



Technical Report

Multiprotocol NAS in NetApp ONTAP

Overview and Best Practices

Justin Parisi, NetApp
April 2021 | TR-4887

Abstract

This technical report describes how multiprotocol NAS access works in NetApp® ONTAP® data management software and the best practices for multiprotocol environments.

TABLE OF CONTENTS

Overview	5
What is NAS?	5
What is CIFS/SMB?.....	5
What is NFS?	5
Homogenous versus heterogenous NAS environments	5
Why use multiprotocol NAS?	6
Common challenges	6
How multiprotocol NAS in ONTAP works: The Basics	7
Namespace and file system concepts	7
Network access	14
Access points: Volumes, shares and exports	20
Authentication and name mapping	26
Authorization and permissions.....	30
General best practices	31
Multiprotocol best practices	31
Advanced multiprotocol concepts	33
Multiprotocol NAS file locking	33
Special character considerations	36
Qtree considerations	37
Advanced name mapping concepts.....	47
NAS redirects and global sharing	49
Native CIFS and NFS file auditing	75
Troubleshooting multiprotocol NAS	75
Appendix A: Multiprotocol NAS terminology.....	91
Appendix B: NFS server options.....	93
Appendix C: CIFS/SMB server options.....	97
Where to find additional information	99
Version history.....	100

LIST OF TABLES

Table 1) Limitations of existing security styles.....	26
Table 2) Name mappings and security styles.....	27

Table 3) Limits on local users and groups in clustered Data ONTAP	28
Table 4) Decision matrix for NAS volume and qtree security styles.	31
Table 5) LDAP client schema options – name mapping.	49
Table 6) NFS credential cache settings.....	81
Table 7) Multiprotocol NAS terminology.	91
Table 8) NFS server options that can impact multiprotocol NAS – ONTAP 9.8 and later.....	93
Table 9) CIFS server options that can impact multiprotocol NAS – ONTAP 9.8 and later.....	97

LIST OF FIGURES

Figure 1) Multiprotocol NAS basic operations.	7
Figure 2) Cluster namespace.	8
Figure 3) Load-sharing mirror protection of vsroot volumes.	9
Figure 4) FlexVol design with junctioned architecture for >100TB capacity.	10
Figure 5) FlexVol and FlexGroup volumes architecture comparison.	11
Figure 6) NetApp FlexCache volumes.....	13
Figure 7) Sparse volume details.....	14
Figure 8) Single LIF NAS interaction.	15
Figure 9) Multiple LIF NAS interaction.....	16
Figure 10) NAS clients in a segmented network.....	19
Figure 11) Qtree export specification – ONTAP System Manager.	22
Figure 12) Reordering the Rule Index in ONTAP System Manager.	24
Figure 13) Quota reports – ONTAP System Manager.	41
Figure 14) Quota volume status – ONTAP System Manager.	41
Figure 15) Quota rules – ONTAP System Manager.	42
Figure 16) CIFS Symlink, relative path - same volume.	53
Figure 17) CIFS symlink, absolute path, same volume – default behavior.....	54
Figure 18) CIFS Symlink, absolute path – same volume.....	54
Figure 19) CIFS symlink, absolute path, different volume – default behavior.....	55
Figure 20) CIFS widelink redirect – same SVM.....	56
Figure 21) CIFS symlink – before and after proper configuration.	56
Figure 22) CIFS symlink, different volume/same SVM – widelink.	57
Figure 23) Windows server SMB share.	58
Figure 24) CIFS symlink – widelink to the Windows server.	59
Figure 25) CIFS symlink versus direct connection to a Windows SMB share.	59
Figure 26) Local file symlink – local locality, symlinks_and_widelinks share property.....	60
Figure 27) Local file symlink – local locality, symlinks,no_strict_security share property.	61
Figure 28) Local file symlink – widelink locality, symlinks_and_widelinks share property.	61
Figure 29) Symlink from root of share with junctioned volumes and ../ symlink path.....	62
Figure 30) Symlink from share with redirect to another share – absolute path.....	63

Figure 31) Remote file symlink – local locality, symlinks_and_widelinks share property.....	63
Figure 32) Remote file symlink – Blank file, no_strict_security.....	64
Figure 33) CIFS symlink – no_strict_security.....	66
Figure 34) CIFS symlink – no_strict_security navigation.....	66
Figure 35) CIFS symlink – no_strict_security no set.....	67
Figure 36) CIFS share access error.....	69
Figure 37) Junction path views: Reparse point enabled versus disabled.....	73
Figure 38) Symlink views: Reparse point enabled versus disabled.....	74
Figure 39) Windows DFS with ONTAP as a target.....	74
Figure 40) Permissions view before UNIX SIDs resolve.....	88
Figure 41) Security tab on UNIX security styles.....	89
Figure 42) Security tab on UNIX security styles – permission change.....	89
Figure 43) Security tab hidden on UNIX security styles.....	90

LIST OF BEST PRACTICES

Best Practice 1: Network Design with FlexGroup.....	17
Best Practice 2: Use Some Form of DNS Load Balancing.....	18
Best Practice 3: Special Character Handling – Recommended ONTAP Version.....	36
Best Practice 4: UTF-8 or utf8mb4?.....	37

Overview

This technical report covers multiprotocol NAS access on NetApp storage systems running ONTAP data management software. Multiprotocol NAS access allows enterprise and scale-out storage systems to provide access to clients running both Linux and Windows-based operating systems through the NFS and CIFS/SMB protocols, respectively. For multiprotocol NAS terms, see the “Multiprotocol NAS terminology” section of this doc.

What is NAS?

Multiprotocol NAS is essentially what the name suggests – unified NAS access through multiple NAS protocols. Leveraging multiprotocol NAS on NetApp storage systems allows users on all operating systems to seamlessly access the same datasets, regardless of the type of protocol being used. The protocols involved in multiprotocol environments are CIFS/SMB and NFS.

Is multiprotocol NAS also called mixed mode?

A common misconception is that multiprotocol NAS is also called mixed mode. This misconception creates confusion when implementing a NetApp storage system running NAS, as the concept of a mixed security style also exists. A mixed security style is covered later in this document (in the section titled “Security styles”).

What is CIFS/SMB?

[CIFS/SMB](#) is the way that users share files across Ethernet-based networks primarily on operating systems running Microsoft Windows. CIFS is the native file sharing protocol introduced in Windows 2000 and leverages SMB as the underlying protocol for communication between the client and server in modern operating systems.

CIFS/SMB is also used on other non-Windows operating systems such as Apple, Linux, and Solaris through third-party implementations, such as Samba. Support of CIFS/SMB on non-Windows operating systems on NetApp storage systems varies and can be found on the [Interoperability Matrix Tool \(IMT\)](#).

Note: While CIFS and SMB are different in many ways, this document uses the terms interchangeably.

What is NFS?

[NFS](#) is the way that users share file across Ethernet-based networks primarily on operating systems running Linux, Solaris, UNIX, HP-UX, and so on. NFS follows a series of standards defined by the [Internet Engineering Task Force](#) (IETF) through documents called [Request for Comments](#) (RFC). These standards are followed by all major NFS client and server vendors that intend on delivering enterprise-level NFS access. NFS depends on a series of underlying messages that depend on the version of NFS being used. For more information about NFS in ONTAP, see [TR-4067: NFS Best Practices and Implementation Guide](#).

Homogenous versus heterogeneous NAS environments

Some sites have pure Windows or pure UNIX environments in which all data is accessed using only one of the following:

- CIFS/SMB and NTFS file security
- NFS and UNIX file security (mode bits or NFSv4.x access control lists (ACLs))

However, many sites must enable data sets to be accessed from both Windows and UNIX clients. For these environments, ONTAP has native multiprotocol NAS support. After the user is authenticated on the network and has both appropriate share or export permissions and the necessary file-level permissions, data can be accessed by the user from UNIX hosts using NFS or from Windows hosts using CIFS/SMB.

Leveraging multiprotocol NAS access does not require using [mixed security style \(sometimes referred to as mixed mode\)](#) volumes and/or qtrees, even though it is available as an option.

Why use multiprotocol NAS?

Using multiprotocol NAS with ONTAP data management software delivers several distinct advantages. When data sets can be seamlessly accessed simultaneously by clients using different NAS protocols, the following benefits can be achieved:

- Reduce the overall storage administrator management tasks.
- Require only a single copy of data to be stored for NAS access from multiple clients.
- Protocol agnostic NAS allows storage administrators to control the style of ACL and access control being presented to end users.
- Centralize identity management operations in a NAS environment.

ONTAP data management software has provided enterprise-class multiprotocol NAS access for over 25 years and counting. With the advent of scale-out ONTAP clusters and NetApp ONTAP FlexGroup volumes, storage administrators are allowed even more flexibility with multiprotocol NAS environments.

Use cases

Some of the most common ways multiprotocol NAS is used include, but are not limited to:

- Home directories
- Source code repositories
- Research and engineering shares
- Image repositories
- Audio and video editing/rendering

Common challenges

Multiprotocol NAS access is desirable by many organizations for its flexibility, but there is a perception of difficulty in multiprotocol NAS that creates a specific set of challenges that are unique to the concept of sharing across protocols. This perception is grounded in reality, but only if the underlying infrastructure has not been prepared for multiprotocol NAS access. For example, standing up a Lightweight Directory Access Protocol (LDAP) server for identity management needs can greatly simplify multiprotocol NAS environments.

These challenges include, but are not limited to:

- Requirement of knowledge across multiple protocols, operating systems and storage systems.
- Working knowledge of name service servers, such as DNS, LDAP, NIS, and so on.
- External factors such as:
 - Dealing with multiple departments and IT groups (such as Windows group, and UNIX group)
 - Company acquisitions
 - Domain consolidations
 - Reorganizations
 - Many moving parts

Despite these very real challenges, multiprotocol NAS setup, configuration, and access can be simple and seamlessly integrated into any environment, provided the best practices are followed. This document will help you configure and manage multiprotocol NAS in the simplest way possible.

How multiprotocol NAS in ONTAP works: The basics

At a high level, multiprotocol NAS in ONTAP uses a combination of name mapping and permissions styles to deliver consistent data access regardless of the protocol in use. That means whether you're accessing a file from NFS or SMB, you can be assured that users with access to those files can access them, and users without access to those files cannot access them.

When a NAS client requests access to a volume in ONTAP, a series of things happen behind the scenes to provide the most transparent experience to the end user.

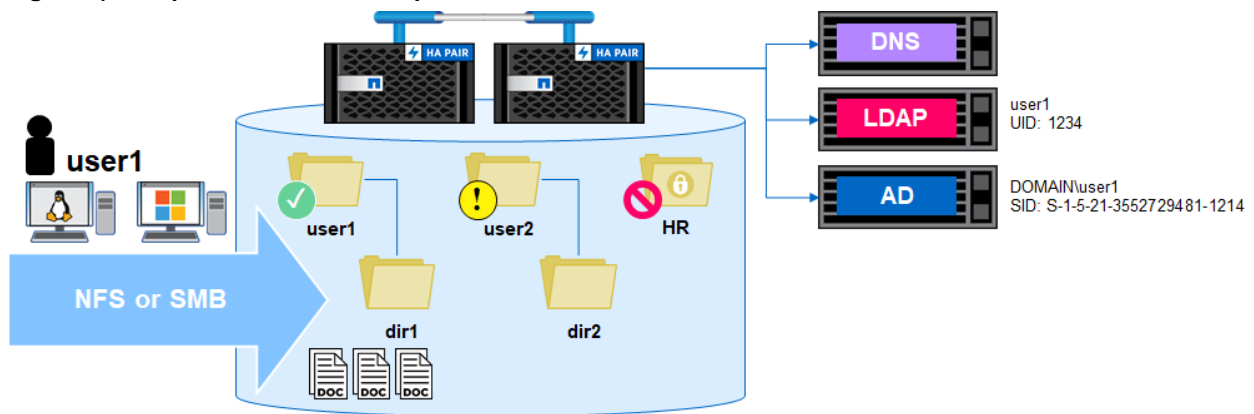
This process is controlled by how ONTAP is configured, but general concepts still apply:

1. A NAS client makes a NAS connection to the ONTAP storage VM.
2. The NAS client passes user identity information to ONTAP.
3. ONTAP checks to make sure the NAS client/user has access to the NAS share.
4. ONTAP takes that user and maps it to a valid user that ONTAP can find in its name services.
5. ONTAP uses that user to compare against the file-level permissions in the system.
6. Those permissions control the level of access the user has.

In Figure 1, user1 authenticates to a share in an ONTAP storage virtual machine (SVM) through either SMB or NFS. ONTAP finds the user in LDAP and Active Directory and maps the user 1:1. After that happens, the user is verified as user1 and gets user1's access.

In this instance, they get full control on their own folder, read access to user2's folder, and no access to the HR folder. This is all based on the ACLs specified in the file system.

Figure 1) Multiprotocol NAS basic operations.



The rest of this section covers other concepts that pertain to multiprotocol NAS access.

Namespace and file system concepts

In ONTAP, you can deploy SVMs to act as secure tenants in the cluster that can provide isolated, unique file systems for NAS clients. SVMs can have their own volumes, network interfaces, name services, and Active Directory configurations, permission models, and can act as a single namespace for NAS environments.

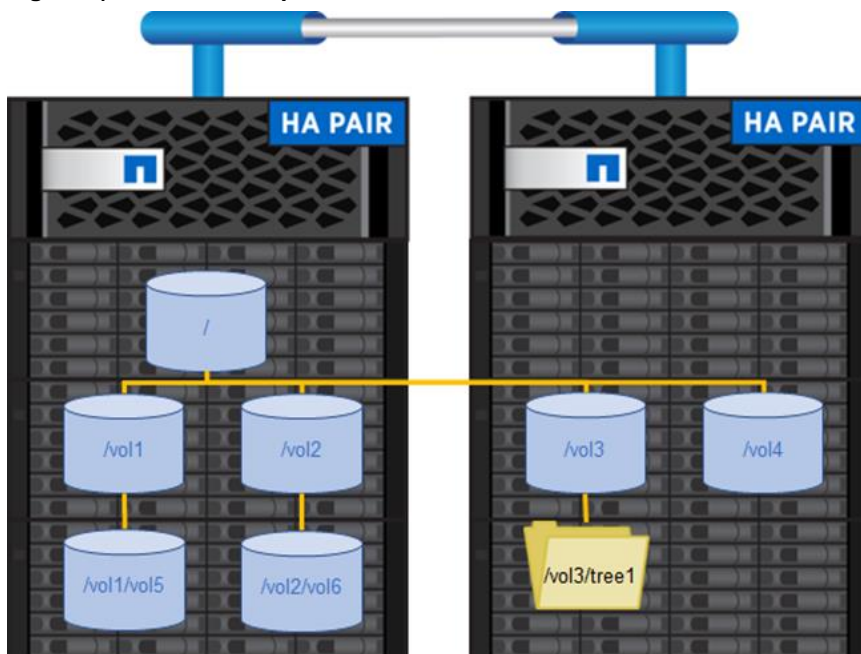
The cluster namespace

A namespace in ONTAP is a collection of file systems hosted across different nodes in the cluster to provide scalable performance and capacity. Each SVM has a file namespace that consists of a single root

volume. This namespace starts at the location of `/`. Subsequent volumes and qtrees all traverse `/` and have their export paths defined by the volume option `-junction-path`. The SVM namespace can consist of one or more volumes linked by means of junctions that connect from a named junction inode in one volume to the root directory of another volume. A cluster can have more than one SVM, but each SVM only has one vsroot and one `/`, which results in each SVM having a unique set of file system IDs. This prevents volumes in different SVMs from sharing file system IDs/file handles and avoids issues mounting NFS exports in multitenant environments.

All the volumes belonging to the SVM are linked into the global namespace in that cluster using the `/` export path. The cluster namespace is mounted at a single point in the cluster. The top directory of the cluster namespace within a cluster (`/`) is a synthetic directory containing entries for the root directory of each SVM namespace in the cluster. The volumes in a namespace can be NetApp FlexVol® volumes or FlexGroup volumes.

Figure 2) Cluster namespace.



Protecting your namespace

A vsroot volume lives only on a single node in a cluster, even though the SVM is accessible through multiple nodes. Because the vsroot is how NFS clients traverse the namespace, it is vital to NFS operations.

```
cluster::> vol offline -vserver NFS -volume vsroot
```

```
Warning: Offlining root volume vsroot of Vserver NFS will make all volumes on that Vserver inaccessible.
```

```
Do you want to continue? {y|n}: y
```

```
Volume "NFS:vsroot" is now offline.
```

If the vsroot volume is somehow unavailable, then NFS clients issues whenever the vsroot volume is needed to traverse the file system.

This process includes, but may not be limited to, the following behaviors:

- Mount requests will hang.
- If `/` is mounted, traversal (`cd`) from `/` to another volume, running operations such as `ls`, will hang.

- Unmount operations might fail because the mount is busy, even when the volume is back online.
- If a volume is already mounted below “/” (such as /vol11), then reads/writes/listings will still succeed.

Load-sharing mirrors (LS mirrors) in ONTAP is a way to leverage the NetApp ONTAP SnapMirror® capability to increase vsroot resiliency.

Note: LS mirrors are supported only with vsroot volumes. To share load across data volumes, consider using “NetApp FlexCache volumes” instead.

When LS mirrors are available for the vsroot volume, NFSv3 operations are able to leverage the LS mirror destination volumes to traverse the file system. When LS mirrors are in use, it is possible to access the source volume through the .admin folder within the NFS mount.

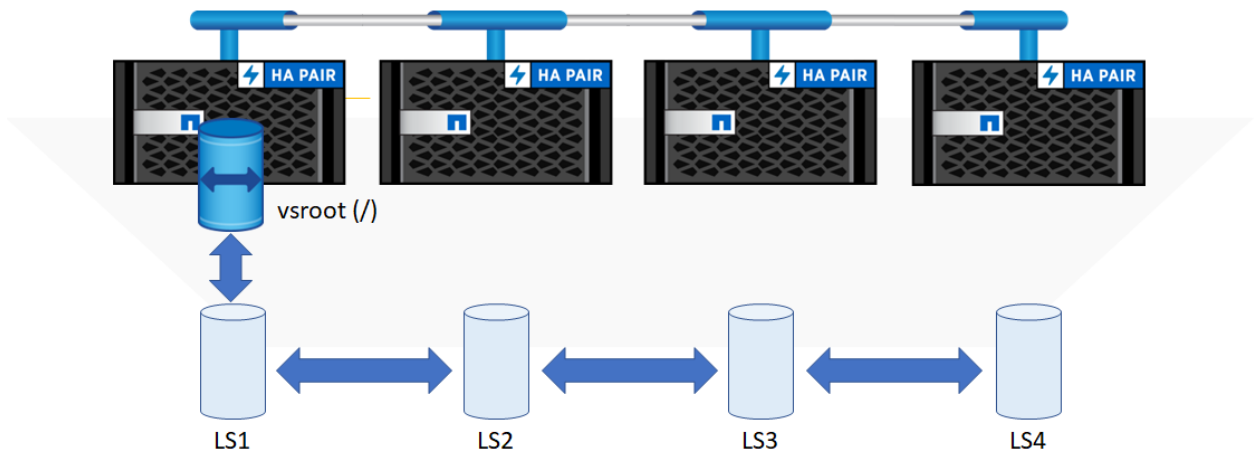
For more information, see [Creating and initializing load-sharing mirror relationships](#).

NetApp highly recommends creating LS mirror relationships for vsroot volumes in NFSv3 environments.

Note: NFSv4.x clients are unable to use LS mirror volumes to traverse file systems due to the nature of the NFSv4.x protocol.

Figure 3 shows how load-sharing mirrors can provide access to “/” in the event the vsroot is unavailable.

Figure 3) Load-sharing mirror protection of vsroot volumes.



To create a load-sharing mirror for the vsroot volume, complete the following steps:

1. Typically, the vsroot volume is 1GB in size. Verify the vsroot volume size prior to creating new volumes and make sure the new volumes are all the same size.
2. Create a destination volume to mirror the vsroot on each node in the cluster. For example, in a four-node cluster, create four new volumes with the -type DP.
3. Create a new SnapMirror relationship from the vsroot source to each new DP volume that you created. Specify a schedule for updates depending on the change rate of your namespace root. For example, hourly if you create new volumes regularly; daily if you do not.
4. Initialize SnapMirror using the `initialize-ls-set` command.

Pseudo file systems

The ONTAP architecture has made it possible to have a true pseudo file system, which complies with the [RFC 7530](#) NFSv4 standards.

Servers that limit NFS access to "shares" or "exported" file systems should provide a pseudo-file system into which the exported file systems can be integrated, so that clients can browse the

server's namespace. The clients' view of a pseudo-file system will be limited to paths that lead to exported file systems.

And in [section 7.3](#):

NFSv4 servers avoid this namespace inconsistency by presenting all the exports within the framework of a single-server namespace. An NFSv4 client uses LOOKUP and REaddir operations to browse seamlessly from one export to another. Portions of the server namespace that are not exported are bridged via a "pseudo-file system" that provides a view of exported directories only. A pseudo-file system has a unique fsid and behaves like a normal, read-only file system.

ONTAP removed the `/vol` requirement for exported volumes seen in ONTAP operating in 7-Mode; instead, it uses a more standardized approach to the pseudo file system. Because of this, you can now seamlessly integrate an existing NFS infrastructure with NetApp storage because `/` is truly `/` and not a redirector to `/vol/vol0`, as it was in 7-Mode.

A pseudo file system applies only in ONTAP if the permissions flow from more restrictive to less restrictive. For example, if the `vsroot` (mounted to `/`) has more restrictive permissions than a data volume (such as `/volname`) does, then pseudo file system concepts apply.

Having a pseudo file system allows storage administrators to create their own file system namespaces, if they desire, by way of mounting volumes to other volumes using junction paths. This concept is illustrated in Figure 2

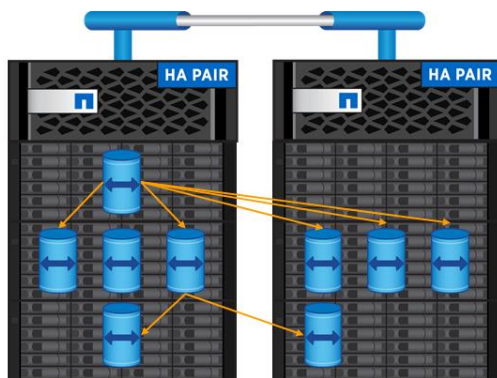
Junction paths

In ONTAP, you can create your own folder tree structure for NAS deployments by using volumes and qtrees – and even folders – as mount points within the same cluster namespace by way of junction paths.

With junction paths, you can mount a volume to another volume, a volume to a qtree, or a volume to a subdirectory. This provides granular control over the namespace for storage administrators and some flexibility for data protection and management.

Figure 4 illustrates the FlexVol design with junctioned architecture for >100TB capacity.

Figure 4) FlexVol design with junctioned architecture for >100TB capacity.



Note: Both FlexVol and FlexGroup volumes can be used in junction path architectures.

FlexVol volumes

The flexible volume, NetApp FlexVol software, was introduced in ONTAP software in 2005 as part of the Data ONTAP 7.0 (Data ONTAP operating in 7-Mode) release. The goal was to take a storage file system and virtualize it across a hardware construct to provide flexible storage administration in an ever-changing data center.

FlexVol volumes could be grown or shrunk nondisruptively and allocated to the storage operating system as [thin-provisioned containers](#) to enable overprovisioning of storage systems. Doing so allowed storage administrators the freedom to allocate space as consumers demanded it.

Qtrees

Qtrees allow a storage administrator to create folders from the ONTAP GUI or CLI to provide logical separation of data within a volume. Qtrees provide flexibility in data management by enabling unique export policies, unique security styles, quotas, and granular statistics.

Qtrees have multiple use cases and are useful for home directory workloads because qtrees can be named to reflect the user names of users accessing data, and dynamic shares can be created to provide access based on a username.

The following list provides more information about qtrees in FlexGroup volumes:

- Qtrees appear as directories to clients.
- Qtrees are able to be created at the volume level; you cannot currently create qtrees below directories to create qtrees that are subdirectories.
- Qtrees cannot be replicated using SnapMirror. SnapMirror currently is only performed at the volume level. If you want more granular replication with a volume, use [junction paths](#).
- A maximum of 4,995 qtrees is supported per volume. Quota monitoring and enforcement (enforcement in ONTAP 9.5 and later for FlexGroup volumes) can be applied at the qtree or user level.

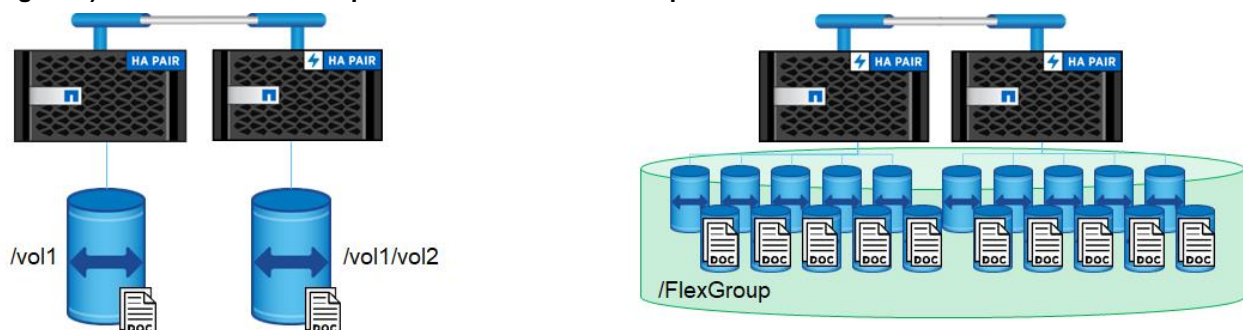
FlexGroup volumes

FlexGroup volumes took the concept of a FlexVol volume and extended it across nodes in a cluster by grouping together multiple FlexVol volumes and presenting them to clients as a single container. Doing this extended limitations well beyond 100TB and two billion files, as well as added additional performance benefits for workloads that could bottleneck on the resources a single FlexVol volume provided.

With FlexGroup volumes, a storage administrator can easily provision a massive single namespace in a matter of seconds. FlexGroup volumes have virtually no capacity or file count constraints outside of the physical limits of hardware and the total volume limits of ONTAP. Limits are determined by the overall number of constituent member volumes that work in collaboration to dynamically balance load and space allocation evenly across all members. There is no required maintenance or management overhead with a FlexGroup volume. You simply create the volume and share it with your NAS clients. ONTAP does the rest.

Figure 5 compares the architecture of the FlexVol and FlexGroup volumes.

Figure 5) FlexVol and FlexGroup volumes architecture comparison.



NetApp FlexGroup volumes: Powering the NetApp workloads

One of the truest tests of a software feature is this: Does the creator of the software use its own features?

The answer to this question is a resounding yes. NetApp leverages FlexGroup volumes in its own development environment, in NetApp Active IQ data lakes, and for use with numerous other workload use cases.

For more information about how NetApp uses FlexGroup volumes for Active IQ, see the following resources:

- [ONTAP FlexGroup Technology Powers NetApp's Massive Active IQ Data Lake](#)
- [Tech ONTAP Podcast Episode 182: NetApp on NetApp – FlexGroups and Active IQ](#)

Choosing volume styles: FlexGroup or FlexVol?

When deploying volumes to use with NFS workloads, you have two choices of available volume styles:

- **FlexVol volumes** are the standard volume type available in ONTAP and span a single node's hardware.
- **FlexGroup volumes** are volumes that are made of up multiple FlexVol member volumes spanning multiple hardware domains in a cluster that provide a number of advantages over FlexVol volumes including:
 - Volume sizes greater than 100TB (20PB tested).
 - File counts greater than two billion (400 billion tested).
 - Multithreaded metadata operations that provide 2–6x performance for high-ingest workloads.
 - Ability to use multiple nodes in a cluster to automatically balance workloads.
 - FlexVol-like management for ease of use.
 - Nondisruptive expansion when a volume reaches capacity.

In most NFS workloads, a FlexGroup volume provides more benefits over FlexVol volumes. The main caveat when deciding is to check the feature parity between the volume styles to see whether necessary features in your environment are supported. For more information about FlexGroup volumes, including a deeper look at deployment and decision points, see [TR-4571: NetApp FlexGroup Volumes Best Practices and Implementation](#).

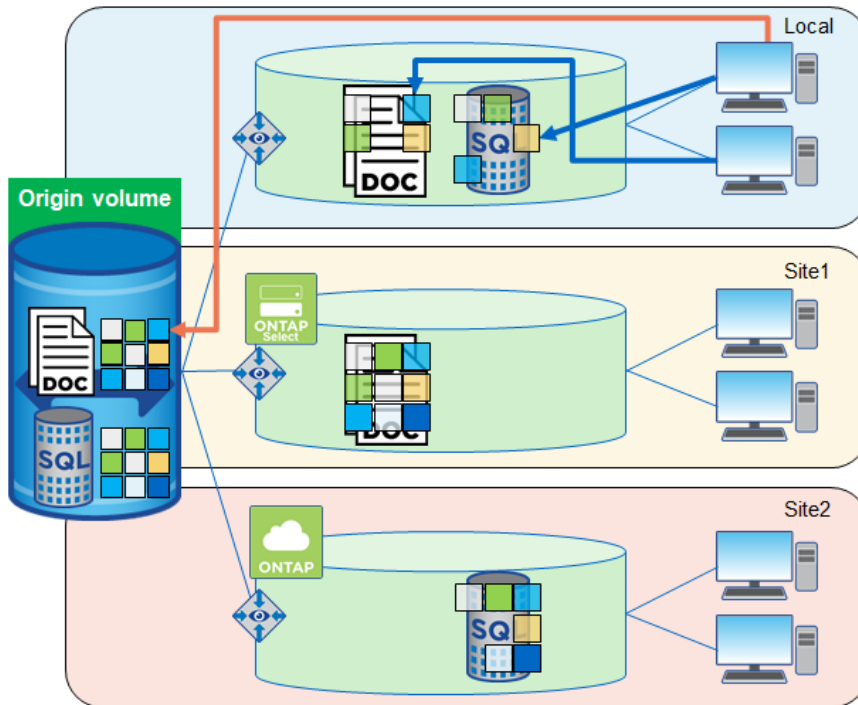
NetApp FlexCache volumes

NetApp FlexCache® technology in ONTAP provides a writable, persistent cache of a volume in a remote place that is consistent, coherent, and current.

A cache is a temporary storage location that resides between a host and a source of data. The objective of a cache is to store frequently accessed portions of source data in a way that allows the data to be served faster than it would be by fetching the data from the source. Caches are most beneficial in read-intensive environments where data is accessed more than once and is shared by multiple hosts.

Figure 6 illustrates the NetApp FlexCache volumes.

Figure 6) NetApp FlexCache volumes.



A cache can serve data faster in one of two ways:

- The cache system is faster than the system with the data source. This can be achieved through faster storage (for example, solid-state drives (SSD) versus HDD), increased processing power, or increased (or faster) memory in the platform that serves the cache.
- The storage space for the cache is physically closer to the host, so it does not take as long to reach the data.

Caches are implemented with different architectures, policies, and semantics so that the integrity of the data is protected as it is stored in the cache and served to the host.

