule, which is applied through the connection object schedule. Scheduled replication occurs independently of change notification.

Note

- You do not need to configure the connection object schedule unless you are creating a manual intersite replication topology that does not use the KCC automatic connection objects.

The KCC uses a two-step process to compute the schedule of an intersite connection.

1. The schedules of the site links traversed by a connection are merged together.
2. This merged schedule is modified so that it is available at only certain periods. The length of those periods is equal to the maximum replication interval of the site links traversed by this connection.

By using Active Directory Sites and Services, you can manually revise the schedule on a connection object, but such an override is effective for only administrator-owned connection objects.

Replication Interval

For each site link object, you can specify a value for the replication interval (frequency), which determines how often replication occurs over the site link during the time that the schedule allows. For example, if the schedule allows replication between 02:00 hours and 04:00 hours, and the replication interval is set for 30 minutes, replication can occur up to four times during the scheduled time.

Tech Savvy Productions

The default replication interval is 180 minutes, or 3 hours. When the KCC creates a connection between a domain controller in one site and a domain controller in another site, the replication interval of the connection is the maximum interval along the minimum-cost path of site link objects from one end of the connection to the other.

Interaction of Replication Schedule and Interval

When multiple site links are required to complete replication for all sites, the replication interval settings on each site link combine to affect the entire length of the connection between sites. In addition, when schedules on each site link are not identical, replication can occur only when the schedules overlap.

Suppose that site A and site B have site link AB, and site B and site C have site link BC. When a domain controller in site A replicates with a domain controller in site C, it can do so only as often as the maximum interval that is set for site link AB and site link BC allows. The following table shows the site link settings that determine how often and during what times replication can occur between domain controllers in site A, site B, and site C.

Replication Interval and Schedule Settings for Two Site Links

Site Link	Replication Interval	Schedule
AB	30 minutes	12:00 hours to 04:00 hours
BC	60 minutes	01:00 hours to 05:00 hours

Given these settings, a domain controller in site A can replicate with a domain controller in site B according to the AB site link schedule and interval, which is once every 30 minutes between the hours of 12:00 and 04:00. However, assuming that there is no site link AC, a domain controller in site A can replicate with a domain controller in site C between the hours of 01:00 and 04:00, which is where the schedules on the two site links intersect. Within that timespan, they can replicate once every 60 minutes, which is the greater of the two replication intervals.

KCC and Topology Generation

Tech Savvy Productions

The Knowledge Consistency Checker (KCC) is a dynamic-link library (DLL) that runs as a distributed application on every domain controller. The KCC on each domain controller modifies data in its local instance of the directory in response to forest-wide changes, which are made known to the KCC by changes to data in the configuration directory partition.

The KCC generates and maintains the replication topology for replication within sites and between sites by converting KCC-defined and administrator-defined (if any) connection objects into a configuration that is understood by the directory replication engine. By default, the KCC reviews and makes modifications to the Active Directory replication topology every 15 minutes to ensure propagation of data, either directly or transitively, by creating and deleting connection objects as needed. The KCC recognizes changes that occur in the environment and ensures that domain controllers are not orphaned in the replication topology.

Operating independently, the KCC on each domain controller uses its own view of the local replica of the configuration directory partition to arrive at the same intrasite topology. One KCC per site, the ISTG, determines the intersite replication topology for the site. Like the KCC that runs on each domain controller within a site, the instances of the ISTG in different sites do not communicate with each other. They independently use the same algorithm to produce a consistent, well-formed spanning tree of connections. Each site constructs its own part of the tree and, when all have run, a working replication topology exists across the enterprise.

The predictability of all KCCs allows scalability by reducing communication requirements between KCC instances. All KCCs agree on where connections will be formed, ensuring that redundant replication does not occur and that all parts of the enterprise are connected.

The KCC performs two major functions:

- Configures appropriate replication connections (connection objects) on the basis of the existing cross-reference, server, NTDS settings, site, site link, and site link bridge objects and the current status of replication.
- Converts the connection objects that represent inbound replication to the local domain controller into the replication agreements that are actually

Tech Savvy Productions

used by the replication engine. These agreements, called replica links, accommodate replication of a single directory partition from the source to the destination domain controller.

Intervals at Which the KCC Runs

By default, the KCC runs its first replication topology check five minutes after the domain controller starts. The domain controller then attempts initial replication with its intrasite replication partners. If a domain controller is being used for multiple other services, such as DNS, WINS, or DHCP, extending the replication topology check interval can ensure that all services have started before the KCC begins using CPU resources.

You can edit the registry to modify the interval between startup and the time the domain controller first checks the replication topology.

Note

- If you must edit the registry, use extreme caution. Registry information is provided here as a reference for use by only highly skilled directory service administrators. It is recommended that you do not directly edit the registry unless, as in this case, there is no Group Policy or other Windows tools to accomplish the task. Modifications to the registry are not validated by the registry editor or by Windows before they are applied, and as a result, incorrect values can be stored. Storage of incorrect values can result in unrecoverable errors in the system.

Modifying the interval between startup and the time the domain controller first checks the replication topology requires changing the **Repl topology update delay (secs)** entry in HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NTDS\Parameters as appropriate:

- **Value:** Number of seconds to wait between the time Active Directory starts and the KCC runs for the first time.
- **Default:** 300 seconds (5 minutes)
- **Data type:** REG_DWORD

Tech Savvy Productions

Thereafter, as long as services are running, the KCC on each domain controller checks the replication topology every 15 minutes and makes changes as necessary.

Modifying the interval at which the KCC performs topology review requires changing the **Repl topology update period (secs)** entry in HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NTDS\Parameters as appropriate:

- **Value:** Number of seconds between KCC topology updates
- **Default:** 900 seconds (15 minutes)
- **Data type:** REG_DWORD

Objects that the KCC Requires to Build the Replication Topology

The following objects, which are stored in the configuration directory partition, provide the information required by the KCC to create the replication topology:

- **Cross-reference.** Each directory partition in the forest is identified in the Partitions container by a cross-reference object. The attributes of this object are used by the replication system to locate the domain controllers that store each directory partition.
- **Server.** Each domain controller in the forest is identified as a server object in the Sites container.
- **NTDS Settings.** Each server object that represents a domain controller has a child NTDS Settings object. Its presence identifies the server as having Active Directory installed. The NTDS Settings object must be present for the server to be considered by the KCC for inclusion in the replication topology.
- **Site.** The presence of the above objects also indicates to the KCC the site in which each domain controller is located for replication. For example, the distinguished name of the NTDS Settings object contains the name of the site in which the server object that represents the domain controller exists.
- **Site link.** A site link must be available between any set of sites and its schedule and cost properties evaluated for routing decisions.

Tech Savvy Productions

- **Site link bridge.** If they exist, site link bridge objects and properties are evaluated for routing decisions.

If the domain controller is physically located in one site but its server object is configured in a different site, the domain controller will attempt intrasite replication with a replication partner that is in the site of its server object. In this scenario, the improper configuration of servers in sites can affect network bandwidth.

If a site object exists for a site that has no domain controllers, the KCC does not consider the site when generating the replication topology.

Topology Generation Phases

The KCC generates the replication topology in two phases:

- **Evaluation.** During the evaluation phase, the KCC evaluates the current topology, determines whether replication failures have occurred with the existing connections, and constructs whatever new connection objects are required to complete the replication topology.
- **Translation.** During the translation phase, the KCC implements, or "translates," the decisions that were made during the evaluation phase into agreements between the replication partners. During this phase, the KCC writes to the **repsFrom** attribute on the local domain controller (for intrasite topology) or on all bridgehead servers in a site (for intersite topology) to identify the replication partners from which each domain controller pulls replication. For more information about the information in the replication agreement, see "How the Active Directory Replication Model Works."

KCC Modes and Scopes

Because individual KCCs do not communicate directly to generate the replication topology, topology generation occurs within the scope of either a single domain controller or a single site. In performing the two topology generation phases, the KCC has three modes of operation. The following table identifies the modes and scope for each mode.

Modes and Scopes of KCC Topology Generation

Tech Savvy Productions

KCC Mode	Performing Domain Controllers	Scope	Description
Intrasite	All	Local server	Evaluate all servers in a site and create connection objects locally on this server from servers in the same site that are adjacent to this server in the ring topology.
Intersite	One domain controller per site that has the ISTG role	Local site	Evaluate the servers in all sites and create connection objects both locally and on other servers in the site from servers in different sites.
Link translation	All	Local server	Translate connection objects into replica links (partnerships) for each server relative to each directory partition that it holds.

Topology Evaluation and Connection Object Generation

The KCC on a destination domain controller evaluates the topology by reading the existing connection objects. For each connection object, the KCC reads attribute values of the NTDS Settings object (class **nTDSDSA**) of the source domain controller (indicated by the **fromServer** value on the connection object) to determine what directory partitions its destination domain controller has in common with the source domain controller.

Topology evaluation for all domain controllers

To determine the connection objects that need to be generated, the KCC uses information stored in the attributes of the NTDS Settings object that is associated with each server object, as follows:

- For all directory partitions, the multivalued attribute **hasMasterNCs** stores the distinguished names of all directory partitions that are stored on that domain controller.

Tech Savvy Productions

- For all domain controllers, the value of the **options** attribute indicates whether that domain controller is configured to host the global catalog.
- The **hasPartialReplicaNCs** attribute contains the set of partial-replica directory partitions (global catalog read-only domain partitions) that are located on the domain controller that is represented by the server object.

Topology evaluation for domain controllers running Windows Server 2003

For all domain controllers that are running Windows Server 2003, the **msDS-HasDomainNCs** attribute of the NTDS Settings object contains the name of the domain directory partition that is hosted by the domain controller.

In forests that have the forest functional level of at least Windows Server 2003 or Windows Server 2003 interim, the following additional information is used by the KCC to evaluate the topology for application directory partitions and to generate the needed connections:

- The linked multivalued attribute **msDS-NC-Replica-Locations** on cross-reference objects stores the distinguished names of NTDS Settings objects for all domain controllers that are configured to host a replica of the corresponding application directory partition.

Note

  - When you remove Active Directory from a server that hosts an application directory partition, its corresponding entry in this multivalued attribute is automatically dropped because **msDS-NC-Replica-Locations** is a linked attribute.

- Application directory partition replica locations are determined by matching the values of the **hasMasterNCs** attribute with the values of the **msDS-NC-Replica-Locations** linked multivalued attribute of cross-reference objects. The **msDS-NC-Replica-Locations** attribute holds distinguished name references to the NTDS Settings objects for domain controllers that have been configured to store replicas of the application directory partition. The **msDS-NC-Replica-Locations** attribute facilitates the enumeration of *existing* replicas for a given application directory partition. Connection objects can

Tech Savvy Productions

then be created between the domain controllers that hold matching replicas.

Be aware that due to replication latency, the configuration of replicas in attribute values does not guarantee the existence of the replica on a given server. For example, you can designate a domain controller as a global catalog server by clicking the **Global Catalog** check box on the NTDS Settings object properties in Active Directory Sites and Services. However, until all of the partial domain directory partitions have replicated to that domain controller and global-catalog-specific SRV records are registered in DNS, it is not a functioning global catalog server (does not advertise as a global catalog server in DNS). Similarly, observing the NTDS Settings name for a server in the **msDS-NC-Replica-Locations** attribute on the cross-reference object does not indicate that the replica has necessarily been fully replicated to that server.

Connection Translation

All KCCs process their connection objects and translate them into connection agreements, also called “replica links,” between pairs of domain controllers. At specified intervals, Active Directory replicates data from the source domain controller to the destination for directory partitions that they have in common. These replication agreements do not appear in the administrative tools; the replication engine uses them internally to track the directory partitions that are to be replicated from specified servers.

For each directory partition that two domain controllers have in common and that matches the full and partial characteristics of a replication source, the KCC creates (or updates) a replication agreement on the destination domain controller. Replication agreements take the form of entries for each source domain controller in the **repsFrom** attribute on the topmost object of each directory partition replica. This value is stored and updated locally on the domain controller and is not replicated. The KCC updates this attribute each time it runs.

For example, suppose a connection object is created between two domain controllers from different domains. Assuming that neither of these domain controllers is a global catalog server and neither stores an application directory partition, the KCC identifies the only two directory partitions that the domain controllers have in common — the schema directory partition and the

Tech Savvy Productions

configuration directory partition. If a connection object links domain controllers in the same domain, at least three directory partitions are replicated: the schema directory partition, the configuration directory partition, and the domain directory partition.

In contrast, if the connection object that is created establishes replication between two domain controllers that are global catalog servers, then in addition to the directory partitions the domain controllers have in common, a partial replica of each additional domain directory partition in the forest is also replicated between the two domain controllers over the same connection.

For more information about replication agreements, see "How the Active Directory Replication Model Works."

Read-only and Writable Replicas

When computing the replication topology, the KCC must consider whether a replica is writable or read-only. For each potential set of replication partners in the topology, the considerations are as follows:

- A writable replica can receive updates from a corresponding writable replica.
- A read-only replica can receive updates from a corresponding writable replica.
- A read-only replica can receive updates from a corresponding read-only replica.
- A writable replica *cannot* receive updates from a corresponding read-only replica.

In Windows 2000 forests, for any one domain directory partition, the KCC calculates two topologies: one for the writable replicas and one for the read-only replicas. This calculation allows redundant connections for read-only replicas under certain conditions.

The improved KCC spanning tree algorithm eliminates redundancy that can occur in Windows 2000. The algorithm computes only one topology with slightly different behavior for replicating the global catalog. The KCC on a domain

Tech Savvy Productions

controller that is not a global catalog server does not consider global catalog servers in its calculations for read-only domain replicas because it never replicates read-only data *from* a global catalog server.

Automated Intrasite Topology Generation

For replication within a site, a topology is generated and then optimized to minimize the number of hops to three. The means by which the three-hop minimum is achieved varies according to the number of domain controllers that are hosted in the site as well as the presence of global catalog servers. Generally, the intrasite topology is formed in a ring. The topology becomes more complex as the number of servers increases. However, the KCC can accommodate thousands of domain controllers in a site.

