Technical Report

Clustered Data ONTAP
NFS Best Practice and Implementation Guide

Justin Parisi, Bikash Roy Choudhury, NetApp
February 2014 | TR-4067

Executive Summary

This report serves as an NFSv3 and NFSv4 operational guide and an overview of the clustered NetApp® Data ONTAP® 8.2 operating system with a focus on NFSv4. It details steps in the configuration of an NFS server, NFSv4 features, and the differences between clustered Data ONTAP and Data ONTAP operating in 7-Mode.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 Introduction

As more and more data centers evolve from application-based silos to server virtualization and scale-out systems, storage systems have evolved to support this change. Clustered NetApp Data ONTAP 8.2 provides shared storage for enterprise and scale-out storage for various applications such as databases, server virtualization, and home directories. It provides a solution for emerging workload challenges in which data is growing in size and becoming more complex and unpredictable. Clustered Data ONTAP 8.2 is unified storage software that scales out to provide efficient performance and support of multi-tenancy and data mobility. This scale-out architecture provides large scalable containers to store petabytes of data. It also upgrades, rebalances, replaces, and redistributes load without disruption, which means that the data is perpetually alive and active.

1.1 Scope

This document covers the following topics:

- Introduction to clustered Data ONTAP
- Architecture of clustered Data ONTAP
- Setting up an NFS server in clustered Data ONTAP
- Configuring export policies and rules
- 7-Mode and clustered Data ONTAP differences and similarities for NFS access-cache implementation
- Multiprotocol user mapping
- Mapping of NFS options in 7-Mode to clustered Data ONTAP
- Configuration of NFS v4 features in clustered Data ONTAP, such as user ID mapping, delegations, ACLs, and referrals

Note: This document is not intended to provide information on migration from 7-Mode to clustered Data ONTAP; it is specifically about NFSv3 and NFSv4 implementation in clustered Data ONTAP and the steps required to configure it.

1.2 Intended Audience and Assumptions

This technical report is for storage administrators, system administrators, and data center managers. It assumes basic familiarity with the following:

- NetApp FAS systems and the Data ONTAP operating system
- Network file sharing protocols (NFS in particular)

Note: This document contains advanced and diag-level commands. Exercise caution when using these commands. If there are questions or concerns using these commands, contact NetApp Support for assistance.

2 Overview of Clustered Data ONTAP

2.1 Business Challenges with Traditional Storage

- **Capacity scaling**
  Capacity expansion in traditional storage systems might require downtime, either during physical installation or when redistributing existing data across the newly installed capacity.
- **Performance scaling**
  Standalone storage systems might lack the I/O throughput to meet the needs of large-scale enterprise applications.

- **Availability**
  Traditional storage systems often have single points of failure that can affect data availability.
- **Right-sized SLAs**
  Not all enterprise data requires the same level of service (performance, resiliency, and so on). Traditional storage systems support a single class of service, which often results in poor utilization or unnecessary expense.
- **Cost**
  With rapid data growth, storage is consuming a larger and larger portion of shrinking IT budgets.
- **Complicated management**
  Discrete storage systems and their subsystems must be managed independently. Existing resource virtualization does not extend far enough in scope.

2.2 Clustered Data ONTAP 8.2

Clustered NetApp Data ONTAP 8.2 helps to achieve results and get products to market faster by providing the throughput and scalability needed to meet the demanding requirements of high-performance computing and digital media content applications. It also facilitates high levels of performance, manageability, and reliability for large Linux®, UNIX®, or Microsoft® Windows® clusters.

Features of clustered Data ONTAP include:

- Scale-up, scale-out, and scale-down are possible with numerous nodes using a global namespace.
- Storage virtualization with Storage Virtual Machines (SVMs) eliminates physical boundaries of a single controller (memory, CPU, ports, disks, and so on).
- Nondisruptive operations (NDO) are available when you redistribute load or rebalance capacity combined with network load balancing options within the cluster for upgrading or expanding its nodes.
- NetApp storage efficiency features like Snapshot™ copies, thin provisioning, space-efficient cloning, deduplication, data compression, and RAID-DP® technology are also available.

Solutions for the previously mentioned business challenges can be addressed by using the scale-out clustered Data ONTAP approach.

- **Scalable Capacity**
  Grow capacity incrementally, on demand, through the nondisruptive addition of storage shelves and growth of storage containers (pools, LUNs, file systems). Support nondisruptive redistribution of existing data to the newly provisioned capacity as needed via volume moves.
- **Scalable Performance—Pay as You Grow**
  Grow performance incrementally, on demand and nondisruptively, through the addition of storage controllers in small, economical (pay-as-you-grow) units.
- **High Availability**
  Leverage highly available pairs to provide continuous data availability in the face of individual component faults.
- **Flexible, Manageable Performance**
  Support different levels of service and provide the ability to dynamically modify the service characteristics associated with stored data by nondisruptively migrating data to slower, less costly disks and/or by applying quality-of-service (QoS) criteria.
- **Scalable Storage Efficiency**
  Control costs through the use of scale-out architectures that employ commodity components. Grow capacity and performance on an as-needed (pay-as-you-go) basis. Increase utilization through thin provisioning and data deduplication.
- **Unified Management**
  Provide a single point of management across the cluster. Leverage policy-based management to streamline configuration, provisioning, replication, and backup. Provide a flexible monitoring and reporting structure implementing an exception-based management model. Virtualize resources

across numerous controllers so that volumes become simple-to-manage logical entities that span storage controllers for performance and dynamic redistribution of data.

3 Architecture

3.1 Important Components of Clustered Data ONTAP

Storage Virtual Machine (SVM)

- An SVM is a logical file system namespace capable of spanning beyond the boundaries of physical nodes in a cluster.
  - Clients can access virtual servers from any node in the cluster, but only through the associated logical interfaces (LIFs).
  - Each SVM has a root volume under which additional volumes are mounted, extending the namespace.
  - It can span several physical nodes.
  - It is associated with one or more logical interfaces; clients access the data on the virtual server through the logical interfaces that can live on any node in the cluster.

Logical Interface (LIF)

- A logical interface is essentially an IP address with associated characteristics, such as a home port, a list of ports for failover, a firewall policy, a routing group, and so on.
  - Client network data access is through logical interfaces dedicated to the SVM.
  - An SVM can have more than one LIF. You can have many clients mounting one LIF or one client mounting several LIFs.
  - This means that IP addresses are no longer tied to a single physical interface.

Aggregates

- An aggregate is a RAID-level collection of disks; it can contain more than one RAID group.
  - Aggregates serve as resources for SVMs and are shared by all SVMs.

Flexible Volumes

- A volume is a logical unit of storage. The disk space that a volume occupies is provided by an aggregate.
  - Each volume is associated with one individual aggregate and therefore with one physical node.
  - In clustered Data ONTAP, data volumes are owned by an SVM.
  - Volumes can be moved from aggregate to aggregate with the DataMotion™ for Volumes feature, without loss of access to the client. This provides more flexibility to move volumes within a single namespace to address issues such as capacity management and load balancing.

3.2 Cluster Namespace

A cluster namespace is a collection of file systems hosted from different nodes in the cluster. Each SVM has a file namespace that consists of a single root volume. The SVM namespace consists 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.

All the volumes belonging to the SVM are linked into the global namespace in that cluster. 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.

Figure 1) Cluster namespace.

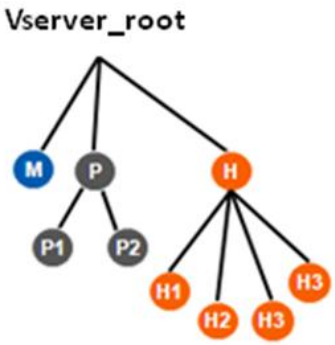
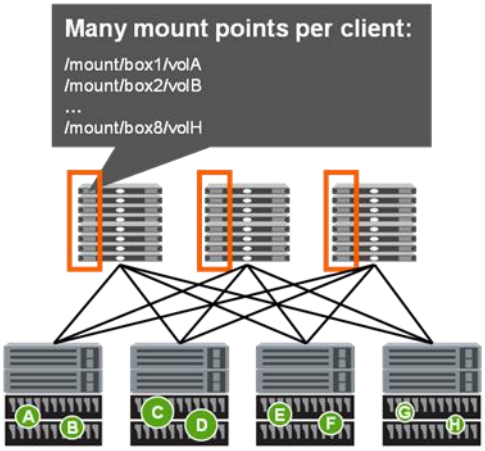
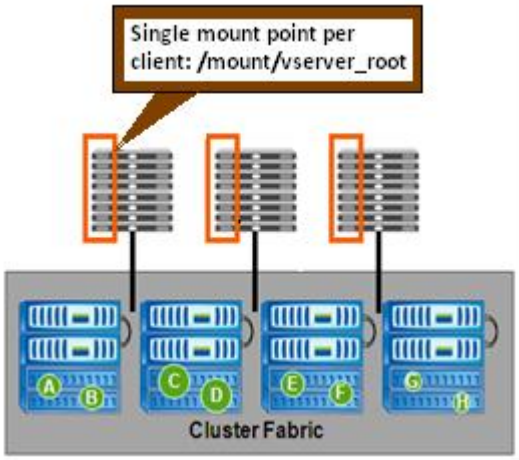


Table 1) Benefits of a cluster namespace.

Without a Cluster Namespace	With a Cluster Namespace
	
• Change mapping for thousands of clients when moving or adding data • Difficult to manage • Very complex to change • Doesn't scale	• Namespace unchanged as data moves • Much easier to manage • Much easier to change • Seamlessly scales to petabytes

3.3 Steps to Bring Up a Clustered Data ONTAP NFS Server

NetApp assumes that the following configuration steps have been completed before you proceed with setting up a clustered Data ONTAP NFS server.

- Clustered Data ONTAP 8.2 installation and configuration
- Aggregate creation
- SVM creation
- LIF creation

- Volume creation
- Valid NFS license applied

Note: NFS server creation and options are explained in detail in the “File Access and Protocols Management Guide” for the version of clustered Data ONTAP being used.

Export Policies in Clustered Data ONTAP

Instead of the flat export files found in 7-Mode, clustered Data ONTAP offers export policies as containers for export policy rules to control security. These policies are stored in the replicated database, thus making exports available across every node in the cluster, rather than isolated to a single node. A NetApp cluster can support 70k export policy rules per cluster for systems using less than 16GB of RAM and 140k export policy rules on systems using more than 16GB of RAM. Each HA pair can handle up to 10,240 export policy rules. There is no limit on export policies. Volumes that are created without specifying the policy will get assigned the default policy.

A newly created SVM contains an export policy called “default.” This export policy cannot be deleted, although it can be renamed or modified. Each volume created in the SVM inherits the “default” export policy and the rules assigned to it. Because export policy rules are inherited by default, NetApp recommends opening all access to the root volume of the SVM (vsroot) when a rule is assigned. Setting any rules for the “default” export policy that restrict the vsroot denies access to the volumes created under that SVM because vsroot is “/” in the path to “/junction” and factors into the ability to mount and traverse. 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 (0755 permissions). In clustered Data ONTAP 8.2, 0755 is the default security set on volumes. The default owner is UID 0 and the default group is GID 1. To control data volume access, separate export policies and rules can be set for every volume under the vsroot.

Each volume has only one export policy, although numerous volumes can use the same export policy. An export policy can contain several rules to allow granularity in access control. With this flexibility, a user can choose to balance workload across numerous volumes, yet can assign the same export policy to all volumes. **Remember, export policies are containers for export policy rules.** If a policy is created with no rule, that effectively denies access to everyone. Always create a rule with a policy to allow access to a volume.

Export policy and export policy rule creation (including examples) is specified in detail in the “File Access and Protocols Management Guide” for the version of clustered Data ONTAP being used.

- Use the `vserver export-policy` commands to set up export rules; this is equivalent to the `/etc/exports` file in 7-Mode.
- All exports are persistent across system restarts, and this is why temporary exports cannot be defined.
- There is a global namespace per virtual server; this maps to the *actual=path* syntax in 7-Mode. In clustered Data ONTAP, a volume can have a designated junction path that is different from the volume name. Therefore, the *–actual* parameter found in the `/etc/exports` file is no longer applicable. This applies to both NFSv3 and NFSv4.
- In clustered Data ONTAP, an export rule has the granularity to provide different levels of access to a volume for a specific client or clients, which has the same effect as fencing in the case of 7-Mode.
- Export policy rules affect CIFS access in clustered Data ONTAP by default versions prior to 8.2. For more information on how export policies can be applied to volumes hosting CIFS shares, see the “File Access and Protocols Management Guide” for the version of clustered Data ONTAP being used.

Refer to Table 16 in the appendix for NFSv3 config options that are modified in clustered Data ONTAP.

3.4 Translation of NFS Export Policy Rules from 7-Mode to Clustered Data ONTAP

Export Policy Sharing and Rule Indexing

Clustered Data ONTAP exports do not follow the 7-Mode model of file-based access definition, in which the file system path ID is described first and then the clients who want to access the file system path are specified. Clustered Data ONTAP export policies are sets of rules that describe access to a volume. Exports are applied at the volume level, rather than to explicit paths as in 7-Mode.

Policies can be associated with one or more volumes.

For example, in 7-Mode exports could look like this:

```
/vol/test_vol        -sec=sys,rw=172.17.44.42,root=172.17.44.42
/vol/datastore1_sata    -sec=sys,rw,nosuid
```

In clustered Data ONTAP, export rules would look like this:

```
                           Policy        Rule     Access       Client                 RO
Vserver             Name        Index   Protocol   Match               Rule
------------ --------------- ------  -------- --------------------- ------------------
vs1_nfs3     nfs3_policy1    1           any        0.0.0.0/0             any
vs2_nfs4     nfs4_policy1    1           any        0.0.0.0/0             any
```

7-Mode supports subvolume or nested exports; Data ONTAP supports exporting `/vol/volX` and `/vol/volX/dir`. Clustered Data ONTAP currently does not support subvolume or nested exports. The concept of subvolume exports does not exist because the export path applicable for a particular client's access is specified at mount time based on the mount path.

Clustered Data ONTAP did not support qtree exports prior to 8.2.1. In previous releases, a qtree could not be a junction in the namespace independent of its containing volume because the "export permissions" were not specified separately for each qtree. The export policy and rules of the qtree's parent volume were used for all the qtrees contained within it. This is different from the 7-Mode qtree implementation, in which each qtree is a point in the namespace where export policies can be specified.

In 8.2.1, qtree exports will be available for NFSv3 exports only. The export policy can be specified at the qtree level or inherited from the parent volume. Qtree export policies and rules work exactly how volume export policies and rules work.

UNIX Users and Groups

The UID and GID that a cluster will leverage depends on how the SVM has been configured with regard to name mapping and name switch. The name service switch (ns-switch) option for SVMs specifies the source or sources that are searched for network information and the order in which they are searched. Possible values include nis, file, and ldap. This parameter provides the functionality of the /etc/nsswitch.conf file on UNIX systems.

The name mapping switch (nm-switch) option for SVMs specifies the sources that are searched for name mapping information and the order in which they are searched. Possible values include file and ldap.

If NIS or LDAP are specified for name services and/or name mapping, then the cluster will contact the specified servers for UID and GID information. Connectivity to NIS and LDAP will attempt to use a data LIF in the SVM by default. Therefore, data LIFs must be routable to name service servers. Management LIFs will be used in the event a data LIF is not available to service a request. If data LIFs are not able to communicate with name service servers, then there might be some latency in authentication requests that will manifest as latency in data access.

If desired, name service and name mapping communication can be forced over the management network by default. This can be useful in environments in which an SVM does not have access to name service and name mapping servers.

To force all authentication requests over the management network:

```
cluster::> set diag
cluster::> vserver modify -vserver vs0 -protocol-services-use-data-lifs false
```

NetApp recommends leaving this option as "true" since management networks are often more bandwidth-limited than data networks (1Gb versus 10Gb), which can result in authentication latency in some cases.

If local files are used, then the cluster will leverage the unix-user and unix-group tables created for the specified SVM. Because no remote servers are being used, there will be little to no authentication latency. However, in large environments, managing large lists of unix-users and groups can be daunting and mistake prone.

NetApp recommends leveraging either NIS or LDAP for name services in larger environments.

Unix-users and groups are not created by default when creating an SVM using the `vserver create` command. However, using System Manager or the `vserver setup` command will create the default users of root (0), pcuser (65534), and nobody (65535) and default groups of daemon (1), root (0), pcuser (65534), and nobody (65535).

```
cluster::> unix-user show -vserver vs0
  (vserver services unix-user show)
               User            User   Group  Full
Vserver        Name            ID     ID     Name
-------------- --------------- ------ ------ --------------------------------
vs0            nobody          65535  65535  -
vs0            pcuser          65534  65534  -
vs0            root            0      1      -
3 entries were displayed.

cluster::> unix-group show -vserver vs0
  (vserver services unix-group show)
Vserver        Name                ID
-------------- ------------------- ----------
nfs            daemon              1
nfs            nobody              65535
nfs            pcuser              65534
nfs            root                0
4 entries were displayed.
```

NetApp recommends using System Manager or `vserver setup` to avoid configuration mistakes when creating new SVMs.

The Anon User

The "anon" user ID specifies a UNIX user ID or user name that is mapped to client requests that arrive without valid NFS credentials. This can include the root user. Clustered Data ONTAP determines a user's file access permissions by checking the user's effective UID against the SVM's specified name-mapping and name-switch methods. Once the effective UID is determined, the export policy rule is leveraged to determine what access that UID has.

Note: The `-anon` option in export policy rules allows specification of a UNIX user ID or user name that is mapped to client requests that arrive without valid NFS credentials (including the root user). The default value of `-anon`, if not specified in export policy rule creation, is 65534. This UID is normally associated with the user name "nobody" or "nfsnobdy" in Linux environments. NetApp appliances use 65534 as the user "pcuser," which is generally used for multiprotocol operations, Because of this difference, if using local files and NFSv4, the name string for users mapped to 65534 might not match, which might cause files to be written as the user specified in the `/etc/idmapd.conf` file on the client (Linux) or `/etc/default/nfs` file (Solaris).

The Root User

The "root" user must be explicitly configured in clustered Data ONTAP to specify which machine has "root" access to a share, or else "anon=0" must be specified. Alternatively, the `-superuser` option can be used if more granular control over root access is desired. If these settings are not configured properly, "permission denied" might be encountered when accessing an NFS share as the "root" user (0). If the `-anon` option is not specified in export policy rule creation, the root user ID is mapped to the "nobody" user (65534). There are several ways to configure root access to an NFS share.

AUTH Types

When an NFS client authenticates, an AUTH type is sent. An AUTH type specifies how the client is attempting to authenticate to the server and completely depends on client-side configuration. Supported AUTH types include:

- AUTH_NONE/AUTH_NULL
  This AUTH type specifies that the request coming in has no identity (NONE or NULL) and will be mapped to the anon user. See http://www.ietf.org/rfc/rfc1050.txt and http://www.ietf.org/rfc/rfc2623.txt for details.
- AUTH_SYS/AUTH_UNIX
  This AUTH type specifies that the user is authenticated at the client (or system) and will come in as an identified user. See http://www.ietf.org/rfc/rfc1050.txt and http://www.ietf.org/rfc/rfc2623.txt for details.
- AUTH_RPCGSS
  This is kerberized NFS authentication.

Squashing Root

The following examples show how to squash root to anon in various configuration scenarios.

