# Contents

**Deciding whether to use this guide** ............................................................. 7  
**Understanding NAS file access** ................................................................. 8  
  Namespaces and junction points ................................................................ 8  
  What the typical NAS namespace architectures are .............................. 9  
  How ONTAP controls access to files ....................................................... 11  
  Authentication-based restrictions ......................................................... 12  
  File-based restrictions ........................................................................ 12  
  How ONTAP handles NFS client authentication ................................. 12  
  How ONTAP uses name services ............................................................ 13  
  How ONTAP grants CIFS file access from NFS clients ....................... 13  
  How the NFS credential cache works ................................................... 13  
**Creating and managing data volumes in NAS namespaces** ............... 15  
  Creating data volumes with specified junction points .......................... 15  
  Creating data volumes without specifying junction points .................. 16  
  Mounting or unmounting existing volumes in the NAS namespace ....... 17  
  Displaying volume mount and junction point information .................... 18  
**Configuring security styles** ..................................................................... 19  
  How security styles affect data access .................................................. 19  
  What the security styles and their effects are ...................................... 19  
  Where and when to set security styles ............................................... 20  
  How to decide which security style to use on SVMs ......................... 20  
  How security style inheritance works ............................................... 20  
  How ONTAP preserves UNIX permissions ......................................... 21  
  How to manage UNIX permissions using the Windows Security tab .... 21  
  Configuring security styles on SVM root volumes .............................. 21  
  Configuring security styles on FlexVol volumes .................................. 22  
  Configuring security styles on qtrees ................................................ 22  
**Setting up file access using NFS** ............................................................... 24  
  Securing NFS access using export policies ........................................... 24  
  How export policies control client access to volumes or qtrees .......... 24  
  Default export policy for SVMs ............................................................ 25  
  How export rules work ....................................................................... 25  
  How to handle clients with an unlisted security type ......................... 26  
  How security types determine client access levels ............................. 28  
  How to handle superuser access requests .......................................... 30  
  How ONTAP uses export policy caches ............................................. 31  
  How the access cache works .............................................................. 32  
  How access cache parameters work ............................................... 33  
  Removing an export policy from a qtree ............................................ 33  
  Validating qtree IDs for qtree file operations .................................... 34  
  Export policy restrictions and nested junctions for FlexVol volumes .. 34
Commands for managing export rules ................................................................. 70
Configuring the NFS credential cache ............................................................... 70
  Reasons for modifying the NFS credential cache time-to-live .................... 70
  Configuring the time-to-live for cached NFS user credentials .................... 71
Managing export policy caches ............................................................................. 72
  Flushing export policy caches ........................................................................ 72
  Displaying the export policy netgroup queue and cache ............................... 73
  Checking whether a client IP address is a member of a netgroup ................. 74
  Optimizing access cache performance .......................................................... 75
Managing file locks ............................................................................................. 75
  About file locking between protocols ........................................................... 76
  How ONTAP treats read-only bits ................................................................. 76
  How ONTAP differs from Windows on handling locks on share path
    components .............................................................................................. 77
  Displaying information about locks ............................................................. 77
  Breaking locks .............................................................................................. 78
How FPolicy first-read and first-write filters work with NFS ............................... 79
Modifying the NFSv4.1 server implementation ID ............................................ 79
Managing NFSv4 ACLs ....................................................................................... 80
  Benefits of enabling NFSv4 ACLs ................................................................. 80
  How NFSv4 ACLs work .................................................................................. 81
  Enabling or disabling modification of NFSv4 ACLs ..................................... 81
  How ONTAP uses NFSv4 ACLs to determine whether it can delete a
    file ........................................................................................................... 82
  Enabling or disabling NFSv4 ACLs ............................................................... 82
Managing NFSv4 file delegations ....................................................................... 83
  Enabling or disabling NFSv4 read file delegations ...................................... 83
  Enabling or disabling NFSv4 write file delegations ..................................... 83
Configuring NFSv4 file and record locking ....................................................... 84
  About NFSv4 file and record locking ......................................................... 84
  Specifying the NFSv4 locking lease period ................................................ 84
  Specifying the NFSv4 locking grace period ................................................. 85
How NFSv4 referrals work .................................................................................. 85
Enabling or disabling NFSv4 referrals ............................................................. 86
Displaying NFS statistics .................................................................................. 86
Displaying DNS statistics ................................................................................. 87
Displaying NIS statistics .................................................................................. 89
Support for VMware vStorage over NFS ......................................................... 90
Enabling or disabling VMware vStorage over NFS ....................................... 91
Enabling or disabling rquota support .............................................................. 91
NFSv3 and NFSv4 performance improvement by modifying the TCP transfer
  size ............................................................................................................... 92
Modifying the NFSv3 and NFSv4 TCP maximum transfer size ....................... 92
Configuring the number of group IDs allowed for NFS users .......................... 93
Controlling root user access to NTFS security-style data ................................ 95
Deciding whether to use the NFS Reference

This guide describes ONTAP file access features available for the NFS protocol. It shows you how to enable an NFS server and how to export volumes or qtrees. It includes planning guidance and detailed implementation instructions.

You should use this guide under the following circumstances:

• You want to understand the range of ONTAP NFS protocol capabilities.
• You want to perform less common configuration and maintenance tasks, not basic NFS configuration.
• You want to use the command-line interface (CLI), not OnCommand System Manager or an automated scripting tool.

If you want to create a basic configuration using best practices, you should choose among the following documentation:

• NFSv3 Configuration Express Guide (basic configuration using OnCommand System Manager)
  NFS express configuration
• NFS Configuration Power Guide (advanced configuration using the CLI)
  NFS configuration

If you require additional configuration or conceptual information, you should choose among the following documentation:

• Conceptual background for NFS features
  ONTAP concepts
• Network and LIF management
  Network and LIF management
• Auditing configuration and management using native and external server implementations
  SMB/CIFS and NFS auditing and security tracing
• Command reference
  ONTAP 9 commands
• Automation of management tasks
  NetApp Documentation: OnCommand Workflow Automation (current releases)
• Technical Reports (TRs), which include additional information about ONTAP technology and interaction with external services
**Understanding NAS file access**

You can configure file servers on an SVM that allow NAS clients to access storage over NFS, CIFS, or both concurrently. In a multiprotocol configuration, clients access files regardless of the protocol they are connecting with or the type of authentication they require.

The NFS and CIFS servers you create are logical entities that enable the SVM to present a single namespace to clients. The namespace offers access to file storage on many cluster nodes via a single NFS mount or CIFS share.

**Namespaces and junction points**

A NAS namespace is a logical grouping of volumes joined together at junction points to create a single file system hierarchy. A client with sufficient permissions can access files in the namespace without specifying the location of the files in storage. Junctioned volumes can reside anywhere in the cluster.

Rather than mounting every volume containing a file of interest, NAS clients mount an NFS export or access an SMB share. The export or share represents the entire namespace or an intermediate location within the namespace. The client accesses only the volumes mounted below its access point.

You can add volumes to the namespace as needed. You can create junction points directly below a parent volume junction or on a directory within a volume. A path to a volume junction for a volume named “vol3” might be `/vol1/vol2/vol3`, or `/vol1/dir2/vol3`, or even `/dir1/dir2/vol3`. The path is called the junction path.

Every SVM has a unique namespace. The SVM root volume is the entry point to the namespace hierarchy.

**Note:** To ensure that data remains available in the event of a node outage or failover, you should create a load-sharing mirror copy for the SVM root volume.
Example

The following example creates a volume named “home4” located on SVM vs1 that has a junction path /eng/home:

```
cluster1::> volume create -vserver vs1 -volume home4 -aggregate aggr1 -size 1g -junction-path /eng/home
[Job 1642] Job succeeded: Successful
```

What the typical NAS namespace architectures are

There are several typical NAS namespace architectures that you can use as you create your SVM name space. You can choose the namespace architecture that matches your business and workflow needs.

The top of the namespace is always the root volume, which is represented by a slash (/). The namespace architecture under the root falls into three basic categories:

- A single branched tree, with only a single junction to the root of the namespace
- Multiple branched trees, with multiple junction points to the root of the namespace
- Multiple stand-alone volumes, each with a separate junction point to the root of the name space

Namespace with single branched tree

An architecture with a single branched tree has a single insertion point to the root of the SVM namespace. The single insertion point can be either a junctioned volume or a directory beneath the root. All other volumes are mounted at junction points beneath the single insertion point (which can be a volume or a directory).

For example, a typical volume junction configuration with the above namespace architecture might look like the following configuration, where all volumes are junctioned below the single insertion point, which is a directory named “data”: 
Namespace with multiple branched trees

An architecture with multiple branched trees has multiple insertion points to the root of the SVM namespace. The insertion points can be either junctioned volumes or directories beneath the root. All other volumes are mounted at junction points beneath the insertion points (which can be volumes or directories).

For example, a typical volume junction configuration with the above namespace architecture might look like the following configuration, where there are three insertion points to the root volume of the SVM. Two insertion points are directories named “data” and “projects”. One insertion point is a junctioned volume named “audit”:

<table>
<thead>
<tr>
<th>Vserver</th>
<th>Volume</th>
<th>Active</th>
<th>Junction Path</th>
<th>Path Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>vs1</td>
<td>audit</td>
<td>true</td>
<td>/audit</td>
<td>RW_volume</td>
</tr>
<tr>
<td>vs1</td>
<td>audit_logs1</td>
<td>true</td>
<td>/audit/logs1</td>
<td>RW_volume</td>
</tr>
<tr>
<td>vs1</td>
<td>audit_logs2</td>
<td>true</td>
<td>/audit/logs2</td>
<td>RW_volume</td>
</tr>
<tr>
<td>vs1</td>
<td>audit_logs3</td>
<td>true</td>
<td>/audit/logs3</td>
<td>RW_volume</td>
</tr>
<tr>
<td>vs1</td>
<td>eng</td>
<td>true</td>
<td>/data/eng</td>
<td>RW_volume</td>
</tr>
<tr>
<td>vs1</td>
<td>mktg1</td>
<td>true</td>
<td>/data/mktg1</td>
<td>RW_volume</td>
</tr>
</tbody>
</table>

(1) SVM root

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

1

2

3

4

5

6

7

8

9

10

| NFS Reference |
Namespace with multiple stand-alone volumes

In an architecture with stand-alone volumes, every volume has an insertion point to the root of the SVM namespace; however, the volume is not junctioned below another volume. Each volume has a unique path, and is either junctioned directly below the root or is junctioned under a directory below the root.

For example, a typical volume junction configuration with the above namespace architecture might look like the following configuration, where there are five insertion points to the root volume of the SVM, with each insertion point representing a path to one volume.

<table>
<thead>
<tr>
<th>Vserver</th>
<th>Volume</th>
<th>Active</th>
<th>Junction Path</th>
<th>Path Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>vs1</td>
<td>eng</td>
<td>true</td>
<td>/eng</td>
<td>RW_volume</td>
</tr>
<tr>
<td>vs1</td>
<td>mktg</td>
<td>true</td>
<td>/vol/mktg</td>
<td>RW_volume</td>
</tr>
<tr>
<td>vs1</td>
<td>project1</td>
<td>true</td>
<td>/project1</td>
<td>RW_volume</td>
</tr>
<tr>
<td>vs1</td>
<td>project2</td>
<td>true</td>
<td>/project2</td>
<td>RW_volume</td>
</tr>
<tr>
<td>vs1</td>
<td>sales</td>
<td>true</td>
<td>/sales</td>
<td>RW_volume</td>
</tr>
<tr>
<td>vs1</td>
<td>vs1_root</td>
<td>-</td>
<td>/</td>
<td>-</td>
</tr>
</tbody>
</table>

How ONTAP controls access to files

ONTAP controls access to files according to the authentication-based and file-based restrictions that you specify.

When a client connects to the storage system to access files, ONTAP has to perform two tasks:

- Authentication
ONTAP has to authenticate the client by verifying the identity with a trusted source. In addition, the authentication type of the client is one method that can be used to determine whether a client can access data when configuring export policies (optional for CIFS).

- **Authorization**
  ONTAP has to authorize the user by comparing the user's credentials with the permissions configured on the file or directory and determining what type of access, if any, to provide.

To properly manage file access control, ONTAP must communicate with external services such as NIS, LDAP, and Active Directory servers. Configuring a storage system for file access using CIFS or NFS requires setting up the appropriate services depending on your environment in ONTAP.

**Authentication-based restrictions**

With authentication-based restrictions, you can specify which client machines and which users can connect to the storage virtual machine (SVM).

ONTAP supports Kerberos authentication from both UNIX and Windows servers.

**File-based restrictions**

ONTAP evaluates three levels of security to determine whether an entity is authorized to perform a requested action on files and directories residing on an SVM. Access is determined by the effective permissions after evaluation of the three security levels.

Any storage object can contain up to three types of security layers:

- **Export (NFS) and share (SMB) security**
  Export and share security applies to client access to a given NFS export or SMB share. Users with administrative privileges can manage export and share-level security from SMB and NFS clients.

- **Storage-Level Access Guard file and directory security**
  Storage-Level Access Guard security applies to SMB and NFS client access to SVM volumes. Only NTFS access permissions are supported. For ONTAP to perform security checks on UNIX users for access to data on volumes for which Storage-Level Access Guard has been applied, the UNIX user must map to a Windows user on the SVM that owns the volume.

  **Note:** If you view the security settings on a file or directory from an NFS or SMB client, you will not see Storage-Level Access Guard security. Storage-Level Access Guard security cannot be revoked from a client, even by a system (Windows or UNIX) administrator.

- **NTFS, UNIX, and NFSv4 native file-level security**
  Native file-level security exists on the file or directory that represents the storage object. You can set file-level security from a client. File permissions are effective regardless of whether SMB or NFS is used to access the data.

**How ONTAP handles NFS client authentication**

NFS clients must be properly authenticated before they can access data on the SVM. ONTAP authenticates the clients by checking their UNIX credentials against the name services that you configure.

When an NFS client connects to the SVM, ONTAP obtains the UNIX credentials for the user by checking different name services, depending on the name services configuration of the SVM. ONTAP can check credentials for local UNIX accounts, NIS domains, and LDAP domains. At least one of them must be configured so that ONTAP can successfully authenticate the user. You can specify multiple name services and the order in which ONTAP searches them.

In a pure NFS environment with UNIX volume security styles, this configuration is sufficient to authenticate and provide the proper file access for a user connecting from an NFS client.
If you are using mixed, NTFS, or unified volume security styles, ONTAP must obtain a CIFS user name for the UNIX user for authentication with a Windows domain controller. This can happen either by mapping individual users using local UNIX accounts or LDAP domains, or by using a default CIFS user instead. You can specify which name services ONTAP searches in which order, or specify a default CIFS user.

**How ONTAP uses name services**

ONTAP uses name services to obtain information about users and clients. ONTAP uses this information to authenticate users accessing data on or administering the storage system, and to map user credentials in a mixed environment.

When you configure the storage system, you must specify what name services you want ONTAP to use for obtaining user credentials for authentication. ONTAP supports the following name services:

- Local users (file)
- External NIS domains (NIS)
- External LDAP domains (LDAP)

You use the `vserver services name-service ns-switch` command family to configure SVMs with the sources to search for network information and the order in which to search them. These commands provide the equivalent functionality of the `/etc/nsswitch.conf` file on UNIX systems.

When an NFS client connects to the SVM, ONTAP checks the specified name services to obtain the UNIX credentials for the user. If name services are configured correctly and ONTAP can obtain the UNIX credentials, ONTAP successfully authenticates the user.

In an environment with mixed security styles, ONTAP might have to map user credentials. You must configure name services appropriately for your environment to allow ONTAP to properly map user credentials.

ONTAP also uses name services for authenticating SVM administrator accounts. You must keep this in mind when configuring or modifying the name service switch to avoid accidentally disabling authentication for SVM administrator accounts. For more information about SVM administration users, see the *Administrator Authentication and RBAC Power Guide*.

**Administrator authentication and RBAC**

**How ONTAP grants CIFS file access from NFS clients**

ONTAP uses Windows NT File System (NTFS) security semantics to determine whether a UNIX user, on an NFS client, has access to a file with NTFS permissions.

ONTAP does this by converting the user’s UNIX User ID (UID) into a CIFS credential, and then using the CIFS credential to verify that the user has access rights to the file. A CIFS credential consists of a primary Security Identifier (SID), usually the user’s Windows user name, and one or more group SIDs that correspond to Windows groups of which the user is a member.

The time ONTAP takes converting the UNIX UID into a CIFS credential can be from tens of milliseconds to hundreds of milliseconds because the process involves contacting a domain controller. ONTAP maps the UID to the CIFS credential and enters the mapping in a credential cache to reduce the verification time caused by the conversion.

**How the NFS credential cache works**

When an NFS user requests access to NFS exports on the storage system, ONTAP must retrieve the user credentials either from external name servers or from local files to authenticate the user. ONTAP
then stores these credentials in an internal credential cache for later reference. Understanding how the
NFS credential caches works enables you to handle potential performance and access issues.

Without the credential cache, ONTAP would have to query name services every time an NFS user
requested access. On a busy storage system that is accessed by many users, this can quickly lead to
serious performance problems, causing unwanted delays or even denials to NFS client access.

With the credential cache, ONTAP retrieves the user credentials and then stores them for a
predetermined amount of time for quick and easy access should the NFS client send another request.
This method offers the following advantages:

• It eases the load on the storage system by handling fewer requests to external name servers (such
  as NIS or LDAP).
• It eases the load on external name servers by sending fewer requests to them.
• It speeds up user access by eliminating the wait time for obtaining credentials from external
  sources before the user can be authenticated.

ONTAP stores both positive and negative credentials in the credential cache. Positive credentials
means that the user was authenticated and granted access. Negative credentials means that the user
was not authenticated and was denied access.

By default, ONTAP stores positive credentials for 24 hours; that is, after initially authenticating a
user, ONTAP uses the cached credentials for any access requests by that user for 24 hours. If the user
requests access after 24 hours, the cycle starts over: ONTAP discards the cached credentials and
obtains the credentials again from the appropriate name service source. If the credentials changed on
the name server during the previous 24 hours, ONTAP caches the updated credentials for use for the
next 24 hours.

By default, ONTAP stores negative credentials for two hours; that is, after initially denying access to
a user, ONTAP continues to deny any access requests by that user for two hours. If the user requests
access after 2 hours, the cycle starts over: ONTAP obtains the credentials again from the appropriate
name service source. If the credentials changed on the name server during the previous two hours,
ONTAP caches the updated credentials for use for the next two hours.
Creating and managing data volumes in NAS namespaces

To manage file access in a NAS environment, you must manage data volumes and junction points on your storage virtual machine (SVM). This includes planning your namespace architecture, creating volumes with or without junction points, mounting or unmounting volumes, and displaying information about data volumes and NFS server or CIFS server namespaces.

Related concepts

Namespaces and junction points on page 8

Creating data volumes with specified junction points

You can specify the junction point when you create a data volume. The resultant volume is automatically mounted at the junction point and is immediately available to configure for NAS access.

Before you begin

The aggregate in which you want to create the volume must already exist.

Note: The following characters cannot be used in the junction path:

* # " > < | ? \n
In addition, the junction path length cannot be more than 255 characters.

Steps

1. Create the volume with a junction point:

   ```
   volume create -vserver vserver_name -volume volume_name -aggregate aggregate_name -size {integer|KB|MB|GB|TB|PB} -security-style {ntfs|unix|mixed} -junction-path junction_path
   ```

   The junction path must start with the root (/) and can contain both directories and junctioned volumes. The junction path does not need to contain the name of the volume. Junction paths are independent of the volume name.

   Specifying a volume security style is optional. If you do not specify a security style, ONTAP creates the volume with the same security style that is applied to the root volume of the storage virtual machine (SVM). However, the root volume's security style might not be the security style you want applied to the data volume you create. The recommendation is to specify the security style when you create the volume to minimize difficult-to-troubleshoot file-access issues.

   The junction path is case insensitive; /ENG is the same as /eng. If you create a CIFS share, Windows treats the junction path as if it is case sensitive. For example, if the junction is /ENG, the path of a CIFS share must start with /ENG, not /eng.

   There are many optional parameters that you can use to customize a data volume. To learn more about them, see the man pages for the `volume create` command.

2. Verify that the volume was created with the desired junction point:

   ```
   volume show -vserver vserver_name -volume volume_name -junction
   ```
Creating data volumes without specifying junction points

You can create a data volume without specifying a junction point. The resultant volume is not automatically mounted, and is not available to configure for NAS access. You must mount the volume before you can configure SMB shares or NFS exports for that volume.

Before you begin

The aggregate in which you want to create the volume must already exist.

Steps

1. Create the volume without a junction point by using the following command:

   ```
   volume create -vserver vserver_name -volume volume_name -aggregate aggregate_name -size {integer[KB|MB|GB|TB|PB]} -security-style {ntfs|unix|mixed}
   ```

   Specifying a volume security style is optional. If you do not specify a security style, ONTAP creates the volume with the same security style that is applied to the root volume of the storage virtual machine (SVM). However, the root volume's security style might not be the security style you want applied to the data volume. The recommendation is to specify the security style when you create the volume to minimize difficult-to-troubleshoot file-access issues.

   There are many optional parameters that you can use to customize a data volume. To learn more about them, see the man pages for the `volume create` command.

2. Verify that the volume was created without a junction point:

   ```
   volume show -vserver vserver_name -volume volume_name -junction
   ```

   Example

   The following example creates a volume named “sales” located on SVM vs1 that is not mounted at a junction point:

   ```
   cluster1::> volume create -vserver vs1 -volume sales -aggregate aggr3 -size 20GB
   [Job 3406] Job succeeded: Successful
   ```

   ```
   cluster1::> volume show -vserver vs1 -junction
   ```
Mounting or unmounting existing volumes in the NAS namespace

A volume must be mounted on the NAS namespace before you can configure NAS client access to data contained in the storage virtual machine (SVM) volumes. You can mount a volume to a junction point if it is not currently mounted. You can also unmount volumes.

About this task

If you unmount a volume, all data within the junction point, including data in volumes with junction points contained within the unmounted volume's namespace, are inaccessible to NAS clients. When you unmount a volume, data within the volume is not lost. Additionally, existing volume export policies and SMB shares created on the volume or on directories and junction points within the unmounted volume are retained. If you remount the unmounted volume, NAS clients can access the data contained within the volume using existing export policies and SMB shares.

Steps

1. Perform the desired action:

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Enter the command...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mount a volume</td>
<td>volume mount -vserver vserver_name -volume volume_name -junction-path junction_path</td>
</tr>
<tr>
<td>Unmount a volume</td>
<td>volume unmount -vserver vserver_name -volume volume_name</td>
</tr>
</tbody>
</table>

2. Verify that the volume is in the desired mount state:

   volume show -vserver vserver_name -volume volume_name -junction

Examples

The following example mounts a volume named “sales” located on SVM vs1 to the junction point /sales:

```
cluster1::> volume mount -vserver vs1 -volume sales -junction-path /sales
```

The following example unmounts a volume named “data” located on SVM vs1:

```
cluster1::> volume unmount -vserver vs1 -volume data
```
Displaying volume mount and junction point information

You can display information about mounted volumes for storage virtual machines (SVMs) and the junction points to which the volumes are mounted. You can also determine which volumes are not mounted to a junction point. You can use this information to understand and manage your SVM namespace.

**Step**

1. Perform the desired action:

<table>
<thead>
<tr>
<th>If you want to display...</th>
<th>Enter the command...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary information about mounted and unmounted volumes on the SVM</td>
<td><code>volume show -vserver vserver_name -junction</code></td>
</tr>
<tr>
<td>Detailed information about mounted and unmounted volumes on the SVM</td>
<td><code>volume show -vserver vserver_name -volume volume_name -instance</code></td>
</tr>
</tbody>
</table>
| Specific information about mounted and unmounted volumes on the SVM | a. If necessary, you can display valid fields for the `-fields` parameter by using the following command: `volume show -fields ?`
| | b. Display the desired information by using the `-fields` parameter: `volume show -vserver vserver_name -fields fieldname,...` |

**Examples**

The following example displays a summary of mounted and unmounted volumes on SVM vs1:

```
cluster1::> volume show -vserver vs1 -junction

<table>
<thead>
<tr>
<th>Vserver</th>
<th>Volume</th>
<th>Junction</th>
<th>Active</th>
<th>Junction Path</th>
<th>Path Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>vs1</td>
<td>data</td>
<td>true</td>
<td>/data</td>
<td>RW_volume</td>
<td></td>
</tr>
<tr>
<td>vs1</td>
<td>home4</td>
<td>true</td>
<td>/eng/home</td>
<td>RW_volume</td>
<td></td>
</tr>
<tr>
<td>vs1</td>
<td>vs1_root</td>
<td>-</td>
<td>/</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>vs1</td>
<td>sales</td>
<td>true</td>
<td>/sales</td>
<td>RW_volume</td>
<td></td>
</tr>
</tbody>
</table>
```

The following example displays information about specified fields for volumes located on SVM vs2:

```
cluster1::> volume show -vserver vs2 -fields vserver,volume,aggregate,size,state,type,security-style,junction-path,junction-parent,node

<table>
<thead>
<tr>
<th>vserver</th>
<th>volume</th>
<th>aggregate</th>
<th>size</th>
<th>state</th>
<th>type</th>
<th>security-style</th>
<th>junction-path</th>
<th>junction-parent</th>
<th>node</th>
</tr>
</thead>
<tbody>
<tr>
<td>vs2</td>
<td>data1</td>
<td>aggr3</td>
<td>2GB</td>
<td>online</td>
<td>RW</td>
<td>unix</td>
<td>/data2</td>
<td>vs2_root</td>
<td>node3</td>
</tr>
<tr>
<td>vs2</td>
<td>data2</td>
<td>aggr3</td>
<td>1GB</td>
<td>online</td>
<td>RW</td>
<td>ntfs</td>
<td>/data2/d2_1</td>
<td>vs2_root</td>
<td>node3</td>
</tr>
<tr>
<td>vs2</td>
<td>data2_1</td>
<td>aggr3</td>
<td>8GB</td>
<td>online</td>
<td>RW</td>
<td>ntfs</td>
<td>/data2/d2_2</td>
<td>data2</td>
<td>node3</td>
</tr>
<tr>
<td>vs2</td>
<td>pubs</td>
<td>aggr1</td>
<td>1GB</td>
<td>online</td>
<td>RW</td>
<td>unix</td>
<td>/publications</td>
<td>vs2_root</td>
<td>node1</td>
</tr>
<tr>
<td>vs2</td>
<td>images</td>
<td>aggr3</td>
<td>2TB</td>
<td>online</td>
<td>RW</td>
<td>ntfs</td>
<td>/images</td>
<td>vs2_root</td>
<td>node3</td>
</tr>
<tr>
<td>vs2</td>
<td>logs</td>
<td>aggr1</td>
<td>1GB</td>
<td>online</td>
<td>RW</td>
<td>unix</td>
<td>/logs</td>
<td>vs2_root</td>
<td>node1</td>
</tr>
<tr>
<td>vs2</td>
<td>vs2_root</td>
<td>aggr3</td>
<td>1GB</td>
<td>online</td>
<td>RW</td>
<td>ntfs</td>
<td>/</td>
<td>-</td>
<td>node3</td>
</tr>
</tbody>
</table>
```
Configuring security styles

You configure security styles on FlexVol volumes and qtrees to determine the type of permissions ONTAP uses to control access and what client type can modify these permissions.