Simplified Ring Topology Generation

A simplified process for creating the topology for replication within a site begins as follows:

- The KCC generates a list of all servers in the site that hold that directory partition.
- These servers are connected in a ring.
- For each neighboring server in the ring from which the current domain controller is to replicate, the KCC creates a connection object if one does not already exist.

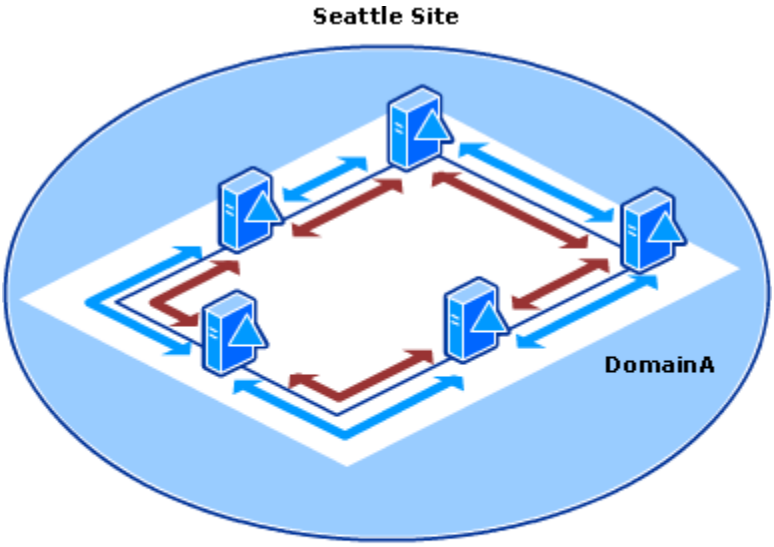
This simple approach guarantees a topology that tolerates a single failure. If a domain controller is not available, it is not included in the ring that is generated by the list of servers. However, this topology, with no other adjustments, accommodates only seven servers. Beyond this number, the ring would require more than three hops for some servers.

The simplest case scenario — seven or fewer domain controllers, all in the same domain and site — would result in the replication topology shown in the following diagram. The only directory partitions to replicate are a single domain directory partition, the schema directory partition, and the configuration directory partition. Those topologies are generated first, and at that point, sufficient connections to replicate each directory partition have already been created.

Tech Savvy Productions

In the next series of diagrams, the arrows indicate one-way or two-way replication of the type of directory partitions indicated in the Legend.

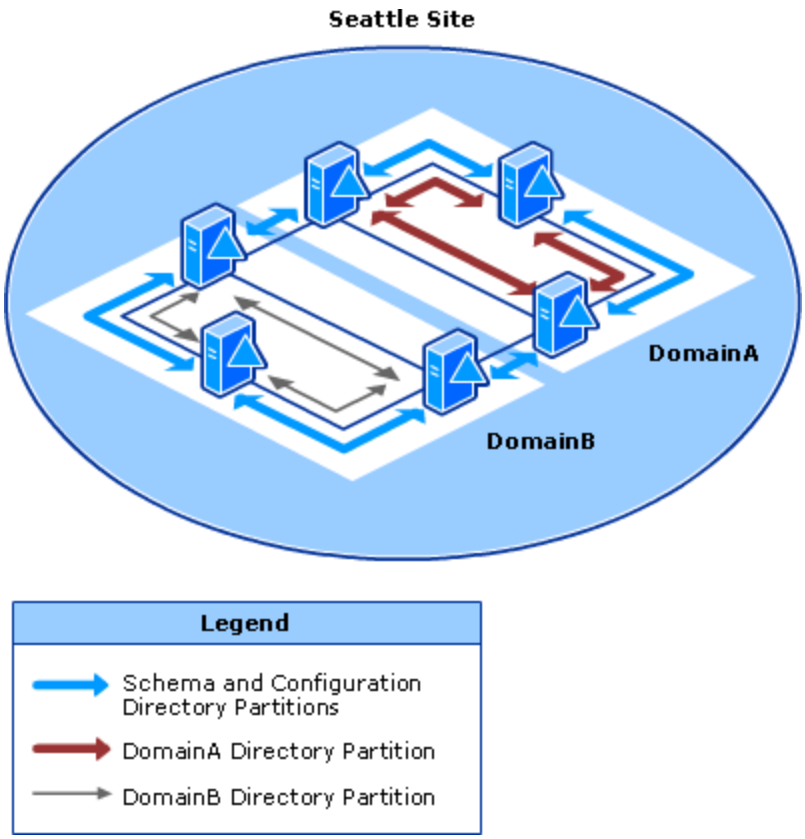
Simple Ring Topology that Requires No Optimization



Because a ring topology is created for each directory partition, the topology might look different if domain controllers from a second domain were present in the site. The next diagram illustrates the topology for domain controllers from two domains in the same site with no global catalog servers defined in the site.

Ring Topology for Two Domains in a Site that Has No Global Catalog Server

Tech Savvy Productions



The next diagram illustrates replication between a global catalog server and three domains to which the global catalog server does not belong. When a global catalog server is added to the site in DomainA, additional connections are required to replicate updates of the other domain directory partitions to the global catalog server. The KCC on the global catalog server creates connection objects to replicate from domain controllers for each of the other domain directory partitions within the site, or from another global catalog server, to update the read-only partitions. Wherever a domain directory partition is replicated, the KCC also uses the connection to replicate the schema and configuration directory partitions.

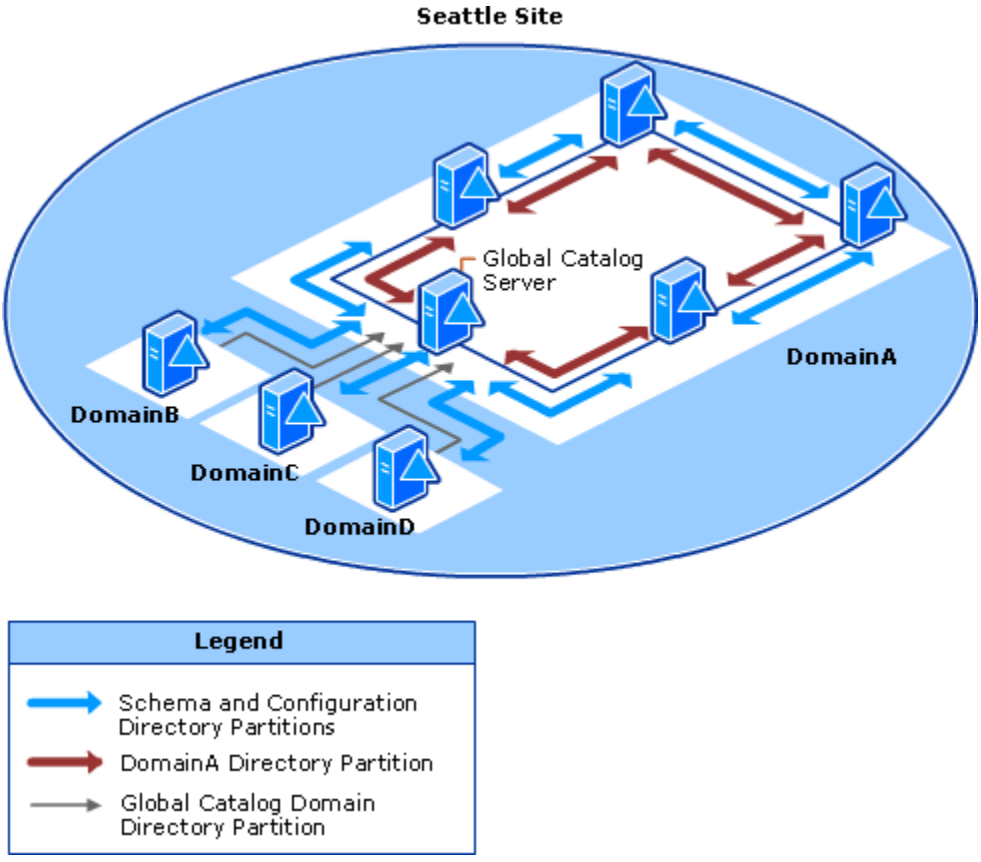
Note

- Connection objects are generated independently for the configuration and schema directory partitions (one connection) and for the separate domain and application directory partitions, unless a connection from the same source to destination domain controllers already exists for one directory

Tech Savvy Productions

partition. In that case, the same connection is used for all (duplicate connections are not created).

Intrasite Topology for Site with Four Domains and a Global Catalog Server



Expanded Ring Topology Within a Site

When the number of servers in a site grows beyond seven, the KCC estimates the number of connections that are needed so that if a change occurs at any one domain controller, there are as many replication partners as needed to ensure that no domain controller is more than three replication hops from another domain controller (that is, a change takes no more than three hops before it reaches another domain controller that has not already received the change by another path). These optimizing connections are created at random and are not necessarily created on every third domain controller.

The KCC adds connections automatically to optimize a ring topology within a site, as follows:

Tech Savvy Productions

- Given a set of nodes in a ring, create the minimum number of connections, *n*, that each server must have to ensure a path of no more than three hops to another server.

Given *n*, topology generation proceeds as follows.

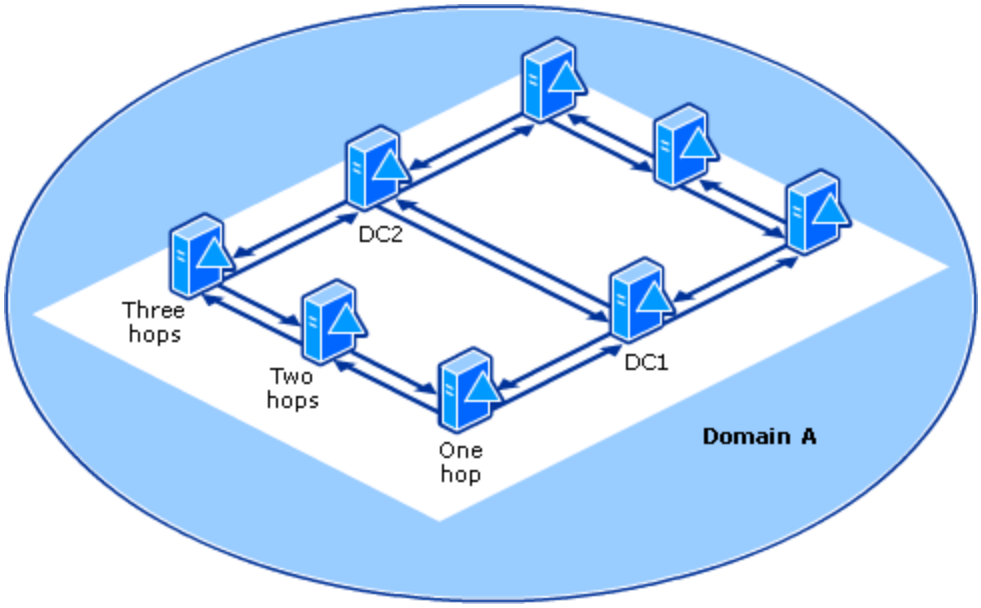
- If the local server does not have *n* extra connections, the KCC does the following:
    - Chooses *n* other servers randomly in the site as source servers.
    - For each of those servers, creates a connection object.

This approach approximates the minimum-hop goal of three servers. In addition, it grows well, because as the site grows in server count, old optimizing connections are still useful and are not removed. Also, every time an additional 9 to 11 servers are added, a connection object is deleted at random; then a new one is created, ideally having one of the new servers as its source. This process ensures that, over time, the additional connections are distributed well over the entire site.

The following diagram shows an intrasite ring topology with optimizing connections in a site that has eight domain controllers in the same domain. Without optimizing connections, the hop count from DC1 to DC2 is more than three hops. The KCC creates optimizing connections to limit the hop count to three hops. The two one-way inbound optimizing connections accommodate all directory partitions that are replicated between the two domain controllers.

Intrasite Topology with Optimizing Connections

Tech Savvy Productions



Excluded Nonresponding Servers

The KCC automatically rebuilds the replication topology when it recognizes that a domain controller has failed or is unresponsive.

The criteria that the KCC uses to determine when a domain controller is not responsive depend upon whether the server computer is within the site or not. Two thresholds must be reached before a domain controller is declared "unavailable" by the KCC:

- The requesting domain controller must have made *n* attempts to replicate from the target domain controller.
    - For replication between sites, the default value of *n* is 1 attempt.
    - For replication within a site, the following distinctions are made between the two immediate neighbors (in the ring) and the optimizing connections:

For immediate neighbors, the default value of *n* is 0 failed attempts. Thus, as soon as an attempt fails, a new server is tried.

For optimizing connections, the default value of *n* is 1 failed attempt. Thus, as soon as a second failed attempt occurs, a new server is tried.

Tech Savvy Productions

- A certain amount of time must have passed since the last successful replication attempt.
    - For replication between sites, the default time is 2 hours.
    - For replication within a site, a distinction is made between the two immediate neighbors (in the ring) and the optimizing connections:

For immediate neighbors, the default time is 2 hours.

For optimizing connections, the default value is 12 hours.

You can edit the registry to modify the thresholds for excluding nonresponding servers.

Note

- If you must edit the registry, use extreme caution. Registry information is provided here as a reference for use by only highly skilled directory service administrators. It is recommended that you do not directly edit the registry unless, as in this case, there is no Group Policy or other Windows tools to accomplish the task. Modifications to the registry are not validated by the registry editor or by Windows before they are applied, and as a result, incorrect values can be stored. Storage of incorrect values can result in unrecoverable errors in the system.

Modifying the thresholds for excluding nonresponding servers requires editing the following registry entries in HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NTDS\Parameters, with the data type REG_DWORD. You can modify these values to any desired value as follows:

For replication between sites, use the following entries:

- **IntersiteFailuresAllowed**

Value: Number of failed attempts

Default: 1

- **MaxFailureTimeForIntersiteLink (sec)**

Value: Time that must elapse before being considered unavailable, in seconds

Tech Savvy Productions

Default: 7200 (2 hours)

For optimizing connections within a site, use the following entries:

- **NonCriticalLinkFailuresAllowed**

Value: Number of failed attempts

Default: 1

- **MaxFailureTimeForNonCriticalLink**

Value: Time that must elapse before considered unavailable, in seconds

Default: 43200 (12 hours)

For immediate neighbor connections within a site, use the following entries:

- **CriticalLinkFailuresAllowed**

Value: Number of failed attempts

Default: 0

- **MaxFailureTimeForCriticalLink**

Value: Time that must elapse before considered unavailable, in seconds

Default: 7200 (2 hours)

When the original domain controller begins responding again, the KCC automatically restores the replication topology to its pre-failure condition the next time that the KCC runs.