Example 1: Root is squashed to the anon user via superuser for all NFS clients using AUTH_SYS/AUTH_UNIX; other AUTH types are denied access.

```
cluster::> vserver export-policy rule show -policyname root_squash -instance
  (vserver export-policy rule show)

                                Vserver: vs0
                            Policy Name: root_squash
                             Rule Index: 1
                        Access Protocol: nfs     ← only NFS is allowed (NFSv3 and v4)
Client Match Hostname, IP Address, Netgroup, or Domain: 0.0.0.0/0   ← all clients
                         RO Access Rule: sys     ← only AUTH_SYS is allowed
                         RW Access Rule: sys     ← only AUTH_SYS is allowed
User ID To Which Anonymous Users Are Mapped: 65534   ← mapped to 65534
                  Superuser Security Types: none    ← superuser (root) squashed to anon user
              Honor SetUID Bits in SETATTR: true
                 Allow Creation of Devices: true

cluster::> volume show -vserver vs0 -volume nfsvol -fields policy
vserver volume policy
------- ------ ------------
vs0     nfsvol root_squash

[root@centos6 /]# mount -o nfsvers=3 cluster:/nfsvol /mnt
[root@centos6 /]# cd /mnt

[root@centos6 mnt]# touch root_squash

[root@centos6 mnt]# ls -la
total 116
drwxrwxrwx.  3 root   root   106496 Apr 24  2013 .
dr-xr-xr-x. 26 root   root     4096 Apr 24 11:24 ..
drwxr-xr-x.  2 root   daemon   4096 Apr 18 12:54 junction
-rw-r--r--.  1 nobody nobody      0 Apr 24 11:33 root_squash
```

```
[root@centos6 mnt]# ls -lan
drwxrwxrwx.  3     0     0 106496 Apr 24  2013 .
dr-xr-xr-x. 26     0     0   4096 Apr 24 11:24 ..
drwxrwxrwx. 12     0     0   4096 Apr 24 11:05 .snapshot
drwxr-xr-x.  2     0     1   4096 Apr 18 12:54 junction
-rw-r--r--.  1 65534 65534      0 Apr 24  2013 root_squash

[root@centos6 /]# mount -o sec=krb5 cluster:/nfsvol /mnt
mount.nfs: access denied by server while mounting cluster:/nfsvol
```

Example 2: Root is squashed to the anon user via superuser for a specific client; AUTH_SYS and AUTH_NONE (null) are allowed.

```
cluster::> vserver export-policy rule show -policyname root_squash_client -instance
  (vserver export-policy rule show)

                                Vserver: vs0
                            Policy Name: root_squash_client
                             Rule Index: 1
                        Access Protocol: nfs     ← only NFS is allowed (NFSv3 and v4)
Client Match Hostname, IP Address, Netgroup, or Domain: 10.10.100.25       ← just this client
                         RO Access Rule: sys,none       ← AUTH_SYS and AUTH_NONE are allowed
                         RW Access Rule: sys,none       ← AUTH_SYS and AUTH_NONE are allowed
User ID To Which Anonymous Users Are Mapped: 65534   ← mapped to 65534
                  Superuser Security Types: none    ← superuser (root) squashed to anon user
              Honor SetUID Bits in SETATTR: true
                 Allow Creation of Devices: true

cluster::> volume show -vserver vs0 -volume nfsvol -fields policy
vserver volume policy
------- ------ ------------
vs0     nfsvol root_squash_client

[root@centos6 /]# mount -o nfsvers=3 cluster:/nfsvol /mnt
[root@centos6 /]# cd /mnt

[root@centos6 mnt]# touch root_squash_client

[root@centos6 mnt]# ls -la
drwxrwxrwx.  3 root   root   106496 Apr 24  2013 .
dr-xr-xr-x. 26 root   root     4096 Apr 24 11:24 ..
drwxr-xr-x.  2 root   daemon   4096 Apr 18 12:54 junction
-rw-r--r--.  1 nfsnobody nfsnobody      0 Apr 24  2013 root_squash_client

[root@centos6 mnt]# ls -lan
drwxrwxrwx.  3     0     0 106496 Apr 24  2013 .
dr-xr-xr-x. 26     0     0   4096 Apr 24 11:24 ..
drwxrwxrwx. 12     0     0   4096 Apr 24 11:05 .snapshot
drwxr-xr-x.  2     0     1   4096 Apr 18 12:54 junction
-rw-r--r--.  1 65534 65534      0 Apr 24  2013 root_squash_client
```

Example 3: Root is squashed to the anon user via superuser for a specific set of clients using AUTH_RPCGSS (Kerberos) and only NFSv4 and CIFS are allowed.

```
cluster::> vserver export-policy rule show -policyname root_squash_krb5 -instance
  (vserver export-policy rule show)

                                Vserver: vs0
                            Policy Name: root_squash_krb5
                             Rule Index: 1
                        Access Protocol: nfs4,cifs      ← only NFSv4 and CIFS are allowed
Client Match Hostname, IP Address, Netgroup, or Domain: 10.10.100.0/24     ← just clients with
an IP address of 10.10.100.X
                         RO Access Rule: krb5    ← only AUTH_RPCGSSD is allowed
                         RW Access Rule: krb5    ← only AUTH_RPCGSSD is allowed
User ID To Which Anonymous Users Are Mapped: 65534   ← mapped to 65534
                  Superuser Security Types: none    ← superuser (root) squashed to anon user
```

```
               Honor SetUID Bits in SETATTR: true
                  Allow Creation of Devices: true

cluster::> volume show -vserver vs0 -volume nfsvol -fields policy
vserver volume policy
------- ------ -----------
vs0     nfsvol root_squash

[root@centos6 /]# mount -o nfsvers=3 cluster:/nfsvol /mnt
mount.nfs: access denied by server while mounting cluster:/nfsvol

[root@centos6 /]# mount -t nfs4 cluster:/nfsvol /mnt
mount.nfs4: Operation not permitted

[root@centos6 /]# mount -t nfs4 -o sec=krb5 krbsn:/nfsvol /mnt
[root@centos6 /]# cd /mnt

[root@centos6 mnt]# touch root_squash_krb5

[root@centos6 mnt]# ls -la
drwxrwxrwx.  3 root   root   106496 Apr 24  2013 .
dr-xr-xr-x. 26 root   root     4096 Apr 24 11:24 ..
drwxr-xr-x.  2 root   daemon   4096 Apr 18 12:54 junction
-rw-r--r--.  1 nobody nobody      0 Apr 24 11:50 root_squash_krb5

[root@centos6 mnt]# ls -lan
drwxrwxrwx.  3  0  0 106496 Apr 24  2013 .
dr-xr-xr-x. 26  0  0   4096 Apr 24 11:24 ..
drwxr-xr-x.  2  0  1   4096 Apr 18 12:54 junction
-rw-r--r--.  1 99 99      0 Apr 24 11:50 root_squash_krb5

NOTE: Note the UID of 99; this occurs in NFSv4 when the user name cannot map into the NFSv4
domain. /var/log/messages confirms this:
Apr 23 10:54:23 centos6 nfsidmap[1810]: nss_getpwnam: name 'pcuser' not found in domain
nfsv4domain.netapp.com'
```

In the above examples, when the root user requests access to a mount, it will map to the anon UID. In this case, the UID is 65534. This prevents unwanted root access from specified clients to the NFS share. Because “sys” is specified as the rw and ro access rules in the first two examples, only clients using AUTH_SYS will gain access. The third example shows a possible configuration using Kerberized NFS authentication. Setting the access protocol to NFS allows only NFS access to the share (including NFSv3 and NFSv4). If multiprotocol access is desired, then the access protocol must be set to allow NFS and CIFS. NFS access can be limited to only NFSv3 or NFSv4 here as well.

Root Is Root

The following examples show how to enable the root user to come into an NFS share as the root user.

Example 1: Root is allowed access as root via superuser for all clients only for AUTH_SYS; AUTH_NONE and AUTH_SYS are allowed rw and ro access; all other anon access is mapped to 65534.

```
cluster::> vserver export-policy rule show -policyname root_allow_anon_squash -instance
  (vserver export-policy rule show)

                                Vserver: vs0
                            Policy Name: root_allow_anon_squash
                             Rule Index: 1
                        Access Protocol: nfs     ← only NFS is allowed (NFSv3 and v4)
Client Match Hostname, IP Address, Netgroup, or Domain: 0.0.0.0/0   ← all clients
                          RO Access Rule: sys,none       ← AUTH_SYS and AUTH_NONE allowed
                          RW Access Rule: sys,none       ← AUTH_SYS and AUTH_NONE allowed
User ID To Which Anonymous Users Are Mapped: 65534   ← mapped to 65534
                  Superuser Security Types: sys     ← superuser for AUTH_SYS only
               Honor SetUID Bits in SETATTR: true

cluster::> volume show -vserver vs0 -volume nfsvol -fields policy
vserver volume policy
```

```
------- ------ -----------
vs0     nfsvol root_allow_anon_squash

[root@centos6 /]# mount -o nfsvers=3 cluster:/nfsvol /mnt
[root@centos6 /]# cd /mnt

[root@centos6 mnt]# touch root_allow_anon_squash_nfsv3

[root@centos6 mnt]# ls -la
drwxrwxrwx.  3 root      root      106496 Apr 24  2013 .
dr-xr-xr-x. 26 root      root        4096 Apr 24 11:24 ..
drwxrwxrwx. 12 root      root        4096 Apr 24 11:05 .snapshot
drwxr-xr-x.  2 root      bin         4096 Apr 18 12:54 junction
-rw-r--r--.  1 root      root           0 Apr 24  2013 root_allow_anon_squash_nfsv3

[root@centos6 mnt]# ls -lan
drwxrwxrwx.  3  0  0 106496 Apr 24  2013 .
dr-xr-xr-x. 26  0  0   4096 Apr 24 11:24 ..
drwxr-xr-x.  2  0  1   4096 Apr 18 12:54 junction
-rw-r--r--.  1  0  0      0 Apr 24 11:56 root_allow_anon_squash_nfsv3
```

Example 2: Root is allowed access as root via superuser for AUTH_RPCGSS (krb5) only; anon access is mapped to 65534; AUTH_SYS and AUTH_RPCGSS are allowed, but only via NFSv4.

```
cluster::> vserver export-policy rule show -policyname root_allow_krb5_only -instance
  (vserver export-policy rule show)

                                Vserver: vs0
                            Policy Name: root_allow_krb5_only
                             Rule Index: 1
                        Access Protocol: nfs4    ← only NFSv4 is allowed
Client Match Hostname, IP Address, Netgroup, or Domain: 0.0.0.0/0   ← all clients
                          RO Access Rule: sys,krb5       ← AUTH_SYS and AUTH_RPCGSS allowed
                          RW Access Rule: sys,krb5       ← AUTH_SYS and AUTH_RPCGSS allowed
User ID To Which Anonymous Users Are Mapped: 65534   ← mapped to 65534
                  Superuser Security Types: krb5    ← superuser via AUTH_RPCGSS only
              Honor SetUID Bits in SETATTR: true

cluster::> volume show -vserver vs0 -volume nfsvol -fields policy
vserver volume policy
------- ------ -----------
vs0     nfsvol root_allow_krb5_only

[root@centos6 /]# mount -o nfsvers=3 cluster:/nfsvol /mnt
mount.nfs: access denied by server while mounting cluster:/nfsvol

[root@centos6 /]# mount -t nfs4 cluster:/nfsvol /mnt
[root@centos6 /]# cd /mnt

[root@centos6 mnt]# touch root_allow_krb5_only_notkrb5

[root@centos6 mnt]# ls -la
drwxrwxrwx.  3 root   root   106496 Apr 24  2013 .
dr-xr-xr-x. 26 root   root     4096 Apr 24 11:24 ..
drwxr-xr-x.  2 root   daemon   4096 Apr 18 12:54 junction
-rw-r--r--.  1 nobody nobody      0 Apr 24  2013 root_allow_krb5_only_notkrb5

[root@centos6 mnt]# ls -lan
drwxrwxrwx.  3  0  0 106496 Apr 24  2013 .
dr-xr-xr-x. 26  0  0   4096 Apr 24 11:24 ..
drwxr-xr-x.  2  0  1   4096 Apr 18 12:54 junction
-rw-r--r--.  1 99 99      0 Apr 24  2013 root_allow_krb5_only_notkrb5


NOTE: Again, the UID of an unmapped user in NFSv4 is 99. This is controlled via /etc/idmapd.conf
in Linux and /etc/default/nfs in Solaris.

[root@centos6 /]# mount -t nfs4 -o sec=krb5 cluster:/nfsvol /mnt
[root@centos6 /]# kinit
Password for root@KRB5DOMAIN.NETAPP.COM:
```

```
[root@centos6 /]# cd /mnt

[root@centos6 mnt]# touch root_allow_krb5_only_krb5mount

[root@centos6 mnt]# ls -la
drwxrwxrwx.  3 root   root   106496 Apr 24  2013 .
dr-xr-xr-x. 26 root   root     4096 Apr 24 11:24 ..
drwxr-xr-x.  2 root   daemon   4096 Apr 18 12:54 junction
-rw-r--r--.  1 root   daemon      0 Apr 24  2013 root_allow_krb5_only_krb5mount

[root@centos6 mnt]# ls -lan
drwxrwxrwx.  3  0  0 106496 Apr 24  2013 .
dr-xr-xr-x. 26  0  0   4096 Apr 24 11:24 ..
drwxr-xr-x.  2  0  1   4096 Apr 18 12:54 junction
-rw-r--r--.  1  0  1      0 Apr 24  2013 root_allow_krb5_only_krb5mount
```

Example 3: Root and all anonymous users are allowed access as root via anon=0, but only for AUTH_SYS and AUTH_RPCGSS over NFSv4.

```
cluster::> vserver export-policy rule show -policyname root_allow_anon0 -instance
  (vserver export-policy rule show)

                                Vserver: vs0
                            Policy Name: root_allow_anon0
                             Rule Index: 1
                        Access Protocol: nfs4    ← only NFSv4 is allowed
Client Match Hostname, IP Address, Netgroup, or Domain: 0.0.0.0/0   ← all clients
                         RO Access Rule: krb5, sys      ← AUTH_SYS and AUTH_RPCGSS allowed
                         RW Access Rule: krb5, sys      ← AUTH_SYS and AUTH_RPCGSS allowed
User ID To Which Anonymous Users Are Mapped: 0       ← mapped to 0
                  Superuser Security Types: none    ← superuser (root) squashed to anon user
              Honor SetUID Bits in SETATTR: true
                 Allow Creation of Devices: true


cluster::> volume show -vserver vs0 -volume nfsvol -fields policy
vserver volume policy
------- ------ -----------
vs0     nfsvol root_allow_anon0

[root@centos6 /]# mount -o nfsvers=3 cluster:/nfsvol /mnt
mount.nfs: access denied by server while mounting cluster:/nfsvol

[root@centos6 /]# mount -t nfs4 cluster:/nfsvol /mnt
[root@centos6 /]# cd /mnt

[root@centos6 mnt]# touch root_allow_anon0

[root@centos6 mnt]# ls -la
drwxrwxrwx.  3 root   root   106496 Apr 24  2013 .
dr-xr-xr-x. 26 root   root     4096 Apr 24 11:24 ..
drwxr-xr-x.  2 root   daemon   4096 Apr 18 12:54 junction
-rw-r--r--.  1 root   daemon      0 Apr 24  2013 root_allow_anon0

[root@centos6 mnt]# ls -lan
drwxrwxrwx.  3  0  0 106496 Apr 24  2013 .
dr-xr-xr-x. 26  0  0   4096 Apr 24 11:24 ..
drwxr-xr-x.  2  0  1   4096 Apr 18 12:54 junction
-rw-r--r--.  1  0  1      0 Apr 24  2013 root_allow_anon0


[root@centos6 /]# mount -t nfs4 -o sec=krb5 cluster:/nfsvol /mnt
[root@centos6 /]# cd /mnt

[root@centos6 mnt]# touch root_allow_anon0_krb5

[root@centos6 mnt]# ls -la
drwxrwxrwx.  3 root   root   106496 Apr 24  2013 .
dr-xr-xr-x. 26 root   root     4096 Apr 24 11:24 ..
drwxr-xr-x.  2 root   daemon   4096 Apr 18 12:54 junction
```

```
-rw-r--r--.  1 root   daemon      0 Apr 24  2013 root_allow_anon0_krb5

[root@centos6 mnt]# ls -lan
drwxrwxrwx.  3  0  0 106496 Apr 24  2013 .
dr-xr-xr-x. 26  0  0   4096 Apr 24 11:24 ..
drwxr-xr-x.  2  0  1   4096 Apr 18 12:54 junction
-rw-r--r--.  1  0  1      0 Apr 24  2013 root_allow_anon0_krb5
```

Showmount in Clustered Data ONTAP

Clustered Data ONTAP does not support the `showmount` command from NFS clients. The reasoning behind this is that, due to performance considerations, clusters can potentially have thousands of export rules, so a query for all exports can be process intensive. Additionally, exports are not in flat files and are applied to volumes as rules, so the export path and export rules would live in two different places.

Example of `showmount –e` in 7-Mode:

```
[root@centos6 /]# showmount -e 10.61.84.240
Export list for 10.61.84.240:
/vol/unix      (everyone)
/vol/Test      (everyone)
/vol/vol0/home (everyone)
/vol/vol0      (everyone)
/vol/Test2     (everyone)
/vol/mixed     10.61.179.164
```

Example of `showmount –e` in clustered Data ONTAP:

```
[root@centos6 /]# showmount -e 10.61.92.34
Export list for 10.61.92.34:
/ (everyone)
```

When running a showmount in clustered Data ONTAP, the NFS server would be an SVM IP. The SVM has a vsroot volume mounted to /, which is the volume returned in the showmount. All other volumes are mounted below that mount point. In the above example, / is shown as allowing everyone. This is the export policy rule for / in the SVM being queried:

```
cluster::> vol show -vserver vs0 -volume vsroot -fields policy
  (volume show)
vserver volume      policy
------- ----------- -------
vs0     vsroot default

cluster::> export-policy rule show -vserver vs0 -policyname default -instance
  (vserver export-policy rule show)

                                Vserver: vs0
                            Policy Name: default
                             Rule Index: 1
                          Access Protocol: any
Client Match Hostname, IP Address, Netgroup, or Domain: 0.0.0.0/0
                            RO Access Rule: any
                            RW Access Rule: any
User ID To Which Anonymous Users Are Mapped: 65534
                  Superuser Security Types: any
              Honor SetUID Bits in SETATTR: true
                 Allow Creation of Devices: true
```

If the export policy rule is changed to allow just a host, the `showmount –e` output does not change:

```
cluster::> export-policy rule modify -vserver vs0 -policyname default -ruleindex 1 -clientmatch
10.61.179.164
  (vserver export-policy rule modify)

cluster::> export-policy rule show -vserver vs0 -policyname default -instance
```

```
  (vserver export-policy rule show)


                                    Vserver: vs0
                                Policy Name: default
                                 Rule Index: 1
                            Access Protocol: any
Client Match Hostname, IP Address, Netgroup, or Domain: 10.61.179.164
                              RO Access Rule: any
                              RW Access Rule: any
User ID To Which Anonymous Users Are Mapped: 65534
                    Superuser Security Types: any
                Honor SetUID Bits in SETATTR: true
                   Allow Creation of Devices: true

[root@centos6 /]# showmount -e 10.61.92.34
Export list for 10.61.92.34:
/ (everyone)
```

Thus, for clustered Data ONTAP, showmount isn't really useful in most cases. To get similar functionality to showmount, leverage SSH or the Data ONTAP SDK to extract the desired information. The fields to extract would be:

- Junction-path from the `volume show` command/ZAPI
- Policy from the `volume show` command/ZAPI
- Any desired fields from the export policy rule set in the policy assigned to the volume

Showmount Plug-in for Clustered Data ONTAP

The support tool chest now contains a showmount plug-in for clustered Data ONTAP. This plug-in has limited support and should be used only in situations where showmount is required.

3.5 Creating Local Netgroups

When creating export policies and rules, netgroup names can be specified instead of an IP address and mask bits to match clients to an export rule. A netgroup is a named collection of arbitrary IP addresses that is stored in an NIS map.

Export policies are not specific to any one virtual server; however, because each virtual server has an independent NIS domain and the set of IP addresses that a netgroup matches depends on NIS, each netgroup-based rule can match different clients on different virtual servers that have different NIS domains.

Netgroup creation is covered in the "File Access and Protocols Management Guide" for the version of clustered Data ONTAP being used.

4 NFSv4.x in Clustered Data ONTAP

NFSv4.0 and NFSv4.1 were introduced for the first time in clustered Data ONTAP starting with Data ONTAP 8.1.

Advantages of Using NFSv4.x

- Firewall-friendly because NFSv4 uses only a single port (2049) for its operations
- Advanced and aggressive cache management, like delegation in NFSv4.0 (does not apply in NFSv4.1)
- Mandatory strong RPC security flavors that employ cryptography
- Internationalization
- Compound operations

- Works only with TCP
- Stateful protocol (not stateless like NFSv3)
- Kerberos configuration for efficient authentication mechanisms (uses 3DES for encryption)
- No replication support
- Migration (for dNFS) using referrals
- Support of access control that is compatible with UNIX and Windows
- String-based user and group identifiers
- Parallel access to data (does not apply for NFSv4.0)

4.1 NFSv4.0

Recently there has been a major increase in the adoption of NFSv4 for various business requirements. While customers prepare to migrate their existing setup and infrastructure from NFSv3 to NFSv4, some environmental changes must be made before moving to NFSv4. One of them is "id domain mapping," as mentioned later in this table.

Some production environments have the challenge to build new naming service infrastructures like NIS or LDAP for string-based name mapping to be functional in order to move to NFSv4. With the new "numeric_id" option, setting name services does not become an absolute requirement. The "numeric_id" feature must be supported and enabled on the server as well as on the client. With this option enabled, the user and groups exchange UIDs/GIDs between the client and server just as with NFSv3. However, for this option to be enabled and functional, NetApp recommends having a supported version of the client and the server. Today the first available client that supports this feature is Fedora15 on kernel 3.0 and later.

In clustered Data ONTAP 8.1, a new option called v4-id-numerics was added. With this option enabled, even if the client does not have access to the name mappings, IDs can be sent in the user name and group name fields and the server accepts them and treats them as representing the same user as would be represented by a v2/v3 UID or GID having the corresponding numeric value.

Note: To access this command, you must be in diag mode. Commands related to diag mode should be used with caution, and NetApp recommends that you contact the NetApp Support team for further advice.

Note: Note that -v4-id-numerics should be enabled only if the client supports it.