For information about the security style of Infinite Volumes, see the Infinite Volumes Management Guide.

Infinite volumes management

How security styles affect data access

Each volume and qtree on the storage system has a security style. The security style determines what type of permissions are used for data on volumes when authorizing users. Security styles can differ between volume types.

What the security styles and their effects are

There are four different security styles: UNIX, NTFS, mixed, and unified. Each security style has a different effect on how permissions are handled for data. You must understand the different effects to ensure that you select the appropriate security style for your purposes.

It is important to understand that security styles do not determine what client types can or cannot access data. Security styles only determine the type of permissions ONTAP uses to control data access and what client type can modify these permissions.

For example, if a volume uses UNIX security style, SMB clients can still access data (provided that they properly authenticate and authorize) due to the multiprotocol nature of ONTAP. However, ONTAP uses UNIX permissions that only UNIX clients can modify using native tools.

<table>
<thead>
<tr>
<th>Security style</th>
<th>Clients that can modify permissions</th>
<th>Permissions that clients can use</th>
<th>Resulting effective security style</th>
<th>Clients that can access files</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNIX</td>
<td>NFS</td>
<td>NFSv3 mode bits</td>
<td>UNIX</td>
<td>NFS and SMB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NFSv4.x ACLs</td>
<td>UNIX</td>
<td></td>
</tr>
<tr>
<td>NTFS</td>
<td>SMB</td>
<td>NTFS ACLs</td>
<td>NTFS</td>
<td></td>
</tr>
<tr>
<td>Mixed</td>
<td>NFS or SMB</td>
<td>NFSv3 mode bits</td>
<td>UNIX</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>NFSv4.x ACLs</td>
<td>UNIX</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>NTFS ACLs</td>
<td>NTFS</td>
<td></td>
</tr>
<tr>
<td>Unified</td>
<td>NFS or SMB</td>
<td>NFSv3 mode bits</td>
<td>UNIX</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>NFSv4.1 ACLs</td>
<td>UNIX</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>NTFS ACLs</td>
<td>NTFS</td>
<td></td>
</tr>
</tbody>
</table>

FlexVol volumes support UNIX, NTS, and mixed security styles. When the security style is mixed or unified, the effective permissions depend on the client type that last modified the permissions because users set the security style on an individual basis. If the last client that modified permissions was an NFSv3 client, the permissions are UNIX NFSv3 mode bits. If the last client was an NFSv4 client, the permissions are NFSv4 ACLs. If the last client was an SMB client, the permissions are Windows NTFS ACLs.
Note: ONTAP initially sets some default file permissions. By default, the effective security style on all data in UNIX, mixed, and unified security style volumes is UNIX and the effective permissions type is UNIX mode bits (0755 unless specified otherwise) until configured by a client as allowed by the default security style. By default, the effective security style on all data in NTFS security style volumes is NTFS and has an ACL allowing full control to everyone.

Where and when to set security styles

Security styles can be set on FlexVol volumes (both root or data volumes) and qtrees. Security styles can be set manually at the time of creation, inherited automatically, or changed at a later time.

How to decide which security style to use on SVMs

To help you decide which security style to use on a volume, you should consider two factors. The primary factor is the type of administrator that manages the file system. The secondary factor is the type of user or service that accesses the data on the volume.

When you configure the security style on a volume, you should consider the needs of your environment to ensure that you select the best security style and avoid issues with managing permissions. The following considerations can help you decide:

<table>
<thead>
<tr>
<th>Security style</th>
<th>Choose if...</th>
</tr>
</thead>
</table>
| UNIX           | • The file system is managed by a UNIX administrator.  
• The majority of users are NFS clients.  
• An application accessing the data uses a UNIX user as the service account. |
| NTFS           | • The file system is managed by a Windows administrator.  
• The majority of users are SMB clients.  
• An application accessing the data uses a Windows user as the service account. |
| Mixed          | The file system is managed by both UNIX and Windows administrators and users consist of both NFS and SMB clients. |

How security style inheritance works

If you do not specify the security style when creating a new FlexVol volume or a qtree, it inherits its security style in different ways.

Security styles are inherited in the following manner:

• A FlexVol volume inherits the security style of the root volume of its containing SVM.
• A qtree inherits the security style of its containing FlexVol volume.
• A file or directory inherits the security style of its containing FlexVol volume or qtree.
How ONTAP preserves UNIX permissions

When files in a FlexVol volume that currently have UNIX permissions are edited and saved by Windows applications, ONTAP can preserve the UNIX permissions.

When applications on Windows clients edit and save files, they read the security properties of the file, create a new temporary file, apply those properties to the temporary file, and then give the temporary file the original file name.

When Windows clients perform a query for the security properties, they receive a constructed ACL that exactly represents the UNIX permissions. The sole purpose of this constructed ACL is to preserve the file's UNIX permissions as files are updated by Windows applications to ensure that the resulting files have the same UNIX permissions. ONTAP does not set any NTFS ACLs using the constructed ACL.

How to manage UNIX permissions using the Windows Security tab

If you want to manipulate UNIX permissions of files or folders in mixed security-style volumes or qtrees on SVMs, you can use the Security tab on Windows clients. Alternatively, you can use applications that can query and set Windows ACLs.

• Modifying UNIX permissions
  You can use the Windows Security tab to view and change UNIX permissions for a mixed security-style volume or qtree. If you use the main Windows Security tab to change UNIX permissions, you must first remove the existing ACE you want to edit (this sets the mode bits to 0) before you make your changes. Alternatively, you can use the Advanced editor to change permissions.
  If mode permissions are used, you can directly change the mode permissions for the listed UID, GID, and others (everyone else with an account on the computer). For example, if the displayed UID has r-x permissions, you can change the UID permissions to rwx.

• Changing UNIX permissions to NTFS permissions
  You can use the Windows Security tab to replace UNIX security objects with Windows security objects on a mixed security-style volume or qtree where the files and folders have a UNIX effective security style.
  You must first remove all listed UNIX permission entries before you can replace them with the desired Windows User and Group objects. You can then configure NTFS-based ACLs on the Windows User and Group objects. By removing all UNIX security objects and adding only Windows Users and Groups to a file or folder in a mixed security-style volume or qtree, you change the effective security style on the file or folder from UNIX to NTFS.
  When changing permissions on a folder, the default Windows behavior is to propagate these changes to all subfolders and files. Therefore, you must change the propagation choice to the desired setting if you do not want to propagate a change in security style to all child folders, subfolders, and files.

Configuring security styles on SVM root volumes

You configure the storage virtual machine (SVM) root volume security style to determine the type of permissions used for data on the root volume of the SVM.

Steps

1. Use the vserver create command with the -rootvolume-security-style parameter to define the security style.

   The possible options for the root volume security style are unix, ntfs, or mixed. You cannot use unified security style because it only applies to Infinite Volumes.
2. Display and verify the configuration, including the root volume security style of the SVM you created:

```
vserver show -vserver vserver_name
```

### Configuring security styles on FlexVol volumes

You configure the FlexVol volume security style to determine the type of permissions used for data on FlexVol volumes of the storage virtual machine (SVM).

**Steps**

1. Perform one of the following actions:

<table>
<thead>
<tr>
<th>If the FlexVol volume...</th>
<th>Use the command...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does not yet exist</td>
<td><code>volume create</code> and include the <code>-security-style</code> parameter to specify the security style.</td>
</tr>
<tr>
<td>Already exists</td>
<td><code>volume modify</code> and include the <code>-security-style</code> parameter to specify the security style.</td>
</tr>
</tbody>
</table>

The possible options for the FlexVol volume security style are `unix`, `ntfs`, or `mixed`. You cannot use `unified` security style because it only applies to Infinite Volumes.

If you do not specify a security style when creating a FlexVol volume, the volume inherits the security style of the root volume.

For more information about the `volume create` or `volume modify` commands, see the *Logical Storage Management Guide*.

2. To display the configuration, including the security style of the FlexVol volume you created, enter the following command:

```
volume show -volume volume_name -instance
```

**Related information**

*Logical Storage Management Guide*

### Configuring security styles on qtrees

You configure the qtree volume security style to determine the type of permissions used for data on qtrees.

**Steps**

1. Perform one of the following actions:

<table>
<thead>
<tr>
<th>If the qtree...</th>
<th>Use the command...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does not exist yet</td>
<td><code>volume qtree create</code> and include the <code>-security-style</code> parameter to specify the security style.</td>
</tr>
<tr>
<td>Already exists</td>
<td><code>volume qtree modify</code> and include the <code>-security-style</code> parameter to specify the security style.</td>
</tr>
</tbody>
</table>

The possible options for the qtree security style are `unix`, `ntfs`, or `mixed`. You cannot use `unified` security style because it only applies to Infinite Volumes.

If you do not specify a security style when creating a qtree, the default security style is `mixed`. 
For more information about the `volume qtree create` or `volume qtree modify` commands, see the *Logical Storage Management Guide*.

2. To display the configuration, including the security style of the qtree you created, enter the following command:

   ```bash
   volume qtree show -qtree qtree_name -instance
   ```

Related information

*Logical Storage Management Guide*
Setting up file access using NFS

You must complete a number of steps to allow clients access to files on storage virtual machines (SVMs) using NFS. There are some additional steps that are optional depending on the current configuration of your environment.

For clients to be able to access files on SVMs using NFS, you must complete the following tasks:

1. Enable the NFS protocol on the SVM.
   You must configure the SVM to allow data access from clients over NFS.

2. Create an NFS server on the SVM.
   An NFS server is a logical entity on the SVM that enables the SVM to serve files over NFS. You must create the NFS server and specify the NFS protocol versions you want to allow.

3. Configure export policies on the SVM.
   You must configure export policies to make volumes and qtrees available to clients.

4. Configure the NFS server with the appropriate security and other settings depending on the network and storage environment.
   This step might include configuring Kerberos, LDAP, NIS, name mappings, and local users.

Securing NFS access using export policies

You can use export policies to restrict NFS access to volumes or qtrees to clients that match specific parameters. When provisioning new storage, you can use an existing policy and rules, add rules to an existing policy, or create a new policy and rules. You can also check the configuration of export policies.

**Note:** Beginning in ONTAP 9.3, you can enable export policy configuration checking as a background job that records any rules violations in an error rule list. The `vserver export-policy config-checker` commands invoke the checker and display results, which you can use to verify your configuration and delete erroneous rules from the policy.

The commands only validate export configuration for host names, netgroups, and anonymous users.

How export policies control client access to volumes or qtrees

Export policies contain one or more export rules that process each client access request. The result of the process determines whether the client is denied or granted access and what level of access. An export policy with export rules must exist on the storage virtual machine (SVM) for clients to access data.

You associate exactly one export policy with each volume or qtree to configure client access to the volume or qtree. The SVM can contain multiple export policies. This enables you to do the following for SVMs with multiple volumes or qtrees:

- Assign different export policies to each volume or qtree of the SVM for individual client access control to each volume or qtree in the SVM.
- Assign the same export policy to multiple volumes or qtrees of the SVM for identical client access control without having to create a new export policy for each volume or qtree.

If a client makes an access request that is not permitted by the applicable export policy, the request fails with a permission-denied message. If a client does not match any rule in the export policy, then access is denied. If an export policy is empty, then all accesses are implicitly denied.
You can modify an export policy dynamically on a system running ONTAP.

**Default export policy for SVMs**

Each SVM has a default export policy that contains no rules. An export policy with rules must exist before clients can access data on the SVM. Each FlexVol volume contained in the SVM must be associated with an export policy.

When you create an SVM, the storage system automatically creates a default export policy called `default` for the root volume of the SVM. You must create one or more rules for the default export policy before clients can access data on the SVM. Alternatively, you can create a custom export policy with rules. You can modify and rename the default export policy, but you cannot delete the default export policy.

When you create a FlexVol volume in its containing SVM, the storage system creates the volume and associates the volume with the default export policy for the root volume of the SVM. By default, each volume created in the SVM is associated with the default export policy for the root volume. You can use the default export policy for all volumes contained in the SVM, or you can create a unique export policy for each volume. You can associate multiple volumes with the same export policy.

**How export rules work**

Export rules are the functional elements of an export policy. Export rules match client access requests to a volume against specific parameters you configure to determine how to handle the client access requests.

An export policy must contain at least one export rule to allow access to clients. If an export policy contains more than one rule, the rules are processed in the order in which they appear in the export policy. The rule order is dictated by the rule index number. If a rule matches a client, the permissions of that rule are used and no further rules are processed. If no rules match, the client is denied access.

You can configure export rules to determine client access permissions using the following criteria:

- The file access protocol used by the client sending the request, for example, NFSv4 or SMB.
- A client identifier, for example, host name or IP address.
- The security type used by the client to authenticate, for example, Kerberos v5, NTLM, or AUTH_SYS.

If a rule specifies multiple criteria, and the client does not match one or more of them, the rule does not apply.

### Example

The export policy contains an export rule with the following parameters:

- `-protocol nfs3`
- `-clientmatch 10.1.16.0/255.255.255.0`
- `-rorule any`
- `-rwrule any`

The client access request is sent using the NFSv3 protocol and the client has the IP address 10.1.17.37.

Even though the client access protocol matches, the IP address of the client is in a different subnet from the one specified in the export rule. Therefore, client matching fails and this rule does not apply to this client.
Example
The export policy contains an export rule with the following parameters:

- -protocol nfs
- -clientmatch 10.1.16.0/255.255.255.0
- -rorule any
- -rwrule any

The client access request is sent using the NFSv4 protocol and the client has the IP address 10.1.16.54.

The client access protocol matches and the IP address of the client is in the specified subnet. Therefore, client matching is successful and this rule applies to this client. The client gets read-write access regardless of its security type.

Example
The export policy contains an export rule with the following parameters:

- -protocol nfs3
- -clientmatch 10.1.16.0/255.255.255.0
- -rorule any
- -rwrule krb5,ntlm

Client #1 has the IP address 10.1.16.207, sends an access request using the NFSv3 protocol, and authenticated with Kerberos v5.

Client #2 has the IP address 10.1.16.211, sends an access request using the NFSv3 protocol, and authenticated with AUTH_SYS.

The client access protocol and IP address matches for both clients. The read-only parameter allows read-only access to all clients regardless of the security type they authenticated with. Therefore both clients get read-only access. However, only client #1 gets read-write access because it used the approved security type Kerberos v5 to authenticate. Client #2 does not get read-write access.

How to handle clients with an unlisted security type

When a client presents itself with a security type that is not listed in an access parameter of an export rule, you have the choice of either denying access to the client or mapping it to the anonymous user ID instead by using the option `none` in the access parameter.

A client might present itself with a security type that is not listed in an access parameter because it was authenticated with a different security type or was not authenticated at all (security type AUTH_NONE). By default, the client is automatically denied access to that level. However, you can add the option `none` to the access parameter. As a result, clients with an unlisted security style are mapped to the anonymous user ID instead. The `-anon` parameter determines what user ID is assigned to those clients. The user ID specified for the `-anon` parameter must be a valid user that is configured with permissions you deem appropriate for the anonymous user.

Valid values for the `-anon` parameter range from 0 to 65535.
User ID assigned to -anon | Resulting handling of client access requests
--- | ---
0 - 65533 | The client access request is mapped to the anonymous user ID and gets access depending on the permissions configured for this user.
65534 | The client access request is mapped to the user nobody and gets access depending on the permissions configured for this user. This is the default.
65535 | The access request from any client is denied when mapped to this ID and the client presents itself with security type AUTH_NONE. The access request from clients with user ID 0 is denied when mapped to this ID and the client presents itself with any other security type.

When using the option `none`, it is important to remember that the read-only parameter is processed first. Consider the following guidelines when configuring export rules for clients with unlisted security types:

<table>
<thead>
<tr>
<th>Read-only includes none</th>
<th>Read-write includes none</th>
<th>Resulting access for clients with unlisted security types</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>No</td>
<td>Denied</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>Denied because read-only is processed first</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Read-only as anonymous</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Read-write as anonymous</td>
</tr>
</tbody>
</table>

**Example**

The export policy contains an export rule with the following parameters:

- `-protocol nfs3`
- `-clientmatch 10.1.16.0/255.255.255.0`
- `-rorule sys, none`
- `-rwrule any`
- `-anon 70`

Client #1 has the IP address 10.1.16.207, sends an access request using the NFSv3 protocol, and authenticated with Kerberos v5.

Client #2 has the IP address 10.1.16.211, sends an access request using the NFSv3 protocol, and authenticated with AUTH_SYS.

Client #3 has the IP address 10.1.16.234, sends an access request using the NFSv3 protocol, and did not authenticate (meaning security type AUTH_NONE).

The client access protocol and IP address matches for all three clients. The read-only parameter allows read-only access to clients with their own user ID that authenticated with AUTH_SYS. The read-only parameter allows read-only access as the anonymous user with user ID 70 to clients that authenticated using any other security type. The read-write parameter allows read-write access to any security type, but in this case only applies to clients already filtered by the read-only rule.
Therefore, clients #1 and #3 get read-write access only as the anonymous user with user ID 70. Client #2 gets read-write access with its own user ID.

**Example**

The export policy contains an export rule with the following parameters:

- `-protocol nfs3`
- `-clientmatch 10.1.16.0/255.255.255.0`
- `-rorule sys,none`
- `-rwrule none`
- `-anon 70`

Client #1 has the IP address 10.1.16.207, sends an access request using the NFSv3 protocol, and authenticated with Kerberos v5.

Client #2 has the IP address 10.1.16.211, sends an access request using the NFSv3 protocol, and authenticated with AUTH_SYS.

Client #3 has the IP address 10.1.16.234, sends an access request using the NFSv3 protocol, and did not authenticate (meaning security type AUTH_NONE).

The client access protocol and IP address matches for all three clients. The read-only parameter allows read-only access to clients with their own user ID that authenticated with AUTH_SYS. The read-only parameter allows read-only access as the anonymous user with user ID 70 to clients that authenticated using any other security type. The read-write parameter allows read-write access only as the anonymous user.

Therefore, client #1 and client #3 get read-write access only as the anonymous user with user ID 70. Client #2 gets read-only access with its own user ID but is denied read-write access.

---

**How security types determine client access levels**

The security type that the client authenticated with plays a special role in export rules. You must understand how the security type determines the levels of access the client gets to a volume or qtree.

The three possible access levels are as follows:

1. Read-only
2. Read-write
3. Superuser (for clients with user ID 0)

Because the access level by security type is evaluated in this order, you must observe the following rules when constructing access level parameters in export rules:

<table>
<thead>
<tr>
<th>For a client to get access level...</th>
<th>These access parameters must match the client's security type...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal user read-only</td>
<td>Read-only (-rorule)</td>
</tr>
<tr>
<td>Normal user read-write</td>
<td>Read-only (-rorule) and read-write (-rwrule)</td>
</tr>
<tr>
<td>Superuser read-only</td>
<td>Read-only (-rorule) and -superuser</td>
</tr>
<tr>
<td>Superuser read-write</td>
<td>Read-only (-rorule) and read-write (-rwrule) and -superuser</td>
</tr>
</tbody>
</table>
The following are valid security types for each of these three access parameters:

- any
- none
- never
  This security type is not valid for use with the `-superuser` parameter.
- krb5
- krb5i
- krb5p
- ntlm
- sys

When matching a client's security type against each of the three access parameters, there are three possible outcomes:

<table>
<thead>
<tr>
<th>If the client's security type...</th>
<th>Then the client...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matches the one specified in the access parameter.</td>
<td>Gets access for that level with its own user ID.</td>
</tr>
<tr>
<td>Does not match the one specified, but the access parameter includes the option <code>none</code>.</td>
<td>Gets access for that level but as the anonymous user with the user ID specified by the <code>-anon</code> parameter.</td>
</tr>
<tr>
<td>Does not match the one specified and the access parameter does not include the option <code>none</code>.</td>
<td>Does not get any access for that level. This does not apply to the <code>-superuser</code> parameter because it always includes <code>none</code> even when not specified.</td>
</tr>
</tbody>
</table>

**Example**

The export policy contains an export rule with the following parameters:

- `-protocol nfs3`
- `-clientmatch 10.1.16.0/255.255.255.0`
- `-rorule any`
- `-rrule sys,krb5`
- `-superuser krb5`

Client #1 has the IP address 10.1.16.207, has user ID 0, sends an access request using the NFSv3 protocol, and authenticated with Kerberos v5.

Client #2 has the IP address 10.1.16.211, has user ID 0, sends an access request using the NFSv3 protocol, and authenticated with AUTH_SYS.

Client #3 has the IP address 10.1.16.234, has user ID 0, sends an access request using the NFSv3 protocol, and did not authenticate (AUTH_NONE).

The client access protocol and IP address matches all three clients. The read-only parameter allows read-only access to all clients regardless of security type. The read-write parameter allows read-write access to clients with their own user ID that authenticated with AUTH_SYS or Kerberos v5. The superuser parameter allows superuser access to clients with user ID 0 that authenticated with Kerberos v5.
Therefore, client #1 gets superuser read-write access because it matches all three access parameters. Client #2 gets read-write access but not superuser access. Client #3 gets read-only access but not superuser access.

How to handle superuser access requests

When you configure export policies, you need to consider what you want to happen if the storage system receives a client access request with user ID 0, meaning as a superuser, and set up your export rules accordingly.

In the UNIX world, a user with the user ID 0 is known as the superuser, typically called root, who has unlimited access rights on a system. Using superuser privileges can be dangerous for several reasons, including breach of system and data security.

By default, ONTAP maps clients presenting with user ID 0 to the anonymous user. However, you can specify the -superuser parameter in export rules to determine how to handle clients presenting with user ID 0 depending on their security type. The following are valid options for the -superuser parameter:

- any
- none
- This is the default setting if you do not specify the -superuser parameter.
- krb5
- ntlm
- sys

There are two different ways how clients presenting with user ID 0 are handled, depending on the -superuser parameter configuration:

<table>
<thead>
<tr>
<th>If the -superuser parameter and the client's security type...</th>
<th>Then the client...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Match</td>
<td>Gets superuser access with user ID 0.</td>
</tr>
<tr>
<td>Do not match</td>
<td>Gets access as the anonymous user with the user ID specified by the -anon parameter and its assigned permissions. This is regardless of whether the read-only or read-write parameter specifies the option none.</td>
</tr>
</tbody>
</table>

If a client presents with user ID 0 to access a volume with NTFS security style and the -superuser parameter is set to none, ONTAP uses the name mapping for the anonymous user to obtain the proper credentials.

**Example**

The export policy contains an export rule with the following parameters:

- `-protocol nfs3`
- `-clientmatch 10.16.0/255.255.255.0`
- `-rorule any`
- `-rwrule krb5,ntlm`
- `-anon 127`
Client #1 has the IP address 10.1.16.207, has user ID 746, sends an access request using the NFSv3 protocol, and authenticated with Kerberos v5.

Client #2 has the IP address 10.1.16.211, has user ID 0, sends an access request using the NFSv3 protocol, and authenticated with AUTH_SYS.

The client access protocol and IP address matches for both clients. The read-only parameter allows read-only access to all clients regardless of the security type they authenticated with. However, only client #1 gets read-write access because it used the approved security type Kerberos v5 to authenticate.

Client #2 does not get superuser access. Instead, it gets mapped to anonymous because the -superuser parameter is not specified. This means it defaults to none and automatically maps user ID 0 to anonymous. Client #2 also only gets read-only access because its security type did not match the read-write parameter.

Example

The export policy contains an export rule with the following parameters:

- `-protocol nfs3`
- `-clientmatch 10.1.16.0/255.255.255.0`
- `-rorule any`
- `-rwrule krb5,ntlm`
- `-superuser krb5`
- `-anon 0`

Client #1 has the IP address 10.1.16.207, has user ID 0, sends an access request using the NFSv3 protocol, and authenticated with Kerberos v5.

Client #2 has the IP address 10.1.16.211, has user ID 0, sends an access request using the NFSv3 protocol, and authenticated with AUTH_SYS.

The client access protocol and IP address matches for both clients. The read-only parameter allows read-only access to all clients regardless of the security type they authenticated with. However, only client #1 gets read-write access because it used the approved security type Kerberos v5 to authenticate. Client #2 does not get read-write access.

The export rule allows superuser access for clients with user ID 0. Client #1 gets superuser access because it matches the user ID and security type for the read-only and -superuser parameters. Client #2 does not get read-write or superuser access because its security type does not match the read-write parameter or the -superuser parameter. Instead, client #2 is mapped to the anonymous user, which in this case has the user ID 0.

How ONTAP uses export policy caches

To improve system performance, ONTAP uses local caches to store information such as host names and netgroups. This enables ONTAP to process export policy rules more quickly than retrieving the information from external sources. Understanding what the caches are and what they do can help you troubleshoot client access issues.

You configure export policies to control client access to NFS exports. Each export policy contains rules, and each rule contains parameters to match the rule to clients requesting access. Some of these parameters require ONTAP to contact an external source, such as DNS or NIS servers, to resolve objects such as domain names, host names, or netgroups.
These communications with external sources take a small amount of time. To increase performance, ONTAP reduces the amount of time it takes to resolve export policy rule objects by storing information locally on each node in several caches.

