Technical Report

# Namespaces in Clustered Data ONTAP

Jay Goldfinch, NetApp
January 2014 | TR-4129

## Namespaces in Clustered Data ONTAP

This technical report is an introduction to the key namespace concepts in the clustered NetApp® Data ONTAP® operating system. Throughout this paper, unless otherwise specified, the term *Data ONTAP* refers to clustered Data ONTAP. References to 7-Mode are stated specifically.

**TABLE OF CONTENTS**

**LIST OF FIGURES**

# 1 Introduction

In clustered Data ONTAP, NAS clients access flexible volumes through a storage virtual machine (SVM). SVMs abstract storage services from their underlying hardware.

**Note:**   In the clustered Data ONTAP command line, an SVM is referred to as a Vserver.

Flexible volumes containing NAS data are junctioned into the owning SVM in a hierarchy. This hierarchy presents NAS clients with a unified view of the storage, regardless of the physical location of flexible volumes inside the cluster.

When a flexible volume is created within the SVM, the administrator specifies the junction path for the flexible volume. The junction path is a directory location under the root of the SVM where the flexible volume can be accessed. A flexible volume's name and its junction path do not need to be the same.

Junction paths allow each flexible volume to be browsable, like a directory or folder. NFS clients can access multiple flexible volumes using a single mount point. CIFS clients can access multiple flexible volumes using a single CIFS share.

A *namespace* consists of a group of volumes connected using junction paths. It is the hierarchy of flexible volumes within a single SVM as presented to NAS clients.

As storage or performance requirements for an SVM change, the SVM's namespace can adapt accordingly. Additional flexible volumes can be added nondisruptively to increase capacity, with no client-side configuration changes. Flexible volumes in an SVM with higher performance requirements can be physically moved nondisruptively to aggregates with faster storage. Flexible volumes can be physically moved to different aggregates inside the cluster without affecting their logical location in the SVM's namespace.

**Figure 1) An SVM namespace. A dotted line represents a junction. A solid line represents a traditional directory or folder. The client view (top) is independent of the physical storage layout (bottom).**
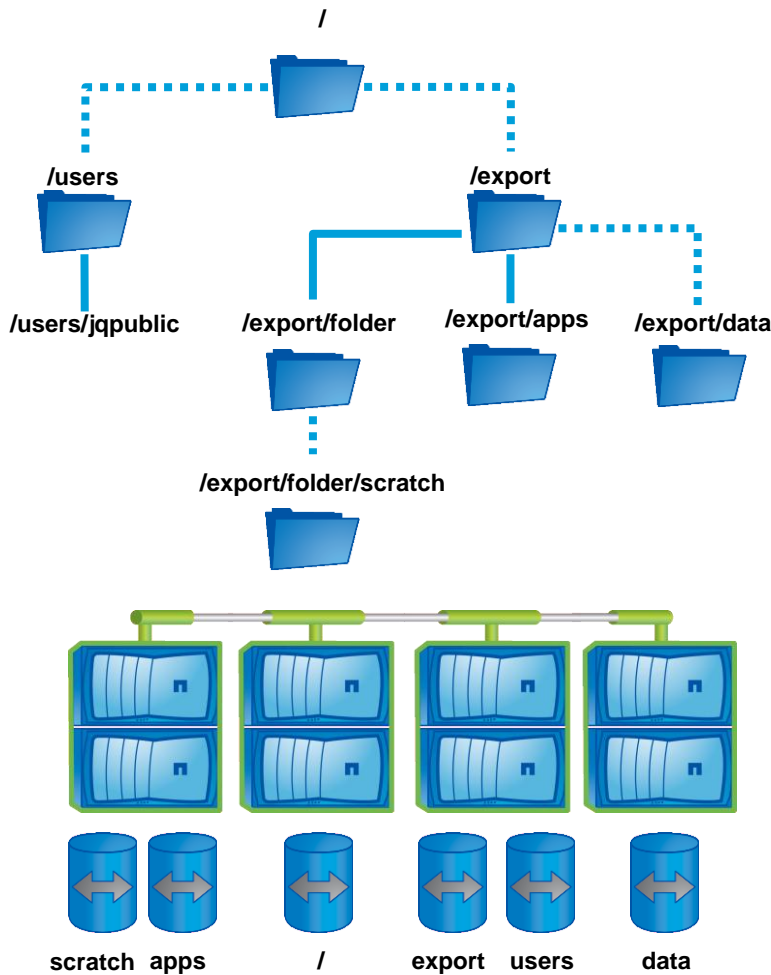


## 1.1 Purpose and Scope

This document describes the best practices and recommendations for creating a unified NAS namespace using flexible volumes and clustered Data ONTAP.