Table 2) Enabling numeric ID support for NFSv4 in clustered Data ONTAP.

Category	Commands
Enable NFSv4.0.	
	`cluster::> vserver nfs modify -vserver test_vs1 -access true -v4.0 enabled -tcp enabled`
	Verification

	`cluster::> vserver nfs show -vserver test_vs1`
	
	`Vserver: test_vs1`
	`General NFS Access: true`
	`NFS v3: enabled`
	`NFS v4.0: enabled`
	`UDP Protocol: enabled`
	`TCP Protocol: enabled`
	`Spin Authentication: disabled`
	`Default Windows User: -`
	`NFSv4.0 ACL Support: disabled`
	`NFSv4.0 Read Delegation Support: disabled`
	`NFSv4.0 Write Delegation Support: disabled`

	NFSv4 ID Mapping Domain: defaultv4iddomain.com NFSv4.1 Minor Version Support: disabled Rquota Enable: disabled NFSv4.1 Parallel NFS Support: enabled NFSv4.1 ACL Support: disabled NFS vStorage Support: disabled
Set up NFSv4 user ID mapping.	**Note:** On a clustered Data ONTAP system, the command to turn on the v4-id-numerics option follows.
	cluster::> set diag Warning: These diagnostic commands are for use by NetApp personnel only. Do you want to continue? {y|n}: y cluster::> vserver nfs modify -vserver testvs1 -v4-numeric-ids enabled
	Verification
	cluster::> vserver nfs show -vserver testvs1 -fields v4-numeric-ids Vserver v4-numeric-ids ------- -------------- testvs1 enabled
	If the `v4-id-numerics` option is disabled, the server only accepts user name/group name of the form user@domain or group@domain. The NFSv4 domain name is a pseudo-domain name that both the client and storage controller must agree upon before they can execute NFSv4 operations. The NFSv4 domain name might or might not be equal to the NIS or DNS domain name, but it must be a string that both the NFSv4 client and server understand. This is a two-step process in which the Linux client and clustered Data ONTAP system are configured with the NFSv4 domain name. **On the clustered Data ONTAP system:** The default value of the NFS option -v4-id-domain is defaultv4iddomain.com.
	cluster::> vserver nfs modify -vserver test_vs1 -v4-id-domain nfsv4domain.netapp.com
	Verification
	cluster::> ::> vserver nfs show -vserver test_vs1 -fields v4-id-domain

	`Vserver  v4-id-domain`
	`-------- ------------------------------`
	`test_vs1 nfsv4domain.netapp.com`

	This section describes how the domain name can be changed on the client. **Solaris.** Edit the `/etc/default/nfs` file and change NFSMAPID_DOMAIN to that set for the server. Reboot the client for the change to take effect. **Linux.** Make the necessary adjustments to `/etc/idmapd.conf`. Restart the idmapd process to have the change take effect. NOTE: Restarting idmapd varies per client. Rebooting the server is an option as well.

	`[root@linuxlinux /]# vi /etc/idmapd.conf`
	`[General]`
	
	`Verbosity = 0`
	`Pipefs-Directory = /var/lib/nfs/rpc_pipefs`
	`Domain = nfsv4domain.netapp.com`
	
	`[mapping]`
	
	`Nobody-User = nobody`
	`Nobody-Group = nobody`
	
	`[Translation]`
	`Method = nsswitch`

Create a UNIX group with GID 1 and assign it to the SVM.	**Note:** Whenever a volume is created, it is associated with UID 0 and GID 1 by default. NFSv3 ignores this, whereas NFSv4 is sensitive to the UID and GID mapping. If GID 1 was not previously created, follow these steps to create one.
	`cluster::> vserver services unix-group create -vserver test_vs1 -name daemon -id 1`
	Verification

	`cluster::> vserver services unix-group show -vserver test_vs1`
	
	`Vserver  Name  ID`
	`-------------- -------------------- ----------`
	`test_vs1  daemon  1`
	`test_vs1  root  0`
	`2 entries were displayed.`

Mounting the client over NFSv4	**On the client:**
	`[root@linux /]# mkdir -p  /home/root/mnt/nfs4/` `[root@linux /]# mount 172.17.37.135:/path01 /home/root/mnt/nfs4/`
	Verification

	`[root@linux /]# mount`
	`172.17.37.135:/path01 on /home/root/mnt/test_vs1 type nfs (rw,vers=3,addr=172.17.37.135)`
	`172.17.37.135:/path01 on /home/root/mnt/ nfs4 type nfs (rw,vers=4,addr=172.17.37.135,clientaddr=172.17.44.42)`

	Note: Linux clients must mount the file system from the NetApp storage with a "-t nfs4" option. However, RHEL 6.0 and later mount NFSv4 by default. Solaris10 clients mount the file system over NFSv4 by default when NFSv4 is enabled on the NetApp storage appliance. For mounting over NFSv3, "vers=3" must be explicitly specified on the mounts. **Note:** A volume can be mounted via NFSv3 and NFSv4.

Configure UID and GID Name Mappings

Use any of three ways of modifying `file/nis/ldap`. The order of mapping is specified using the commands shown in Table 3.

Table 3) Configuring UID and GID mapping.

Category	Commands
Configure name-mapping methodologies.	
	`cluster::> vserver modify –vserver test_vs1 –ns-switch nis,ldap –nm-switch file`
Configure LDAP.	**Create an LDAP client.**
	`cluster::> vserver services ldap client show` `This table is currently empty.` LDAP using Active Directory®: `cluster::> vserver services ldap client create -client-config AD_LDAP -servers 10.10.10.100 –ad-domain domain.netapp.com -bind-as-cifs-server true -schema AD-IDMU -port 389 -query-timeout 3 -min-bind-level sasl -base-dn DC=domain,DC=netapp,DC=com -base-scope subtree -preferred-ad-servers 10.10.10.100`  Non-Active Directory LDAP (such as OpenLDAP): `cluster::> vserver services ldap client create -client-config OPENLDAP -schema RFC-2307 –servers 10.10.10.101 -port 389 -query-timeout 3 -min-bind-level simple -base-dn DC=openldap,DC=netapp,DC=com -base-scope subtree`
	Verification

	LDAP using Active Directory:
	`cluster::> vserver services ldap client show -instance`
	`Client Configuration Name: AD_LDAP`
	`LDAP Server List: 10.10.10.100`
	`Active Directory Domain: domain.netapp.com`

	Preferred Active Directory Servers: 10.10.10.100 Bind Using the Vserver's CIFS Credentials: true Schema Template: AD-IDMU LDAP Server Port: 389 Query Timeout (sec): 3 Minimum Bind Authentication Level: sasl Bind DN (User): - Base DN:DC=domain,DC=netapp, DC=com Base Search Scope: subtree Non-Active Directory LDAP (such as OpenLDAP): cluster::> vserver services ldap client show -instance Client Configuration Name: OPENLDAP LDAP Server List: 10.10.10.101 Active Directory Domain: - Preferred Active Directory Servers: - Bind Using the Vserver's CIFS Credentials: truefalse Schema Template: RFC-2307 LDAP Server Port: 389 Query Timeout (sec): 3 Minimum Bind Authentication Level: sasl Bind DN (User): - Base DN:DC=openldap,DC=netapp, DC=com Base Search Scope: subtree
	Create an LDAP server.
	cluster::> vserver services ldap show This table is currently empty. cluster::> vserver services ldap create -vserver test_vs1 -client-config ldapclient1 -client-enabled true
	Verification
	cluster::> vserver services ldap show Client Client Vserver Configuration Enabled ---------- ---------- ------- test_vs1 ldapclient1 true
Configure NIS.	
	cluster::> vserver services nis-domain create -vserver test_vs1 -domain nisdom.netapp.com -active true -servers 10.10.10.110
	Verification
	cluster::> vserver services nis-domain show NIS Vserver Domain Active Server ---------- ---------- ------ ---------- test_vs1 nisdom.netapp.com true 10.10.10.110

Viewing Active NFS Connections in the Cluster

In clustered Data ONTAP, it is possible to view active NFS connections across all SVMs and nodes in the cluster via the `network connections active show` command. This command allows filtering of IPs, services, and other features to provide more useful and granular information. This can be used in place of classic netstat commands found in 7-Mode.

Example:

```
cluster::> network connections active show
    show           show-clients   show-lifs      show-protocols show-services

cluster::> network connections active show -node node1 -service nfs*
             Vserver   Interface         Remote
      CID Ctx Name     Name:Local Port   Host:Port            Protocol/Service
--------- --- -------- ----------------- -------------------- ----------------
Node: node1
286571835   6 vs0          data:2049         10.61.179.164:763    TCP/nfs

cluster::> network connections active show -node node2 -service nfs*
There are no entries matching your query.
```

NFSv4 Access Control Lists (ACLs)

The NFSv4 protocol can provide access control in the form of NFSv4 Access Control Lists (ACLs), which are similar in concept to those found in CIFS. An NFSv4 ACL consists of individual Access Control Entries (ACEs), each of which provides an access control directive to the server. Clustered Data ONTAP 8.2 supports a maximum of 1,024 ACEs.

Benefits of Enabling NFSv4 ACLs

The benefits of enabling NFSv4 ACLs include the following:

- Granular control of user access to files and directories
- Better NFS security
- Improved interoperability with CIFS
- Removal of the NFS limitation of 16 groups per user with AUTH_SYS security

Compatibility Between NFSv4 ACLs and Windows (NTFS) ACLs

NFSv4 ACLs are different from Windows file-level ACLs (NTFS ACLs), but Data ONTAP can map NFSv4 ACLs to Windows ACLs for viewing on Windows platforms. Permissions displayed to NFS clients for files that have Windows ACLs are "display" permissions, and the permissions used for checking file access are those of the Windows ACL.

Note: Data ONTAP does not support POSIX ACLs.

How NFSv4 ACLs Work

When a client sets an NFSv4 ACL on a file during a SETATTR operation, the NetApp storage system sets that ACL on the object, replacing any existing ACL. If there is no ACL on a file, then the mode permissions on the file are calculated from OWNER@, GROUP@, and EVERYONE@. If there are any existing SUID/SGID/STICKY bits on the file, they are not affected.

When a client gets an NFSv4 ACL on a file during the course of a GETATTR operation, the NetApp system reads the NFSV4 ACL associated with the object and constructs a list of ACEs and returns it to the client. If the file has an NT ACL or mode bits, then an ACL is constructed from mode bits and is returned to the client.

Access is denied if a DENY ACE is present in the ACL, and access is granted if an ALLOW ACE exists. However, access is also denied if neither of the ACEs is present in the ACL.

A security descriptor consists of a Security ACL (SACL) and a Discretionary ACL (DACL). When NFSv4 interoperates with CIFS, the DACL is one-to-one mapped with NFSv4 and CIFS. The DACL consists of the ALLOW and the DENY ACEs.

Table 4) Enabling NFSv4 access control lists.

Category	Commands
Modify the NFSv4 server to enable ACLs by enabling the `-v4.0-acl` option.	`cluster::> vserver nfs modify -vserver test_vs1 -v4.0-acl enabled`
	Verification

	`cluster::> vserver nfs show -vserver test_vs1 -fields v4.0-acl,v4.0`
	`Vserver  v4.0    v4.0-acl`
	`-------- ------- --------`
	`test_vs1 enabled enabled`

On a Linux client	**Note:** After you enable ACLs on the server, the `nfs4_setfacl` and `nfs4_getfacl` commands are required on the Linux client to set or get NFSv4 ACLs on a file or directory, respectively. To avoid problems with earlier implementations, use RHEL 5.8 or RHEL 6.2 and later for using NFSv4 ACLs in clustered Data ONTAP. The following example illustrates the use of the `-e` option to set the ACLs on the file or directory from the client. To learn more about the types of ACEs that can be used, refer to the following links: www.linuxcertif.com/man/1/nfs4_setfacl/145707/ http://linux.die.net/man/5/nfs4_acl

	`[root@linux /]# mount 172.17.37.135:/path01 /home/root/mnt/nfs4/`
	`[root@linux /]# mount`
	`172.17.37.135:/path01 on /home/root/mnt/ nfs4 type nfs`
	`(rw,vers=4,addr=172.17.37.135,clientaddr=172.17.44.42)`
	
	`[root@linux /]# cd /home/root/mnt/nfs4`
	`[root@linux nfs4]# ls -al`
	`total 8`
	`drwxr-xr-x. 2 root root 4096 Jul 27 12:56 ./`
	`drwxr-xr-x. 3 root root 4096 Jul 27 12:56 ../`
	`[root@linux nfs4] # touch aa`
	`[root@linux nfs4] # nfs4_setfacl -e aa`

	## Editing NFSv4 ACL for file: `/home/root/mnt/ nfs4/aa`:

	`A::OWNER@:rwatTnNcCy`
	`D::OWNER@:x`
	`A:g:GROUP@:rtncy`
	`D:g:GROUP@:waxTC`
	`A::EVERYONE@:rtncCy`
	`D::EVERYONE@:waxT`

Category	Commands
	Verification

	`[root@linux nfs4] # nfs4_getfacl aa`
	`A::OWNER@:rwatTnNcCy`
	`D::OWNER@:x`
	`A:g:GROUP@:rtncy`
	`D:g:GROUP@:waxTC`
	`A::EVERYONE@:rtncCy`
	`D::EVERYONE@:waxT`

A client using NFSv4 ACLs can set and view ACLs for files and directories on the system. When a new file or subdirectory is created in a directory that has an ACL, the new file or subdirectory inherits all ACEs in the ACL that have been tagged with the appropriate inheritance flags. For access checking, CIFS users are mapped to UNIX users. The mapped UNIX user and that user's group membership are checked against the ACL.

If a file or directory has an ACL, that ACL is used to control access no matter which protocol—NFSv3, NFSv4, or CIFS—is used to access the file or directory and is used even if NFSv4 is no longer enabled on the system.

Files and directories inherit ACEs from NFSv4 ACLs on parent directories (possibly with appropriate modifications) as long as the ACEs have been tagged with the correct inheritance flags. Clustered Data ONTAP 8.2 supports up to 1,024 ACEs. This can be controlled via the following command:

```
cluster::> nfs server modify –vserver vs0 –v4-acl-max-aces [number up to 1024]
```

Prior to clustered Data ONTAP 8.2, the maximum ACE limit was 400. If reverting to a version of Data ONTAP prior to 8.2, files or directories with more than 400 ACEs will have their ACLs dropped and the security will revert to mode bit style.

When a file or directory is created as the result of an NFSv4 request, the ACL on the resulting file or directory depends on whether the file creation request includes an ACL or only standard UNIX file access permissions, and whether the parent directory has an ACL.

- If the request includes an ACL, that ACL is used.
- If the request includes only standard UNIX file access permissions but the parent directory has an ACL, the ACEs in the parent directory's ACL are inherited by the new file or directory as long as the ACEs have been tagged with the appropriate inheritance flags.

Note: A parent ACL is inherited even if `-v4.0-acl` is set to `off`.

- If the request includes only standard UNIX file access permissions and the parent directory does not have an ACL, the client file mode is used to set standard UNIX file access permissions.
- If the request includes only standard UNIX file access permissions and the parent directory has a non-inheritable ACL, a default ACL based on the mode bits passed into the request is set on the new object.

ACL Formatting

NFSv4.x ACLs have specific formatting. The following is an ACE set on a file:

`A::ldapuser@domain.netapp.com:rwatTnNcCy`

The preceding follows the ACL format guidelines of:

`type:flags:principal:permissions`

A type of “A” means allow. The flags are not set in this case, because the principal is not a group and does not include inheritance. Also, because the ACE is not an AUDIT entry, there is no need to set the audit flags.

For more information on NFSv4.x ACLs, see http://linux.die.net/man/5/nfs4_acl.

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.

Mixed Security Style Considerations

Mixed qtree styles can cause issues with permissions if not set up properly. It can also be confusing to know what permissions are set on a file or folder when using mixed security style, since the NFS or CIFS clients might not display the ACLs properly. Mixed security style can get messy when clients are modifying permissions, even with identity management in place.

Best Practice

Choose either NTFS or UNIX style security unless there is a specific recommendation from an application vendor to use mixed mode.

- For any NT user, the user's SID is mapped to a UNIX ID and the NFSv4 ACL is then checked for access for that UNIX ID. Regardless of which permissions are displayed, the actual permissions set on the file take effect and are returned to the client.
- If a file has an NT ACL and a UNIX client does a `chmod`, `chgrp`, or `chown`, the NT ACL is dropped.

In clustered Data ONTAP 8.1.x and prior versions, run the following command on the node that owns the volume:

```
cluster::> node run -node nodename “fsecurity show /vol/volname”
```

```
In clustered Data ONTAP 8.2 and later, use the following command:cluster::> vserver security
file-directory show -vserver vs0 -path /junction-path
```

Explicit DENY

NFSv4 permissions may include explicit DENY attributes for OWNER, GROUP, and EVERYONE. That is because NFSv4 ACLs are "default-deny," which means that if an ACL is not explicitly granted by an ACE, then it is denied.

```
sh-4.1$ nfs4_getfacl /mixed
A::ldapuser@domain.netapp.com:ratTnNcCy
A::OWNER@:rwaDxtTnNcCy
D::OWNER@:
A:g:GROUP@:rxtncy
D:g:GROUP@:waDTC
A::EVERYONE@:rxtncy
D::EVERYONE@:waDTC
```

DENY ACEs should be avoided whenever possible, since they can be confusing and complicated. When DENY ACEs are set, users might be denied access when they expect to be granted access. This is because the ordering of NFSv4 ACLs affects how they are evaluated.

The above set of ACEs is the equivalent to 755 in mode bits. That means:

- Owner has full rights
- Groups have read only
- Others have read only

However, even if permissions are adjusted to the 775 equivalent, access can be denied due to the explicit DENY set on EVERYONE.

For example, the user "ldapuser" belongs to the group "Domain Users."

```
sh-4.1$ id
uid=55(ldapuser) gid=513(Domain Users) groups=513(Domain Users),503(unixadmins)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Permissions on the volume "mixed" are 775. The owner is root and the group is "Domain Users:"

```
[root@centos6 /]# nfs4_getfacl /mixed
A::OWNER@:rwaDxtTnNcCy
D::OWNER@:
A:g:GROUP@:rwaDxtTnNcy
D:g:GROUP@:C
A::EVERYONE@:rxtncy
D::EVERYONE@:waDTC

[root@centos6 /]# ls -la | grep mixed
drwxrwxr-x.   3 root     Domain Users   4096 Apr 30 09:52 mixed
```

Since "ldapuser" is a member of Domain Users, it should have write access to the volume, and it does:

```
[root@centos6 /]# su ldapuser
sh-4.1$ cd /mixed
sh-4.1$ ls -la
total 12
drwxrwxr-x.  3 root Domain Users 4096 Apr 30 09:52 .
dr-xr-xr-x. 28 root root         4096 Apr 29 15:24 ..
drwxrwxrwx.  6 root root         4096 Apr 30 08:00 .snapshot
sh-4.1$ touch newfile
sh-4.1$ nfs4_getfacl /mixed

sh-4.1$ ls -la
total 12
drwxrwxr-x.  3 root     Domain Users 4096 Apr 30 09:56 .
dr-xr-xr-x. 28 root     root         4096 Apr 29 15:24 ..
drwxrwxrwx.  6 root     root         4096 Apr 30 08:00 .snapshot
-rw-r--r--.  1 ldapuser Domain Users    0 Apr 30 09:56 newfile
```

However, if the ACLs are reordered and the explicit DENY for EVERYONE is placed ahead of group, then "ldapuser" is denied access to write to the same volume it just had access to write to:

```
[root@centos6 /]# nfs4_getfacl /mixed
A::OWNER@:rwaDxtTnNcCy
D::OWNER@:
A::EVERYONE@:rxtncy
D::EVERYONE@:waDTC
A:g:GROUP@:rwaDxtTnNcy

[root@centos6 /]# su ldapuser
sh-4.1$ cd /mixed
sh-4.1$ ls -la
total 12
drwxrwxr-x.  3 root     Domain Users 4096 Apr 30 09:56 .
dr-xr-xr-x. 28 root     root         4096 Apr 29 15:24 ..
drwxrwxrwx.  6 root     root         4096 Apr 30 08:00 .snapshot
-rw-r--r--.  1 ldapuser Domain Users    0 Apr 30 09:56 newfile

sh-4.1$ touch newfile2
touch: cannot touch `newfile2': Permission denied
```

If the explicit DENY rule is removed, the desired access is restored:

```
[root@centos6 /]# nfs4_getfacl /mixed
A::OWNER@:rwaDxtTnNcCy
D::OWNER@:
A::EVERYONE@:rxtncy
A:g:GROUP@:rwaDxtTnNcy