Fully Optimized Ring Topology Generation

Taking the addition of extra connections, management of nonresponding servers, and growth-management mechanisms into account, the KCC proceeds to fully optimize intrasite topology generation. The appropriate connection objects are created and deleted according to the available criteria.

Note

- Connection objects from nonresponding servers are not deleted because the condition is expected to be transient.

Tech Savvy Productions

Automated Intersite Topology Generation

To produce a replication topology for hundreds of domains and thousands of sites in a timely manner and without compromising domain controller performance, the KCC must make the best possible decision when confronted with the question of which network link to use to replicate a given directory partition between sites. Ideally, connections occur only between servers that contain the same directory partition(s), but when necessary, the KCC can also use network paths that pass through servers that do not store the directory partition.

Intersite topology generation and associated processes are improved in Windows Server 2003 in the following ways:

- Improved scalability: A new spanning tree algorithm achieves greater efficiency and scalability when the forest has a functional level of Windows Server 2003. For more information about this new algorithm, see "Improved KCC Scalability in Windows Server 2003 Forests" later in this section.
- Less network traffic: A new method of communicating the identity of the ISTG reduces the amount of network traffic that is produced by this process. For more information about this method, see "Intersite Topology Generator" later in this section.
- Multiple bridgehead servers per site and domain, and initial bridgehead server load balancing: An improved algorithm provides random selection of multiple bridgehead servers per domain and transport (the Windows 2000 algorithm allows selection of only one). The load among bridgehead servers is balanced the first time connections are generated. For more information about bridgehead server load balancing, see "Windows Server 2003 Multiple Bridgehead Selection" later in this section.

Factors Considered by the KCC

The spanning tree algorithm used by the KCC that is running as the ISTG to create the intersite replication topology determines how to connect all the sites that need to be connected with the minimum number of connections and the least cost. The algorithm must also consider the fact that each domain controller has at least three directory partitions that potentially require synchronization with other

Tech Savvy Productions

sites, not all domain controllers store the same partitions, and not all sites host the same domains.

The ISTG considers the following factors to arrive at the intersite replication topology:

- Location of domain directory partitions (calculate a replication topology for each domain).
- Bridgehead server availability in each site (at least one is available).
- All explicit site links.
- With automatic site link bridging in effect, consider all implicit paths as a single path with a combined cost.
- With manual site link bridging in effect, consider the implicit combined paths of only those site links included in the explicit site link bridges.
- With no site link bridging in effect, where the site links represent hops between domain controllers in the same domain, replication flows in a store-and-forward manner through sites.

Improved KCC Scalability in Windows Server 2003 Forests

KCC scalability is greatly improved in Windows Server 2003 forests over its capacity in Windows 2000 forests. Windows 2000 forests scale safely to support 300 sites, whereas Windows Server 2003 forests have been tested to 3,000 sites. This level of scaling is achieved when the forest functional level is Windows Server 2003. At this forest functional level, the method for determining the least-cost path from each site to every other site for each directory partition is significantly more efficient than the method that is used in a Windows 2000 forest or in a Windows Server 2003 forest that has a forest functional level of Windows 2000.

Windows 2000 Spanning Tree Algorithm

The ability of the KCC to generate the intersite topology in Windows 2000 forests is limited by the amount of CPU time and memory that is consumed when the KCC computes the replication topology in large environments that use transitive (bridged) site links. In a Windows 2000 forest, a potential disadvantage of bridging

Tech Savvy Productions

all site links affects only very large networks (generally, greater than 100 sites) where periods of high CPU activity occur every 15 minutes when the KCC runs. By default, the KCC creates a single bridge for the entire network, which generates more routes that must be processed than if automatic site link bridging is not used and manual site link bridges are applied selectively.

In a Windows 2000 forest, or in a Windows Server 2003 forest that has a forest functional level of Windows 2000, the KCC reviews the comparison of multiple paths to and from every destination and computes the spanning tree of the least-cost path. The spanning tree algorithm works as follows:

- Computes a cost matrix by identifying each site pair (that is, each pair of bridgehead servers in different sites that store the directory partition) and the cost on the site link connecting each pair.

Note

  - This matrix is actually computed by Intersite Messaging and used by the KCC.

- By using the costs computed in the matrix, builds a spanning tree between sites that store the directory partition.

This method becomes inefficient when there are a large number of sites.

Note

- CPU time and memory is not an issue in a Windows 2000 forest as long as the following criteria apply:
- D is the number of domains in your network
- S is the number of sites in your network
- (1 + D) * S^2 <= 100,000

Windows Server 2003 Spanning Tree Algorithm

A more efficient spanning tree algorithm improves efficiency and scalability of replication topology generation in Windows Server 2003 forests. When the forest functional level is either Windows Server 2003 or Windows Server 2003 interim, the improved algorithm takes effect and computes a minimum-cost spanning tree

Tech Savvy Productions

of connections between the sites that host a particular directory partition, but eliminates the inefficient cost matrix. Thus, the KCC directly determines the lowest-cost spanning tree for each directory partition, considering the schema and configuration directory partitions as a single tree. Where the spanning trees overlap, the KCC generates a single connection between domain controllers for replication of all common directory partitions.

In a Windows Server 2003 forest, both versions of the KCC spanning tree algorithms are available. The algorithm for Windows 2000 forests is retained for backwards compatibility with the Windows 2000 KCC. It is not possible for the two algorithms to run simultaneously in the same enterprise.

DFS Site Costing and Windows Server 2003 SP1 Site Options

When the forest functional level is Windows Server 2003 or Windows Server 2003 interim and the ISTG does not use Intersite Messaging to calculate the intersite cost matrix, DFS can still use Intersite Messaging to compute the cost matrix for its site-costing functionality, provided that the **Bridge all site links** option is enabled. In branch office deployments, where the large number of sites and site links makes automatic site link bridging too costly in terms of the replication connections that are generated, the **Bridge all site links** option is usually disabled on the IP container (CN=IP,CN=Inter-Site Transports,CN=Sites,CN=Configuration,DC=*ForestRootDomain*). If the **Bridge all site links** option is disabled, DFS is unable to use Intersite Messaging to calculate site costs.

When the forest functional level is Windows Server 2003 or Windows Server 2003 interim and the ISTG in a site is running Windows Server 2003 with SP1, you can have the **Bridge all site links** option enabled, and then use a site option to turn off automatic site link bridging for KCC operation without hampering the ability of DFS to use Intersite Messaging to calculate the cost matrix. This site option is set by running the command **repadmin /siteoptions W2K3_BRIDGES_REQUIRED**. This option is applied to the NTDS Site Settings object (CN=NTDS Site Settings,CN=*SiteName*,CN=Sites,CN=Configuration,DC=*ForestRootDomain*). When this method is used to disable automatic site link bridging (as opposed to disabling **Bridge all site links**), default Intersite Messaging options enable the site-costing calculation to occur for DFS.

Tech Savvy Productions

Note

The site option on the NTDS Site Settings object can be set on any domain controller, but it does not take effect until replication of the change reaches the ISTG role holder for the site.

Intersite Topology Generator

The KCC on the domain controller that has the ISTG role creates the inbound connections on all domain controllers in its site that require replication with domain controllers in other sites. The sum of these connections for all sites in the forest forms the intersite replication topology.

A fundamental concept in the generation of the topology within a site is that each server does its part to create a site-wide topology. In a similar manner, the generation of the topology between sites depends on each site doing its part to create a forest-wide topology between sites.

ISTG Role Ownership and Viability

The owner of the ISTG role is communicated through normal Active Directory replication. Initially, the first domain controller in the site is the ISTG role owner. It communicates its role ownership to other domain controllers in the site by writing the distinguished name of its child NTDS Settings object to the **interSiteTopologyGenerator** attribute of the NTDS Site Settings object for the site. As a change to the configuration directory partition, this value is replicated to all domain controllers in the forest.

The ISTG role owner is selected automatically. The role ownership does not change unless:

- The current ISTG role owner becomes unavailable.
- All domain controllers in the site are running Windows 2000 and one of them is upgraded to Windows Server 2003.

If at least one domain controller in a site is running Windows Server 2003, the ISTG role is assumed by a domain controller that is running Windows Server 2003.

Tech Savvy Productions

The viability of the current ISTG is assessed by all other domain controllers in the site. The need for a new ISTG in a site is established differently, depending on the forest functional level that is in effect.

- Windows 2000 functional level: At 30-minute intervals, the current ISTG notifies every other domain controller of its existence and availability by writing the **interSiteTopologyGenerator** attribute of the NTDS Site Settings object for the site. The change replicates to every domain controller in the forest. The KCC on each domain controller monitors this attribute for its site to verify that it has been written. If a period of 60 minutes elapses without a modification to the attribute, a new ISTG declares itself.
- Windows Server 2003 or Windows Server 2003 interim functional level: Each domain controller maintains an up-to-dateness vector, which contains an entry for each domain controller that holds a full replica of any directory partition that the domain controller replicates. On domain controllers that are running Windows Server 2003, this up-to-dateness vector contains a timestamp that indicates the times that it was last contacted by its replication partners, including both direct and indirect partners (that is, every domain controller that replicates a directory partition that is stored by this domain controller). The timestamp is recorded whether or not the local domain controller actually received any changes from the partner. Because all domain controllers store the schema and configuration directory partitions, every domain controller is guaranteed to have the ISTG for its site among the domain controllers in its up-to-dateness vector.

This timestamp eliminates the need to receive periodic replication of the updated **interSiteTopologyGenerator** attribute from the current ISTG. When the timestamp indicates that the current ISTG has not contacted the domain controller in the last 120 minutes, a new ISTG declares itself.

The Windows Server 2003 method eliminates the network traffic that is generated by periodically replicating the **interSiteTopologyGenerator** attribute update to every domain controller in the forest.

ISTG Eligibility

Tech Savvy Productions

When at least one domain controller in a site is running Windows Server 2003, the eligibility for the ISTG role depends on the operating system of the domain controllers. When a new ISTG is required, each domain controller computes a list of domain controllers in the site. All domain controllers in the site arrive at the same ordered list. Eligibility is established as follows:

- If no domain controllers in the site are running Windows Server 2003, all domain controllers that are running Windows 2000 Server are eligible. The list of eligible servers is ordered by GUID.
- If at least one domain controller in the site is running Windows Server 2003, all domain controllers that are running Windows Server 2003 are eligible. In this case, the entries in the list are sorted first by operating system and then by GUID. In a site in which some domain controllers are running Windows 2000 Server, domain controllers that are running Windows Server 2003 remain at the top of the list and use the GUID in the same manner to maintain the order.

The domain controller that is next in the list of servers after the current owner declares itself the new ISTG by writing the **interSiteTopologyGenerator** attribute on the NTDS Site Settings object.

If the current ISTG is temporarily disconnected from the topology, as opposed to being shut down, and a new ISTG declares itself in the interim, then two domain controllers can temporarily assume the ISTG role. When the original ISTG resumes replication, it initially considers itself to be the current ISTG and creates inbound replication connection objects, which results in duplicate intersite connections. However, as soon as the two ISTGs replicate with each other, the last domain controller to write the **intersiteTopologyGenerator** attribute continues as the single ISTG and removes the duplicate connections.

Bridgehead Server Selection

Bridgehead servers can be selected in the following ways:

- Automatically by the ISTG from all domain controllers in the site.

Tech Savvy Productions

- Automatically by the ISTG from all domain controllers that are identified as preferred bridgehead servers, if any preferred bridgehead servers are assigned. Preferred bridgehead servers must be assigned manually.
- Manually by creating a connection object on a domain controller in one site from a domain controller in a different site.

By default, when at least one domain controller in a site is running Windows Server 2003 (regardless of forest functional level), any domain controller that hosts a domain in the site is a candidate to be an ISTG-selected bridgehead server. If preferred bridgehead servers are selected, candidates are limited to this list. The connections from remote servers are distributed among the available candidate bridgehead servers in each site. The selection of multiple bridgehead servers per domain and transport is new in Windows Server 2003. The ISTG uses an algorithm to evaluate the list of domain controllers in the site that can replicate each directory partition. This algorithm is improved in Windows Server 2003 to randomly select multiple bridgehead servers per directory partition and transport. In sites containing only domain controllers that are running Windows 2000 Server, the ISTG selects only one bridgehead server per directory partition and transport.

When bridgehead servers are selected by the ISTG, the ISTG ensures that each directory partition in the site that has a replica in any other site can be replicated to and from that site or sites. Therefore, if a single domain controller hosts the only replica of a domain in a specific site and the domain has domain controllers in another site or sites, that domain controller must be a bridgehead server in its site. In addition, that domain controller must be able to connect to a bridgehead server in the other site that also hosts the same domain directory partition.

Note

- If a site has a global catalog server but does not contain at least one domain controller of every domain in the forest, then at least one bridgehead server must be a global catalog server.

Preferred Bridgehead Servers

Because bridgehead servers must be able to accommodate more replication traffic than non-bridgehead servers, you might want to control which servers have this responsibility. To specify servers that the ISTG can designate as bridgeheads, you

Tech Savvy Productions

can select domain controllers in the site that you want the ISTG to always consider as preferred bridgehead servers for the specified transport. These servers are used exclusively to replicate changes collected from the site to any other site over that transport. Designating preferred bridgehead servers also serves to exclude those domain controllers that, for reasons of capability, you do *not* want to be used as bridgehead servers.

Depending on the available transports, the directory partitions that must be replicated, and the availability of global catalog servers, multiple bridgehead servers might be required to replicate full and partial copies of directory data from one site to another.