FlexCache technology offers the following benefits:

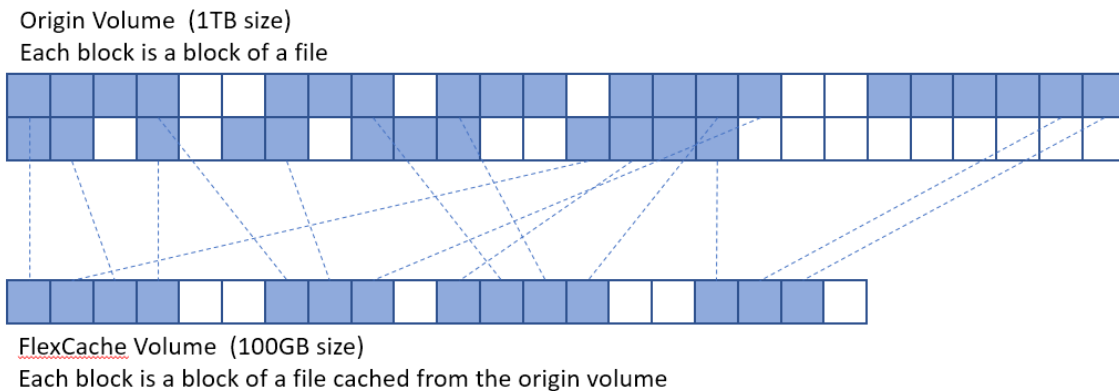
- Improved performance by providing load distribution
- Reduced latency by locating data closer to the point of client access
- Enhanced availability by serving cached data in a network disconnection situation

FlexCache technology provides all of the above advantages while maintaining cache coherency, data consistency, data currency, and efficient use of storage in a scalable and high-performing manner.

A FlexCache volume is a sparse container; not all files from the origin dataset are cached, and, even then, not all data blocks of a cached inode can be present in the cache. Storage is used efficiently by prioritizing retention of the working dataset (recently used data). Figure 7 shows the sparse volume details.

With FlexCache technology, the management of disaster recovery and other corporate data strategies only needs to be implemented at the origin. Because data management is only on the source, FlexCache technology enables better and more efficient use of resources and simpler data management and disaster recovery strategies. For EDA workloads, SemiWiki discusses how FlexCache volumes offer a way for geographically dispersed design teams can stay in sync with a current cache of the working dataset in [Concurrency and Collaboration – Keeping a Dispersed Design Team in Sync with NetApp](#).

Figure 7) Sparse volume details.



Use cases

The FlexCache in ONTAP design offers the most benefits for specific use cases, and those specific use cases are listed as ideal. Other use cases for a FlexCache volume are possible, but the benefits have not been fully vetted. In most instances, the use case is limited to the supported feature set. Non-ideal use cases are not discouraged, but you should compare the benefits of FlexCache to the costs associated with the non-ideal use case.

Ideal use cases

Because FlexCache is limited to a write-around model, it works better with workloads that are read heavy. Writes incur latency, and, when there are fewer writes, the latency does not affect the overall performance of the application accessing the dataset. Some examples include, but are not limited to, the following:

- Electronic design automation
- Media rendering
- Artificial intelligence (AI), machine learning (ML), and deep learning (DL) workloads
- Unstructured NAS data such as home directories
- Software-build environments such as Git
- Common tool distribution
- Hot volume performance balancing
- Cloud bursting, acceleration, and caching
- Stretched NAS volumes across NetApp MetroCluster™ configurations

Network access

As with any centralized storage solution, the network is a key component to providing a positive experience for end users. This section covers some network concepts in ONTAP, as well as network-adjacent concepts that are critical for NAS deployments, such as DNS.

Data LIFs

ONTAP presents IP addresses for clients through data LIFs. These are virtual IP addresses that reside on physical network ports and automatically migrate to other network ports in the event of node or port failure and can be manually migrated if you need to perform node maintenance, evacuate a node from the cluster, or simply want to change the home port of the LIF.

Data LIFs can use ifgrps or VLANs as their underlying ports and can be configured to fail over only to ports where clients have network connectivity.

For more information about data LIFs in ONTAP, see: [Configuring LIFs](#).

Data LIF considerations in NAS environments

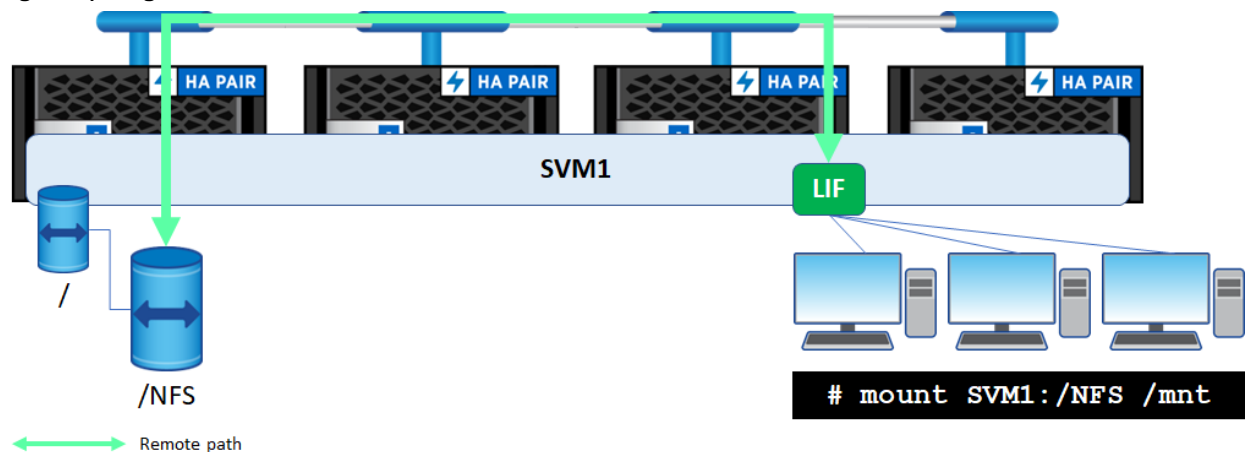
Data LIFs can live on any physical port in a cluster that is added to a valid broadcast domain. These data LIFs are configured with SVM-aware routing mechanisms that allow the correct pathing of Ethernet traffic in an SVM, regardless of where a valid data LIF lives in the cluster. When designing a network for NAS interaction, one of two approaches can be taken.

Option #1: Simplicity approach - Single LIF per SVM

Essentially, all it takes to access NAS data in ONTAP is a single network IP address that is routable to network clients. In many environments, a single network interface will suffice for NAS workloads. If the underlying physical network port fails, or if a storage node is taken over by its HA partner, then the network IP address migrates to another working port in the cluster. Using a single network interface reduces the number of IP addresses needed, but it also limits the amount of potential network bandwidth that would be available to a workload. Sending all NAS traffic to a single node in the cluster also limits the number of resources (such as CPU and RAM) available, therefore, if a workload is expected to require high throughput or connect hundreds to thousands of clients, then option #2 might be a better choice.

Figure 8 shows the single LIF NAS interaction.

Figure 8) Single LIF NAS interaction.



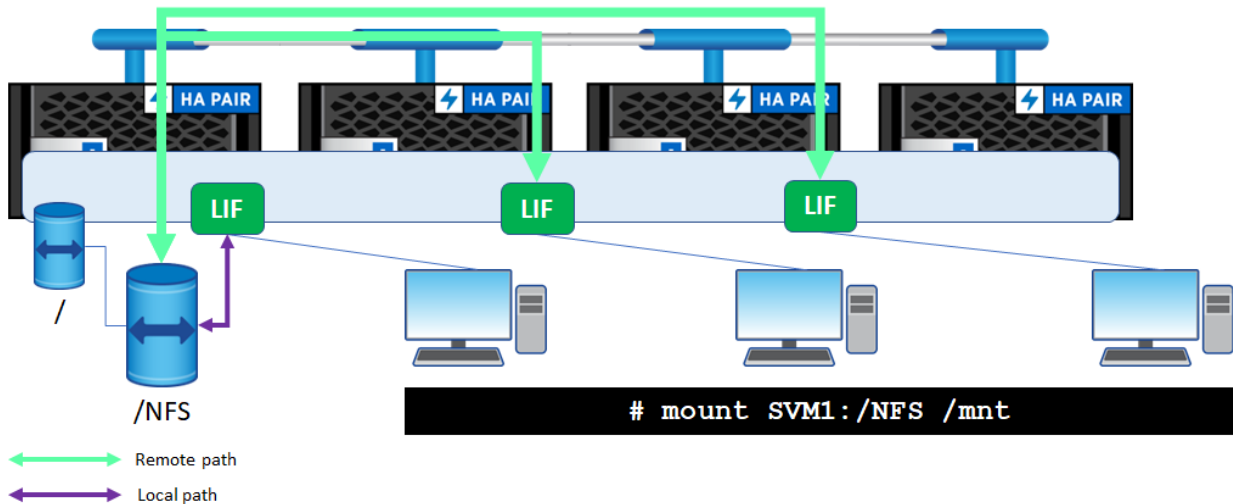
Option #2: Performance approach - Multiple data LIFs per SVM

In an ONTAP cluster, multiple nodes can be made available for NAS connectivity and storage. Remember, ONTAP clusters using NAS only can scale up to 24 nodes. Multiple nodes mean multiple physical resources, such as CPU/RAM/network interfaces. As a result, having more than one data LIF in an SVM can add considerable performance to your NAS workloads. Spreading network connections across nodes alleviates CPU and network port contention, as well as avoiding scenarios where a node might have too many TCP connections. For network load balancing of NAS connections, round robin DNS, on-box DNS, or standard load balancing hardware can be leveraged. For more information about on-box DNS (and how to configure it), see [TR-4523: DNS Load Balancing in ONTAP](#).

For situations where the best possible performance is required, or where many clients will be accessing a NAS device at the same time, creating multiple data LIFs per SVM is a sound approach. Additionally, using load balancing NAS features such as NFS referrals, CIFS autolocation, and pNFS requires a data LIF on each node where data resides.

Figure 9 shows the multiple LIF NAS interaction.

Figure 9) Multiple LIF NAS interaction.



Data LIF locality recommendations

In ONTAP, you have the ability to leverage data locality features such as NFS referrals, CIFS autolocation, and pNFS for NAS traffic regardless of where the volumes live in a cluster. For NFS referrals and CIFS autolocation, the initial TCP connection is automatically redirected to a network interface that is local to the requested volume. If the volume being used is a FlexGroup volume, then NFS referrals and CIFS autolocation should not be used.

pNFS provides a metadata path on the initial mount request, but all reads and writes are automatically redirected to local volumes through pNFS layout calls. pNFS is only available for the NFSv4.1 protocol and only with NFS clients that support it. For more information about pNFS, see [TR-4067: NFS Best Practice and Implementation Guide](#).

Without autolocation features, managing the data LIF locality to avoid the cluster network adds management complexity that might not be worth the effort, because the performance impact for most NAS workloads is negligible. Ideally, NAS connections connect to data LIFs that are local to the volumes, but with FlexGroup volumes/scale-out NAS and larger cluster backend networks, this becomes less important.

Data locality benefits and considerations

This section describes the benefits and considerations of data locality in ONTAP and how to approach these concepts with simplicity in mind.

- Ability to spread the load across nodes and leverage all the available hardware in a cluster.** When creating volumes and network interfaces, consider deploying workloads across multiple nodes in the cluster to maximize the performance headroom. Why pay for hardware you do not use?
Simplicity approach: ONTAP provides automated provisioning of storage when using ONTAP System Manager. This approach takes into account available performance headroom and attempts to place new volumes on nodes that are less utilized. Additionally, FlexGroup volumes provision across multiple nodes in a cluster and automatically balance workloads to a single namespace.
- Ability to balance network connections across multiple cluster nodes.** Clusters are single entities, as are SVMs. But they do have underlying hardware that has its own maximums, such as the number of connections and so on.

Simplicity approach: Create multiple data LIFs per SVM and mask those interfaces behind a DNS round robin name or DNS forwarding zone leveraging the ONTAP on-box DNS feature. In addition, leverage FlexGroup volumes to spread workloads across multiple nodes.

- **Ability to enable data locality in the event of a volume move.** If you move a volume to another node, you can be certain you still have a local path to the data if every node has a data LIF for the SVM. When moving volumes in ONTAP to new nodes, existing TCP connections remain in place for NAS clients. As a result, these NAS operations traverse the cluster network.

Simplicity approach: Do nothing. In most cases, NAS clients do not notice a difference in performance to these NAS shares. For NFSv4.1, consider using pNFS.

General network best practices for NAS

The following is a list of general best practices for networking in NAS environments.

Best Practice 1: Network Design with FlexGroup

When you design a NAS solution in ONTAP, consider the following networking best practices, regardless of the volume style:

- Create at least one data LIF per node, per SVM to confirm a path to each node.
- Present multiple IP addresses to clients behind a single fully qualified domain name (FQDN) by using some form of DNS load balancing. For DNS load balancing details, see [TR-4523: DNS Load Balancing in ONTAP](#).
- When possible, use LACP ports to host data LIFs for throughput and failover considerations.
- When you manually mount clients, spread the TCP connections across cluster nodes evenly. Otherwise, allow DNS load balancing to handle the client TCP connection distribution.
- For clients that do frequent mounts and unmounts, consider using [on-box DNS](#) to help balance the load. If clients are not mounted and unmounted frequently, on-box DNS does not help much.
- If the workload is that of a mount storm (that is, hundreds or thousands of clients mounting at the same time), use off-box DNS load balancing and/or consider using [NetApp FlexCache volumes](#). A mount storm to a single node can result in a denial of service to clients or performance issues.
- If you're using NFSv4.1, consider leveraging pNFS for data localization and parallel connections to files. pNFS works best with sequential I/O workloads; high metadata workloads might bottleneck over the single metadata server connection.
- For SMB3 workloads, consider enabling the multichannel and large MTU features on the CIFS server.
- If you're using jumbo frames on your network, make sure the jumbo frames are enabled at each endpoint in the network architecture; mismatched jumbo frame configurations can introduce hard-to-diagnose performance issues for any volume type.
- NFS clients can get greater performance with multiple mount points from the same client connected to the same volume in ONTAP across multiple network interfaces. However, this configuration can introduce complexity. If your NFS client supports it, use nconnect, which is covered in [TR-4067: NFS in NetApp ONTAP](#).

LACP considerations

There are valid reasons to use an LACP port on client-facing networks. A common and appropriate use case is to offer resilient connections for clients that connect to the file server over the SMB 1.0 protocol. Because the SMB 1.0 protocol is stateful and maintains session information at higher levels of the OSI stack, LACP offers protection when file servers are in an HA configuration. Later implementation of the SMB protocol can deliver resilient network connections without the need to set up LACP ports. For more information, see [TR-4100: Nondisruptive Operations with SMB File Shares](#).

LACP can provide benefits to throughput and resiliency, but you should consider the complexity of maintaining LACP environments when you are deciding. Even if LACP is involved, you should still use multiple data LIFs.

DNS load-balancing considerations

DNS load balancing (both off-box and on-box) provides a method to spread network connections across nodes and ports in a cluster. Ultimately, the decision of which method of DNS load-balancing to use comes down to the storage and network administrators' goals. For more information about DNS load balancing, see [TR-4523: DNS Load Balancing in ONTAP](#).

Best Practice 2: Use Some Form of DNS Load Balancing

When possible, use some form of DNS load balancing with multiprotocol NAS environments.

On-box DNS or off-box DNS?

ONTAP provides a method to service DNS queries through an on-box DNS server. This method factors in a node's CPU and throughput to help determine which available data LIF is the best one to service NAS access requests.

- Off-box DNS is configured by way of the DNS administrator creating multiple "A" name records with the same name on an external DNS server that provides round-robin access to data LIFs.
- For workloads that create mount-storm scenarios, the ONTAP on-box DNS server cannot keep up and balance properly, so it's preferable to use off-box DNS.

As a best practice, NetApp recommends creating at least one data LIF per node per SVM. However, to decide on how to deploy your data LIFs, review the "Data LIF considerations in NAS environments" section. If you do deploy multiple data LIFs, it is prudent to mask the IP addresses behind a DNS alias through DNS load balancing. The DNS name provides a user-friendly, easy-to-remember access point to the storage. If you plan on creating a DNS entry for multiple data LIFs and are leveraging Kerberos, make sure the DNS A/AAAA records match the assigned Kerberos SPN for the SVM, or that there is a canonical name (CNAME) that redirects to the proper A/AAAA record. Otherwise, Kerberos authentication will fail.

- For more in-depth information about DNS load balancing, including a decision matrix, see [TR-4523: DNS Load Balancing in ONTAP](#).
- For more information about NFS Kerberos and the impact of DNS names on Kerberos, see [TR-4616: NFS Kerberos in ONTAP](#).

LIF service policies

ONTAP 9.6 and later introduces [LIF service policies](#), which replace the concept of roles on network data interfaces in ONTAP. LIF policies can be applied or removed to a network interface to allow/disallow traffic without needing to recreate the network interface.

You can see which service policy your interfaces have by running the following command:

```
cluster::*> net int show -vserver DEMO -lif data -fields service-policy
(network interface show)
vserver lif    service-policy
-----
DEMO      data default-data-files
```

LIF service policies create several default policies, but you can also add custom policies. These are the following default policies, which allow SAN, NAS, or management traffic. Only one policy can be assigned to a data LIF at a time.

```
cluster::*> network interface service-policy show -vserver DEMO
Vserver   Policy                               Service: Allowed Addresses
-----
DEMO
  default-data-blocks  data-core: 0.0.0.0/0, ::/0
                        data-iscsi: 0.0.0.0/0, ::/0
                        data-fpolicy-client: 0.0.0.0/0, ::/0
  default-data-files    data-core: 0.0.0.0/0, ::/0
                        data-nfs: 0.0.0.0/0, ::/0
                        data-cifs: 0.0.0.0/0, ::/0
                        data-flexcache: 0.0.0.0/0, ::/0
                        data-fpolicy-client: 0.0.0.0/0, ::/0
  default-management    data-core: 0.0.0.0/0, ::/0
                        management-ssh: 0.0.0.0/0, ::/0
                        management-https: 0.0.0.0/0, ::/0
                        data-fpolicy-client: 0.0.0.0/0, ::/0
```

If you wish to create policies that allow only NFS or only CIFS/SMB, you can use `network interface service-policy create`, or you can add or remove services with `network interface service-policy add-service` or `network interface service-policy remove-service`. This can all be done without taking an outage.

For multiprotocol NAS, use the default-data-files policy.

For more information, see [LIFs and service policies in ONTAP 9.6 and later](#).

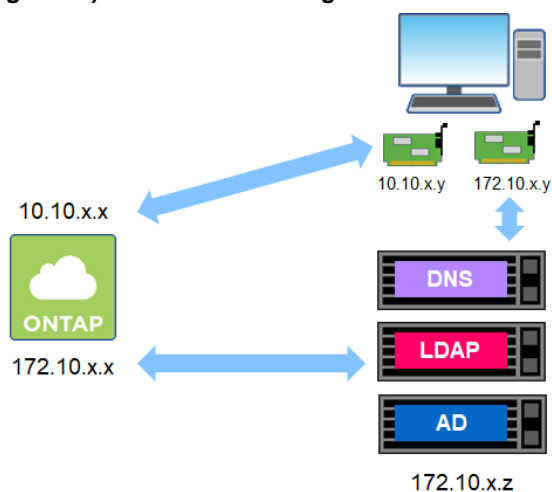
Name service connectivity

When implementing multiprotocol NAS, name services play a huge part in the functionality of the solution. As a result, your ONTAP SVM's network interfaces require network access to your name service servers. In some cases, NAS clients might live in a segmented network with multiple interfaces that have isolated connections to the storage and name services. This can be a data LIF or a management LIF dedicated for name service connectivity.

One such example is if you host your ONTAP storage in the cloud, while your NAS clients and domain services all reside on-premises. In those scenarios, you would need to provide network access to both the NAS clients and the name services.

Figure 10 shows NAS clients in a segmented network.

Figure 10) NAS clients in a segmented network.



Name service connectivity to ONTAP is needed for the following examples:

- Active Directory (for CIFS/SMB connectivity and user lookup)
- LDAP (for UNIX user and group identities and netgroups)
- DNS (for domain services)

Note: For NFS Kerberos, access is not required in order to name services. For more information, see [TR-4616: NFS Kerberos in ONTAP](#).

Access points: Volumes, shares and exports

When you provision a volume in ONTAP, you're simply carving out a portion of available capacity to use. It's not until the volume is mounted in the namespace and export policies and/or shares are created that NAS clients are able to access the volume via NFS or CIFS/SMB. This section describes some things to consider when presenting storage to NAS clients.

CIFS/SMB shares

To gain access to a volume in ONTAP through the CIFS protocol, create a CIFS share for the proper junction path of that volume. You can do this during the initial volume creation in ONTAP System Manager, or afterwards, in the CLI or System Manager.

1. To create a CIFS share during the volume creation, simply select the Share via SMB/CIFS option.

A screenshot of a configuration form titled 'Share via SMB/CIFS'. It features a checked checkbox at the top. Below it, there are two sections: 'GRANT ACCESS TO USER(S)' with a text input field containing 'Everyone', and 'PERMISSION' with a dropdown menu showing 'Full Control' and a blue downward arrow.

2. To create a CIFS after the volume has been created, click Storage > Shares > Add.

A screenshot of the 'Add Share' form. It contains several input fields: 'SHARE NAME' with 'CIFS', 'STORAGE VOLUME' with a dropdown showing 'DEMO', 'FOLDER NAME' with '/CIFS' and a 'Browse' button, and a 'DESCRIPTION' text area. At the bottom, there is an 'ACCESS PERMISSION' table.

User/Group	User Type	Access Permission
Everyone	Windows	Full Control

Below the table is a '+ Add' link.

CIFS share properties

When creating a CIFS share, you can assign different properties to the share, depending on what your application requires. In addition, you can remove properties that you no longer need.

By default, if you don't specify the share properties at share creation, the following properties are applied:

```
oplocks
browsable
changenotify
show-previous-versions
```

Note: `changenotify` might cause performance issues in high file count environments, particularly when using FlexGroup volumes. For more information, see [TR-4571: NetApp ONTAP FlexGroup Volumes](#).

Other available share properties include:

```
showsnapshot      attributecache      continuously-available
branchcache       access-based-enumeration namespace-caching
encrypt-data
```

These properties are described in the [vserver cifs share properties add](#) product document.

CIFS share ACLs

To control who can and cannot access the CIFS/SMB share, you can assign share permissions in ONTAP. This feature leverages the existing CIFS server to look up users and groups in Active Directory to properly translate the ACLs. Share permissions control the level of access a user or group can have when accessing the share, but are overridden by file and folder permissions. For example, if user1 has Full Control to the share named Documents, but has read-only access assigned to the actual folder permissions, then user1 will only have read access to the share.

To assign [share-level permissions in ONTAP](#), use ONTAP System Manager; the command line (`cifs share access-control`) or a Windows client (either share properties or the [MMC](#)).

In multiprotocol NAS environments, only CIFS/SMB clients leverage Windows share permissions. NFS clients use NFS exports for initial access to shares. CIFS/SMB export policies can also be configured in ONTAP. For more information, see “CIFS/SMB clients and export policies.”

Note: NFS and CIFS/SMB clients both honor file- and folder-level permissions.

CIFS/SMB clients and export policies

By default, ONTAP uses CIFS shares to control access for CIFS/SMB clients. However, if you want to control access by client hostnames/IP addresses/subnets rather than by users and groups, you can enable the use of export policies for CIFS shares.

For information on how to do that, see: [How export policies are used with SMB access](#).

NFS exports

Volumes in ONTAP are shared out to NFS clients by exporting a path that is accessible to a client or set of clients. When a volume is mounted to the SVM's namespace, a file handle is created and presented to NFS clients when requested in a mount command. Permissions to these exports are defined by export policies and rules, which are configurable by storage administrators.

Export policy and rule concepts

ONTAP offers export policies as containers for export policy rules to control security. These policies are stored in a replicated database, thus making exports available across every node in the cluster, rather than isolated to a single node.

To provide or restrict NFS access to these volumes, export policy rules are created. These rules can define read, write and root access, as well as specifying client lists. Multiple rules can exist in a policy and multiple clients can exist in a single rule.

The default export policy

A newly created SVM contains an export policy called Default. This export policy cannot be deleted, although it can be renamed or modified. When an NFS server is created, the Default policy is automatically created and applied to the vsroot volume. However, this Default policy does not have export rules; therefore, access to volumes using the default export policy will not be able to mount until the rules are added. When new volumes are created, if the export policy is not defined, then the export policy of the vsroot volume is inherited.

Vsroot and volume traversal

Because export policies are inherited by default, NetApp recommends opening read access to the root volume of the SVM (vsroot) to NFS clients when a rule is assigned. Setting any rules for the default export policy that restrict read access to the vsroot denies traversal to the volumes created under that SVM and will cause mounts to fail. That is because vsroot is “/” in the path to `/junction` and factors into the ability to mount and traverse.

Qtree exports

In ONTAP, it is possible to set export policies and rules for volumes, as well as underlying [qtrees](#). This offers a way to restrict/allow client access to storage-managed directories in ONTAP, which can help storage administrators more easily manage workloads such as home directories.

By default, qtrees inherit the export policy of the parent volume. You can explicitly choose or create an export policy and rule when creating qtrees in ONTAP System Manager (Figure 11), or by using the `-export-policy` CLI option.

Figure 11) Qtree export specification – ONTAP System Manager.

The screenshot shows the 'Add Qtree' window in the ONTAP System Manager interface. It contains the following fields and options:

- NAME:** A text input field containing 'SMtree'.
- VOLUME:** A dropdown menu showing 'flexvol'.
- Enable quota:** A checkbox that is checked.
- SECURITY STYLE:** A dropdown menu showing 'Inherit security style from the volume'.
- EXPORT POLICY:** Two radio buttons: 'Inherit policy from the volume' (unselected) and 'Select an existing policy' (selected). Below the selected option is a search box labeled 'EXPORT POLICY' with the text 'Search for objects.' and a dropdown arrow.
- Export policy considerations:** A blue hyperlink text.

Access control to vsroot

To control access to read/write to vsroot, use the volume `unix-permissions` and/or ACLs. NetApp recommends restricting the ability for nonowners of the volume to write to vsroot (at most, 0755 permissions).

When volumes are created, the following values are the defaults, unless specified otherwise:

- 0755 is the default UNIX security set on volumes.

- The default owner is UID 0 and the default group is GID 1.

To provide the ability to traverse vsroot that also prevents read/list access to NFS clients that might mount “/”, there are two approaches.

Option 1: Lock down UNIX mode bits on vsroot

The simplest way to lock down vsroot to users is to manage the ownership and permissions from the cluster:

1. Create a local UNIX user that is unique to the SVM. For example, this UNIX user could be the same name as the SVM itself.
2. Set the vsroot volume to the new UNIX user. Most NFS clients have a root user, which means, by default, the vsroot volume might have too much access by the root user.
3. Use UNIX permissions that limit groups and others to only traverse permissions, but leaves desired permissions for the volume owner (for example, 0611).

Option 2: Use NFSv4.x or NTFS ACLs to lock down vsroot

Another way to lock down vsroot is to leverage ACLs to limit permissions to traverse for everyone except a select few users or groups. This can be done via NFSv4.x ACLs (even if you mount using NFSv3) or via NTFS permissions in environments that are also serving CIFS/SMB protocols. For information on using NFSv4.x ACLs with NFSv3 mounts, see [TR-4067](#).

Export Policy Rules: Options

Export policy rules have multiple options available for configuration. Most export policy rule options can be viewed by using the `export-policy rule show` command or by using ONTAP System Manager.

Export policy rule options are covered in the product documentation. However, the following export policy rule options are unique to multiprotocol NAS functionality.

protocol

Use this policy to control which protocols are allowed access. The protocol options include any, NFS, NFS3, NFS4, and CIFS.

ntfs-unix-security-ops

Use this policy to control how the permission changes from NFS clients are handled on NTFS security style volumes. The options are Fail (permission changes fail with an error) or Ignore (permission changes fail silently).

allow-dev

Use this policy to control the creation/deletion of device files. However, with multiprotocol NAS, device files can only be created/deleted from NFS clients. For more information, see bug [337385](#).

Export policy rules: Inheritance

In ONTAP, the export policy rules affect only the volumes and qtrees to which they are applied. For example, if the SVM root volume has a restrictive export policy rule that limits root access to a specific client or subset of clients, the data volumes that exist under the SVM root volume (which is mounted at “/”) honor only the export policies applied to them. The one exception is if a volume has an export policy rule that denies read access to a client and that client must traverse that volume in that path. There is currently no concept of bypass traverse checking for NFS in ONTAP. For an example of an export policy rule inheritance, see [TR-4067](#).

Export policy rules: Index

ONTAP offers storage administrators the ability to set the priority for export policy rules so that they are honored in a specific order. The policy is evaluated when access is attempted and the rules are read in order from 0 to 999999999.

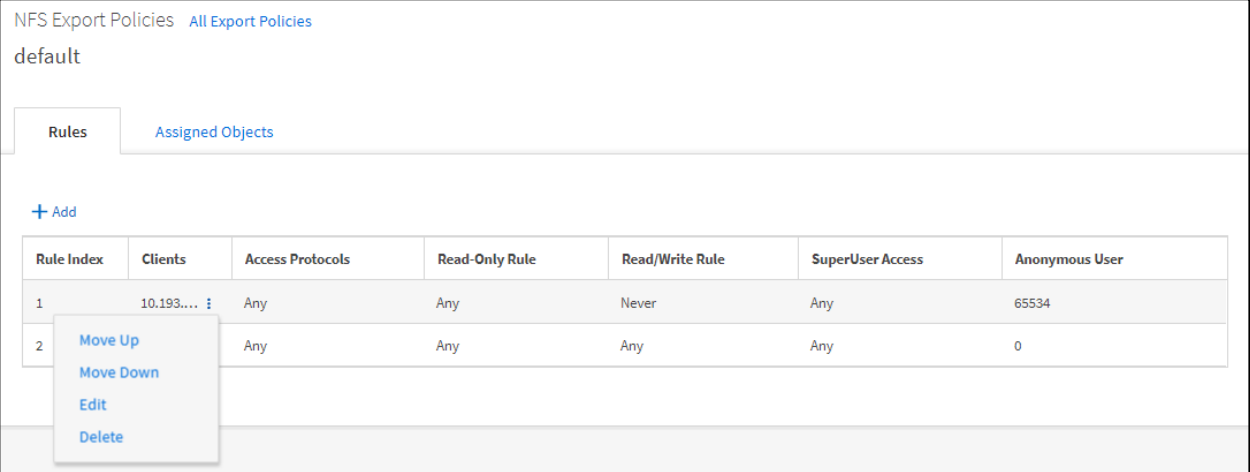
Note: A rule index of 999999999 is an absolute maximum, but NetApp does not recommend it. Use more sensible numbers for the index.

If a rule index with a higher number (such as 1) is read and has allowed access for a subnet but later a host that is in that subnet is denied access through a rule at a lower index (such as 99), then that host is granted access based on the rule that allows access being read earlier in the policy.

Conversely, if a client is denied access through an export policy rule at a higher index and then allowed access through a global export policy rule later in the policy (such as 0.0.0.0/0 client match), then that client is denied access.

It is possible to reorder the rule index for policy rules with the `export-policy rule setindex` command, or in ONTAP System Manager using Move Up/Move Down (Figure 12).

Figure 12) Reordering the Rule Index in ONTAP System Manager.



NFS Export Policies All Export Policies

default

Rules Assigned Objects

+ Add

Rule Index	Clients	Access Protocols	Read-Only Rule	Read/Write Rule	SuperUser Access	Anonymous User
1	10.193....	Any	Any	Never	Any	65534
2		Any	Any	Any	Any	0

Move Up
Move Down
Edit
Delete

It is important to consider the order of the export policy rules when determining the access that is and is not allowed for clients in ONTAP. If you use multiple export policy rules, be sure the rules that deny or allow access to a broad range of clients do not step on rules that deny or allow access to those same clients. Rule index ordering factors in when rules are read; higher-number rules override lower-number rules in the index.

Note: When you are using more granular rules (such as for a specific client, such as an administrative host), they should be placed higher up in the rule index. Broader access rules should be placed lower. For example, an administrative host rule would be at rule index 1 and a policy for 0.0.0.0/0 would be at index 99.

Export policy rules: Clientmatch

The Clientmatch option in an export policy rule that allows storage administrators to define an access list for mounting NFS exports, as well as a way to control access permissions at a high level after a client is able to mount the export.

Valid entries for the NFS export policy rule Clientmatch include:

- IP addresses
- Host names

- Domains
- Subnets
- Netgroups

Note: In ONTAP 9.1 and later, it's possible to define multiple comma-separated IP addresses or host names in a single rule, rather than needing to create unique policy rules for each.