[root@centos6 /]# su ldapuser

sh-4.1$ cd /mixed

sh-4.1$ ls -la
total 12
drwxrwxr-x.  3 root     Domain Users 4096 Apr 30 09:56 .
dr-xr-xr-x. 28 root     root         4096 Apr 29 15:24 ..
drwxrwxrwx.  6 root     root         4096 Apr 30 08:00 .snapshot
-rw-r--r--.  1 ldapuser Domain Users    0 Apr 30 09:56 newfile

sh-4.1$ touch newfile2

sh-4.1$ ls -la
total 12
drwxrwxr-x.  3 root     Domain Users 4096 Apr 30 10:06 .
dr-xr-xr-x. 28 root     root         4096 Apr 29 15:24 ..
drwxrwxrwx.  6 root     root         4096 Apr 30 08:00 .snapshot
-rw-r--r--.  1 ldapuser Domain Users    0 Apr 30 09:56 newfile
-rw-r--r--.  1 ldapuser Domain Users    0 Apr 30 10:06 newfile2
```

Best Practice

It is a best practice to set DENY ACEs only when absolutely necessary.

NFSv4 ACL Preservation

By default, NFSv4 ACLs can be affected by setting mode bits on a file or folder. If an NFSv4 ACE has been configured and a chmod is used, the ACE will be removed. This behavior can be avoided by setting the following on the NetApp storage system:

```
cluster::> set diag
cluster::*> nfs server modify -vserver vs0 -v4-acl-preserve enabled
```

NetApp recommends this option in environments using NFSv3 and NFSv4 on the same NFS exports.

ACL Preservation in Action

This is a newly created UNIX-style volume:

```
cluster::> volume show -vserver vs0 -volume unix -fields security-style,
unix-permissions,user,group
vserver volume user group security-style unix-permissions
------- ------ ---- ----- -------------- ----------------
vs0     unix   0    1     unix           ---rwxr-xr-x

cluster ::> vserver security file-directory show -vserver vs0 -path /unix

                Vserver: vs0
              File Path: /unix
         Security Style: unix
        Effective Style: unix
         DOS Attributes: 10
 DOS Attributes in Text: ----D---
Expanded Dos Attributes: -
           Unix User Id: 0
          Unix Group Id: 1
         Unix Mode Bits: 755
 Unix Mode Bits in Text: rwxr-xr-x
                   ACLs: -
```

In the above example, the volume (/unix) has 755 permissions. That means the owner has ALL access, the owning group has READ/EXECUTE access, and everyone else has READ/EXECUTE access.

Even though there are no NFSv4 ACLs in the fsecurity output, there are default values set that can be viewed from the client:

```
[root@centos6 /]# mount -t nfs4 krbsn:/unix /unix
[root@centos6 /]# ls -la | grep unix
drwxr-xr-x.   2 root     daemon         4096 Apr 30 11:24 unix
[root@centos6 /]# nfs4_getfacl /unix
A::OWNER@:rwaDxtTnNcCy
A:g:GROUP@:rxtncy
A::EVERYONE@:rxtncy
```

The NFSv4 ACLs above show the same—the owner has ALL access, the owning group has READ/EXECUTE access, and everyone else has READ/EXECUTE access. The default mode bits are tied to the NFSv4 ACLs.

When mode bits are changed, the NFSv4 ACLs are also changed:

```
[root@centos6 /]# chmod 775 /unix
[root@centos6 /]# ls -la | grep unix
drwxrwxr-x.   2 root     daemon         4096 Apr 30 11:24 unix
[root@centos6 /]# nfs4_getfacl /unix
A::OWNER@:rwaDxtTnNcCy
A:g:GROUP@:rwaDxtTnNcy
A::EVERYONE@:rxtncy
```

When a user ACE is added to the ACL, the entry is reflected in the ACL on the appliance. In addition, the entire ACL is now populated. Note that the ACL is in SID format.

```
[root@centos6 /]# nfs4_setfacl -a A::ldapuser@nfsv4domain.netapp.com:ratTnNcCy /unix
[root@centos6 /]# nfs4_getfacl /unix
A::ldapuser@nfsv4domain.netapp.com:ratTnNcCy
A::OWNER@:rwaDxtTnNcCy
A:g:GROUP@:rwaDxtTnNcy
A::EVERYONE@:rxtncy

cluster::> vserver security file-directory show -vserver vs0 -path /unix

                Vserver: vs0
              File Path: /unix
         Security Style: unix
```

```
        Effective Style: unix
         DOS Attributes: 10
 DOS Attributes in Text: ----D---
Expanded Dos Attributes: -
           Unix User Id: 0
          Unix Group Id: 1
         Unix Mode Bits: 775
 Unix Mode Bits in Text: rwxrwxr-x
                   ACLs: NFSV4 Security Descriptor
                         Control:0x8014
                         DACL - ACEs
                           ALLOW-S-1-8-55-0x16019d
                           ALLOW-S-1-520-0-0x1601ff
                           ALLOW-S-1-520-1-0x1201ff-IG
                           ALLOW-S-1-520-2-0x1200a9
```

To see the translated ACLs, use fsecurity from node shell on the node that owns the volume:

```
cluster::> node run -node node2 fsecurity show /vol/unix

[/vol/unix - Directory (inum 64)]
  Security style: Unix
  Effective style: Unix

  DOS attributes: 0x0010 (----D---)

  Unix security:
    uid: 0
    gid: 1
    mode: 0775 (rwxrwxr-x)

  NFSv4 security descriptor:
    DACL:
      Allow - uid: 55 - 0x0016019d
      Allow - OWNER@ - 0x001601ff
      Allow - GROUP@ - 0x001201ff
      Allow - EVERYONE@ - 0x001200a9 (Read and Execute)
    SACL:
      No entries.
```

When a change is made to the mode bit when NFSv4 ACLs are present, the NFSv4 ACL that was just set will get wiped by default:

```
[root@centos6 /]# chmod 755 /unix
[root@centos6 /]# ls -la | grep unix
drwxr-xr-x.   2 root     daemon         4096 Apr 30 11:24 unix
[root@centos6 /]# nfs4_getfacl /unix
A::OWNER@:rwaDxtTnNcCy
A:g:GROUP@:rxtncy
A::EVERYONE@:rxtncy

cluster::> node run -node node2 fsecurity show /vol/unix

[/vol/unix - Directory (inum 64)]
  Security style: Unix
  Effective style: Unix

  DOS attributes: 0x0010 (----D---)

  Unix security:
    uid: 0
    gid: 1
    mode: 0755 (rwxr-xr-x)

  No security descriptor available.
```

To control this behavior in clustered Data ONTAP, use the following diag-level option:

```
cluster::> set diag
cluster::*> nfs server modify -vserver vs0 -v4-acl-preserve [enabled|disabled]
```

Once the option is enabled, the ACL will stay intact when mode bits are set.

```
[root@centos6 /]# nfs4_setfacl -a A::ldapuser@nfsv4domain.netapp.com:ratTnNcCy /unix
[root@centos6 /]# ls -la | grep unix
drwxr-xr-x.   2 root     daemon         4096 Apr 30 11:24 unix
[root@centos6 /]# nfs4_getfacl /unix
A::ldapuser@nfsv4domain.netapp.com:ratTnNcCy
A::OWNER@:rwaDxtTnNcCy
A:g:GROUP@:rxtncy
A::EVERYONE@:rxtncy

cluster::> vserver security file-directory show -vserver vs0 -path /unix

               Vserver: vs0
             File Path: /unix
         Security Style: unix
        Effective Style: unix
         DOS Attributes: 10
 DOS Attributes in Text: ----D---
Expanded Dos Attributes: -
           Unix User Id: 0
          Unix Group Id: 1
         Unix Mode Bits: 755
 Unix Mode Bits in Text: rwxr-xr-x
                   ACLs: NFSV4 Security Descriptor
                         Control:0x8014
                         DACL - ACEs
                           ALLOW-S-1-8-55-0x16019d
                           ALLOW-S-1-520-0-0x1601ff
                           ALLOW-S-1-520-1-0x1200a9-IG
                           ALLOW-S-1-520-2-0x1200a9

cluster::> node run -node node2 fsecurity show /vol/unix

[/vol/unix - Directory (inum 64)]
  Security style: Unix
  Effective style: Unix

  DOS attributes: 0x0010 (----D---)

  Unix security:
    uid: 0
    gid: 1
    mode: 0755 (rwxr-xr-x)

  NFSv4 security descriptor:
    DACL:
      Allow - uid: 55 - 0x0016019d
      Allow - OWNER@ - 0x001601ff
      Allow - GROUP@ - 0x001200a9 (Read and Execute)
      Allow - EVERYONE@ - 0x001200a9 (Read and Execute)
    SACL:
      No entries.
```

Note that the ACL is still intact after mode bits get set:

```
[root@centos6 /]# chmod 777 /unix
[root@centos6 /]# ls -la | grep unix
drwxrwxrwx.   2 root     daemon         4096 Apr 30 11:24 unix
[root@centos6 /]# nfs4_getfacl /unix
A::ldapuser@win2k8.ngslabs.netapp.com:ratTnNcCy
A::OWNER@:rwaDxtTnNcCy
A:g:GROUP@:rwaDxtTnNcy
A::EVERYONE@:rwaDxtTnNcy

cluster::> vserver security file-directory show -vserver vs0 -path /unix
```

```
               Vserver: vs0
             File Path: /unix
       Security Style: unix
      Effective Style: unix
       DOS Attributes: 10
 DOS Attributes in Text: ----D---
Expanded Dos Attributes: -
         Unix User Id: 0
        Unix Group Id: 1
       Unix Mode Bits: 777
 Unix Mode Bits in Text: rwxrwxrwx
                 ACLs: NFSV4 Security Descriptor
                       Control:0x8014
                       DACL - ACEs
                         ALLOW-S-1-8-55-0x16019d
                         ALLOW-S-1-520-0-0x1601ff
                         ALLOW-S-1-520-1-0x1201ff-IG
                         ALLOW-S-1-520-2-0x1201ff

cm6080-rtp2::*> node run -node node2 fsecurity show /vol/unix

[/vol/unix - Directory (inum 64)]
  Security style: Unix
  Effective style: Unix

  DOS attributes: 0x0010 (----D---)

  Unix security:
    uid: 0
    gid: 1
    mode: 0777 (rwxrwxrwx)

  NFSv4 security descriptor:
    DACL:
      Allow - uid: 55 - 0x0016019d
      Allow - OWNER@ - 0x001601ff
      Allow - GROUP@ - 0x001201ff
      Allow - EVERYONE@ - 0x001201ff
    SACL:
      No entries.
```

NFSv4 Delegations

NFSv4 introduces the concept of delegations that provide an aggressive cache, which is different from the ad hoc caching that NFSv3 provides. There are two forms of delegations—read and write. Delegations provide more cache correctness rather than improving performance. For delegations to work, a supported UNIX client is required along with the right delegation options enabled on the NetApp controller. These options are disabled by default. When a server determines to delegate a complete file or part of the file to the client, the client caches it locally and avoids additional RPC calls to the server. This reduces a lot of GETATTR calls in case of read delegations. Reads can be delegated to numerous clients, but writes can be delegated only to one client at a time. The server reserves the right to recall the delegation for any valid reason. The server determines to delegate the file under two scenarios—a confirmed call-back path from the client that the server uses to recall a delegation if needed and when the client sends an OPEN function for a file.

Why Use Read or Write Delegations?

Delegations can be used to improve the read and write performance of certain applications. For example, web applications that have numerous readers of one or more files on the same client and across clients that also generate copious amounts of metadata operations like GETATTRs and LOOKUPs could request read delegations from the NetApp controller for local access to improve performance and response time. Delegating the whole file or certain ranges of bytes to the client's local memory avoids additional RPC calls over the wire for metadata operations. If the file or byte offset is rewritten during the delegation, the

delegation is recalled. Although this is necessary to acquire updates, the delegation recall can affect read performance. Therefore, write delegations are typically granted for single writer applications. Read and write delegations can improve I/O performance, but that depends on the client hardware and operating system. For instance, low-memory client platforms would not handle delegations very well.

In 7-Mode, read and write delegations are set using the following commands:

```
option.nfs.v4.read_delegation on
```

```
option.nfs.v4.write_delegation on
```

In clustered Data ONTAP, read or write delegations can be set during the creation of the NFS server or when modifying an existing NFS server. There is no production impact when enabling delegations on an existing NFS server other than the features delegations bring.

Enabling or Disabling NFSv4 Read File Delegations

Goal	How to
Enable read file delegations	`vserver nfs modify -vserver vserver_name –v4.0-read-delegation enabled`
Disable read file delegations	`vserver nfs modify -vserver vserver_name -v4.0-read-delegation disabled`

Enabling or Disabling NFSv4 Write File Delegations

Goal	How to
Enable write file delegations	`vserver nfs modify -vserver vserver_name –v4.0-write-delegation enabled`
Disable write file delegations	`vserver nfs modify -vserver vserver_name -v4.0-write-delegation disabled`

Note: Both the file read and write delegation options take effect as soon as they are changed. There is no need to reboot or restart NFS.

NFSv4 Locking

For NFSv4 clients, Data ONTAP supports the NFSv4 file-locking mechanism, maintaining the state of all file locks under a lease-based model. In accordance with RFC 3530, Data ONTAP "defines a single lease period for all state held by an NFS client. If the client does not renew its lease within the defined period, all state associated with the client's lease may be released by the server." The client can renew its lease explicitly or implicitly by performing an operation, such as reading a file. Furthermore, Data ONTAP defines a grace period, which is a period of special processing in which clients attempt to reclaim their locking state during a server recovery.

Locks are issued by Data ONTAP to the clients on a lease basis. The server checks the lease on each client every 30 seconds. In the case of a client reboot, the client can reclaim all the valid locks from the server once it has restarted. If a server reboots, then upon restarting it does not issue any new locks to the clients for a grace period of 45 seconds (tunable in clustered Data ONTAP to a maximum of 90 seconds), after which the locks are issued to the requesting clients. The lease time of 30 seconds can be tuned based on the application requirements.

Table 5) NFS lease and grace periods.

Term	Definition (see RFC 3530 for more information)
Lease	The time period in which Data ONTAP irrevocably grants a lock to a client
Grace period	The time period in which clients attempt to reclaim their locking state from Data ONTAP during server recovery

Specifying the NFSv4 Locking Lease Period

To specify the NFSv4 locking lease period (the time period in which Data ONTAP irrevocably grants a lock to a client), you can modify the `-v4-lease-seconds` option. By default, this option is set to 30. The minimum value for this option is 10. The maximum value for this option is the locking grace period, which you can set with the `locking.lease_seconds` option.

NFSv4.x Referrals

Clustered Data ONTAP 8.1 introduced NFSv4.x referrals. A referral directs a client to another LIF in the SVM. The NFSv4.x client uses this referral to direct its access over the referred path to the target LIF from that point forward. Referrals are issued when there is a LIF in the SVM that resides on the cluster node where the data volume resides. In other words, if a cluster node receives an NFSv4.x request for a nonlocal volume, it is able to refer the client to the local path for that volume by means of the LIF. This allows clients faster access to the data via a direct path and avoids extra traffic on the cluster network.

How they work

When a mount request is sent, the request will act as a normal NFSv4.x mount operation. However, once the DH LOOKUP call is made, the server (NetApp cluster) will respond with the GETFH status of “NFS4ERR_MOVED” to notify the client that the volume being accessed does not live where the LIF being requested lives. The server will then send a LOOKUP call to the client, notifying it of the IP (via the fs_location4 value) on the node where the data volume lives. This works regardless of whether a client is mounting via DNS name or IP. However, the client will report that it is mounted to the IP specified rather than the IP returned to the client from the server.

For example:

The data volume lives on node1:

```
cluster::> volume show -vserver vs0 -volume nfsvol -fields node
vserver volume node
------- ------ -------------
vs0     nfsvol node1
```

The data LIF lives on node2:

```
cluster::> net int show -vserver vs0 -lif data2 -fields curr-node,home-node
  (network interface show)
vserver lif   home-node     curr-node     address
------- ----- ------------- ------------- -----------
vs0     data2 node2         node2         10.61.92.37
```

There is also a data LIF on node1:

```
cluster::> net int show -vserver vs0 -curr-node node1 -role data
  (network interface show)
            Logical    Status     Network            Current       Current Is
Vserver     Interface  Admin/Oper Address/Mask       Node          Port    Home
----------- ---------- ---------- ------------------ ------------- ------- ----
vs0
            data1         up/up    10.61.92.34/24     node1         e0a     true
```

The client makes a mount request to the data LIF on node2, at the IP address 10.61.92.37:

```
[root@centos6 /]# mount -t nfs4 10.61.92.37:/nfsvol /mnt
```

The mount location looks to be at the IP address specified by the client:

```
[root@centos6 /]# mount | grep /mnt
10.61.92.37:/nfsvol on /mnt type nfs4 (rw,addr=10.61.92.37,clientaddr=10.61.179.164)
```

But the cluster shows that the connection was actually established to node1, where the data volume lives. No connection was made to node2:

```
cluster::> network connections active show -node node1 -service nfs*
             Vserver   Interface         Remote
      CID Ctx Name     Name:Local Port   Host:Port            Protocol/Service
--------- --- -------- ----------------- -------------------- ----------------
Node: node1
286571835   6 vs0          data:2049         10.61.179.164:763    TCP/nfs

cluster::> network connections active show -node node2 -service nfs*
There are no entries matching your query.
```

Because clients might become “confused” about which IP address they are actually connected to as per the `mount` command, NetApp recommends using host names in mount operations.

Best Practice

NetApp highly recommends that there be at least one data LIF per node per SVM so that a local path is always available to data volumes. NetApp also recommends leveraging some form of DNS load balancing so that all LIFs in an SVM are being used equally in NFS requests. On-box DNS load balancing is one option available in clustered Data ONTAP and is covered in depth in the "Clustered Data ONTAP Networking Best Practice Guide" (TR-4847).

If a volume moves to another aggregate on another node, the NFSv4.x clients must unmount and remount the file system manually in order to be referred to the new location of the volume. This provides a direct data path for the client to reach the volume in its new location. If the manual mount/unmount process is not done, the client can still access the volume in its new location, but I/O requests would then take a remote path. NFSv4.x referrals are enabled by default on newer Linux clients, such as RHEL 5.4 and later releases.

If a volume is junctioned below other volumes, the referral will use the volume being mounted to refer as the local volume. For example:

- A client wants to mount vol2
- Vol2's junction is /vol1/vol2
- Vol1 lives on node1; vol2 lives on node2
- A mount is made to cluster:/vol1/vol2
- The referral will return the IP address of a LIF that lives on node2, regardless of what IP address is returned from DNS for the hostname "cluster"
- The mount will use the LIF local to vol2 on node2

In a mixed client environment, if any of the clients do not support referrals, then the `-v4.0-referrals` option should not be enabled. If the option is enabled and clients that do not support referrals get a referral from the server, that client will be unable to access the volume and will experience failures. See RFC 3530 for more details on referrals.

Table 6) Configuring NFSv4.x referrals.

Category	Commands
Configure NFSv4.x referrals.	To enable referrals on an SVM requires advanced privilege.

	`cluster::> set advanced`
	
	`Warning: These advanced commands are potentially dangerous; use them only when directed to do so by NetApp personnel.`
	`Do you want to continue? {y|n}: y`
	
	`For NFSv4.0:`
	`cluster::*> vserver nfs modify -vserver test_vs1 -v4.0-referrals enabled -v4-fsid-change enabled`
	
	`For NFSv4.1:`
	`cluster::*> vserver nfs modify -vserver test_vs1 -v4.1-referrals enabled -v4-fsid-change enabled`

	Verification

	`cluster::*> vserver nfs show -vserver test_vs1 -fields v4.0-referrals,v4-fsid-change`
	`Vserver  v4-fsid-change v4.0-referrals`
	`-------- -------------- --------------`
	`test_vs1 enabled        enabled`
	
	`cluster::*> vserver nfs show -vserver test_vs1 -fields v4.1-referrals,v4-fsid-change`
	`Vserver  v4-fsid-change v4.1-referrals`
	`-------- -------------- --------------`

	test_vs1 enabled enabled

Refer to Table 17 in the “NFSv4 Option Changes in Clustered Data ONTAP” section for more information.

NFSv4.x Fastpath in Clustered Data ONTAP 8.2.x

Starting in clustered Data ONTAP 8.2, NFS fastpath was introduced to potentially improve NFSv4 performance for READs and WRITES by bypassing the internal processing of NFSv4 packets into clustered Data ONTAP–centric packets when the data request is made on a LIF that is local to the node hosting the volume. When combined with other features like pNFS or referrals, localized data can be guaranteed for each READ and WRITE request, thus allowing consistent use of the NFSv4 fastpath. NFSv3 has always had an NFS fastpath concept. NFS fastpath is enabled by default.

NFSv4.x Multithreaded Operations in Clustered Data ONTAP 8.2.x

Starting in clustered Data ONTAP 8.2, NFSv4 operations will now be able to leverage numerous CPUs, allowing NFSv4 to use parallel processing to help improve performance. The performance improvement will depend on a variety of factors, including number of clients, existing non-NFS load on the system, number of CPUs, volume layout across the nodes, and number of data LIFs in an SVM. Workload types can also be a determining factor in performance. NFSv4 multithreading is available by default on all clustered Data ONTAP systems.

NFSv4.x Stateless Migration

NFSv4 referrals also brought NFSv4 stateless migration support in clustered Data ONTAP 8.1. Stateless migration is also in clustered Data ONTAP 8.2 and will include support only for Oracle® dNFS. Stateful migration is not presently available in clustered Data ONTAP.

Migration is an NFSv4.x feature that allows a file system to move from one server to another without client disruption. Migration enablement requires enabling referrals and the option `-v4-fsid-change` on the NFS server. Migration is a diag-level option. Enabling migration assumes the following about the solution:

- All clients accessing the NFSv4.x server on the SVM are stateless.
- All clients accessing the NFSv4.x server on the SVM support migrations.
- The NFSv4.x clients **do not** use the following:
  - Locking
  - Share reservations
  - Delegations
  - OPEN for file access
- The NFSv4.x clients **do** use the following:
  - READ, WRITE, and SETATTR with special stateid of all bits 0
  - OPEN only to create a file and close it right away
- The NFSv4.x clients do not have a state established on the NFS server.

NFS migration support can be useful in the following scenarios in clustered Data ONTAP:

- Volume moves
- LIF migration/failover

NFSv4.x Snapshots

In previous versions of NFS (v2/v3), the .snapshot directory was visible to clients. This was exposed at the mount point and was visible at every directory. However, because NFSv4.x does not use the MOUNT protocol, the .snapshot directory is not visible, but is accessible from anywhere in the NFSv4.x mount. To access snapshots via NFSv4.x, simply navigate to the .snapshot directory manually.