<table>
<thead>
<tr>
<th>Cache name</th>
<th>Type of information stored</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access</td>
<td>Mappings of clients to corresponding export policies</td>
</tr>
<tr>
<td>Name</td>
<td>Mappings of UNIX user names to corresponding UNIX user IDs</td>
</tr>
<tr>
<td>ID</td>
<td>Mappings of UNIX user IDs to corresponding UNIX user IDs and extended UNIX group IDs</td>
</tr>
<tr>
<td>Host</td>
<td>Mappings of host names to corresponding IP addresses</td>
</tr>
<tr>
<td>Netgroup</td>
<td>Mappings of netgroups to corresponding IP addresses of members</td>
</tr>
<tr>
<td>Showmount</td>
<td>List of exported directories from SVM namespace</td>
</tr>
</tbody>
</table>

If you change information on the external name servers in your environment after ONTAP retrieved and stored it locally, the caches might now contain outdated information. Although ONTAP refreshes caches automatically after certain time periods, different caches have different expiration and refresh times and algorithms.

Another possible reason for caches to contain outdated information is when ONTAP attempts to refresh cached information but encounters a failure when attempting to communicate with name servers. If this happens, ONTAP continues to use the information currently stored in the local caches to prevent client disruption.

As a result, client access requests that are supposed to succeed might fail, and client access requests that are supposed to fail might succeed. You can view and manually flush some of the export policy caches when troubleshooting such client access issues.

**How the access cache works**

ONTAP uses an access cache to store the results of export policy rule evaluation for client access operations to a volume or qtree. This results in performance improvements because the information can be retrieved much faster from the access cache than going through the export policy rule evaluation process every time a client sends an I/O request.

Whenever an NFS client sends an I/O request to access data on a volume or qtree, ONTAP must evaluate each I/O request to determine whether to grant or deny the I/O request. This evaluation involves checking every export policy rule of the export policy associated with the volume or qtree. If the path to the volume or qtree involves crossing one or more junction points, this might require performing this check for multiple export policies along the path.

Note that this evaluation occurs for every I/O request sent from an NFS client, such as read, write, list, copy and other operations; not just for initial mount requests.

After ONTAP has identified the applicable export policy rules and decided whether to allow or deny the request, ONTAP then creates an entry in the access cache to store this information.

When an NFS client sends an I/O request, ONTAP notes the IP address of the client, the ID of the SVM, and the export policy associated with the target volume or qtree, and first checks the access cache for a matching entry. If a matching entry exists in the access cache, ONTAP uses the stored information to allow or deny the I/O request. If a matching entry does not exist, ONTAP then goes through the normal process of evaluating all applicable policy rules as explained above.

Beginning with ONTAP 9, access cache entries that are not actively used are not refreshed. This reduces unnecessary and wasteful communication with external name servers.

Retrieving the information from the access cache is much faster than going through the entire export policy rule evaluation process for every I/O request. Therefore, using the access cache greatly improves performance by reducing the overhead of client access checks.
How access cache parameters work

Several parameters control the refresh periods for entries in the access cache. Understanding how these parameters work enables you to modify them to tune the access cache and balance performance with how recent the stored information is.

The access cache stores entries consisting of one or more export rules that apply to clients attempting to access volumes or qtrees. These entries are stored for a certain amount of time before they are refreshed. The refresh time is determined by access cache parameters and depends on the type of access cache entry.

Beginning with ONTAP 9, you can specify access cache parameters for individual SVMs. This allows the parameters to differ according to SVM access requirements. Access cache entries that are not actively used are not refreshed, which reduces unnecessary and wasteful communication with external name servers.

<table>
<thead>
<tr>
<th>Access cache entry type</th>
<th>Description</th>
<th>Refresh period in seconds</th>
</tr>
</thead>
</table>
| Positive entries        | Access cache entries that have not resulted in access denial to clients. | Minimum: 300  
                           |                          | Maximum: 86,400  
                           |                          | Default: 3,600 |
| Negative entries        | Access cache entries that have resulted in access denial to clients. | Minimum: 60  
                           |                          | Maximum: 86,400  
                           |                          | Default: 3,600 |

**Example**

An NFS client attempts to access a volume on a cluster. ONTAP matches the client to an export policy rule and determines that the client gets access based on the export policy rule configuration. ONTAP stores the export policy rule in the access cache as a positive entry. By default, ONTAP keeps the positive entry in the access cache for one hour (3,600 seconds), and then automatically refreshes the entry to keep the information current.

To prevent the access cache from filling up unnecessarily, there is an additional parameter to clear existing access cache entries that have not been used for a certain time period to decide client access. This `-harvest-timeout` parameter has an allowed range of 60 through 2,592,000 seconds and a default setting of 86,400 seconds.

Removing an export policy from a qtree

If you decide you do not want a specific export policy assigned to a qtree any longer, you can remove the export policy by modifying the qtree to inherit the export policy of the containing volume instead. You can do this by using the `volume qtree modify` command with the `-export-policy` parameter and an empty name string ("").

**Steps**

1. To remove an export policy from a qtree, enter the following command:

   ```
   volume qtree modify -vserver vserver_name -qtree-path /vol/volume_name/qttree_name -export-policy ""
   ```

2. Verify that the qtree was modified accordingly by entering the following command:

   ```
   volume qtree show -qtree qtree_name -fields export-policy
   ```
Validating qtree IDs for qtree file operations

ONTAP can perform an optional additional validation of qtree IDs. This validation ensures that client file operation requests use a valid qtree ID and that clients can only move files within the same qtree. You can enable or disable this validation by modifying the -validate-qtree-export parameter. This parameter is enabled by default.

About this task

This parameter is only effective when you have assigned an export policy directly to one or more qtrees on the storage virtual machine (SVM).

Steps

1. Set the privilege level to advanced:
   ```bash
   set -privilege advanced
   ``

2. Perform one of the following actions:

<table>
<thead>
<tr>
<th>If you want qtree ID validation to be...</th>
<th>Enter the following command...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled</td>
<td>vserver nfs modify -vserver vserver_name -validate-qtree-export enabled</td>
</tr>
<tr>
<td>Disabled</td>
<td>vserver nfs modify -vserver vserver_name -validate-qtree-export disabled</td>
</tr>
</tbody>
</table>

3. Return to the admin privilege level:
   ```bash
   set -privilege admin
   ``

Export policy restrictions and nested junctions for FlexVol volumes

If you configured export policies to set a less restrictive policy on a nested junction but a more restrictive policy on a higher level junction, access to the lower level junction might fail.

You should ensure that higher level junctions have less restrictive export policies than lower level junctions.

Using Kerberos with NFS for strong security

You can use Kerberos to provide strong authentication between SVMs and NFS clients to provide secure NFS communication. Configuring NFS with Kerberos increases the integrity and security of NFS client communications with the storage system.

ONTAP support for Kerberos

Kerberos provides strong secure authentication for client/server applications. Authentication provides verification of user and process identities to a server. In the ONTAP environment, Kerberos provides authentication between storage virtual machines (SVMs) and NFS clients.

In ONTAP 9, the following Kerberos functionality is supported:

- Kerberos 5 authentication with integrity checking (krb5i)
  Krb5i uses checksums to verify the integrity of each NFS message transferred between client and server. This is useful both for security reasons (for example, to ensure that data has not been tampered with) and for data integrity reasons (for example, to prevent data corruption when using NFS over unreliable networks).
• Kerberos 5 authentication with privacy checking (krb5p)
  Krb5p uses checksums to encrypt all the traffic between client and the server. This is more secure
and also incurs more load.

• 128-bit and 256-bit AES encryption
  Advanced Encryption Standard (AES) is an encryption algorithm for securing electronic data.
ONTAP now supports AES with 128-bit keys (AES-128) and AES with 256-bit keys (AES-256)
encryption for Kerberos for stronger security.

• SVM-level Kerberos realm configurations
  SVM administrators can now create Kerberos realm configurations at the SVM level. This means
that SVM administrators no longer have to rely on the cluster administrator for Kerberos realm
configuration and can create individual Kerberos realm configurations in a multi-tenancy
environment.

Requirements for configuring Kerberos with NFS

Before you configure Kerberos with NFS on your system, you must verify that certain items in your
network and storage environment are properly configured.

  Note: The steps to configure your environment depend on what version and type of client
operating system, domain controller, Kerberos, DNS, etc., that you are using. Documenting all
these variables is beyond the scope of this document. For more information, see the respective
documentation for each component.

For a detailed example of how to set up ONTAP and Kerberos 5 with NFSv3 and NFSv4 in an
environment using Windows Server 2008 R2 Active Directory and Linux hosts, see technical
report 4073.

The following items should be configured first:

Network environment requirements

• Kerberos
  You must have a working Kerberos setup with a key distribution center (KDC), such as Windows
Active Directory based Kerberos or MIT Kerberos.
NFS servers must use “nfs” as the primary component of their machine principal.

• Directory service
  You must use a secure directory service in your environment, such as Active Directory or
OpenLDAP, that is configured to use LDAP over SSL/TLS.

• NTP
  You must have a working time server running NTP. This is necessary to prevent Kerberos
authentication failure due to time skew.

• Domain name resolution (DNS)
  Each UNIX client and each SVM LIF must have a proper service record (SRV) registered with
the KDC under forward and reverse lookup zones. All participants must be properly resolvable
via DNS.

• User accounts
  Each client must have a user account in the Kerberos realm. NFS servers must use “nfs” as the
primary component of their machine principal.

NFS client requirements

• NFS
Each client must be properly configured to communicate over the network using NFSv3 or NFSv4. Clients must support RFC1964 and RFC2203.

- **Kerberos**
  Each client must be properly configured to use Kerberos authentication, including the following details:
  - Encryption for TGS communication is enabled. AES-256 for strongest security.
  - The most secure encryption type for TGT communication is enabled.
  - The Kerberos realm and domain are configured correctly.
  - GSS is enabled.

When using machine credentials:
  - Do not run `gssd` with the `-n` parameter.
  - Do not run `kinit` as the root user.

- Each client must use the most recent and updated operating system version. This provides the best compatibility and reliability for AES encryption with Kerberos.

- **DNS**
  Each client must be properly configured to use DNS for correct name resolution.

- **NTP**
  Each client must be synchronizing with the NTP server.

- **Host and domain information**
  Each client’s `/etc/hosts` and `/etc/resolv.conf` files must contain the correct host name and DNS information, respectively.

- **Keytab files**
  Each client must have a keytab file from the KDC. The realm must be in uppercase letters. The encryption type must be AES-256 for strongest security.

- Optional: For best performance, clients benefit from having at least two network interfaces: one for communicating with the local area network and one for communicating with the storage network.

**Storage system requirements**

- **NFS license**
  The storage system must have a valid NFS license installed.

- **CIFS license**
  The CIFS license is optional. It is only required for checking Windows credentials when using multiprotocol name mapping. It is not required in a strict UNIX-only environment.

- **SVM**
  You must have at least one SVM configured on the system.

- **DNS on the SVM**
  You must have configured DNS on each SVM.

- **NFS server**
  You must have configured NFS on the SVM.
• AES encryption
  For strongest security, you must configure the NFS server to allow only AES-256 encryption for Kerberos.

• CIFS server
  If you are running a multiprotocol environment, you must have configured CIFS on the SVM. The CIFS server is required for multiprotocol name mapping.

• Volumes
  You must have a root volume and at least one data volume configured for use by the SVM.

• Root volume
  The root volume of the SVM must have the following configuration:

<table>
<thead>
<tr>
<th>Name</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security style</td>
<td>UNIX</td>
</tr>
<tr>
<td>UID</td>
<td>root or ID 0</td>
</tr>
<tr>
<td>GID</td>
<td>root or ID 0</td>
</tr>
<tr>
<td>UNIX permissions</td>
<td>777</td>
</tr>
</tbody>
</table>

  In contrast to the root volume, data volumes can have either security style.

• UNIX groups
  The SVM must have the following UNIX groups configured:

<table>
<thead>
<tr>
<th>Group name</th>
<th>Group ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>daemon</td>
<td>1</td>
</tr>
<tr>
<td>root</td>
<td>0</td>
</tr>
<tr>
<td>pcuser</td>
<td>65534 (created automatically by ONTAP when you create the SVM)</td>
</tr>
</tbody>
</table>

• UNIX users
  The SVM must have the following UNIX users configured:

<table>
<thead>
<tr>
<th>User name</th>
<th>User ID</th>
<th>Primary group ID</th>
<th>Comment</th>
</tr>
</thead>
</table>
  | nfs       | 500     | 0                | Required for GSS INIT phase
             |          |                  | The first component of the NFS client user SPN is used as the user. |
  | pcuser    | 65534   | 65534            | Required for NFS and CIFS multiprotocol use
             |          |                  | Created and added to the pcuser group automatically by ONTAP when you create the SVM. |
  | root      | 0       | 0                | Required for mounting |

  The nfs user is not required if a Kerberos-UNIX name mapping exists for the SPN of the NFS client user.

• Export policies and rules
  You must have configured export policies with the necessary export rules for the root and data volumes and qtrees. If all volumes of the SVM are accessed over Kerberos, you can set the export rule options `--rorule`, `--rwrule`, and `--superuser` for the root volume to `krb5`, `krb5i`, or `krb5p`. 
Kerberos-UNIX name mapping

If you want the user identified by the NFS client user SPN to have root permissions, you must create a name mapping to root.

Related information

- NetApp Interoperability Matrix Tool
- System administration
- Logical storage management

Specifying the user ID domain for NFSv4

To specify the user ID domain, you can set the \(-v4\)-id-domain option.

About this task

By default, ONTAP uses the NIS domain for NFSv4 user ID mapping, if one is set. If an NIS domain is not set, the DNS domain is used. You might need to set the user ID domain if, for example, you have multiple user ID domains. The domain name must match the domain configuration on the domain controller. It is not required for NFSv3.

Step

1. Enter the following command:

   \texttt{vserver nfs modify -vserver vserver\_name -v4-id-domain NIS\_domain\_name}

Configuring name services

Depending on the configuration of your storage system, ONTAP needs to be able to look up host, user, group, or netgroup information to provide proper access to clients. You must configure name services to enable ONTAP to access local or external name services to obtain this information.

How ONTAP name service switch configuration works

ONTAP stores name service configuration information in a table that is the equivalent of the \texttt{/etc/nsswitch.conf} file on UNIX systems. You must understand the function of the table and how ONTAP uses it so that you can configure it appropriately for your environment.

The ONTAP name service switch table determines which name service sources ONTAP consults in which order to retrieve information for a certain type of name service information. ONTAP maintains a separate name service switch table for each SVM.

Database types

The table stores a separate name service list for each of the following database types:

<table>
<thead>
<tr>
<th>Database type</th>
<th>Defines name service sources for...</th>
<th>Valid sources are...</th>
</tr>
</thead>
<tbody>
<tr>
<td>hosts</td>
<td>Converting host names to IP addresses</td>
<td>files, dns</td>
</tr>
<tr>
<td>group</td>
<td>Looking up user group information</td>
<td>files, nis, ldap</td>
</tr>
<tr>
<td>passwd</td>
<td>Looking up user information</td>
<td>files, nis, ldap</td>
</tr>
<tr>
<td>netgroup</td>
<td>Looking up netgroup information</td>
<td>files, nis, ldap</td>
</tr>
<tr>
<td>namemap</td>
<td>Mapping user names</td>
<td>files, ldap</td>
</tr>
</tbody>
</table>
Source types

The sources specify which name service source to use for retrieving the appropriate information.

<table>
<thead>
<tr>
<th>Specify source type...</th>
<th>To look up information in...</th>
<th>Managed by the command families...</th>
</tr>
</thead>
<tbody>
<tr>
<td>files</td>
<td>Local source files</td>
<td>vserver services name-service unix-user</td>
</tr>
<tr>
<td></td>
<td></td>
<td>vserver services name-service unix-group</td>
</tr>
<tr>
<td></td>
<td></td>
<td>vserver services name-service netgroup</td>
</tr>
<tr>
<td></td>
<td></td>
<td>vserver services name-service dns hosts</td>
</tr>
<tr>
<td>nis</td>
<td>External NIS servers as specified in the NIS domain configuration of the SVM</td>
<td>vserver services name-service nis-domain</td>
</tr>
<tr>
<td>ldap</td>
<td>External LDAP servers as specified in the LDAP client configuration of the SVM</td>
<td>vserver services name-service ldap</td>
</tr>
<tr>
<td>dns</td>
<td>External DNS servers as specified in the DNS configuration of the SVM</td>
<td>vserver services name-service dns</td>
</tr>
</tbody>
</table>

Even if you plan to use NIS or LDAP for both data access and SVM administration authentication, you should still include files and configure local users as a fallback in case NIS or LDAP authentication fails.

Protocols used to access external sources

To access the servers for external sources, ONTAP uses the following protocols:

<table>
<thead>
<tr>
<th>External name service source</th>
<th>Protocol used for access</th>
</tr>
</thead>
<tbody>
<tr>
<td>NIS</td>
<td>UDP</td>
</tr>
<tr>
<td>DNS</td>
<td>UDP</td>
</tr>
<tr>
<td>LDAP</td>
<td>TCP</td>
</tr>
</tbody>
</table>

Example

The following example displays the name service switch configuration for the SVM svm_1:

```
cluster1::*> vserver services name-service ns-switch show -vserver svm_1

<table>
<thead>
<tr>
<th>Vserver</th>
<th>Database</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>svm_1</td>
<td>hosts</td>
<td>files, dns</td>
</tr>
<tr>
<td>svm_1</td>
<td>group</td>
<td>files</td>
</tr>
<tr>
<td>svm_1</td>
<td>passwd</td>
<td>files</td>
</tr>
<tr>
<td>svm_1</td>
<td>netgroup</td>
<td>nis, files</td>
</tr>
</tbody>
</table>
```

To look up IP addresses for hosts, ONTAP first consults local source files. If the query does not return any results, DNS servers are checked next.

To look up user or group information, ONTAP consults only local sources files. If the query does not return any results, the lookup fails.
To look up netgroup information, ONTAP first consults external NIS servers. If the query does not return any results, the local netgroup file is checked next.

There are no name service entries for name mapping in the table for the SVM svm_1. Therefore, ONTAP consults only local source files by default.

**Related information**


**Using LDAP**

An LDAP server enables you to centrally maintain user information. If you store your user database on an LDAP server in your environment, you can configure your storage system to look up user information in your existing LDAP database.

- Before configuring LDAP for ONTAP, you should verify that your site deployment meets best practices for LDAP server and client configuration. In particular, the following conditions must be met:
  - The domain name of the LDAP server must match the entry on the LDAP client.
  - If the LDAP server requires session security measures, you must configure them in the LDAP client.
    - The following session security options are available:
      - LDAP signing (provides data integrity checking) and LDAP signing and sealing (provides data integrity checking and encryption)
      - LDAP over TLS (encryption)
  - To enable signed and sealed LDAP queries, the following services must be configured:
    - LDAP servers must support the GSSAPI (Kerberos) SASL mechanism.
    - LDAP servers must have DNS A/AAAA records as well as PTR records set up on the DNS server.
    - Kerberos servers must have SRV records present on the DNS server.
  - To enable TLS encrypted LDAP queries, the following services must be configured:
    - The LDAP server must be enabled for TLS.
      - As of ONTAP 9.0, SSL is no longer supported.
    - A certificate server must already be configured in the domain.
  - To enable LDAP referral chasing (in ONTAP 9.5 and later), the following conditions must be satisfied:
    - Both domains should be configured with one of the following trust relationships:
      - Two-way
      - One-way, where the primary trusts the referral domain
      - Parent-child
    - DNS must be configured to resolve all referred server names.
    - Domain passwords should be same to authenticate when --bind-as-cifs-server set to true.

**Note:** The following configurations are not currently supported with LDAP referral chasing:
- LDAP signing and sealing (the `-session-security` option)
- Encrypted TLS connections (the `-use-start-tls` option)
- Communications over LDAPS port 636
- LDAP clients on an admin SVM

- You must enter an LDAP schema when configuring the LDAP client on the SVM. In most cases, one of the default ONTAP schemas will be appropriate. However, if the LDAP schema in your environment differs from these, you must create a new LDAP client schema for ONTAP before creating the LDAP client. Consult with your LDAP administrator about requirements for your environment.

- Using LDAP for host name resolution is not supported.

Securing LDAP session communication

There are two ways to enable LDAP secure sessions for queries to the AD server. You can use LDAP signing and sealing, which encrypts traffic (sealing) and protects the integrity (signing) of session traffic between the storage virtual machine (SVM) LDAP client and the LDAP server. Alternatively, you can use LDAP over TLS to encrypt all session traffic.

LDAP signing and sealing concepts

Beginning in ONTAP 9, you can configure signing and sealing to enable LDAP session security on queries to an Active Directory (AD) server. You must configure the NFS server security settings on the storage virtual machine (SVM) to correspond to those on the LDAP server.

Signing confirms the integrity of the LDAP payload data using secret key technology. Sealing encrypts the LDAP payload data to avoid transmitting sensitive information in clear text. An LDAP Security Level option indicates whether the LDAP traffic needs to be signed, signed and sealed, or neither. The default is `none`.

LDAP signing and sealing on CIFS traffic is enabled on the SVM with the `-session-security-for-ad-ldap` option to the `vserver cifs security modify` command.

LDAP over TLS concepts

You must understand certain terms and concepts about how ONTAP uses TLS to secure LDAP communication. ONTAP can use LDAP over TLS for setting up authenticated sessions between Active Directory-integrated LDAP servers or UNIX-based LDAP servers.

Terminology

There are certain terms that you should understand about how ONTAP uses LDAP over SSL to secure LDAP communication.

LDAP

(Lightweight Directory Access Protocol) A protocol for accessing and managing information directories. LDAP is used as an information directory for storing objects such as users, groups, and netgroups. LDAP also provides directory services that manage these objects and fulfill LDAP requests from LDAP clients.

SSL

(Secure Sockets Layer) A protocol developed for sending information securely over the Internet. Beginning with ONTAP 9, SSL is no longer supported.
TLS
(Transport Layer Security) An IETF standards track protocol that is based on the earlier SSL specifications. It is the successor to SSL.

LDAP over TLS
(Also known as LDAPS) A protocol that uses TLS to secure communication between LDAP clients and LDAP servers. The terms LDAP over SSL and LDAP over TLS are sometimes used interchangeably, but only TLS is supported by ONTAP 9 and later.

Start TLS
(Also known as start_tls, STARTTLS, and StartTLS) A mechanism to provide secure communication by using the TLS protocols.

ONTAP uses STARTTLS for securing LDAP communication, and uses the default LDAP port (389) to communicate with the LDAP server. The LDAP server must be configured to allow connections over LDAP port 389; otherwise, LDAP TLS connections from the storage virtual machine (SVM) to the LDAP server fail.

How ONTAP uses LDAP over TLS
By default, LDAP communications between client and server applications are not encrypted. This means that it is possible to use a network monitoring device or software and view the communications between LDAP client and server computers. This is especially problematic when an LDAP simple bind is used because the credentials (user name and password) used to bind the LDAP client to the LDAP server are passed over the network unencrypted.

The TLS protocol runs above TCP/IP and below higher-level protocols, such as LDAP. They use TCP/IP on behalf of the higher-level protocols, and in the process, permit a TLS-enabled server to authenticate itself to a TLS-enabled client and permit both machines to establish an encrypted connection. These capabilities address fundamental security concerns about communication over the Internet and other TCP/IP networks.

ONTAP supports TLS server authentication, which enables the SVM LDAP client to confirm the LDAP server's identity during the bind operation. TLS-enabled LDAP clients can use standard techniques of public-key cryptography to check that a server's certificate and public ID are valid and have been issued by a certificate authority (CA) listed in the client's list of trusted CAs.

LDAP supports STARTTLS to encrypt communications using TLS. STARTTLS begins as a plaintext connection over the standard LDAP port (389), and that connection is then upgraded to TLS.

ONTAP supports the following:
• LDAP over TLS for SMB-related traffic between the Active Directory-integrated LDAP servers and the SVM
• LDAP over TLS for LDAP traffic for name mapping and other UNIX information
  Either Active Directory-integrated LDAP servers or UNIX-based LDAP servers can be used to store information for LDAP name mapping and other UNIX information, such as users, groups, and netgroups.
• Self-signed root CA certificates
  When using an Active-Directory integrated LDAP, the self-signed root certificate is generated when the Windows Server Certificate Service is installed in the domain. When using an UNIX-based LDAP server for LDAP name mapping, the self-signed root certificate is generated and saved by using means appropriate to that LDAP application.

By default, LDAP over TLS is disabled.
Enabling LDAP RFC2307bis support

If you want to use LDAP and require the additional capability to use nested group memberships, you can configure ONTAP to enable LDAP RFC2307bis support.

Before you begin

You must have created a copy of one of the default LDAP client schemas that you want to use.

About this task

In LDAP client schemas, group objects use the memberUid attribute. This attribute can contain multiple values and lists the names of the users that belong to that group. In RFC2307bis enabled LDAP client schemas, group objects use the uniqueMember attribute. This attribute can contain the full distinguished name (DN) of another object in the LDAP directory. This enables you to use nested groups because groups can have other groups as members.

The user should not be a member of more than 256 groups including nested groups. ONTAP ignores any groups over the 256 group limit.

By default, RFC2307bis support is disabled.

Steps

1. Set the privilege level to advanced:
   ```
   set -privilege advanced
   ```

2. Modify the copied RFC2307 LDAP client schema to enable RFC2307bis support:
   ```
   vserver services name-service ldap client schema modify -vserver vserver_name -schema schema-name -enable-rfc2307bis true
   ```

3. Modify the schema to match the object class supported in the LDAP server:
   ```
   vserver services name-service ldap client schema modify -vserver vserver-name -schema schema_name -group-of-unique-names-object-class object_class
   ```

4. Modify the schema to match the attribute name supported in the LDAP server:
   ```
   vserver services name-service ldap client schema modify -vserver vserver-name -schema schema_name -unique-member-attribute attribute_name
   ```

5. Return to the admin privilege level:
   ```
   set -privilege admin
   ```

Related references

- Commands for managing LDAP client configurations on page 68

Configuration options for LDAP directory searches

You can optimize LDAP directory searches, including user, group, and netgroup information, by configuring the ONTAP LDAP client to connect to LDAP servers in the most appropriate way for your environment. You need to understand when the default LDAP base and scope search values suffice and which parameters to specify when custom values are more appropriate.