This document does not provide detailed steps for configuring the namespace. For configuration steps, refer to the clustered Data ONTAP documentation.

This document does not provide detailed information on configuring a namespace with NetApp Infinite Volume. For details on configuring NetApp Infinite Volume, see TR-4037: "Introduction to NetApp Infinite Volume" or TR-4178: "Infinite Volume Deployment and Implementation Guide."

## 1.2 Intended Audience

This document is intended for technical audiences. It assumes that the reader has basic knowledge of NAS, IP networking, NetApp storage systems, clustered Data ONTAP, client authentication, and directory services.

# 2   Junctions

A *junction* joins together two flexible volumes in the same SVM. A junction in a parent flexible volume redirects client requests to a child flexible volume. A flexible volume must be junctioned into an SVM's namespace for clients to access it.

A junction is analogous to a mount in a UNIX® system, where a directory serves as an entry point to another file system.

Junctions differ from NFS automounters and client-side mounts in several respects:

- Junctions allow the nondisruptive movement of a flexible volume from one physical node to another while preserving client connectivity, NetApp Snapshot™ copies, and replication relationships.

- Junctions require no client-side configuration or automount map.

- Junctions allow multiple flexible volumes under a single NFSv3 export, reducing the number of mounts used by the client.

- Junctions are centrally managed in the Data ONTAP cluster.

- File permissions on a junction path are not controlled by clients. They are controlled using the permissions of the flexible volume that exists at the junction path.

No file or directory can exist in the full path to be used for the junction, but the parent directory containing the junction must exist before the junction is created. For example, if an administrator wants to junction a flexible volume into the namespace at /x/y/z, there must be a directory or junction at /x/y but there cannot be an existing file, directory, or junction at /x/y/z.

An SVM's top-level flexible volume is known as the *SVM root volume.* The SVM root volume forms the root of the flexible volume hierarchy in an SVM. It is the parent, grandparent, or ancestor of every flexible volume in the SVM's namespace.

A junction can exist beneath any directory in a parent flexible volume. A flexible volume can be junctioned to the namespace in only one location. This prevents the same flexible volume from appearing more than once in the namespace.

Junctions are implemented as a file system entry in the parent flexible volume. Information about the physical location of the child flexible volume is maintained by the cluster. If an administrator moves the child flexible volume to a different aggregate inside the cluster, the cluster sends client traffic to the new location automatically.

Junctions can be created by using the `-junction-path` argument to `volume create`, the `volume mount` command, or the Namespace panel in the NetApp OnCommand® System Manager.

## 2.1   Location Transparency

The directory hierarchy visible to the clients is independent of the physical layout of the flexible volumes inside the cluster.

When an administrator issues a `volume move` command, a flexible volume is moved from one physical aggregate to another. Client traffic is redirected to the new location automatically upon completion. The client-visible path information remains the same.

After the storage administrator has created a junction, it cannot be deleted from a client. An attempt to delete a directory that is a junction path results in an error.

## 2.2   Namespace Mobility

Just as a flexible volume can be physically moved from one aggregate to another, flexible volumes can be logically moved inside the namespace. This gives an administrator the flexibility to quickly reorganize

large amounts of data without copying the data, losing Snapshot copies and replication relationships, or losing storage efficiency.

For example, suppose that a flexible volume under a CIFS share needs to be migrated quarterly. Using junctions, the flexible volume located under `\\finance.acme.com\currentquarter` can be relocated to `\\finance.acme.com\archives\Q1Y15` without copy operations or client configuration changes.

Note that the flexible volume previously junctioned under `currentquarter` can be attached at any directory or subdirectory. Unlike a directory rename operation, a flexible volume can be rejunctioned anywhere in the namespace. Because rejunctioning changes the location of a flexible volume inside the namespace, it is not transparent to client access. The client now has to access the data at the new directory location.

After the old flexible volume has been moved to a new directory, a new flexible volume can be junctioned under `current_quarter`.

## 2.3   Efficient File Distribution

Junctions can be used to distribute directories across flexible volumes. This is valuable in environments that stress the limits of individual NAS systems.

High-file-count environments can stress the capabilities of individual nodes, aggregates, and flexible volumes. Junctions can be used to create a distributed namespace that reduces demand on individual resources. TR-3537: "High-File-Count Environment Best Practices," presents an in-depth discussion of strategies to use in these situations.