```
For example:
[root@centos6 ~]# mount -t nfs4 10.61.92.37:/nfsvol /mnt
[root@centos6 ~]# cd /mnt
[root@centos6 mnt]# ls -la | grep snapshot
[root@centos6 mnt]# cd .snapshot
[root@centos6 .snapshot]# ls -la
drwxrwxrwx. 12 root     root          4096 Apr 25 16:05 .
drwxrwxrwx.  3 root     root        106496 Apr 24 16:01 ..
drwxrwxrwx.  4 root     root        106496 Apr 18 14:50 daily.2013-04-24_0010
drwxrwxrwx.  2 root     root        106496 Mar 12 19:54 weekly.2013-04-14_0015
drwxrwxrwx.  4 root     root        106496 Apr 18 14:50 weekly.2013-04-21_0015
```

4.2 NFSv4.1

NFSv4.1 support began in clustered Data ONTAP 8.1. NFSv4.1 is considered a minor version of NFSv4. Even though the NFSv4.1 RFC 5661 suggests that directory delegations and session trunking are available, there is currently no client support, nor is there currently support in clustered Data ONTAP.

Parallel NFS (pNFS) is a major addition to NFSv4.1. By default NFSv4.1 is disabled. It can be enabled by specifying the `-v4.1` option and setting it to enabled when creating an NFS server on an SVM. NFSv4.0 support must also be enabled in order to enable NFSv4.1 support.

Table 7) Enabling NFSv4.1.

Category	Commands
Enable NFSv4.1.	
	`cluster::> vserver nfs modify -vserver test_vs1 -v4.0 enabled -v4.1 enabled`
	Verification – Note v4.0 and v4.1 are both enabled.

	`cluster::> vserver nfs show -vserver test_vs1 -fields v4.0,v4.1`
	`Vserver  v4.0    v4.1`
	`-------- ------- -------`
	`test_vs1 enabled enabled`

Parallel Network File System (pNFS)

Parallel NFS (pNFS) is a new part of NFS version 4.1 standards. NFSv4.1, which follows Request for Comments (RFC) 5661, is a minor release of NFSv4. NFSv4.1 does not modify any NFSv4 features and functionalities. With traditional NFS versions 3, 4, and 4.1, the metadata and data shared the same I/O path. With pNFS, there is now an NFS feature that handles metadata and data on different I/O paths. A metadata server handles all the metadata activities from the client, while the data servers provide a direct path for data access. As explained in RFC 5661, "Parallel data access is controlled by recallable objects known as 'layouts,' which are integrated into the protocol locking model. Clients direct requests for data access to a set of data servers specified by the layout via a data storage protocol which may be NFSv4.1 or may be another protocol."

pNFS support began in clustered Data ONTAP 8.1 for files only and continues with enhancements in clustered Data ONTAP 8.2. There is no Data ONTAP 7G/7-Mode support for pNFS. Current client support for pNFS is very limited, but NetApp does support all clients that support pNFS and follow the RFC specifications. By default the pNFS option is enabled, but it is only active if NFSv4.0 and NFSv4.1 support also are enabled. NetApp does not currently recommend pNFS for metadata-heavy workloads.

pNFS requires a client that also supports pNFS. Currently, RHEL 6.4 is the only commercial Linux distribution that has full pNFS support.

How pNFS Works

pNFS defines the notion of a device, which is generated by the server (NetApp NFS server) and sent to the client. This helps the client locate the data and send requests directly over the path local to that data. Data ONTAP generates one pNFS device per flexible volume. The metadata path will not change, so metadata requests might still be remote. In clustered Data ONTAP's pNFS implementation, every data LIF is considered an NFS server, so NetApp strongly recommends that each node owns at least one data LIF per NFS SVM.

The device contains information about the following:

- Volume constituents
- Network location of the constituents

The device information is cached to the local node's NAS protocol stack for improved performance.

To see pNFS devices in the cluster, use the following diag-level command:

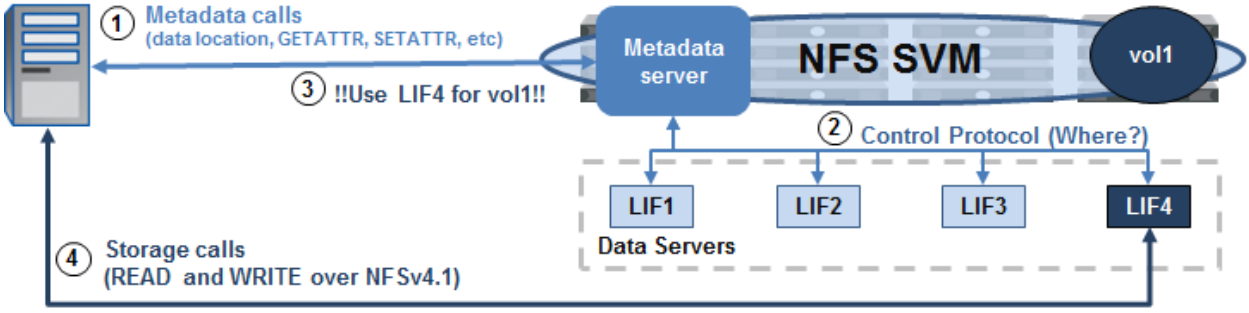
```
cluster::> set diag
cluster::*> vserver nfs pnfs devices cache show
```

There are three main components of pNFS:

- Metadata server
  - Handles all nondata traffic such as GETATTR, SETATTR, and so on
  - Responsible for maintaining metadata that informs the clients of the file locations
  - Located on the NetApp NFS server
- Data server
  - Stores file data and responds to READ and WRITE requests
  - Located on the NetApp NFS server
  - Inode information will also reside here
- Clients

These components will leverage three different protocols. The **control protocol** will be the way the metadata and data servers stay in sync. The **pNFS protocol** is used between clients and the metadata server. pNFS supports file-, block-, and object-based **storage protocols**, but NetApp currently only supports file-based pNFS.

Figure 2) pNFS data workflow.



1. The client makes a data request to the cluster.
2. The metadata server works to find the location of the data if the location is not already cached.
3. The location of the data is returned to the client via the control path.
4. The client begins operations over the specified data LIF returned from the metadata server.

NFSv4.1 Delegations

In clustered Data ONTAP 8.2, support for NFSv4.1 delegations was added. NFSv4.1 delegations are very similar to NFSv4.0 delegations, but are part of the v4.1 protocol rather than v4.0. Below is a table that covers the new additions to NFSv4.1 and how they benefit an environment over NFSv4.0. These additions are covered in detail in RFC 5661.

Table 8) NFSv4.1 delegation benefits.

NFSv4.1 Delegation Feature	Benefit Versus NFSv4.0 Delegation

NFSv4.1 Delegation Feature	Benefit Versus NFSv4.0 Delegation
EXCHANGE_ID is used	In NFSv4.0, SETCLIENTID was used. EXCHANGE_ID replaces SETCLIENTID and enables a client ID to be assigned before any other client operations take place. As per RFC 5661, "The only NFSv4.1 operations possible before a client ID is established are those needed to establish the client ID."
Callbacks use the same TCP connection as the forechannel	In NFSv4.0, callbacks use different TCP connections than the forechannel. Using the same TCP connection for callbacks provides better performance for delegations and is more firewall friendly.
New OPEN request options: • OPEN4_SHARE_ACCESS_WANT_DELEG_MASK • OPEN4_SHARE_ACCESS_WANT_NO_PREFERENCE • OPEN4_SHARE_ACCESS_WANT_READ_DELEG • OPEN4_SHARE_ACCESS_WANT_WRITE_DELEG • OPEN4_SHARE_ACCESS_WANT_ANY_DELEG • OPEN4_SHARE_ACCESS_WANT_NO_DELEG	NFSv4.1 provides more precise control to clients for acquisition of delegations than NFSv4.0. These new options enable more OPEN scenarios to be covered to prevent problems issuing or reclaiming delegations.

For information regarding pNFS with RHEL 6.4, see TR-4063: Parallel Network File System Configuration and Best Practices for Clustered Data ONTAP.

4.3 NFS Auditing

NFS auditing is new in clustered Data ONTAP 8.2. In 7-Mode, NFS auditing required CIFS to function properly. That is no longer the case in clustered Data ONTAP. NFS auditing can now be set up independently and does not require a CIFS license.

The following section covers the setup and use of NFS auditing.

NFS Audit Setup

The use of NFS auditing does not require CIFS, but does require the use of NFSv4.x ACLs. Therefore, this option must be enabled on the SVM, along with NFSv4.x. This is because of the need to set an AUDIT type ACE on the file or directory to enable NFS auditing. After the AUDIT ACE is set, auditing will take place for NFSv3 and NFSv4.x operations.

Enabling Auditing on Clustered Data ONTAP System

To enable NFSv4.x and NFSv4.x ACLs, see the sections on NFSv4.x and NFS ACLs.

After NFSv4.x and NFSv4.x ACLs are enabled, enable NFS auditing with the following command:

```
::> vserver audit create -vserver nfs -destination /unix -rotate-size 100MB -rotate-limit 0
```

This command will enable auditing for NFS and CIFS access on the junction path "/unix" for the SVM named "nfs."

After auditing is enabled on the clustered Data ONTAP system, the AUDIT ACEs should be created.

Creating NFSv4 AUDIT ACEs

To create an NFSv4 AUDIT ACE, mount the volume on which auditing was enabled using NFSv4.x. After the volume is mounted, create an AUDIT ACE on the volume, files, and/or directories where auditing is required.

An AUDIT ACE can be used to track ALLOW or DENY for a variety of operations, including:

- Read
- Write
- Execute
- Append
- Delete

For information on all of the ACE permissions in NFSv4, see http://linux.die.net/man/5/nfs4_acl.

Each Linux client will use a different method of assigning NFSv4.x ACEs. In RHEL/CentOS/Fedora, the commands `nfs4_setacl` and `nfs4_getacl` are used.

An AUDIT ACE will leverage flags to specify if auditing should be for successes, failures, or both. AUDIT ACEs will use the ACE type of U.

Example of Setting an AUDIT ACE

```
# nfs4_setfacl -a  U:SF:ldapuser@domain.netapp.com:rwatTnNcCy /mnt
```

After the AUDIT ACE is applied and the user that is being audited attempts access, the events will get logged to an XML file on the volume.

Example of an Audit Event Logged

```
- <Event>
- <System>
  <Provider Name="Netapp-Security-Auditing" />
  <EventID>4663</EventID>
  <EventName>Get Object Attributes</EventName>
  <Version>1</Version>
  <Source>NFSv3</Source>
  <Level>0</Level>
  <Opcode>0</Opcode>
  <Keywords>0x8020000000000000</Keywords>
  <Result>Audit Success</Result>
  <TimeCreated SystemTime="2013-08-08T20:36:05.011243000Z" />
  <Correlation />
  <Channel>Security</Channel>
  <Computer>e284de25-3edc-11e2-92d0-123478563412/525c9a2c-dce2-11e2-b94f-123478563412</Computer>
  <Security />
  </System>
- <EventData>
  <Data Name="SubjectIP" IPVersion="4">10.61.179.150</Data>
  <Data Name="SubjectUnix" Uid="10000" Gid="503" Local="false" />
  <Data Name="ObjectServer">Security</Data>
  <Data Name="ObjectType">Directory</Data>
  <Data Name="HandleID">00000000000453;00;00000040;3a2cada4</Data>
  <Data Name="ObjectName">/</Data>
  <Data Name="InformationRequested">File Type; File Size; Last Accessed Time; Last Metadata
Modfied Time; Last Modified Time; Unix Mode; Unix Owner; Unix Group;</Data>
  </EventData>
```

```
</Event>
```

4.4 NFS on Windows

It is possible to use NFS with clustered Data ONTAP systems on Windows operating systems either by installing the native Windows NFS tools (such as Services for NFS in Windows 2008 and later) or by third-party tools, such as Hummingbird/OpenText NFS client, Cygwin, and so on. However, locking with NFSv3 is not currently supported in clustered Data ONTAP. To use NFS on Windows, either use NFSv4.x or disable the NLM portion of the NFS client.

Use of NFSv4.x will require interaction with an LDAP server for ID to name mapping, as well as a valid NFSv4 ID domain.

PCNFS, WebNFS, and HCLNFS are not supported with clustered Data ONTAP storage systems.

4.5 NFS Using Apple OS

NFS mounts are also possible using Apple® OS via the Finder or terminal windows. For complete mount options in the Apple OS, use the `man mount_nfs` command in a terminal window. When using Apple clients for NFS, there are some things to keep in mind.

Apple OS Disables Root by Default

Apple disables the root user (UID 0) by default in its OS. Users are instead required to log in with a user name other than root and use sudo if performing root-level tasks. It is possible to reenable the root user.

Apple UIDs Start at 501

The Apple UID structure starts at UID 501. This UID is not a default UNIX user in clustered Data ONTAP, nor does it exist in most name service servers. This happens for every Apple OS client in an environment, so it is possible that multiple users will exist with the same UID. The options to handle this are as follows:

- Create a user on the cluster or in a name service server with UID 501 to authenticate all Apple users.
- Change the UID on the Apple OS for each user who intends to use NFS on Apple.

Use of Apple NFS with NTFS Security Style Volumes

Apple NFS handles NTFS security style volumes differently than Linux NFS clients. Therefore, copies/writes to an NFS mount via Finder applications will fail by default when NTFS security style is used. This issue occurs when the Apple client attempts an EXCLUSIVE CREATE operation on the file, which is only allowed by SMB clients in clustered Data ONTAP.

As a workaround, the NFS server option –ntfs-unix-security-ops can be set to ignore to allow NTFS security style volumes to work properly with NFS mounts on Apple. See bug 723115 for more information.

NFS Rootonly Operations Do Not Work as Expected with Apple OS*

In clustered Data ONTAP 8.2, the NFS server options `-mount-rootonly` and `-nfs-rootonly` were introduced. By default, mount-rootonly is enabled, and nfs-rootonly is disabled. Apple OS behavior when mounting via NFS defaults to always use reserved ports for the MOUNT protocol and nonreserved ports for the NFS protocol. The Linux NFS mount option of resvport/noresvport applies in the Apple OS, but noresvport does not control the client's MOUNT port sent to the cluster. Therefore, Apple NFS clients will always use ports in range <1024 for MOUNT. There presently is not a known method to change this behavior, so Apple technical support would need to be engaged to use nonreserved ports for NFS MOUNT calls. For NFSv4.x mounts, this does not matter, because NFSv4.x does not leverage the MOUNT protocol. NFS client ports for NFS operations (port 2049) can be controlled using the resvport/noresvport mount options, but the NFS server option on the cluster would need to be toggled to honor the client behavior. This would affect all versions of NFS.

Additionally, when attempting to mount with the resvport option specified in the Apple OS, the sudo command would need to be used, because root is disabled and the -users option is not specified by default.

*When using the Finder to mount NFS, mount options cannot be specified.

5 Multiprotocol User Mapping

Multiprotocol functionality includes the ability to map UNIX user identities (UIDs) to NT identities (SIDs). This mapping involves contacting an NT domain controller to do name-to-SID lookups. Because this translation is time consuming and must be performed for every NFS access of a file with NT security, these mappings are cached. In clustered Data ONTAP, credentials are cached in two locations: the NAS protocol stack and the Security Daemon (SecD).

NAS Protocol Caching

The NAS protocol stack is unique per node and handles the translation of NAS protocol packets into cluster-aware packets to be passed through the cluster network on to the WAFL® file system. The NAS protocol stack credential cache did not age out prior to clustered Data ONTAP 8.2, but now ages out every 20 minutes so that stale credentials are not kept on the system. The NAS credential cache can be viewed and flushed manually via diag-level commands. Keep in mind that to flush a NAS cache for a specific node one must be logged in to a management interface local to that node (such as the node management LIF). NAS protocol caches are flushed as a whole per SVM. Once a credential is flushed, it must be repopulated into cache, which can affect latency on new connections. Existing connections are not affected by flushing this cache. However, NetApp recommends flushing caches only at the direction of NetApp Support.

Note: Diag-level commands must be used with caution.

Example:

```
cluster::> set diag
cluster::*> diag nblade credentials show -vserver vs0 -unix-user-name root
Getting credential handles.
1 handles found....

Getting cred 0 for user.
          Global Virtual Server: 8
          Cred Store Uniquifier: 23
Cifs SuperUser Table Generation: 0
               Locked Ref Count: 0
                     Info Flags: 1
          Alternative Key Count: 0
        Additional Buffer Count: 0
                Allocation Time: 0 ms
                      Hit Count: 0 ms
                   Locked Count: 0 ms
Windows Creds:
        Flags: 0
        Primary Group: S-0-0
Unix Creds:
        Flags: 0
        Domain ID: 0
        Uid: 0
        Gid: 1
        Additional Gids:

cluster::*> diag nblade credentials flush -vserver vs0
FlushCredStore succeeded flushing 2 entries
```

SecD Caching

SecD is a user space application that runs on a per-node basis. The SecD application handles name service lookups such as DNS, NIS, and LDAP, as well as credential queries, caching, and name mapping. Because SecD is responsible for so many functions, caching plays an important role in its operations. SecD contains two types of caches: LRU and DB style.

LRU-Style Caches

LRU caches are “Least Recently Used” cache types and will age out individual entries at a specified timeout value based on how long it has been since the entry was last accessed. LRU cache timeout values are viewable and configurable via diag-level commands in the cluster.

In the following example, the “sid-to-name” cache (responsible for Windows SID to UNIX user name caching) allows a default of 2,500 max entries, which will stay in cache for 86.400 seconds:

```
cluster::> set diag
cluster::*> diag secd cache show-config –node node1 –cache-name sid-to-name
Current Entries: 0
    Max Entries: 2500
 Entry Lifetime: 86400
```

Caches can be manually flushed, but can only be flushed one at a time on a per-SVM basis:

```
cluster::> set diag
cluster::*> diag secd cache clear –node node1 –vserver vs0 –cache-name sid-to-name
```

DB-Style Caches

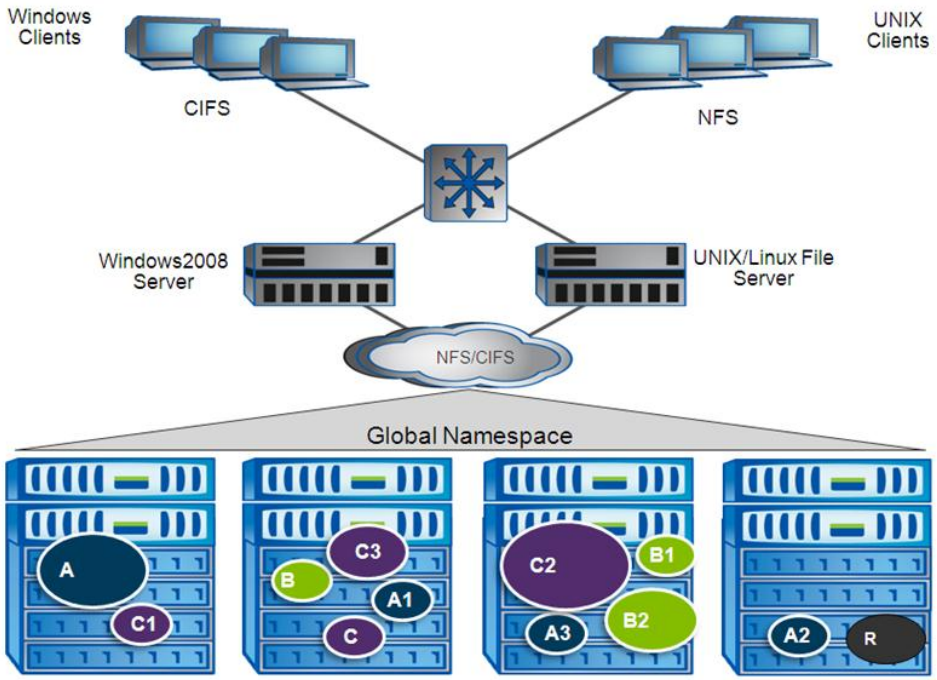
DB-style caches are caches that time out as a whole. These caches do not have maximum entries configured and are rarer than LRU-style caches.

Caches can be flushed in their entirety rather than per node, but both methods involve disrupting the node. One way is to reboot the node via storage failover/giveback. The other method is to restart the SecdD process via the following diag-level command:

```
cluster::> set diag
cluster::*> diag secd restart –node node1
```

NetApp does not recommend adjusting SecD caches unless directed by NetApp Support.

Figure 3) Multiprotocol user mapping.



5.1 User Name Mapping During Multiprotocol Access

Data ONTAP performs a number of steps when attempting to map user names. Name mapping can take place for one of two reasons:

- The user name needs to be mapped to a UID
- The user name needs to be mapped to a Windows SID

Name Mapping Functionality

The method of user mapping will depend on the security style of the volume being accessed. If a volume with UNIX security style is accessed via NFS, then a UID will need to be translated from the user name to determine access. If the volume is NTFS security style, then the UNIX user name will need to map to a Windows user name/SID for NFS requests because the volume will use NTFS-style ACLs. All access decisions will be made by the NetApp device based on credentials, group membership, and permissions on the volume.

By default, NTFS security style volumes are set to 777 permissions, with a UID and GID of 0, which generally translates to the “root” user. NFS clients will see these volumes in NFS mounts with this security setting, but users will not have full access to the mount. The access will be determined by which Windows user the NFS user is mapped to.

The cluster will use the following order of operations to determine the name mapping:

1. 1:1 implicit name mapping
   a. Example: WINDOWS\john maps to UNIX user john implicitly
   b. In the case of LDAP/NIS, this generally is not an issue
2. `Vserver name-mapping` rules
   a. If no 1:1 name mapping exists, SecD checks for name mapping rules
   b. Example: WINDOWS\john maps to UNIX user unixjohn

3. Default Windows/UNIX user
    a. If no 1:1 name mapping and no name mapping rule exist, SecD will check the NFS server for a default Windows user or the CIFS server for a default UNIX user
    b. By default, pcuser is set as the default UNIX user in CIFS servers when created using System Manager 3.0 or `vserver setup`
    c. By default, no default Windows user is set for the NFS server
4. If none of the above exist, then authentication will fail
    a. In most cases in Windows, this manifests as the error “A device attached is not functioning”
    b. In NFS, a failed name mapping will manifest as access or permission denied

Name mapping and name switch sources will depend on the SVM configuration. See the “File Access and Protocols Management Guide” for the specified version of clustered Data ONTAP for configuration details.

Best Practice
It is a best practice to configure an identity management server such as LDAP with Active Directory for large multiprotocol environments.

Table 9) Configuring CIFS for multiprotocol access.

Category	Commands
Add CIFS license.	**Note:** None of the CIFS-related operations can be initiated without adding the CIFS license key.
	`cluster::> license add -license-code XXXXXXXXXXXXXXX`
Enable CIFS.	
	`cluster::> vserver modify -vserver test_vs1 -allowed-protocols nfs,cifs`
	Verification

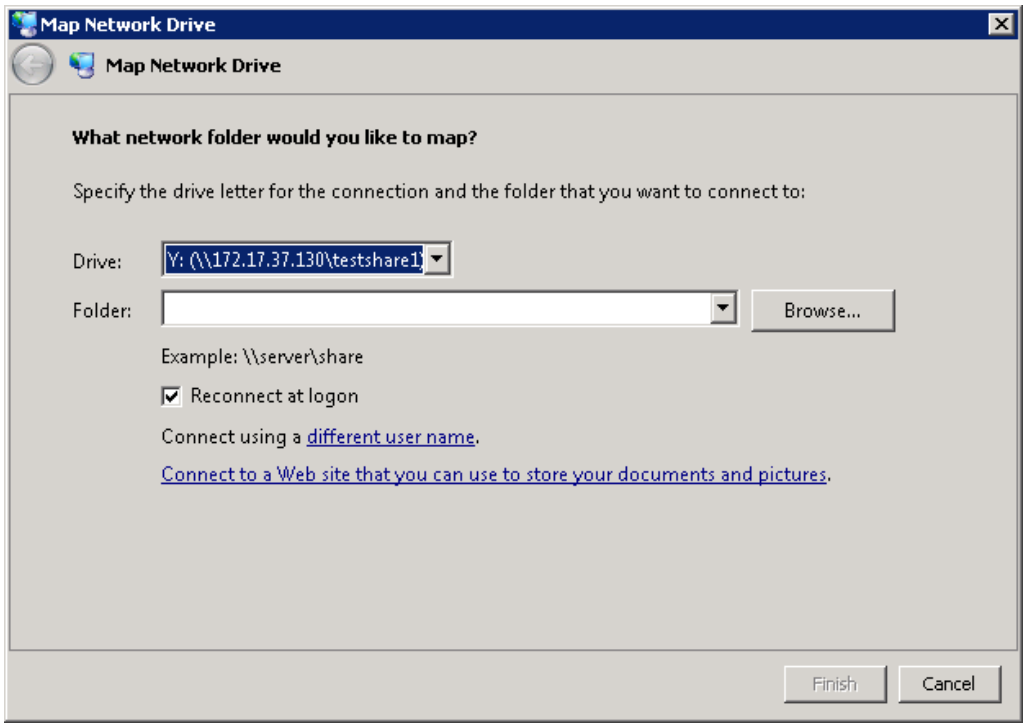
	`cluster::> vserver show -instance -vserver test_vs1`
	
	`vserver: test_vs1`
	`vserver Type: cluster`
	`vserver UUID: 51fdb806-b862-11e0-9980-123478563412`
	`Root Volume: test_vs1`
	`Aggregate: aggr1_Cluster01`
	`Name Service Switch: file, ldap`
	`Name Mapping Switch: file`
	`NIS Domain: -`
	`Root Volume Security Style: unix`
	`LDAP Client: ldapclient1`
	`Language: C`
	`Snapshot Policy: default`
	`Comment:`

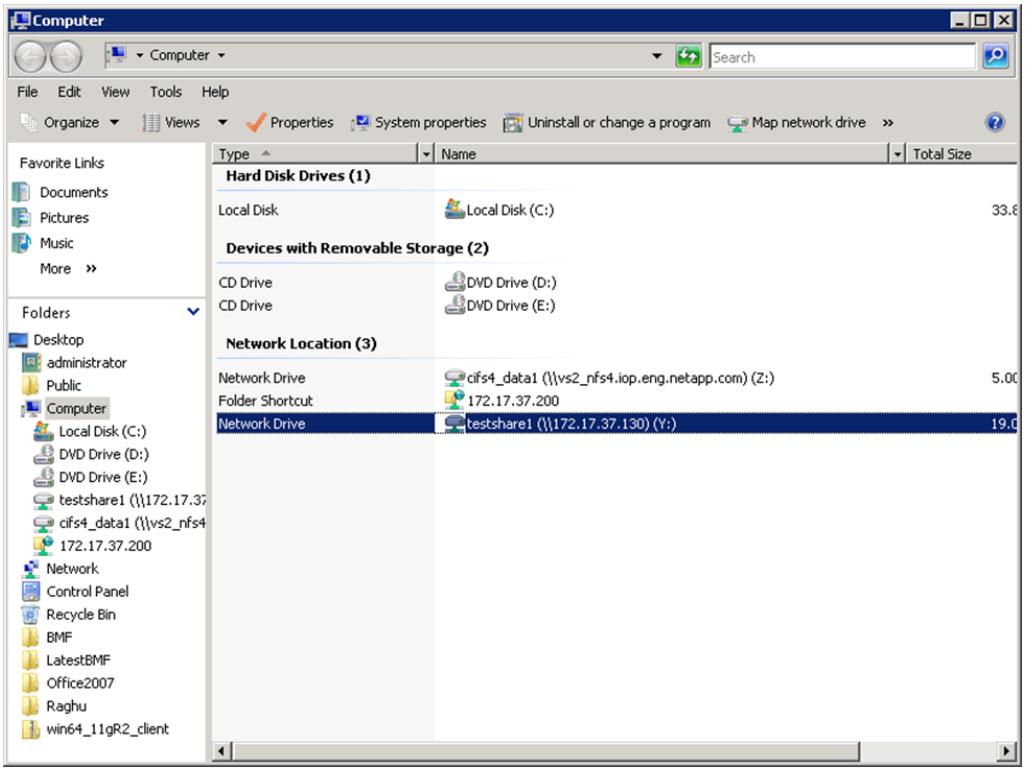
	Anti-Virus On-Access Policy: default Quota Policy: default List of Aggregates Assigned: - Limit on Maximum Number of Volumes allowed: unlimited vserver Admin State: running Allowed Protocols: nfs, cifs Disallowed Protocols: fcp, iscsi
Configure DNS server.	A DNS server must be created and configured properly to provide the name services to resolve the LIF names assigned to the network ports.
	`cluster::> vserver services dns create -vserver test_vs1 -domains domain.netapp.com -state enabled -timeout 2 -attempts 1 -name-servers 172.17.32.100`
	Verification
	`cluster::> vserver services dns show -vserver test_vs1` Vserver: test_vs1 Domains: domain.netapp.com Name Servers: 172.17.32.100 Enable/Disable DNS: enabled Timeout (secs): 2 Maximum Attempts: 1
Create CIFS server.	
	`cluster::> cifs create -vserver test_vs1 -cifs-server test_vs1_cifs -domain domain.netapp.com`
	Verification
	`cluster::> cifs server show` Server Domain/Workgroup Authentication vserver Name Name Style test_vs1 TEST_VS1_CIFS DOMAIN domain
Create CIFS share.	
	`cluster::> cifs share create -vserver test_vs1 -share-name testshare1 -path /testshare1`
	Verification
	`cluster::> vserver cifs share show -vserver test_vs1 -share-name testshare1` vserver: test_vs1 Share: testshare1 CIFS Server NetBIOS Name: TEST_VS1_CIFS

	Path: /testshare1 Share Properties: oplocks browsable changenotify Symlink Properties: - File Mode Creation Mask: - Directory Mode Creation Mask: - Share Comment: - Share ACL: Everyone / Full Control File Attribute Cache Lifetime: -
Make sure that the default UNIX user is set to pcuser.	Make sure that the default UNIX user is set to a valid existing user. In clustered Data ONTAP 8.2 and later, this is set to pcuser by default. Previous versions of clustered Data ONTAP need to be set manually.
	cluster::> cifs options show –vserver test_vs1 vserver: test_vs1 Default Unix User: - ←------- not mapped to pcuser Read Grants Exec: disabled WINS Servers: -
	Create the UNIX group pcuser.
	cluster::> unix-group create -vserver test_vs1 -name pcuser -id 65534 Verification: cluster::> unix-group show -vserver test_vs1 (vserver services unix-group show) vserver Name ID -------------- -------------------- ---------- test_vs1 daemon 1 test_vs1 pcuser 65534 test_vs1 root 0 3 entries were displayed.
	Create the UNIX user pcuser.
	cluster::> unix-user create -vserver test_vs1 -user pcuser -id 65534 -primary-gid 65534 -full-name pcuser Verification: cluster::> unix-user show -vserver test_vs1 (vserver services unix-user show) vserver Name ID -------------- -------------------- ---------- test_vs1 pcuser 65534 test_vs1 root 0 2 entries were displayed.

	Map the default UNIX user to pcuser.

	`cluster::> cifs options modify -vserver test_vs1 -default-unix-user pcuser`
	
	`Verification:`
	
	`cluster::> cifs options show –vserver test_vs1`
	
	`Vserver: test_vs1`
	
	`Default Unix User: pcuser  <-------  mapped to pcuser`
	`Read Grants Exec: disabled`
	`WINS Servers: -`

Attempt to map the CIFS share.	

	
For more information	Before you attempt name mapping, verify that the default UNIX user is mapped to “pcuser.” By default, no UNIX user is associated with the Vserver. For more information, including how to create name mapping rules, see the “File Access and Protocols Management Guide” for the specified version of clustered Data ONTAP.

Using Local Files for Authentication

In clustered Data ONTAP, there is no concept of /etc/passwd, /etc/usermap.cfg or other flat files. Instead, everything is contained within database table entries that are replicated across all nodes in the cluster for consistency and locality.

For local file authentication, users are created and managed at an SVM level for multi-tenancy. For instance, if there are two SVMs in a cluster, both SVMs will have independent UNIX user and group lists. To manage these lists, the commands `vserver services unix-user` and `vserver services unix-group` are leveraged.

These commands control the following:

- User name
- UID/GID
- Group membership (primary and auxiliary)

Users and groups can be either created manually or loaded from URI. For information on the procedure to load from URI, see the File Access and Protocol Guide for the release of clustered Data ONTAP running on the system.

Example of Creating Local UNIX User

```
cluster::> vserver services unix-user create -vserver vs0 -user testuser -id 101 -primary-gid 101
```

Example of Creating Local UNIX Group

```
cluster::> vserver services unix-group create -vserver vs0 -name testgroup -id 101
```

Example of Adding a Local UNIX User to a Local UNIX Group

```
cluster::> vserver services unix-group adduser -vserver vs0 -name testgroup -username testuser
```

Using local users and groups can be beneficial in smaller environments with a handful of users, because the cluster would not need to authenticate to an external source. This prevents latency for lookups, as well as the chance of failed lookups due to failed connections to name servers.

For larger environments, it is recommended to use a name server such as NIS or LDAP to service UID/GID translation requests.

Best Practice
UNIX users will always have primary GIDs. When specifying a primary GID, whether with local users or name services, be sure the primary GID exists in the specified nm-switch and ns-switch locations. Using primary GIDs that do not exist can cause authentication failures in clustered Data ONTAP 8.2 and prior.

Default Local Users

When an SVM is created via vserver setup or System Manager, default local UNIX users and groups are created, along with default UIDs and GIDs.

The following 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
```

Rules to Convert User Mapping Information in 7-Mode in Clustered Data ONTAP

* Name mappings with IP addresses are not supported in clustered Data ONTAP.

Table 10) 7-Mode to clustered Data ONTAP mapping.

7-Mode Mapping	Clustered Data ONTAP			
	-direction	-pattern	-replacement	-position
X => Y	Win-UNIX	X	Y	–
X <= Y	UNIX-Win	Y	X	–
X == Y	UNIX-Win/ Win-UNIX	X/Y	Y/X	–

For further information on CIFS configuration and name mapping, refer to TR-3967: Deployment and Best Practices Guide for Clustered Data ONTAP 8.1 Windows File Services.

5.2 Unified Security Style

Infinite Volumes were introduced in clustered Data ONTAP 8.1.1 with support for NFSv3. Unified security style was introduced in clustered Data ONTAP 8.2 to support CIFS and NFSv4 for Infinite Volumes. Unified security style is intended to provide ubiquitous access control in a multiprotocol environment rather than prioritizing behavior on a particular protocol.

Infinite Volumes use only Unified security style. This style is not currently available for FlexVol® volumes.

For detailed information on Infinite Volumes, see TR-4037: Introduction to NetApp Infinite Volume and TR-4178: Infinite Volume Deployment and Implementation Guide.

What Is Unified Security Style?

Unified security style consolidates file permission management for both UNIX and Windows users and groups. Windows and UNIX users can view and manage permissions on files regardless of the current effective style and regardless of the protocol previously used to set permissions on those files.

UNIX, NTFS, and Mixed Security Styles

Data ONTAP operating in 7-Mode and clustered Data ONTAP support three security styles for FlexVol volumes: UNIX, NTFS, and Mixed. These security styles prioritize the network protocol when managing permissions, but at the expense of other protocols. For example, Windows clients cannot view UNIX-style ACLs, and UNIX clients cannot view NTFS ACLs. In Mixed style, although both UNIX and Windows clients can set ACLs, they are unable to view ACLs set by the other, and when an ACL is set it blindly overwrites the existing permissions. Table 11 describes the behavior and limitations of each security style.

Table 111) Limitations of existing security styles.

Security Style	Limitations
UNIX	• Windows clients cannot set attributes • NTFS-style ACLs are not effective; only NFSv4 ACLs and mode bits are effective • UNIX mode bits can be merged into an NFSv4 ACL
NTFS	• UNIX clients cannot set attributes • Only NTFS-style ACLs are effective; NFSv4 ACLs and mode bits are not effective
Mixed	• Both Windows and UNIX clients can set attributes • UNIX mode bits can be merged into an NFSv4 ACL, but they cannot be merged into an NTFS ACL • Only one style of ACL can be honored on an object – Applying UNIX-style ACLs drops NTFS-style ACLs – Applying NTFS-style ACLs drops UNIX-style ACLs

Note: These limitation apply to all objects in NetApp storage (files, directories, volumes, qtrees, and LUNs).

Unified Security Style Behavior in Clustered Data ONTAP

Unified security style in clustered Data ONTAP eliminates many of the caveats and restrictions imposed by the UNIX, NTFS, and Mixed security styles. Unified security style provides ubiquitous access control and management for both UNIX and Windows clients.

In Unified security style:

- ACLs and permissions can be viewed by UNIX and Windows clients regardless of the on-disk effective security style; that is, regardless of the protocol previously used to set ownership or permissions.
- ACLs and permissions can be modified by UNIX and Windows clients regardless of the on-disk effective security style; that is, regardless of the protocol previously used to set ownership or permissions.
- UNIX mode bits can be merged into an existing ACL regardless of the on-disk effective security style; that is, regardless of whether the ACL is an NFSv4 ACL or an NTFS ACL.
- ACEs in NTFS ACLs can represent UNIX or Windows principals (users or groups).
    - Current NFSv4 clients and servers support a single NFSv4 domain, and all principals must be mapped into that NFSv4 domain. For this reason, NFSv4 ACLs set by NFSv4 clients contain only NFSv4 principals.

To control the NFSv4 ACL preservation option, use the following command:

```
cluster::> set advanced
cluster::*> nfs server modify -vserver [SVM] -v4-acl-preserve enabled
```

In clustered Data ONTAP, it is possible to view the effective security style and ACLs of an object in storage by using the `vserver security file-directory` command set. Currently, the command does not autocomplete for SVMs with content repository enabled, so the SVM name must be entered manually.

Example:

```
::> vserver security file-directory show -vserver infinite -path /infinitevolume/CIFS

               Vserver: infinite
             File Path: /infinitevolume/CIFS
        Security Style: mixed
       Effective Style: ntfs
        DOS Attributes: 10
 DOS Attributes in Text: ----D---
Expanded Dos Attributes: -
          Unix User Id: 500
         Unix Group Id: 512
        Unix Mode Bits: 777
 Unix Mode Bits in Text: rwxrwxrwx
                   ACLs: NTFS Security Descriptor
                         Control:0x8504
                         Owner:DOMAIN\Administrator
                         Group:DOMAIN\Domain Users
                         DACL - ACEs
                           ALLOW-S-1-520-0-0x1f01ff-OI|CI
                           ALLOW-S-1-520-1-0x1201ff-OI|CI
                           ALLOW-S-1-520-2-0x1201ff-OI|CI
                           ALLOW-DOMAIN\unified1-0x1f01ff-OI|CI
                           ALLOW-DOMAIN\Administrator-0x1f01ff-OI|CI
```

```
                          ALLOW-DOMAIN\unifiedgroup-0x1f01ff-OI|CI

::> vserver security file-directory show -vserver infinite -path /infinitevolume/NFS

               Vserver: infinite
             File Path: /infinitevolume/NFS
       Security Style: mixed
      Effective Style: unix
       DOS Attributes: 10
 DOS Attributes in Text: ----D---
Expanded Dos Attributes: -
         Unix User Id: 100059
        Unix Group Id: 10008
       Unix Mode Bits: 777
 Unix Mode Bits in Text: rwxrwxrwx
                 ACLs: NFSV4 Security Descriptor
                       Control:0x8014
                       DACL - ACEs
                         ALLOW-S-1-8-10001-0x16019f
                         ALLOW-S-1-520-0-0x1601ff
                         ALLOW-S-1-520-1-0x1201ff-IG
                         ALLOW-S-1-520-2-0x1201ff
```

In this example, a volume named `infinite` contains a folder with effective security style of UNIX called `NFS` and an effective NTFS style folder called `CIFS`. The effective style reflects the protocol that last applied an ACL to the object and, although both folders indicate Mixed security style, the behavior is Unified security style. Table 12 shows the main differences between the Mixed and Unified security styles.

Table 12) Mixed mode versus Unified security style.