LDAP client search options for user, group, and netgroup information can help avoid failed LDAP queries, and therefore failed client access to storage systems. They also help ensure that the searches are as efficient as possible to avoid client performance issues.
**Default base and scope search values**

The LDAP base is the default base DN that the LDAP client uses to perform LDAP queries. All searches, including user, group, and netgroup searches, are done using the base DN. This option is appropriate when your LDAP directory is relatively small and all relevant entries are located in the same DN.

If you do not specify a custom base DN, the default is `root`. This means that each query searches the entire directory. Although this maximizes the chances of success of the LDAP query, it can be inefficient and result in significantly decreased performance with large LDAP directories.

The LDAP base scope is the default search scope that the LDAP client uses to perform LDAP queries. All searches, including user, group, and netgroup searches, are done using the base scope. It determines whether the LDAP query searches only the named entry, entries one level below the DN, or the entire subtree below the DN.

If you do not specify a custom base scope, the default is `subtree`. This means that each query searches the entire subtree below the DN. Although this maximizes the chances of success of the LDAP query, it can be inefficient and result in significantly decreased performance with large LDAP directories.

**Custom base and scope search values**

Optionally, you can specify separate base and scope values for user, group, and netgroup searches. Limiting the search base and scope of queries this way can significantly improve performance because it limits the search to a smaller subsection of the LDAP directory.

If you specify custom base and scope values, they override the general default search base and scope for user, group, and netgroup searches. The parameters to specify custom base and scope values are available at the advanced privilege level.

<table>
<thead>
<tr>
<th>LDAP client parameter...</th>
<th>Specifies custom...</th>
</tr>
</thead>
<tbody>
<tr>
<td>-base-dn</td>
<td>Base DN for all LDAP searches</td>
</tr>
<tr>
<td></td>
<td>Multiple values can be entered if needed (for example, if LDAP referral chasing is enabled in ONTAP 9.5 and later releases).</td>
</tr>
<tr>
<td>-base-scope</td>
<td>Base scope for all LDAP searches</td>
</tr>
<tr>
<td>-user-dn</td>
<td>Base DNs for all LDAP user searches</td>
</tr>
<tr>
<td></td>
<td>This parameter also applies to user name-mapping searches.</td>
</tr>
<tr>
<td>-user-scope</td>
<td>Base scope for all LDAP user searches</td>
</tr>
<tr>
<td></td>
<td>This parameter also applies to user name-mapping searches.</td>
</tr>
<tr>
<td>-group-dn</td>
<td>Base DNs for all LDAP group searches</td>
</tr>
<tr>
<td>-group-scope</td>
<td>Base scope for all LDAP group searches</td>
</tr>
<tr>
<td>-netgroup-dn</td>
<td>Base DNs for all LDAP netgroup searches</td>
</tr>
<tr>
<td>-netgroup-scope</td>
<td>Base scope for all LDAP netgroup searches</td>
</tr>
</tbody>
</table>

**Multiple custom base DN values**

If your LDAP directory structure is more complex, it might be necessary for you to specify multiple base DNs to search multiple parts of your LDAP directory for certain information. You can specify multiple DNs for the user, group, and netgroup DN parameters by separating them with a semicolon (`;`) and enclosing the entire DN search list with double quotes (`"`). If a DN contains a semicolon, you must add an escape character (`\`) immediately before the semicolon in the DN.
Note that the scope applies to the entire list of DNs specified for the corresponding parameter. For example, if you specify a list of three different user DNs and subtree for the user scope, then LDAP user searches search the entire subtree for each of the three specified DNs.

Beginning with ONTAP 9.5, you can also specify LDAP referral chasing, which allows the ONTAP LDAP client to refer look-up requests to other LDAP servers if an LDAP referral response is not returned by the primary LDAP server. The client uses that referral data to retrieve the target object from the server described in the referral data. To search for objects present in the referred LDAP servers, the base-dn of the referred objects can be added to the base-dn as part of LDAP client configuration. However, referred objects are only looked up when referral chasing is enabled (using the `--referral-enabled true` option) during LDAP client creation or modification.

**Related references**

*Commands for managing LDAP client configurations* on page 68

### Improving performance of LDAP directory netgroup-by-host searches

If your LDAP environment is configured to allow netgroup-by-host searches, you can configure ONTAP to take advantage of this and perform netgroup-by-host searches. This can significantly speed up netgroup searches and reduce possible NFS client access issues due to latency during netgroup searches.

**Before you begin**

Your LDAP directory must contain a `netgroup.byhost` map.

Your DNS servers should contain both forward (A) and reverse (PTR) lookup records for NFS clients.

When you specify IPv6 addresses in netgroups, you must always shorten and compress each address as specified in RFC 5952.

**About this task**

NIS servers store netgroup information in three separate maps called `netgroup`, `netgroup.byuser`, and `netgroup.byhost`. The purpose of the `netgroup.byuser` and `netgroup.byhost` maps is to speed up netgroup searches. ONTAP can perform netgroup-by-host searches on NIS servers for improved mount response times.

By default, LDAP directories do not have such a `netgroup.byhost` map like NIS servers. It is possible, though, with the help of third-party tools, to import a NIS `netgroup.byhost` map into LDAP directories to enable fast netgroup-by-host searches. If you have configured your LDAP environment to allow netgroup-by-host searches, you can configure the ONTAP LDAP client with the `netgroup.byhost` map name, DN, and search scope for faster netgroup-by-host searches.

Receiving the results for netgroup-by-host searches faster enables ONTAP to process export rules faster when NFS clients request access to exports. This reduces the chance of delayed access due to netgroup search latency issues.

**Steps**

1. Obtain the exact full distinguished name of the NIS `netgroup.byhost` map you imported into your LDAP directory.

   The map DN can vary depending on the third-party tool you used for import. For best performance, you should specify the exact map DN.

2. Set the privilege level to advanced:

   ```bash
   set --privilege advanced
   ```
3. Enable netgroup-by-host searches in the LDAP client configuration of the storage virtual machine (SVM):

```bash
vserver services name-service ldap client modify -vserver vserver_name -client-config config_name -is-netgroup-byhost-enabled true -netgroup-byhost-dn netgroup-by-host_map_distinguished_name -netgroup-byhost-scope netgroup-by-host_search_scope

-is-netgroup-byhost-enabled {true|false} enables or disables netgroup-by-host search for LDAP directories. The default is false.

-netgroup-byhost-dn netgroup-by-host_map_distinguished_name specifies the distinguished name of the netgroup.byhost map in the LDAP directory. It overrides the base DN for netgroup-by-host searches. If you do not specify this parameter, ONTAP uses the base DN instead.

-netgroup-byhost-scope {base|onelevel|subtree} specifies the search scope for netgroup-by-host searches. If you do not specify this parameter, the default is subtree.

If the LDAP client configuration does not exist yet, you can enable netgroup-by-host searches by specifying these parameters when creating a new LDAP client configuration using the `vserver services name-service ldap client create` command.

**Note:** Starting in ONTAP 9.2, the field `-ldap-servers` replaces the field `-servers`. This new field can take either a hostname or an IP address for the LDAP server.

4. Return to the admin privilege level:

```bash
set -privilege admin
```

**Example**

The following command modifies the existing LDAP client configuration named “ldap_corp” to enable netgroup-by-host searches using the netgroup.byhost map named “nisMapName=netgroup.byhost,dc=corp,dc=example,dc=com” and the default search scope subtree:

```bash
cluster1::*> vserver services name-service ldap client modify -vserver vs1 -client-config ldap_corp -is-netgroup-byhost-enabled true -netgroup-byhost-dn nisMapName=netgroup.byhost,dc=corp,dc=example,dc=com
```

**After you finish**

The netgroup.byhost and netgroup maps in the directory must be kept in sync at all times to avoid client access issues.

**Related information**

*IETF RFC 5952: A Recommendation for IPv6 Address Text Representation*

**Displaying LDAP statistics**

Beginning with ONTAP 9.2, you can display LDAP statistics for storage virtual machines (SVMs) on a storage system to monitor the performance and diagnose issues.

**Before you begin**

- You must have configured an LDAP client on the SVM.
- You must have identified LDAP objects from which you can view data.
Step

1. View the performance data for counter objects:

```
  statistics show
```

Examples

The following example shows the performance data for object `secd_external_service_op`:

```
cluster::*>> statistics show -vserver vserverName -object secd_external_service_op -instance "vserverName:LDAP (NIS & Name Mapping):GetUserInfoFromName:1.1.1.1"
```

Object: secd_external_service_op
Instance: vserverName:LDAP (NIS & Name Mapping):GetUserInfoFromName:1.1.1.1
Start-time: 4/13/2016 22:15:38
End-time: 4/13/2016 22:15:38
Scope: vserverName

<table>
<thead>
<tr>
<th>Counter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>instance_name</td>
<td>vserverName:LDAP (NIS &amp; Name Mapping):GetUserInfoFromName:1.1.1.1</td>
</tr>
<tr>
<td>last_modified_time</td>
<td>1460610787</td>
</tr>
<tr>
<td>node_name</td>
<td>nodeName</td>
</tr>
<tr>
<td>num_not_found_responses</td>
<td>1</td>
</tr>
<tr>
<td>num_request_failures</td>
<td>1</td>
</tr>
<tr>
<td>num_requests_sent</td>
<td>1</td>
</tr>
<tr>
<td>num_responses_received</td>
<td>1</td>
</tr>
<tr>
<td>num_successful_responses</td>
<td>0</td>
</tr>
<tr>
<td>num_timeouts</td>
<td>0</td>
</tr>
<tr>
<td>operation</td>
<td>GetUserInfoFromName</td>
</tr>
<tr>
<td>process_name</td>
<td>secd</td>
</tr>
<tr>
<td>request_latency</td>
<td>52131us</td>
</tr>
</tbody>
</table>

Configuring name mappings

ONTAP uses name mapping to map CIFS identities to UNIX identities, Kerberos identities to UNIX identities, and UNIX identities to CIFS identities. It needs this information to obtain user credentials and provide proper file access regardless of whether they are connecting from an NFS client or a CIFS client.

There are two exceptions where you do not have to use name mapping:

- You configure a pure UNIX environment and do not plan to use CIFS access or NTFS security style on volumes.
- You configure the default user to be used instead.

In this scenario, name mapping is not required because instead of mapping every individual client credential all client credentials are mapped to the same default user.

Note that you can use name mapping only for users, not for groups.

However, you can map a group of individual users to a specific user. For example, you can map all AD users that start or end with the word SALES to a specific UNIX user and to the user’s UID.

How name mapping works

When ONTAP has to map credentials for a user, it first checks the local name mapping database and LDAP server for an existing mapping. Whether it checks one or both and in which order is determined by the name service configuration of the SVM.

- For Windows to UNIX mapping
If no mapping is found, ONTAP checks whether the lowercase Windows user name is a valid user name in the UNIX domain. If this does not work, it uses the default UNIX user provided that it is configured. If the default UNIX user is not configured and ONTAP cannot obtain a mapping this way either, mapping fails and an error is returned.

- For UNIX to Windows mapping
  If no mapping is found, ONTAP tries to find a Windows account that matches the UNIX name in the SMB domain. If this does not work, it uses the default SMB user, provided that it is configured. If the default CIFS user is not configured and ONTAP cannot obtain a mapping this way either, mapping fails and an error is returned.

Machine accounts are mapped to the specified default UNIX user by default. If no default UNIX user is specified, machine account mappings fail.

- Beginning with ONTAP 9.5, you can map machine accounts to users other than the default UNIX user.
- In ONTAP 9.4 and earlier, you cannot map machine accounts to other users.
  Even if name mappings for machine accounts are defined, the mappings are ignored.

**Multidomain searches for UNIX user to Windows user name mappings**

ONTAP supports multidomain searches when mapping UNIX users to Windows users. All discovered trusted domains are searched for matches to the replacement pattern until a matching result is returned. Alternatively, you can configure a list of preferred trusted domains, which is used instead of the discovered trusted domain list and is searched in order until a matching result is returned.

**How domain trusts affect UNIX user to Windows user name mapping searches**

To understand how multidomain user name mapping works, you must understand how domain trusts work with ONTAP. Active Directory trust relationships with the CIFS server's home domain can be a bidirectional trust or can be one of two types of unidirectional trusts, either an inbound trust or an outbound trust. The home domain is the domain to which the CIFS server on the SVM belongs.

- **Bidirectional trust**
  With bidirectional trusts, both domains trust each other. If the CIFS server's home domain has a bidirectional trust with another domain, the home domain can authenticate and authorize a user belonging to the trusted domain and vice versa.
  UNIX user to Windows user name mapping searches can be performed only on domains with bidirectional trusts between the home domain and the other domain.

- **Outbound trust**
  With an outbound trust, the home domain trusts the other domain. In this case, the home domain can authenticate and authorize a user belonging to the outbound trusted domain.
  A domain with an outbound trust with the home domain is not searched when performing UNIX user to Windows user name mapping searches.

- **Inbound trust**
  With an inbound trust, the other domain trusts the CIFS server's home domain. In this case, the home domain cannot authenticate or authorize a user belonging to the inbound trusted domain.
  A domain with an inbound trust with the home domain is not searched when performing UNIX user to Windows user name mapping searches.

**How wildcards (*) are used to configure multidomain searches for name mapping**

Multidomain name mapping searches are facilitated by the use of wildcards in the domain section of the Windows user name. The following table illustrates how to use wildcards in the domain part of a name mapping entry to enable multidomain searches:
### How multidomain name searches are performed

You can choose one of two methods for determining the list of trusted domains used for multidomain name searches:

- Use the automatically discovered bidirectional trust list compiled by ONTAP
- Use the preferred trusted domain list that you compile

If a UNIX user is mapped to a Windows user with a wildcard used for the domain section of the user name, the Windows user is looked up in all the trusted domains as follows:

- If a preferred trusted-domain list is configured, the mapped Windows user is looked up in this search list only, in order.
- If a preferred list of trusted domains is not configured, then the Windows user is looked up in all the bidirectionally trusted domains of the home domain.
- If there are no bidirectionally trusted domains for the home domain, the user is looked up in the home domain.

If a UNIX user is mapped to a Windows user without a domain section in the user name, the Windows user is looked up in the home domain.

For more information about managing the list of bidirectional trusted domains, see the SMB/CIFS Reference.

### Name mapping conversion rules

An ONTAP system keeps a set of conversion rules for each SVM. Each rule consists of two pieces: a pattern and a replacement. Conversions start at the beginning of the appropriate list and perform a substitution based on the first matching rule. The pattern is a UNIX-style regular expression. The replacement is a string containing escape sequences representing subexpressions from the pattern, as in the UNIX sed program.

### Creating a name mapping

You can use the vserver name-mapping create command to create a name mapping. You use name mappings to enable Windows users to access UNIX security style volumes and the reverse.

### About this task

For each SVM, ONTAP supports up to 1,024 name mappings for each direction.
Step

1. Create a name mapping:

   \texttt{vserver name-mapping create -vserver vserver\_name -direction \{krb-unix|win-unix|unix-win\} -position integer -pattern text -replacement text}

   \textbf{Note:} The -pattern and -replacement statements can be formulated as regular expressions. You can also use the -replacement statement to explicitly deny a mapping to the user by using the null replacement string " " (the space character). See the \texttt{vserver name-mapping create} man page for details.

When Windows-to-UNIX mappings are created, any SMB clients that have open connections to the ONTAP system at the time the new mappings are created must log out and log back in to see the new mappings.

\begin{table}[h]
\centering
\begin{tabular}{|l|}
\hline
Examples \tabularnewline
\hline
\texttt{The following command creates a name mapping on the SVM named vs1. The mapping is a mapping from UNIX to Windows at position 1 in the priority list. The mapping maps the UNIX user johnd to the Windows user ENG\JohnDoe.} \tabularnewline
\texttt{vsl1::> vserver name-mapping create -vserver vs1 -direction unix-win -position 1 -pattern johnd -replacement "ENG\\JohnDoe"} \tabularnewline
\hline
\texttt{The following command creates another name mapping on the SVM named vs1. The mapping is a mapping from Windows to UNIX at position 1 in the priority list. Here the pattern and replacement include regular expressions. The mapping maps every CIFS user in the domain ENG to users in the LDAP domain associated with the SVM.} \tabularnewline
\texttt{vsl1::> vserver name-mapping create -vserver vs1 -direction win-unix -position 1 -pattern "ENG\((.+\))" -replacement \"\1\"} \tabularnewline
\hline
\texttt{The following command creates another name mapping on the SVM named vs1. Here the pattern includes "$" as an element in the Windows user name that must be escaped. The mapping maps the windows user ENG\john$ops to UNIX user john_ops.} \tabularnewline
\texttt{vsl1::> vserver name-mapping create -direction win-unix -position 1 -pattern ENG\john$ops -replacement john\_ops} \tabularnewline
\hline
\end{tabular}
\end{table}

Related references

\textit{Commands for managing name mappings} on page 51

Configuring the default user

You can configure a default user to use if all other mapping attempts fail for a user, or if you do not want to map individual users between UNIX and Windows. Alternatively, if you want authentication of non-mapped users to fail, you should not configure a default user.

About this task

For CIFS authentication, if you do not want to map each Windows user to an individual UNIX user, you can instead specify a default UNIX user.

For NFS authentication, if you do not want to map each UNIX user to an individual Windows user, you can instead specify a default Windows user.
Step

1. Perform one of the following actions:

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Enter the following command...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure the default UNIX user</td>
<td><code>vserver cifs options modify -default-unix-user user_name</code></td>
</tr>
<tr>
<td>Configure the default Windows user</td>
<td><code>vserver nfs modify -default-win-user user_name</code></td>
</tr>
</tbody>
</table>

Related concepts

- Configuring name mappings on page 47

Related tasks

- Handling NFS access to NTFS volumes or qtrees for unknown UNIX users on page 56

Commands for managing name mappings

There are specific ONTAP commands for managing name mappings.

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Use this command...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a name mapping</td>
<td><code>vserver name-mapping create</code></td>
</tr>
<tr>
<td>Insert a name mapping at a specific position</td>
<td><code>vserver name-mapping insert</code></td>
</tr>
<tr>
<td>Display name mappings</td>
<td><code>vserver name-mapping show</code></td>
</tr>
<tr>
<td>Exchange the position of two name mappings</td>
<td><code>vserver name-mapping swap</code></td>
</tr>
</tbody>
</table>
  
  **Note:** A swap is not allowed when name-mapping is configured with an ip-qualifier entry.
| Modify a name mapping | `vserver name-mapping modify` |
| Delete a name mapping | `vserver name-mapping delete` |
| Validate the correct name mapping | `vserver security file-directory show-effective-permissions -vserver vsl -win-user-name user1 -path / -share-name sh1` |

See the man page for each command for more information.

Related tasks

- Creating a name mapping on page 49

Enabling access for Windows NFS clients

ONTAP supports file access from Windows NFSv3 clients. This means that clients running Windows operating systems with NFSv3 support can now access files on NFSv3 exports on the cluster. To
successfully use this functionality, you must properly configure the storage virtual machine (SVM) and be aware of certain requirements and limitations.

**Before you begin**

NFSv3 must be enabled on the SVM.

**About this task**

By default, Windows NFSv3 client support is disabled.

Windows NFSv3 clients do not support the network status monitor (NSM) protocol. As a result, Windows NFSv3 client sessions might experience disruptions during storage failover and volume move operations.

**Steps**

1. **Enable Windows NFSv3 client support:**
   
   ```
   vserver nfs modify -vserver vserver_name -v3-ms-dos-client enabled
   ```

2. **On all SVMs that support Windows NFSv3 clients, disable the `enable-ejukebox` and `v3-connection-drop` parameters:**
   
   ```
   vserver nfs modify -vserver vserver_name -enable-ejukebox false -v3-connection-drop disabled
   ```

   Windows NFSv3 clients can now mount exports on the storage system.

3. **Ensure that each Windows NFSv3 client uses hard mounts by specifying the `-o mtype=hard` option.**

   This is required to ensure reliable mounts.

   **Example**

   ```
   mount -o mtype=hard \10.53.33.10\vol\vol1 z:\
   ```

**How to view NFS exports of SVMs on NFS client**

NFS clients can use the `showmount -e` command to see a list of exports available on an NFS server. This can help users identify the file system they want to mount on the NFS server.

By default, ONTAP allows NFS clients to view the export list. You can use the `showmount -e` command to see a list of exports available on an NFS server. For viewing the export list, NFSv3 should be enabled on the SVM.

**Example**

The following command shows the showmount feature on the SVM named vs1:

```
cluster1 : : > vserver nfs show -vserver vs1 -fields showmount
vserver showmount
------- -------
vs1      enabled
```

The following command executed on an NFS client displays the list of exports on an NFS server with the IP address 10.63.21.9:
showmount -e 10.63.21.9
Export list for 10.63.21.9:
/unix (everyone)
/unix/unix1 (everyone)
/ unix/unix2 (everyone)
/ (everyone)
Managing file access using NFS

After you have enabled NFS on the storage virtual machine (SVM) and configured it, there are a number of tasks you might want to perform to manage file access using NFS.

Enabling or disabling NFSv3

You can enable or disable NFSv3 by modifying the \(-v3\) option. This allows file access for clients using the NFSv3 protocol. By default, NFSv3 is enabled.

Step

1. Perform one of the following actions:

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Enter the command...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable NFSv3</td>
<td><code>vserver nfs modify -vserver vserver_name -v3 enabled</code></td>
</tr>
<tr>
<td>Disable NFSv3</td>
<td><code>vserver nfs modify -vserver vserver_name -v3 disabled</code></td>
</tr>
</tbody>
</table>

Enabling or disabling NFSv4.0

You can enable or disable NFSv4.0 by modifying the \(-v4.0\) option. This allows file access for clients using the NFSv4.0 protocol. By default, NFSv4.0 is disabled.

About this task

NFSv4.0 is not supported for storage virtual machines (SVMs) with Infinite Volume.

Step

1. Perform one of the following actions:

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Enter the following command...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable NFSv4.0</td>
<td><code>vserver nfs modify -vserver vserver_name -v4.0 enabled</code></td>
</tr>
<tr>
<td>Disable NFSv4.0</td>
<td><code>vserver nfs modify -vserver vserver_name -v4.0 disabled</code></td>
</tr>
</tbody>
</table>

Enabling or disabling NFSv4.1

You can enable or disable NFSv4.1 by modifying the \(-v4.1\) option. This allows file access for clients using the NFSv4.1 protocol. By default, NFSv4.1 is disabled.

Step

1. Perform one of the following actions:
Enabling or disabling pNFS

pNFS improves performance by allowing NFS clients to perform read/write operations on storage devices directly and in parallel, bypassing the NFS server as a potential bottleneck. To enable or disable pNFS (parallel NFS), you can modify the `-v4.1-pnfs` option. By default pNFS is enabled.

**Before you begin**

NFSv4.1 support is required to be able to use pNFS.

If you want to enable pNFS, you must first disable NFS referrals. They cannot both be enabled at the same time.

If you use pNFS with Kerberos on SVMs, you must enable Kerberos on every LIF on the SVM.

**Step**

1. Perform one of the following actions:

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Enter the command...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable pNFS</td>
<td><code>vserver nfs modify -vserver vserver_name -v4.1-pnfs enabled</code></td>
</tr>
<tr>
<td>Disable pNFS</td>
<td><code>vserver nfs modify -vserver vserver_name -v4.1-pnfs disabled</code></td>
</tr>
</tbody>
</table>

**Related tasks**

*Enabling or disabling NFSv4 referrals* on page 86

*Enabling or disabling NFSv4.1* on page 54

Controlling NFS access over TCP and UDP

You can enable or disable NFS access to storage virtual machines (SVMs) over TCP and UDP by modifying the `-tcp` and `-udp` parameters, respectively. This enables you to control whether NFS clients can access data over TCP or UDP in your environment.

**About this task**

These parameters only apply to NFS. They do not affect auxiliary protocols. For example, if NFS over TCP is disabled, mount operations over TCP still succeed. To completely block TCP or UDP traffic, you can use export policy rules.

**Note:** You must turn off the SnapDiff RPC Server before you disable TCP for NFS to avoid a command failed error. You can disable TCP by using the command `vserver snapdiff-rpc-server off -vserver vserver name`. 
Step

1. Perform one of the following actions:

<table>
<thead>
<tr>
<th>If you want NFS access to be...</th>
<th>Enter the command...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled over TCP</td>
<td>vserver nfs modify -vserver vserver_name -tcp enabled</td>
</tr>
<tr>
<td>Disabled over TCP</td>
<td>vserver nfs modify -vserver vserver_name -tcp disabled</td>
</tr>
<tr>
<td>Enabled over UDP</td>
<td>vserver nfs modify -vserver vserver_name -udp enabled</td>
</tr>
<tr>
<td>Disabled over UDP</td>
<td>vserver nfs modify -vserver vserver_name -udp disabled</td>
</tr>
</tbody>
</table>

Related concepts

*NFSv3 and NFSv4 performance improvement by modifying the TCP transfer size* on page 92

Controlling NFS requests from nonreserved ports

You can reject NFS mount requests from nonreserved ports by enabling the `-mount-rootonly` option. To reject all NFS requests from nonreserved ports, you can enable the `-nfs-rootonly` option.

**About this task**

By default, the option `-mount-rootonly` is **enabled**.

By default, the option `-nfs-rootonly` is **disabled**.

These options do not apply to the NULL procedure.

Step

1. Perform one of the following actions:

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Enter the command...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allow NFS mount requests from nonreserved ports</td>
<td>vserver nfs modify -vserver vserver_name -mount-rootonly disabled</td>
</tr>
<tr>
<td>Reject NFS mount requests from nonreserved ports</td>
<td>vserver nfs modify -vserver vserver_name -mount-rootonly enabled</td>
</tr>
<tr>
<td>Allow all NFS requests from nonreserved ports</td>
<td>vserver nfs modify -vserver vserver_name -nfs-rootonly disabled</td>
</tr>
<tr>
<td>Reject all NFS requests from nonreserved ports</td>
<td>vserver nfs modify -vserver vserver_name -nfs-rootonly enabled</td>
</tr>
</tbody>
</table>

Handling NFS access to NTFS volumes or qtrees for unknown UNIX users

If ONTAP cannot identify UNIX users attempting to connect to volumes or qtrees with NTFS security style, it therefore cannot explicitly map the user to a Windows user. You can configure
ONTAP to either deny access to such users for stricter security or map them to a default Windows user to ensure a minimum level of access for all users.

**Before you begin**

A default Windows user must be configured if you want to enable this option.

**About this task**

If a UNIX user tries to access volumes or qtrees with NTFS security style, the UNIX user must first be mapped to a Windows user so that ONTAP can properly evaluate the NTFS permissions. However, if ONTAP cannot look up the name of the UNIX user in the configured user information name service sources, it cannot explicitly map the UNIX user to a specific Windows user. You can decide how to handle such unknown UNIX users in the following ways:

- Deny access to unknown UNIX users.
  This enforces stricter security by requiring explicit mapping for all UNIX users to gain access to NTFS volumes or qtrees.

- Map unknown UNIX users to a default Windows user.
  This provides less security but more convenience by ensuring that all users get a minimum level of access to NTFS volumes or qtrees through a default Windows user.

**Steps**

1. Set the privilege level to advanced:
   ```
   set -privilege advanced
   ```

2. Perform one of the following actions:

<table>
<thead>
<tr>
<th>If you want the default Windows user for unknown UNIX users...</th>
<th>Enter the command...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled</td>
<td>vserver nfs modify -vserver vserver_name -map-unknown-uid-to-default-windows-user enabled</td>
</tr>
<tr>
<td>Disabled</td>
<td>vserver nfs modify -vserver vserver_name -map-unknown-uid-to-default-windows-user disabled</td>
</tr>
</tbody>
</table>

3. Return to the admin privilege level:
   ```
   set -privilege admin
   ```

**Related concepts**

*Configuring name mappings* on page 47

**Related tasks**

*Configuring the default user* on page 50
Considerations for clients that mount NFS exports using a nonreserved port

The `-mount-rootonly` option must be disabled on a storage system that must support clients that mount NFS exports using a nonreserved port even when the user is logged in as root. Such clients include Hummingbird clients and Solaris NFS/IPv6 clients.

If the `-mount-rootonly` option is enabled, ONTAP does not allow NFS clients that use nonreserved ports, meaning ports with numbers higher than 1,023, to mount NFS exports.

Performing stricter access checking for netgroups by verifying domains

By default, ONTAP performs an additional verification when evaluating client access for a netgroup. The additional check ensures that the client's domain matches the domain configuration of the storage virtual machine (SVM). Otherwise, ONTAP denies client access.

About this task

When ONTAP evaluates export policy rules for client access and an export policy rule contains a netgroup, ONTAP must determine whether a client's IP address belongs to the netgroup. For this purpose, ONTAP converts the client's IP address to a host name using DNS and obtains a fully qualified domain name (FQDN).

If the netgroup file only lists a short name for the host and the short name for the host exists in multiple domains, it is possible for a client from a different domain to obtain access without this check.

To prevent this, ONTAP compares the domain that was returned from DNS for the host against the list of DNS domain names configured for the SVM. If it matches, access is allowed. If it does not match, access is denied.

This verification is enabled by default. You can manage it by modifying the `-netgroup-dns-domain-search` parameter, which is available at the advanced privilege level.

Steps

1. Set the privilege level to advanced:
   
   set -privilege advanced

2. Perform the desired action:
   
   If you want domain verification for netgroups to be...
   
   Enter...
   
   | Enabled            | vserver nfs modify -vserver vserver_name -netgroup-dns-domain-search enabled |
   |                   |Disabled            | vserver nfs modify -vserver vserver_name -netgroup-dns-domain-search disabled |

3. Set the privilege level to admin:
   
   set -privilege admin
Modifying ports used for NFSv3 services

The NFS server on the storage system uses services such as mount daemon and Network Lock Manager to communicate with NFS clients over specific default network ports. In most NFS environments the default ports work correctly and do not require modification, but if you want to use different NFS network ports in your NFSv3 environment, you can do so.

Before you begin

Changing NFS ports on the storage system requires that all NFS clients reconnect to the system, so you should communicate this information to your users in advance of making the change.

About this task

You can set the ports used by the NFS mount daemon, Network Lock Manager, Network Status Monitor, and NFS quota daemon services for each storage virtual machine (SVM). The port number change affects NFS clients accessing data over both TCP and UDP. Ports for NFSv4 and NFSv4.1 cannot be changed.

Steps

1. Set the privilege level to advanced:
   ```bash
   set -privilege advanced
   ```

2. Disable access to NFS:
   ```bash
   vserver nfs modify -vserver vserver_name -access false
   ```

3. Set the NFS port for the specific NFS service:
   ```bash
   vserver nfs modify -vserver vserver_name nfs_port_parameter port_number
   ```

<table>
<thead>
<tr>
<th>NFS port parameter</th>
<th>Description</th>
<th>Default port</th>
</tr>
</thead>
<tbody>
<tr>
<td>-mountd-port</td>
<td>NFS mount daemon</td>
<td>635</td>
</tr>
<tr>
<td>-nlm-port</td>
<td>Network Lock Manager</td>
<td>4045</td>
</tr>
<tr>
<td>-nsm-port</td>
<td>Network Status Monitor</td>
<td>4046</td>
</tr>
<tr>
<td>-rquotad-port</td>
<td>NFS quota daemon</td>
<td>4049</td>
</tr>
</tbody>
</table>

Besides the default port, the allowed range of port numbers is 1024 through 65535. Each NFS service must use a unique port.

4. Enable access to NFS:
   ```bash
   vserver nfs modify -vserver vserver_name -access true
   ```

5. Use the `network connections listening show` command to verify the port number changes.

6. Return to the admin privilege level:
   ```bash
   set -privilege admin
   ```

Example

The following commands set the NFS Mount Daemon port to 1113 on the SVM named vs1:

```bash
vserver nfs modify -vserver vs1 -mountd-port 1113
```
vs1::> set -privilege 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
vs1::*> vserver nfs modify -vserver vs1 -access false
vs1::*> vserver nfs modify -vserver vs1 -mountd-port 1113
vs1::*> vserver nfs modify -vserver vs1 -access true
vs1::*> network connections listening show

<table>
<thead>
<tr>
<th>Vserver Name</th>
<th>Interface Name:Local Port</th>
<th>Protocol/Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster</td>
<td>cluster1-01_clus_1:7700</td>
<td>TCP/ctlopcp</td>
</tr>
<tr>
<td>vs1</td>
<td>data1:4046</td>
<td>TCP/sm</td>
</tr>
<tr>
<td>vs1</td>
<td>data1:4046</td>
<td>UDP/sm</td>
</tr>
<tr>
<td>vs1</td>
<td>data1:4045</td>
<td>TCP/nlm-v4</td>
</tr>
<tr>
<td>vs1</td>
<td>data1:4045</td>
<td>UDP/nlm-v4</td>
</tr>
<tr>
<td>vs1</td>
<td>data1:1113</td>
<td>TCP/mount</td>
</tr>
<tr>
<td>vs1</td>
<td>data1:1113</td>
<td>UDP/mount</td>
</tr>
</tbody>
</table>

vs1::*> set -privilege admin

Commands for managing NFS servers

There are specific ONTAP commands for managing NFS servers.

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Use this command...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create an NFS server</td>
<td>vserver nfs create</td>
</tr>
<tr>
<td>Display NFS servers</td>
<td>vserver nfs show</td>
</tr>
<tr>
<td>Modify an NFS server</td>
<td>vserver nfs modify</td>
</tr>
<tr>
<td>Delete an NFS server</td>
<td>vserver nfs delete</td>
</tr>
</tbody>
</table>

See the man page for each command for more information.

Troubleshooting name service issues

When clients experience access failures due to name service issues, you can use the vserver services name-service getxbyyyy command family to manually perform various name service lookups and examine the details and results of the lookup to help with troubleshooting.

About this task

- For each command, you can specify the following:
  - Name of the node or storage virtual machine (SVM) to perform the lookup on.
    This enables you to test name service lookups for a specific node or SVM to narrow the search for a potential name service configuration issue.
  - Whether to show the source used for the lookup.
    This enables you to check whether the correct source was used.
• ONTAP selects the service for performing the lookup based on the configured name service switch order.

• These commands are available at the advanced privilege level.

Steps

1. Perform one of the following actions:

<table>
<thead>
<tr>
<th>To retrieve the...</th>
<th>Use the command...</th>
</tr>
</thead>
</table>
| IP address of a host name | vserver services name-service getxxbyyyy getaddrinfo  
|                       | vserver services name-service getxxbyyyy gethostbyname (IPv4 addresses only)  |
| Members of a group by group ID | vserver services name-service getxxbyyyy getgrbygid  |
| Members of a group by group name | vserver services name-service getxxbyyyy getgrbyname  |
| List of groups a user belongs to | vserver services name-service getxxbyyyy getgrlist  |
| Host name of an IP address | vserver services name-service getxxbyyyy getnameinfo  
|                       | vserver services name-service getxxbyyyy gethostbyaddr (IPv4 addresses only)  |
| User information by user name | vserver services name-service getxxbyyyy getpwnamename  
|                       | You can test name resolution of RBAC users by specifying the -use-rbac parameter as true.  |
| User information by user ID | vserver services name-service getxxbyyyy getpwnamoid  
|                       | You can test name resolution of RBAC users by specifying the -use-rbac parameter as true.  |
| Netgroup membership of a client | vserver services name-service getxxbyyyy netgrp  |
| Netgroup membership of a client using netgroup-by-host search | vserver services name-service getxxbyyyy netgrppbyhost  |

Example

The following example shows a DNS lookup test for the SVM vs1 by attempting to obtain the IP address for the host acast1.eng.example.com:

```
cluster1::*> vserver services name-service getxxbyyyy getaddrinfo -vserver vs1 -hostname acast1.eng.example.com -address-family all -show-source true
Source used for lookup: DNS
Host name: acast1.eng.example.com
Canonical Name: acast1.eng.example.com
IPv4: 10.72.8.29
```

The following example shows a NIS lookup test for the SVM vs1 by attempting to retrieve user information for a user with the UID 501768:
The following example shows an LDAP lookup test for the SVM vs1 by attempting to retrieve user information for a user with the name ldap1:

```
cluster1::*> vserver services name-service getxbyname -vserver vs1 -username ldap1 -use-rbac false -show-source true
Source used for lookup: LDAP
pw_name: ldap1
pw_passwd: {crypt}JSPM6yc/ilIX6
pw_uid: 10001
pw_gid: 3333
pw_gecos: ldap1 user
pw_dir: /u/ldap1
pw_shell: /bin/csh
```

The following example shows a netgroup lookup test for the SVM vs1 by attempting to find out whether the client dnshost0 is a member of the netgroup lnetgroup136:

```
cluster1::*> vserver services name-service getxbyname netgrp -vserver vs1 -netgroup lnetgroup136 -client dnshost0 -show-source true
Source used for lookup: LDAP
dnshost0 is a member of lnetgroup136
```

2. Analyze the results of the test you performed and take the necessary action.

**Example**

<table>
<thead>
<tr>
<th>If the...</th>
<th>Check the...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host name or IP address lookup failed or yielded incorrect results</td>
<td>DNS configuration</td>
</tr>
<tr>
<td>Lookup queried an incorrect source</td>
<td>Name service switch configuration</td>
</tr>
</tbody>
</table>
| User or group lookup failed or yielded incorrect results | Name service switch configuration  
Source configuration (local files, NIS domain, LDAP client)  
Network configuration (for example, LIFs and routes) |

**Related information**

Verifying name service connections

Starting in ONTAP 9.2, you can check DNS and LDAP name servers to verify that they are connected to ONTAP. These commands are available at the admin privilege level.

About this task

You can check for a valid DNS or LDAP name service configuration on an as-needed basis using the name service configuration checker. This validation check can be initiated at the command line or in System Manager.

Step

1. Use the appropriate command to check a name service configuration. The UI displays the status of the configured servers.

<table>
<thead>
<tr>
<th>To check ...</th>
<th>Use this command ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNS configuration status</td>
<td>vserver services name-service dns check</td>
</tr>
<tr>
<td>LDAP configuration status</td>
<td>vserver services name-service ldap check</td>
</tr>
</tbody>
</table>

Example

```
cluster1::> vserver services name-service dns check -vserver vs0

Vserver         Name Server     Status  Status Details
--------------  --------------- ------- -------------------------
vs0            10.11.12.13     up      Response time (msec): 55
vs0            10.11.12.14     up      Response time (msec): 70
vs0            10.11.12.15     down    Connection refused.

Example

cluster1::> vserver services name-service ldap check -vserver vs0

Vserver: vs0
Client Configuration Name: c1
LDAP Status: up
LDAP Status Details: Successfully connected to LDAP server "10.11.12.13".
```

Configuration validation is successful if at least one of the configured servers (name-servers/ldap-servers) is reachable and providing the service. A warning is shown if some of the servers are not reachable.

Commands for managing name service switch entries

You can manage name service switch entries by creating, displaying, modifying, and deleting them.

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Use this command...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a name service switch entry</td>
<td>vserver services name-service ns-switch create</td>
</tr>
</tbody>
</table>
If you want to... | Use this command...
---|---
Display name service switch entries | vserver services name-service ns-switch show
Modify a name service switch entry | vserver services name-service ns-switch modify
Delete a name service switch entry | vserver services name-service ns-switch delete

See the man page for each command for more information.

**Related information**


### Commands for managing name service cache

You can manage name service cache by modifying the time to live (TTL) value. The TTL value determines how long name service information is persistent in cache.

<table>
<thead>
<tr>
<th>If you want to modify the TTL value for...</th>
<th>Use this command...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unix users</td>
<td>vserver services name-service cache unix-user settings</td>
</tr>
<tr>
<td>Unix groups</td>
<td>vserver services name-service cache unix-group settings</td>
</tr>
<tr>
<td>Unix netgroups</td>
<td>vserver services name-service cache netgroups settings</td>
</tr>
<tr>
<td>Hosts</td>
<td>vserver services name-service cache hosts settings</td>
</tr>
<tr>
<td>Group membership</td>
<td>vserver services name-service cache group-membership settings</td>
</tr>
</tbody>
</table>

**Related information**

*ONTAP 9 commands*

### Commands for managing name mappings

There are specific ONTAP commands for managing name mappings.

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Use this command...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a name mapping</td>
<td>vserver name-mapping create</td>
</tr>
<tr>
<td>Insert a name mapping at a specific position</td>
<td>vserver name-mapping insert</td>
</tr>
<tr>
<td>Display name mappings</td>
<td>vserver name-mapping show</td>
</tr>
<tr>
<td>Exchange the position of two name mappings</td>
<td>vserver name-mapping swap</td>
</tr>
</tbody>
</table>

**Note:** A swap is not allowed when name-mapping is configured with an ip-qualifier entry.
If you want to... | Use this command...
---|---
Modify a name mapping | `vserver name-mapping modify`
Delete a name mapping | `vserver name-mapping delete`
Validate the correct name mapping | `vserver security file-directory show-effective-permissions -vserver vs1 -win-user-name user1 -path / -share-name sh1`

See the man page for each command for more information.

**Related tasks**

- *Creating a name mapping* on page 49

---

### Commands for managing local UNIX users

There are specific ONTAP commands for managing local UNIX users.

If you want to... | Use this command...
---|---
Create a local UNIX user | `vserver services name-service unix-user create`
Load local UNIX users from a URI | `vserver services name-service unix-user load-from-uri`
Display local UNIX users | `vserver services name-service unix-user show`
Modify a local UNIX user | `vserver services name-service unix-user modify`
Delete a local UNIX user | `vserver services name-service unix-user delete`

See the man page for each command for more information.

### Commands for managing local UNIX groups

There are specific ONTAP commands for managing local UNIX groups.

If you want to... | Use this command...
---|---
Create a local UNIX group | `vserver services name-service unix-group create`
Add a user to a local UNIX group | `vserver services name-service unix-group adduser`
Load local UNIX groups from a URI | `vserver services name-service unix-group load-from-uri`
Display local UNIX groups | `vserver services name-service unix-group show`
Modify a local UNIX group | `vserver services name-service unix-group modify`
If you want to... | Use this command...
---|---
Delete a user from a local UNIX group | `vserver services name-service unix-group deluser`
Delete a local UNIX group | `vserver services name-service unix-group delete`

See the man page for each command for more information.

Limits for local UNIX users, groups, and group members

ONTAP introduced limits for the maximum number of UNIX users and groups in the cluster, and commands to manage these limits. These limits can help avoid performance issues by preventing administrators from creating too many local UNIX users and groups in the cluster.

There is a limit for the combined number of local UNIX user groups and group members. There is a separate limit for local UNIX users. The limits are cluster-wide. Each of these new limits is set to a default value that you can modify up to a preassigned hard limit.

<table>
<thead>
<tr>
<th>Database</th>
<th>Default limit</th>
<th>Hard limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local UNIX users</td>
<td>32,768</td>
<td>65,536</td>
</tr>
<tr>
<td>Local UNIX groups and group members</td>
<td>32,768</td>
<td>65,536</td>
</tr>
</tbody>
</table>

Managing limits for local UNIX users and groups

There are specific ONTAP commands for managing limits for local UNIX users and groups. Cluster administrators can use these commands to troubleshoot performance issues in the cluster believed to be related to excessive numbers of local UNIX users and groups.

About this task

These commands are available to the cluster administrator at the advanced privilege level.

Step

1. Perform one of the following actions:

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Use the command...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display information about local UNIX user limits</td>
<td><code>vserver services unix-user max-limit show</code></td>
</tr>
<tr>
<td>Display information about local UNIX group limits</td>
<td><code>vserver services unix-group max-limit show</code></td>
</tr>
<tr>
<td>Modify local UNIX user limits</td>
<td><code>vserver services unix-user max-limit modify</code></td>
</tr>
<tr>
<td>Modify local UNIX group limits</td>
<td><code>vserver services unix-group max-limit modify</code></td>
</tr>
</tbody>
</table>

See the man page for each command for more information.
Commands for managing local netgroups

You can manage local netgroups by loading them from a URI, verifying their status across nodes, displaying them, and deleting them.

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Use the command...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load netgroups from a URI</td>
<td><code>vserver services name-service netgroup load</code></td>
</tr>
<tr>
<td>Verify the status of netgroups across nodes</td>
<td><code>vserver services name-service netgroup status</code>&lt;br&gt;Available at the advanced privilege level and higher.</td>
</tr>
<tr>
<td>Display local netgroups</td>
<td><code>vserver services name-service netgroup file show</code></td>
</tr>
<tr>
<td>Delete a local netgroup</td>
<td><code>vserver services name-service netgroup file delete</code></td>
</tr>
</tbody>
</table>

See the man page for each command for more information.

Related tasks

- **Checking whether a client IP address is a member of a netgroup** on page 74
- **Performing stricter access checking for netgroups by verifying domains** on page 58

Commands for managing NIS domain configurations

There are specific ONTAP commands for managing NIS domain configurations.

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Use this command...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a NIS domain configuration</td>
<td><code>vserver services name-service nis-domain create</code></td>
</tr>
<tr>
<td>Display NIS domain configurations</td>
<td><code>vserver services name-service nis-domain show</code></td>
</tr>
<tr>
<td>Display binding status of a NIS domain configuration</td>
<td><code>vserver services name-service nis-domain show-bound</code></td>
</tr>
<tr>
<td>Display NIS statistics</td>
<td><code>vserver services name-service nis-domain show-statistics</code>&lt;br&gt;Available at the advanced privilege level and higher.</td>
</tr>
<tr>
<td>Clear NIS statistics</td>
<td><code>vserver services name-service nis-domain clear-statistics</code>&lt;br&gt;Available at the advanced privilege level and higher.</td>
</tr>
<tr>
<td>Modify a NIS domain configuration</td>
<td><code>vserver services name-service nis-domain modify</code></td>
</tr>
<tr>
<td>Delete a NIS domain configuration</td>
<td><code>vserver services name-service nis-domain delete</code></td>
</tr>
</tbody>
</table>

See the man page for each command for more information.
**Commands for managing LDAP client configurations**

There are specific ONTAP commands for managing LDAP client configurations.

**Note:** SVM administrators cannot modify or delete LDAP client configurations that were created by cluster administrators.

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Use this command...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create an LDAP client configuration</td>
<td><code>vserver services name-service ldap client create</code></td>
</tr>
<tr>
<td>Display LDAP client configurations</td>
<td><code>vserver services name-service ldap client show</code></td>
</tr>
<tr>
<td>Modify an LDAP client configuration</td>
<td><code>vserver services name-service ldap client modify</code></td>
</tr>
<tr>
<td>Change the LDAP client BIND password</td>
<td><code>vserver services name-service ldap client modify-bind-password</code></td>
</tr>
<tr>
<td>Delete an LDAP client configuration</td>
<td><code>vserver services name-service ldap client delete</code></td>
</tr>
</tbody>
</table>

See the man page for each command for more information.

**Related concepts**

*Configuration options for LDAP directory searches* on page 43

**Commands for managing LDAP configurations**

There are specific ONTAP commands for managing LDAP configurations.

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Use this command...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create an LDAP configuration</td>
<td><code>vserver services name-service ldap create</code></td>
</tr>
<tr>
<td>Display LDAP configurations</td>
<td><code>vserver services name-service ldap show</code></td>
</tr>
<tr>
<td>Modify an LDAP configuration</td>
<td><code>vserver services name-service ldap modify</code></td>
</tr>
<tr>
<td>Delete an LDAP configuration</td>
<td><code>vserver services name-service ldap delete</code></td>
</tr>
</tbody>
</table>

See the man page for each command for more information.

**Commands for managing LDAP client schema templates**

There are specific ONTAP commands for managing LDAP client schema templates.

**Note:** SVM administrators cannot modify or delete LDAP client schemas that were created by cluster administrators.
### Managing file access using NFS

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Use this command...</th>
</tr>
</thead>
</table>
| Copy an existing LDAP schema template | `vserver services name-service ldap client schema copy`  
Available at the advanced privilege level and higher. |
| Display LDAP schema templates | `vserver services name-service ldap client schema show` |
| Modify an LDAP schema template | `vserver services name-service ldap client schema modify`  
Available at the advanced privilege level and higher. |
| Delete an LDAP schema template | `vserver services name-service ldap client schema delete`  
Available at the advanced privilege level and higher. |

See the man page for each command for more information.

## Commands for managing NFS Kerberos interface configurations

There are specific ONTAP commands for managing NFS Kerberos interface configurations.

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Use this command...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable NFS Kerberos on a LIF</td>
<td><code>vserver nfs kerberos interface enable</code></td>
</tr>
<tr>
<td>Display NFS Kerberos interface configurations</td>
<td><code>vserver nfs kerberos interface show</code></td>
</tr>
<tr>
<td>Modify an NFS Kerberos interface configuration</td>
<td><code>vserver nfs kerberos interface modify</code></td>
</tr>
<tr>
<td>Disable NFS Kerberos on a LIF</td>
<td><code>vserver nfs kerberos interface disable</code></td>
</tr>
</tbody>
</table>

See the man page for each command for more information.

## Commands for managing NFS Kerberos realm configurations

There are specific ONTAP commands for managing NFS Kerberos realm configurations.

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Use this command...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create an NFS Kerberos realm configuration</td>
<td><code>vserver nfs kerberos realm create</code></td>
</tr>
<tr>
<td>Display NFS Kerberos realm configurations</td>
<td><code>vserver nfs kerberos realm show</code></td>
</tr>
<tr>
<td>Modify an NFS Kerberos realm configuration</td>
<td><code>vserver nfs kerberos realm modify</code></td>
</tr>
<tr>
<td>Delete an NFS Kerberos realm configuration</td>
<td><code>vserver nfs kerberos realm delete</code></td>
</tr>
</tbody>
</table>

See the man page for each command for more information.
Commands for managing export policies

There are specific ONTAP commands for managing export policies.

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Use this command...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display information about export policies</td>
<td><code>vserver export-policy show</code></td>
</tr>
<tr>
<td>Rename an export policy</td>
<td><code>vserver export-policy rename</code></td>
</tr>
<tr>
<td>Copy an export policy</td>
<td><code>vserver export-policy copy</code></td>
</tr>
<tr>
<td>Delete an export policy</td>
<td><code>vserver export-policy delete</code></td>
</tr>
</tbody>
</table>

See the man page for each command for more information.

Commands for managing export rules

There are specific ONTAP commands for managing export rules.

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Use this command...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create an export rule</td>
<td><code>vserver export-policy rule create</code></td>
</tr>
<tr>
<td>Display information about export rules</td>
<td><code>vserver export-policy rule show</code></td>
</tr>
<tr>
<td>Modify an export rule</td>
<td><code>vserver export-policy rule modify</code></td>
</tr>
<tr>
<td>Delete an export rule</td>
<td><code>vserver export-policy rule delete</code></td>
</tr>
</tbody>
</table>

Note: If you have configured multiple identical export rules matching different clients, be sure to keep them in sync when managing export rules.

See the man page for each command for more information.

Configuring the NFS credential cache

ONTAP uses a credential cache to store information needed for user authentication for NFS export access to provide faster access and improve performance. You can configure how long information is stored in the credential cache to customize it for your environment.

Reasons for modifying the NFS credential cache time-to-live

There are several scenarios when modifying the NFS credential cache time-to-live (TTL) can help resolve issues. You should understand what these scenarios are as well as the consequences of making these modifications.

Reasons

Consider changing the default TTL under the following circumstances:

<table>
<thead>
<tr>
<th>Issue</th>
<th>Remedial action</th>
</tr>
</thead>
<tbody>
<tr>
<td>The name servers in your environment are experiencing performance degradation due to a high load of requests from ONTAP.</td>
<td>Increase the TTL for cached positive and negative credentials to reduce the number of requests from ONTAP to name servers.</td>
</tr>
</tbody>
</table>
### Issue

<table>
<thead>
<tr>
<th>The name server administrator made changes to allow access to NFS users that were previously denied.</th>
<th>Decrease the TTL for cached negative credentials to reduce the time NFS users have to wait for ONTAP to request fresh credentials from external name servers so they can get access.</th>
</tr>
</thead>
<tbody>
<tr>
<td>The name server administrator made changes to deny access to NFS users that were previously allowed.</td>
<td>Reduce the TTL for cached positive credentials to reduce the time before ONTAP requests fresh credentials from external name servers so the NFS users are now denied access.</td>
</tr>
</tbody>
</table>

#### Consequences

You can modify the length of time individually for caching positive and negative credentials. However, you should be aware of both the advantages and disadvantages of doing so.

<table>
<thead>
<tr>
<th>If you...</th>
<th>The advantage is...</th>
<th>The disadvantage is...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Increase the positive credential cache time</td>
<td>ONTAP sends requests for credentials to name servers less frequently, reducing the load on name servers.</td>
<td>It takes longer to deny access to NFS users that previously were allowed access but are not anymore.</td>
</tr>
<tr>
<td>Decrease the positive credential cache time</td>
<td>It takes less time to deny access to NFS users that previously were allowed access but are not anymore.</td>
<td>ONTAP sends requests for credentials to name servers more frequently, increasing the load on name servers.</td>
</tr>
<tr>
<td>Increase the negative credential cache time</td>
<td>ONTAP sends requests for credentials to name servers less frequently, reducing the load on name servers.</td>
<td>It takes longer to grant access to NFS users that previously were not allowed access but are now.</td>
</tr>
<tr>
<td>Decrease the negative credential cache time</td>
<td>It takes less time to grant access to NFS users that previously were not allowed access but are now.</td>
<td>ONTAP sends requests for credentials to name servers more frequently, increasing the load on name servers.</td>
</tr>
</tbody>
</table>

### Configuring the time-to-live for cached NFS user credentials

You can configure the length of time that ONTAP stores credentials for NFS users in its internal cache (time-to-live, or TTL) by modifying the NFS server of the storage virtual machine (SVM). This enables you to alleviate certain issues related to high load on name servers or changes in credentials affecting NFS user access.

#### About this task

These parameters are available at the advanced privilege level.

#### Steps

1. Set the privilege level to advanced:
   ```
   set -privilege advanced
   ```
2. Perform the desired action:
If you want to modify the TTL for cached...

<table>
<thead>
<tr>
<th>Positive credentials</th>
<th>Use the command...</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>\texttt{vserver nfs modify -vserver vserver_name _cached-cred-positive-ttl _time_to_live}</td>
</tr>
<tr>
<td></td>
<td>The TTL is measured in milliseconds. The default is 24 hours (86,400,000 milliseconds). The allowed range for this value is 1 minute (60000 milliseconds) through 7 days (604,800,000 milliseconds).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Negative credentials</th>
<th>Use the command...</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>\texttt{vserver nfs modify -vserver vserver_name _cached-cred-negative-ttl _time_to_live}</td>
</tr>
<tr>
<td></td>
<td>The TTL is measured in milliseconds. The default is 2 hours (7,200,000 milliseconds). The allowed range for this value is 1 minute (60000 milliseconds) through 7 days (604,800,000 milliseconds).</td>
</tr>
</tbody>
</table>

3. Return to the admin privilege level:

\texttt{set -privilege admin}

### Managing export policy caches

ONTAP uses several export policy caches to store information related to export policies for faster access. There are certain tasks you can perform to manage export policy caches for troubleshooting purposes.

#### Flushing export policy caches

Flushing export policy caches manually (\texttt{vserver export-policy cache flush}) removes potentially outdated information and forces ONTAP to retrieve current information from the appropriate external resources. This can help resolve a variety of issues related to client access to NFS exports.

**About this task**

Export policy cache information might be outdated due to the following reasons:

- A recent change to export policy rules
- A recent change to host name records in name servers
- A recent change to netgroup entries in name servers
- Recovering from a network outage that prevented netgroups from being fully loaded

**Steps**

1. If you do not have name service cache enabled, perform one of the following actions in advance privilege mode:

<table>
<thead>
<tr>
<th>If you want to flush...</th>
<th>Enter the command...</th>
</tr>
</thead>
<tbody>
<tr>
<td>All export policy caches (except for showmount)</td>
<td>\texttt{vserver export-policy cache flush -vserver vserver_name}</td>
</tr>
<tr>
<td>The export policy rules access cache</td>
<td>\texttt{vserver export-policy cache flush -vserver vserver_name -cache access}</td>
</tr>
</tbody>
</table>

You can include the optional \texttt{-node} parameter to specify the node on which you want to flush the access cache.
### Displaying the export policy netgroup queue and cache

ONTAP uses the netgroup queue when importing and resolving netgroups and it uses the netgroup cache to store the resulting information. When troubleshooting export policy netgroup related issues, you can use the `vserver export-policy netgroup queue show` and `vserver export-policy netgroup cache show` commands to display the status of the netgroup queue and the contents of the netgroup cache.

**Step**

1. Perform one of the following actions:

<table>
<thead>
<tr>
<th>To display the export policy netgroup...</th>
<th>Enter the command...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queue</td>
<td><code>vserver export-policy netgroup queue show</code></td>
</tr>
<tr>
<td>Cache</td>
<td><code>vserver export-policy netgroup cache show -vserver vserver_name</code></td>
</tr>
</tbody>
</table>

See the man page for each command for more information.
Checking whether a client IP address is a member of a netgroup

When troubleshooting NFS client access issues related to netgroups, you can use the `vserver export-policy netgroup check-membership` command to help determine whether a client IP is a member of a certain netgroup.

About this task

Checking netgroup membership enables you to determine whether ONTAP is aware that a client is or is not a member of a netgroup. It also lets you know whether the ONTAP netgroup cache is in a transient state while refreshing netgroup information. This information can help you understand why a client might be unexpectedly granted or denied access.

Step

1. Check the netgroup membership of a client IP address:

   ```
   vserver export-policy netgroup check-membership -vserver vserver_name -netgroup netgroup_name -client-ip client_ip
   ```

   The command can return the following results:

   • The client is a member of the netgroup.
     This was confirmed through a reverse lookup scan or a netgroup-by-host search.

   • The client is a member of the netgroup.
     It was found in the ONTAP netgroup cache.

   • The client is not a member of the netgroup.

   • The membership of the client cannot yet be determined because ONTAP is currently refreshing the netgroup cache.
     Until this is done, membership cannot be explicitly ruled in or out. Use the `vserver export-policy netgroup queue show` command to monitor the loading of the netgroup and retry the check after it is finished.

Example

The following example checks whether a client with the IP address 172.17.16.72 is a member of the netgroup mercury on the SVM vs1:

```
cluster1:/> vserver export-policy netgroup check-membership -vserver vs1 -netgroup mercury -client-ip 172.17.16.72
```

Related tasks

*Performing stricter access checking for netgroups by verifying domains* on page 58

Related references

*Commands for managing local netgroups* on page 67
Optimizing access cache performance

You can configure several parameters to optimize the access cache and find the right balance between performance and how current the information stored in the access cache is.

About this task

When you configure the access cache refresh periods, keep the following in mind:

- Higher values mean entries stay longer in the access cache. The advantage is better performance because ONTAP spends less resources on refreshing access cache entries. The disadvantage is that if export policy rules change and access cache entries become stale as a result, it takes longer to update them. As a result, clients that should get access might get denied, and clients that should get denied might get access.

- Lower values mean ONTAP refreshes access cache entries more often. The advantage is that entries are more current and clients are more likely to be correctly granted or denied access. The disadvantage is a decrease in performance because ONTAP spends more resources refreshing access cache entries.

Steps

1. Set the privilege level to advanced:
   set -privilege advanced
2. Perform the desired action:

<table>
<thead>
<tr>
<th>To modify the...</th>
<th>Enter...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Refresh period for positive entries</td>
<td>vserver export-policy access-cache config modify-all-vservers -refresh-period-positive timeout_value</td>
</tr>
<tr>
<td>Refresh period for negative entries</td>
<td>vserver export-policy access-cache config modify-all-vservers -refresh-period-negative timeout_value</td>
</tr>
<tr>
<td>Timeout period for old entries</td>
<td>vserver export-policy access-cache config modify-all-vservers -harvest-timeout timeout_value</td>
</tr>
</tbody>
</table>

3. Verify the new parameter settings:
   vserver export-policy access-cache config show-all-vservers

4. Return to the admin privilege level:
   set -privilege admin

Managing file locks

You can display information about the current locks for an SVM as a first step to determining why a client cannot access a volume or file. You can use this information if you need to break file locks.

Related concepts

Managing NFSv4 file delegations on page 83

Related tasks

Configuring NFSv4 file and record locking on page 84
About file locking between protocols

File locking is a method used by client applications to prevent a user from accessing a file previously opened by another user. How ONTAP locks files depends on the protocol of the client.

If the client is an NFS client, locks are advisory; if the client is an SMB client, locks are mandatory.

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

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

- In mixed or NTFS volumes, file manipulation operations such as `rm`, `rmdir`, and `mv` can cause the NFS application to fail.
- NFS read and write operations are denied by SMB deny-read and deny-write open modes, respectively.
- NFS write operations fail when the written range of the file is locked with an exclusive SMB bytelock.

In UNIX security-style volumes, NFS unlink and rename operations ignore SMB lock state and allow access to the file. All other NFS operations on UNIX security-style volumes honor SMB lock state.

How ONTAP treats read-only bits

The read-only bit is set on a file-by-file basis to reflect whether a file is writable (disabled) or read-only (enabled).

SMB clients that use Windows can set a per-file read-only bit. NFS clients do not set a per-file read-only bit because NFS clients do not have any protocol operations that use a per-file read-only bit.

ONTAP can set a read-only bit on a file when an SMB client that uses Windows creates that file. ONTAP can also set a read-only bit when a file is shared between NFS clients and SMB clients.

Some software, when used by NFS clients and SMB clients, requires the read-only bit to be enabled.

For ONTAP to keep the appropriate read and write permissions on a file shared between NFS clients and SMB clients, it treats the read-only bit according to the following rules:

- NFS treats any file with the read-only bit enabled as if it has no write permission bits enabled.
- If an NFS client disables all write permission bits and at least one of those bits had previously been enabled, ONTAP enables the read-only bit for that file.
- If an NFS client enables any write permission bit, ONTAP disables the read-only bit for that file.
- If the read-only bit for a file is enabled and an NFS client attempts to discover permissions for the file, the permission bits for the file are not sent to the NFS client; instead, ONTAP sends the permission bits to the NFS client with the write permission bits masked.
- If the read-only bit for a file is enabled and an SMB client disables the read-only bit, ONTAP enables the owner’s write permission bit for the file.
- Files with the read-only bit enabled are writable only by root.

**Note:** Changes to file permissions take effect immediately on SMB clients, but might not take effect immediately on NFS clients if the NFS client enables attribute caching.
How ONTAP differs from Windows on handling locks on share path components

Unlike Windows, ONTAP does not lock each component of the path to an open file while the file is open. This behavior also affects SMB share paths.

Because ONTAP does not lock each component of the path, it is possible to rename a path component above the open file or share, which can cause problems for certain applications, or can cause the share path in the SMB configuration to be invalid. This can cause the share to be inaccessible.

To avoid issues caused by renaming path components, you can apply security settings that prevent users or applications from renaming critical directories.

Displaying information about locks

You can display information about the current file locks, including what types of locks are held and what the lock state is, details about byte-range locks, sharelock modes, delegation locks, and opportunistic locks, and whether locks are opened with durable or persistent handles.

About this task

The client IP address cannot be displayed for locks established through NFSv4 or NFSv4.1.

By default, the command displays information about all locks. You can use command parameters to display information about locks for a specific storage virtual machine (SVM) or to filter the command's output by other criteria.

The `vserver locks show` command displays information about four types of locks:

- Byte-range locks, which lock only a portion of a file.
- Share locks, which lock open files.
- Opportunistic locks, which control client-side caching over SMB.
- Delegations, which control client-side caching over NFSv4.x.

By specifying optional parameters, you can determine important information about each lock type.

See the man page for the command for more information.

Step

1. Display information about locks by using the `vserver locks show` command.

Examples

The following example displays summary information for an NFSv4 lock on a file with the path `/vol1/file1`. The sharelock access mode is write-deny_none, and the lock was granted with write delegation:

```
cluster1::> vserver locks show
Vserver: vs0
Volume Object Path            LIF Protocol Lock Type Client
----------------------------- --------- ----------- ----------- -------
voll /vol1/file1             lif1  nfsv4    share-level -
Sharelock Mode: write-deny_none
Delegation Type: write
```

The following example displays detailed oplock and sharelock information about the SMB lock on a file with the path `/data2/data2_2/intro.pptx`. A durable handle is granted on
the file with a share lock access mode of write-deny_none to a client with an IP address of 10.3.1.3. A lease oplock is granted with a batch oplock level:

```
cluster1::> vserver locks show -instance -path /data2/data2_2/intro.pptx
Vserver: vs1
  Volume: data2_2
Logical Interface: lif2
  Object Path: /data2/data2_2/intro.pptx
  Lock UUID: 553cf484-7030-4998-88d3-1125adbb00b7
  Lock Protocol: cifs
  Lock Type: share-level
    Node Holding Lock State: node3
    Lock State: granted
    Bytelock Starting Offset: -
    Number of Bytes Locked: -
    Bytelock is Mandatory: -
    Bytelock is Exclusive: -
    Bytelock is Superlock: -
    Bytelock is Soft: -
    Oplock Level: -
    Shared Lock Access Mode: write-deny_none
      Shared Lock is Soft: false
      Delegation Type: -
      Client Address: 10.3.1.3
      SMB Open Type: durable
      SMB Connect State: connected
      SMB Expiration Time [Secs]: -
      SMB Open Group ID: 78a9dc59d45ace211998100059a3c7a0a00f70da0f8ffffcd445b0300000000
Vserver: vs1
  Volume: data2_2
Logical Interface: lif2
  Object Path: /data2/data2_2/test.pptx
  Lock UUID: 302fd7b1-f7bf-47ae-9981-f0dcb6a224f9
  Lock Protocol: cifs
  Lock Type: op-lock
    Node Holding Lock State: node3
    Lock State: granted
    Bytelock Starting Offset: -
    Number of Bytes Locked: -
    Bytelock is Mandatory: -
    Bytelock is Exclusive: -
    Bytelock is Superlock: -
    Bytelock is Soft: -
    Oplock Level: batch
    Shared Lock Access Mode: -
      Shared Lock is Soft: -
      Delegation Type: -
      Client Address: 10.3.1.3
      SMB Open Type: -
      SMB Connect State: connected
      SMB Expiration Time [Secs]: -
      SMB Open Group ID: 78a9dc59d45ace211998100059a3c7a0a00f70da0f8ffffcd445b0300000000
```

**Breaking locks**

When file locks are preventing client access to files, you can display information about currently held locks, and then break specific locks. Examples of scenarios in which you might need to break locks include debugging applications.

**About this task**

The `vserver locks break` command is available only at the advanced privilege level and higher. The man page for the command contains detailed information.

**Steps**

1. To find the information you need to break a lock, use the `vserver locks show` command.
   
   The man page for the command contains detailed information.

2. Set the privilege level to advanced:

   ```
   set -privilege advanced
   ```
3. Perform one of the following actions:

<table>
<thead>
<tr>
<th>If you want to break a lock by specifying...</th>
<th>Enter the command...</th>
</tr>
</thead>
<tbody>
<tr>
<td>The SVM name, volume name, LIF name, and file path</td>
<td>vserver locks break (-vserver) vserver_name (-volume) volume_name (-path) path (-lif) lif</td>
</tr>
</tbody>
</table>

| The lock ID | vserver locks break \(-lockid\) UUID |

4. Return to the admin privilege level:

```
set -privilege admin
```

## How FPolicy first-read and first-write filters work with NFS

NFS clients experience high response time during high traffic of read/write requests when the FPolicy is enabled using an external FPolicy server with read/write operations as monitored events. For NFS clients, the use of first-read and first-write filters in the FPolicy reduces the number of FPolicy notifications and improves performance.

In NFS, the client does I/O on a file by fetching its handle. This handle might remain valid across reboots of the server and the client. Therefore, the client is free to cache the handle and send requests on it without retrieving handles again. In a regular session, lots of reads/write requests are sent to the file server. If notifications are generated for all these requests, it might result in the following issues:

- A larger load due to additional notification processing, and higher response time.
- A large number of notifications being sent to the FPolicy server even though the server unaffected by all of the notifications.

After receiving the first read/write request from a client for a particular file, a cache entry is created and the read/write count is incremented. This request is marked as the first-read/write operation, and an FPolicy event is generated. Before you plan and create your FPolicy filters for an NFS client, you should understand the basics of how FPolicy filters work.

- **First-read**: Filters the client read requests for first-read.
  - When this filter is used for NFS events, the `-file-session-io-grouping-count` and `-file-session-io-grouping-duration` settings determine the first-read request for which FPolicy is processed.

- **First-write**: Filters the client write requests for first-write.
  - When this filter is used for NFS events, the `-file-session-io-grouping-count` and `-file-session-io-grouping-duration` settings determine the first-write request for which FPolicy processed.

The following options are added in NFS servers database.

```
file-session-io-grouping-count: Number of I/O Ops on a File to Be Clubbed and Considered as One Session for Event Generation
file-session-io-grouping-duration: Duration for Which I/O Ops on a File to Be Clubbed and Considered as One Session for Event Generation
```

## Modifying the NFSv4.1 server implementation ID

The NFSv4.1 protocol includes a server implementation ID that documents the server domain, name, and date. You can modify the server implementation ID default values. Changing the default values...
can be useful, for example, when gathering usage statistics or troubleshooting interoperability issues. For more information, see RFC 5661.

**About this task**

The default values for the three options are as follows:

<table>
<thead>
<tr>
<th>Option</th>
<th>Option name</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>NFSv4.1 Implementation ID Domain</td>
<td>-v4.1-implementation-domain</td>
<td>netapp.com</td>
</tr>
<tr>
<td>NFSv4.1 Implementation ID Name</td>
<td>-v4.1-implementation-name</td>
<td>Cluster version name</td>
</tr>
<tr>
<td>NFSv4.1 Implementation ID Date</td>
<td>-v4.1-implementation-date</td>
<td>Cluster version date</td>
</tr>
</tbody>
</table>

**Steps**

1. Set the privilege level to advanced:
   ```
   set -privilege advanced
   ```
2. Perform one of the following actions:
   ```
   If you want to modify the NFSv4.1 implementation ID...
   Enter the command...
   ```
   ```
   Domain
   ```
   ```
   vserver nfs modify -v4.1-implementation-domain domain
   ```
   ```
   Name
   ```
   ```
   vserver nfs modify -v4.1-implementation-name name
   ```
   ```
   Date
   ```
   ```
   vserver nfs modify -v4.1-implementation-date date
   ```
3. Return to the admin privilege level:
   ```
   set -privilege admin
   ```

**Managing NFSv4 ACLs**

You can enable, disable, set, modify, and view NFSv4 access control lists (ACLs).

**Benefits of enabling NFSv4 ACLs**

There are many benefits to enabling NFSv4 ACLs.

The benefits of enabling NFSv4 ACLs include the following:

- Finer-grained control of user access for files and directories
- Better NFS security
- Improved interoperability with CIFS
- Removal of the NFS limitation of 16 groups per user
How NFSv4 ACLs work

A client using NFSv4 ACLs can set and view ACLs on 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 ACL Entries (ACEs) in the ACL that have been tagged with the appropriate inheritance flags.

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, the new object is created only with mode bits.

Enabling or disabling modification of NFSv4 ACLs

When ONTAP receives a chmod command for a file or directory with an ACL, by default the ACL is retained and modified to reflect the mode bit change. You can disable the -v4-acl-preserve parameter to change the behavior if you want the ACL to be dropped instead.

**About this task**

When using unified security style, this parameter also specifies whether NTFS file permissions are preserved or dropped when a client sends a chmod, chgroup, or chown command for a file or directory.

The default for this parameter is enabled.

**Steps**

1. Set the privilege level to advanced:
   
   ```
   set -privilege advanced
   ```

2. Perform one of the following actions:
   
<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Enter the following command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable retention and modification of existing NFSv4 ACLs (default)</td>
<td>vserver nfs modify -vserver vserver_name -v4-acl-preserve enabled</td>
</tr>
<tr>
<td>Disable retention and drop NFSv4 ACLs when changing mode bits</td>
<td>vserver nfs modify -vserver vserver_name -v4-acl-preserve disabled</td>
</tr>
</tbody>
</table>

3. Return to the admin privilege level:
   
   ```
   set -privilege admin
   ```
How ONTAP uses NFSv4 ACLs to determine whether it can delete a file

To determine whether it can delete a file, ONTAP uses a combination of the file's DELETE bit, and the containing directory's DELETE_CHILD bit. For more information, see the NFS 4.1 RFC 5661.

Enabling or disabling NFSv4 ACLs

To enable or disable NFSv4 ACLs, you can modify the -v4.0-acl and -v4.1-acl options. These options are disabled by default.

About this task

The -v4.0-acl or -v4.1-acl option controls the setting and viewing of NFSv4 ACLs; it does not control enforcement of these ACLs for access checking.

NFSv4.0 ACLs are not supported for storage virtual machines (SVMs) with Infinite Volume.

Step

1. Perform one of the following actions:

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Then...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable NFSv4.0 ACLs</td>
<td>Enter the following command:</td>
</tr>
<tr>
<td></td>
<td><code>vserver nfs modify -vserver vserver_name -v4.0-acl enabled</code></td>
</tr>
<tr>
<td>Disable NFSv4.0 ACLs</td>
<td>Enter the following command:</td>
</tr>
<tr>
<td></td>
<td><code>vserver nfs modify -vserver vserver_name -v4.0-acl disabled</code></td>
</tr>
<tr>
<td>Enable NFSv4.1 ACLs</td>
<td>Enter the following command:</td>
</tr>
<tr>
<td></td>
<td><code>vserver nfs modify -vserver vserver_name -v4.1-acl enabled</code></td>
</tr>
<tr>
<td>Disable NFSv4.1 ACLs</td>
<td>Enter the following command:</td>
</tr>
<tr>
<td></td>
<td><code>vserver nfs modify -vserver vserver_name -v4.1-acl disabled</code></td>
</tr>
</tbody>
</table>

Modifying the maximum ACE limit for NFSv4 ACLs

You can modify the maximum number of allowed ACEs for each NFSv4 ACL by modifying the parameter -v4-acl-max-aces. By default, the limit is set to 400 ACEs for each ACL. Increasing this limit can help ensure successful migration of data with ACLs containing over 400 ACEs to storage systems running ONTAP.

About this task

Increasing this limit might impact performance for clients accessing files with NFSv4 ACLs.

Steps

1. Set the privilege level to advanced:
   ```
   set -privilege advanced
   ```

2. Modify the maximum ACE limit for NFSv4 ACLs by entering the following command:
   ```
   vserver nfs modify -v4-acl-max-aces max_ace_limit
   ```
   The valid range of `max_ace_limit` is 192 to 1024.
Managing file access using NFS

3. Return to the admin privilege level:
   
   set -privilege admin

Managing NFSv4 file delegations

You can enable and disable NFSv4 file delegations and retrieve NFSv4 file delegation statistics.

Related concepts

Managing file locks on page 75

Enabling or disabling NFSv4 read file delegations

To enable or disable NFSv4 read file delegations, you can modify the `-v4.0-read-delegation` or `-v4.1-read-delegation` option. By enabling read file delegations, you can eliminate much of the message overhead associated with the opening and closing of files.

About this task

By default, read file delegations are disabled.

The disadvantage of enabling read file delegations is that the server and its clients must recover delegations after the server reboots or restarts, a client reboots or restarts, or a network partition occurs.

NFSv4 file delegations are not supported on storage virtual machines (SVMs) with Infinite Volume.

Step

1. Perform one of the following actions:

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Then...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable NFSv4 read file delegations</td>
<td>Enter the following command: <code>vserver nfs modify -vserver vserver_name -v4.0-read-delegation enabled</code></td>
</tr>
<tr>
<td>Enable NFSv4.1 read file delegations</td>
<td>Enter the following command: <code>vserver nfs modify -vserver vserver_name -v4.1-read-delegation enabled</code></td>
</tr>
<tr>
<td>Disable NFSv4 read file delegations</td>
<td>Enter the following command: <code>vserver nfs modify -vserver vserver_name -v4.0-read-delegation disabled</code></td>
</tr>
<tr>
<td>Disable NFSv4.1 read file delegations</td>
<td>Enter the following command: <code>vserver nfs modify -vserver vserver_name -v4.1-read-delegation disabled</code></td>
</tr>
</tbody>
</table>

Result

The file delegation options take effect as soon as they are changed. There is no need to reboot or restart NFS.

Enabling or disabling NFSv4 write file delegations

To enable or disable write file delegations, you can modify the `-v4.0-write-delegation` or `-v4.1-write-delegation` option. By enabling write file delegations, you can eliminate much of
the message overhead associated with file and record locking in addition to opening and closing of files.

About this task

By default, write file delegations are disabled.

The disadvantage of enabling write file delegations is that the server and its clients must perform additional tasks to recover delegations after the server reboots or restarts, a client reboots or restarts, or a network partition occurs.

NFSv4 file delegations are not supported on storage virtual machines (SVMs) with Infinite Volume.

Step

1. Perform one of the following actions:

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Then...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable NFSv4 write file delegations</td>
<td>Enter the following command:</td>
</tr>
<tr>
<td></td>
<td>vserver nfs modify -vserver vserver_name -v4.0-write-delegation enabled</td>
</tr>
<tr>
<td>Enable NFSv4.1 write file delegations</td>
<td>Enter the following command:</td>
</tr>
<tr>
<td></td>
<td>vserver nfs modify -vserver vserver_name -v4.1-write-delegation enabled</td>
</tr>
<tr>
<td>Disable NFSv4 write file delegations</td>
<td>Enter the following command:</td>
</tr>
<tr>
<td></td>
<td>vserver nfs modify -vserver vserver_name -v4.0-write-delegation disabled</td>
</tr>
<tr>
<td>Disable NFSv4.1 write file delegations</td>
<td>Enter the following command:</td>
</tr>
<tr>
<td></td>
<td>vserver nfs modify -vserver vserver_name -v4.1-write-delegation disabled</td>
</tr>
</tbody>
</table>

Result

The file delegation options take effect as soon as they are changed. There is no need to reboot or restart NFS.

Configuring NFSv4 file and record locking

You can configure NFSv4 file and record locking by specifying the locking lease period and grace period.

Related concepts

Managing file locks on page 75

About NFSv4 file and record locking

For NFSv4 clients, ONTAP supports the NFSv4 file-locking mechanism, maintaining the state of all file locks under a lease-based model.

Specifying the NFSv4 locking lease period

To specify the NFSv4 locking lease period (that is, the time period in which ONTAP irrevocably grants a lock to a client), you can modify the -v4-lease-seconds option. Shorter lease periods
speed up server recovery while longer lease periods are beneficial for servers handling a very large amount of clients.

**About this task**

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.

**Steps**

1. Set the privilege level to advanced:
   ```bash
   set -privilege advanced
   ```

2. Enter the following command:
   ```bash
   vserver nfs modify -vserver vserver_name -v4-lease-seconds number_of_seconds
   ```

3. Return to the admin privilege level:
   ```bash
   set -privilege admin
   ```

**Specifying the NFSv4 locking grace period**

To specify the NFSv4 locking grace period (that is, the time period in which clients attempt to reclaim their locking state from ONTAP during server recovery), you can modify the `-v4-grace-seconds` option.

**About this task**

By default, this option is set to 45.

**Steps**

1. Set the privilege level to advanced:
   ```bash
   set -privilege advanced
   ```

2. Enter the following command:
   ```bash
   vserver nfs modify -vserver vserver_name -v4-grace-seconds number_of_seconds
   ```

3. Return to the admin privilege level:
   ```bash
   set -privilege admin
   ```

**How NFSv4 referrals work**

When you enable NFSv4 referrals, ONTAP provides “intra-SVM” referrals to NFSv4 clients. Intra-SVM referral is when a cluster node receiving the NFSv4 request refers the NFSv4 client to another logical interface (LIF) on the storage virtual machine (SVM).

The NFSv4 client should access the path that received the referral at the target LIF from that point onward. The original cluster node provides such a referral when it determines that there exists a LIF in the SVM that is resident on the cluster node on which the data volume resides, thereby enabling the clients faster access to the data and avoiding extra cluster communication.
Enabling or disabling NFSv4 referrals

You can enable NFSv4 referrals on storage virtual machines (SVMs) by enabling the options `v4-fsid-change` and `v4.0-referrals` or `v4.1-referrals`. Enabling NFSv4 referrals can result in faster data access for NFSv4 clients that support this feature.

**Before you begin**

If you want to enable NFS referrals, you must first disable parallel NFS. You cannot enable both at the same time.

**Steps**

1. Set the privilege level to advanced:
   ```
   set -privilege advanced
   ```

2. Perform one of the following actions:

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Enter the command...</th>
</tr>
</thead>
</table>
   | Enable NFSv4 referrals | `vserver nfs modify -vserver vserver_name -v4-fsid-change enabled`  
                           | `vserver nfs modify -vserver vserver_name -v4.0-referrals enabled` |
   | Disable NFSv4 referrals | `vserver nfs modify -vserver vserver_name -v4.0-referrals disabled` |
   | Enable NFSv4.1 referrals | `vserver nfs modify -vserver vserver_name -v4-fsid-change enabled`  
                            | `vserver nfs modify -vserver vserver_name -v4.1-referrals enabled` |
   | Disable NFSv4.1 referrals | `vserver nfs modify -vserver vserver_name -v4.1-referrals disabled` |

3. Return to the admin privilege level:
   ```
   set -privilege admin
   ```

**Related tasks**

*Enabling or disabling pNFS* on page 55

Displaying NFS statistics

You can display NFS statistics for storage virtual machines (SVMs) on the storage system to monitor performance and diagnose issues.

**Steps**

1. Use the `statistics catalog object show` command to identify the NFS objects from which you can view data.

**Example**

```
statistics catalog object show -object nfs*
```
For more information about the `statistics` commands, see the *Performance Monitoring Power Guide*.

2. Use the `statistics start` and optional `statistics stop` commands to collect a data sample from one or more objects.

3. Use the `statistics show` command to view the sample data.