The ISTG recognizes preferred bridgehead servers by reading the **bridgeheadTransportList** attribute of the server object. When this attribute has a value that is the distinguished name of the transport container that the server uses for intersite replication (IP or SMTP), the KCC treats the server as a preferred bridgehead server. For example, the value for the IP transport is CN=IP,CN=Inter-Site Transports,CN=Sites,CN=Configuration,DC=*ForestRootDomainName*. You can use Active Directory Sites and Services to designate a preferred bridgehead server by opening the server object properties and placing either the IP or SMTP transport into the preferred list, which adds the respective transport distinguished name to the **bridgeheadTransportList** attribute of the server.

The **bridgeheadServerListBl** attribute of the transport container object is a backlink attribute of the **bridgeheadTransportList** attribute of the server object. If the **bridgeheadServerListBl** attribute contains the distinguished name of at least one server in a site, then the KCC uses only preferred bridgehead servers to replicate changes for that site, according to the following rules:

- If at least one server is designated as a preferred bridgehead server, updates to the domain directory partition hosted by that server can be replicated only from a preferred bridgehead server. If at the time of replication no preferred bridgehead server is available for that directory partition, replication of that directory partition does not occur.
- If any bridgehead servers are designated but no domain controller is designated as a preferred bridgehead server for a specific directory

Tech Savvy Productions

partition that has replicas in another site or sites, the KCC selects a domain controller to act as the bridgehead server, if one is available that can replicate the directory partition to the other site or sites.

Therefore, to use preferred bridgehead servers effectively, be sure to:

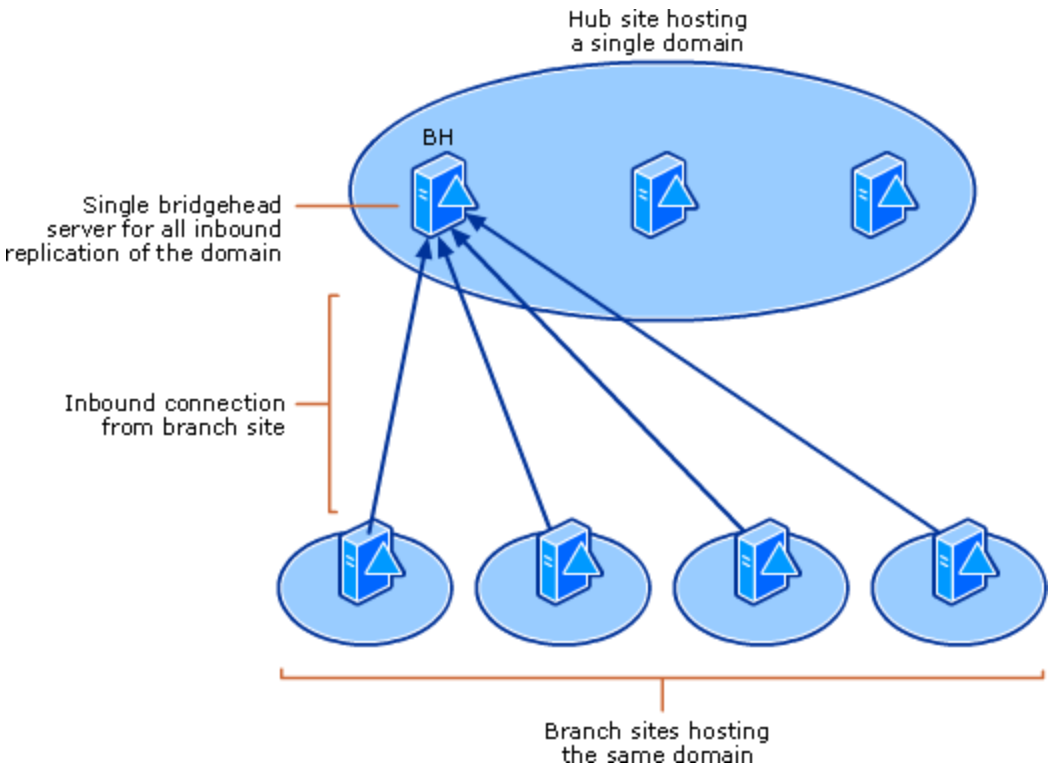
- Assign at least two or more bridgehead servers for each of the following:
    - Any domain directory partition that has a replica in any other site.
    - Any application directory partition that has a replica in another site.
    - The schema and configuration directory partitions (one bridgehead server replicates both) if no domains in the site have replicas in other sites.
- If the site has a global catalog server, select the global catalog server as one of the preferred bridgehead servers.

Windows 2000 Single Bridgehead Selection

In a Windows 2000 forest or in a Windows Server 2003 forest that has a forest functional level of Windows 2000, the ISTG selects a single bridgehead server per directory partition and transport. The selection changes only when the bridgehead server becomes unavailable. The following diagram shows the automatic bridgehead server (BH) selection that occurs in the hub site where all domain controllers host the same domain directory partition and multiple sites have domain controllers that host that domain directory partition.

Windows 2000 Single Bridgehead Server in a Hub Site that Serves Multiple Branch Sites

Tech Savvy Productions



Windows Server 2003 Multiple Bridgehead Selection

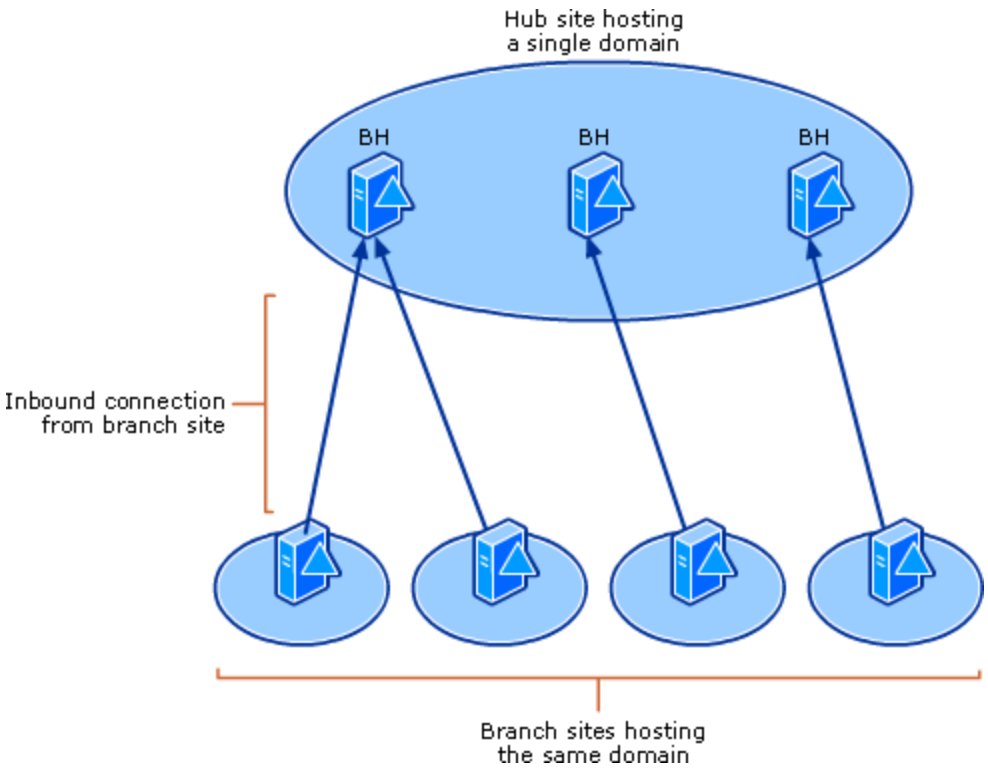
When at least one domain controller in a site is running Windows Server 2003 (and thereby becomes the ISTG), the ISTG begins performing random load balancing of new connections. This load balancing occurs by default, although it can be disabled.

When creating a new connection, the KCC must choose endpoints from the set of eligible bridgeheads in the two endpoint sites. Whereas in Windows 2000 the ISTG always picks the same bridgehead for all connections, in Windows Server 2003 it picks one randomly from the set of possible bridgeheads.

Assuming two out of three of the domain controllers have been added to the site since the ISTG was upgraded to Windows Server 2003, the following diagram shows the connections that might exist on domain controllers in the hub site to accommodate the four branch sites that have domain controllers for the same domain.

Random Bridgehead Server Selection in a Hub Site in which the ISTG Runs Windows Server 2003

Tech Savvy Productions



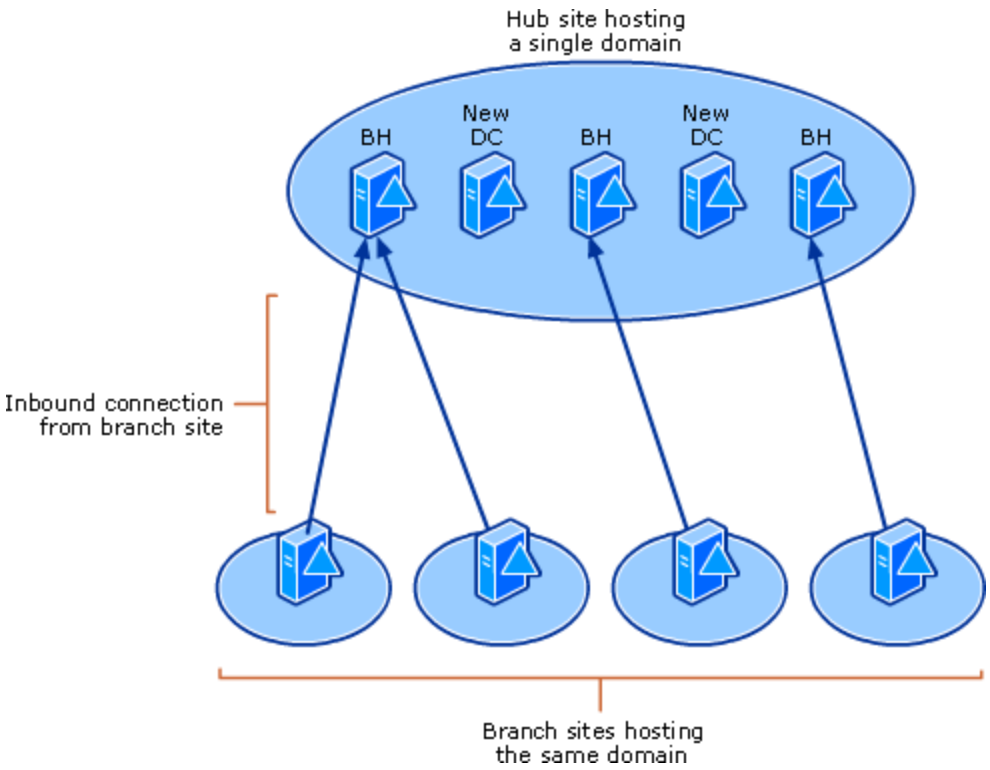
If one or more new domain controllers are added to the hub site, the inbound connections on the existing bridgehead servers are not automatically re-balanced. The next time it runs, the ISTG considers the newly added server(s) as follows:

- If preferred bridgehead servers are not selected in the site, the ISTG considers the newly added servers as candidate bridgehead servers and creates new connections randomly if new connections are needed. It does not remove or replace the existing connections.
- If preferred bridgehead servers are selected in the site, the ISTG does not consider the newly added servers as candidate bridgehead servers unless they are designated as preferred bridgehead servers.

The initial connections remain in place until a bridgehead server becomes unavailable, at which point the KCC randomly replaces the connection on any available bridgehead server. That is, the endpoints do not change automatically for the existing bridgehead servers. In the following diagram, two new domain controllers are added to the hub site, but the existing connections are not redistributed.

New Domain Controllers with No New Connections Created

Tech Savvy Productions



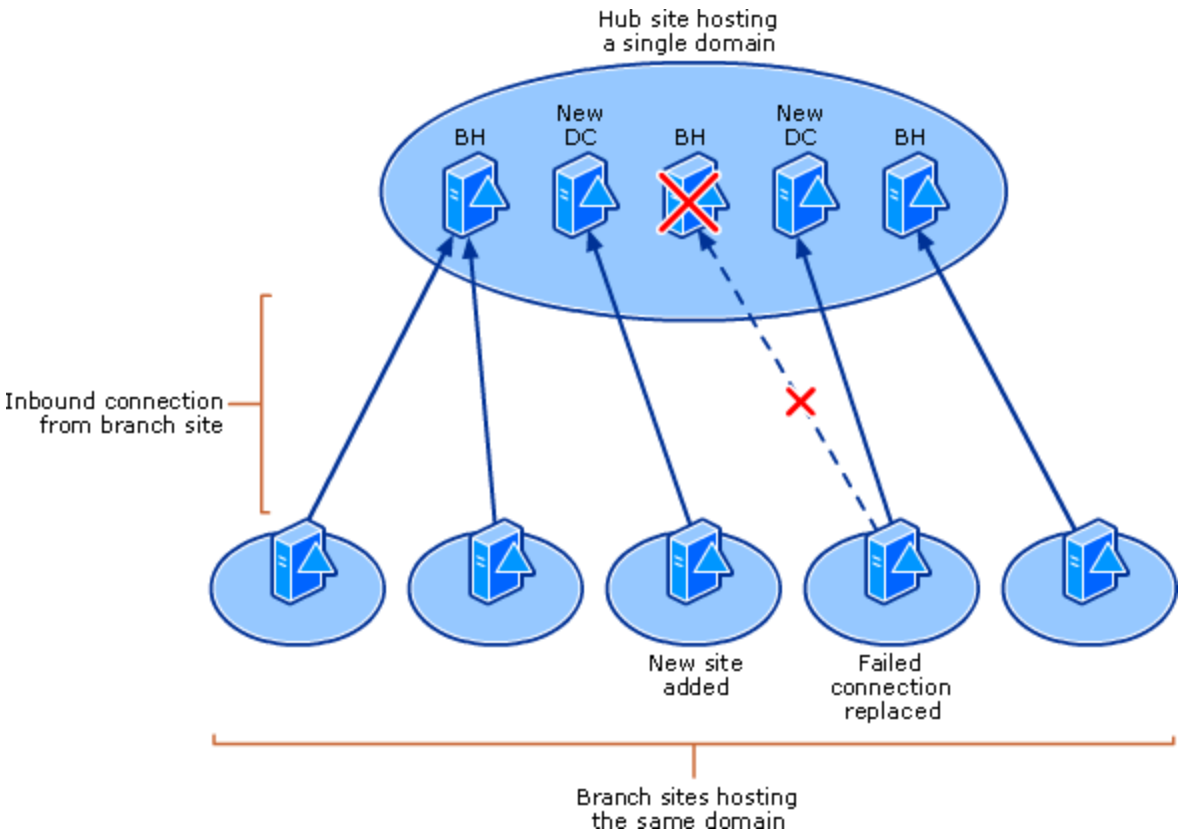
Although the ISTG does not rebalance the load among the existing bridgehead servers in the hub site after the initial connections are created, it does consider the added domain controllers as candidate bridgehead servers under either of the following conditions:

- A new site is added that requires a bridgehead server connection to the hub site.
- An existing connection to the hub site becomes unavailable.