The following considerations should be made:

- When host names are used in the clientmatch field or in netgroups, a working DNS server or manual host entries must be available to resolve the host names to IP addresses.
- When netgroups are used, an @ sign should be appended to the front of the netgroup to let ONTAP know that you are specifying a netgroup rather than a host name.
- If you're relying on name services for name resolution or netgroup lookups, make sure there is a data LIF in the SVM that can reach the necessary name services.

For more information on name services, see [TR-4668: Name Services Best Practice Guide](#).

Netgroups

Using netgroups is a way to centrally manage a large number of hosts. To control access to exports, add a single group name rather than the entire list of hosts. If you need to add or remove hosts, add or remove them through the netgroup, rather than managing export policies and rules.

ONTAP supports the use of netgroups for use with export policies through the following methods:

- Local files
- LDAP
- NIS

For information about using netgroups with NFS, see [TR-4067: Network File Systems \(NFS\) in NetApp ONTAP](#).

For information about netgroups with LDAP, see [TR-4835: How to Configure LDAP in ONTAP](#).

Note: For multiprotocol NAS, the use of netgroups apply only when using export policies and rules.

Security styles

CIFS/SMB and NFS use very different permission models for user and group access. As a result, ONTAP must be configured to honor the desired permission model for protocol access. For NFS-only environments, the decision is simple – use UNIX security styles.

When using multiprotocol NAS, keep in mind that newly created volumes inherit the security style of the SVM root volume if you do not specify the security style during creation. For example, if the SVM root (vsroot) volume is NTFS security style, new volumes will all be NTFS security style unless you use the `-security-style` option (or equivalent field in ONTAP System Manager). You can always change security styles after a volume is created, but if you create a security style incorrectly based on the defaults, you might not be aware of any issues until you start getting calls about access/permissions.

If NFS and CIFS/SMB are required, then the decision should be made based on two main concepts:

- What protocol will users manage permissions from the most?
- What is the desired permission management endpoint? In other words, do users require the ability to manage permissions from NFS clients or Windows clients? Or both?

Volume security styles are really permission styles.

ONTAP volume/qtree security styles

ONTAP offers three volume security styles to choose from for volumes and qtrees.

UNIX

The UNIX security style provides UNIX-style permissions, such as basic mode bits (Owner/Group/Everyone access with standard Read/Write/Execute permissions, such as 0755) and NFSv4.x ACLs. POSIX ACLs are not supported.

NTFS

The NTFS security style provides identical functionality as Windows SMB permissions, with granular user and groups in ACLs and detailed security/audit permissions.

Mixed

The Mixed security style takes the concepts of UNIX and NTFS security styles and applies one as an effective style based on the protocol that last modified the ACL. For example, if a Windows SMB client changes permissions of a file or folder in a mixed security style volume, then that file or folder takes on NTFS as the effective security style and applies the necessary ACLs. If an NFS client changes the permissions on that same file or folder later, then the effective security style changes to UNIX. This provides the ability for multiple clients to manage permissions and works best for applications that require this functionality.

As a best practice, NetApp does not recommend a mixed security style unless your application has a direct requirement.

Table 1 lists the limitations of existing security styles.

Table 1) Limitations of existing security styles.

Security Style	Limitations
UNIX	<ul style="list-style-type: none">• Windows clients can only set UNIX permission attributes through SMBs that map to UNIX attributes (Read/Write/Execute only; no special permissions).• NFSv4.x ACLs don't have GUI or ONTAP CLI management.• If a file or folder has NFSv4.x ACLs, the Windows GUI cannot display them.
NTFS	<ul style="list-style-type: none">• UNIX clients cannot set attributes through NFS.• NFS clients show only approximated permissions when viewing ACLs when the NFS option <code>-ntacl-display-permissive-perms</code> is disabled (default is disabled).
Mixed	<ul style="list-style-type: none">• Both Windows and UNIX clients can set attributes.• Only one style of ACL can be honored on an object.<ul style="list-style-type: none">– Applying UNIX-style ACLs drops NTFS-style ACLs.– Applying NTFS-style ACLs drops UNIX-style ACLs.• The last protocol used to modify the ACL determines the file's effective security style.

Authentication and name mapping

Authentication in NAS is how ONTAP determines that a user is who they claim to be. This authentication ensures that the expected access to files and folders is delivered, whether it is access granted or access denied.

Name mapping

After it is determined that a user is who they claim to be, the name mapping is leveraged to connect Windows user identities to UNIX user identities, because Windows and UNIX permission semantics are so different. Name mapping is done only at a user level; group names are not mapped. Instead, group memberships are gathered by ONTAP after a name mapping is done.

For more information, also see the following resources:

- [How name mapping works](#) (NFS Guide)
- [How name mapping works](#) (CIFS/SMB Guide)

Name mappings take place in the following order:

1. ONTAP checks for a 1:1 (symmetric) name mapping. For example, UNIX user `netapp` maps to Windows user `DOMAIN\netapp`.
2. If a 1:1 mapping does not exist, the name map `ns-switch` database is consulted for name service sources for name mapping. By default, local files are used (through `vserver` name-mapping rule entries), but LDAP can also be used for name map entries. For more information, see [TR-4835: How to Configure LDAP in ONTAP](#).
3. If the name mapping rules do not exist for the user, then ONTAP attempts to use the default user name set on the CIFS/SMB or NFS server. By default, CIFS/SMB uses `pcuser` as a default UNIX user (`-default-unix-user`). NFS servers have no default Windows user (`-default-win-user`) set.
4. If no user can be mapped, the NAS request fails.

Name mapping functionality based on security style

The direction in which the name mapping occurs (Windows to UNIX or UNIX to Windows) depends on which protocol is being used, but also on which security style is applied to a volume. A Windows client always requires a Windows-to-UNIX name mapping, whether or not that user is applied to review permissions depends on the security style. Conversely, an NFS client only needs to use a UNIX-to-Windows name mapping if the NTFS security style is in use.

Table 2 summarizes name mapping direction and security styles.

Table 2) Name mappings and security styles.

Protocol	Security Style	Name mapping direction	Permissions applied
CIFS/SMB	UNIX	Windows to UNIX	UNIX (mode-bits or NFSv4.x ACLs)
CIFS/SMB	NTFS	Windows to UNIX	NTFS ACLs (based on Windows SID accessing share)
CIFS/SMB	Mixed	Windows to UNIX	Depends on effective security style
NFSv3	UNIX	None	UNIX (mode-bits or NFSv4.x ACLs*)
NFSv4.x	UNIX	Numeric ID to UNIX user name	UNIX (mode-bits or NFSv4.x ACLs)
NFS	NTFS	UNIX to Windows	NTFS ACLs (based on mapped Windows user SID)
NFS	Mixed	Depends on effective security style	Depends on effective security style

* NFSv4.x ACLs can be applied using an NFSv4.x administrative client and honored by NFSv3 clients.

Local files

ONTAP SVMs can have their own unique name service configurations, which includes local files. Local files in ONTAP are not files at all, but instead are entries in a replicated database, where each node has copy. In the event of a node failure, the cluster continues normal operations because the other nodes in the cluster know what the configuration is.

In ONTAP, local files can be used for:

- UNIX users and groups
- Name mapping
- Netgroups
- DNS/host entries

Unlike external name services, local file entries do have limits on the number of entries allowed.

By default, ONTAP SVMs support up to 64,000 entries for local UNIX users and groups.

If local files are the primary name service, and more than 64,000 entries are required, then enabling scaled/file-only mode is a good option.

Limits

The following section (Table 3) covers the limits for using local users and groups in ONTAP. These limits are cluster-wide.

Table 3) Limits on local users and groups in clustered Data ONTAP.

	Local UNIX Users/Groups	Scaled-Mode Users/Groups
Local users and groups maximum entries	65,536	Users: 400K Groups: 15k Group memberships: 3000k SVMs: 6
Scaled-mode users and groups maximum file sizes	N/A	Passwd file size (users): 10MB* Group file size: 25MB* *group and passwd file sizes can be overridden with <code>-skip-file-size-check</code> but larger file sizes have not been tested

As previously mentioned, the local UNIX user and group limits are cluster-wide and affect clusters with multiple SVMs. Thus, if a cluster has four SVMs, then the maximum number of users in each SVM must add up to the maximum limit set on the cluster.

For example:

- SVM1 has 2,000 local UNIX users.
- SVM2 has 40,000 local UNIX users.
- SVM3 has 20 local UNIX users.
- SVM4 would then have 23,516 local UNIX users available to be created.

Any attempted creation of any UNIX user or group beyond the limit would result in an error message.

For example:

```
cluster::> unix-group create -vserver NAS -name test -id 12345
Error: command failed: Failed to add "test" because the system limit of {limit number}
```

```
"local unix groups and members" has been reached.
```

Scaled mode/file-only mode

The scaled mode/file-only mode feature for local users and groups in ONTAP 9.1 and later allows storage administrators to expand the limits of local users and groups by enabling a diag-level name service option and then use the load-from-uri functionality to load files into the cluster to provide a larger number of users and groups. Scaled mode/file-only mode can also add performance improvements to name service lookups, because there is no longer a need to have external dependencies on name service servers, networks, and so on. However, this performance comes at the expense of ease of management of the name services, because file management adds overhead to the storage management and introduces more potential for human error. Additionally, local file management must be done per cluster, adding an extra layer of complexity.

To enable this option for users and groups, run the `vserver services name-service unix-user file-only` and `vserver services name-service unix-group file-only` commands:

After the mode is enabled, run the following command to load the user and group file from URI:

```
cluster::*> vserver services name-service unix-user load-from-uri
```

Note: To load files larger than 10MB for users and 25MB for groups, use the `-skip-file-size-check` option.

When using file-only mode, individual operations on users and groups are not allowed. This configuration is not currently supported in MetroCluster or SVM disaster recovery (SVM DR) scenarios.

Can you still use external name services when using file-only mode?

File-only mode does not mean you cannot use LDAP or NIS as a name service; it means that management of local users and groups is done with files only (as opposed to replicated database entries). When file-only mode is enabled, LDAP and NIS lookups still work properly.

Default local users

When an SVM is created using SVM Setup or System Manager, the default local UNIX users and groups (and the default UIDs and GIDs) are created.

The following example shows these users and groups:

```
cluster::*> vserver services unix-user show -vserver vs0
      User      User  Group  Full
Vserver  Name    ID    ID    Name
-----
nfs      nobody   65535 65535 -
nfs      pcuser    65534 65534 -
nfs      root      0      0      -

cluster::*> vserver services unix-group show -vserver vs0
Vserver  Name    ID
-----
nfs      daemon  1
nfs      nobody  65535
nfs      pcuser  65534
nfs      root    0
```

Note: When using file-only mode, be sure the preceding users exist in the files being used to manage the cluster. After file-only mode is enabled, the default users are removed if the uploaded file does not include them.

Local user impact

When file-only mode is enabled, the default local users of root, pcuser, and nobody are removed if the file being loaded does not have the users. When using file-only mode, be sure to include the local users and groups in your password/group files.

Name services and external identity providers

NetApp recommends delivering the host name and user/group identities to NAS clients and ONTAP through external name services and identity providers, such as LDAP, NIS, and DNS. This best practice ensures that all endpoints involved in the NAS communication agree on who the users are, their numeric IDs, groups of which they are members, which IP addresses map to the host names, and so on, rather than needing to maintain hundreds or even thousands of local files across multiple clients and storage systems.

Centralizing name services also delivers a central management location, where removing a user from a group only has to be performed once, rather than many times across clients. This process also reduces human error that could result in outages or undesired access rights.

For information about configuring LDAP for UNIX identities and net groups, see [TR-4835: How to Configure LDAP in ONTAP](#).

Authorization and permissions

After a user is authenticated, the file and folder access is then controlled by authorization. User identities are used to populate caches with group membership information, and the file and folder permissions determine which level of access the user will get.

Access control entries and access control lists

Each file and folder in ONTAP has ACLs associated with it. These ACLs contain access control entries (ACEs) that determine the level of access users and groups have on the file or folder. Each file or folder can have up to 1,024 ACEs, but in general, it's best to use fewer ACEs on files and folders for performance and manageability. The best way to approach file and folder permissions is to use groups.

File and folder permissions in ONTAP follow the same standard rules as Windows and UNIX permission models. ONTAP supports three types of permission structures:

- [NTFS ACLs](#)
- [NFSv4 ACLs](#)
- [UNIX mode bits](#)

With multiprotocol NAS, these permissions structures are honored regardless of the type being used and the access protocol.

The type of permissions used depends on the [security style](#) in use.

ACL interaction with different security styles

The security semantics of a volume are determined by its security style and its ACL (NFSv4 or NTFS).

For a volume with UNIX security style:

- NFSv4 ACLs and mode bits are effective.
- NTFS ACLs are not effective.
- Windows clients cannot set attributes.

For a volume with NTFS security style:

- NFSv4 ACLs are not effective.
- NTFS ACLs and mode bits are effective.

- UNIX clients cannot set attributes.

For a volume with mixed security style:

- NFSv4 ACLs and mode bits are effective.
- NTFS ACLs are effective.
- Both Windows and UNIX clients can set attributes.

General best practices

This section covers a range of best practices in multiprotocol environments that help ensure the best possible results.

Multiprotocol best practices

When using multiprotocol in ONTAP, leveraging best practices can make life infinitely easier for storage administrators. The storage system is designed to integrate seamlessly into NAS environments leveraging both CIFS and NFS, provided the environment already follows best practices.

For optimal security, performance and interoperability, the best practices described in this section should be followed.

Choose a security style

ONTAP provides a variety of volume and qtree security styles for NAS file systems. A common preconception is that when using both CIFS and NFS, mixed security styles should be used. However, it's better to use either NTFS or UNIX security style volumes and qtrees in a vast majority of cases, with the exception of using mixed mode when an application vendor specifically calls for that security style or if the effective security style of a volume needs to change when a user makes permissions modifications. The design consideration should be made based on the following questions:

- What operating system/NAS protocol are a majority of clients using?
- How granular should the permissions be?
- Do the NAS clients support the latest and greatest protocol features and versions?

Use Table 4 as a decision matrix for selecting the proper volume and qtree security styles. In the table, an X represents the design consideration, with the final result in the last two columns. If both columns are selected, that means either choice is acceptable and should be chosen based on how important each of the security style features are over one another.

Table 4) Decision matrix for NAS volume and qtree security styles.

Security Style	Mostly NFS	Mostly CIFS/SMB	Need for Granular Security	Ability for Clients to Change Permissions from Any Protocol
UNIX	X	–	X (with NFSv4.x ACLs)	–
NTFS	–	X	X	–
Mixed	–	–	X	X

Note: Volume security styles and their pros and cons are described in Security styles.

Use LDAP for identity management

There are a variety of options when selecting which name service switch (ns-switch) to use. Although local files and NIS are valid options, LDAP is the recommended option for the following reasons:

- **LDAP is future-proof.** As more and more NFS clients add support for NFSv4.x, the need for NFSv4 ID domains that contain an up-to-date list of users and groups accessible from clients and storage is needed to ensure optimal security and guaranteed access when access is defined. Having an identity management server that provides 1:1 name mappings for Windows and NFS users alike greatly simplifies life for storage administrators, not just in the present, but for years to come. And when multiprotocol environments inevitably grow, storage administrators can take solace in the fact that...
- **LDAP is more scalable.** Local UNIX users and groups are capped at a soft default limit of 32,768 per cluster and can be extended to a hard limit 65,536. However, in multitenant environments with multiple SVMs, or in environments that exceed that number of users, the cluster limit will be reached and no more users will be allowed to be added. NIS servers do not have as low of a limit, but have their own issues, such as the fact that...
- **LDAP is more secure.** LDAP offers security in the form of how a storage system can connect to the LDAP server to make requests for user information. LDAP servers can allow the following bind levels when used with ONTAP:
 - Anonymous
 - Simple password
 - SASL
 - Kerberos

NIS does not offer any level of security. Passwords are weakly encrypted and sent over the wire in the clear. NIS is hard to firewall, as there is no standard port. Clients have no way to ensure that the NIS server being used is actually an NIS server.

NIS+ uses more secure encryption in line with what LDAP can do, but it's hard to set up, and if administrators are going to replace NIS, they often choose LDAP over NIS+ because...

- **LDAP is more robust.** NIS, NIS+, and local files offer basic information such as UID, GID, password, home directories, and so on. However, LDAP offers those attributes and many more. The additional attributes that LDAP uses makes multiprotocol management much more integrated with LDAP versus NIS. In fact...
- **Microsoft Active Directory is built on LDAP.** By default, Microsoft Active Directory uses an LDAP back-end for its user and group entries. However, this LDAP database does not contain UNIX style attributes. These are added when the LDAP schema is extended through Identity Management for UNIX (Windows 2003R2 and later), Service for UNIX (Windows 2003 and earlier), or third-party LDAP tools such as Centrify. Because Microsoft uses LDAP as a back-end, it makes LDAP the perfect solution for environments that choose to leverage CIFS in a domain.

For more information about LDAP in conjunction with Active Directory and ONTAP, see [TR-4073: Secure Unified Authentication](#).

For best practices with individual protocols, see [TR-4067: NFS Best Practices and Implementation Guide](#) and [TR-4191: Best Practices Guide for Windows File Services](#).

Use local files as a fail safe

In rare instances, connectivity to all configured LDAP servers can be lost. In these cases, it is important to ensure there are fail safes in place to allow administrator access to data until LDAP connectivity can be restored. As such, local UNIX users and groups matching the following example should be created.

```
cluster::> unix-user show -vserver SVM
(vserver services unix-user show)
Vserver      User      User      Group      Full
Name         Name      ID        ID          Name
-----
SVM          nobody    65535     65535
SVM          pcuser    65534     65534
SVM          root      0         1
```



```
3 entries were displayed.
```

```
cluster::> unix-group show -vserver SVM
(vserver services unix-group show)
```

Vserver	Name	ID
SVM	daemon	1
SVM	nobody	65535
SVM	pcuser	65534
SVM	root	0

```
4 entries were displayed.
```

Note: By default, these users and groups are created when CIFS is set up.

Advanced multiprotocol concepts

This section attempts to go beyond the basics of multiprotocol NAS in ONTAP. It describes more advanced topics such as NFS and CIFS server options specific to multiprotocol, common issues, troubleshooting steps, and workarounds. For a complete list of SMB/CIFS and NFS server options as they pertain to multiprotocol NAS, review the following sections:

- Appendix B: NFS server options
- Appendix C: CIFS/SMB server options

Multiprotocol NAS file locking

File locking is a way that applications can preserve the integrity of a file when it is open and in use by notifying other clients that attempt to open the file that it is currently locked. With NFS, file locking mechanisms depend on the NFS version being used. SMB locks are the same regardless of SMB version in use.

NFSv3 locking

NFSv3 uses ancillary protocols such as Network Lock Manager (NLM) and Network Status Monitor (NSM) to coordinate file locks between the NFS client and server. NLM helps establish and release locks, while NSM notifies peers of server reboots. With NFSv3 locking, when a client reboots, the server has to release the locks. When a server reboots, the client reminds the server of the locks it held. In some cases, the lock mechanisms don't communicate properly and stale locks are left over on the server and must be manually cleared.

NFSv4.x locking

NFSv4.x uses a lease-based locking model that is integrated within the NFS protocol. This means there are no ancillary services to maintain or worry about; all the locking is encapsulated in the NFSv4.x communication.

When a server or client reboots, if the lock cannot be reestablished during a specified grace period, then the lock will expire. ONTAP NFS servers control this lock timeout period with the options `-v4-grace-seconds` and `-v4-lease-seconds`.

- `-v4-lease-seconds` refers to how long a lease is granted before the client has to renew the lease. The default is 30 seconds, with a minimum of 10 seconds and maximum of -1 second of the value of `-v4-grace-seconds`.
- `-v4-grace-seconds` refers to how long a client attempts to reclaim a lock from ONTAP during a reboot of a node (such as during failovers/givebacks). The default is 45 seconds and can be modified with a range of +1 second of the `-v4-lease-seconds` value and a maximum of 90 seconds.

In rare instances, locks might not be freed as quickly as stated by the lease seconds value, which results in the locks being freed over the course of two lease periods. For example, if the grace period is set to 45 seconds, it might take 90 seconds to free the lock. For more information, see [bug 957529](#).

SMB locking

SMB uses [opportunistic locking](#), which is a way to improve performance by caching files on the local client. By caching data locally, the network traffic is reduced while a client works on a file. When the client is finished with the file, the changes are applied to the file on the server and the file is checked back in for others to edit. While the file is being edited, no one else is allowed to make changes, which provides [data coherency](#) on the file.

There are [four types](#) of opportunistic locks (oplocks):

- **Batch.** This oplock is used for files that are frequently opened and closed. A client delays sending a close request when a batch oplock is in use. If another open occurs on that file before the close request is sent, the close request is canceled. This helps improve overall performance.
- **Level 1 oplocks/exclusive locks.** A level 1 oplock is when a client caches the file locally and tracks changes in the local copy before committing to the server, under the assumption that no other clients will be making changes. This is similar to how Microsoft Office creates ~files. This improves performance by reducing the number of round trips made between the client and server.
- **Level 2 oplocks.** A level 2 oplock is when a client locks a file, but relinquishes the lock to allow other clients read/write access. Level 2 oplocks cache only reads. This is generally how OneDrive and SharePoint issue locks.
- **Filter oplocks.** A filter opportunistic lock locks a file so that it cannot be opened for either write or delete access. All clients must be able to share the file. Filter oplocks are different from level 2 oplocks in that it allows open operations for reads to occur without share violations.

In ONTAP, it is possible to configure CIFS/SMB shares to never use oplocks. One use case where disabling oplocks might make sense is in scenarios where an unreliable network connection to the storage from clients might be in place (such as SMB over a WAN or through a NAT) and older SMB versions are in use (such as SMB 1.0). Under some circumstances, if a process has an exclusive oplock on a file and a second process attempts to open the file, the first process must invalidate cached data and flush writes and locks. The client must then relinquish the oplock and access to the file. If there is a network failure during this flush, cached write data might be lost. For details on this and how to manage oplocks, see the [ONTAP 9 Documentation Center section on oplocks](#).

Note: The most recent versions of the SMB protocol have features that help mitigate the impact of network instability to SMB shares and locks, such as persistent file handles.

Multiprotocol NAS lock behavior

When using file locks in multiprotocol NAS environments, be aware of the difference in behavior depending on the NAS protocol in use:

- If the NAS client is SMB, file locks are mandatory locks.
- If the NAS client is NFS, file locks are advisory locks.

What this means

Because of the differences between the NFS and SMB file locks, an NFS client might fail to access a file previously opened by an SMB application.

The following activity occurs when an NFS client attempts to access a file locked by an SMB application:

- In mixed or NTFS volumes, file manipulation operations such as `rm`, `rmdir`, and `mv` can cause the NFS application to fail.

- NFS read and write operations are denied by SMB deny-read and deny-write open modes, respectively.
- NFS write operations fail when the written range of the file is locked with an exclusive SMB bytelock. In UNIX security-style volumes, NFS unlink and rename operations ignore SMB lock state and allow access to the file. All other NFS operations on UNIX security-style volumes honor SMB lock state.

Lock types

The various types of NAS locks include:

- **Shared locks.** Shared locks can be used by multiple processes at the same time and can only be issued if there are no exclusive locks on a file. These locks are intended for read-only work, but can be used for writes (such as with a database).
- **Exclusive locks.** These locks operate the same as exclusive locks in CIFS/SMB; however, only one process can use the file when there is an exclusive lock. If any other processes have locked the file, an exclusive lock cannot be issued, unless that process was [forked](#).
- **Delegations.** Delegation locks are used only with NFSv4.x and are assigned when the NFS server options are enabled and the client supports NFSv4.x delegations. Delegations provide a way to cache operations on the client side by creating a soft lock to the file being used by a client. This process helps improve some aspects of performance for operations by reducing the number of calls being made between the client and server and are similar to SMB opportunistic locks. For more information about NFS delegations, see [TR-4067: Network File Systems \(NFS\) in NetApp ONTAP](#).
- **Byte-range locks.** Rather than locking an entire file, byte-range locks only lock a portion of a file.
- **Opportunistic locks.** This lock is the standard way that SMB locks files. For details, see “SMB locking.”

Note: NFS locking behavior is dependent on the type of lock, the client OS version, and the NFS version being used. Be sure to test locking in your environment to gauge the expected behavior.

For more information about file locking in ONTAP, see the product documentation section called [“Managing file locks.”](#)

Manually establishing locks on an NFS client

To test NFS locks, the client has to tell the NFS server to establish a lock. However, not all applications use locks. For example, an application such as “vi” does not lock a file; instead, it creates a hidden swap file in the same folder and then commits writes to that file when the application is closed. Then the old file is deleted and the swap file gets renamed to the file name.

However, there are utilities to manually establish locks. For example, [flock](#) can lock files. To establish a lock on a file, complete the following steps:

1. Run `exec` to assign a numeric ID.

```
# exec 4<>v4user_file
```

2. Use `flock` to create a shared or exclusive lock on the file.

```
# flock

Usage:
flock [options] <file|directory> <command> [command args]
flock [options] <file|directory> -c <command>
flock [options] <file descriptor number>

Options:
-s --shared          get a shared lock
-x --exclusive      get an exclusive lock (default)
-u --unlock          remove a lock
-n --nonblock        fail rather than wait
```

```
-w --timeout <secs>      wait for a limited amount of time
-E --conflict-exit-code <number>  exit code after conflict or timeout
-o --close               close file descriptor before running command
-c --command <command>   run a single command string through the shell

-h, --help              display this help and exit
-V, --version            output version information and exit

# flock -n 4
```

3. Check the ONTAP SVM for the lock.

```
cluster::*> vserver locks show -vserver DEMO
```

Notice: Using this command can impact system performance. It is recommended that you specify both the vserver and the volume when issuing this command to minimize the scope of the command's operation. To abort the command, press Ctrl-C.

```
Vserver: DEMO
Volume  Object Path          LIF          Protocol  Lock Type  Client
-----
home    /home/v4user_file          data2        nlm       byte-range  10.x.x.x
        Bytelock Offset (Length): 0 (18446744073709551615)
```

4. Unlock the file:

```
# flock -u -n 4
```

Manually locking files allows you to test the file open and edit the interactions, as well as to see how file locks handle storage failover events.

Special character considerations

Most common text characters in Unicode (when they are encoded with UTF-8 format) use encoding that is equal to or smaller than three bytes. This common text includes all modern written languages, such as Chinese, Japanese, and German. However, with the popularity of special characters such as the [emoji](#), some UTF-8-character sizes have grown beyond three bytes. For example, a [trophy symbol](#) is a character that requires four bytes in UTF-8 encoding.

Special characters include, but are not limited to, the following:

- Emojis
- Music symbols
- Mathematical symbols

When a special character is written to a FlexGroup volume, the following behavior occurs:

```
# mkdir /flexgroup4TB/🏆
mkdir: cannot create directory '/flexgroup4TB/\360\237\217\206': Permission denied
```

In the preceding example, `\360\237\217\206` is hex `0xF0 0x9F 0x8F 0x86` in UTF-8, which is a trophy symbol.

ONTAP software did not natively support UTF-8 sizes that were greater than three bytes in NFS, as indicated in [bug 229629](#). To handle the character sizes that exceeded three bytes, ONTAP placed the extra bytes into an area in the operating system known as `bagofbits`. These bits were stored until the client requested them. Then the client interpreted the character from the raw bits. FlexVol technology supports `bagofbits`, and FlexGroup volumes added support for `bagofbits` in ONTAP 9.2.

Best Practice 3: Special Character Handling – Recommended ONTAP Version

For optimal special character handling, use ONTAP 9.5 or later and the `utf8mb4` volume language.

Also, ONTAP has an event management system message for issues with `bagofbits` handling, which includes how to identify the offending file ID.

Message Name: `waf1.bagofbits.name`
Severity: ERROR

Corrective Action: Use the "volume file show-inode" command with the file ID and volume name information to find the file path. Access the parent directory from an NFSv3 client and rename the entry using Unicode characters.

Description: This message occurs when a read directory request from an NFSv4 client is made to a Unicode-based directory in which directory entries with no NFS alternate name contain non-Unicode characters.

Support for utf8mb4 volume language

As previously mentioned, special characters might exceed the supported three bytes UTF-8 encoding that is natively supported. ONTAP then uses the `bagofbits` functionality to allow these characters to work.

This method for storing inode information is not ideal; therefore, starting in ONTAP 9.5, `utf8mb4` volume language support was added. When a volume uses this language, special characters that are four bytes in size are stored properly and not in `bagofbits`.

Volume language is used to convert names sent by NFSv3 clients to Unicode, and to convert on-disk Unicode names to the encoding expected by NFSv3 clients. In legacy situations in which NFS hosts are configured to use non-UTF-8 encodings, you should use the corresponding volume language. The use of UTF-8 has recently become almost universal; therefore, the volume language is most likely to be UTF-8.

NFSv4 requires the use of UTF-8, so there is no need to use non-UTF-8 encoding for NFSv4 hosts. Similarly, CIFS uses Unicode natively, therefore, it works with any volume language. However, use of `utf8mb4` is recommended because files with Unicode names above the basic plane are not converted properly on non-`utf8mb4` volumes.

Volume language can only be set on a volume at creation by using the `-language` option. You cannot convert a volume's language. To use files with a new volume language, create the volume and migrate the files by using a utility such as the XCP Migration Tool.

Best Practice 4: UTF-8 or utf8mb4?

If you're running ONTAP 9.5 or later, it is best to use the `utf8mb4` volume language to help prevent issues with file name translation unless clients are unable to support the language.

Qtree considerations

Qtrees are a way for storage administrators to present to end users the ONTAP-managed folders that reside inside of a volume and provide the following features:

- Quota monitoring and enforcement
- Unique export policies and rules
- Unique security styles
- Qtree quality of service (QoS)
- Qtree statistics

Going forward, qtrees are considered the chosen data management point in ONTAP, and they will continue to see more enhancements.

This section focuses on some considerations you should make when using qtrees.

Qtrees and file moves

A qtree is considered a unique file system in ONTAP. Although it looks like a directory from a NAS client perspective, some operations might behave differently than if they were an actual directory. One example of this scenario is moving a file between qtrees in the same volume.

- When a file move is performed in a volume across directories, the file is simply renamed to a new name. This process happens within seconds because that is a move inside of the same file system.
- When a file move occurs between two qtrees, the file is copied to the new location rather than being renamed. This process causes the operation to take much longer. This file move behavior occurs whether the qtree lives in a FlexVol or a FlexGroup volume.

Qtree IDs and rename behavior

After a noninherited export policy is applied to a qtree, NFS file handles change slightly when dealing with operations between qtrees. ONTAP validates qtree IDs in NFS operations, which impacts things such as file renames/moves when moving to or from a qtree in the same volume as the source folder or qtree. This is considered a security feature, which helps prevent unwanted access across qtrees, such as in home directory scenarios. However, simply applying export policy rules and permissions can achieve similar goals.

For example, a move or rename to or from a qtree in the same volume results in access denied. The same move or rename to or from a qtree in a different volume results in the file being copied. With larger files, the copy behavior can make it seem like a move operation is taking an unusually long time, where most move operations are near-instantaneous, because they are simple file renames when in the same file system/volume.

How the renames in qtrees behave is controlled by the Advanced Privilege option, which is described in the NetApp Knowledge Base article [Permission denied while moving files between qtrees when NFS option '-validate-qtree-export' is enabled](#).

This article describes how the following behaviors are of different operations:

Assuming that file permissions allow and that client is allowed by export policies to access both source and destination volume/qtree, these are the current permutations with the 'validate-qtree-export' flag enabled or disabled:

Enabled:

- Rename in same volume and qtree: SUCCESS
- Rename in same volume, different qtrees: EACCESS
- Rename between volumes where qtree IDs differ: EACCESS
- Rename between volumes where qtree IDs match: XDEV

Disabled:

- Rename in same volume and qtree: SUCCESS
- Rename in same volume, different qtrees: SUCCESS
- Rename between volumes where qtree IDs differ: XDEV
- Rename between volumes where qtree IDs match: XDEV

Note: NFS3ERR_XDEV and NFS3ERR_ACCESS are defined in [RFC-1813](#).