```plaintext
Example: Monitoring NFSv3 performance
The following example shows performance data for the NFSv3 protocol.
The following command starts data collection for a new sample:

```bash
vsl::> statistics start -object nfsv3 -sample-id nfs_sample
```

The following command shows data from the sample by specifying counters that show the number of successful read and write requests versus the total number of read and write requests:

```bash
vsl::> statistics show -sample-id nfs_sample -counter read_total|write_total|read_success|write_success
```

Object: nfsv3
Instance: vsl
Start-time: 2/11/2013 15:38:29
End-time: 2/11/2013 15:38:41
Cluster: cluster1

<table>
<thead>
<tr>
<th>Counter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>read_success</td>
<td>40042</td>
</tr>
<tr>
<td>read_total</td>
<td>40042</td>
</tr>
<tr>
<td>write_success</td>
<td>1492052</td>
</tr>
<tr>
<td>write_total</td>
<td>1492052</td>
</tr>
</tbody>
</table>

Related information

*Performance monitoring express setup*

**Displaying DNS statistics**

You can display DNS statistics for storage virtual machines (SVMs) on the storage system to monitor performance and diagnose issues.

**Steps**

1. Use the `statistics catalog object show` command to identify the DNS objects from which you can view data.

   ```bash
   statistics catalog object show -object external_service_op*
   ```

   For more information about the `statistics` commands, see the *Performance Monitoring Power Guide*.

2. Use the `statistics start` and `statistics stop` commands to collect a data sample from one or more objects.
3. Use the `statistics show` command to view the sample data.

### Monitoring DNS statistics

The following examples show performance data for DNS queries. The following commands start data collection for a new sample:

```
vsl::*> statistics start -object external_service_op -sample-id dns_sample1
vsl::*> statistics start -object external_service_op_error -sample-id dns_sample2
```

The following command displays data from the sample by specifying counters that display the number of DNS queries sent versus the number of DNS queries received, failed, or timed out:

```
vsl::*> statistics show -sample-id dns_sample1 -counter num_requests_sent|num_responses_received|num_successful_responses|num_timeouts|num_request_failures|num_not_found_responses
```

Object: external_service_op
Start-time: 3/8/2016 11:15:21
End-time: 3/8/2016 11:16:52
Elapsed-time: 91s
Scope: vs1

<table>
<thead>
<tr>
<th>Counter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>num_not_found_responses</td>
<td>0</td>
</tr>
<tr>
<td>num_request_failures</td>
<td>0</td>
</tr>
<tr>
<td>num_requests_sent</td>
<td>1</td>
</tr>
<tr>
<td>num_responses_received</td>
<td>1</td>
</tr>
<tr>
<td>num_successful_responses</td>
<td>1</td>
</tr>
<tr>
<td>num_timeouts</td>
<td>0</td>
</tr>
</tbody>
</table>

6 entries were displayed.

The following command displays data from the sample by specifying counters that display the number of times a specific error was received for a DNS query on the particular server:

```
vsl::*> statistics show -sample-id dns_sample2 -counter server_ip_address|error_string|count
```

Object: external_service_op_error
End-time: 3/8/2016 11:24:25
Elapsed-time: 64s
Scope: vs1

<table>
<thead>
<tr>
<th>Counter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>count</td>
<td>1</td>
</tr>
<tr>
<td>error_string</td>
<td>NXDOMAIN</td>
</tr>
<tr>
<td>server_ip_address</td>
<td>10.72.219.109</td>
</tr>
</tbody>
</table>

3 entries were displayed.

### Related information

*Performance monitoring express setup*
Displaying NIS statistics

You can display NIS statistics for storage virtual machines (SVMs) on the storage system to monitor performance and diagnose issues.

Steps

1. Use the `statistics catalog object show` command to identify the NIS objects from which you can view data.

   **Example**

   ```
   statistics catalog object show -object external_service_op*
   ```

   For more information about the statistics commands, see the Performance Monitoring Power Guide.

2. Use the `statistics start` and `statistics stop` commands to collect a data sample from one or more objects.

3. Use the `statistics show` command to view the sample data.

---

### Monitoring NIS statistics

The following examples display performance data for NIS queries. The following commands start data collection for a new sample:

````
vs1::*> statistics start -object external_service_op -sample-id nis_sample1
vs1::*> statistics start -object external_service_op_error -sample-id nis_sample2
```

The following command displays data from the sample by specifying counters that show the number of NIS queries sent versus the number of NIS queries received, failed, or timed out:

````
vs1::*> statistics show -sample-id nis_sample1 -counter instance|num_requests_sent|num_responses_received|num_successful_responses|num_timeouts|num_request_failures|num_not_found_responses
Object: external_service_op
End-time: 3/8/2016 11:27:56
Elapsed-time: 17s
Scope: vs1
```

<table>
<thead>
<tr>
<th>Counter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>num_not_found_responses</td>
<td>0</td>
</tr>
<tr>
<td>num_request_failures</td>
<td>1</td>
</tr>
<tr>
<td>num_requests_sent</td>
<td>2</td>
</tr>
<tr>
<td>num_responses_received</td>
<td>1</td>
</tr>
<tr>
<td>num_successful_responses</td>
<td>1</td>
</tr>
<tr>
<td>num_timeouts</td>
<td>0</td>
</tr>
</tbody>
</table>

6 entries were displayed.

The following command displays data from the sample by specifying counters that show the number of times a specific error was received for a NIS query on the particular server:

````
vs1::*> statistics show -sample-id nis_sample2 -counter server_ip_address|error_string|count
Object: external_service_op_error
```

---
Support for VMware vStorage over NFS

ONTAP supports certain VMware vStorage APIs for Array Integration (VAAI) features in an NFS environment.

Supported features

The following features are supported:

- **Copy offload**
  Enables an ESXi host to copy virtual machines or virtual machine disks (VMDKs) directly between the source and destination data store location without involving the host. This conserves ESXi host CPU cycles and network bandwidth. Copy offload preserves space efficiency if the source volume is sparse.

- **Space reservation**
  Guarantees storage space for a VMDK file by reserving space for it.

Limitations

VMware vStorage over NFS has the following limitations:

- **vStorage is not supported on storage virtual machines (SVMs) with Infinite Volumes.**
- **Copy offload operations can fail in the following scenarios:**
  - While running wafliron on the source or destination volume because it temporarily takes the volume offline
  - While moving either the source or destination volume
  - While moving either the source or destination LIF
  - While performing takeover or giveback operations
  - While performing switchover or switchback operations
- **Server-side copy can fail due to file handle format differences in the following scenario:**
  You attempt to copy data from SVMs that have currently or had previously exported qtrees to SVMs that have never had exported qtrees. To work around this limitation, you can export at least one qtree on the destination SVM.

Related information

*NetApp Knowledgebase Answer 1001100: What VAAI offloaded operations are supported by Data ONTAP?*
Enabling or disabling VMware vStorage over NFS

You can enable or disable support for VMware vStorage over NFS on storage virtual machines (SVMs) by using the `vserver nfs modify` command.

About this task

By default, support for VMware vStorage over NFS is disabled.

Steps

1. Display the current vStorage support status for SVMs by entering the following command:
   
   ```bash
   vserver nfs show -vserver vserver_name -instance
   ```

2. Perform one of the following actions:

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Enter the following command...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable VMware vStorage support</td>
<td><code>vserver nfs modify -vserver vserver_name -vstorage enabled</code></td>
</tr>
<tr>
<td>Disable VMware vStorage support</td>
<td><code>vserver nfs modify -vserver vserver_name -vstorage disabled</code></td>
</tr>
</tbody>
</table>

After you finish

You must install the NFS Plug-in for VMware VAAI before you can use this functionality. For more information, see *Installing the NetApp NFS Plug-in for VMware VAAI*.

Related information

*NetApp Documentation: NetApp NFS Plug-in for VMware VAAI*

Enabling or disabling rquota support

ONTAP supports the remote quota protocol version 1 (rquota v1). The rquota protocol enables NFS clients to obtain quota information for users and groups from a remote machine. You can enable rquota on storage virtual machines (SVMs) by using the `vserver nfs modify` command.

About this task

By default, rquota is disabled.

Step

1. Perform one of the following actions:

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Enter the following command...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable rquota support for SVMs</td>
<td><code>vserver nfs modify -vserver vserver_name -rquota enable</code></td>
</tr>
<tr>
<td>Disable rquota support for SVMs</td>
<td><code>vserver nfs modify -vserver vserver_name -rquota disable</code></td>
</tr>
</tbody>
</table>

For more information about quotas, see the *Logical Storage Management Guide*. 
NFSv3 and NFSv4 performance improvement by modifying the TCP transfer size

You can improve the performance of NFSv3 and NFSv4 clients connecting to storage systems over a high-latency network by modifying the TCP maximum transfer size.

When clients access storage systems over a high-latency network, such as a wide area network (WAN) or metro area network (MAN) with a latency over 10 milliseconds, you might be able to improve the connection performance by modifying the TCP maximum transfer size. Clients accessing storage systems in a low-latency network, such as a local area network (LAN), can expect little to no benefit from modifying these parameters. If the throughput improvement does not outweigh the latency impact, you should not use these parameters.

To determine whether your storage environment would benefit from modifying these parameters, you should first conduct a comprehensive performance evaluation of a poorly performing NFS client. Review whether the low performance is because of excessive round trip latency and small request on the client. Under these conditions, the client and server cannot fully use the available bandwidth because they spend the majority of their duty cycles waiting for small requests and responses to be transmitted over the connection.

By increasing the NFSv3 and NFSv4 request size, the client and server can use the available bandwidth more effectively to move more data per unit time; therefore, increasing the overall efficiency of the connection.

Keep in mind that the configuration between the storage system and the client might vary. The storage system and the client supports maximum size of 1 MB for transfer operations. However, if you configure the storage system to support 1 MB maximum transfer size but the client only supports 64 KB, then the mount transfer size is limited to 64 KB or less.

Before modifying these parameters, you must be aware that it results in additional memory consumption on the storage system for the period of time necessary to assemble and transmit a large response. The more high-latency connections to the storage system, the higher the additional memory consumption. Storage systems with high memory capacity might experience very little effect from this change. Storage systems with low memory capacity might experience noticeable performance degradation.

The successful use of these parameter relies on the ability to retrieve data from multiple nodes of a cluster. The inherent latency of the cluster network might increase the overall latency of the response. Overall latency tends to increase when using these parameters. As a result, latency sensitive workloads might show negative impact.

Modifying the NFSv3 and NFSv4 TCP maximum transfer size

You can modify the `-tcp-max-xfer-size` option to configure maximum transfer sizes for all TCP connections using the NFSv3 and NFSv4.x protocols.

About this task

You can modify these options individually for each storage virtual machine (SVM).

Beginning in ONTAP 9, the `v3-tcp-max-read-size` and `v3-tcp-max-write-size` options are obsolete. You must use the `-tcp-max-xfer-size` option instead.
Steps

1. Set the privilege level to advanced:
   ```bash
   set -privilege advanced
   ```

2. Perform one of the following actions:

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Enter the command...</th>
</tr>
</thead>
</table>
   | Modify the NFSv3 or NFSv4 TCP maximum transfer size | `vserver nfs modify -vserver vserver_name -tcp-max-xfer-size integer_max_xfer_size`

   **Option** | **Range** | **Default**
   --- | --- | ---
   `-tcp-max-xfer-size` | 8192 to 1048576 bytes | 65536 bytes

   **Note:** The maximum transfer size that you enter must be a multiple of 4 KB (4096 bytes). Requests that are not properly aligned negatively affect performance.

3. Use the `vserver nfs show -fields tcp-max-xfer-size` command to verify the changes.

4. If any clients use static mounts, unmount and remount for the new parameter size to take effect.

**Example**

The following command sets the NFSv3 and NFSv4.x TCP maximum transfer size to 1048576 bytes on the SVM named vs1:

```bash
vs1::> vserver nfs modify -vserver vs1 -tcp-max-xfer-size 1048576
```

Configuring the number of group IDs allowed for NFS users

By default, ONTAP supports up to 32 group IDs when handling NFS user credentials using Kerberos (RPCSEC_GSS) authentication. When using AUTH_SYS authentication, the default maximum number of group IDs is 16, as defined in RFC 5531. You can increase the maximum up to 1,024 if you have users who are members of more than the default number of groups.

**About this task**

If a user has more than the default number of group IDs in their credentials, the remaining group IDs are truncated and the user might receive errors when attempting to access files from the storage system. You should set the maximum number of groups, per SVM, to a number that represents the maximum groups in your environment.

The following table shows the two parameters of the `vserver nfs modify` command that determine the maximum number of group IDs in three sample configurations:

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Settings</th>
<th>Resulting group IDs limit</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-extended-groups-limit</code></td>
<td><code>32</code> <code>disabled</code></td>
<td>RPCSEC_GSS: 32</td>
</tr>
<tr>
<td><code>-auth-sys-extended-groups</code></td>
<td>These are the default settings.</td>
<td></td>
</tr>
</tbody>
</table>
### Parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Settings</th>
<th>Resulting group IDs limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>-extended-groups-limit</td>
<td>256</td>
<td>RPCSEC_GSS: 256 AUTH_SYS: 16</td>
</tr>
<tr>
<td>-auth-sys-extended-groups</td>
<td>disabled</td>
<td></td>
</tr>
<tr>
<td>-extended-groups-limit</td>
<td>512</td>
<td>RPCSEC_GSS: 512 AUTH_SYS: 512</td>
</tr>
<tr>
<td>-auth-sys-extended-groups</td>
<td>enabled</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** Some older NFS clients might not be compatible with AUTH_SYS extended groups.

### Steps

1. Set the privilege level to advanced:
   ```
   set -privilege advanced
   ```

2. Perform the desired action:

<table>
<thead>
<tr>
<th>If you want to set the maximum number of allowed auxiliary groups...</th>
<th>Enter the command...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Only for RPCSEC_GSS and leave AUTH_SYS set to the default value of 16</td>
<td>vserver nfs modify -vserv vserver_name -extended-groups-limit (32-1024) -auth-sys-extended-groups disabled</td>
</tr>
<tr>
<td>For both RPCSEC_GSS and AUTH_SYS</td>
<td>vserver nfs modify -vserv vserver_name -extended-groups-limit (32-1024) -auth-sys-extended-groups enabled</td>
</tr>
</tbody>
</table>

3. Verify the `-extended-groups-limit` value and verify whether AUTH_SYS is using extended groups:
   ```
   vserver nfs show -vserv vserver_name -fields auth-sys-extended-groups,extended-groups-limit
   ```

4. Return to the admin privilege level:
   ```
   set -privilege admin
   ```

### Example

The following example enables extended groups for AUTH_SYS authentication and sets the maximum number of extended groups to 512 for both AUTH_SYS and RPCSEC_GSS authentication. These changes are made only for clients who access the SVM named vs1:

```bash
vs1::> set -privilege 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

vs1::*> vserver nfs modify -vserv vsl -auth-sys-extended-groups enabled -extended-groups-limit 512

vs1::*> vserver nfs show -vserv vsl -fields auth-sys-extended-groups,extended-groups-limit

vserver auth-sys-extended-groups extended-groups-limit
-------- ---------------------
vs1 enabled 512

vs1::*> set -privilege admin
```
Controlling root user access to NTFS security-style data

You can configure ONTAP to allow NFS clients access to NTFS security-style data and NTFS clients to access NFS security-style data. When using NTFS security style on an NFS data store, you must decide how to treat access by the root user and configure the storage virtual machine (SVM) accordingly.

About this task

When a root user accesses NTFS security-style data, you have two options:

- Map the root user to a Windows user like any other NFS user and manage access according to NTFS ACLs.
- Ignore NTFS ACLs and provide full access to root.

Steps

1. Set the privilege level to advanced:
   ```
   set -privilege advanced
   ```

2. Perform the desired action:
   ```
   If you want the root user to... Enter the command...
   Be mapped to a Windows user  vserver nfs modify -vserver vserver_name -ignore-nt-acl-for-root disabled
   Bypass the NT ACL check  vserver nfs modify -vserver vserver_name -ignore-nt-acl-for-root enabled
   ```

   By default, this parameter is disabled.

   If this parameter is enabled but there is no name mapping for the root user, ONTAP uses a default SMB administrator credential for auditing.

3. Return to the admin privilege level:
   ```
   set -privilege admin
   ```
Appendix: Supported NFS versions and clients

Before you can use NFS in your network, you need to know which NFS versions and clients ONTAP supports.

ONTAP supports the following major and minor NFS protocol versions:

- NFSv3
- NFSv4.0
- NFSv4.1
- pNFS

For the latest information about which NFS clients ONTAP supports, see the Interoperability Matrix.

NetApp Interoperability Matrix Tool

NFSv4.0 functionality supported by ONTAP

ONTAP supports all the mandatory functionality in NFSv4.0 except the SPKM3 and LIPKEY security mechanisms.

The following NFSv4 functionality is supported:

**COMPOUND**
- Allows a client to request multiple file operations in a single remote procedure call (RPC) request.

**File delegation**
- Allows the server to delegate file control to some types of clients for read and write access.

**Pseudo-fs**
- Used by NFSv4 servers to determine mount points on the storage system. There is no mount protocol in NFSv4.

**Locking**
- Lease-based. There are no separate Network Lock Manager (NLM) or Network Status Monitor (NSM) protocols in NFSv4.

For more information about the NFSv4.0 protocol, see RFC 3530.

Related concepts
- Managing NFSv4 file delegations on page 83
- Managing file locks on page 75

Related tasks
- Configuring NFSv4 file and record locking on page 84

Limitations of ONTAP support for NFSv4

You should be aware of several limitations of ONTAP support for NFSv4.

- The delegation feature is not supported by every client type.
• In ONTAP 9.4 and earlier releases, names with non-ASCII characters on volumes other than UTF8 volumes are rejected by the storage system.
   In ONTAP 9.5 and later releases, volumes created with the utf8mb4 language setting and mounted using NFS v4 are no longer subject to this restriction.

• All file handles are persistent; the server does not give volatile file handles.

• Migration and replication are not supported.

• NFSv4 clients are not supported with read-only load-sharing mirrors.
   ONTAP routes NFSv4 clients to the source of the load-sharing mirror for direct read and write access.

• Named attributes are not supported.

• All recommended attributes are supported, except for the following:
  ◦ archive
  ◦ hidden
  ◦ homogeneous
  ◦ mimetype
  ◦ quota_avail_hard
  ◦ quota_avail_soft
  ◦ quota_used
  ◦ system
  ◦ time_backup

  Note: Although it does not support the quota* attributes, ONTAP does support user and group quotas through the RQUOTA side band protocol.

ONTAP support for NFSv4.1
ONTAP supports the NFSv4.1 protocol to allow access for NFSv4.1 clients.

By default NFSv4.1 is disabled. You can enable it by specifying the \-v4.1 option and setting it to enabled when creating an NFS server on the storage virtual machine (SVM).

ONTAP does not support NFSv4.1 directory and file level delegations.

Related tasks
Enabling or disabling NFSv4.1 on page 54

ONTAP support for parallel NFS
ONTAP supports parallel NFS (pNFS). The pNFS protocol offers performance improvements by giving clients direct access to the data of a set of files distributed across multiple nodes of a cluster. It helps clients locate the optimal path to a volume.

Related tasks
Enabling or disabling pNFS on page 55
Use of hard mounts

When troubleshooting mounting problems, you need to be sure that you are using the correct mount type. NFS supports two mount types: soft mounts and hard mounts. You should use only hard mounts for reliability reasons.

You should not use soft mounts, especially when there is a possibility of frequent NFS timeouts. Race conditions can occur as a result of these timeouts, which can lead to data corruption.
Appendix: NFS and SMB file and directory naming dependencies

File and directory naming conventions depend on both the network clients’ operating systems and the file-sharing protocols, in addition to language settings on the ONTAP cluster and clients.

The operating system and the file-sharing protocols determine the following:

- Characters a file name can use
- Case-sensitivity of a file name

ONTAP supports multi-byte characters in file, directory, and qtree names, depending on the ONTAP release.

Characters a file or directory name can use

If you are accessing a file or directory from clients with different operating systems, you should use characters that are valid in both operating systems.

For example, if you use UNIX to create a file or directory, do not use a colon (:) in the name because the colon is not allowed in MS-DOS file or directory names. Because restrictions on valid characters vary from one operating system to another, see the documentation for your client operating system for more information about prohibited characters.

Case-sensitivity of file and directory names in a multiprotocol environment

File and directory names are case-sensitive for NFS clients and case-insensitive but case-preserving for CIFS clients. You must understand what the implications are in a multiprotocol environment and the actions you might need to take when specifying the path while creating SMB shares and when accessing data within the shares.

If an SMB client creates a directory named testdir, both SMB and NFS clients display the file name as testdir. However, if an SMB user later tries to create a directory name TESTDIR, the name is not allowed because, to the SMB client, that name currently exists. If an NFS user later creates a directory named TESTDIR, NFS and SMB clients display the directory name differently, as follows:

On NFS clients, you see both directory names as they were created, for example testdir and TESTDIR, because directory names are case-sensitive.

SMB clients use the 8.3 names to distinguish between the two directories. One directory has the base file name. Additional directories are assigned an 8.3 file name.

- On SMB clients, you see testdir and TESTDI~1.
- ONTAP creates the TESTDI~1 directory name to differentiate the two directories.
  In this case, you must use the 8.3 name when specifying a share path while creating or modifying a share on a storage virtual machine (SVM).

Similarly for files, if an SMB client creates test.txt, both SMB and NFS clients display the file name as test.txt. However, if an SMB user later tries to create Test.txt, the name is not allowed because, to the SMB client, that name currently exists. If an NFS user later creates a file named Test.txt, NFS and SMB clients display the file name differently, as follows:
On NFS clients, you see both file names as they were created, test.txt and Text.txt, because file names are case-sensitive.

SMB clients use the 8.3 names to distinguish between the two files. One file has the base file name. Additional files are assigned an 8.3 file name.

• On SMB clients, you see test.txt and TEST~1.TXT.
• ONTAP creates the TEST~1.TXT file name to differentiate the two files.

How ONTAP creates file and directory names

ONTAP creates and maintains two names for files or directories in any directory that has access from an SMB client: the original long name and a name in 8.3 format.

For file or directory names that exceed the eight character name or the three character extension limit (for files), ONTAP generates an 8.3-format name as follows:

• It truncates the original file or directory name to six characters, if the name exceeds six characters.
• It appends a tilde (~) and a number, one through five, to file or directory names that are no longer unique after being truncated.
  If it runs out of numbers because there are more than five similar names, it creates a unique name that bears no relation to the original name.
• In the case of files, it truncates the file name extension to three characters.