Mixed	Unified
• NFS clients cannot view an existing NTFS-style ACL • NFS clients can only blindly overwrite an existing NTFS-style ACL • NFS mode bits cannot be merged into an existing NTFS-style ACL • NFS principals (users or groups) cannot be represented in an NTFS-style ACL • Windows clients cannot view an existing NFSv4 ACL • Windows clients can only blindly overwrite an existing NFSv4 ACL	• NFS clients can view and modify existing NTFS-style ACLs – Group mapping has been added to support NFSv4 clients. Both users and groups can be mapped into the NFSv4 domain. – If an NFS client saves mode bits, the mode bits can be merged into an existing ACL. – Independently configurable for NFS ACLs or NTFS-style ACLs. • Windows clients can view and modify existing NFSv4 ACLs – UNIX principals may appear in NTFS-style ACLs. UNIX principals are distinguished by a unix-user or unix-group prefix.

Note: The effective style indicates the protocol most recently used to set the ACL in all security styles. The difference in Unified security style is that the effective style does not indicate ACL management restrictions or limitations.

Unified Security Style Behavior in NFSv3

The NFSv3 protocol does not support ACLs. Therefore, when a client mounts an Infinite Volume via NFSv3, only the mode bits are visible to that client. Mode bits are the classic `rwx` style of permissions that can be numerically represented 0-7 for owner, group, and other. For more information about mode bit permissions, see File Permission Modes from Oracle.

ACLs are still honored for access control. UNIX to Windows name mapping is required to interpret Windows principals in NTFS-style ACLs. A UNIX user would need to map to a valid Windows user in order to interpret the Windows principal in NTFS-style ACEs.

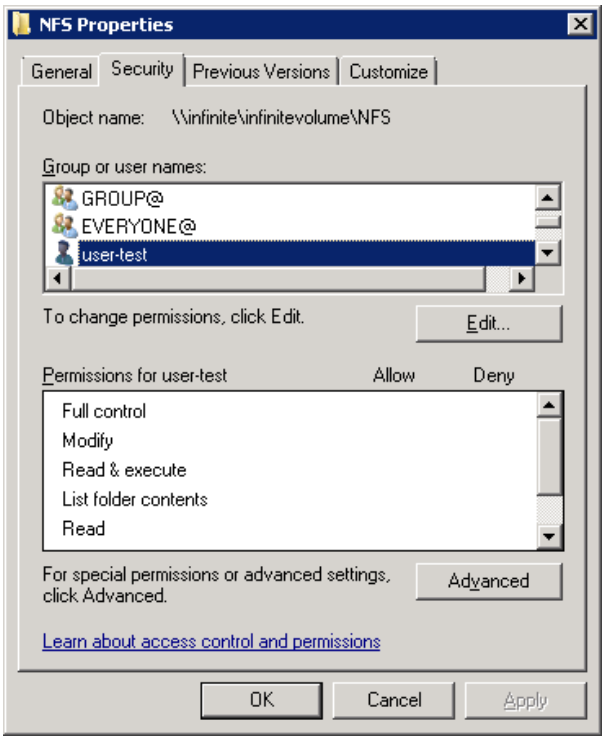
Unified Security Style Behavior in NFSv4.x

Infinite Volumes currently support NFSv4.1 only in clustered Data ONTAP 8.2. NFSv4.1 must be enabled and configured on the cluster, and clients must be NFSv4.1 capable. A single identity mapping domain should be available (only one domain per SVM is supported in clustered Data ONTAP). NetApp highly recommends enabling NFSv4 ACL preservation when using NFSv4 ACLs.

When NFSv4 ACLs are used, the ACLs map directly to Windows SIDs, allowing unified access to files and directories.

However, when an NFSv4 ACL cannot be mapped to a Windows SID, the ACL represents itself with `user-` or `group-` in the list, and a dummy SID is created in Data ONTAP, using the UID or GID of the user being represented.

Example:




```
cluster::> vserver security file-directory show -vserver infinite -path /infinitevolume/NFS

                Vserver: infinite
              File Path: /infinitevolume/NFS
        Security Style: mixed
       Effective Style: unix
        DOS Attributes: 10
 DOS Attributes in Text: ----D---
Expanded Dos Attributes: -
          Unix User Id: 100059
         Unix Group Id: 10008
        Unix Mode Bits: 777
 Unix Mode Bits in Text: rwxrwxrwx
                  ACLs: NFSV4 Security Descriptor
                        Control:0x8014
                        DACL - ACEs
```

```
                           ALLOW-S-1-8-10001-0x16019f
                           ALLOW-S-1-520-0-0x1601ff
                           ALLOW-S-1-520-1-0x1201ff-IG
                           ALLOW-S-1-520-2-0x1201ff
cluster::> set diag
cluster::*> diag secd authentication translate -node node1 -vserver infinite -sid S-1-8-10001
domain\user-test (User)
cluster::*> diag secd authentication translate -node node1 -vserver infinite -win-name
domain\user-test
S-1-8-10001
```

The other NFSv4 ACLs listed on the object are the default EVERYONE@, GROUP@, and OWNER@ ACLs.

```
cluster::*> diag secd authentication translate -node node1 -vserver infinite -sid S-1-520-0
OWNER@ (Well known group)

cluster::*> diag secd authentication translate -node node1 -vserver infinite -sid S-1-520-1
GROUP@ (Well known group)

cluster::*> diag secd authentication translate -node node1 -vserver infinite -sid S-1-520-2
EVERYONE@ (Well known group)
```

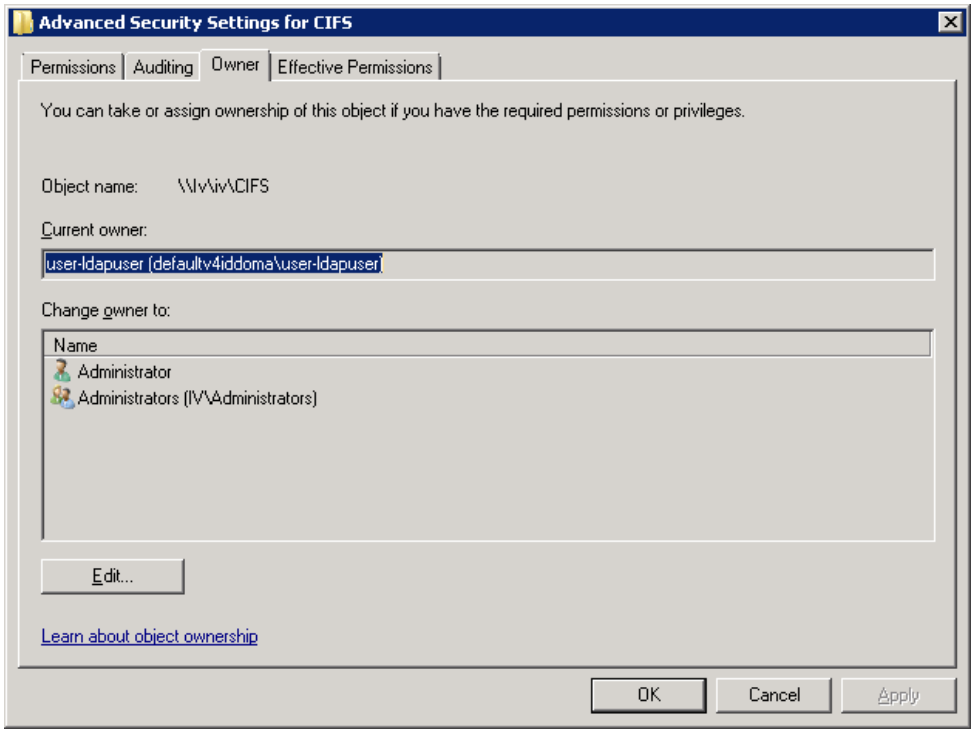
These default ACLs get set on every object and reflect the mode bit translation for NFSv3 backward compatibility.

Example:

```
# ls -la | grep NFS
drwxrwxrwx.  2 unified1      unifiedgroup  4096 Nov  1 13:46 NFS

# nfs4_getfacl /infinitevol/NFS
A::test@domain.win2k8.netapp.com:rwatTnNcCy
A::OWNER@:rwaDxtTnNcCy
A:g:GROUP@:rwaDxtTnNcy
A::EVERYONE@:rwaDxtTnNcy

# chmod 755 NFS

# ls -la | grep NFS
drwxr-xr-x.  2 unified1      unifiedgroup  4096 Nov  1 13:46 NFS

# nfs4_getfacl /infinitevol/NFS
A::test@domain.win2k8.netapp.com:rwatTnNcCy
A::OWNER@:rwaDxtTnNcCy
A:g:GROUP@:rxtncy
A::EVERYONE@:rxtncy
```

Unified Security Style Behavior in NFSv4 ID Domain

Unified security style leverages the NFSv4 ID domain attribute on the NFS server to formulate unified ACLs. The default value of this is `defaultv4iddomain.com`. Therefore, users may appear with the following format, even if NFSv4.x is not used:



To avoid this behavior, set the `v4-id-domain` option in the NFS server even if NFSv4.x is not being used.

Example:

```
cluster::> nfs server modify -vserver infinite -v4-id-domain domain.win2k8.netapp.com
```

Unified Security Style Behavior in CIFS

Infinite Volumes currently support SMB version 1.0 only. NTFS-style ACLs are supported and operate identically to FlexVol. volumes With Unified security style, however, NTFS ACLs are retained when UNIX mode bits are applied. This behavior is similar to the NFSv4 ACL preserve option, but it cannot be managed from the command line.

Unreachable Attributes

If an Infinite Volume data constituent is offline, the `unreachable-attr-action` attribute on the volume controls how data access behaves for inaccessible attributes.

There are two options: return-generated and wait.

- **Return-generated** returns default values for the attributes, which appear to the client as a file size of 0 and timestamps that are in the past. This is the default setting.
- **Wait** causes the volume to return a RETRY error, which can cause some clients to appear to hang because they retry the request indefinitely.

Default Users

Clustered Data ONTAP has a concept of default users for CIFS and NFS access. These defaults users provide authentication and mapping for users that do not map into a valid name service source. With Infinite Volume, group mapping is also available to map Windows groups to groups that can be recognized in the NFSv4 domain. The default group concept is relevant for Infinite Volume only. User mapping provides a default user for ACLs, as well as for user authentication.

The following CIFS options allow modification of the default user and group mapping to a valid UNIX user or group for all CIFS requests:

```
cluster::> cifs options show -vserver infinite -default-unix-
    -default-unix-user  -default-unix-group
```

Note: Pcuser is the default user created during CIFS server creation. It uses UID 65534, which often maps to the `nfsnobody` user on UNIX clients. Consider changing this user if a different user/UID is desired.

Note: By default, no default group is set for CIFS servers.

The following NFS options allow modification of the default user and group mapping to a valid Windows user for all NFS requests:

```
cluster::> nfs server modify -vserver infinite -default-win-
    -default-win-user  -default-win-group
```

Note: By default, no default Windows user or group is set for the NFS server.

Infinite Volume Export Policies

When SVMs are created for Infinite Volumes, several default export policies are created. These policies contain default rules, which are applied to the volume in the SVM. In clustered Data ONTAP 8.2, export policies apply only to NFS by default. Previous versions of clustered Data ONTAP used export policies for CIFS access as well.

The following default polices are created when an SVM is created for an Infinite Volume:

```
default
repos_root_readonly_export_policy
```

When an Infinite Volume is added, two additional policies are also created:

```
default
repos_namespace_export_policy
repos_restricted_export_policy
repos_root_readonly_export_policy
```

These policies have the following default rules:

```
                                    Vserver: IV
                                Policy Name: repos_namespace_export_policy
                                 Rule Index: 1
                             Access Protocol: any
Client Match Hostname, IP Address, Netgroup, or Domain: 0.0.0.0/0
                               RO Access Rule: any
                               RW Access Rule: any
User ID To Which Anonymous Users Are Mapped: 0
                     Superuser Security Types: any
                 Honor SetUID Bits in SETATTR: true
                    Allow Creation of Devices: true
                  NTFS Unix Security Options: fail
          Vserver NTFS Unix Security Options: -
                       Change Ownership Mode: restricted
               Vserver Change Ownership Mode: -
```

```
                                    Vserver: IV
                                Policy Name: repos_namespace_export_policy
                                 Rule Index: 2
                            Access Protocol: any
Client Match Hostname, IP Address, Netgroup, or Domain: ::0/0
                              RO Access Rule: any
                              RW Access Rule: any
User ID To Which Anonymous Users Are Mapped: 0
                    Superuser Security Types: any
                Honor SetUID Bits in SETATTR: true
                   Allow Creation of Devices: true
                  NTFS Unix Security Options: fail
          Vserver NTFS Unix Security Options: -
                      Change Ownership Mode: restricted
               Vserver Change Ownership Mode: -

                                    Vserver: IV
                                Policy Name: repos_root_readonly_export_policy
                                 Rule Index: 1
                            Access Protocol: any
Client Match Hostname, IP Address, Netgroup, or Domain: 0.0.0.0/0
                              RO Access Rule: any
                              RW Access Rule: never
User ID To Which Anonymous Users Are Mapped: 0
                    Superuser Security Types: any
                Honor SetUID Bits in SETATTR: true
                   Allow Creation of Devices: true
                  NTFS Unix Security Options: fail
          Vserver NTFS Unix Security Options: -
                      Change Ownership Mode: restricted
               Vserver Change Ownership Mode: -

                                    Vserver: IV
                                Policy Name: repos_root_readonly_export_policy
                                 Rule Index: 2
                            Access Protocol: any
Client Match Hostname, IP Address, Netgroup, or Domain: ::0/0
                              RO Access Rule: any
                              RW Access Rule: never
User ID To Which Anonymous Users Are Mapped: 0
                    Superuser Security Types: any
                Honor SetUID Bits in SETATTR: true
                   Allow Creation of Devices: true
                  NTFS Unix Security Options: fail
          Vserver NTFS Unix Security Options: -
                      Change Ownership Mode: restricted
               Vserver Change Ownership Mode: -
```

Note: The policies named “default” and “repos_restricted_export_policy” do not contain any rules by default.

For information about how these rules affect access, see section 3.4, ““Translation of NFS Export Policy Rules from 7-Mode to Clustered Data ONTAP.”

Infinite Volume Junction Paths

By default, if no junction path is specified when creating an Infinite Volume, the path is `/NS`. This differs from the FlexVol behavior, in which a junction path is created only if one has been specified. To control this behavior, either specify the junction path at volume creation or unmount and remount the Infinite Volume to the desired path.

6 NFS Performance Monitoring and Data Gathering

In clustered Data ONTAP, EMS messages are viewed differently than they are in 7-Mode. In 7-Mode, the /etc/messages file located in /vol/vol0 can be viewed via CLI with rdfile or via NFS or CIFS. Clustered Data ONTAP currently does not provide NAS protocol visibility for logs. However, there are various ways to view the log files.

Viewing Log Files

To view EMS errors:

```
cluster::> event log show
```

SecD Troubleshooting

SecD provides a number of diag-level commands to troubleshoot authentication and permissions issues. The following information shows examples of commands to use for various scenarios. All commands are at the diagnostic level (denoted by * in the CLI prompt). Exercise caution while at the diagnostic level.

Check name mapping functionality

```
cluster::*> diag secd name-mapping show -node node1 -vserver vs0 -direction unix-win -name
ldapuser
ldapuser maps to WIN2K8\ldapuser
```

Translate user names and groups into SIDs or UIDs

```
cluster::*> diag secd authentication translate -node node1 -vserver vs0 -unix-user-name ldapuser
55

cluster::*> secd authentication translate -node node1 -vserver vs0 -win-name DOMAIN\ldapuser
S-1-5-21-2216667725-3041544054-3684732124-1123

cluster::*> secd authentication translate -node node1 -vserver vs0 -unix-group-name unixadmins
503
```

Enable/disable debug-level logging in SecD

```
cluster::*> diag secd trace set -node <nodename> -trace-all [yes|no]
```

Restart SecD process (NOTE: This is a disruptive operation)

```
cluster::*> diag secd restart -node <nodename>
```

Check user name credentials and group membership as SecD sees them

```
cluster::*> diag secd authentication show-creds -node node1 -vserver vs0 -unix-user-name ldapuser
-list-name true -list-id true

 UNIX UID: 55 (ldapuser) <> Windows User: S-1-5-21-2216667725-3041544054-3684732124-1123
(DOMAIN\ldapuser (Domain User))

 GID: 513 (Domain Users)
 Supplementary GIDs:
  503  (unixadmins)

 Windows Membership:
  S-1-5-21-2216667725-3041544054-3684732124-513     DOMAIN\Domain Users (Domain group)
  S-1-5-21-2216667725-3041544054-3684732124-1108     DOMAIN\unixadmins (Domain group)
  S-1-5-32-545     BUILTIN\Users (Alias)
 User is also a member of Everyone, Authenticated Users, and Network Users

 Privileges (0x80):
```

NFS Troubleshooting Basics

The following section covers some NFS troubleshooting basics to help assist in resolving configuration issues that cause access problems with NFS mounts.

When troubleshooting NFS access issues, it's important to note where in the process the NFS request is failing. For example, a failure during mount will generally have a much different root cause than a failure during file access.

Export policies and rules are some of the most common issues in clustered Data ONTAP NFS access issues. For information on export policies and rules, see the section earlier in this document on export policies and rules.

Cannot Mount

The following section covers common errors and scenarios in which an NFS mount fails to a clustered Data ONTAP system. It also covers how to resolve the issue.

Table 13) Common mount failures.

Error	What to Check	How to Resolve
Access denied by server while mounting	NFS client - If using Kerberos, is the configuration correct? See TR-4073 for details. - If using AUTH_UNIX or AUTH_SYS, does the user resolve in the name service? NFS server - NFS server options - Can the user be resolved by the server into a UID? - Is SecD running? - Export policy rule(s) of the volume - Export policy rule(s) of the parent volume(s) Common mistakes in export policies include: - No rule defined in export policy - Clientmatch is incorrect (such as 0.0.0.0 instead of 0.0.0.0/0 for all clients) - Client match does not allow client attempting	Review the NFS client configuration. Review the NFS server configuration, export policies, and rules and make corrections.

	access - Access protocol does not allow NFS - RO policy is set to incorrect value - User is squashed to the anon user, which does not have permissions to the volume NFS server options that could cause access-denied errors during mount include: - NFS mount root only	
Requested NFS version or transport protocol is not supported	NFS server - NFS server options for the NFS version - TCP/UDP settings - NFS server is running Network/firewall - Verify that the firewall is not blocking NFS or related ports - Verify that the data LIF allows NFS SVM - Verify that the SVM allows NFS as a protocol	Review the NFS server configuration to verify that the protocol and NFS version are enabled and the server is running. Review the network settings and data LIFs to verify that NFS is allowed. Review the SVM to verify that NFS is allowed.
Mount hangs indefinitely	Network - Is the LIF up? Can it be pinged? - Is the DNS entry correct? - Is the LIF at home? Are failover groups configured properly? - Is the firewall blocking any of the NFS ports?	Review the network settings. Review the data LIFs on the cluster.
Mounting failed, reason given by server: No such file or directory	Mount syntax - Is the right mount path specified?	Review the mount syntax. Review the NFS volume.

	NFS server - Is the junction the same as the mount path? - Is the volume mounted? - If using LS mirrors, have they been updated? NFS client - If volume permission changes have been made, has the volume been remounted? - Is the volume mounted with no access cache?	
Mount point is busy or already mounted	NFS client - Is something already mounted to that mount point?	Review the output of the `mount` command.
Mount point/test does not exist	NFS client - Does the mount point exist?	Use a valid mount point.
Only root can do that	NFS client - Does the user have permission to mount? NFS server - Is root-only mount set?	Check client and server configuration.
Operation not permitted	NFS client - Does the user have root access? NFS server - Does the client export as root? - Is superuser set properly?	Check export policies and rules. Check client configuration for root access.

For information regarding mount issues using NFS Kerberos, see TR-4073: Secure Unified Authentication for more details.

Permission Denied/Access Issues

The following section covers issues in which an NFS mount succeeds but accessing the mount fails; it also covers how to resolve the issue. Not all scenarios are covered.

Table 14) Common access issues.

Error	What to Check	How to Resolve
Permission denied (while accessing mount/reading/writing)	NFS server - Do the data volume's security settings permit the user access? - Do the parent volume's security settings permit the user access? - What is the volume's security style? Does the user attempting access map to a valid Windows user if the security style is NTFS? - If able to `cd` but not able to read or write, but UNIX permissions seem to allow access to all users, does the cluster know the user attempting access?	Verify and modify the volume's security. Verify and modify the export policy rule to allow access. Verify that the user can map properly into name service. Verify that the user exists in name service.
Permission denied (while attempting chmod/chown/chgrp) Operation not permitted (while chown/chmod/chgrp)	NFS server - Is chown allowed by anyone other than root? - Is the user the owner of the file? NFS client - Is the user root?	Change the NFS server and export policy rule options for chown to "unrestricted."
Not a directory (when traversing Snapshot directory)	NFS client - Check the kernel version	See Bugzilla 798809.

Files Written as "Nobody"

The following section covers issues in which NFSv4 clients show file ownership as the "nobody" user; it also covers how to resolve the issue. Not all scenarios are covered.