To change the behavior of renames/moves across qtrees, modify `-validate-qtree-export` to disabled. For more information, see [Validating qtree IDs for qtree file operations](#).

Note: Other than allowing renames across qtrees, there is no known negative impact to disabling the `-validate-qtree-export` option.

File handle impact for qtree exports

Normally, the NFS export file handles that are handed out to clients are 32 bytes or less in size. However, with qtree exports, an extra few bytes are added to create 40-byte file handles. In most clients, this byte size is not an issue, but older clients ([such as HPUX 10.20, introduced in 1996](#)) might have problems

mounting these exports. Be sure to test older client connectivity in a separate test SVM before enabling qtree exports, because there is currently no way to change the file handle behavior after qtree exports are enabled.

Mounting multiple qtrees in the same volume on the same NFS client

Although qtrees effectively act as independent file systems, if they live in the same volume, the NFS conversation between client and server will involve the same MSID/file handle from the parent volume. This can result in the NFS client seeing the qtrees as the same file system mounted twice and the used space will be the same regardless of what is actually being used in each qtree.

For example, these two qtrees are mounted to the same client at different mount points.

```
# mount | grep qtree
10.193.67.214:/testvol/qtree1 on /mnt/qtree1 type nfs
10.193.67.214:/testvol/qtree2 on /mnt/qtree2 type nfs
```

They both show the same space usage before we copy a file.

```
# df -h | grep qtree
10.193.67.214:/testvol/qtree1 973G 2.0M 973G 1% /mnt/qtree1
10.193.67.214:/testvol/qtree2 973G 2.0M 973G 1% /mnt/qtree2
```

Then, we copy a 3.8GB file to qtree1. Both qtrees show the same space used.

```
# cp debian-8.2.0-amd64-DVD-1.iso /mnt/qtree1/
# df -h | grep qtree
10.193.67.214:/testvol/qtree1 973G 3.8G 970G 1% /mnt/qtree1
10.193.67.214:/testvol/qtree2 973G 3.8G 970G 1% /mnt/qtree2
```

We can get around this by applying a simple monitoring quota to one of the qtrees. Just by doing this, the proper space usage is seen.

```
cluster::*> quota report -vserver NFS
Vserver: NFS
```

Volume	Tree	Type	ID	-----Disk----- Used Limit	-----Files----- Used Limit	Quota Specifier
testvol	qtree1	tree	1	3.73GB -	2 -	qtree1
testvol	qtree2	tree	2	0B -	1 -	qtree2
testvol		tree	*	0B -	0 -	*

```
# df -h | grep qtree
10.193.67.214:/testvol/qtree1 973G 3.8G 970G 1% /mnt/qtree1
10.193.67.214:/testvol/qtree2 970G 0 970G 0% /mnt/qtree2
```

Subdirectory exports

Qtrees can be exported through NFS, which provides a single-level subdirectory path to define unique export policies and rules for clients. However, individual directories cannot have export policies and rules applied to them, and qtrees currently can only be created at the volume level in ONTAP. If your environment requires exports lower in the directory tree, a combination of volumes, qtrees, and junction paths can be used to simulate subdirectory exports. However, this does not secure the entire path, because each level in the junction path has to allow read access to the export policy rules for the clients to allow traversal.

For example, you could create a subdirectory export similar to this:

```
/volume1/qtree1/volume2/qtree2/volume3/qtree3
```


Each object in the path above can be exported to NFS clients with unique policies and rules. For greater levels of security for these folders, consider using NTFS security styles/ACLs or Kerberos for NFS.

User and group owners

Starting in ONTAP 9.8, you can set the user and group owner of a qtree from the ONTAP CLI with `qtree create` or `qtree modify`. In previous releases, this was setting done through the NAS protocol from a client. This setting is only currently available through the CLI or REST API. There is no ZAPI or ONTAP System Manager support.

[-user <user name>]	User ID
[-group <group name>]	Group ID

Managing quotas

ONTAP supports [user/group and tree quotas](#) for use with NAS operations. Quotas provide a method for storage administrators to monitor and control space and file count usage in the storage system.

FlexGroup volumes also support quotas. The level of support for these quotes can be broken down into the following categories:

- Support for quota reporting in ONTAP 9.3.
- Support for FPolicy, which can provide quota enforcement from third-party vendors, such as DefendX (formerly NTP) in ONTAP 9.4.
- Enforcement of quotas (that is, setting hard and soft limits for capacity and file count) is supported in ONTAP 9.5 and later.

User and group quota considerations

To implement user or group quotas, the cluster must be able to resolve the specified user name or group. This requirement means that the user or group must exist locally on the SVM or within a resolvable name service server, such as Active Directory, LDAP, or NIS. If a user or group cannot be found by the SVM, then the quota rule is not created. If a user or group quota fails to create because of an invalid user, the command line issues the following error:

```
Error: command failed: User name user not found. Reason: SecD Error: object not found.
```

ONTAP System Manager delivers a similar message. Use the `event log show` command to investigate the issue further. For more information about configuring name services for identity management in ONTAP, see [TR-4835: How to Configure LDAP in ONTAP](#) and [TR-4668: Name Services Best Practices Guide](#).

Creating a user or group quota

User and group quotas can be created to report or enforce capacity or file count limits on a per-user basis. These quotas would be used in scenarios where multiple users or groups share the same namespace or qtree. These steps are the same for FlexVol and FlexGroup volumes.

Creating a quota – ONTAP System Manager

To create a user or group quota in ONTAP System Manager, navigate the left menu to Storage > Quotas. A page with three tabs is displayed: Reports, Rules, and Volume Status.

Reports show you the current quota tracking for users, groups, and qtrees.

Figure 13) Quota reports – ONTAP System Manager.

Quotas

Reports Rules Volume Status

DEMO X Download Show / Hide Filter

Type	Volume	Storage VM	Qtree	Users	Group	% Space Used	% Files Used
user	home	DEMO	-	root	-	4.65 GB used No Hard Limit	25 used No Hard Limit
user	home	DEMO	-	14	-	4 KB used No Hard Limit	2 used No Hard Limit
user	home	DEMO	-	apache	-	383 MB used No Hard Limit	2 used No Hard Limit
user	home	DEMO	-	Podcast	-	0 Bytes used No Hard Limit	2 used No Hard Limit
user	home	DEMO	-	admin	-	4.65 GB used No Hard Limit	2 used No Hard Limit
user	home	DEMO	-	BUILTIN\Administrat...	-	0 Bytes used No Hard Limit	15 used No Hard Limit
user	home	DEMO	-	squash	-	0 Bytes used No Hard Limit	3 used No Hard Limit
user	home	DEMO	-	1003	-	12 KB used No Hard Limit	5 used No Hard Limit
user	home	DEMO	-	prof1	-	0 Bytes used No Hard Limit	11 used No Hard Limit
user	home	DEMO	-	1108	-	0 Bytes used No Hard Limit	1 used No Hard Limit

Volume status shows whether quotas are on or off for the volume.

Figure 14) Quota volume status – ONTAP System Manager.

Quotas

Reports Rules Volume Status

Tech_ONTAP X Download Show / Hide Filter

Volume Name	Status	Quota Rules
Tech_ONTAP	Off	0 rules

Rules is where you would create new quotas for users, groups, or qtrees. Click Add and enter the information for the user, group, or qtree quota in the dialog box. After the rule is created, ONTAP System Manager performs all of the necessary steps to enable and activate the quota.

Figure 15) Quota rules – ONTAP System Manager.

Add Quota

QUOTA TARGET

Tech_ONTAP

podcast_tree

If your quota target is a volume, leave qtree blank.

☒ Enable Quota

QUOTA TYPE

☒ Qtree
Enforce usage limits for a qtree within a volume.

☐ User
Enforce usage limits for all users or a specific user.

☐ Group
Enforce usage limits for all groups or a specific group.

Quota Limit

Space Limit

HARD LIMIT

600

GB

SOFT LIMIT

300

GB

File Limit

HARD LIMIT

9

Hundred

SOFT LIMIT

6

Hundred

Save

Cancel

Quotas

ReportsRulesVolume Status

+ Add

SearchDownloadShow / HideFilter

Type	Volume	Storage VM	Qtree	Users	Group	Space Limit (Soft/Hard)	Files Limit (Soft/Hard)
tree	Tech_ONTAP	DEMO	podcast_tree	-	-	300 GB / 600 GB	600 / 900
tree	Tech_ONTAP	DEMO	All Qtrees			Unlimited / Unlimited	Unlimited / Unlimited

Quotas

ReportsRulesVolume Status

SearchDownloadShow / HideFilter

Type	Volume	Storage VM	Qtree	Users	Group	% Space Used	% Files Used
tree	Tech_ONTAP	DEMO	podcast_tree	-	-	<div><div></div></div> 0%	<div><div></div></div> 0%

Creating a user or group quota – CLI

To create a user or group reporting quota for a specific user or group by using the CLI, run the following command at the admin privilege level:

```
cluster::> quota policy rule create -vserver SVM1 -policy-name default -volume flexgroup -type
[user|group] -target [username or groupname] -qtree ""
```

To create a user or group reporting quota for all users or groups by using the CLI, run the following command at the admin privilege level. The target is provided as an asterisk to indicate all:

```
cluster::> quota policy rule create -vserver SVM1 -policy-name default -volume flexgroup -type
[user|group] -target * -qtree ""
```

Creating a tree reporting quota from the CLI

To create a tree reporting quota for a specific user or group by using the CLI, run the following command at the admin privilege level:

```
cluster::> quota policy rule create -vserver DEMO -policy-name tree -volume flexgroup_local -type
tree -target qtree
```

To enable quotas, use `quota on` or `quota resize`.

```
cluster::> quota on -vserver DEMO -volume flexgroup_local
[Job 9152] Job is queued: "quota on" performed for quota policy "tree" on volume
"flexgroup_local" in Vserver "DEMO".

cluster::> quota resize -vserver DEMO -volume flexgroup_local
[Job 9153] Job is queued: "quota resize" performed for quota policy "tree" on volume
"flexgroup_local" in Vserver "DEMO".

cluster::> quota show -vserver DEMO -volume flexgroup_local

      Vserver Name: DEMO
      Volume Name: flexgroup_local
      Quota State: on
      Scan Status: -
      Logging Messages: -
      Logging Interval: -
      Sub Quota Status: none
      Last Quota Error Message: -
      Collection of Quota Errors: -
      User Quota enforced: false
      Group Quota enforced: false
      Tree Quota enforced: true
```

The following example shows a `quota report` command on a volume with a tree quota specified:

```
cluster::> quota report -vserver DEMO -volume flexgroup_local
Vserver: DEMO
```

Volume	Tree	Type	ID	----Disk----		----Files-----		Quota Specifier
				Used	Limit	Used	Limit	
flexgroup_local	qtree	tree	1	0B	-	1	-	qtree

The used files and disk space are monitored and incremented as new files are created.

```
cluster::> quota report -vserver DEMO -volume flexgroup_local
Vserver: DEMO
```

Volume	Tree	Type	ID	----Disk----		----Files-----		Quota Specifier
				Used	Limit	Used	Limit	
flexgroup_local	qtree	tree	1	13.77MB	-	4	-	qtree

Quota enforcement example

When quota enforcement is enabled on a qtree or for a user/group, ONTAP disallows new file creations or writes after a quota is exceeded. This helps storage administrators have greater control over how much data is being written to a volume or qtree.

In addition, when a quota is exceeded, an event management system message is logged at the DEBUG severity level to notify storage administrators of the quota violation. You can configure these messages so that the system forwards them as SNMP traps or as syslog messages.

In this example, a quota has been set with a hard limit of 1GB and 10 files.

```
cluster::*> quota policy rule show -vserver DEMO
```

Vserver: DEMO			Policy: tree			Volume: flexgroup_local		
Type	Target	Qtree	User Mapping	Disk Limit	Soft Disk Limit	Files Limit	Soft Files Limit	Threshold
tree	qtree	""	-	1GB	-	10	-	-

When a user tries to copy a 1.2GB file to the qtree, ONTAP reports an out of space error.

```
[root@centos7 qtree]# cp /SANscreenServer-x64-7.3.1-444.msi /FGlocal/qtree/  
cp: failed to close '/FGlocal/qtree/SANscreenServer-x64-7.3.1-444.msi': No space left on device
```

The file is partially written, but it is unusable because it is missing data.

```
# ls -alh  
total 1.1G  
drwxr-xr-x  2 root root  4.0K Jul 19 15:44 .  
drwxr-xr-x 11 root root  4.0K Jun 28 15:10 ..  
-rw-r--r--  1 root root    0 Dec 12  2017 newfile1  
-rw-r--r--  1 root root    0 Dec 12  2017 newfile2  
-rw-r--r--  1 root root 1021M Jul 19  2018 SANscreenServer-x64-7.3.1-444.msi
```

ONTAP then reports the quota as exceeded.

```
cluster::*> quota report -vserver DEMO  
Vserver: DEMO
```

Volume	Tree	Type	ID	---Disk---		---Files---		Quota
				Used	Limit	Used	Limit	Specifier
flexgroup_local	qtree	tree	1	1.01GB	1GB	5	10	qtree

The same behavior occurs for file count limits. In this example, the file count limit is 10 and the qtree already has five files. An extra five files meets our limit.

```
[root@centos7 /]# su student1  
sh-4.2$ cd ~  
sh-4.2$ pwd  
/home/student1  
sh-4.2$ touch file1  
sh-4.2$ touch file2  
sh-4.2$ touch file3  
sh-4.2$ touch file4  
sh-4.2$ touch file5  
touch: cannot touch 'file5': Disk quota exceeded
```

```
cluster::*> quota report -vserver DEMO  
Vserver: DEMO
```

---Disk---		---Files---		Quota
------------	--	-------------	--	-------

Volume	Tree	Type	ID	Used	Limit	Used	Limit	Specifier
flexgroup_local	qtree	tree	1	1.01GB	1GB	5	10	qtree
home		user	student1	NTAP\student1	4KB	1GB	10	student1

2 entries were displayed.

From the event logs, the quota violations is displayed.

```
cluster::*> event log show -message-name quota.exceeded
Time                Node                Severity          Event
-----
7/19/2018 16:27:54  node02                DEBUG             quota.exceeded: ltype="hard", volname="home",
app="", volident="@vserver:7e3cc08e-d9b3-11e6-85e2-00a0986b1210", limit_item="file",
limit_value="10", user="uid=1301", qtree="treeid=1", vfiler=""
7/19/2018 15:45:02  node01                DEBUG             quota.exceeded: ltype="hard",
volname="flexgroup_local", app="", volident="@vserver:7e3cc08e-d9b3-11e6-85e2-00a0986b1210",
limit_item="disk", limit_value="1048576", user="", qtree="treeid=1", vfiler=""
```

Quota scan completion times

When a quota initialization or resize takes place, ONTAP must perform some background tasks to complete the necessary work to reflect quota usage accurately. These tasks take time, which depends on a number of factors covered below.

Initialization completion time

The time it takes for quotas to initialize on a volume or qtree depends on the following factors:

- **The number of files and folders in a volume.** More files mean a longer initialization, while file size does not affect the initialization time.
- **Type of volume.** FlexVol volume scans can take longer than FlexGroup volume scans because FlexGroup quota scans are performed in parallel across the nodes on which the FlexGroup volume resides.
- **Type of hardware and load on system.** Heavily loaded systems with many files can result in scans that take hours.

You can check quota initialization status by running the following the command:

```
quota show -volume volname -instance
```

Quota resize completion time

[Quota resize](#) is used when a quota policy is changed. The resize performs a scan with the new limits. This process also has some considerations for time to completion:

- The resize only scans using the newly added rules; therefore, it completes faster than an initialize.
- Resize typically completes in a matter of seconds because it has to do less than quotas on/off.
- Use resize instead of toggling quotas on/off because resize completes faster.
- Quota resize can run up to 100 concurrent jobs; after 100 jobs, the resize operations must wait in a queue.
- More concurrent scans can impact resize performance and add time to the job completion.

User-mapping considerations with quotas

User mapping in multiprotocol environments (data access from both SMB and NFS) for quotas occurs at the member volume level. Eventually, all member volumes agree on the user mapping. However,

sometimes there might be a discrepancy, such as when user mapping fails or times out when doing a name mapping that succeeded on another member. This means that at least one member considers the user to be part of a user-mapped pair, and at least one other member considers it to be a discrete record.

At worst, enforcement of the quota rules can be inconsistent until the issue is resolved. For instance, a user might be able to briefly overrun a quota limit.

An event management system message is sent when user mapping results are coordinated.

```
cluster::*> event route show -message-name fg.quota.usermapping.result -instance

                                Message Name: fg.quota.usermapping.result
                                Severity: NOTICE
                                Corrective Action: (NONE)
                                Description: This message occurs when the quota mapper
decides whether to map the Windows quota record and the UNIX quota record of a user into a single
multiuser record.
```

Tree quota considerations

SVMs in ONTAP can have a maximum of five quota policies, but only one policy can be active at a time. To see the active policy in an SVM, run the following commands:

```
cluster::> vserver show -vserver DEMO -fields quota-policy
vserver quota-policy
-----
DEMO      default
```

Note: Currently, you cannot view this information in ONTAP System Manager.

The default policy is adequate in most cases and does not need to be changed. When `quota on` is issued, the active policy is used—not the policy that was assigned to a volume. Therefore, it's possible to think you applied a quota and rules to a volume, but `quota on` fails.

The following example applies a quota policy to a volume:

```
cluster::*> quota policy show -vserver DEMO -policy-name tree

Vserver: DEMO
Policy Name: tree
Last Modified: 10/19/2017 11:25:20
Policy ID: 42949672962

cluster::*> quota policy rule show -vserver DEMO -policy-name tree -instance

Vserver: DEMO
Policy Name: tree
Volume Name: flexgroup_local
Type: tree
Target: tree1
Qtree Name: ""
User Mapping: -
Disk Limit: -
Files Limit: -
Threshold for Disk Limit: -
Soft Disk Limit: -
Soft Files Limit: -
```

Turning on quotas produces an error because the SVM has `default` assigned for quotas and does not contain any rules.

```
cluster::*> quota on -vserver DEMO -volume flexgroup_local -foreground true

Error: command failed: No valid quota rules found in quota policy default for volume
flexgroup_local in Vserver DEMO.
```

When you add a rule to default, the `quota on` command works, but the SVM does not use the new tree policy.

```
cluster::*> quota policy rule create -vserver DEMO -policy-name default -volume flexgroup_local -
type tree -target ""

cluster::*> quota on -vserver DEMO -volume flexgroup_local -foreground true
[Job 8063] Job succeeded: Successful

cluster::*> vserver show -vserver DEMO -fields quota-policy
vserver quota-policy
-----
DEMO      default
```

To use the necessary policy, you must modify the SVM and then turn quotas off and back on.

```
cluster::*> vserver modify -vserver DEMO -quota-policy tree

cluster::*> quota off -vserver DEMO *

cluster::*> quota policy rule delete -vserver DEMO -policy-name default *
1 entry was deleted.

cluster::*> quota on -vserver DEMO -volume flexgroup_local -foreground true
[Job 8084] Job succeeded: Successful
```

How clients see space when quotas are enabled

When quotas are enabled for a qtree in ONTAP, the clients only see the available space as reported by that quota.

For example, this is a quota for qtree1:

```
cluster::*> quota report -vserver DEMO -volume flexgroupDS -tree qtree1
Vserver: DEMO
```

Volume	Tree	Type	ID	-----Disk----- Used Limit	-----Files----- Used Limit	Quota Specifier
flexgroupDS	qtree1	tree	1	0B 500GB	1 -	qtree1

This is how much space that volume actually has:

```
cluster::*> vol show -vserver DEMO -volume flexgroupDS -fields size
vserver volume      size
-----
DEMO      flexgroupDS 10TB
```

This is what the client sees for space for that volume:

```
# df -h /mnt/nas2
Filesystem      Size  Used Avail Use% Mounted on
demo:/flexgroupDS 9.5T  4.5G  9.5T   1% /mnt/nas2
```

This is what is reported for that qtree:

```
# df -h /mnt/nas2/qtree1/
Filesystem      Size  Used Avail Use% Mounted on
demo:/flexgroupDS 500G    0  500G   0% /mnt/nas2
```

Advanced name mapping concepts

Name mapping in ONTAP can be as simple as having Windows and UNIX users with the same user names for implicit 1:1 name mapping to occur without the need for additional configurations. If ONTAP

can find the user names for Windows and UNIX users in a name service, and they match, everything will work successfully.

However, there are more complexities to name mappings, especially if the user names don't match or if you have multiple domains for ONTAP to search.

This section attempts to cover some of those complexities.

Regex and wildcards

Name mapping rules in ONTAP allow you to use [regular expressions \(regex\)](#) and wildcard values to configure name mapping rules for asymmetric user names. Wildcards can be useful when a multiple user names have the same general differences between UNIX and Windows names. For example, if all Windows user names have a period between the first and last names (such as `alice.smith`), but UNIX user names have an underscore (such as `alice_smith`), you can use regex to ensure those users always map to one another.

This is an example of Windows-to-UNIX regex name mapping that replaces periods with underscores:

```
vserver name-mapping create -vserver DEMO -direction win-unix -position 1 -pattern  
(.+)\\((.+)\\. (.+) -replacement \\2_\\3
```

This is an example of UNIX-to-Windows regex name mapping that replaces underscores with periods:

```
vserver name-mapping create -vserver DEMO -direction unix-win -position 2 -pattern (.+)_(.+) -  
replacement \\1\\.\\2
```

For more information, see [Name mapping conversion rules](#).

Mapping Windows admin users to root

In a multiprotocol NAS solution, you might want Windows admin users to have access to files and folders the same way root does in NFS/UNIX environments. One such use case is for data migrations, where an administrative user might need access to copy files and modify ACLs globally, without needing to be added to an ACL.

There are two main methods to accomplish this:

- For granular control over a user becoming root, you can create a win-unix name mapping rule that maps the desired user name to the root user. This method allows you to avoid applying a global rule to all administrative users present in the local BUILTIN\Administrators group in the SVM.
- If you want all users in the SVM's local BUILTIN\Administrators group to gain root access to files and folders in the SVM, use the CIFS server option `-is-admin-users-mapped-to-root-enabled`.

Note: If your use case is simply for data migrations, use the BUILTIN\Backup Operators instead. For information about which privileges are on each local group, see [List of supported privileges](#).

Mapping Windows clients to user names

In addition to mapping user names to user names, you can also map individual clients or subnets to user names by using name mapping rules. This is useful when you want to limit the access rights at a client or subnet level.

For example, if you want all clients in the 10.10.10.x/24 subnet (where you are running an application that requires specific Windows NTFS permissions) to map to the Windows user `DOMAIN\application`, use the following name mapping rule:

```
vserver name-mapping create -vserver SVM -direction unix-win -position 2 -pattern root -  
replacement DOMAIN\application -address 10.10.10.0/24
```


Using LDAP for name mapping

Although ONTAP provides a way to create explicit name mapping rules locally on the SVM, there is a limit of 1,024 rules allowed. In some cases, you might need more rules, or you simply prefer a centrally managed name mapping server, such as LDAP.

In these instances, use LDAP to act as your name mapping server by populating LDAP attributes with the UNIX or Windows user names you want to map and then specifying that attribute in the following LDAP client schema fields. Table 5 lists those attributes and what they do.

Table 5) LDAP client schema options – name mapping.

LDAP Schema Attribute	What It Does
<code>-windows-to-unix-object-class</code>	Provides the LDAP attribute to define the Windows-to-UNIX name mapping object class. Object classes are used to group multiple LDAP objects to enable faster searches. The default value in AD-IDMU is <code>User</code> . For RFC 2037 schemas, the value is set to <code>posixAccount</code> .
<code>-windows-to-unix-attribute</code>	Provides the LDAP attribute for the value that is used for mapping a Windows user to a UNIX user. The default value for AD-IDMU schemas in ONTAP is <code>sAMAccountName</code> . For RFC 2307 schemas, the value defaults to <code>windowsAccount</code> .
<code>-windows-to-unix-no-domain-prefix</code>	This option controls whether the attribute value in <code>-windows-to-unix-attribute</code> has the domain prefix added to it. (The default is <code>false</code> .) Because <code>sAMAccountName</code> is represented by a single user name (rather than <code>DOMAIN\username</code>) and because <code>msDS-PrincipalName</code> is not a value that can be used in LDAP search, domain prefixes might be necessary to enable functional asymmetric name mapping. The need for this value depends on the LDAP schema and attributes that are being used, as well as whether multiple domain name mappings are present for multiple unique Windows domains.
<code>-windows-account-attribute</code>	This option controls which LDAP schema attribute to use when mapping UNIX names to Windows names. The default value of this attribute is <code>sAMAccountName</code> , which is the standard field that is used for Windows accounts when new users are created.

After the LDAP client is configured for name mapping lookups, modify the ns-switch database `namemap` to use `ldap`, `files`. If your Windows and UNIX user names match or are symmetric (for example, `johns` in Windows is `johns` in UNIX), then no action is required. For more information about using LDAP for asymmetric name mapping, see [TR-4835: How to Configure LDAP in ONTAP](#).

Note: Specify an external service in the name map database only if an external service is actually being used for asymmetric name mappings. If you specify a server that does not have any name mapping rules configured, name mapping lookups can add latency to requests and creates slow authentication or failures.

NAS redirects and global sharing

Sharing files across a local network is generally easy; networks are reliable and locking semantics don't have many complexities to contend with. However, to share NAS datasets across a WAN to multiple sites, or even across multiple file systems in the same network, things can get a bit complicated. This

section covers a few scenarios that apply, such as symbolic link (symlinks)/widelinks, using Distributed File System (DFS) and FlexCache volumes.

Symlinks and widelinks

ONTAP supports the use of both symlinks and widelinks to redirect traffic from folders or files in a NAS share to other locations, even remote, non-NetApp storage. This feature allows storage administrators to create and present single namespaces that are transparent to clients regardless of where that data lives.

What is a symlink?

A symlink is a file that contains a reference to another file or directory in the form of an absolute or relative path.

- **Absolute paths** point to the same location in a file system, regardless of the current working directory or where a symlink resides. This path should always start with a “/” in the path.
- **Relative paths** start at a given working directory. While this is a shorter and simpler way to define links, it can also create Path Not Found errors if the wrong path is defined.

In ONTAP, you can create symlinks from UNIX clients over NFS or through PowerShell by referencing this knowledge base article: [How to create a symbolic link in ONTAP without the need for NFS.](#)

Note: You currently cannot create symlinks from CIFS/SMB clients. For more information, see [bug 930915](#).

What is a widelink?

A widelink is a symlink that allows you to extend your NAS namespace outside of the storage system to other NAS devices. This can include other ONTAP instances or even non-NetApp storage – including Windows DFS. In ONTAP, you can create symlinks and widelinks by specifying the appropriate CIFS share `-symlink-properties` options:

```
cluster::*> cifs share modify -vserver DEMO -share-name share -symlink-properties ?
enable          (DEPRECATED)-Enable both local symlinks and wide links for read-write
hide            (DEPRECATED)-Hide both symlinks and wide links
read_only       (DEPRECATED)-Enable symlinks for read-only
symlinks        Enable symlinks only for read-write, DFS is not advertised
symlinks_and_widelinks Enable both local symlinks and wide links, DFS is advertised
disable         Disable both local symlinks and wide links, DFS is not advertised
no_strict_security Allow clients to follow symlinks outside share boundaries
```

What is a hardlink?

A hardlink is a link that can be used to link to a file rather than a directory. Unlike symlinks and widelinks, a hardlink cannot span different file systems and cannot link directories. In ONTAP, that means a hardlink cannot span:

- Different volumes
- Different qtrees
- Between Snapshot copies and an active file system
- Different SVMs
- Different storage systems

Folders/directories in the same volume are not considered different file systems; however, a hardlink can point to a file that is multiple levels deep in the same volume.

If you attempt to create a hardlink that crosses file system boundaries, the following error is displayed:

```
ln: failed to create hard link 'hard-link' => '/path/to/file: Invalid cross-device link
```

When a hardlink is created, the file has multiple inode access points. For example, the same file contents on both the file and hardlink are shown:

```
# cat /mnt/client1/dir1/dir2/linked-dir/hard-file
this is a linked file

# cat /mnt/client1/hard-link
this is a linked file
```

To find if a file that has a hard link, first use `ls -li` to find the inode number and then find all the files with the same inode number.

For example:

```
# ls -li | grep hard-link
1146684405 hard-link

# find /mnt/client1 -inum 1146684405
/mnt/client1/dir1/dir2/linked-dir/hard-file
/mnt/client1/hard-link
```

UNIX/NFS symlinks

When creating UNIX/NFS symlinks, there are no special configurations or considerations to make. If the path being linked to exists on the client, these NFS symlinks will work as expected.

CIFS symlink paths

For UNIX-created symlinks with CIFS/SMB shares, map the symlink path to the proper CIFS share path in order for the link to work properly. When creating CIFS symlinks, there are two main points to consider:

- The storage system can overlay a UNIX-style symbolic link with a DFS referral; therefore, CIFS clients are also redirected. If the links are relative links, and stay within the share, the storage system knows how to map these transparently.
- If a symbolic link is absolute, or points to a different export, you can create a mapping rule to make sure the links resolve CIFS clients to the appropriate destination (whether it's a different CIFS share on the same node, or a share on a different CIFS server).

CIFS symlink paths are necessary to help ONTAP redirect the link to the proper file or directory. These paths are created by the running the command `cifs symlink create`.

Note: The `cifs symlink create` command does not create symlinks; instead, it creates path mappings. Symlinks still need to be created on NFS clients or by using PowerShell.

Creating CIFS symlink mappings

The [How to make symbolic links \(widelinks\) work for CIFS clients on clustered Data ONTAP](#) Knowledge Base article (in the section titled “Seven Important Things to Consider”) describes in simple terms what you need to do in order to make CIFS symlinks work properly:

- The command `cifs symlink create` does not create a symlink.
- A symlink is required and can only be created from an NFS client or through the PowerShell toolkit.
- The CIFS symlink map entry must exist on the Vserver that has the share that contains the symbolic link, not the destination.
- The share on the Vserver with the symbolic link you are mapping must have symbolic links enabled or set to read-only. Verify this setting by running the `cifs share show` command.
- The Vserver parses the contents (destination path) of the symbolic link to use for mapping, not the name of the link, nor the path of the link itself. To see what a link points to, that is, what needs to be mapped, perform an `ls -l` on the directory with the link, and check the destination path.

- Symbolic link mapping to CIFS referrals only works for directories, not for files.
- If the only purpose of the symbolic links is to redirect CIFS clients. It is acceptable if the UNIX path of the symbolic link destination does not actually exist inside ONTAP or on an NFS client—the symlink can exist purely for the purpose of the CIFS map. If the relevant mapping is correct and matches the destination path of the symlink, redirects work for CIFS even if the link itself does not work for UNIX or Linux clients.

Examples of CIFS symlinks

The following sections provide examples of CIFS symlinks:

- [CIFS symlink in the same volume using a relative path.](#)
- [CIFS symlink in the same volume using an absolute path.](#)
- [CIFS widelink in the same SVM/different volume.](#)
- [CIFS widelink to a non-NetApp CIFS share.](#)
- [CIFS symlink to a local file.](#)
- [CIFS symlink to a remote file.](#)

While attempting to create symlinks, the CIFS/SMB client and ONTAP sometimes disagree on the state of the symlink path resolutions. If you experience unexpected behavior or things don't appear work properly, see the "Caches and symlink errors" section for workarounds.

For more information about DFS behavior and symlinks with MacOS, see: [DFS links do not work on MAC OS client, with ONTAP 9.5 and symlinks enabled.](#)

For other examples, see [How to Create a symlink and widelink in Clustered Data ONTAP.](#)

CIFS symlink: Same volume, relative path

This example shows how to create a link at the root of the volume that uses the relative path of the volume to redirect to a folder that is three levels deep in the same volume.