Junctions can also be used to distribute an SVM's working set. This allows a storage administrator to distribute frequently accessed data across the cluster. Node-bound physical resources such as buffer cache, CPU, and NetApp Flash Cache™ intelligent caching can be used more effectively when data is distributed across nodes.


# 3   Export Policies

An *export policy* is used to control client access to a flexible volume. Export policies specify those clients, protocols, and users that have read access, write access, superuser access, or no access at all.

Each flexible volume has an export policy associated with it. A single export policy can be used by multiple flexible volumes.

Starting with clustered Data ONTAP 8.2.1, export policies can also be applied to individual qtrees inside a flexible volume. These qtree export policies are supported for NFSv3 access only. Qtree export policies are intended for the transition of existing Data ONTAP 7-Mode environments that have a large number of individually exported qtrees. 7-Mode qtrees that are being used only for quotas do not require migration to individually exported clustered Data ONTAP qtrees.

Deduplication, SnapMirror[®], SnapVault[®], quality of service (QoS), and Snapshot retention policies are managed at the flexible volume level. Similarly, nondisruptive volume movement is available at the volume level but not at the qtree level. For these reasons, NetApp recommends the use of volume export policies when possible. If a qtree does not have an export policy, it inherits the export policy of its containing flexible volume.

The administration of export policies and rules is described in the "File Access and Protocols Management Guide" and TR-4067: "Clustered Data ONTAP NFS Implementation Guide." Refer to those documents for additional guidance. TR-4052: "Clustered Data ONTAP Transition Guide" discusses in detail the migration of qtrees from 7-Mode to clustered Data ONTAP. Export policies are also discussed in section 5, "Protocol-Specific Considerations."

## 3.1 Usage

Administrators can use export policies to restrict client access in NFS environments that do not use Kerberos. When necessary, administrators can also use export policies to supplement other access controls.

NetApp does not recommend restrictive export policies for modern CIFS or kerberized NFS environments. In these environments, each user is centrally password authenticated. Therefore, permissive export policies can safely export all data to every client. Users and administrators can control access over CIFS or NFS by using file permissions. Administrators can also control access over CIFS by using share permissions.

## 3.2 Rules

An export policy is composed of an ordered set of rules. When a client request is made, the rules are checked in sequence to determine access rights. The first rule that matches the request is applied.

Rules can use the following criteria to determine access permissions:

- A client specification in the form of a host name, a netmask, a netgroup, or a domain name
- A protocol specification
- An external authentication type

A rule can grant or deny:

- Read access
- Write access
- Superuser access
- Rights to set user ID or to set group ID (UNIX only)
- Device file creation rights (UNIX only)

A rule can also specify how to handle access requests from a root UNIX user. A root user can be mapped to another user ID, require more stringent authentication, or be denied access.

## 3.3 Namespace Accessibility

Each flexible volume can have its own export policy. This can affect namespace accessibility and capacity planning.

To reach a descendant flexible volume, all of its ancestor flexible volumes must allow read access. If a client cannot read from an ancestor flexible volume, that client will not be able to traverse the namespace to reach a descendant flexible volume. In some cases, this is desirable. In many cases, it is not. For this reason, NetApp recommends that flexible volumes at or near the root of the namespace have export policies that allow read access to all clients that use the SVM.

If a flexible volume is not intended to hold user data, its export policy should deny write access. For example, it is common for SVM root volumes to house only junctions to other flexible volumes. These SVM root volumes are small. To prevent capacity exhaustion, they should not contain user data. An export policy can be used to prevent write access to these volumes.

# 4 Storage Considerations

Size specifications, Snapshot copies, Snapshot schedules, SnapMirror relationships, and storage efficiency are all managed at the FlexVol® level. Because a group of flexible volumes can be junctioned together to form a single namespace, the storage administrator has a great deal of flexibility in architecting policies at different points in the namespace.

For example, if the storage administrator has a flexible volume of archived data junctioned at `/archive`, that admin might want to enable deduplication and compression. If the storage administrator has another flexible volume containing virtual machine images junctioned at `/vms`, that admin might want to enable deduplication but not compression. If there is a third flexible volume containing temporary storage junctioned at `/scratch`, the admin might not want to enable deduplication or compression.

Each of these flexible volumes might also have different Snapshot schedules, SnapMirror relationships, provisioning, autogrow specifications, quotas, and alerts.

## 4.1   Load-Sharing Mirrors

A load-sharing mirror of a source flexible volume is a full, read-only copy of that flexible volume. Load-sharing mirrors are used to transparently off-load client read requests. Client write requests will fail unless directed to a specific writable path.