The following diagram illustrates the inbound connection that is possible on a new domain controller in the hub site to accommodate a failed connection on one of the existing hub site bridgehead servers. In addition, a new branch site is added and a new inbound connection can potentially be created on the new domain controller in the hub site. However, because the selection is random, there is no guarantee that the ISTG creates the connections on the newly added domain controllers.

Possible New Connections for Added Site and Failed Connection

Tech Savvy Productions



Using ADLB to Balance Hub Site Bridgehead Server Load

In large hub-and-spoke topologies, you can implement the redistribution of existing bridgehead server connections by using the Active Directory Load Balancing (ADLB) tool (Adlb.exe), which is included with the "Windows Server 2003 Active Directory Branch Office Deployment Guide." ADLB makes it possible to dynamically redistribute the connections on bridgehead servers. This application works independently of the KCC, but uses the connections that are created by the KCC. Connections that are manually created are not moved by ADLB. However, manual connections are factored into the load-balancing equations that ADLB uses.

The ISTG is limited to making modifications in its site, but ADLB modifies both inbound and outbound connections on eligible bridgehead servers and offers schedule staggering for outbound connections.

For more information about how bridgehead server load balancing and schedule staggering work with ADLB, see the "Windows Server 2003 Active Directory

Tech Savvy Productions

Branch Office Planning and Deployment Guide" on the Web at https://go.microsoft.com/fwlink/?linkID=28523.

Network Ports Used by Replication Topology

By default, RPC-based replication uses dynamic port mapping. When connecting to an RPC endpoint during Active Directory replication, the RPC run time on the client contacts the RPC endpoint mapper on the server at a well-known port (port 135). The server queries the RPC endpoint mapper on this port to determine what port has been assigned for Active Directory replication on the server. This query occurs whether the port assignment is dynamic (the default) or fixed. The client never needs to know which port to use for Active Directory replication.

Note

- An endpoint comprises the protocol, local address, and port address.

In addition to the dynamic port 135, other ports that are required for replication to occur are listed in the following table.

Port Assignments for Active Directory Replication

Service Name	UDP	TCP
LDAP	389	389
LDAP		636 (Secure Sockets Layer [SSL])
LDAP		3268 (global catalog)
Kerberos	88	88
DNS	53	53
SMB over IP	445	445

Replication within a domain also requires FRS using a dynamic RPC port.

Setting Fixed Replication Ports Across a Firewall

For each service that needs to communicate across a firewall, there is a fixed port and protocol. Normally, the directory service and FRS use dynamically allocated ports that require a firewall to have a wide range of ports open. Although FRS

Tech Savvy Productions

cannot be restricted to a fixed port, you can edit the registry to restrict the directory service to communicate on a static port.

Note

- If you must edit the registry, use extreme caution. Registry information is provided here as a reference for use by only highly skilled directory service administrators. It is recommended that you do not directly edit the registry unless, as in this case, there is no Group Policy or other Windows tools to accomplish the task. Modifications to the registry are not validated by the registry editor or by Windows before they are applied, and as a result, incorrect values can be stored. Storage of incorrect values can result in unrecoverable errors in the system.

Restricting the directory service to using a fixed port requires editing the **TCP/IP Port** registry entry (REG_DWORD), located in:

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\NTDS\Parameters

Changing this registry entry on a domain controller and restarting it causes the directory service to use the TCP port named in the registry entry. For example, port 49152 is DWORD=0000c000 (hexadecimal).

Related Information

The following resources contain additional information that is relevant to this section.

- Active Directory Replication Model Technical Reference
- TCP/IP Technical Reference
- FRS Technical Reference
- DNS Support for Active Directory Technical Reference
- DFS Technical Reference

Tech Savvy Productions

Troubleshooting Active Directory Replication Problems

In this article

1. Introduction and resources for troubleshooting Active Directory replication
2. Event and tool solution recommendations
3. Ruling out intentional disruptions or hardware failures
4. Responding to failure of an outdated server running Windows 2000 Server

Applies to: Windows Server 2022, Windows Server 2019, Windows Server 2016, Windows Server 2012 R2, Windows Server 2012

- It can help you quickly identify and fix common Active Directory replication issues

Active Directory replication problems can have several different sources. For example, Domain Name System (DNS) problems, networking issues, or security problems can all cause Active Directory replication to fail.

The rest of this topic explains tools and a general methodology to fix Active Directory replication errors. The following subtopics cover symptoms, causes, and how to resolve specific replication errors:

Introduction and resources for troubleshooting Active Directory replication

Inbound or outbound replication failure causes Active Directory objects that represent the replication topology, replication schedule, domain controllers, users, computers, passwords, security groups, group memberships, and Group Policy to be inconsistent between domain controllers. Directory inconsistency and replication failure cause either operational failures or inconsistent results, depending on the domain controller that is contacted for the operation, and can prevent the application of Group Policy and access control permissions. Active Directory Domain Services (AD DS) depends on network connectivity, name resolution, authentication and authorization, the directory database, the replication topology, and the replication engine. When the root cause of a

Tech Savvy Productions

replication problem is not immediately obvious, determining the cause among the many possible causes requires systematic elimination of probable causes.

For a UI-based tool to help monitor replication and diagnose errors, download and run the Microsoft Support and Recovery Assistant tool, **or** use the Active Directory Replication Status Tool if you only want to analyze the replication status.

For a comprehensive document that describes how you can use the Repadmin tool to troubleshoot Active Directory replication is available; see Monitoring and Troubleshooting Active Directory Replication Using Repadmin.

For information about how Active Directory replication works, see the following technical references:

- Active Directory Replication Model Technical Reference
- Active Director Replication Topology Technical Reference

Event and tool solution recommendations

Ideally, the red (Error) and yellow (Warning) events in the Directory Service event log suggest the specific constraint that is causing replication failure on the source or destination domain controller. If the event message suggests steps for a solution, try the steps that are described in the event. The Repadmin tool and other diagnostic tools also provide information that can help you resolve replication failures.

For detailed information about using Repadmin for troubleshooting replication problems, see Monitoring and Troubleshooting Active Directory Replication Using Repadmin.

Ruling out intentional disruptions or hardware failures

Sometimes replication errors occur because of intentional disruptions. For example, when you troubleshoot Active Directory replication problems, rule out intentional disconnections and hardware failures or upgrades first.

Intentional disconnections

If replication errors are reported by a domain controller that is attempting replication with a domain controller that has been built in a staging site and is currently offline awaiting its deployment in the final production site (a remote

Tech Savvy Productions

site, such as a branch office), you can account for those replication errors. To avoid separating a domain controller from the replication topology for extended periods, which causes continuous errors until the domain controller is reconnected, consider adding such computers initially as member servers and using the install from media (IFM) method to install Active Directory Domain Services (AD DS). You can use the Ntdsutil command-line tool to create installation media that you can store on removable media (CD, DVD, or other media) and ship to the destination site. Then, you can use the installation media to install AD DS on the domain controllers at the site, without the use of replication.

Hardware failures or upgrades

If replication problems occur as a result of hardware failure (for example, failure of a motherboard, disk subsystem, or hard drive), notify the server owner so that the hardware problem can be resolved.

Periodic hardware upgrades can also cause domain controllers to be out of service. Ensure that your server owners have a good system of communicating such outages in advance.

Firewall configuration

By default, Active Directory replication remote procedure calls (RPCs) occur dynamically over an available port through the RPC Endpoint Mapper (RPCSS) on port 135. Make sure that Windows Firewall with Advanced Security and other firewalls are configured properly to allow for replication. For information about specifying the port for Active Directory replication and port settings, see article 224196 in the Microsoft Knowledge Base.

For information about the ports that Active Directory replication uses, see Active Directory Replication Tools and Settings.

For information about managing Active Directory replication over firewalls, see Active Directory Replication over Firewalls.

Responding to failure of an outdated server running Windows 2000 Server

If a domain controller running Windows 2000 Server has failed for longer than the number of days in the tombstone lifetime, the solution is always the same:

Tech Savvy Productions

1. Move the server from the corporate network to a private network.
2. Either forcefully remove Active Directory or reinstall the operating system.
3. Remove the server metadata from Active Directory so that the server object cannot be revived.

You can use a script to clean up server metadata on most Windows operating systems. For information about using this script, see Remove Active Directory Domain Controller Metadata.

By default, NTDS Settings objects that are deleted are revived automatically for a period of 14 days. Therefore, if you do not remove server metadata (use Ntdsutil or the script mentioned previously to perform metadata cleanup), the server metadata is reinstated in the directory, which prompts replication attempts to occur. In this case, errors will be logged persistently as a result of the inability to replicate with the missing domain controller.

Root causes

If you rule out intentional disconnections, hardware failures, and outdated Windows 2000 domain controllers, the remainder of replication problems almost always have one of the following root causes:

- Network connectivity: The network connection might be unavailable, or network settings are not configured properly.
- Name resolution: DNS misconfigurations are a common cause of replication failures.
- Authentication and authorization: Authentication and authorization problems cause "Access denied" errors when a domain controller tries to connect to its replication partner.
- Directory database (store): The directory database might not be able to process transactions fast enough to keep up with replication time-outs.
- Replication engine: If intersite replication schedules are too short, replication queues might be too large to process in the time that is required by the outbound replication schedule. In this case, replication of some

Tech Savvy Productions

changes can be stalled indefinitely potentially, long enough to exceed the tombstone lifetime.

- Replication topology: Domain controllers must have intersite links in AD DS that map to real wide area network (WAN) or virtual private network (VPN) connections. If you create objects in AD DS for the replication topology that are not supported by the actual site topology of your network, replication that requires the misconfigured topology fails.

General approach to fixing problems

Use the following general approach to fixing replication problems:

1. Monitor replication health daily, or use Repadmin.exe to retrieve replication status daily.
2. Attempt to resolve any reported failure in a timely manner by using the methods that are described in event messages and this guide. If software might be causing the problem, uninstall the software before you continue with other solutions.
3. If the problem that is causing replication to fail cannot be resolved by any known methods, remove AD DS from the server and then reinstall AD DS. For more information about reinstalling AD DS, see Decommissioning a Domain Controller.
4. If AD DS cannot be removed normally while the server is connected to the network, use one of the following methods to resolve the problem:
    - Force AD DS removal in Directory Services Restore Mode (DSRM), clean up server metadata, and then reinstall AD DS.
    - Reinstall the operating system, and rebuild the domain controller.

For more information about forcing removal of AD DS, see Forcing the Removal of a Domain Controller.

Using Repadmin to retrieve replication status

Replication status is an important way for you to evaluate the status of the directory service. If replication is working without errors, you know the domain

Tech Savvy Productions

controllers that are online. You also know that the following systems and services are working:

- DNS infrastructure
- Kerberos authentication protocol
- Windows Time service (W32time)
- Remote procedure call (RPC)
- Network connectivity

Use Repadmin to monitor replication status daily by running a command that assesses the replication status of all the domain controllers in your forest. The procedure generates a .csv file that you can open in Microsoft Excel and filter for replication failures.

You can use the following procedure to retrieve the replication status of all domain controllers in the forest.

Requirements

Membership in **Enterprise Admins**, or equivalent, is the minimum required to complete this procedure.

Tools:

- Repadmin.exe
- Excel (Microsoft Office)

To generate a repadmin /showrepl spreadsheet for domain controllers

1. Open a Command Prompt as an administrator: On the Start menu, right-click Command Prompt, and then click Run as administrator. If the User Account Control dialog box appears, provide Enterprise Admins credentials, if required, and then click Continue.
2. At the command prompt, type the following command, and then press ENTER: repadmin /showrepl * /csv > showrepl.csv
3. Open Excel.

Tech Savvy Productions

4. Click the Office button, click Open, navigate to showrepl.csv, and then click Open.
5. Hide or delete column A as well as the Transport Type column, as follows:
6. Select a column that you want to hide or delete.
    - To hide the column, right-click the column, and then click Hide.
    - To delete the column, right-click the selected column, and then click Delete.
7. Select row 1 beneath the column heading row. On the View tab, click Freeze Panes, and then click Freeze Top Row.
8. Select the entire spreadsheet. On the Data tab, click Filter.
9. In the Last Success Time column, click the down arrow, and then click Sort Ascending.
10. In the Source DC column, click the filter down arrow, point to Text Filters, and then click Custom Filter.
11. In the Custom AutoFilter dialog box, under Show rows where, click does not contain. In the adjacent text box, type del to eliminate from view the results for deleted domain controllers.
12. Repeat step 11 for the Last Failure Time column, but use the value does not equal, and then type the value 0.
13. Resolve replication failures.

For every domain controller in the forest, the spreadsheet shows the source replication partner, the time that replication last occurred, and the time that the last replication failure occurred for each naming context (directory partition). By using Autofilter in Excel, you can view the replication health for working domain controllers only, failing domain controllers only, or domain controllers that are the least or most current, and you can see the replication partners that are replicating successfully.

Replication problems and resolutions

Tech Savvy Productions

Replication problems are reported in event messages and in various error messages that occur when an application or service attempts an operation. Ideally, these messages are collected by your monitoring application or when you retrieve replication status.

Most replication problems are identified in the event messages that are logged in the Directory Service event log. Replication problems might also be identified in the form of error messages in the output of the repadmin /showrepl command.

repadmin /showrepl error messages that indicate replication problems

To identify Active Directory replication problems, use the repadmin /showrepl command, as described in the previous section. The following table shows error messages that this command generates, along with the root causes of the errors and links to topics that provide solutions for the errors.