- The relative folder path is `dir1/dir2/linked-dir`.
- The link is created by running the following command in the NFS mount:

```
ln -s dir1/dir2/linked-dir rel-link
```

When that link is created, the expected results from NFS are displayed. Both the symlink (`rel-link`) and the folder path (`dir1/dir2/linked-dir/`) show the same file.

```
# ls -la rel-link/
total 8
drwxr-xr-x 2 root root 4096 Feb 15 16:41 .
drwxr-xr-x 3 root root 4096 Feb 15 16:40 ..
-rw-r--r-- 1 root root 0 Feb 15 16:41 rel-link-file

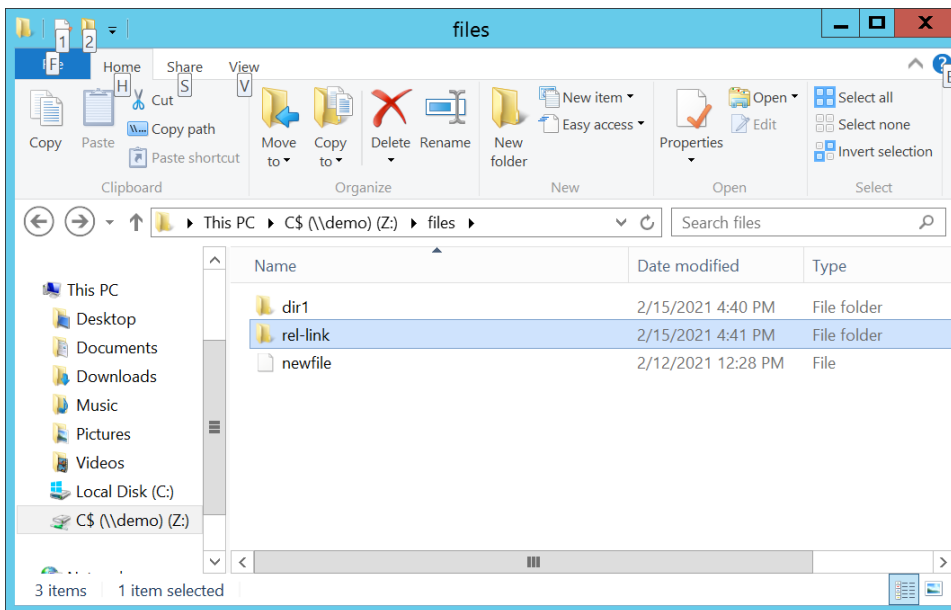
# ls -la dir1/dir2/linked-dir/
total 8
drwxr-xr-x 2 root root 4096 Feb 15 16:41 .
drwxr-xr-x 3 root root 4096 Feb 15 16:40 ..
-rw-r--r-- 1 root root 0 Feb 15 16:41 rel-link-file
```

In ONTAP, a symlink with a relative path in the same volume redirects without needing to create a special CIFS symlink mapping, provided the symlink property is set on the CIFS share. By default, all CIFS shares already have this property set; therefore, relative CIFS symlinks should work out of the box without any special configuration required.

In the CIFS/SMB share, the link appears as a directory or a shortcut.

Note: How the symlink appears (as a shortcut or a file/directory) depends on the SMB version in use and is controlled through the options described in “Junction paths and reparse points.”

Figure 16) CIFS Symlink, relative path - same volume.



CIFS symlink: Same volume, absolute path

When you create a symlink with an absolute path, then you are telling the link that the path being used is always the path to be used, regardless of where in the namespace your link resides.

When this type of link is created, the default behavior in ONTAP is different.

The following example creates a link at the root of the volume that uses the absolute path of the NFS mount to redirect to a folder that is three levels deep in the same volume:

- The absolute folder path is `/mnt/client1/dir1/dir2/linked-dir`.
- The link is created by running the following command in the NFS mount:

```
ln -s /mnt/client1/dir1/dir2/linked-dir abs-link
```

When that link is created, the expected results from NFS are displayed. Both the absolute path symlink (`abs-link`) and the folder path (`dir1/dir2/linked-dir/`) show the same file.

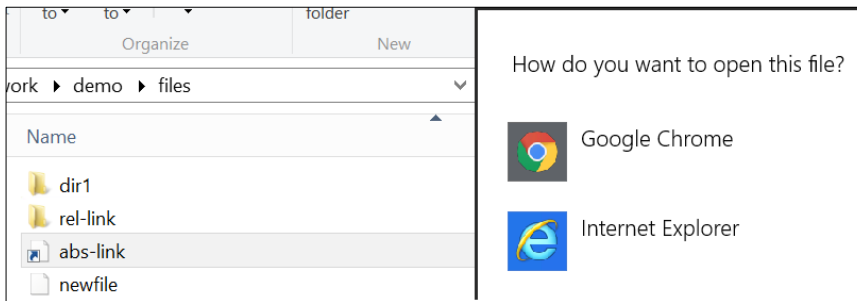
```
# touch abs-link/abs-link-file
# ls -la abs-link/
total 8
drwxr-xr-x 2 root root 4096 Feb 15 17:04 .
drwxr-xr-x 3 root root 4096 Feb 15 16:40 ..
-rw-r--r-- 1 root root 0 Feb 15 17:04 abs-link-file
-rw-r--r-- 1 root root 0 Feb 15 16:41 rel-link-file

# ls -la dir1/dir2/linked-dir/
total 8
drwxr-xr-x 2 root root 4096 Feb 15 17:04 .
drwxr-xr-x 3 root root 4096 Feb 15 16:40 ..
-rw-r--r-- 1 root root 0 Feb 15 17:04 abs-link-file
-rw-r--r-- 1 root root 0 Feb 15 16:41 rel-link-file
```

However, from the CIFS/SMB share, there is a shortcut file that doesn't redirect anywhere. Instead, there is a prompt asking how to open that file.

Note: How the symlink appears (as a shortcut or a file/directory) depends on the SMB version in use. The symlink is controlled through the options described in “Junction paths and repare points.”

Figure 17) CIFS symlink, absolute path, same volume – default behavior.

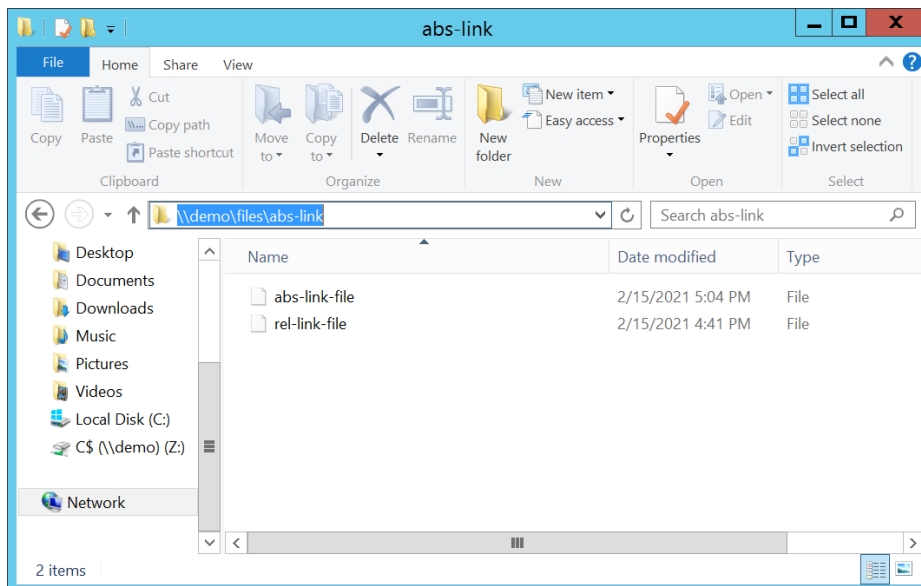


To let ONTAP know that the link in this directory is a symlink, define a CIFS symlink path by running the following command:

```
cluster::> cifs symlink create -vserver DEMO -unix-path /mnt/client1/ -cifs-path / -cifs-server DEMO -locality local -share-name files
```

In this example, the `-unix-path` is defined with the mount path of the NFS client we created the symlink with (`/mnt/client1`). Because the file is in the same volume, use “local” as the `-locality` value. Even if that NFS client is unmounted, ONTAP still knows how to redirect the link within the CIFS share.

Figure 18) CIFS Symlink, absolute path – same volume.



CIFS symlink: Different volume/share (widelink)

You now need to link from one volume to another in the same namespace. This is a significant step because each volume is considered a unique file system to NAS clients. You must treat the way you link to these volumes a little differently.

In this example, create a link at the root of a volume that uses the absolute path of another NFS mount to redirect to a folder that is three levels deep in a different volume. Do not use a relative path here unless you use a path higher in the directory tree. Use the same directory you’ve been using, but the link will reside in a different volume.

- The absolute folder path for the linked volume is `/mnt/client1/dir1/dir2/linked-dir`.
- Create the link to the `client1` mount by running the following command in the `client2` NFS mount:

```
ln -s /mnt/client1/dir1/dir2/linked-dir remote-link
```

These are the two mounted volumes:

```
# mount | grep client
DEMO:/files on /mnt/client1 type nfs
DEMO:/flexgroup_16 on /mnt/client2 type nfs
```

When that link is created, the expected results from NFS are displayed. Both the remote symlink (`remote-link`) and the folder path (`/mnt/client1/dir1/dir2/linked-dir/`) show the same file that was created from the symlink path.

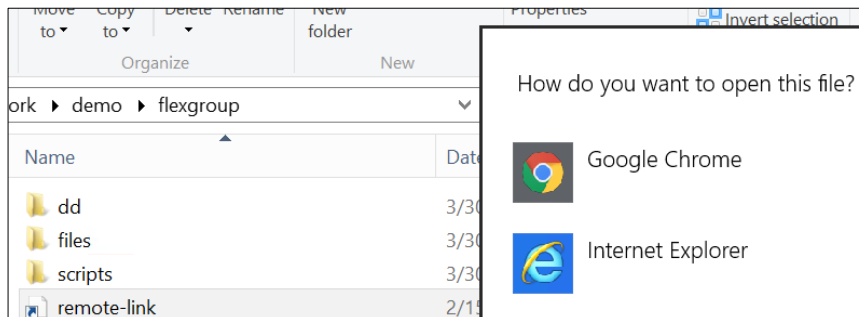
```
# touch /mnt/client2/remote-link/remote-file
# ls -la remote-link/
total 8
drwxr-xr-x 2 root root 4096 Feb 15 17:32 .
drwxr-xr-x 3 root root 4096 Feb 15 16:40 ..
-rw-r--r-- 1 root root 0 Feb 15 17:04 abs-link-file
-rw-r--r-- 1 root root 0 Feb 15 16:41 rel-link-file
-rw-r--r-- 1 root root 0 Feb 15 17:32 remote-file

# ls -la /mnt/client1/dir1/dir2/linked-dir/
total 8
drwxr-xr-x 2 root root 4096 Feb 15 17:32 .
drwxr-xr-x 3 root root 4096 Feb 15 16:40 ..
-rw-r--r-- 1 root root 0 Feb 15 17:04 abs-link-file
-rw-r--r-- 1 root root 0 Feb 15 16:41 rel-link-file
-rw-r--r-- 1 root root 0 Feb 15 17:32 remote-file
```

By default, the CIFS share on the SMB client shows a shortcut file that doesn't redirect anywhere.

Note: How the symlink is displayed (as a shortcut or a file/directory) depends on the SMB version in use. The symlink is controlled through the options described in "Junction paths and reparse points."

Figure 19) CIFS symlink, absolute path, different volume – default behavior.



Again, ONTAP must be informed of where that redirect should go. In the previous examples, a symlink path was created with `-unix-path` defined as `/mnt/client1`. This symlink points to `/mnt/client1`, but exists in `/mnt/client2`, so we'd need a new CIFS symlink entry to inform ONTAP where the path is supposed to redirect.

Because we're spanning file systems, this type of link is considered a widelink. To change the `-locality` option, run the following command :

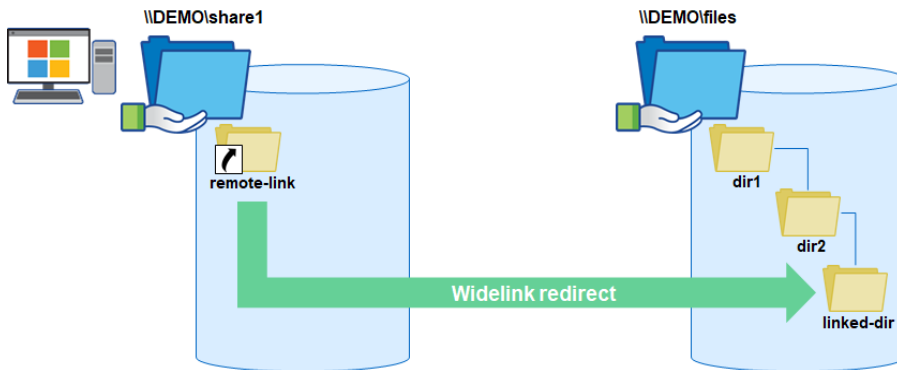
```
cluster::*> cifs symlink modify -vserver DEMO -unix-path /mnt/client1/ -locality widelink
```

In addition, enable both the source and destination CIFS shares should have `symlinks_and_widelinks` for `-symlink-properties`. If you define the symlink path as a widelink and don't change the `-symlink-properties` to `widelink`, then existing links will break (such as the absolute path link we created previously).

```
cluster::*> cifs share modify -vserver DEMO -share-name source -symlink-properties  
symlinks_and_widelinks  
  
cluster::*> cifs share modify -vserver DEMO -share-name destination -symlink-properties  
symlinks_and_widelinks
```

This CIFS symlink mapping is how ONTAP redirects.

Figure 20) CIFS widelink redirect – same SVM



After the mapping is complete, the link shows as a shortcut folder or regular folder (rather than a shortcut file) that correctly redirects to the proper linked location and we can see the files we're supposed to see.

Note: How the symlink appears (as a shortcut or a file/directory) depends on the SMB version in use. The symlink is controlled through the options described in “Junction paths and reparse points.”

Figure 21) CIFS symlink – before and after proper configuration.

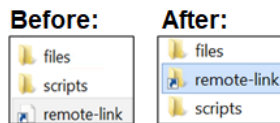
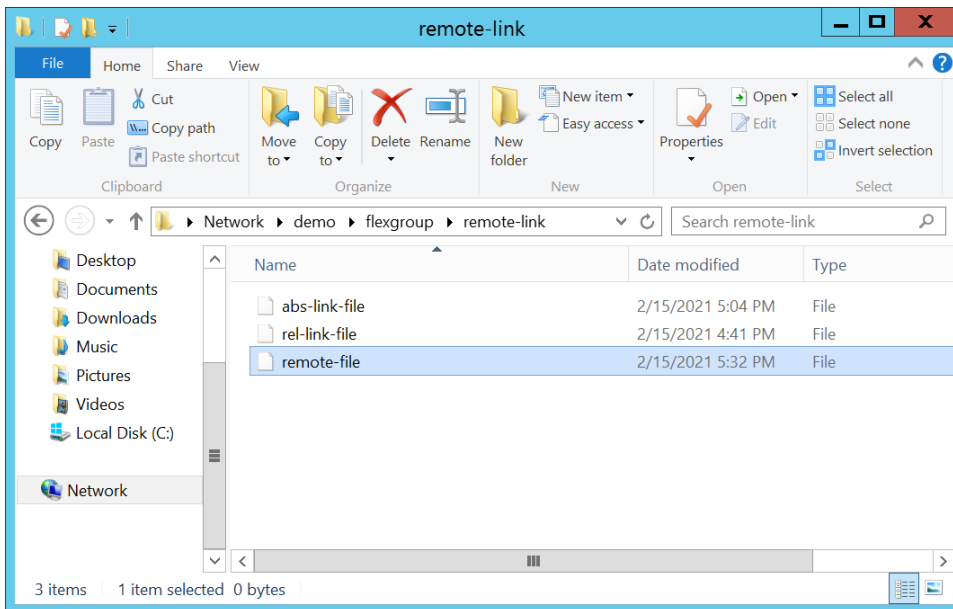


Figure 22) CIFS symlink, different volume/same SVM – widelink.



CIFS symlink: Non-NetApp CIFS share (widelink)

ONTAP can also set up CIFS symlinks that redirect to CIFS/SMB servers that are not hosted by ONTAP storage systems, which means ONTAP can act as a DFS namespace similarly to Windows servers.

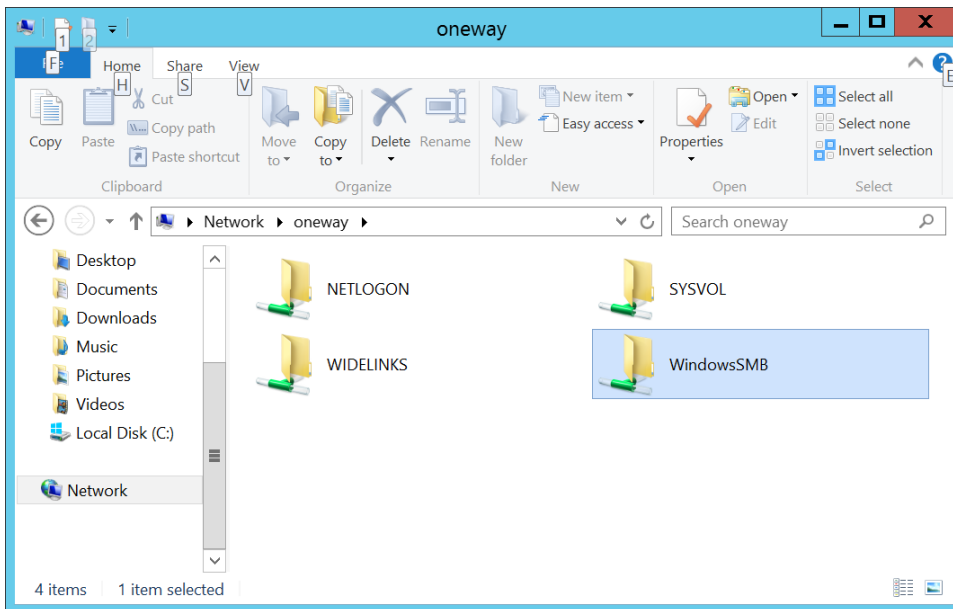
This example uses a CIFS symlink hosted in an ONTAP volume that points to a CIFS share hosted on a Windows server. Because this is a Windows server, the symlink from the NFS client can't be tested, but you can leverage the same concepts described in the previous sections to create a working widelink across SMB servers.

- The `-unix-path` for the linked Windows share is `/mnt/winclient/WindowsSMB/`.
- Create the link to the `client1` mount by running the following command in the `client2` NFS mount:

```
ln -s /mnt/winclient/WindowsSMB/WindowsSMB-link win-widelink
```

The Windows share that was created on the Windows server is named `ONEWAY.NTAP.LOCAL`, which means the CIFS server name is `ONEWAY`.

Figure 23) Windows server SMB share.



To create a CIFS symlink path for that Windows server, run the following command. Remember that the CIFS/SMB share on the ONTAP SVM where the symlink resides must have the `-symlink-property` value `symlinks_and_widelinks` set and that `-locality` must be set to `widelink`.

```
cluster::*> cifs symlink create -vserver DEMO -unix-path /mnt/winclient/WindowsSMB/ -cifs-path /  
-cifs-server ONEWAY -locality widelink -share-name WindowsSMB  
  
cluster::*> cifs share show -vserver DEMO -symlink-properties symlinks_and_widelinks -fields  
symlink-properties  
vserver share-name symlink-properties  
-----  
DEMO files symlinks_and_widelinks  
DEMO flexgroup symlinks_and_widelinks
```

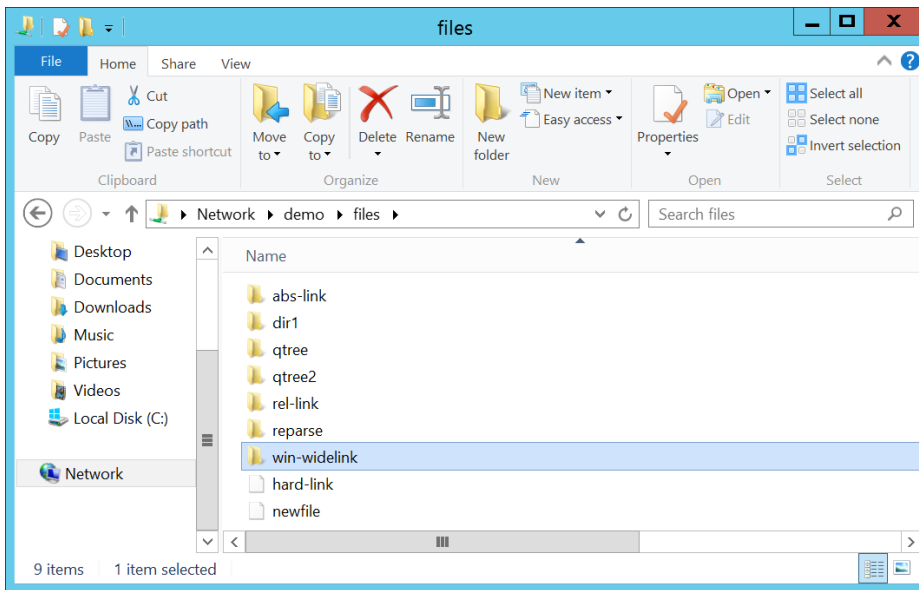
The `cifs symlink` command includes the following configuration breakdown:

- The `-unix-path` path is used to create the symlink (`/mnt/winclient/WindowsSMB/`).
- `-cifs-path` is set to `/`, because that is where you will start your navigation.
- `-cifs-server` is the name of the destination CIFS/SMB server. In this example, the Windows server name is `ONEWAY`.
- `-locality` is `widelink` because you're crossing file systems.
- `-share-name` is the name of the destination CIFS/SMB share `WindowsSMB`.

After you create the CIFS symlink path, navigate to the newly created symlink. From there, either a shortcut icon or a folder icon is displayed.

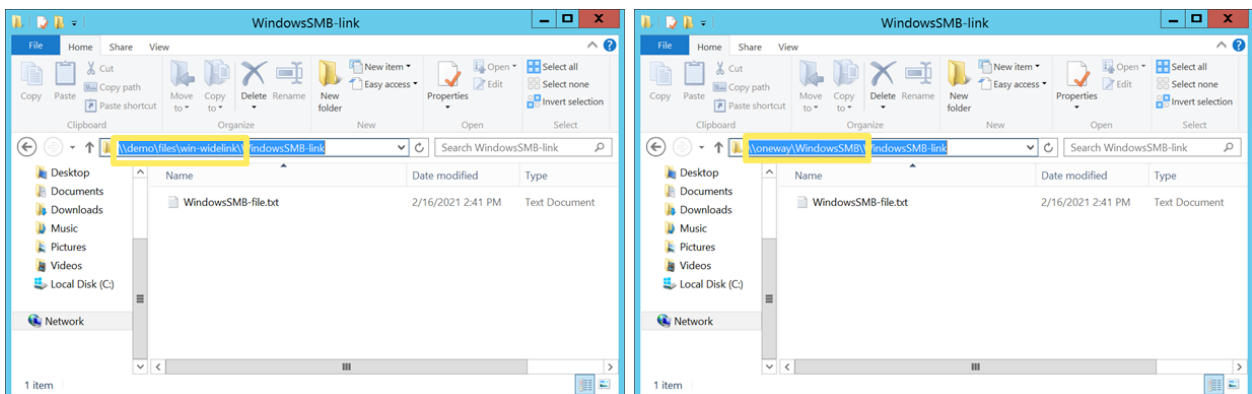
Note: How the symlink is displayed (as a shortcut or a file/directory) depends on the SMB version in use. The symlink is controlled through the options described in “Junction paths and reparse points.”

Figure 24) CIFS symlink – widelink to the Windows server.



After you navigate to the widelink folder, you will see the same contents as what you'd see if you navigated to the Windows SMB share directly.

Figure 25) CIFS symlink versus direct connection to a Windows SMB share.



From the Windows client, the `dfsutil diag` command displays the following path resolution:

```
C:\>dfsutil diag viewdfspath \\demo\files\win-widelink
```

The DFS Path <\\demo\files\win-widelink> resolves to -> \\ONEWAY\WindowsSMB

Note: This procedure works for any CIFS/SMB server that supports DFS referrals.

The next section provides more information about using `dfsutil`.

CIFS symlink: Link to a local file

It is also possible to create symlinks that point to files that CIFS/SMB clients can see as files. To create these symlinks, the following requirements must be met:

- UNIX/NFS created symlink to a file.
- CIFS symlink path in ONTAP using `-locality local`.

- The CIFS share with `-symlink-properties` is set to `symlinks` or `symlinks_and_widelinks`.
- Depending on the configuration, the `-symlink-properties` option `no_strict_security` is optional.

The following example is the file symlink that was created through NFS. This symlink lives in the same volume as the file to which it links.

```
# ls -la | grep file-symlink
lrwxrwxrwx 1 root root          45 Feb 18 10:12 file-symlink.txt ->
/mnt/client1/dir1/dir2/linked-dir/linked-file
# pwd
/mnt/client1

# cat file-symlink.txt
This is a file symlink.

# cat /mnt/client1/dir1/dir2/linked-dir/linked-file
This is a file symlink.
```

The following example is the CIFS symlink path in ONTAP:

```
cluster::*> cifs symlink show -vserver DEMO -unix-path /mnt/client1/

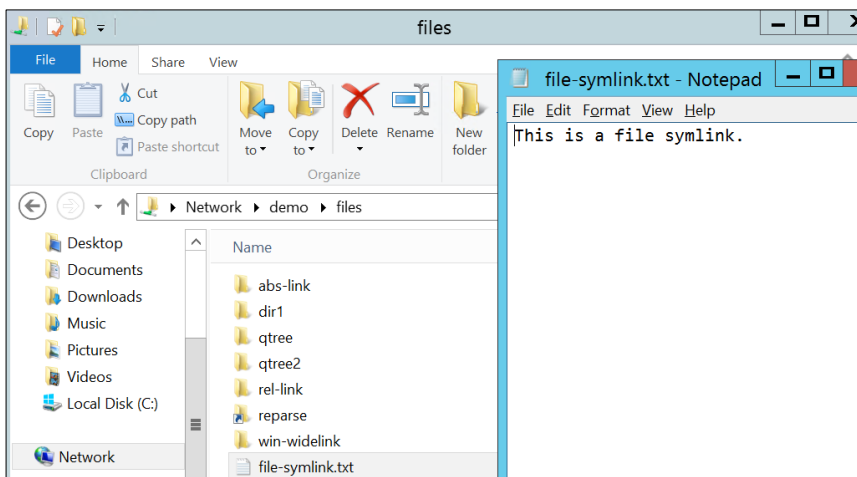
      Vserver: DEMO
      UNIX Path: /mnt/client1/
      CIFS Share: files
      CIFS Path: /
Remote NetBIOS Server Name: DEMO
  Local or Wide Symlink: local
      Home Directory: false
```

And the CIFS share `-symlink-properties` value:

```
cluster::*> cifs share show -vserver DEMO -share-name files,flexgroup -fields symlink-properties
vserver share-name symlink-properties
-----
DEMO     files          symlinks_and_widelinks
```

When the above procedure is used, the symlink file appears on the SMB client.

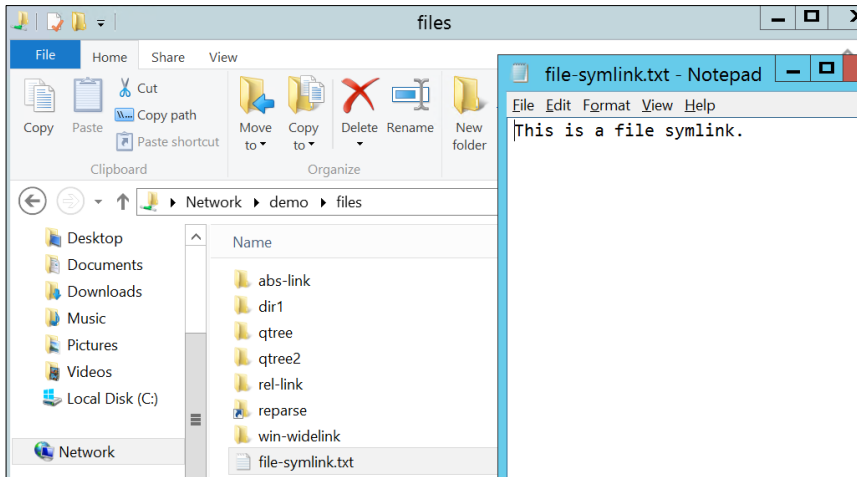
Figure 26) Local file symlink – local locality, `symlinks_and_widelinks` share property.



If you change the CIFS share `-symlink-properties` value to `no_strict_security`, the local symlink still works.

```
cluster::*> cifs share modify -vserver DEMO -share-name files -symlink-properties  
symlinks,no_strict_security
```

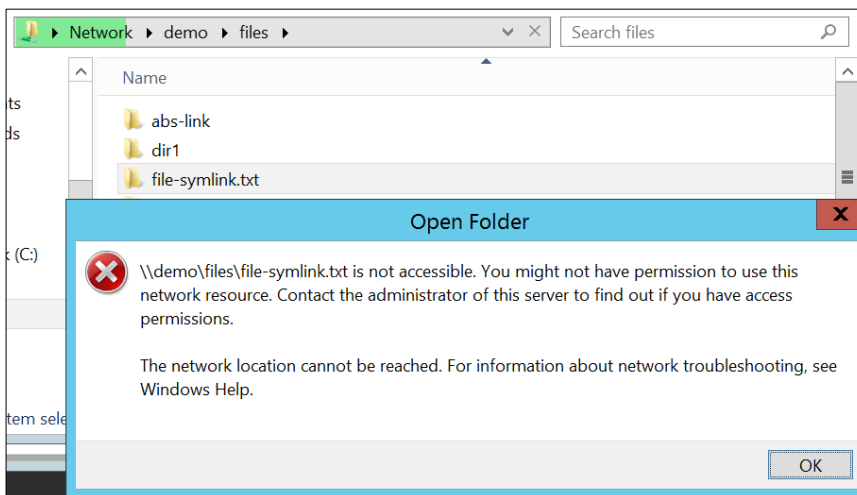
Figure 27) Local file symlink – local locality, symlinks,no_strict_security share property.



If you change the symlink path to `-locality widelink`, the symlink displays in Windows as a folder, and the file won't open properly.

```
cluster::*> cifs symlink modify -vserver DEMO -unix-path /mnt/client1/ -locality widelink
```

Figure 28) Local file symlink – widelink locality, symlinks_and_widelinks share property.



Widelink entries have the following limitations:

- Even if the destination of the widelink is a file, it appears as a directory in directory listings.
- The system API for opening the file correctly follows the widelink, but this process might confuse certain applications. To avoid this problem, create a widelink that resolves to a directory, rather than a file.
- Widelinks cannot direct a client to a nonshared area on the destination machine.

CIFS symlink: Link to a remote file

Creating a file symlink that links to a file in the same volume is straightforward, as listed in “CIFS symlink: Link to a local file.”

However, when you want to link to a file that leaves the confines of the volume, you are faced with some challenges:

- Symlinks that leave a volume boundary are generally considered widelinks.
- Widelinks are displayed as directories in SMB clients.

So, how do you create a file symlink that appears as a file and redirects to another share?

There are two main options:

- Create a CIFS symlink mapping at the root of a folder with junctioned volumes, with a symlink that uses ../ to redirect up the file path to the top level of the directory. Users never leave the share.
- Create a CIFS symlink mapping to the destination share that uses an absolute path to the desired file. Users travers shares through symlink.

Figure 29 shows each of these options.

Figure 29) Symlink from root of share with junctioned volumes and ../ symlink path.