For example, if an NFS client creates a file named specifications.html, the 8.3 format file name created by ONTAP is specif~1.htm. If this name already exists, ONTAP uses a different number at the end of the file name. For example, if an NFS client then creates another file named specifications_new.html, the 8.3 format of specifications_new.html is specif~2.htm.

How ONTAP handles multi-byte file, directory, and qtree names

Beginning with ONTAP 9.5, support for 4-byte UTF-8 encoded names enables the creation and display of UTF-16 file names with supplementary characters, also known as surrogate pairs, which was not available in earlier releases. 4-byte UTF-8 encoded characters can be displayed in the names of files, directories, and qtrees.

To enable support for 4-byte UTF-8 encoded names, a new utf8mb4 language code is available for the vserver and volume command families.

• You must create a new volume in one of the following ways:
  ◦ Setting the volume -language option explicitly:
    ```
    volume create -language utf8mb4 {...}
    ```
  ◦ Inheriting the volume -language option from an SVM that has been created with or modified for the option: vserver [create|modify] -language utf8mb4 {...}
    ```
    volume create {...}
    ```
• You cannot modify existing volumes for utf8mb4 support; you must create a new utf8mb4-ready volume, and then migrate the data using client-based copy tools.
You can update SVMs for utf8mb4 support, but existing volumes retain their original language codes.
Note: LUN names with 4-byte UTF-8 characters are not currently supported.

- Unicode character data is typically represented in Windows applications using the 16-bit Unicode Transformation Format (UTF-16).
  - Characters in the basic multilingual plane (BMP) of UTF-16 are represented as single 16-bit code points (comprising 3 bytes).
  - Characters in the additional 16 supplementary planes are represented as pairs of 16-bit code points that are referred to as surrogate pairs.

- When you create file names on systems running ONTAP 9.4 or earlier that contain valid or invalid supplementary characters, ONTAP rejects the file name and returns an invalid file name error. To avoid this issue, use only BMP characters in file names and avoid using supplementary characters, or upgrade to ONTAP 9.5 or later.

Beginning with ONTAP 9, Unicode characters are allowed in qtree names.

- You can use either the volume qtree command family or System Manager to set or modify qtree names.
- qtree names can include multi-byte characters in Unicode format, such as Japanese and Chinese characters.
- In releases before ONTAP 9.5, only BMP characters (that is, those that could be represented in 3 bytes) were supported.
  
  Note: In releases before ONTAP 9.5, the junction-path of the qtree's parent volume can contain qtree and directory names with Unicode characters. The volume show command displays these names correctly when the parent volume has a UTF-8 language setting. However, if the parent volume language is not one of the UTF-8 language settings, some parts of the junction-path are displayed using a numeric NFS alternate name.

- In 9.5 and later releases, 4-byte characters are supported in qtree names, provided that the qtree is in a volume enabled for utf8mb4.

### Configuring character mapping for SMB file name translation on volumes

NFS clients can create file names that contain characters that are not valid for SMB clients and certain Windows applications. You can configure character mapping for file name translation on volumes to allow SMB clients to access files with NFS names that would otherwise not be valid.

**About this task**

When files created by NFS clients are accessed by SMB clients, ONTAP looks at the name of the file. If the name is not a valid SMB file name (for example, if it has an embedded colon “:” character), ONTAP returns the 8.3 file name that is maintained for each file. However, this causes problems for applications that encode important information into long file names.

Therefore, if you are sharing a file between clients on different operating systems, you should use characters in the file names that are valid in both operating systems.

However, if you have NFS clients that create file names containing characters that are not valid file names for SMB clients, you can define a map that converts the invalid NFS characters into Unicode characters that both SMB and certain Windows applications accept. For example, this functionality supports the CATIA MCAD and Mathematica applications as well as other applications that have this requirement.

You can configure character mapping on a volume-by-volume basis.
You must keep the following in mind when configuring character mapping on a volume:

- **Character mapping is not applied across junction points.**
  You must explicitly configure character mapping for each junction volume.

- **You must make sure that the Unicode characters that are used to represent invalid or illegal characters are characters that do not normally appear in file names; otherwise, unwanted mappings occur.**
  For example, if you try to map a colon (:) to a hyphen (-) but the hyphen (-) was used in the file name correctly, a Windows client trying to access a file named “a-b” would have its request mapped to the NFS name of “a:b” (not the desired outcome).

- **After applying character mapping, if the mapping still contains an invalid Windows character, ONTAP falls back to Windows 8.3 file names.**

- **In FPolicy notifications, NAS audit logs, and security trace messages, the mapped file names are shown.**

- **When a SnapMirror relation of type DP is created, the source volume's character mapping is not replicated on the destination DP volume.**

**Step**

1. **Configure character mapping:**

   ```
   vserver cifs character-mapping create -vserver vserver_name -volume volume_name -mapping mapping_text, ...
   ```

   The mapping consists of a list of source-target character pairs separated by “:”. The characters are Unicode characters entered using hexadecimal digits. For example: 3C:E03C.

   The first value of each `mapping_text` pair that is separated by a colon is the hexadecimal value of the NFS character you want to translate, and the second value is the Unicode value that SMB uses. The mapping pairs must be unique (a one-to-one mapping should exist).

   - **Source mapping**
     The following table shows the permissible Unicode character set for source mapping:

     | Unicode character | Printed character | Description       |
     |-------------------|-------------------|-------------------|
     | 0x01-0x19         | Not applicable    | Non-printing control characters |
     | 0x5C              | \                | Backslash         |
     | 0x3A              | :                | Colon             |
     | 0x2A              | *                | Asterisk          |
     | 0x3F              | ?                | Question mark     |
     | 0x22              | "                | Quotation mark    |
     | 0x3C              | <                | Less than         |
     | 0x3E              | >                | Greater than      |
     | 0x7C              | |                | Vertical line     |
     | 0xB1              | ±                | Plus-minus sign   |

   - **Target mapping**
     You can specify target characters in the “Private Use Area” of Unicode in the following range: U+E0000...U+F8FF.
Example

The following command creates a character mapping for a volume named “data” on storage virtual machine (SVM) vs1:

```
cluster1::> vserver cifs character-mapping create -volume data -mapping 3c:e17c,3e:f17d,2a:f745
cluster1::> vserver cifs character-mapping show
```

Vserver | Volume Name | Character Mapping
--------|-------------|-------------------
vs1     | data        | 3c:e17c, 3e:f17d, 2a:f745

Commands for managing character mappings for SMB file name translation

You can manage character mapping by creating, modifying, displaying information about, or deleting file character mappings used for SMB file name translation on FlexVol volumes.

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Use this command...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create new file character mappings</td>
<td>vserver cifs character-mapping create</td>
</tr>
<tr>
<td>Display information about file character mappings</td>
<td>vserver cifs character-mapping show</td>
</tr>
<tr>
<td>Modify existing file character mappings</td>
<td>vserver cifs character-mapping modify</td>
</tr>
<tr>
<td>Delete file character mappings</td>
<td>vserver cifs character-mapping delete</td>
</tr>
</tbody>
</table>

For more information, see the man page for each command.
Copyright

Copyright © 2019 NetApp, Inc. All rights reserved. Printed in the U.S.

No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

Data contained herein pertains to a commercial item (as defined in FAR 2.101) and is proprietary to NetApp, Inc. The U.S. Government has a non-exclusive, non-transferrable, non-sublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b).
Trademark

NETAPP, the NETAPP logo, and the marks listed on the NetApp Trademarks page are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.

How to send comments about documentation and receive update notifications

You can help us to improve the quality of our documentation by sending us your feedback. You can receive automatic notification when production-level (GA/FCS) documentation is initially released or important changes are made to existing production-level documents.

If you have suggestions for improving this document, send us your comments by email.

doccomments@netapp.com

To help us direct your comments to the correct division, include in the subject line the product name, version, and operating system.

If you want to be notified automatically when production-level documentation is released or important changes are made to existing production-level documents, follow Twitter account @NetAppDoc.

You can also contact us in the following ways:

• NetApp, Inc., 1395 Crossman Ave., Sunnyvale, CA 94089 U.S.
• Telephone: +1 (408) 822-6000
• Fax: +1 (408) 822-4501
• Support telephone: +1 (888) 463-8277
Index

4-byte UTF-8 encoded characters
   how ONTAP handles multi-byte file, directory, and qtree names 100
8.3-format
   how ONTAP uses when creating file and directory names 100

A
   about this guide  
     deciding whether to use the NFS Reference 7
   access  
     how security types determine levels of client 28
     access cache  
       explained 32
       how timeout values work 33
       optimizing performance 75
     access control lists  
       See ACLs
     access levels  
       how security types determine client 28
     access requests  
       mapping to anonymous 26
     access, file  
       tasks for setting up for NFS 24
     accessing  
       handling unknown UNIX users for NTFS volumes or qtrees 56
     ACEs  
       limit for NFSv4 ACLs 82
   ACLs  
     enabling or disabling modification of NFSv4 81
     enabling or disabling NFSv4 82
     limit of ACEs for NFSv4 82
     NFSv4, benefits of enabling 80
     NFSv4, how they work 81
     NFSv4, introduction to managing 80
   anonymous  
     mapping clients to 26
   anonymous access  
     how to configure with export rules 30
   APIs  
     supported VMware vStorage, for NFS 90
   architectures  
     typical NAS namespace 9
   authentication  
     how ONTAP handles NFS client 12
     how ONTAP uses name services for user 13
     requirements for using Kerberos with NFS 34
     support for Kerberos in an ONTAP environment 34
   authentication-based restrictions 12

B
   bits  
     how ONTAP treats read-only 76
   breaking

C
   cache  
     configuring time-to-live for NFS user credentials 71
     reasons for modifying time-to-live for NFS credentials 70
   cache, access  
     explained 32
     optimizing performance 75
   cache, name service  
     commands for managing 64
   cache, NFS credential  
     how it works 13
   caches  
     displaying export policy netgroup 73
     flushing export policy 72
     how ONTAP uses export policy 31
     introduction to configuring NFS credential 70
   case-sensitivity  
     of file and directory names in a multiprotocol environment 99
   character mapping  
     commands for creating, modifying, deleting, or displaying information about SMB file name translations 103
   character mappings  
     enabling for SMB file name translation on FlexVol volumes 101
   checking  
     client membership of netgroups 74
   CIFS  
     how ONTAP grants file access from NFS clients 13
   CIFS share  
     concepts 8
   client access  
     how security types determine levels of 28
     performing stricter checking for netgroups by verifying domains 58
   client authentication  
     how ONTAP handles 12
   client schema templates  
     commands for managing LDAP 68
   clients  
     validating qtree IDs for file operations 34
     commands  
       for managing LDAP client schema templates 68
       for managing local UNIX users 65
       for managing name mappings 51, 64
     commands for managing name-service cache 64
   comments  
     how to send feedback about documentation 106
   components  
     differences between ONTAP and Windows handling of locks on share path 77
   configurations  
     commands for managing NFS Kerberos realm 69
configuring
  default users 50
  number of group IDs allowed for NFS users 93
  security style on FlexVol volumes 22
  security style on qtrees 22
  security style on SVM root volumes 21
conversion rules
  name mapping 49
copying
  export policies 70
  LDAP client schema templates, command for 68
creating
  export rules, command for 70
  LDAP client configurations, command for 68
  LDAP configurations, command for 68
  local UNIX groups, command for 65
  local UNIX users, command for 65
  name service switch entries 63
  NFS Kerberos realm configurations, command for 69
  NIS domain configurations, command for 67
credential cache, NFS
  how it works 13
credential caches
  introduction to configuring NFS 70

D
data access
  introduction to how security styles affect 19
data volumes
  creating with specified junction points for NAS access 15
default users
  configuring 50
deleting
  export policies 70
  export rules, command for 70
  LDAP client configurations, command for 68
  LDAP client schema templates, command for 68
  LDAP configurations, command for 68
  local UNIX groups, command for 65
  local UNIX users, command for 65
  name service switch entries 63
  netgroups, command for 67
  NFS Kerberos realm configurations, command for 69
  NFS servers 60
  NIS domain configurations, command for 67
directory names
  case-sensitivity in a multiprotocol environment 99
  dependencies for NFS and SMB 99
  how ONTAP creates 100
  valid characters for 99
directory searches
  configuration options for LDAP 43
disabling
  NFS Kerberos on a LIF 69
  NFSv3 54
  NFSv4 54
  NFSv4 referrals 86
  parallel NFS 55
  pNFS 55
  rquota support 91
  vStorage over NFS 91
displaying
  export policies 70
  export rules, command for 70
  information about locks 77
  LDAP client configurations, command for 68
  LDAP client schema templates, command for 68
  LDAP configurations, command for 68
  LDAP statistics for SVMs to monitor performance and diagnose issues 46
  local UNIX groups, command for 65
  local UNIX users, command for 65
  name mappings, command for 51, 64
  name service switch entries 63
  netgroups, command for 67
  NFS Kerberos interface configurations 69
  NFS Kerberos realm configurations, command for 69
  NFS servers 60
  NFS statistics 86
  NIS domain configuration binding status, command for 67
  NIS domain configurations, command for 67
  volume mount and junction point information 18
dDNS connections
  troubleshooting 63
dDNS statistics
  displaying for SVMs 87
  monitoring 87
documentation
  how to receive automatic notification of changes to 106
  how to send feedback about 106
domains
  commands for managing NIS 67
  performing stricter access checking for netgroups by verifying 58

E
enabling
  NFS Kerberos on a LIF 69
  NFSv3 54
  NFSv4 54
  NFSv4 referrals 86
  parallel NFS 55
  pNFS 55
  rquota support 91
  vStorage over NFS 91
exchanging
  name mappings, command for 51, 64
export
  NFS 8
export policies
  capability to enable configuration checking 24
  default, for SVMs 25
  flushing caches 72
  how ONTAP uses caches 31
how they control client access to qtrees 24
how they control client access to volumes 24
introduction to securing NFS access using 24
managing 70
removing from qtrees 33
restrictions and nested junctions 34
export policy caches
introduction to managing 72
export policy rules
how the access cache works to store results from 32
export rules
commands for managing 70
how they work 25
how to configure anonymous access 30
how to configure superuser access 30
export-level security
how used to secure access 12
exporting qtrees
requirements for using NFS file access reference guide to understand, plan, and implement 7
exporting volumes
requirements for using NFS file access reference guide to understand, plan, and implement 7
external services
requirements for using Kerberos with NFS 34

F
features, NFS
requirements for using reference guides to understand, plan, and implement 7
feedback
how to send comments about documentation 106
file access
how Data ONTAP controls 11
NFS, managing 54
tasks for setting up for NFS 24
file access, NFS
requirements for using reference guides to understand, plan, and implement 7
file and folder access
how three levels of security used to secure 12
file and record locking
NFSv4, described 84
file delegations
enabling or disabling NFSv4 read 83
enabling or disabling NFSv4 write 83
NFSv4, managing 83
file locking
between protocols, explained 76
file locks
breaking 78
displaying information about 77
introduction to managing 75
file name translations
commands for managing NFS to SMB character mapping for 103
configuring character mapping for SMB on FlexVol volumes 101
file names
case-sensitivity in a multiprotocol environment 99
dependencies for NFS and SMB 99
how ONTAP creates 100
valid characters for 99
file operations
validating qtree IDs for 34
file permissions
effect of security styles on 19
file-level and folder-level security
how used to secure access 12
first-read filters
how they work with NFS clients 79
first-write filters
how they work with NFS clients 79
FlexVol volumes
commands for managing NFS to SMB character mapping on 103
configuring character mappings for SMB file name translation on 101
configuring security style on 22
flushing
export policy caches 72
FPolicy
how first-read and first-write filters work with NFS 79

G
group IDs
configuring number allowed for NFS users 93
group information
configuration options for LDAP directory searches for 43
group memberships
enabling LDAP RFC2307bis support to use nested 43
groups
commands for managing local UNIX 65
local UNIX, managing limits for 66
guides, reference
requirements for using NFS file access guide 7

H
hard mounts
use of 98

I
Infinite Volumes
where to get information about security styles of 19
information
how to send feedback about improving documentation 106
inserting
name mappings, command for 51, 64

J
junction point
and volumes 8
concepts 8
junction points
creating volumes without specified 16
displaying information about volume 18
Kerberos
- commands for managing realm configurations 69
  - for NFS, commands for managing interface configurations 69
  - introduction to using with NFS for strong security 34
  - ONTAP support for 34
  - requirements for configuring with NFS 35
  - requirements for external services 34

LDAP
- commands for managing 68
  - commands for managing client configurations 68
  - commands for managing client schema templates 68
  - configuration options for directory searches 43
  - enabling RFC2307bis support to use nested group memberships 43
  - improving netgroup search performance 45
  - requirements for using as a name service 40
- LDAP connections
  - troubleshooting 63
- LDAP over TLS
  - introduction to configuring and using to secure communication 41
  - introduction to how ONTAP uses to secure LDAP communication 41
- LDAP signing and sealing
  - introduction to configuring and using to secure communication 41
  - introduction to how ONTAP uses to secure LDAP communication 41
- LDAP statistics
  - displaying for SVMs to monitor performance and diagnose issues 46
- limitations
  - of ONTAP support for NFSv4 96
  - of ACEs for NFSv4 ACLs 82
  - load-sharing mirror requirement for root volume 8
- loading
  - netgroups from URI, command for 67
  - local UNIX groups
    - commands for managing 65
    - limits for 66
    - managing limits for 66
  - local UNIX users
    - commands for managing 65
    - limits for 66
    - managing limits for 66
- locking grace period
  - NFSv4, specifying 85
- locking lease period
  - specifying NFSv4 84
- locks
  - breaking 78
  - differences between ONTAP and Windows handling of, on share path components 77
  - displaying information about 77

M
- managing
  - export policies 70
  - export rules, commands for 70
  - LDAP client configurations, commands for 68
  - LDAP client schema templates, commands for 68
  - LDAP configurations, commands for 68
  - limits for local UNIX users and groups 66
  - local UNIX groups, commands for 65
  - local UNIX users, command for 65
  - name mappings, commands for 51, 64
  - name service switch entries 63
  - NFS Kerberos interface configurations, commands for 69
  - NFS Kerberos realm configurations, command for 69
  - NFS servers 60
- mappings, name conversion rules 49
- memberships
  - enabling LDAP RFC2307bis support to use nested group 43
- modifying
  - export rules, command for 70
  - LDAP client configurations, command for 68
  - LDAP client schema templates, command for 68
  - LDAP configurations, command for 68
  - local UNIX groups, command for 65
  - local UNIX users, command for 65
  - name mapping patterns, command for 51, 64
  - name service switch entries 63
  - NFS Kerberos interface configurations 69
  - NFS Kerberos realm configurations, command for 69
  - NFS servers 60
  - NFSv3 and NFSv4 TCP maximum transfer size 92
  - NFSv3 services ports 59
  - NFSv4 ACLs, enabling or disabling capability of 81
  - NIS domain configurations, command for 67
  - server implementation ID 79
- mount requests
  - controlling NFS, from nonreserved ports 56
- mounting
  - NFS exports using nonreserved ports 58
  - volumes in NAS namespaces 17
- mounts
  - use of hard 98

N
- name mapping
  - how it works 47
- name mappings
  - commands for managing 51, 64
  - conversion rules 49
  - creating 49
  - how used 47
  - multidomain searches for UNIX user to Windows user 48
- name service switch entries
  - commands for managing 63
  - creating 63
  - deleting 63
  - displaying 63
NIS statistics
- displaying for SVMs 89
- monitoring 89

NTFS
- controlling root user access 95
- when to choose as security style 20

NTFS volumes or qtrees
- handling access for unknown UNIX users 56

ONTAP
- and namespaces 8
- differences with Windows handling of locks on share path components 77
- virtualization 8
- operations
  - validating qtree IDs for file 34

parallel NFS
- enabling or disabling 55
- parameters, timeout
  - optimizing access cache performance 75
- path components
  - differences between ONTAP and Windows handling of locks on share 77
- performance
  - optimizing access cache 75
- performance, NFSv3 and NFSv4
  - improvement by modifying the TCP transfer size 92

permissions
- effect of security styles on file 19
- how ONTAP preserves UNIX 21
- UNIX, how to manage using Windows Security tab 21

pNFS
- enabling or disabling 55
- support for 97

policies
- flushing caches of export 72
- how ONTAP uses caches in export 31
- removing export, from qtrees 33

policy rules, export
- how the access cache works to store results from 32

ports
- changing for NFSv3 services 59
- controlling NFS mount requests from nonreserved 56

preferred trusted domains
- how used with multidomain searches for user name mapping 48

protocol, NFS
- requirements for using reference guides to understand, plan, and implement 7

protocols
- file locking between, explained 76

qtrees
- configuring security style on 22
- how export policies control client access to 24
- removing an export policy from 33
- validating IDs for file operations 34
- qtrees, exporting
  - requirements for using NFS file access reference guide to understand, plan, and implement 7
- query configuration options
  - LDAP directory search 43
- queues
  - displaying export policy netgroup 73

read file delegations
- enabling or disabling NFSv4 83

read-only bits
- how ONTAP treats 76

realm configurations
- commands for managing NFS Kerberos 69
- reference guides
  - requirements for using NFS file access guide 7
- referrals
  - enabling or disabling NFSv4 86
  - how they work for NFSv4 85

removing
- export policies from qtrees 33
- renaming
  - export policies 70

requirements
- Kerberos with NFS configuration 35

restrictions
- authentication-based 12

RFC2307bis support
- enabling LDAP to use nested group memberships 43

root user
- controlling access to NTFS security-style data 95

root volume
- requirement for load-sharing mirror copy 8

root volumes
- configuring security style on SVM 21

rquotas
- enabling or disabling 91

rules, conversion
- name mapping 49

rules, export policy
- how the access cache works to store results from 32

schema templates
- commands for managing LDAP client 68

secure communication
- introduction to using LDAP signing and sealing or LDAP over TLS for 41

security
- requirements for using Kerberos with NFS 34

three levels used to secure file and folder access 12

security styles
- configuring on FlexVol volumes 22
- configuring on qtrees 22
- configuring on SVM root volumes 21
effects on file permissions 19
how inheritance works 20
how to choose 20
introduction to how they affect data access 19
when and where to set 20
Security tab
how to manage UNIX permissions using Windows 21
security types
how client access levels are determined by 28
how to handle clients with unlisted 26
server implementation ID
modifying 79
server, NFS
requirements for using NFS file access reference
guide to understand, plan, and implement 7
services
how ONTAP uses name 13
services, name
introduction to configuring 38
share
CIFS 8
SMB 8
share path components
differences between ONTAP and Windows handling
of locks on 77
share-level security
how used to secure access 12
signing and sealing, LDAP
introduction to configuring and using to secure
communication 41
SMB
commands for managing NFS character mapping to
103
differences between ONTAP and Windows handling
of locks on share path components 77
file and directory naming dependencies 99
file locking between protocols explained 76
how ONTAP treats read-only bits 76
SMB clients
configuring character mapping for file name
translation on FlexVol volumes 101
SMB share
concepts 8
SMB shares
implications of case-sensitivity of file and directory
names when creating or accessing files in a
multiprotocol environment 99
soft mounts
use of 98
statistics
displaying NFS 86
statistics, DNS
displaying for SVMs 87
monitoring 87
statistics, NIS
displaying for SVMs 89
monitoring 89
Storage Virtual Machine (SVM)
and namespace 8
root volume 8
Storage-Level Access Guard
how used to secure access 12
strong security
requirements for using Kerberos with NFS 34
suggestions
how to send feedback about documentation 106
superuser access
how to configure with export rules 30
surrogate pairs
how ONTAP handles file names containing UTF-16 100
SVM
and namespace 8
default export policy for 25
root volume 8
SVMs
configuring security style on root volume 21
displaying DNS statistics for 87
tasks for setting up file access using NFS 24
SVMs with FlexVol volumes
default export policy for 25
switch name services
how ONTAP uses 38
T
TCP
controlling NFS access over 55
modifying maximum transfer size for NFSv3 and
NFSv4 92
modifying transfer size to improve NFSv3 and
NFSv4 performance 92
templates
commands for managing LDAP schema 68
terminology
LDAP over TLS 41
time-to-live
configuring for cached NFS user credentials 71
reasons for modifying NFS credential cache 70
timeout parameters
optimizing access cache performance 75
timeout values
how they work for access cache 33
TLS, LDAP over
introduction to configuring and using to secure
communication 41
troubleshooting
DNS, LDAP 63
name service issues 60
name services 63
trusted domains
discovered, how used with multidomain searches for
user name mapping 48
Twitter
how to receive automatic notification of
documentation changes 106
U
UDP
controlling NFS access over 55
UNIX
commands for managing local groups 65
commands for managing local users 65
creating name mappings for 49
limits for local users, groups, and group members 66
managing limits for local users and groups 66
when to choose as security style 20
UNIX permissions
  how ONTAP preserves 21
  how to manage using Windows Security tab 21
UNIX users
  handling access to NTFS volumes or qtrees for unknown 56
  volumes in NAS namespaces 17
user authentication
  how ONTAP uses name services for 13
user ID domains
  specifying for NFSv4 38
user information
  configuration options for LDAP directory searches for 43
user name mappings
  preferred trusted domains used with multidomain search for 48
user names
  how mapping works 47
users
  commands for managing local UNIX 65
  local UNIX, managing limits for 66
UTF-16 supplementary characters
  how ONTAP handles file names containing 100

V

verifying
  status of netgroups, command for 67
VMware
  enabling or disabling vStorage over NFS 91

supported vStorage APIs for NFS 90
volume
  and junction points 8
  and namespaces 8
volumes
  commands for managing NFS to SMB character mapping on FlexVol 103
  configuring character mappings for SMB file name translation on FlexVol 101
  configuring security style on FlexVol 22
  creating without specified junction points 16
  displaying mount and junction point information 18
  how export policies control client access to 24
  introduction to creating and managing in NAS namespaces 15
  mounting and unmounting in NAS namespaces 17
volumes, exporting
  requirements for using NFS file access reference guide to understand, plan, and implement 7
volumes, root
  configuring security style on SVM 21
vStorage
  enabling or disabling over NFS 91
  supported APIs for NFS 90

W

Windows
  creating name mappings for 49
  NFS clients, support for 51
Windows Security tab
  how to manage UNIX permissions using 21
write file delegations
  enabling or disabling NFSv4 83