Repadmin error	Root Cause	Solution
The time since last replication with this server has exceeded the tombstone lifetime.	A domain controller has failed inbound replication with the named source domain controller long enough for a deletion to have been tombstoned, replicated, and garbage-collected from AD DS.	Event ID 2042: It has been too long since this machine replicated
No inbound neighbors.	If no items appear in the "Inbound Neighbors" section of the output that is generated by repadmin /showrepl, the domain controller was not able to establish replication links with another domain controller.	Fixing Replication Connectivity Problems (Event ID 1925)
Access is denied.	A replication link exists between two domain controllers, but replication cannot be performed properly as a result of an authentication failure.	Fixing Replication Security Problems

Tech Savvy Productions

Repadmin error	Root Cause	Solution
Last attempt at <date - time> failed with the "Target account name is incorrect."	This problem can be related to connectivity, DNS, or authentication issues. If this is a DNS error, the local domain controller could not resolve the globally unique identifier (GUID)-based DNS name of its replication partner.	Fixing Replication DNS Lookup Problems (Event IDs 1925, 2087, 2088) Fixing Replication Security Problems Fixing Replication Connectivity Problems (Event ID 1925)
LDAP Error 49.	The domain controller computer account might not be synchronized with the Key Distribution Center (KDC).	Fixing Replication Security Problems
Cannot open LDAP connection to local host	The administration tool could not contact AD DS.	Fixing Replication DNS Lookup Problems (Event IDs 1925, 2087, 2088)
Active Directory replication has been preempted.	The progress of inbound replication was interrupted by a higher-priority replication request, such as a request that was generated manually with the repadmin /sync command.	Wait for replication to complete. This informational message indicates normal operation.
Replication posted, waiting.	The domain controller posted a replication request and is waiting for an answer. Replication is in progress from this source.	Wait for replication to complete. This informational message indicates normal operation.

The following table lists common events that might indicate problems with Active Directory replication, along with root causes of the problems and links to topics that provide solutions for the problems.

Tech Savvy Productions

Event ID and source	Root cause	Solution
1311 NTDS KCC	The replication configuration information in AD DS does not accurately reflect the physical topology of the network.	Fixing Replication Topology Problems (Event ID 1311)
1388 NTDS Replication	Strict replication consistency is not in effect, and a lingering object has been replicated to the domain controller.	Fixing Replication Lingering Object Problems (Event IDs 1388, 1988, 2042)
1925 NTDS KCC	The attempt to establish a replication link for a writable directory partition failed. This event can have different causes, depending on the error.	Fixing Replication Connectivity Problems (Event ID 1925) Fixing Replication DNS Lookup Problems (Event IDs 1925, 2087, 2088)
1988 NTDS Replication	The local domain controller has attempted to replicate an object from a source domain controller that is not present on the local domain controller because it may have been deleted and already garbage-collected. Replication does not proceed for this directory partition with this partner until the situation is resolved.	Fixing Replication Lingering Object Problems (Event IDs 1388, 1988, 2042)
2042 NTDS Replication	Replication has not occurred with this partner for a tombstone lifetime, and replication cannot proceed.	Fixing Replication Lingering Object Problems (Event IDs 1388, 1988, 2042)
2087 NTDS Replication	AD DS could not resolve the DNS host name of the source domain controller to an IP address, and replication failed.	Fixing Replication DNS Lookup Problems (Event IDs 1925, 2087, 2088)

Tech Savvy Productions

Event ID and source	Root cause	Solution
2088 NTDS Replication	AD DS could not resolve the DNS host name of the source domain controller to an IP address, but replication succeeded.	Fixing Replication DNS Lookup Problems (Event IDs 1925, 2087, 2088)
5805 Net Logon	A machine account failed to authenticate, which is usually caused by either multiple instances of the same computer name or the computer name not replicating to every domain controller.	Fixing Replication Security Problems

For more information about replication concepts, see Active Directory Replication Technologies.

Next steps

For more information, including support articles specific to error codes see the support article: How to troubleshoot common Active Directory replication errors

Configuring a secure Admin Computer for AD/Server Troubleshooting

Applies to: Windows Server 2022, Windows Server 2019, Windows Server 2016, Windows Server 2012 R2, Windows Server 2012

Tech Savvy Productions

No internet access on this machine, no email access to this machine.

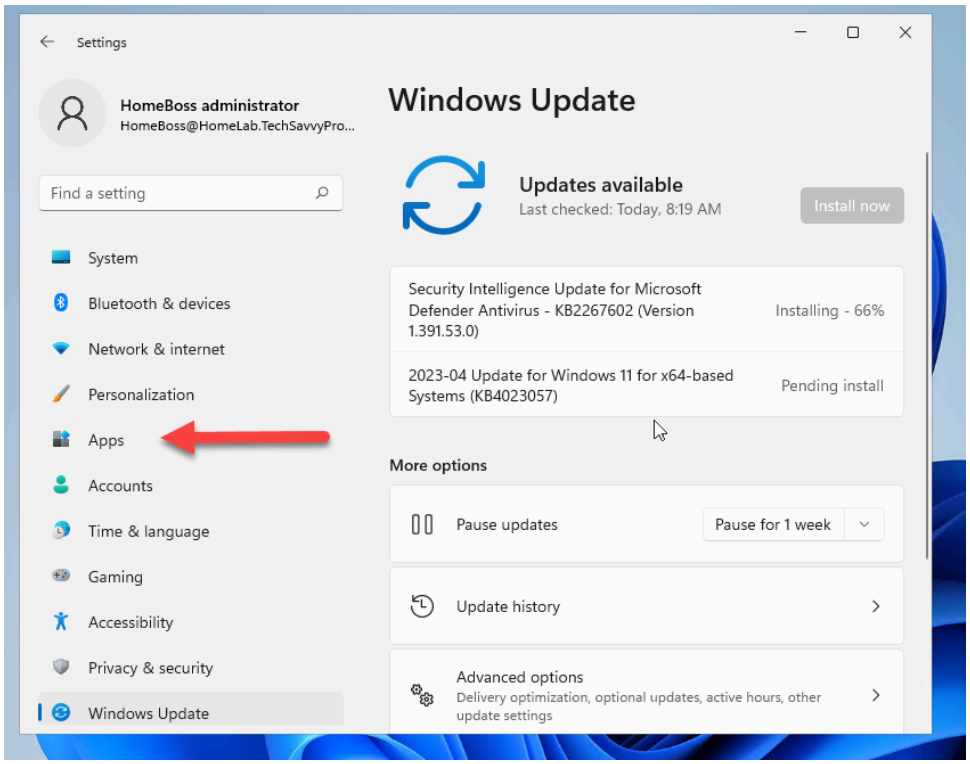
Before you use advanced troubleshooting techniques to identify and fix Active Directory problems, configure your computers for troubleshooting. You should also have a basic understanding of troubleshooting concepts, procedures, and tools.

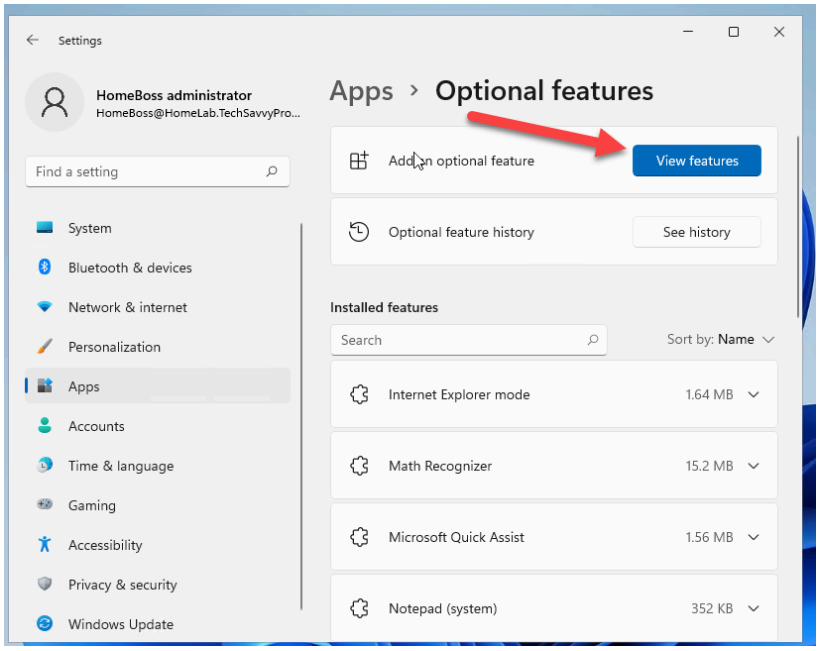
For information about monitoring tools for Windows Server, see the Step-by-Step Guide for [Performance and Reliability Monitoring in Windows Server]

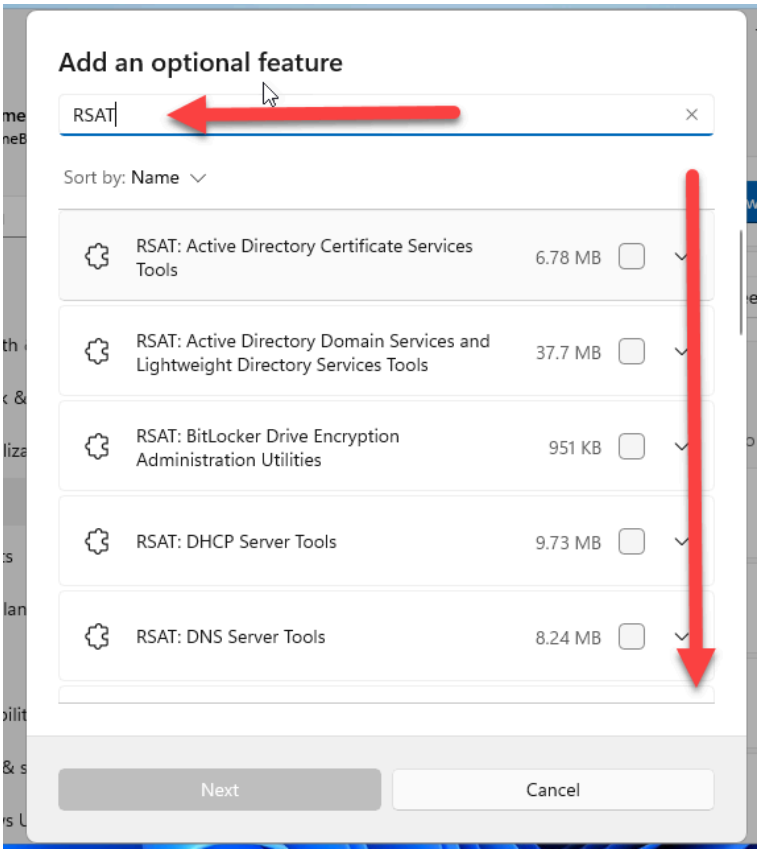
Configuration tasks for troubleshooting

To configure your computer for troubleshooting Active Directory Domain Services (AD DS), perform the following tasks:

Install Remote Server Administration Tools for AD DS







When you install AD DS to create a domain controller, the administrative tools that you use to manage AD DS are installed automatically. If you want to manage domain controllers remotely from a computer that is not a domain controller, you can install the Remote Server Administration Tools (RSAT) on a member server or workstation that is running a supported version of Windows. RSAT replaces Windows Support Tools from Windows Server 2003.

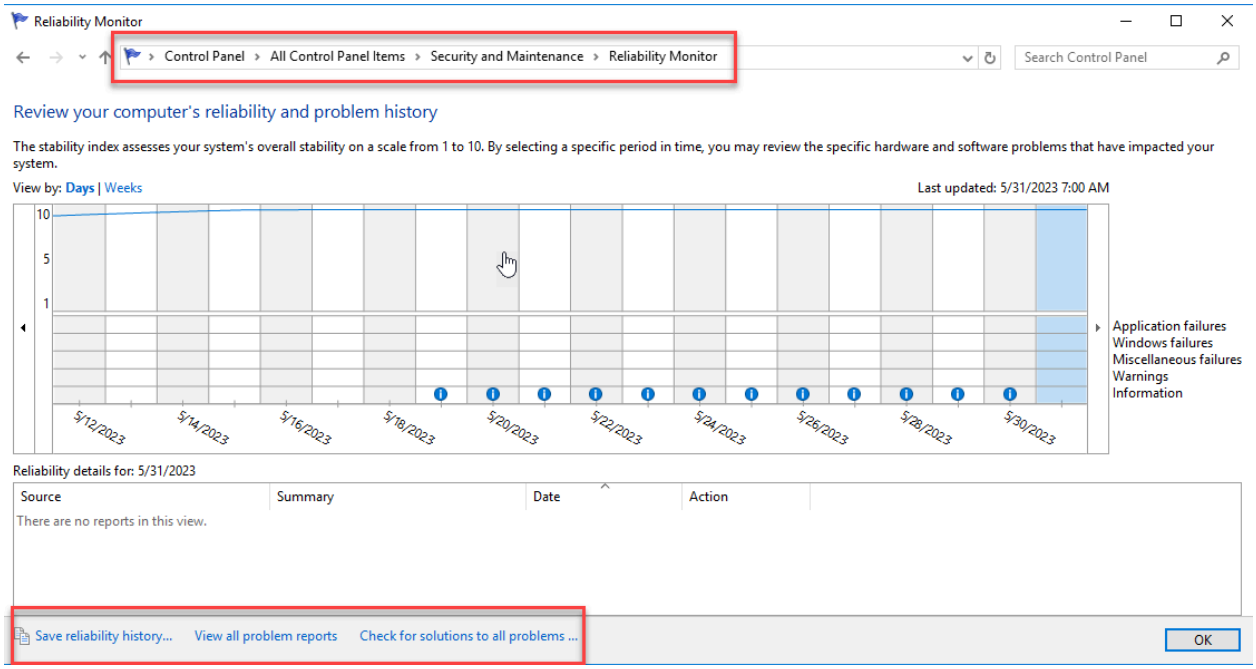
For information about installing RSAT, see the article Remote Server Administration Tools.