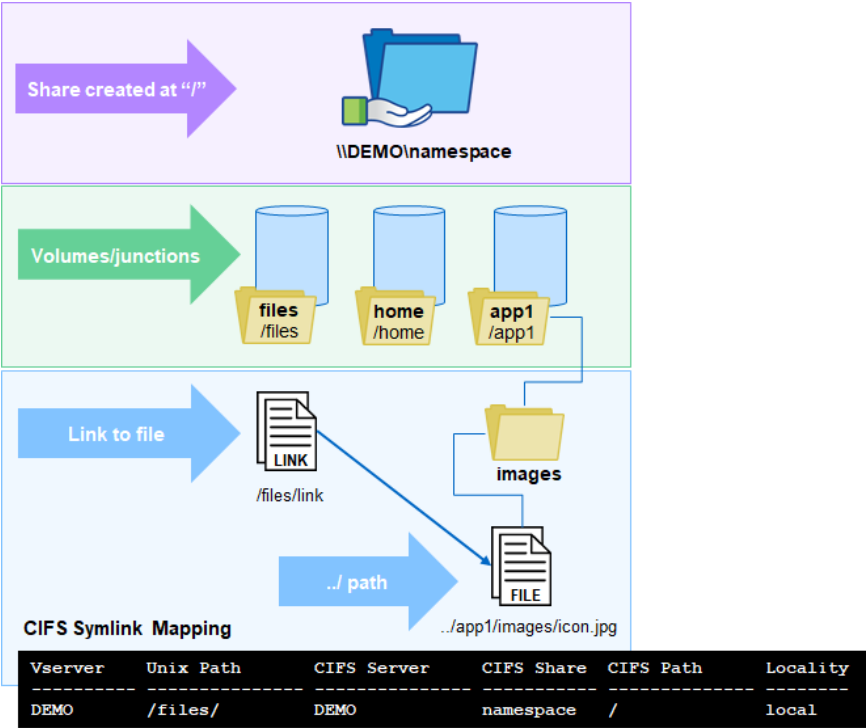
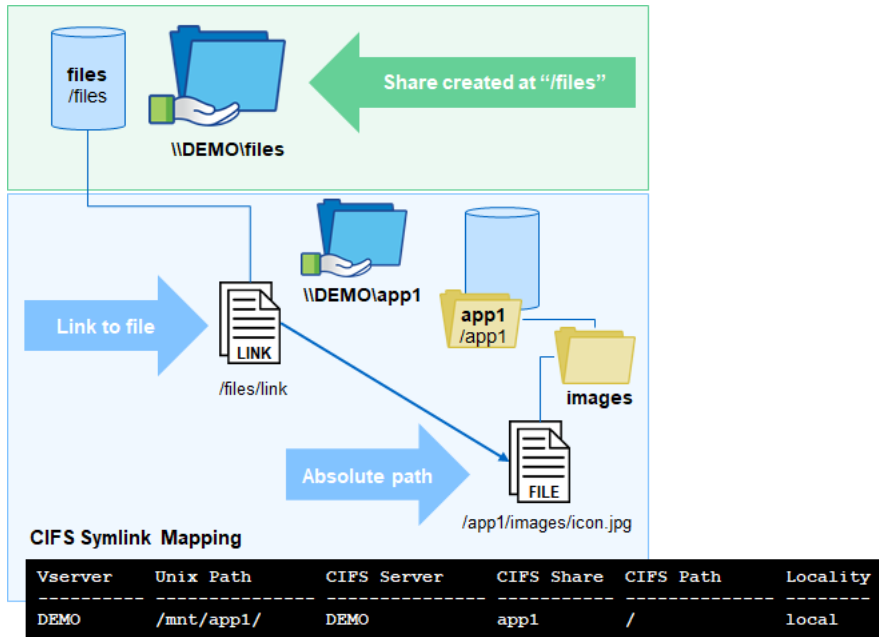


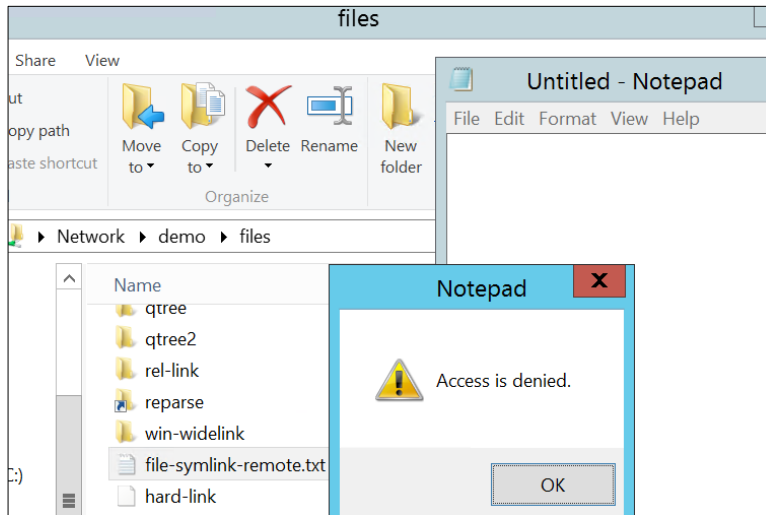
Figure 30) Symlink from share with redirect to another share – absolute path.



Potential issues with file symlinks

In some instances, you might experience issues after creating a CIFS symlink mapping for a file. This section addresses the issues you might experience and their potential causes.

Figure 31) Remote file symlink – local locality, `symlinks_and_widelinks` share property.



Access Denied message

If Access Denied is displayed when you try to open a symlink to a file from SMB clients, verify the following:

- ACLs on the symlink
- ACLs on the destination file
- ACLs on the destination share

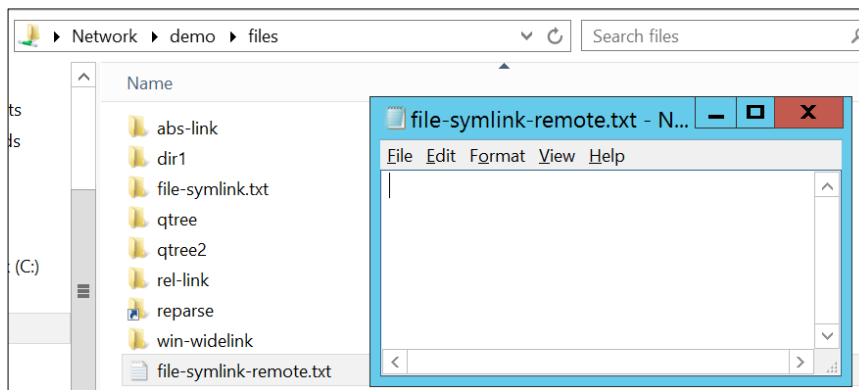
- Does the symlink work from the NFS client?
- CIFS symlink mapping path configuration:
 - Is the share name correct?
 - Is the UNIX path correct?
 - Are you using a relative or absolute path?
 - Are you using local as the locality?
- CIFS share -symlink-properties
 - Is [no_strict_security](#) used?

Note: `no_strict_security` allows symlinks to work across shares by removing some of the SMB enforcement.

Blank file

In some instances, you might be able to open a symlink from a CIFS client, but the file doesn't have the expected content in it.

Figure 32) Remote file symlink – Blank file, `no_strict_security`.



If this issue occurs, verify the following.

- Does the symlink work from the NFS client?
- CIFS symlink mapping path configuration:
 - Is the share name correct?
 - Does a path mapping for the UNIX path in the symlink exist?
 - Is the UNIX path correct?
 - Are you using a relative or absolute path?
 - Are you using local as the locality?
- CIFS share -symlink-properties:
 - Is `no_strict_security` used?

Note: `no_strict_security` allows symlinks to work across shares by removing some of the SMB enforcement. If the path is not mapped properly, a blank file is displayed.

`No_strict_security`, symlinks and DFS advertisements

When creating a CIFS symlink path, you can control whether ONTAP [advertises using DFS](#) when a symlink is configured to leave the boundaries of a share or if the symlink simply follows the path defined without advertising DFS.

The `-symlink-path` option on the CIFS share called `no_strict_security` controls this behavior.

When this option is disabled (not set), and the CIFS symlink path has `-locality` set to `widelink`, the CIFS/SMB client sends `FSCTL_DFS_GET_REFERRALS` to the storage system to see if the storage system is advertising the path through DFS.

In a packet capture, the following packets are displayed when DFS advertisements are used:

Packet from SMB client:

```
1247    13.520643    x.x.x.x  x.x.x.y    SMB2    230    Ioctl Request
FSCTL_DFS_GET_REFERRALS, File: \demo\files\win-widelink
File Name: \demo\files\win-widelink
```

Response packet from ONTAP:

```
1248    13.521286    x.x.x.y  x.x.x.x    SMB2    382    Ioctl Response
FSCTL_DFS_GET_REFERRALS
Path: \demo\files\win-widelink
Alt Path: \demo\files\win-widelink
Node: \ONEWAY\WindowsSMB\WindowsSMB-link
```

Some applications require DFS advertisements to be disabled, but they still must be able to traverse symlinks. In this instance, disable DFS advertisements for the share.

To create a CIFS symlink path mapping that does not advertise through DFS, complete the following steps:

1. Create the symlink as normal on the NFS client.
2. Create the symlink path to use `-locality local` and `-share [destination share name]` values.
3. Use the path used for the symlink in `-unix-path`.
4. Set the `-symlink-properties` to `symlinks, no_strict_security` on the source share.

In the following examples, the CIFS share named is `flexgroup` with a symlink in the root of the share that redirects to the CIFS share `files`.

The following example shows the symlink; it points to `/mnt/xyz`:

```
# ln -s /mnt/xyz nostrict-link
# cd nostrict-link/
# ls -la
total 8
drwxr-xr-x 2 root root 4096 Feb 18 13:15 .
drwxr-xr-x 6 root root 4096 Feb 18 13:15 ..
lrwxrwxrwx 1 root root    8 Feb 18 13:15 xyz -> /mnt/xyz
```

The following example shows the CIFS symlink path mapping:

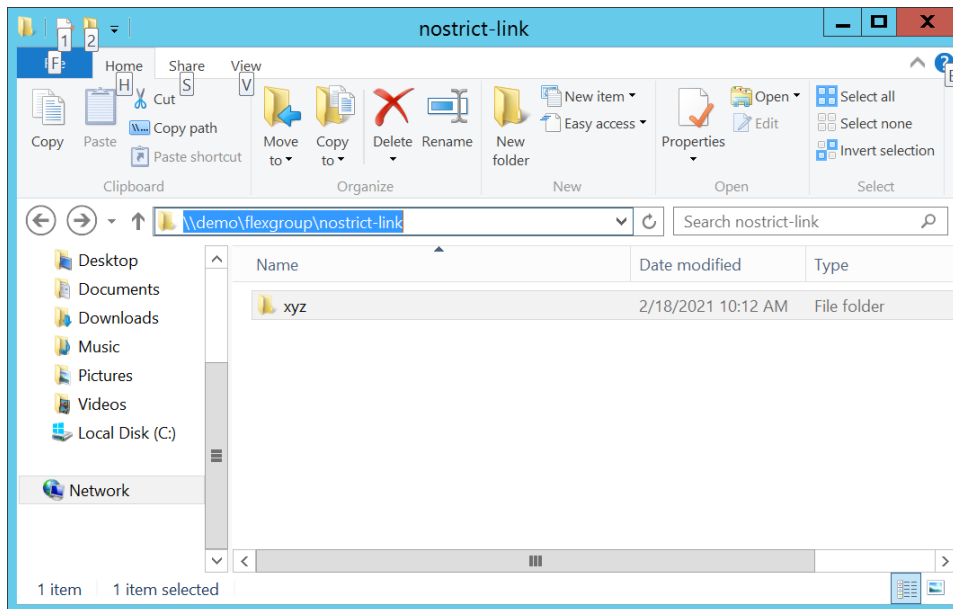
```
cluster::*> cifs symlink create -vserver DEMO -unix-path /mnt/xyz/ -cifs-path / -cifs-server DEMO
-locality local -home-directory false -share-name files
```

The following example shows the CIFS share `-symlink-properties`:

```
DEMO    flexgroup    symlinks,no_strict_security
```

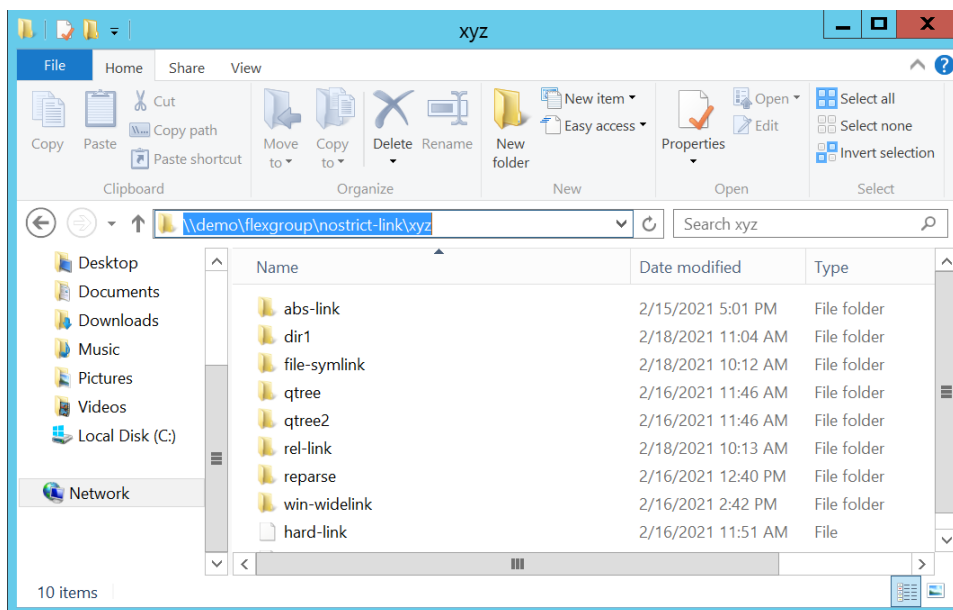
This example shows how the symlink is displayed in the CIFS:

Figure 33) CIFS symlink – no_strict_security.



When you navigate to this folder, it takes us to the share files.

Figure 34) CIFS symlink – no_strict_security navigation.



This is how the client sees the paths:

```
C:\>dfsutil diag viewdfspath \\demo\\flexgroup
<\\demo\\flexgroup> is not a DFS Path
Could not complete the command successfully.
SYSTEM ERROR - The system cannot find the file specified.

C:\>dfsutil diag viewdfspath \\demo\\flexgroup\\nostrict-link
<\\demo\\flexgroup\\nostrict-link> is not a DFS Path
```

```
Could not complete the command successfully.  
SYSTEM ERROR - The system cannot find the file specified.
```

```
C:\>dfsutil diag viewdfspath \\demo\flexgroup\nostrick-link\xyz
```

```
<\\demo\flexgroup\nostrick-link\xyz> is not a DFS Path
```

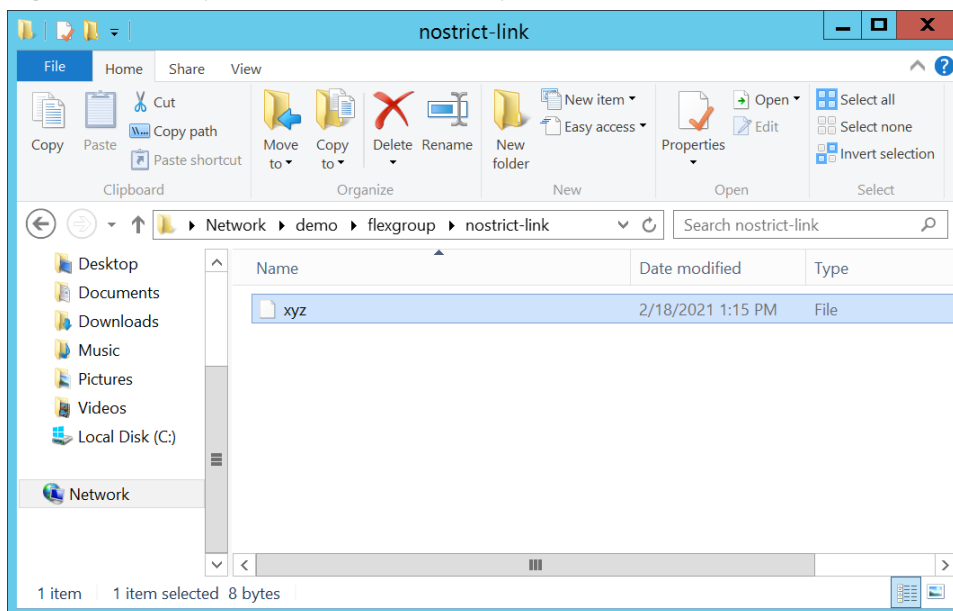
```
Could not complete the command successfully.  
SYSTEM ERROR - The system cannot find the file specified.
```

When the client asks for DFS referrals, ONTAP replies with `STATUS_NOT_FOUND`.

453	5.681603	x.x.x.x.x.x.y SMB2	212	Ioctl Request FSCTL_DFS_GET_REFERRALS, File: \\demo\flexgroup
454	5.681805	x.x.x.y.x.x.x SMB2	131	Ioctl Response, Error: STATUS_NOT_FOUND

If `no_strict_security` is not set, the link appears as a file and does not redirect traffic. To use this link without setting `no_strict_security`, the CIFS symlink path mapping must change the `-locality` to `widelink` and the CIFS share `-symlink-properties` must be changed to `symlinks_and_widelinks`.

Figure 35) CIFS symlink – `no_strict_security` no set.



CIFS symlinks, mtime behavior, and `no_strict_security`

CIFS symlinks aren't technically symlinks, they are reparse points/path redirections with path mappings that are controlled by ONTAP. How CIFS symlinks operate in ONTAP can impact how SMB clients show the mtime values of files, folders, and symlinks—and the default behavior being that the mtime of the symlink is shown, rather than the mtime of the target. mtime behavior with CIFS symlinks with ONTAP is controlled by DFS advertisements; you can change the behavior by using [no_strict_security](#) for the share's `-symlink-properties` option.

The following is an example of a symlink:

```
lrwxrwxrwx 1 root root    8 Feb 18 13:15 xyz -> /mnt/xyz
```

ONTAP was instructed to map the `/mnt/xyz` UNIX path to the CIFS path `\\DEMO\files\dir1\dir2` using a widelink through this CIFS symlink mapping.

```
cluster::*> cifs symlink show -vserver DEMO
```

Vserver	Unix Path	CIFS Server	CIFS Share	CIFS Path	Locality
DEMO	/mnt/xyz/	DEMO	files	/dir1/dir2/	widelink

The symlink lives in the FlexGroup CIFS share; therefore, it has a different volume target (the files CIFS share). As a result, `symlinks_and_widelinks` were used as the `-symlink-properties`.

```
cluster::*> cifs share show -vserver DEMO -share-name files,flexgroup -fields symlink-properties
vserver share-name symlink-properties
-----
DEMO     files      symlinks_and_widelinks
DEMO     flexgroup  symlinks_and_widelinks
```

The client sees the DFS advertisement, and this is the redirection path:

```
C:\>dfsutil diag viewdfspath \\demo\flexgroup\nostrick-link\xyz

The DFS Path <\\demo\flexgroup\nostrick-link\xyz> resolves to -> \\demo\files\dir1\dir2
```

With this configuration, the mtime of the symlink is different from the mtime of the destination directory, which can cause problems for some applications that rely on mtime in order to operate. In the following example, the directory mtime is February 18, 2021 at 12:55PM and the symlink mtime is February 18, 2021 at 1:15PM.

```
C:\>dir /T:W \\demo\files\dir1\
Volume in drive \\demo\files is files
Volume Serial Number is 80F0-4459

Directory of \\demo\files\dir1

02/18/2021  11:04 AM    <DIR>          .
02/23/2021  11:11 AM    <DIR>          ..
02/18/2021  12:55 PM    <DIR>          dir2 <<< this is the target directory

C:\>dir /T:W \\demo\flexgroup\nostrick-link\
Volume in drive \\demo\flexgroup is flexgroup
Volume Serial Number is 80F0-3768

Directory of \\demo\flexgroup\nostrick-link

02/18/2021  01:15 PM    <DIR>          .
02/18/2021  01:15 PM    <DIR>          ..
02/18/2021  01:15 PM    <DIR>          xyz <<< this is the name of the symlink
```

To see the same mtime for the link and the directory, disable the DFS advertisements for that share and instead rely on the `no_strict_security` option to redirect the link. For more information, see “No_strict_security, symlinks and DFS advertisements.” The following example shows how this symlink example was configured to show identical symlink and target directory mtime values.

To make the above scenario work successfully, complete the following steps:

1. Modify the CIFS share `-symlink-property` to `symlink,no_strict_security`.
2. Modify the CIFS symlink mapping `-locality` value to `local`.
3. Flush the [DFS caches and net use cache](#) on the client.

Dfsutil no longer advertises the path.

```
C:\>dfsutil diag viewdfspath \\demo\flexgroup\nostrick-link\xyz

Destination Path <\\demo\flexgroup\nostrick-link\xyz> is inaccessible
Could not complete the command successfully.
SYSTEM ERROR - The network location cannot be reached. For information about network
troubleshooting, see Windows Help.
```

Using the value of the destination directory only, the mtimes for the symlink and directory are now identical.

```
C:\>dir /T:W \\demo\flexgroup\nostrick-link\  
Volume in drive \\demo\flexgroup is flexgroup  
Volume Serial Number is 80F0-3768  
  
Directory of \\demo\flexgroup\nostrick-link  
  
02/18/2021  01:15 PM    <DIR>        .  
02/18/2021  01:15 PM    <DIR>        ..  
02/18/2021  12:55 PM    <DIR>        xyz  
  
C:\>dir /T:W \\demo\files\dir1\dir2  
Volume in drive \\demo\files is files  
Volume Serial Number is 80F0-4459  
  
Directory of \\demo\files\dir1\dir2  
  
02/18/2021  12:55 PM    <DIR>        .  
02/18/2021  11:04 AM    <DIR>        ..  
02/18/2021  12:55 PM    12 nostrict-link
```

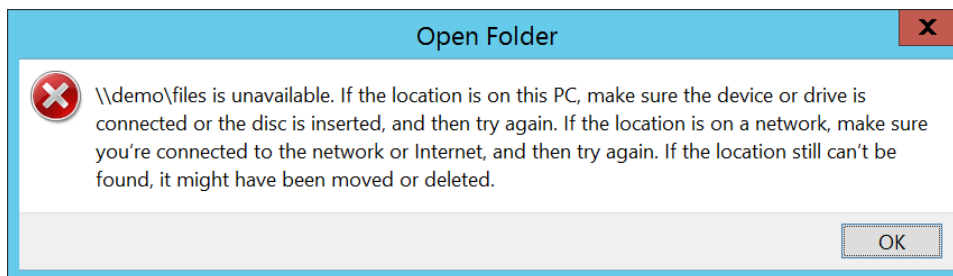
Note: To see different mtimes for symlinks and targets, you must use widelinks. However, when pointed to files, widelinks don't work consistently on SMB clients; therefore, point them to directories only.

Caches and symlink errors

While attempting to configure CIFS symlinks, you might experience issues navigating links and/or accessing CIFS shares as you find the correct path mappings. This issue can cause confusion about what the correct configuration should be, because you might see false negatives or false positives mixed in with valid issues.

For example, you might see the error message shown in Figure 36 when trying to access a CIFS/SMB share that used to work fine.

Figure 36) CIFS share access error.



The majority of these errors are going to be due to client-side and ONTAP-side caching of the SMB share paths. In those instances, you should be aware of three main caches.

Dfsutil cache

Symlinks use DFS for redirections on Windows SMB clients. By caching these paths, the client reduces the amount of network traffic required to resolve paths and improves performance. However, caches can also create scenarios in which things that appear to be working or broken are actually the opposite.

Use the [dfsutil](#) command set to view or flush these caches when you are troubleshooting. The main caches you should focus on are the provider and referral caches.

Dfsutil has a diag command flag that shows us where paths should resolve. This example uses the correctly configured widelink from the section titled, “CIFS symlink: Different volume/share (widelink).”

```
C:\>dfsutil diag viewdfspath \\demo\flexgroup\remote-link
```

The DFS Path <\\demo\flexgroup\remote-link> resolves to -> <\\demo\files\dir1\dir2\linked-dir>

This example shows what the provider cache looks like when it's populated with a symlink path.

```
C:\>dfsutil cache provider
4 entries

Max size 16384 bytes

Current size 666 bytes

Max TTL is 15m0s

\ONEWAY.NTAP.local\sysvol [TTL 8m40s]
    UNC Provider: \Device\LanmanRedirector [Priority: 1]
    Surrogate Provider: \Device\DfsClient
\demo\IPC$ [TTL 9m55s]
    UNC Provider: \Device\LanmanRedirector [Priority: 1]
    Surrogate Provider: (null)
\demo\files [TTL 9m55s]
    UNC Provider: \Device\LanmanRedirector [Priority: 1]
    Surrogate Provider: \Device\DfsClient
\demo\flexgroup [TTL 12m13s]
    UNC Provider: \Device\LanmanRedirector [Priority: 1]
    Surrogate Provider: \Device\DfsClient
```

In the above example, the provider cache offers TTL information—this is how long that entry lives in the cache if it remains unused.

The following example is the referral cache:

```
C:\>dfsutil cache referral
Entry: \demo\files\abs-link
ShortEntry: \demo\files\abs-link
Expires in 0 seconds
UseCount: 0 Type:0x1 ( DFS )
    0:[\DEMO\files\dir1\dir2\linked-dir] AccessStatus: 0 ( ACTIVE )

Entry: \demo\flexgroup\remote-link
ShortEntry: \demo\flexgroup\remote-link
Expires in 1651 seconds
UseCount: 0 Type:0x1 ( DFS )
    0:[\DEMO\files\dir1\dir2\linked-dir] AccessStatus: 0 ( ACTIVE )

Entry: \demo\files
ShortEntry: \demo\files
Expires in 1502 seconds
UseCount: 1 Type:0x81 ( REFERRAL_SVC DFS )
    0:[\demo\files] AccessStatus: 0 ( ACTIVE )

Entry: \demo\flexgroup
ShortEntry: \demo\flexgroup
Expires in 1639 seconds
UseCount: 1 Type:0x81 ( REFERRAL_SVC DFS )
    0:[\demo\flexgroup] AccessStatus: 0 ( ACTIVE )
```

This list displays the link entries and how long they'll remain in the cache until they expire. The abs-link expires in 0 seconds, which means it will never expire on its own. You must manually clear the cache to remove the entry.

To flush these caches, run the following commands:

```
C:\>dfsutil cache provider flush
C:\>dfsutil cache referral flush
```

Net use

In addition to DFS caches, SMB clients also cache SMB connections and credentials. You can view and manage these caches using `net use`.

To view cached CIFS and SMB connections, run the following command:

```
C:\>net use
```

To clear individual cached connections or disconnect mapped drives, run the following commands:

```
C:\>net use /d \\SERVER\share
C:\>net use /d Z:
```

To clear all cached connections and disconnect all mapped drives, run the following command:

```
C:\>net use /d *
```

Note: Running this command in conjunction with `dfsutil` produces better results than running the command first and then flushing the `dfsutil` caches. In some instances, when you run `net use /d`, you might find that the CIFS/SMB share is inaccessible (as seen in Figure 36). In those instances, close the Windows Explorer window, flush the caches again (`net use` and `dfsutil`), and then retry the connection.

ONTAP path component cache

ONTAP provides path caching for both CIFS shares and symlinks. These caches are controlled by the following CIFS server options in Diagnostic Privilege. If needed, you can disable or enable the caches for troubleshooting purposes, but they should remain enabled for normal production workloads unless otherwise directed by NetApp support.

```
[-is-path-component-cache-enabled {true|false}] - Is Path Component Cache Enabled (privilege:
advanced)
This optional parameter specifies whether the path component cache is enabled. The default value
for this parameter is true.

[-is-path-component-cache-symlink-enabled {true|false}] - Is Path Component Cache Symlink
Resolution Enabled (privilege: diagnostic)
This optional parameter specifies whether the symlink resolution for the path component cache is
enabled. The default value of this parameter is true.
```

You can configure the values for these caches. However, leave the default values unless directed otherwise by NetApp support.

```
[-path-component-cache-max-entries <integer>] - Path Component Cache Maximum Entries (privilege:
diagnostic)
This optional parameter specifies the maximum number of entries in an instance of the path
component cache. The default value of this parameter is 5000. The maximum value of this parameter
is 10000.

[-path-component-cache-entry-exp-time <integer>] - Path Component Cache Entry Expiration Time
(privilege: diagnostic)
This optional parameter specifies the maximum expiration time in milliseconds of an entry in the
path component cache. The default value of this parameter is 15000 (15 seconds). The maximum
value of this parameter is 3600000 (1 hour).

[-path-component-cache-symlink-exp-time <integer>] - Path Component Cache Symlink Expiration Time
(privilege: diagnostic)
```

This optional parameter specifies the maximum expiration time in milliseconds of an entry that is a symlink in the path component cache. The default value of this parameter is 15000 (15 seconds). The maximum value of this parameter is 3600000 (1 hour).

`[-path-component-cache-max-session-token-size <integer>]` - Path Component Cache Maximum Session Token Size (privilege: diagnostic)

This optional parameter specifies the maximum session token size for the path component cache. The default value of this parameter is 1000. The maximum value of this parameter is 10000.

To enable statistics for these caches, run the following command in Diagnostic Privilege:

```
cluster::*> statistics start -counter component_cache -object cifs -vserver DEMO
```

To view these statistics, run the following commands:

```
cluster::*> statistics show -object cifs
Object: cifs
Instance: DEMO
Start-time: 2/16/2021 11:10:00
End-time: 2/16/2021 12:29:01
Elapsed-time: 4740s
Scope: DEMO
```

Counter	Value
-----	-----
component_cache	-
Total Components	166
Total Tests	140
Total Hits	19
Junction Hits	0
Symlink Hits	9
No Cache Miss	51
Not Allowed Miss	0
Expired Miss	61
Expired Sym Res Miss	1
Unresolved Junc Miss	0
Unresolved Sym Miss	8
Total Additions	142
Addition Session List	140
Total Purged	0
Total Stale	0
Total Deletions	99

Junction paths and reparse points

ONTAP uses [junction paths](#) within an SVM namespace to direct NAS clients between volumes mounted in the same namespace. Every SVM's namespace starts at the SVM root volume (vsroot) and has a path of `/`. NFS clients can mount that path, provided it is exported to them, and SMB clients can access `/` with the C\$ hidden share that is created by default when CIFS is configured.

By default, ONTAP junction paths appear as [reparse points](#) (essentially a symlink or shortcut) by way of the Advanced Privilege ONTAP CIFS options `is-use-junctions-as-reparse-points-enabled`. [CIFS symlinks](#) appear as shortcuts to SMB 1.0 clients, but as directories to SMB 2.x and 3.x clients, due to how the `widelink-as-reparse-point-versions` option is configured.

The following default settings are for the junction path and reparse point options:

```
cluster::*> cifs options show -vserver DEMO -fields is-use-junctions-as-reparse-points-
enabled,widelink-as-reparse-point-versions
vserver is-use-junctions-as-reparse-points-enabled widelink-as-reparse-point-versions
-----
DEMO    true                                SMB1
```

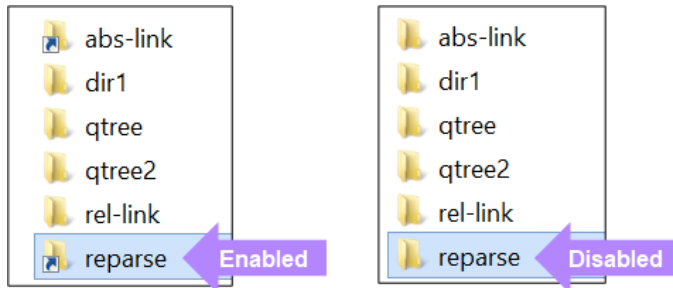
If you want to see symlinks appear as shortcut files on SMB 2.x and SMB 3.x clients, modify the `widelink-as-reparse-point-versions` option to include the desired SMB versions.

```
cluster::*> cifs options modify -vserver DEMO -widelink-as-reparse-point-versions SMB
```


If you want junction paths to appear as directories to SMB clients, disable `is-use-junctions-as-reparse-points-enabled`.

```
cluster::*> cifs options modify -vserver DEMO -is-use-junctions-as-reparse-points-enabled false
```

Figure 37) Junction path views: Reparse point enabled versus disabled.