A stale file handle error occurs when the server file system has changed and the file handle is no longer valid. For example: Client A opens file xxx.yyy for edit, Client B deletes this file, Client A goes to save the edit—Client A will get a stale file handle error.

This can occur not just for operations on individual files, but also due to changes in directory structure.

Table 15) Files written as "nobody" in NFSv4.

Error	What to Check	How to Resolve
No error; files written as the "nobody" user (or some other unexpected user)	NFS client - /var/log/messages file - Is the NFSv4 domain specified in /etc/idmapd.conf? - What is the user name attempting access? Can the client resolve the name in the name service? NFS server - Can the cluster translate the user name to a UID? - Is the NFSv4 domain set? - If the user writing the file is root, does the export policy squash root to the anon user (superuser = none)? - Is the name service working properly?	Fix the NFSv4 domain ID. Verify that the user name matches the NFSv4 domain user name exactly (case sensitive). Verify that the client and cluster can resolve the user name. Adjust the export policy rule to allow superuser access if desired.

Stale File Handle or NFS Mount

The following section covers scenarios in which a stale file handle error might occur and how to resolve them. Not all scenarios are covered.

Error	What to Check	How to Resolve
mount.nfs: Stale file handle	NFS server - Did the junction path for the volume change? - Is the volume mounted? - Does the volume still exist? NFS client - Is the mount already mounted somewhere else on the client?	Verify and remount the volume from the client. Mount the volume from the cluster if it is not mounted.

Cannot open directory: Stale file handle	NFS server - Was the fsid-change option modified?	Remount the volume from the client.
Was not found in /proc/mounts	NFS client - Does the mount show up in /proc/mounts? - Does the mount show (deleted) in the output?	Reboot the client.

Performance Monitoring in Clustered Data ONTAP 8.1 and Earlier

In 7G, `nfsstat -d` was a common and popular command to provide information on NFS operations. In clustered Data ONTAP, `nfsstat -d` does not exist. However, the `statistics` command can be used with a variety of parameters to get details of NFS metadata operations at the individual cluster node level.

```
cluster::*> statistics show-periodic -node node1
-object nfs3 -interval 1
```

null	gattr	sattr	lookup	access	rsym	read	write	create	mkdir	symln	mknod	remove	rmdir	rename	link
0	0	0	0	0	0	15	11	0	0	0	0	0	0	0	0
0	16	0	0	0	0	45	6	0	0	0	0	0	0	0	0
0	0	0	0	0	0	12	2	0	0	0	0	0	0	0	0
0	0	0	0	0	0	47	16	0	0	0	0	0	0	0	0
0	0	0	0	0	0	11	2	0	0	0	0	0	0	0	0
0	22	0	0	9	0	12	5	0	0	0	0	0	0	0	0
0	16	0	0	0	0	49	5	0	0	0	0	0	0	0	0

The following command provides information from each individual volume about NFS workload and latency.

```
cluster::*> statistics show-periodic -node node1
-object volume -instance vs2_nfs4_data3
```

instance name	node name	instance uuid	avg latency	total ops	read data	read latency	read_ops	write data	write latency	write ops	other latency	other ops	nfs read data	nfs read latency	nfs read ops
0	0	0	0us	0	0B	0us	0	0B	0us	0	0us	0	0B	0us	0
0	0	0	0us	0	0B	0us	0	0B	0us	0	0us	0	0B	0us	0
0	0	0	0us	0	0B	0us	0	0B	0us	0	0us	0	0B	0us	0

A proper breakdown of the different NFS frame sizes can be achieved by running the following command for each cluster node.

```
cluster::*> statistics protocol-request-size show -node node1
-stat-type nfs3_*


Node:                    node1
Stat Type:               nfs3_read
                         Value       Delta
--------------           --------    ----------
Average Size:               7437             -
Total Request Count:     4681100             -
0-511:                        34             -
512-1023:                     19             -
```

```
1K-2047:                    37831            -
2K-4095:                       42            -
4K-8191:                       77            -
8K-16383:                 2726657            -
16K-32767:                1250362            -
32K-65535:                 615306            -
64K-131071:                  1252            -
128K - :                    49520            -

Node:                    node1
Stat Type:               nfs3_write
                         Value       Delta
--------------           --------    ----------
Average Size:                8434            -
Total Request Count:
                       107524781            -
0-511:                       3782            -
512-1023:                    4518            -
1K-2047:                   929818            -
2K-4095:                 45939930            -
4K-8191:                  9699755            -
8K-16383:                 3503960            -
16K-32767:               39615020            -
32K-65535:                 543995            -
64K-131071:                657028            -
128K - :                  6626975            -
2 entries were displayed.
```

The following commands identify the type of protocol in use and the details of the RPC calls.

```
cluster::*> statistics oncrpc show-rpc-calls -node node1 -protocol tcp

              Node: node1
 Transport Protocol: tcp
Bad Procedure Calls: 0             -
   Bad Length Calls: 0             -
   Bad Header Calls: 8       0/s:16s
          Bad Calls: 8       0/s:16s
  Bad Program Calls: 0             -
        Total Calls: 116491426      58/s:16s
```

Per-client statistics are also available to identify which client IP addresses are generating what NFS traffic in clustered Data ONTAP.

```
cluster::*> statistics settings modify -client-stats enabled

Warning: System performance may be significantly impacted. Are you sure?
Do you want to continue? {y|n}: y

cluster::*> statistics show -object client
```

```
Node: fas3070c-svl19
    Object.Instance.Counter                                  Value          Delta
    ------------------------------------------- ------------- -------------
    client.172.17.44.106.hostname
                                fas6080c-svl14.iop.eng.netapp.com
            -
    client.172.17.44.106.total-ops                             0             -
    client.172.17.44.106.nfs2-ops                              0             -
    client.172.17.44.106.nfs3-ops                              0             -
    client.172.17.44.106.nfs4-ops                              0             -
    client.172.17.44.106.cifs-ops                              0             -
    client.172.17.44.106.recv-data                          100B             -
    client.172.17.44.106.sent-data                            0B             -
    client.172.17.44.106.recv-packets                          2             -
    client.172.17.44.106.avg-latency-remote                  0us             -
    client.172.17.44.151.hostname
    client.172.17.44.151.total-ops                             0             -
    client.172.17.44.151.nfs2-ops                              0             -
    client.172.17.44.151.nfs3-ops                              0             -
    client.172.17.44.151.nfs4-ops                              0             -
    client.172.17.44.151.cifs-ops                              0             -
```

We can also drill down to details for a single client.

```
cluster::*> statistics show -object client -instance 172.17.44.106

Node: fas3070c-svl19
    Object.Instance.Counter                                  Value          Delta
    ------------------------------------------- ------------- -------------
    client.172.17.44.106.hostname
                                fas6080c-svl14.iop.eng.netapp.com
            -
    client.172.17.44.106.total-ops                             0             -
    client.172.17.44.106.nfs2-ops                              0             -
    client.172.17.44.106.nfs3-ops                              0             -
    client.172.17.44.106.nfs4-ops                              0             -
    client.172.17.44.106.cifs-ops                              0             -
    client.172.17.44.106.recv-data                          200B       1B/s:80s
    client.172.17.44.106.sent-data                            0B             -
    client.172.17.44.106.recv-packets                          4        0/s:80s
    client.172.17.44.106.sent-packets                          0             -
    client.172.17.44.106.avg-latency                         0us             -
    client.172.17.44.106.nlm-ops                               0             -
    client.172.17.44.106.mount-ops                             0             -
    client.172.17.44.106.local-ops                             0             -
    client.172.17.44.106.remote-ops                            0             -
    client.172.17.44.106.avg-latency-local                   0us             -
    client.172.17.44.106.avg-latency-remote                  0us             -
17 entries were displayed.
```

In clustered Data ONTAP, use the `locks show` command to list all the locks assigned to files residing in a specific volume under an SVM.

```
cluster::*> vserver locks show

Vserver: vs2_nfs4
Volume   Object Path              LIFE         Protocol  Lock Type   Client
-------- ------------------------ ---------- --------- ---------- ----------
vs2_nfs4_data2
        /nfs4_ds2/app/grid/product/11.2.0/dbhome_1/oc4j/j2ee/home/persistence/jms.state
                                 vs2_nfs4_data1
                                           nlm       byte-range  172.17.37.103
              Bytelock Offset(Length): 0 (9223372036854775807)
        /nfs4_ds2/app/grid/product/11.2.0/dbhome_1/oc4j/j2ee/home/persistence/oc4jJmsExceptionQueue
                                 vs2_nfs4_data1
                                           nlm       byte-range  172.17.37.103
              Bytelock Offset(Length): 0 (9223372036854775807)
2 entries were displayed.
```

The `locks break` command can be used to remove a lock on a particular file.

```
cluster::*> locks break -vserver vs2 -volume vs2_nfs4_data2 -lif vs2_nfs4_data1 -path
/nfs4_ds2/app/grid/product/11.2.0/dbhome_1/oc4j/j2ee/home/persistence/jms.state
```

`Perfstat8` is also available for clustered Data ONTAP for use in performance collection. Each version of Perfstat improves data collection for clusters.

"Admin" and "diag" user access is needed to run the `perfstat` command.

The following command illustrates how to capture a perfstat for a clustered Data ONTAP cluster. The cluster management IP should always be used. Perfstat will discern the nodes in the cluster and collect data for each node. In this example, the cluster management IP is 172.17.37.200 for a 4-node cluster. This perfstat collects 24 iterations with a sleep time of 300 seconds between iterations. More examples are available from the Perfstat8 tool download page:

https://support.netapp.com/NOW/download/tools/perfstat/perfstat8.shtml

A valid NetApp Support account is required for access to the perfstat8 tool.

```
[root@linux]# ./perfstat8 --verbose -i 24,300 172.17.37.200
```

Performance Monitoring in 8.2 Clustered Data ONTAP

In clustered Data ONTAP 8.2, performance monitoring commands changed slightly as the underlying performance monitoring subsystems get an overhaul. As a result, legacy performance commands use the `statistics-v1` command set, while the newer performance monitoring commands leverage the `statistics` command.

The following should be kept in mind for performance commands in clustered Data ONTAP 8.2:

- NFS per client statistics do not exist under `statistics` in 8.2; they only exist under `statistics-v1`.
- Currently there is no way to zero counters; the only way to zero counters is via reboot.

Note: Newer releases of clustered Data ONTAP will introduce new performance improvements and bug fixes so that statistics-v1 will no longer be necessary.

Appendix

NFSv3 Option Changes in Clustered Data ONTAP

Table 16 shows how to apply the 7-Mode options for NFSv3 in clustered Data ONTAP.

Table 16) NFSv3 configuration options in clustered Data ONTAP.

7-Mode Option	How to Apply in Clustered Data ONTAP	Remark
nfs.response.trace	`vserver nfs modify -vserver vs0vs0 -trace-enabled`	If this option is “on,” it forces all NFS requests that have exceeded the time set in `nfs.response.trigger` to be logged. If this option is "off,” only one message is logged per hour.
nfs.rpcsec.ctx.high	`vserver nfs modify -vserver vs0vs0 -rpcsec-ctx-high`	If set to a value other than zero, it sets a high-water mark on the number of

7-Mode Option	How to Apply in Clustered Data ONTAP	Remark
		stateful RPCSEC_GSS (see RFC 2203) authentication contexts. (Only Kerberos V5 currently produces a stateful authentication state in NFS.) If it is zero, then no explicit high-water mark is set.
nfs.rpcsec.ctx.idle	`vserver nfs modify -vserver vs0vs0 -rpcsec-ctx-idle`	This is the amount of time, in seconds, that an RPCSEC_GSS context (see the description for the nfs.rpcsec.ctx.high option) is permitted to be unused before it is deleted.
nfs.tcp.enable	`vserver nfs modify -vserver vs0vs0 -tcp enabled`	When this option is enabled, the NFS server supports NFS over TCP.
nfs.udp.xfersize	`vserver nfs modify -vserver vs0vs0 -udp-max-xfer-size 32768`	This is the maximum transfer size (in bytes) that the NFSv3 mount protocol should negotiate with the client for UDP transport.
nfs.v3.enable	`vserver nfs modify -vserver vs0vs0 -v3 enabled`	When enabled, the NFS server supports NFS version 3.

NFSv4 Option Changes in Clustered Data ONTAP

Table 17 shows how to apply the 7-Mode options for NFSv4 in clustered Data ONTAP.

Table 17) NFSv4 configuration options in clustered Data ONTAP.

7-Mode Option	How to Apply in Clustered Data ONTAP	Remark
nfs.v4.enable	`vserver nfs modify -vserver vs0 -v4 enabled`	When this option is enabled, the NFS server supports NFS version 4.
nfs.v4.read_delegation	`vserver nfs modify -vserver vs0 -v4-read-delegation`	When this option is enabled, read delegations are supported for NFS version 4.
nfs.v4.write_delegation	`vserver nfs modify -vserver vs0 -v4-write-delegation`	When this option is enabled, write delegations are supported for NFS version 4.
nfs.tcp.xfersize	`vserver nfs modify -vserver vs0 -tcp-max-xfer-size`	This is the maximum transfer size (in bytes) that the NFS mount protocol should negotiate with the client for TCP transport.
nfs.v4.acl.enable	`vserver nfs modify -vserver vs0 -v4-acl`	Enable NFSv4 ACL support.
nfs.v4.reply_drop	`vserver nfs modify -vserver vs0 -v4-reply-drop`	This is a debugging operation to cause requests to be dropped to test client/server resiliency.
nfs.v4.id.domain	`vserver nfs modify -vserver`	This option controls the domain

	`vs0 -v4-id-domain`	portion of the string form of user and group names as defined in the NFS version 4 protocol. The domain name is normally taken from the NIS domain in use or otherwise from the DNS domain. However, if this option is set it overrides this default behavior.
locking.grace_lease_seconds	Currently controlled through nodeshell using the same option; the same value applies to both 7-Mode and clustered Data ONTAP.	
nfs.v4.snapshot.active.fsid.enable	`vserver nfs modify -vserver vs0 -v4-fsid-change`	This affects the behavior of the fsid used for the `.snapshot` directory and entities in the `.snapshot` directory. The default behavior is that they use a different fsid than the active copy of the files in the file system. When this option is enabled, the fsid is identical to that for files in the active file system. The option is "off" by default.
kerberos.file_keytab.principal	`vserver nfs kerberos-config modify -vserver vs0 -spn`	
kerberos.file_keytab.realm	`vserver nfs kerberos-config modify -vserver vs0 -spn`	
nfs.kerberos.enable on/off	`vserver nfs kerberos-config modify -vserver vs0 -kerberos enable/disable`	

kerberos.file_keytab.enable on/off	`kerberos.file_keytab.enable = on:`
	`'vserver services kerberos-realm modify -kdc-vendor Other'`
	
	In this case, the keytab file must be added to the clustered Data ONTAP configuration:
	`'vserver nfs kerberos-config modify -keytab-uri'`
	
	`kerberos.file_keytab.enable = off:`
	`'vserver services kerberos-realm modify -kdc-vendor Microsoft'`	

NFSv3 Port Changes

In clustered Data ONTAP, the mountd port changed from 4046 to 635. The status port also changed from 4047 to 4046. The following shows an example of rpcinfo –p showing a 7-Mode and a clustered Data ONTAP system.

7-Mode rpcinfo:

```
[root@centos6 ~]# rpcinfo -p 10.61.84.240
   program vers proto   port  service
    100003    4   tcp   2049  nfs
    100011    1   udp   4049  rquotad
    100024    1   tcp   4047  status
    100024    1   udp   4047  status
    100021    4   tcp   4045  nlockmgr
    100021    3   tcp   4045  nlockmgr
    100021    1   tcp   4045  nlockmgr
    100021    4   udp   4045  nlockmgr
    100021    3   udp   4045  nlockmgr
    100021    1   udp   4045  nlockmgr
    100005    3   tcp   4046  mountd
    100003    3   tcp   2049  nfs
    100005    2   tcp   4046  mountd
    100005    1   tcp   4046  mountd
    100003    2   tcp   2049  nfs
    100005    3   udp   4046  mountd
    100003    3   udp   2049  nfs
    100005    2   udp   4046  mountd
    100005    1   udp   4046  mountd
    100003    2   udp   2049  nfs
    100000    2   tcp    111  portmapper
    100000    2   udp    111  portmapper
```

Clustered Data ONTAP rpcinfo:

```
[root@centos6 ~]# rpcinfo -p 10.61.92.34
   program vers proto   port  service
    100000    2   udp    111  portmapper
    100000    2   tcp    111  portmapper
    100000    3   udp    111  portmapper
    100000    3   tcp    111  portmapper
    100000    4   udp    111  portmapper
    100000    4   tcp    111  portmapper
    100003    3   udp   2049  nfs
    100003    3   tcp   2049  nfs
    100003    4   tcp   2049  nfs
    400010    1   tcp   2049
    100005    1   udp    635  mountd
    100005    2   udp    635  mountd
    100005    3   udp    635  mountd
    100005    1   tcp    635  mountd
    100005    2   tcp    635  mountd
    100005    3   tcp    635  mountd
    100021    4   udp   4045  nlockmgr
    100021    4   tcp   4045  nlockmgr
    100024    1   udp   4046  status
    100024    1   tcp   4046  status
    100011    1   udp   4049  rquotad
```

NFSv4 User ID Mapping

Clustered Data ONTAP supports "numeric-ids," which can be enabled using the following command at the SVM level.

```
cluster::> set diag
cluster::*> vserver nfs modify -vserver vs0 -v4-numeric-ids enabled
cluster::*> vserver nfs show -vserver vs0 –fields v4-numeric-ids
vserver v4-numeric-ids
```

```
------- --------------
vs0     enabled
```

Disabling and Verifying ID Mapping on the Client

```
[root@localhost /]# cat /etc/idmapd.conf
[General]
#Verbosity = 0
# The following should be set to the local NFSv4 domain name
# The default is the host's DNS domain name.
Domain = local.domain.edu

[root@localhost /]# cat /sys/module/nfs/parameters/nfs4_disable_idmapping
Y

[root@localhost /]# mount -t nfs -o nfsvers=4 10.63.17.87:/vol/nfs /mnt/nfsv4
[root@localhost /]# cd /mnt/nfsv4
[root@localhost nfsv4]# ls -al
total 12
drwxrwxrwt 2 nobody bin   4096 Nov 10 17:27 .
drwxr-xr-x 5 root   root  4096 Nov  9 21:01 ..
```

Following are two test cases in which the users “test” and “mock-build,” creating files without using ID domain mapping just by using UID/GID.

```
[root@localhost nfsv4]# su - test   <-- lets test a REAL user...
[test@localhost ~]$ id
uid=500(test) gid=500(test) groups=500(test)
[test@localhost ~]$ cd /mnt/nfsv4
[test@localhost nfsv4]$ ls -al
total 12
drwxrwxrwt 2 nobody bin   4096 Nov 11 20:20 .
drwxr-xr-x 5 root   root  4096 Nov  9 21:01 ..
```

```
[test@localhost nfsv4]$ touch 1231

[test@localhost nfsv4]$ ls -al
total 12
drwxrwxrwt 2 nobody bin   4096 Nov 11 20:21 .
drwxr-xr-x 5 root   root  4096 Nov  9 21:01 ..
-rw-rw-r-- 1 test   test     0 Nov 11 20:21 1231


[root@localhost nfsv4]# su - mockbuild
[mockbuild@localhost ~]$ cd /mnt/nfsv4
[mockbuild@localhost nfsv4]$ touch mockbird
[mockbuild@localhost nfsv4]$ ls -al
total 12
drwxrwxrwt 2 nobody    bin       4096 Nov 11 20:22 .
drwxr-xr-x 5 root      root      4096 Nov  9 21:01 ..
-rw-rw-r-- 1 test      test         0 Nov 11 20:21 1231
-rw-rw-r-- 1 mockbuild mockbuild    0 Nov 11 20:22 mockbird
```

Because ID domain mapping is not used, the ID mapping falls back to classic UID/GID-style mapping, eliminating the need for an NFSv4 ID domain. However, in large environments, NetApp recommends a centralized name repository for NFSv4.x.

References

- TR-3967: Deployment and Best Practices Guide for Data ONTAP 8.1 Clustered Data ONTAP Windows File Services
- RFC 2203 – RPCSEC_GSS Protocol Specification

- RFC 3530 – Network File System (NFS) Version 4 Protocol
- RFC 5661 – Network File System (NFS) Version 4 Minor Version 1 Protocol
- TR-3580 – NFSv4 Enhancements and Best Practices Guide – Data ONTAP Implementation

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.

NetApp provides no representations or warranties regarding the accuracy, reliability, or serviceability of any information or recommendations provided in this publication, or with respect to any results that may be obtained by the use of the information or observance of any recommendations provided herein. The information in this document is distributed AS IS, and the use of this information or the implementation of any recommendations or techniques herein is a customer's responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. This document and the information contained herein may be used solely in connection with the NetApp products discussed in this document.

Go further, faster®

www.netapp.com