Configure Reliability and Performance Monitor

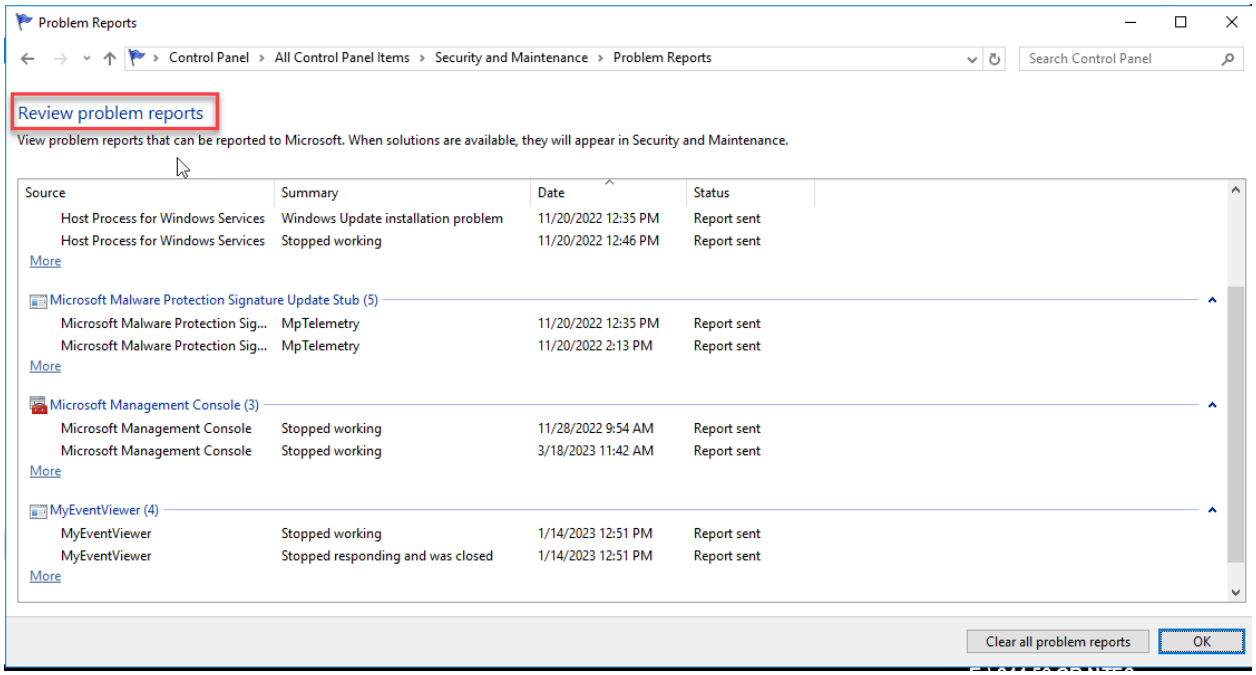
Tech Savvy Productions

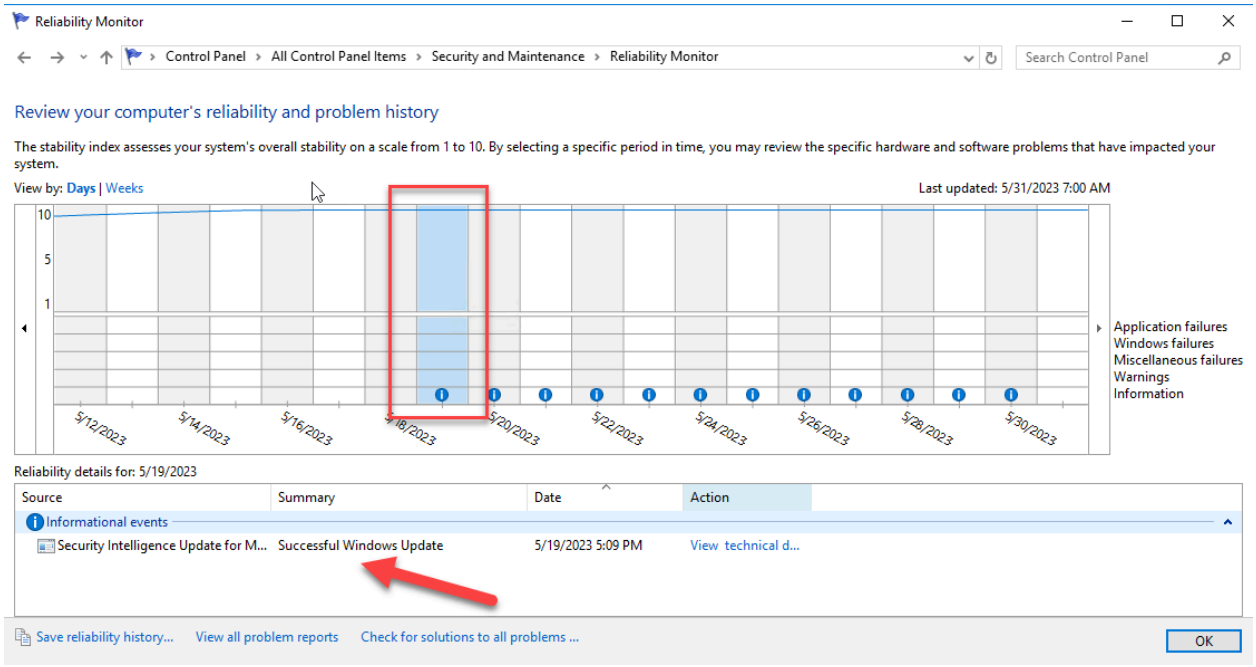
Windows Server includes the Windows Reliability and Performance Monitor, which is a Microsoft Management Console (MMC) snap-in that combines the functionality of previous stand-alone tools, including Performance Logs and Alerts and System Monitor. This snap-in provides a graphical user interface (GUI) for customizing Data Collector Sets and Event Trace Sessions.

Reliability and Performance Monitor also includes Reliability Monitor, an MMC snap-in that tracks changes to the system and compares them to changes in system stability, providing a graphical view of their relationship.

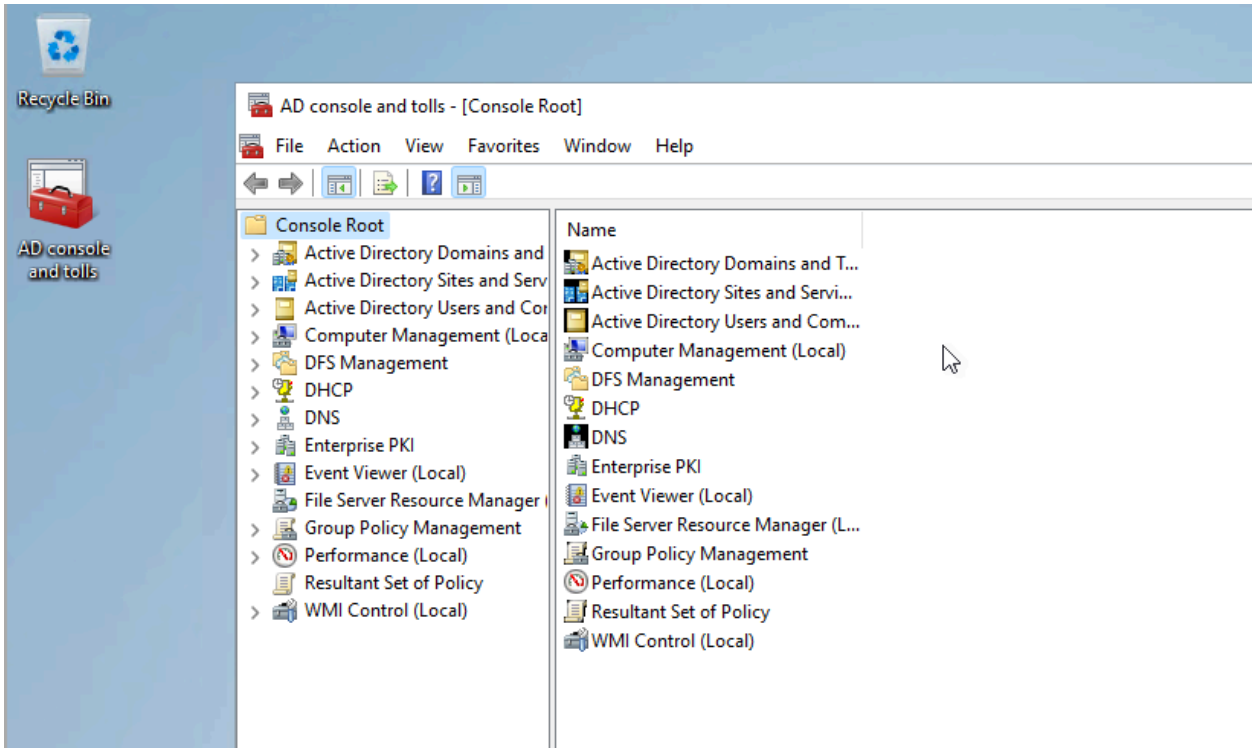


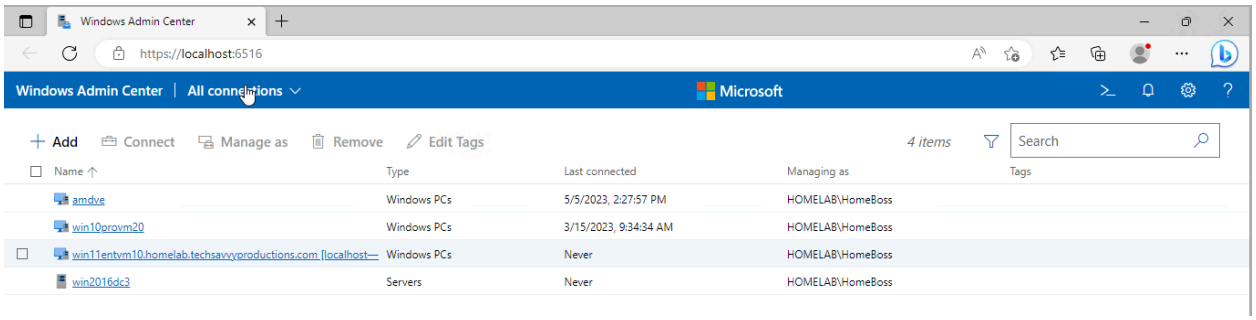
Tech Savvy Productions



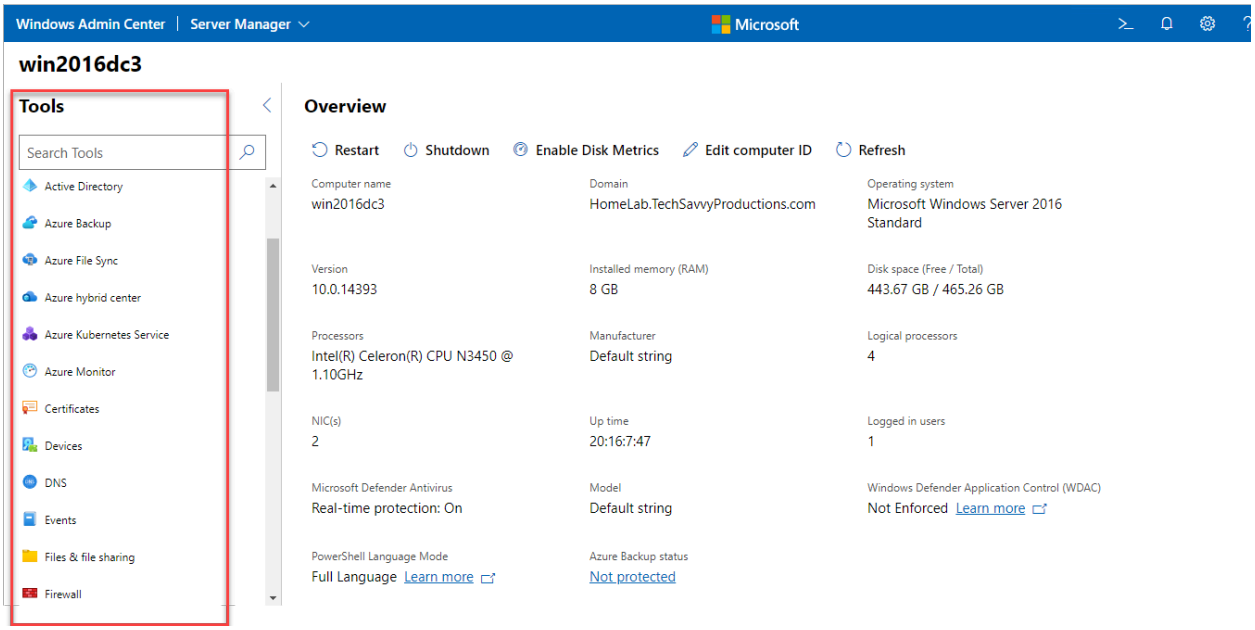


Tech Savvy Productions





Tech Savvy Productions



Set logging levels

If the information that you receive in the Directory Service log in Event Viewer is not sufficient for troubleshooting, raise the logging levels by using the appropriate registry entry in **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NTDS\Diagnostics**.

By default, the logging levels for all entries are set to **0**, which provides the minimum amount of information. The highest logging level is **5**. Increasing the level for an entry causes additional events to be logged in the Directory Service event log.

Use the following procedure to change the logging level for a diagnostic entry. Membership in **Domain Admins**, or equivalent, is the minimum required to complete this procedure.

Warning

We recommend that you do not directly edit the registry unless there is no other alternative. Modifications to the registry are not validated by the registry editor or by Windows before they are applied, and as a result, incorrect values can be stored. This can result in unrecoverable errors in the system. When possible, use Group Policy or other Windows tools, such as MMC snap-ins, to accomplish tasks,

Tech Savvy Productions

rather than editing the registry directly. If you must edit the registry, use extreme caution.

To change the logging level for a diagnostic entry

1. Click **Start** > **Run** > type **regedit** > click **OK**.
2. Navigate to the entry for which you want to set logging in.
    - EXAMPLE: HKEY_LOCAL_MACHINESYSTEMCurrentControlSetServicesNTDSDiagnostics
3. Double-click the entry, and in **Base**, click **Decimal**.
4. In **Value**, type an integer from **0** through **5**, and then click **OK**.