The following examples show how a volume junction path is displayed at the `cmd` prompt with the option enabled and disabled.

is-use-junctions-as-reparse-points-enabled true

```
C:\>dir \\demo\C$
Volume in drive \\demo\C$ is c$
Volume Serial Number is 80F0-3712

Directory of \\demo\C$

02/02/2021  01:09 PM  <DIR>          .
02/02/2021  01:09 PM  <DIR>          ..
07/18/2017  08:37 AM  <JUNCTION>     home [\\?\Volume{80F03713-0000-0000-5879-48F200000040}\]
01/10/2019  09:25 AM  <JUNCTION>     var [\\?\Volume{80F03AA7-0000-0000-5C37-55C400000040}\]
03/09/2017  11:24 AM  <JUNCTION>     flexvol [\\?\Volume{80F0372F-0000-0000-58C181B200000040}\]
```

is-use-junctions-as-reparse-points-enabled false

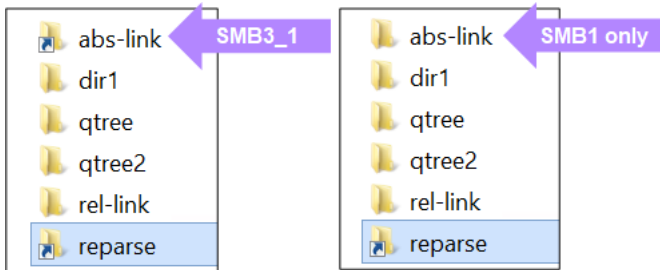
```
C:\>dir \\demo\C$
Volume in drive \\demo\C$ is c$
Volume Serial Number is 80F0-3712

Directory of \\demo\C$

02/02/2021  01:09 PM  <DIR>          .
02/02/2021  01:09 PM  <DIR>          ..
07/18/2017  08:37 AM  <DIR>          home
01/10/2019  09:25 AM  <DIR>          var
03/09/2017  11:24 AM  <DIR>          flexvol
```

The `widelink-as-reparse-point-versions` option controls how the symlinks are viewed. In Figure 38, SMB 3.1 is used as the access protocol. The `abs-link` symlink displays as a shortcut from the SMB 3.1 client when SMB3_1 is added to the option and displays as a folder when the option is left as the default SMB1 value.

Figure 38) Symlink views: Reparse point enabled versus disabled.



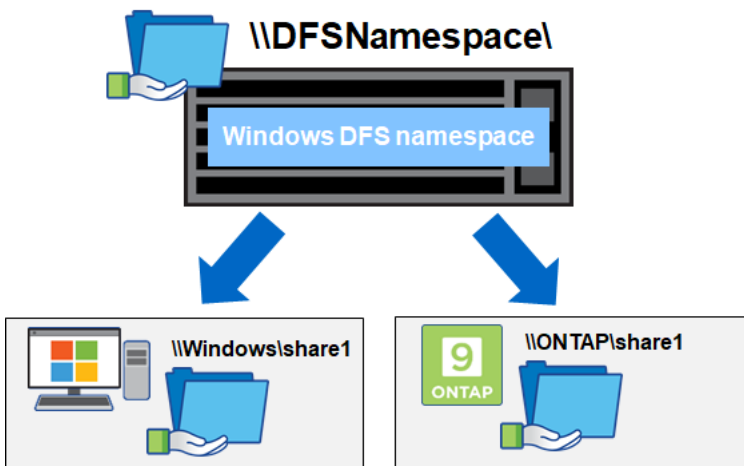
Note: In some instances, if a symlink is not set as a reparse point, some clients might not see the proper amount of free space in the destination path. For more information, see the Knowledge Base article titled [Unable to write to a Symlink CIFS volume due to "not enough space" error](#).

Distributed File Systems

Microsoft Windows supports a functionality called [DFS](#). This feature allows a Windows server to act as a redirector for endpoints for SMB shares, regardless of where those shares reside. DFS targets can be other Windows servers, ONTAP CIFS shares, or other storage system CIFS shares. This feature allows administrators to present a single namespace to CIFS/SMB clients to navigate without end users needing to know multiple IP addresses or host names to access. Instead, everyone connects to the same server and DFS does the rest.

Figure 39 shows Windows DFS with ONTAP as a target.

Figure 39) Windows DFS with ONTAP as a target.



ONTAP supports acting as a DFS target, but does not currently support the DFS-R ([replication](#)) feature. For synchronized SMB share layout across multiple sites, FlexCache volumes using SMB might be a good option.

FlexCache volumes with SMB

Starting in ONTAP 9.8, the SMB protocol is supported at the cache volume so that an SMB share pointing to a FlexCache volume can be created. As with export policies, SMB shares are not replicated with the creation of a FlexCache volume. They are also independent, even if you are creating a FlexCache volume in the same SVM. This also means that different share permissions can be implemented at the

cache. It allows for granular control of cache data access and can also limit caches to read-only if there is a need.

When using SMB shares to access FlexCache data, the origin volume security style will most likely be NTFS. NTFS ACLs and share permission application is highly dependent on the SMB server's configuration in ONTAP so there are a few requirements to ensure that permissions are applied the same at the origin and at the cache. For more information and best practices regarding FlexCache volumes, SMB and multiprotocol NAS, see [TR-4743](#).

Native CIFS and NFS file auditing

ONTAP supports native file and folder auditing for both CIFS/SMB and NFS protocols. With file/folder auditing, storage administrators can track when files are accessed, modified or deleted in the NAS file system without needing to purchase third party monitoring tools.

You can control the NFS and CIFS/SMB auditing by setting the audit ACLs on the volume or folder you want to audit. You can decide whether to save the audit logs as either XML or EVT files, and then store them within the same data volume that you are auditing.

For more information about NFS and CIFS auditing, see the following resources:

- [Deciding whether to use the SMB/CIFS and NFS Auditing and Security Tracing Guide](#)
- [Auditing NAS events on SVMs](#)
- [ONTAP Native NAS Auditing \(SMB and NFS\)](#)

Troubleshooting multiprotocol NAS

This section provides some common multiprotocol NAS issues, as well as the commands to use when troubleshooting issues in ONTAP.

NFS user nfsnobody

In some instances, NFS clients might see file owner/group information display in file listings as nfsnobody.

```
# ls -la | grep newfile
-rwxrwxrwx 1 nfsnobody nfsnobody 0 May 19 13:30 newfile.txt
```

When you list the file with numerics, the owner:group is 65534.

```
# ls -lan | grep newfile
-rwxrwxrwx 1 65534 65534 0 May 19 13:30 newfile.txt
```

On most Linux clients, the user 65534 is nfsnobody. In ONTAP, the user is pcuser.

```
cluster::*> unix-user show -vserver DEMO -id 65534
      User      User  Group  Full
Vserver  Name      ID    ID    Name
-----
DEMO     pcuser      65534 65534
```

pcuser is also the default anonymous user in export policy rules.

```
cluster::*> export-policy rule show -vserver DEMO -policyname default -fields anon
vserver policyname ruleindex anon
-----
DEMO     default      1      65534
DEMO     default      2      65534
DEMO     default      3      65534
```

For the ONTAP cluster file permissions, you might see that the UNIX owner is 65534, but that there are also Windows ACLs and owners that are different.

```
cluster::*> vserver security file-directory show -vserver DEMO -path /data/newfile.txt

      Vserver: DEMO
      File Path: /data/newfile.txt
      File Inode Number: 7088
      Security Style: ntfs
      Effective Style: ntfs
      DOS Attributes: 20
      DOS Attributes in Text: ---A---
      Expanded Dos Attributes: -
      UNIX User Id: 65534
      UNIX Group Id: 65534
      UNIX Mode Bits: 777
      UNIX Mode Bits in Text: rwxrwxrwx
      ACLs: NTFS Security Descriptor
            Control:0x8004
            Owner:NTAP\ntfs
            Group:NTAP\DomainUsers
            DACL - ACEs
                  ALLOW-Everyone-0x1f01ff- (Inherited)
```

If `nfsnobody` or 65534 is displayed in the NFS listings, one of two things is more than likely happening:

- The volume that is being exported to NFS clients is also used by Windows SMB clients, and the Windows users writing to the shares don't map to valid UNIX users and/or groups.
- The volume that is being exported to NFS clients has the anonymous user set to 65534, and something is causing the NFS user to squash to the anonymous user. For more information about user squashing, see [TR-4067: NFS in NetApp ONTAP](#).

To view the Windows-user-to-UNIX-user mapping, run the following command in Advanced Privilege:

```
cluster::*> access-check name-mapping show -vserver DEMO -direction win-unix -name ntfs
'ntfs' maps to 'pcuser'

cluster::*> access-check name-mapping show -vserver DEMO -direction win-unix -name prof1
'prof1' maps to 'prof1'
```

Viewing and managing NFS credentials

In ONTAP 9.3, a global cache for name services was implemented to offer better performance, reliability, resilience, and supportability for NAS credentials and name service server management.

One of those changes was the implementation of the NFS credential cache, which stores NFS user and group information in ONTAP when NFS exports are accessed.

To view and manage these caches, run the `nfs credentials` commands in Advanced Privilege.

```
cluster::*> nfs credentials ?
count          *Count credentials cached by NFS
flush          *Flush credentials cached by NFS
show           *Show credentials cached by NFS
```

Cache entries populate the node where the TCP connection for the NFS mount exists. To view this information, run the following commands on the cluster:

```
cluster::*> nfs connected-clients show -vserver DEMO -client-ip x.x.x.x -fields data-lif-ip -
volume scripts
node          vserver data-lif-ip  client-ip    volume  protocol
-----
Node1         DEMO      x.x.x.y      x.x.x.x    scripts nfs3
```

The command above indicates that the client IP `x.x.x.x` is connected to a data LIF on node1. This information allows you to narrow down which node to focus on for cache entries.

The `nfs credentials count` command allows you to see how many credentials are currently stored in the NFS credential cache. This information helps you to understand the impact of clearing the cache.

```
cluster::*> nfs credentials count -node node1
Number of credentials cached by NFS on node "node1": 4
```

If a user traverses into an ONTAP NFS export, user IDs, group IDs, and so on are all added to the NFS credential cache. In this example, the user named is `prof1`.

```
# id prof1
uid=1102(prof1) gid=10002(ProfGroup) groups=10002(ProfGroup),10000(Domain
Users),1202(group2),1101(group1),1220(sharedgroup),1203(group3)
```

That user has eight different entries: one numeric UID and seven group memberships. The user `prof1` then accesses an NFS export. The credential cache increases by eight.

```
cluster::*> nfs credentials count -node node1
Number of credentials cached by NFS on node "node1": 12
```

This count is for the entire node, not per SVM. If you have multiple SVMs in your environment, the count might not be useful for troubleshooting.

Viewing the NFS credential cache

In addition to showing how many credentials are in the NFS credential cache, you can also view individual cache entries for users and/or groups. If a user in your environment is having access issues, you can look for that user in the cache.

Note: You cannot view the contents of entire credential cache.

In this example, `prof1` accesses the mount. You can view that cache entry as well as the flags that tell you more about the cache entry.

```
cluster::*> nfs credentials show -node node1 -vserver DEMO -unix-user-name prof1

Credentials
-----
                Node: node1
                Vserver: DEMO
                Client IP: -
                Flags: unix-extended-creds-present, id-name-mapping-present
Time since Last Refresh: 52s
Time since Last Access: 44s
Hit Count: 4

UNIX Credentials:
    Flags: 1
    Domain ID: 0
    UID: 1102
    Primary GID: 10002
    Additional GIDs: 10002
                    10000
                    1101
                    1202
                    1203
                    1220

Windows Credentials:
    Flags: -
    User SID: -
    Primary Group SID: -
    Domain SIDs: -
```

```
ID-Name Information:
    Type: user
    ID: 1102
    Name: prof1
```

You can view the entry for the user's primary group.

```
cluster::*> nfs credentials show -node node1 -vserver DEMO -unix-group-name ProfGroup

Credentials
-----
    Node: node1
    Vserver: DEMO
    Client IP: -
    Flags: id-name-mapping-present
    Time since Last Refresh: 64s
    Time since Last Access: 6s
    Hit Count: 2

UNIX Credentials:
    Flags: -
    Domain ID: -
    UID: -
    Primary GID: -
    Additional GIDs: -

Windows Credentials:
    Flags: -
    User SID: -
    Primary Group SID: -
    Domain SIDs: -

ID-Name Information:
    Type: group
    ID: 10002
    Name: ProfGroup
```

You can view the credential cache entries for users and groups down to the client IP that attempted the access.

```
cluster::*> nfs credentials show -node node1 -vserver DEMO -client-ip x.x.x.x -unix-user-id 1102

Credentials
-----
    Node: node1
    Vserver: DEMO
    Client IP: x.x.x.x
    Flags: unix-extended-creds-present, id-name-mapping-present
    Time since Last Refresh: 35s
    Time since Last Access: 34s
    Hit Count: 2
    Reference Count: 4
    Result of Last Update Attempt: no error

UNIX Credentials:
    Flags: 1
    Domain ID: 0
    UID: 1102
    Primary GID: 10002
    Additional GIDs: 10002
    10000
    1101
    1202
    1203
    1220

Windows Credentials:
    Flags: -
    User SID: -
    Primary Group SID: -
```

```

Domain SIDs: -

ID-Name Information:
    Type: user
    ID: 1102
    Name: prof1

```

The credential cache also keeps negative entries (entries that could not be resolved) in cache. Negative entries occur when ONTAP can't resolve the numeric UID to a valid user. In this instance, the UID 1236 cannot be resolved by ONTAP, but attempted access to the NFS export.

```

# su cifsuser
bash-4.2$ cd /scripts/
bash: cd: /scripts/: Permission denied
bash-4.2$ id
uid=1236(cifsuser) gid=1236(cifsuser) groups=1236(cifsuser)

cluster::*> nfs credentials show -node node1 -vserver DEMO -unix-user-id 1236

Credentials
-----
                Node: node1
                Vserver: DEMO
                Client IP: -
                Flags: no-unix-extended-creds, no-id-name-mapping
Time since Last Refresh: 33s
Time since Last Access: 7s
Hit Count: 15

UNIX Credentials:
    Flags: -
    Domain ID: -
    UID: -
    Primary GID: -
    Additional GIDs: -

Windows Credentials:
    Flags: -
    User SID: -
    Primary Group SID: -
    Domain SIDs: -

ID-Name Information:
    Type: -
    ID: -
    Name: -

```

NFS credential cache with NFSv4.x and multiprotocol NAS

The NFS credential cache entries also store Windows credentials and NFSv4 ID mapping credentials.

If the user traverses NFSv4.x exports and maps into the ID domain correctly, the ID-Name Information field is populated:

```

Credentials
-----
                Node: node
                Vserver: DEMO
                Client IP: x.x.x.x
                Flags: unix-extended-creds-present, id-name-mapping-present
Time since Last Refresh: 12s
Time since Last Access: 9s
Hit Count: 2
Reference Count: 4
Result of Last Update Attempt: no error

UNIX Credentials:
    Flags: 1

```

```

        Domain ID: 0
        UID: 1102
        Primary GID: 10002
        Additional GIDs: 10002
                        10000
                        1101
                        1202
                        1203
                        1220

Windows Credentials:
    Flags: -
    User SID: -
    Primary Group SID: -
    Domain SIDs: -

ID-Name Information:
    Type: user
    ID: 1102
    Name: prof1

```

If the user accesses an export that has NTFS permissions/security style, the flag `cifs-creds-present` is displayed, as well as the domain SID information under Windows Credentials:

```

Credentials
-----
        Node: node1
        Vserver: DEMO
        Client IP: x.x.x.x
        Flags: ip-qualifier-configured, unix-extended-creds-present, cifs-creds-
present
        Time since Last Refresh: 19s
        Time since Last Access: 1s
        Hit Count: 9
        Reference Count: 2
        Result of Last Update Attempt: no error

UNIX Credentials:
    Flags: 0
    Domain ID: 0
    UID: 1102
    Primary GID: 10002
    Additional GIDs: 10002
                    10000
                    1101
                    1202
                    1203
                    1220

Windows Credentials:
    Flags: 8320
    User SID: S-1-5-21-3552729481-4032800560-2279794651-1214
    Primary Group SID: S-1-5-21-3552729481-4032800560-2279794651-513
    Domain SIDs: S-1-5-21-3552729481-4032800560-2279794651
                S-1-18
                S-1-1
                S-1-5
                S-1-5-32

ID-Name Information:
    Type: -
    ID: -
    Name: -

```

NFS credential cache settings

The timeout value of the NFS credential cache is controlled by the NFS server options listed in Table 6.

Table 6) NFS credential cache settings.

Option	What It Does	Default Value (ms)
-cached-cred-negative-ttl	This optional parameter specifies the age of the negative cached credentials after which they are cleared from the cache. The value specified must be between 60,000 and 604,800,000.	7,200,000ms
-cached-cred-positive-ttl	This optional parameter specifies the age of the positive cached credentials after which they are cleared from the cache. The value specified must be between 60,000 and 604,800,000.	86,400,000ms (24 hours)
-cached-cred-harvest-timeout	This optional parameter specifies the harvest timeout for cached credentials. The value specified must be between 60,000 and 604,800,000.	86,400,000ms (24 hours)

Cache entries maintain the time since last access/refresh (as seen in the `show` command). If an entry stays idle for a period of time, it eventually is removed from the cache. If the entry is active, it gets refreshed and stays in cache.

These values can be modified to longer or shorter timeout values, depending on the desired effects:

- **Longer cache timeout values** reduce network load and provide faster lookups of users, but can produce more false positives/false negatives as the cache entries are not always in sync with name services.
- **Shorter cache timeout values** increase load on the network and name servers, and can add some latency to name lookups (depending on name service source), but offer more accurate and up-to-date entries.

The NetApp best practice is to leave the values as is. If you need to change the values, be sure to monitor the results and adjust as needed.

Flushing the NFS credential cache

In instances where a user has been added or removed from a group and does not have the desired access, the credential cache entry can be flushed manually, rather than waiting for the cache entry to timeout.

The following command can be run for a UNIX user or numeric ID or a UNIX group or numeric ID. Additionally, this command can be run as granularly as down to the client IP address having the issue.

```
cluster::*> nfs credentials flush -node node1 -vserver DEMO -client-ip x.x.x.x -unix-user-id 1102
Number of matching credentials flushed: 2
```

Note: You can only flush one NFS credential cache entry at a time.

The NFS credential cache is separate from the name-service cache. For information on managing the name-service caches, see [TR-4835: How to Configure LDAP in ONTAP](#).

Export policy rules: Caching

To reduce the number of requests made to a cluster, export policy rules, client host names, and netgroup information are all cached in ONTAP. This feature improves the request performance, as well as alleviates the load on networks and name service servers.

Clientmatch caching

When a clientmatch entry is cached, it remains local to the SVM and then flushes after the cache timeout period is reached or if the export policy rule table is modified. The default cache timeout period is dependent on the version of ONTAP and can be verified by running the command `export-policy access-cache config show` in Admin Privilege.

These are the default values:

```
TTL For Positive Entries (Secs): 3600
TTL For Negative Entries (Secs): 3600
Harvest Timeout (Secs): 86400
```

To view a specific client in the export policy access-cache, run the following commands in Advanced Privilege:

```
cluster::*> export-policy access-cache show -node node-02 -vserver NFS -policy default -address
x.x.x.x

Node: node-02
Vserver: NFS
Policy Name: default
IP Address: x.x.x.x
Access Cache Entry Flags: has-usable-data
Result Code: 0
First Unresolved Rule Index: -
Unresolved Clientmatch: -
Number of Matched Policy Rules: 1
List of Matched Policy Rule Indexes: 2
Age of Entry: 11589s
Access Cache Entry Polarity: positive
Time Elapsed since Last Use for Access Check: 11298s
Time Elapsed since Last Update Attempt: 11589s
Result of Last Update Attempt: 0
List of Client Match Strings: 0.0.0.0/0
```

Host name/DNS caching

When a clientmatch is set to a host name, the name is then resolved to an IP address. This process is based on the order the SVM's name service-switch (ns-switch) uses. For example, if the ns-switch host database is set to `files, dns`, then ONTAP searches for the client match in the local host files and then searches DNS.

After a name lookup, ONTAP caches the result in the hosts cache. This cache's settings are configurable and can be queried and flushed from the ONTAP CLI in Advanced Privilege.

To query the cache, run the following commands:

```
cluster::*> name-service cache hosts forward-lookup show -vserver NFS
(vserver services name-service cache hosts forward-lookup show)

Vserver  Host          IP          Address IP          Create      TTL(sec)
-----  -
NFS      centos7.ntap.local
          Any          Ipv4       x.x.x.x  dns          3/26/2020  3600
                                           16:31:11
```

To view the hosts cache settings, run the following commands:

```
cluster::*> name-service cache hosts settings show -vserver NFS -instance
(vserver services name-service cache hosts settings show)

Vserver: NFS
Is Cache Enabled?: true
Is Negative Cache Enabled?: true
Time to Live: 24h
```

```
Negative Time to Live: 1m
Is TTL Taken from DNS: true
```

In some instances, if an NFS client's IP address changes, the host's entry might need to be flushed to correct access issues.

To flush a hosts cache entry, run the following commands:

```
cluster::*> name-service cache hosts forward-lookup delete -vserver NFS ?
             -host      -protocol -sock-type -flags      -family
```

Netgroup caching

If you are using netgroups in the clientmatch field for export rules, then ONTAP does additional work to contact the netgroup name service server to unpack the netgroup information. The netgroup database in ns-switch determines the order in which ONTAP queries for netgroups. In addition, the method that ONTAP uses for netgroup support depends on whether netgroup.byhost support is enabled or disabled. For more information about netgroup.byhost, see [TR-4835: How to Configure LDAP in ONTAP](#).

- If netgroup.byhost is disabled, then ONTAP queries the entire netgroup and populates the cache with all netgroup entries. If the netgroup has thousands of clients, then the process can take additional time to complete. Netgroup.byhost is disabled by default.
- If netgroup.byhost is enabled, then ONTAP queries the name service only for the host entry and the associated netgroup mapping. This process greatly reduces the amount of time required to query for netgroups because you don't need to look up potentially thousands of clients.

These entries are added to the netgroup cache, which you can find by running the `vserver services name-service cache` commands. These cache entries can be viewed or flushed, and the timeout values can be configured.

To view the netgroups cache settings, run the following commands:

```
cluster::*> name-service cache netgroups settings show -vserver NFS -instance
(vserver services name-service cache netgroups settings show)

                Vserver: NFS
                Is Cache Enabled?: true
Is Negative Cache Enabled?: true
                Time to Live: 24h
                Negative Time to Live: 1m
                TTL for netgroup members: 30m
```

When an entire netgroup is cached, it is placed in the members cache.

```
cluster::*> name-service cache netgroups members show -vserver DEMO -netgroup netgroup1
(vserver services name-service cache netgroups members show)

                Vserver: DEMO
                Netgroup: netgroup1
                Hosts: sles15-1,x.x.x.x
                Create Time: 3/26/2020 12:40:56
                Source of the Entry: ldap
```

When only a single netgroup entry is cached, the `ip-to-netgroup` and `hosts reverse-lookup` caches are populated with the following entry:

```
cluster::*> name-service cache netgroups ip-to-netgroup show -vserver DEMO -host x.x.x.y
(vserver services name-service cache netgroups ip-to-netgroup show)
Vserver  IP Address  Netgroup  Source  Create Time
-----
DEMO     x.x.x.y      netgroup1  ldap    3/26/2020 17:13:09

cluster::*> name-service cache hosts reverse-lookup show -vserver DEMO -ip x.x.x.y
```

(vserver services name-service cache hosts reverse-lookup show)						
Vserver	IP Address	Host	Source	Create Time	TTL(sec)	
DEMO	x.x.x.y	centos8-ipa.centos-ldap.local	dns	3/26/2020 17:13:09	3600	

Cache timeout modification considerations

Cache configurations can be modified to different values if needed:

- **Increasing** the timeout values keep cache entries longer, but can result in inconsistencies in client access if a client changes its IP address. For example, if DHCP is used for client IP addresses and DNS is not updated or the export rule uses IP addresses.
- **Decreasing** the timeout values flushes the cache more frequently for more up-to-date information, but could add additional load to name service servers and add latency to mount requests from clients.

In most cases, leaving the cache timeout values intact is the best approach. For more information and guidance, see [TR-4668: Name Services Best Practices](#) and [TR-4835: How to Configure LDAP in ONTAP](#).

Exportfs support

In ONTAP, `exportfs` is replaced by the `export-policy` and `name-service cache` commands. When running `exportfs`, the following output is displayed:

```
"exportfs" is not supported: use the "vserver export-policy" command.
```

Commands to troubleshoot permissions issues

In most instances, NFS permission issues are fairly straightforward; NFSv3 uses basic rwx mode bits. However, things are more complex when NFSv4 ACLs and/or multiprotocol NAS access and different security styles are involved. This section provides some useful commands for troubleshooting permissions issues in NAS environments. For name service cache information, see “Viewing and managing NFS credentials.” For more detailed information, see [TR-4835: LDAP in NetApp ONTAP](#).

Verifying UNIX UIDs and group memberships

For NFSv3 operations, the UNIX user names and group names aren't hugely important, because the numerics can be passed to verify identity. However, with NFSv4 and NTFS security style objects, numeric IDs must translate to valid UNIX user and group names for proper name resolution. For NFSv4, this numeric-ID-to-name mapping/translation is needed to avoid squashing a user to [nobody](#). In NTFS security styles, the UNIX user name is required to map to a valid Windows user name.

In ONTAP, there are a several commands that you can use to see a UNIX user's ID and group memberships.

For local UNIX users and groups, run the following commands:

```
cluster::> unix-user show
cluster::> unix-group show
```

For basic UID/GID information for all UNIX users (local and name services; Advanced Privilege), run the following command:

```
cluster::*> access-check authentication show-ontap-admin-unix-creds
```

Or run these commands:

```
cluster::*> getxxbyyy getpwbyname -node node1 -vserver DEMO -username profl -show-source true
(vserver services name-service getxxbyyy getpwbyname)
Source used for lookup: LDAP
```

```

pw_name: prof1
pw_passwd:
pw_uid: 1102
pw_gid: 10002
pw_gecos:
pw_dir:
pw_shell:

cluster::*> getxxbyyy getpwbyname -node node1 -vserver DEMO -username host -show-source true
(vserver services name-service getxxbyyy getpwbyname)
Source used for lookup: Files
pw_name: host
pw_passwd: *
pw_uid: 598
pw_gid: 0
pw_gecos:
pw_dir:
pw_shell:

```

To view user information and group memberships (local and name services; Advanced Privilege), run the following commands:

```

cluster::*> getxxbyyy getgrlist -node node1 -vserver DEMO -username prof1
(vserver services name-service getxxbyyy getgrlist)
pw_name: prof1
Groups: 10002 10002 10000 1101 1202 1203 48

```

Viewing user and group information for multiprotocol users

If you have both CIFS/SMB and NFS configured in your environment, you can get a full list of user names, name mapping, IDs, group names, privileges and group memberships from a single command in Advanced Privilege. This is the preferred command to use in multiprotocol environments. The command does not work if there are no SMB/CIFS servers configured.

```

cluster::*> access-check authentication show-creds -node node1 -vserver DEMO -unix-user-name
prof1 -list-name true -list-id true
(vserver services access-check authentication show-creds)

UNIX UID: 1102 (prof1) <> Windows User: S-1-5-21-3552729481-4032800560-2279794651-1110
(NTAP\prof1 (Windows Domain User))

GID: 10002 (ProfGroup)
Supplementary GIDs:
  10002 (ProfGroup)
  10000 (Domain Users)
  1101 (group1)
  1202 (group2)
  1203 (group3)
  48 (apache-group)

Primary Group SID: S-1-5-21-3552729481-4032800560-2279794651-1111 NTAP\ProfGroup (Windows
Domain group)

Windows Membership:
S-1-5-21-3552729481-4032800560-2279794651-1301 NTAP\apache-group (Windows Domain group)
S-1-5-21-3552729481-4032800560-2279794651-1106 NTAP\group2 (Windows Domain group)
S-1-5-21-3552729481-4032800560-2279794651-513 NTAP\DomainUsers (Windows Domain group)
S-1-5-21-3552729481-4032800560-2279794651-1105 NTAP\group1 (Windows Domain group)
S-1-5-21-3552729481-4032800560-2279794651-1107 NTAP\group3 (Windows Domain group)
S-1-5-21-3552729481-4032800560-2279794651-1111 NTAP\ProfGroup (Windows Domain group)
S-1-5-21-3552729481-4032800560-2279794651-1231 NTAP\local-group.ntap (Windows Alias)
S-1-18-2 Service asserted identity (Windows Well known group)
S-1-5-32-551 BUILTIN\Backup Operators (Windows Alias)
S-1-5-32-544 BUILTIN\Administrators (Windows Alias)
S-1-5-32-545 BUILTIN\Users (Windows Alias)
User is also a member of Everyone, Authenticated Users, and Network Users

Privileges (0x22b7):

```

```
SeBackupPrivilege
SeRestorePrivilege
SeTakeOwnershipPrivilege
SeSecurityPrivilege
SeChangeNotifyPrivilege
```

Showing file permissions as seen by ONTAP

When troubleshooting permission issues, you might not have access to view permissions from a NAS client. Or you might want to verify what the NAS client is seeing for permissions with what ONTAP is seeing. To do that, run the following commands:

```
cluster::> file-directory show -vserver DEMO -path /home/profl
(vserver security file-directory show)

      Vserver: DEMO
      File Path: /home/profl
      File Inode Number: 8638
      Security Style: ntfs
      Effective Style: ntfs
      DOS Attributes: 10
DOS Attributes in Text: ----D---
Expanded Dos Attributes: -
      UNIX User Id: 0
      UNIX Group Id: 0
      UNIX Mode Bits: 777
UNIX Mode Bits in Text: rwxrwxrwx
      ACLs: NTFS Security Descriptor
            Control:0x8504
            Owner:NTAP\profl
            Group:BUILTIN\Administrators
            DACL - ACEs
                  ALLOW-Everyone-0x1f01ff-OI|CI
                  ALLOW-NTAP\profl-0x1f01ff-OI|CI
                  ALLOW-NTAP\sharedgroup-0x1200a9-OI|CI
                  ALLOW-NTAP\Administrator-0x1f01ff-OI|CI
```

You can also verify which effective permissions a specific user has to a specific file or directory by running the following commands:

```
cluster::> file-directory show-effective-permissions -vserver DEMO -unix-user-name profl -path
/home/profl
(vserver security file-directory show-effective-permissions)

      Vserver: DEMO
      Windows User Name: NTAP\profl
      Unix User Name: profl
      File Path: /home/profl
      CIFS Share Path: -
Effective Permissions:
      Effective File or Directory Permission: 0x1f01ff
      Read
      Write
      Append
      Read EA
      Write EA
      Execute
      Delete Child
      Read Attributes
      Write Attributes
      Delete
      Read Control
      Write DAC
      Write Owner
      Synchronize
```

Checking export policy access

In some instances, permission issues are attributed to the export policy settings. For example, if your policy is set to allow only reads, then that setting can override any user permissions set on the mount.

ONTAP provides a method to verify the export policy access for the client by running the following command:

```
cluster::> export-policy check-access
```

Using security tracing

To trace permissions issues as they occur, use the security trace filter functionality to trace both NFS and SMB/CIFS permissions.