AD DS Troubleshooting

Applies to: Windows Server 2022, Windows Server 2019, Windows Server 2016, Windows Server 2012 R2

This section includes troubleshooting recommendations and procedures for diagnosing and fixing problems that may occur during Active Directory replication.

Tech Savvy Productions

It focuses on how to respond to Directory Service event log entries and how to interpret messages that tools such as Repadmin.exe and Dcdiag.exe might report.

Repadmin.exe and Dcdiag.exe are available on all domain controllers that run Windows Server 2012 R2 or later versions. For more information about how to use these tools to troubleshoot problems, see the following articles.

- Configuring a Computer for Troubleshooting Active Directory
- Troubleshooting Active Directory Replication Problems

Another useful technology is Event Tracing for Windows (ETW). You can use ETW to troubleshoot LDAP communications among the domain controllers. For more information, see Using ETW to troubleshoot LDAP connections.

You can also install Remote Server Administration Tools (RSAT) on a member server that is running Windows 10. For information about how to install RSAT, see Remote Server Administration Tools.

Tech Savvy Productions

In Active Directory (AD), **the partitions,** also known as **Naming Contexts**, are named based on their function. There are three mandatory partitions and one optional partition:

1. **Domain Partition (also known as Domain Naming Context or DNC)**: This partition contains all objects, such as users, groups, and computers, within a specific domain. Its name will reflect the name of the domain (for example, if the domain is "corp.com," the domain partition could be named "DC=corp,DC=com").
2. **Configuration Partition (also known as Configuration Naming Context or CNC)**: This partition holds information about the physical structure of the forest, such as the site and replication information. It's replicated to all domain controllers in the entire forest. Its name is usually in the format "CN=Configuration,DC=domainname,DC=com."
3. **Schema Partition (also known as Schema Naming Context or SNC)**: This partition contains the schema for the forest. The schema is a set of rules that define the classes of objects and their attributes that can be created within AD. It's also replicated to all domain controllers in the entire forest. Its name is typically "CN=Schema,CN=Configuration,DC=domainname,DC=com."
4. **Application Partition**: This is an optional partition that can be created by applications to store application-specific data. They can be configured to replicate to any set of domain controllers, providing flexibility based on the needs of the application. The name of an application partition can vary depending on the application that created it.

Tech Savvy Productions

Active Directory **Partitions**, or **Naming Contexts,** can be thought of like drawers in a filing cabinet.

Imagine a large, shared office environment where everyone needs access to different types of documents. The office has a huge filing cabinet which is divided into several distinct drawers. Each drawer holds different types of documents:

1. The **Schema drawer** contains the rules and formats for all documents. It includes the definitions of each type of form, such as what fields it should have, the types of information it can contain, etc. This is the Schema Partition in Active Directory, which defines object classes and attributes within the AD.
2. The **Configuration drawer** contains information about the overall office environment, like floor plans, rules, and other settings. This is the Configuration Partition in AD, which stores information about the physical layout and replication topology of the AD environment.
3. The **Domain drawer** contains all the personnel files for the employees in the office - their names, addresses, roles, departments, and other details. There can be multiple domain drawers if the office has several departments. Each Domain Partition in AD contains information about all objects in a domain.
4. The **Application drawer** contains information about the applications and services the office uses. These might be project management systems, internal communication tools, etc. The Application Partition in AD is optional and stores information about applications in the AD environment.

Just as not everyone in the office can access every drawer (for instance, the HR team might be the only ones allowed to access the personnel files), not all data in an Active Directory is accessible to all users or services. Access is controlled based on permissions, similar to how office workers might have keys to specific drawers.

And similarly, changes to some drawers - like the Schema drawer - would affect the entire office and are thus kept to a minimum, the same way changes to the Schema Partition in AD are infrequent and carefully controlled.

Remember, in a distributed environment, multiple copies of this filing cabinet exist, each at a different location (domain controller), and they regularly synchronize to ensure they have the same set of documents.

Tech Savvy Productions

Active Directory (AD) is a distributed system, meaning that its data is stored on multiple servers known as Domain Controllers (DCs). These DCs need to communicate with each other to ensure they all have a consistent copy of the AD data. This communication process is known as replication.

However, if each DC were to communicate with every other DC directly, it could lead to network congestion and inefficiencies. To optimize this, DCs use a more efficient pattern for replication, which is where the **Knowledge Consistency Checker (KCC)** comes into play.

The KCC is a built-in process that runs on all DCs and automatically generates and maintains the **replication topology for the AD forest**. It's like a traffic controller in a busy city, creating efficient routes for data to move from one DC to another.

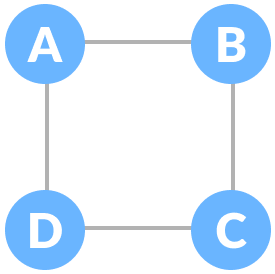
The **KCC uses a mathematical algorithm** to create a minimum replication topology that forms a least-cost spanning tree. A spanning tree is a concept from graph theory where all nodes (DCs, in this case) are connected in such a way that there are no cycles, and all nodes are reachable from any other, but with the least number of connections or 'edges' (replication links).

This **spanning tree** ensures two things:

1. Any DC can reach any other DC either directly or indirectly (through one or more other DCs).
2. The paths are optimized so that replication occurs efficiently with minimal network traffic. (see graphs below)

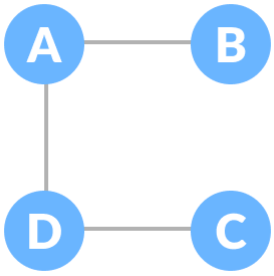
When changes in the network occur, such as the addition or removal of a DC, the KCC will automatically recalculate the replication topology to maintain efficiency. The goal is always to keep the Active Directory data consistent across all DCs while using network resources as efficiently as possible.

Tech Savvy Productions

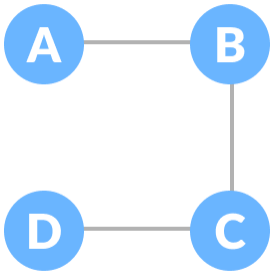


A normal graph

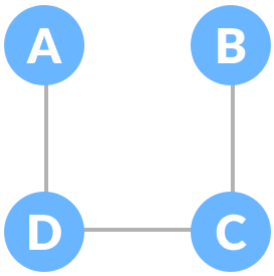
Some of the possible spanning trees that can be created from the above graph are:



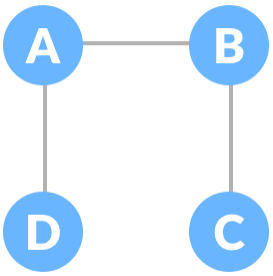
A spanning tree



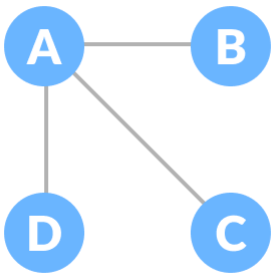
A spanning tree



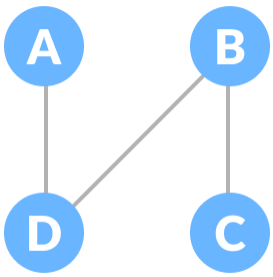
A spanning tree



A spanning tree



A spanning tree



A spanning tree

Tech Savvy Productions

An analogy explaining the difference between relative distinguished name RDN and a distinguished name DN in Active directory

Consider this example. You're meeting a friend at a large music festival with numerous stages, food stalls, and other attractions. The festival is a city within a city for the day, complete with its own map and landmarks.

The Distinguished Name (DN) in Active Directory can be compared to the exact instructions on how to find your friend, starting from the main entrance. For instance, "Enter through the Main Gate, turn right at the Food Court, go past the Jazz Stage, and I'm next to the T-Shirt Stand under the big Oak tree." This complete set of directions from a fixed point of reference (the main entrance) helps you to find your friend in a unique location within the complex festival map. It's a full, unambiguous path to the destination.

Now, let's consider the **Relative Distinguished Name (RDN)**. It's like your friend saying, "I'm under the big Oak tree." This gives you their immediate, unique identifying feature (under the Oak tree), but doesn't provide the full context. Without knowing the layout of the festival, or if there are multiple Oak trees, you might struggle to find them. The RDN is a single, unique identifying attribute - but its meaning is often dependent on the context provided by the rest of the DN.

In Active Directory, an object's DN provides the complete hierarchical path to reach that object, whereas the RDN is just the attribute that distinguishes the object within its immediate container (like an OU or a CN). But just as with the Oak tree at the festival, the RDN alone is typically not enough to find the object within the whole AD structure.

Tech Savvy Productions

In Active Directory, the Distinguished Name (DN) provides a full path to identify an object's exact position in the directory tree hierarchy. The Relative Distinguished Name (RDN) only describes an object's name relative to its immediate parent container.

Consider a user named John Doe in the Users container within the domain corp.com.

The DN for John Doe would be: "CN=John Doe,OU=Users,DC=corp,DC=com"

In this DN:

- "CN" stands for Common Name, and "John Doe" is the CN.
- "OU" stands for Organizational Unit, and "Users" is the OU.
- "DC" stands for Domain Component, and "corp.com" is the DC (split into two parts, "corp" and "com").

The RDN for John Doe would be: "CN=John Doe"

This only includes the object's name (John Doe) and the type of object it is (a common name, CN). The RDN doesn't include any information about where the object is located in the directory tree, which is why it's "relative".

Tech Savvy Productions

Integrate on-premises AD with Azure

Azure Active Directory

This article compares options for integrating your on-premises Active Directory (AD) environment with an Azure network. For each option, a more detailed reference architecture is available.

Many organizations use Active Directory Domain Services (AD DS) to authenticate identities associated with users, computers, applications, or other resources that are included in a security boundary. Directory and identity services are typically hosted on-premises, but if your application is hosted partly on-premises and partly in Azure, there may be latency sending authentication requests from Azure back to on-premises. Implementing directory and identity services in Azure can reduce this latency.

Azure provides two solutions for implementing directory and identity services in Azure:

- Use Azure AD to create an Active Directory domain in the cloud and connect it to your on-premises Active Directory domain. Azure AD Connect integrates your on-premises directories with Azure AD.
- Extend your existing on-premises Active Directory infrastructure to Azure, by deploying a VM in Azure that runs AD DS as a Domain Controller. This architecture is more common when the on-premises network and the Azure virtual network (VNet) are connected by a VPN or ExpressRoute connection. Several variations of this architecture are possible:
    - Create a domain in Azure and join it to your on-premises AD forest.
    - Create a separate forest in Azure that is trusted by domains in your on-premises forest.
    - Replicate an Active Directory Federation Services (AD FS) deployment to Azure.

The next sections describe each of these options in more detail.

Integrate your on-premises domains with Azure AD

Use Azure Active Directory (Azure AD) to create a domain in Azure and link it to an on-premises AD domain.

Tech Savvy Productions

The Azure AD directory is not an extension of an on-premises directory. Rather, it's a copy that contains the same objects and identities. Changes made to these items on-premises are copied to Azure AD, but changes made in Azure AD are not replicated back to the on-premises domain.

You can also use Azure AD without using an on-premises directory. In this case, Azure AD acts as the primary source of all identity information, rather than containing data replicated from an on-premises directory.

Benefits

- You don't need to maintain an AD infrastructure in the cloud. Azure AD is entirely managed and maintained by Microsoft.
- Azure AD provides the same identity information that is available on-premises.
- Authentication can happen in Azure, reducing the need for external applications and users to contact the on-premises domain.

Challenges

- You must configure connectivity with your on-premises domain to keep the Azure AD directory synchronized.
- Applications may need to be rewritten to enable authentication through Azure AD.
- If you wish to authenticate service and computer accounts, you will have to also deploy Azure Active Directory Domain Services.

Reference architecture

- Integrate on-premises Active Directory domains with Azure Active Directory

AD DS in Azure joined to an on-premises forest

Deploy AD Domain Services (AD DS) servers to Azure. Create a domain in Azure and join it to your on-premises AD forest.

Consider this option if you need to use AD DS features that are not currently implemented by Azure AD.

Benefits

Tech Savvy Productions

- Provides access to the same identity information that is available on-premises.
- You can authenticate user, service, and computer accounts on-premises and in Azure.
- You don't need to manage a separate AD forest. The domain in Azure can belong to the on-premises forest.
- You can apply group policy defined by on-premises Group Policy Objects to the domain in Azure.

Challenges

- You must deploy and manage your own AD DS servers and domain in the cloud.
- There may be some synchronization latency between the domain servers in the cloud and the servers running on-premises.

Reference architecture

- Extend Active Directory Domain Services (AD DS) to Azure

AD DS in Azure with a separate forest

Deploy AD Domain Services (AD DS) servers to Azure, but create a separate Active Directory forest that is separate from the on-premises forest. This forest is trusted by domains in your on-premises forest.

Typical uses for this architecture include maintaining security separation for objects and identities held in the cloud, and migrating individual domains from on-premises to the cloud.

Benefits

- You can implement on-premises identities and separate Azure-only identities.
- You don't need to replicate from the on-premises AD forest to Azure.

Challenges

- Authentication within Azure for on-premises identities requires extra network hops to the on-premises AD servers.

Tech Savvy Productions

- You must deploy your own AD DS servers and forest in the cloud, and establish the appropriate trust relationships between forests.

Reference architecture

- Create an Active Directory Domain Services (AD DS) resource forest in Azure

Extend AD FS to Azure

Replicate an Active Directory Federation Services (AD FS) deployment to Azure, to perform federated authentication and authorization for components running in Azure.

Typical uses for this architecture:

- Authenticate and authorize users from partner organizations.
- Allow users to authenticate from web browsers running outside of the organizational firewall.
- Allow users to connect from authorized external devices such as mobile devices.

Benefits

- You can leverage claims-aware applications.
- Provides the ability to trust external partners for authentication.
- Compatibility with large set of authentication protocols.

Challenges

- You must deploy your own AD DS, AD FS, and AD FS Web Application Proxy servers in Azure.
- This architecture can be complex to configure.

Reference architecture

- Extend Active Directory Federation Services (AD FS) to Azure

Tech Savvy Productions