To create a trace filter, run the following commands:

```
cluster::> vserver security trace filter create ?
  -vserver <vserver name>          Vserver
  [-index] <integer>                Filter Index
  [[-protocols] {cifs|nfs}, ...]    Protocols (default: cifs)
  [ -client-ip <IP Address> ]        Client IP Address to Match
  [ -path <TextNoCase> ]             Path
  { [ -windows-name <TextNoCase> ]   Windows User Name
  | [ -unix-name <TextNoCase> ] }     UNIX User Name or User ID
  [ -trace-allow {yes|no} ]          Trace Allow Events (default: no)
  [ -enabled {enabled|disabled} ]    Filter Enabled (default: enabled)
  [ -time-enabled {1..720} ]         Minutes Filter is Enabled (default: 60)
```

If you desire, you can narrow the trace down to specific user names or IP addresses.

```
cluster::> vserver security trace filter modify -vserver DEMO -index 1 -protocols nfs -client-ip
x.x.x.x -trace-allow yes -enabled enabled
```

After the trace is created, the results are displayed in real time. When viewing the results, you can filter by successes, failures, user IDs, protocol, and more.

```
cluster::> vserver security trace trace-result show ?
  [ -instance | -fields <fieldname>, ... ]
  [[-node] <nodename>]                  Node
  [ -vserver <vserver name> ]            Vserver
  [[-seqnum] <integer>]                  Sequence Number
  [ -keytime <Date> ]                    Time
  [ -index <integer> ]                    Index of the Filter
  [ -client-ip <IP Address> ]             Client IP Address
  [ -path <TextNoCase> ]                  Path of the File Being Accessed
  [ -win-user <TextNoCase> ]              Windows User Name
  [ -security-style <security style> ]    Effective Security Style On File
  [ -result <TextNoCase> ]                Result of Security Checks
  [ -unix-user <TextNoCase> ]             UNIX User Name
  [ -session-id <integer> ]               CIFS Session ID
  [ -share-name <TextNoCase> ]            Accessed CIFS Share Name
  [ -protocol {cifs|nfs} ]                Protocol
  [ -volume-name <TextNoCase> ]           Accessed Volume Name
```

The following example demonstrates what a permission/access failure looks like for a specific user:

```
cluster::> vserver security trace trace-result show -node * -vserver DEMO -unix-user 1102 -result
*denied*
```

Vserver: DEMO

Node	Index	Filter Details	Reason
Node2	1	Security Style: UNIX and NFSv4 ACL	Access is denied. The requested permissions are not

granted by the ACE while
setting attributes. Access is
not granted for: "Write DAC"

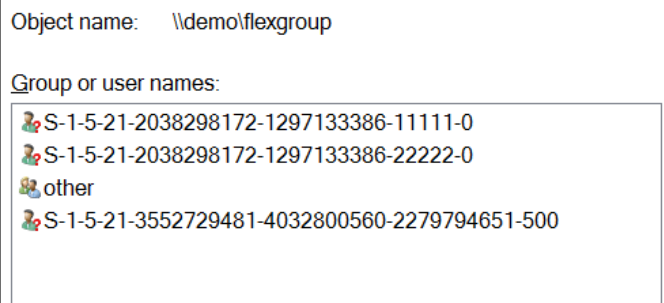
Protocol: nfs
Volume: home
Share: -
Path: /dir
Win-User: -
UNIX-User: 1102
Session-ID: -

Controlling the security tab view on UNIX security style objects

In ONTAP, you can configure the CIFS/SMB server to show or hide the Security tab when volumes or qtrees use UNIX security styles. The option to control this is `is-unix-nt-acl-enabled`.

This option is enabled by default, which means you see a Security tab when a file system object has UNIX security style. Users and groups in the tab show a manufactured SID specific to the SVM, which then resolves to UNIXPermUid, UNIXPermGid, and other names. You can modify permissions from this tab, but only for read, write, and execute values (`rxw`).

Figure 40) Permissions view before UNIX SIDs resolve.

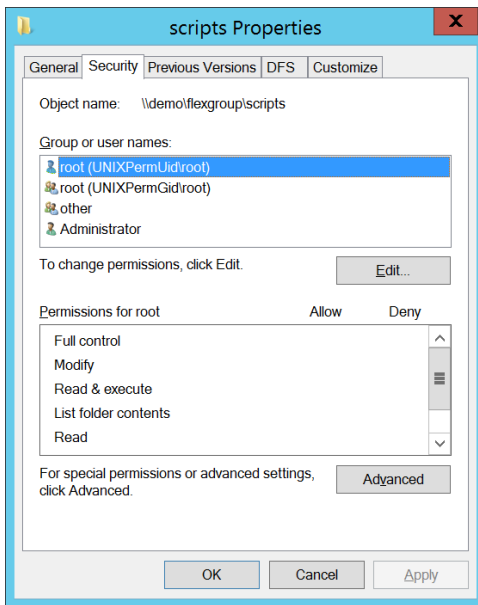


In Figure 41, the scripts folder is UNIX security style and the permissions are 777 (as seen from `vserver security file-directory` show CLI output):

```
cluster::*> vserver security file-directory show -vserver DEMO -path /flexgroup_16/scripts

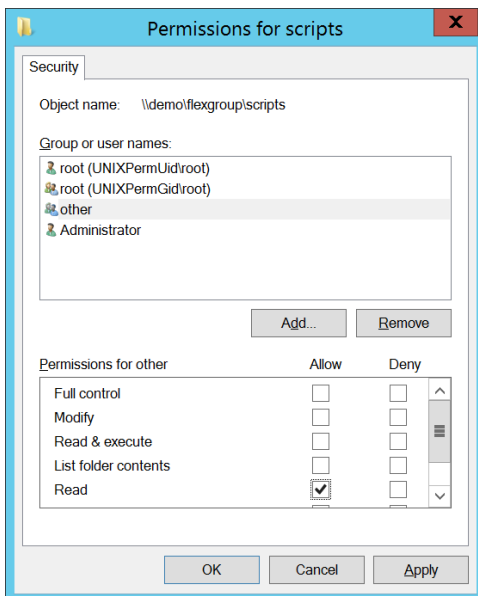
      Vserver: DEMO
      File Path: /flexgroup_16/scripts
      File Inode Number: 96
      Security Style: unix
      Effective Style: unix
      DOS Attributes: 10
      DOS Attributes in Text: ----D---
      Expanded Dos Attributes: -
      UNIX User Id: 0
      UNIX Group Id: 0
      UNIX Mode Bits: 777
      UNIX Mode Bits in Text: rwxrwxrwx
      ACLs: -
```


Figure 41) Security tab on UNIX security styles.



The folder is set to 777 permissions, but you can modify the Security tab to set Other to read.

Figure 42) Security tab on UNIX security styles – permission change.



After this configuration is complete, the permissions change to 774.

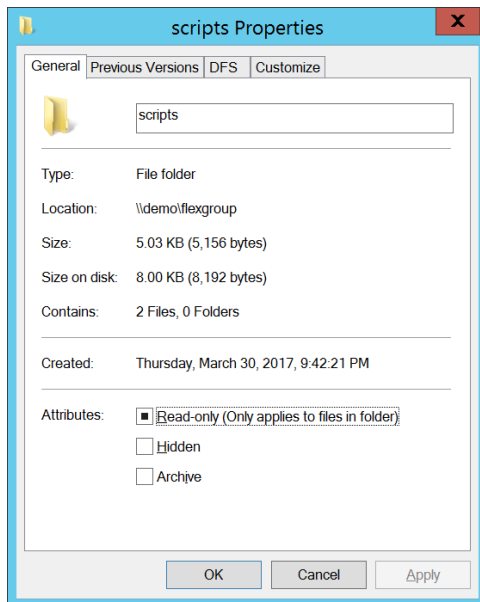
```
cluster::*> vserver security file-directory show -vserver DEMO -path /flexgroup_16/scripts

Vserver: DEMO
File Path: /flexgroup_16/scripts
File Inode Number: 96
Security Style: unix
Effective Style: unix
DOS Attributes: 10
DOS Attributes in Text: ----D---
```

```
Expanded Dos Attributes: -
    UNIX User Id: 0
    UNIX Group Id: 0
    UNIX Mode Bits: 774
UNIX Mode Bits in Text: rwxrwxr--
    ACLs: -
```

When the option is disabled, the security tab is hidden from CIFS/SMB clients on UNIX security style objects. One use case for disabling this option is to prevent unwanted permission changes on UNIX security style objects from SMB clients.

Figure 43) Security tab hidden on UNIX security styles.



Displaying NTFS permissions from NFS clients

When using NTFS security style volumes or qtrees, NFS clients display the mode bits or NFSv4 ACLs for the object as having wide open permissions (777) by default. This can be problematic for users and storage administrators for two primary reasons:

- Applications might depend on the ACLs or mode bits displaying properly for functionality.
- Users who see the mode bits as open might become alarmed, which can result in support tickets and cycles spent on troubleshooting.

Even though an ACL or mode bit shows 777 in NTFS security style volumes, it does not mean that the object allows everyone full access. In ONTAP, NTFS security style volumes control access based on NTFS security and ACLs. Therefore, an NFS client must have a valid UNIX user that maps to a valid Windows user in order to access the volume at all (authentication). After the initial authentication, the mapped user is then used to determine access based on the granular NTFS ACLs.

Data ONTAP 8.3.1 introduced an option called `ntacl-display-permissive-perms`. The default value for this option is Disabled. This default value allows the approximation of interpreted NTFS ACLs on NFS clients mounting NTFS objects, thereby displaying permissions based on minimum access, more closely approximating the real NTFS permissions of the current user in UNIX terms. This helps alleviate concerns and address application compatibility.

The option allows the user accessing the NTFS security-style volume to see the approximate permissions provided based on the user accessing the share. Therefore, users accessing the object might see differing results based on the NTFS security access.

Also, because of the vast difference between NTFS and UNIX-style ACLs, the approximation of permissions might not be exact. For example, if a user has a granular permission provided only in NTFS security semantics, then the NFS client cannot interpret that properly.

Effect of NFSv4 ACLs on Windows Security tab views

When NFSv4 ACLs are present on files, folders, or volumes, SMB 2.0 and later clients are unable to view or modify the Security tab, even if the `-is-unix-nt-acl-enabled` is set to `True`. This is because newer Windows client protocol versions do not support the same SMB calls used in SMB 1.0 to resolve NFSv4 ACLs. For more information, see [bug 928026](#).

Export policy rules: Access verification

ONTAP offers a command (`export-policy check-access`) that allows you to cross-check an export policy's access rule set against a client's access to help determine if an export policy rule is working properly for predeployment as well as for troubleshooting. Its functionality is similar to `exportfs -c` functionality. This command leverages all the normal name service communication and cache interaction that a standard mount from an NFS client would use.

Example of export-policy check-access

```
cluster1::*> vserver export-policy check-access -vserver vs1 -client-ip 1.2.3.4 -volume flex_vol  
-authentication-method sys -protocol nfs3 -access-type read
```

Path	Policy	Policy Owner	Policy Owner Type	Rule Index	Access
/	default	vs1_root	volume	1	read
/dir1	default	vs1_root	volume	1	read
/dir1/dir2	default	vs1_root	volume	1	read
/dir1/dir2/flex1	data	flex_vol	volume	10	read

Appendix A: Multiprotocol NAS terminology

Table 7 lists the terms mentioned throughout this document. Review this section to familiarize yourself with the terminology.

Table 7) Multiprotocol NAS terminology.

Term	Definition
Authentication	Verifying who you are. In ONTAP, this means taking a user name or numeric ID and mapping it to a valid Windows or UNIX user that the ONTAP cluster knows about.
Authorization	After authentication/name mapping, authorization determines what level of access a user or group has in the system. This includes things such as ACEs and ACLs, mode bits, share-level access, exports, and other forms of permissions.
Name mapping	Because Windows and UNIX permissions don't always map 1:1, user names are mapped to the appropriate volume security type to provide proper access regardless of the NAS protocol in use. For details about scenarios where names will map, see "Name mapping."
Logical interface (LIF)	Logical interfaces (or LIFs) are virtual IP addresses in ONTAP that provide data, management, and other network access to the storage system. For NAS protocols, data LIFs are required, as well as specific data LIF service policies .

Term	Definition
Storage virtual machine (SVM)	Storage virtual machines (SVMs) are the way in which storage administrators can provision unique, secure tenants for end users. Each SVM gets its own unique namespace, domain, and name service configurations, NAS protocol configurations, and so on. This provides flexibility when provisioning storage to multiple end users, such as with a service provider. Clusters can have one or up to 1,024 SVMs.
Export policy	Export policies are containers for multiple rules used to determine access for clients attempting to mount NFS (and even SMB, if you choose) shares. Each volume and/or qtree can be assigned a unique export policy.
Export policy rule	Export policies are containers, however, the access level is determined by the export policy rules in ONTAP. A policy can have hundreds of rules, each with multiple clientmatch values per rule. For more information, see “Export policy and rule concepts.”
FlexVol volume	A FlexVol volume is a logical boundary that lives on top of physical storage and provides a mount point or share path for NAS clients. FlexVol volumes each have their own unique file system IDs and can be junctioned to one another in a cluster namespace. FlexVol volumes live on individual nodes in the cluster. FlexVol volumes can grow up to 100TB and can be grown/shrunk as many times as needed.
FlexGroup volume	A FlexGroup volume is a group of FlexVol volumes presented to NAS clients as a large single namespace. FlexGroup volumes can grow much larger than FlexVol volumes (20PB, 200 billion files) and can span multiple nodes in the cluster, which provides performance benefits in workloads that require parallelism. For more information, see TR-4571 .
Namespace	A namespace in ONTAP is the access point for NAS shares. Multiple FlexVol volumes in an ONTAP SVM can be considered multiple namespaces, or they can be considered a single namespace if you use them to construct a directory tree. ONTAP features such as FlexGroup and FlexCache volumes are intended to enhance the concept of a single namespace. For more information, see “Namespace and file system concepts.”
FlexCache volume	FlexCache in ONTAP provides a writable, persistent virtual cache of a volume in a remote place. Caches are beneficial in read-intensive environments where data is accessed more than once and is shared by multiple hosts.
CIFS share	A CIFS (or SMB) share is an access point created for clients accessing NAS via the CIFS/SMB protocol. Generally, this is from Windows clients, but can also be from Linux or MacOS clients.
NFS export	An NFS export is an access point created for clients accessing NAS data through the NFS protocol. Generally, this is done from Linux clients, but can also be done from Windows or MacOS.

Appendix B: NFS server options

NFS servers in ONTAP provide many different options for configurations in ONTAP. Most of these options won't apply to every environment, but some might help you resolve issues unique to your multiprotocol deployment.

Table 8 lists the NFS options available in ONTAP 9.8 and what they are used for. These options are controlled with the `nfs modify` command. Options denoted by asterisks are located in Advanced Privilege.

Table 8) NFS server options that can impact multiprotocol NAS – ONTAP 9.8 and later.

Option	How it can impact multiprotocol NAS
<code>-v4.0/v4.1</code>	Enabling NFS version 4.x requires UNIX users to map to name strings. If not configured properly, client behavior can be unpredictable. If NFSv4.x is not desired on clients, they can still negotiate to NFSv4.x, if not specified otherwise. This can create issues in multiprotocol NAS – specifically regarding name mappings between Windows and UNIX.
<code>-default-win-user</code>	This option is not set by default. When set, all UNIX users that attempt to access a volume or qtree with NTFS permissions fall back to a single default Windows user if no existing Windows name mapping rule or 1:1 user mapping exists. For example, if UNIX user <code>jacksprat</code> attempts to access a volume with NTFS security style, ONTAP attempts to find a Windows user named <code>jacksprat</code> or an existing name mapping rule in LDAP or local files. If none exist, then <code>jacksprat</code> maps to the default Windows user. If the default Windows user is not set, then authentication into ONTAP from NFS to NTFS security style volumes fails because ONTAP views that user as invalid due to not being able to discern permissions.
<code>-ntfs-unix-security-ops*</code>	<p>This option controls the behavior of NFS operations when performed on NTFS security style volumes or qtrees. NFS operations (such as <code>chmod</code>, <code>chown</code>, and so on) cannot be performed on NTFS security style volumes. The default setting on export policy rules (fail) is to send an error message to the NFS client when an operation is attempted. Alternatively, the value can be set to ignore, which allows operations to fail silently.</p> <p>By default, the NFS server value is set to <code>use_export_policy</code>, which means the export policy rule dictates how NFS operations on NTFS security style objects report back to clients. Setting the value explicitly on the NFS server makes all NFS clients act the same. If you want more granular control over this behavior, leave the option as the default and control how it behaves from the individual export policy rules.</p>
<code>-v4-id-domain</code>	When NFSv4.x is enabled, the <code>-v4-id-domain</code> option determines how the user string in ONTAP is formulated. This string should match the string on the NFS clients to ensure proper name string/domain mappings. For example, if the <code>-v4-id-domain</code> option is left as the default <code>v4iddomain.com</code> string and a client tries to find an NFSv4.x user named <code>jacksprat@domain.com</code> , ONTAP is not able to match that string, as <code>jacksprat</code> belongs to

Option	How it can impact multiprotocol NAS
	<p>the default tv4iddomain.com domain. Because jacksprat@domain.com is not identical to jacksprat@defaulttv4iddomain.com, the user is squashed to nobody, which essentially breaks the Windows-UNIX name mappings in multiprotocol NAS environments because nobody is not a valid UNIX user and is certainly not the intended UNIX user.</p>
-v4-acl-preserve	<p>When enabled, this option preserves NFSv4.x ACLs if an NFSv3 chmod or chown operation is attempted. If disabled, running chown/chmod wipes the NFSv4.x ACEs. This option only applies to UNIX or mixed security styles because NFSv4 ACLs cannot be applied to NTFS security styles.</p>
-v4.0-acl/-v4.1-acl	<p>Enabling NFSv4.x ACLs in multiprotocol environments only affects volumes with UNIX or mixed security styles. NTFS security styles only understand NTFS ACLs. You must enable the ACL for the NFS version you are using. For example, if you are using NFSv4.1, enable -v4.1-acl. You don't need to enable both if you only use one of the two NFS versions.</p> <p>Note: ONTAP 9.8 and later supports NFSv4.2, but there is no NFS option to enable it; it is enabled when you enable NFSv4.1.</p>
-v4-numeric-ids	<p>The -v4-numeric-ids option determines whether NFSv4.x users can leverage numeric identifiers when name string matches are not available. In other words, NFSv4.x operates more like NFSv3 for user name resolution, where the domain ID string mapping is not required. By default, this value is set to Enable.</p> <p>In multiprotocol NAS environments, if a user name from NFS arrives as a numeric ID and cannot resolve to a proper user name (through local files/passwd or name services), UNIX security style objects operate as normal. But NTFS security styles require a valid user name to map to a valid Windows user so that permissions can be accurately accounted for.</p> <p>If a user name comes in as numeric ID 1234, and ONTAP cannot find a valid UNIX user name for that numeric ID, then it attempts to map to a Windows user of DOMAIN\1234. Generally, Windows users such as these do not exist; therefore, mapping/authentication for NTFS security styles will fail. This underscores the importance of having a means for UNIX user names to resolve properly in multiprotocol NAS environments, such as LDAP. For more information, see TR-4067.</p>
-auth-sys-extended-groups*	<p>This option controls whether extended groups are enabled or disabled. By default, NFS operations only support up to 16 GIDs per user for auth_sys and 32 GIDs for auth_gss. That means if a user belongs to more groups than NFS can support, the additional groups are dropped from the NFS RPC packet, which causes permissions/access inconsistencies.</p>

Option	How it can impact multiprotocol NAS
	Extended groups can provide support for up to 1,024 groups per user by prefetching the group memberships for a user from a name service and reverse querying group memberships. In multiprotocol NAS environments, enable this option when using name services to ensure all Windows users/groups are properly recognized. For more information, see TR-4067 and TR-4835 .
<code>-extended-groups-limit*</code>	This option determines the maximum number of groups for extended groups—this value can be between 32 and 1,024. Performance impact for this option is generally minimal, provided there is a good network connection to name service servers and an adequate number of LDAP servers to load balance requests.
<code>-map-unknown-uid-to-default-windows-user*</code>	In scenarios where you have NTFS security style volumes, and the numeric NFS ID for the user can't map to a valid Windows user name, this option controls whether or not the unknown UID maps to the default Windows user defined with the <code>-default-win-user</code> option. This means <code>*all*</code> incoming unknown UIDs map to the specified Windows user – even users that intend to act as other users. The default value for this option is Enable, but the <code>-default-win-user</code> value is not set, therefore, the default behavior is for all incoming unknown users to map to no Windows user. As a result, unknown UIDs attempting to gain access to NTFS security styles fail to authenticate. It's generally not recommended to set a default Windows user, but there might be some use instances where you need to allow an application to function properly. In those cases, you might want to dedicate an isolated SVM for those applications, because this is a global option.
<code>-ntacl-display-permissive-perms*</code>	This option controls how end users on NFS clients see NTFS style permissions when issuing commands such as <code>ls -la</code> . Because NFS doesn't understand NTFS security semantics, clients see permissions shown as 777 by default. This creates unnecessary alarm for users (because NTFS ACLs/name mappings control permissions). It can also break some application workflows that are dependent on permissions being seen in a specific way. When <code>-ntacl-display-permissive-perms</code> is set to Enabled, ONTAP sends an approximation of permissions for a user accessing a file to more accurately portray what the user accessing the share can and cannot do.
<code>-v3-ms-dos-client</code>	This option enables whether Windows NFS can be used in an SVM. For details about this option, see TR-4067 . The impact to multiprotocol NAS here depends on the Windows NFS configuration (such as how names/groups are presented to the server), but the same general rules apply to Windows NFS as to regular NFS clients.
<code>-ignore-nt-acl-for-root*</code>	This option controls how the root user in NFS behaves on NTFS security style volumes. By default, this option is set to Disabled, which means the root user must map to a valid Windows user like any other NFS user to help negotiate NTFS permissions. When enabled, the root user ignores all NTFS ACLs and acts more like the UNIX-style root user

Option	How it can impact multiprotocol NAS
	with full read/write access on objects, regardless of the NTFS permissions. Use this option with caution.
<code>-cached-cred-positive-ttl*</code>	This option controls the timeout period for cached credentials in NAS environments. When a user's credentials are successfully queried, ONTAP caches them to reduce the number of times name services need to be contacted. The default timeout value is 86,400,000msecs, which translates to 24 hours. This can impact multiprotocol environments if a user is added or removed from a group, because the access will not be updated until the cache ages out in 24 hours or is manually flushed. For more information about how name services and caches operate in ONTAP, see TR-4668 . For more information about how these options pertain to LDAP, see TR-4835 .
<code>-cached-cred-negative-ttl*</code>	This option controls the timeout period for credentials that have been verified as having negative access. When a user is denied access to a file or folder, the cache is populated for a default of 7,200,000msecs (two hours). This can impact multiprotocol environments if a user is added or removed from a group, because the access will not be updated until the cache ages out in 24 hours or is manually flushed. For more information about how name services and caches operate in ONTAP, see TR-4668 . For more information about how these options pertain to LDAP, see TR-4835 .
<code>-skip-root-owner-write-perm-check*</code>	This option specifies whether permission checks should be skipped for NFS WRITE calls from root/owner. For copying read-only files to a destination folder that has inheritable ACLs, this option must be enabled. Note: When enabled, if an NFS client does not make use of an NFS ACCESS call to check for user-level permissions and then tries to write onto read-only files, the operation will succeed. The default setting is Disabled.
<code>-v4-inherited-acl-preserve*</code>	When using NFSv4 ACLs, this option determines whether or not parent directory mode bits override NFSv4 ACL inheritance. By default, this option is disabled and the behavior is intended for the created file to use the client mode bits instead of the mode bits of inherited parent ACL. This is the expected behavior as per RFC 5661 . However, if you want ACL inheritance instead, enable this option.
<code>-cached-cred-harvest-timeout*</code>	This option controls how long entries in the cache that are not actively being used remain in the cache. For example, if user1 gets credentials cached but never returns to the system to use those credentials, ONTAP removes that entry after the harvest timeout value expires, as not to use unnecessary memory for the cache on stale entries. The default timeout value is 86,400,000 (or 24 hours). TR-4668 covers how name services and caches operate in ONTAP. TR-4835 also covers these options as they pertain to LDAP.

Appendix C: CIFS/SMB server options

CIFS/SMB servers also have a series of configurable options that are useful in multiprotocol NAS configurations. Table 9 lists the CIFS/SMB options available in ONTAP 9.8 and what they are used for. These options are controlled with the `cifs options modify` command. Options denoted by asterisks are located in Advanced Privilege.

Table 9) CIFS server options that can impact multiprotocol NAS – ONTAP 9.8 and later.

Option	How it can impact multiprotocol NAS
<code>-default-unix-user</code>	<p>This options controls which UNIX user is used to map Windows users with no valid UNIX name mapping rules. By default, this user is set to <code>pcuser</code>, which corresponds to numeric ID 65534.</p> <p>If a Windows/SMB client creates a file on a share and the user creating the file maps to the default UNIX user, the file is assigned the owner <code>pcuser</code>. For more information, see the section on “Authentication and name mapping.”</p> <ul style="list-style-type: none">• On NFS clients, 65534 generally maps to the <code>nfsnobody</code> user. As a result, listing files/owners from NFS with <code>pcuser/65534</code> as the owner instead displays <code>nfsnobody</code>. If you see this behavior on your NFS mounts, that likely means the Windows user creating files is not mapping to the expected user name and is falling back to the default UNIX user.• On NTFS security style volumes, the Windows permissions still apply, regardless of the UNIX owner. On UNIX security styles, then having the file owner as <code>pcuser</code> can be problematic.
<code>-read-grants-exec</code>	<p>In UNIX security styles, mode bits and NFSv4 ACLs are used to grant/deny access to files or folders. When a file is set to read-only, the execute bit (x) is not set. This setting can create problems when executing files from CIFS/SMB clients because it requires the execute to be set to work properly by default. In some instances, setting execute on these files is not possible, therefore, ONTAP provides the <code>read-grants-exec</code> CIFS option to bypass this restriction.</p>
<code>-is-local-auth-enabled</code>	<p>Local authentication for CIFS/SMB servers is use of WORKGROUP mode for ONTAP, which is a way to allow CIFS/SMB access without needing a domain controller/Active Directory implementation.</p> <p>In multiprotocol NAS, using WORKGROUP mode still means that the Windows users should map to UNIX users with the same names, or use name mapping rules to map them to different UNIX users.</p> <p>This is option enabled by default.</p>
<code>-is-local-users-and-groups-enabled</code>	<p>Local authentication for CIFS/SMB servers is use of WORKGROUP mode for ONTAP, which is a way to allow CIFS/SMB access without needing a domain controller/Active Directory implementation.</p> <p>In multiprotocol NAS, using WORKGROUP mode still means that the Windows users should map to UNIX users with the same names, or use name mapping rules to map them to different UNIX users.</p>

Option	How it can impact multiprotocol NAS
	This is enabled by default.
<code>-is-exportpolicy-enabled</code>	<p>This option enables the ability to use export policies and rules for CIFS/SMB shares. The advantage to using export policies and rules is providing a way to control access through subnet or host names/IP addresses. This option is disabled by default.</p> <p>An alternative method of limiting access to CIFS/SMB shares is a combination of share-level permissions and mapping SMB clients to Windows user names through the <code>name-mapping create</code> command. For more information, see “Mapping Windows clients to user names.”</p>
<code>-is-unix-nt-acl-enabled</code>	This option controls whether you can view the UNIX permissions on UNIX security style volumes using CIFS/SMB clients and the Security tab. For more information, see “Displaying NTFS permissions from NFS clients.”
<code>-is-trusted-domain-enum-search-enabled</code>	When your CIFS server resides in a domain that has a bidirectional trust, and you want UNIX users to map to Windows users in both domains, enable this option. By default, this option is disabled.
<code>-is-read-only-delete-enabled</code>	This optional parameter controls the deletion of read-only files and directories. NTFS delete semantics forbid the deletion of a file or directory when the read-only attribute is set. UNIX delete semantics ignore it, focusing instead on parent directory permissions, which some applications require. This option is used to select the desired behavior. By default, this option is disabled, yielding NTFS behavior.
<code>-is-unix-extensions-enabled</code>	This option allows for UNIX-based SMB clients (such as MacOS or Linux Samba) to transmit POSIX/UNIX security information over SMB to the UNIX-based SMB client for translation to display the proper POIX/UNIX security. This is not the same as support for POSIX ACLs or extended attributes (xattr), which ONTAP does not support.
<code>-is-search-short-names-enabled</code>	This option controls how ONTAP handles CIFS/SMB 8.3 short names. For more information, see What are the file naming conventions for Network File System(NFS) and SMB/CIFS and maximum length of file names?
<code>-guest-unix-user</code>	This optional parameter specifies that an unauthenticated user coming from any untrusted domain can be mapped to a specified UNIX user for the CIFS server. If the CIFS server cannot authenticate the user against a domain controller for the home domain or a trusted domain or the local database, and this option is enabled, the CIFS server considers the user as a guest user and maps the user to the specified UNIX user. The UNIX user must be a valid user.
<code>-is-admin-users-mapped-to-root-enabled</code>	This option controls whether users that are added to the BUILTINAdministrators group on the SVM are mapped to the root user, which maps those user names to root, writes files as root, has root's permissions, and so on. For details about this option, see “Mapping Windows admin users to root.”

Option	How it can impact multiprotocol NAS
<code>-is-use-junctions-as-reparse-points-enabled</code>	<p>This option is enabled by default and controls how Windows/SMB clients see volumes that are mounted as junction paths in ONTAP.</p> <p>When enabled:</p> <ul style="list-style-type: none"> Junction paths show as <JUNCTION> when using the <code>dir</code> commands in the cmd. Junction paths show as shortcut folders in Windows Explorer. <p>When disabled:</p> <ul style="list-style-type: none"> Junction paths show as regular directories to SMB clients in Windows Explorer and cmd. <p>For more information, see “Junction paths and reparse points.”</p>
<code>-grant-unix-group-perms-to-others</code>	<p>This optional parameter specifies whether the incoming CIFS user, who is not the owner of the file, can be granted the group permission. If the CIFS incoming user is not the owner of UNIX security-style file, and this option is set to True, then at all times the file's group permissions are granted. If the CIFS incoming user is not the owner of UNIX security-style file, and this option is set to False, then the normal UNIX rules are applicable to grant the permissions. The default value of this parameter is False.</p>
<code>-widelink-as-reparse-point-versions</code>	<p>This option controls the SMB versions where widelinks created in the system appear as reparse points. By default, this is set to SMB1, but you can enable this behavior for SMB2 and SMB3. For more information, see “Junction paths and reparse points.”</p>

Where to find additional information

To learn more about the information that is described in this document, review the following documents and/or websites:

- TR-4067: NetApp ONTAP NFS Best Practices and Implementation Guide
<https://www.netapp.com/us/media/tr-4067.pdf>
- TR-4569: Security Hardening Guide for ONTAP 9
<https://www.netapp.com/pdf.html?item=/media/10674-tr4569pdf.pdf>
- TR-4571: NetApp FlexGroup Volumes Best Practices
<https://www.netapp.com/us/media/tr-4571.pdf>
- TR-4616: NFS Kerberos in NetApp ONTAP
<https://www.netapp.com/us/media/tr-4616.pdf>
- TR-4668: Name Services Best Practices - NetApp ONTAP
<https://www.netapp.com/us/media/tr-4668.pdf>
- TR-4743: FlexCache in ONTAP
<https://www.netapp.com/pdf.html?item=/media/7336-tr4743pdf.pdf>
- TR-4835: How to Configure LDAP in NetApp ONTAP
<https://www.netapp.com/media/19423-tr-4835.pdf>

Version history

Version	Date	Document version history
Version 1.0	April 2021	Initial release.

Refer to the [Interoperability Matrix Tool \(IMT\)](#) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

Copyright Information

Copyright © 2021 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

Data contained herein pertains to a commercial item (as defined in FAR 2.101) and is proprietary to NetApp, Inc. The U.S. Government has a non-exclusive, non-transferrable, non-sublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b).

Trademark Information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.

TR-4887-0421