Load-sharing mirrors can be used to enable the availability of the data in the source flexible volume. Load-sharing mirrors provide read-only access to the contents of the source flexible volume even if the source becomes unavailable. A load-sharing mirror can also be promoted to become the read-write volume.

A cluster might have many load-sharing mirrors of a single source flexible volume. When load-sharing mirrors are used, every node in the cluster should have a load-sharing mirror of the source flexible volume. The node that currently hosts the source flexible volume should also have a load-sharing mirror. Identical load-sharing mirrors on the same node yield no performance benefit.

Load-sharing mirrors are updated on demand or on a schedule that is defined by the cluster administrator. Writes made to the mirrored flexible volume are not visible to readers of that flexible volume until the load-sharing mirrors are updated. Similarly, junctions added in the source flexible volume are not visible to readers until the load-sharing mirrors are updated. Therefore, NetApp recommends load-sharing mirrors for flexible volumes that are frequently read but infrequently written to.

SVM root volumes are typically small, contain only junctions to other volumes, do not contain user data, are frequently read, and are infrequently updated. SVM root volumes must be available for clients to traverse other volumes in the namespace. This makes SVM root volumes good candidates for mirroring across different nodes in the cluster.

In versions of clustered Data ONTAP prior to 8.2, load-sharing mirrors were used to distribute access to read-only datasets. Clustered Data ONTAP 8.2 introduces NetApp FlexCache® technology, which can also be used to distribute read access but provides write access and is space efficient. FlexCache is described in detail in section 4.2.

Load-sharing mirrors are capable of supporting NAS only (CIFS/NFSv3). They do not support NFSv4 clients or SAN client protocol connections (FC, FCoE, or iSCSI).

For more information about load-sharing mirrors, see the "Logical Storage Management Guide" and TR-4015: "SnapMirror Configuration and Best Practices Guide for Clustered Data ONTAP 8.2."

## 4.2   FlexCache

FlexCache is available starting in clustered Data ONTAP 8.2. FlexCache volumes are used to accelerate read requests and metadata access to an origin volume. The origin volume must reside on a node in the same cluster as the FlexCache volumes.

Multiple FlexCache volumes can be used to accelerate access to an origin volume. A FlexCache volume corresponding to a single origin volume can be placed on every node in a cluster, allowing read performance to scale well beyond what the origin volume alone could provide.

If a FlexCache volume exists on the same node as a logical interface (LIF), read requests arriving at the LIF are directed to the local FlexCache volume. However, if the origin volume is present on the same

node, all client I/O goes to the origin volume, not to the FlexCache volume. It is still desirable to have a FlexCache volume on the node hosting the source volume, in case the source volume is moved to an aggregate on a different node.

If there is no local FlexCache volume on the node hosting the client LIF, all access is directed to the origin volume. In other words, a client request is serviced by a FlexCache volume only if the data LIF accessed by the client resides on the same node as the FlexCache volume. If no such FlexCache volume exists, then all access is directed to the origin volume.

FlexCache volumes cache individual blocks from the origin volume. The first time a FlexCache volume receives a read request for a block, it must retrieve that block from the origin volume. The origin volume also grants a read delegation to the FlexCache volume. Subsequent requests for that block can be met by the FlexCache volume without consulting the origin.

FlexCache volumes act as a write-through cache, meaning that any writes are automatically directed to the origin volume. A write is successful when it is completed on the origin and any read delegations previously granted to other FlexCache volumes have been revoked.

Unlike load-sharing mirrors, FlexCache volumes provide coherent, transparent write access to all users of the cached volume. FlexCache volumes do not require the use of a special writable path or a mirror update operation.

FlexCache is NAS only. FlexCache volumes are effective in caching requests for NFSv3, NFSv4, SMB2.x, and SMB 3 clients. pNFS (NFS v4.1) is unaware of a FlexCache volume; pNFS clients always send their I/O requests to the origin volume.

Because FlexCache volumes do not contain a complete copy of the origin volume, they do not provide availability if the origin volume becomes unreachable. If this level of protection is required, consider using load-sharing mirrors.

FlexCache in clustered Data ONTAP is currently available only for intracluster caching; therefore  NetApp recommends 7-Mode FlexCache when caching NFS access to a clustered Data ONTAP origin volume over a WAN. 7-Mode FlexCache can be used to cache individual volumes only. Junctions cannot be traversed when accessed through a 7-Mode FlexCache volume.

NetApp recommends using Microsoft® BranchCache® when caching CIFS access over a WAN. Clustered Data ONTAP 8.2 can act as a BranchCache content server. BranchCache is used to cache individual files; it will not cache a namespace or a directory structure.

## 4.3   Cloning and Data Protection

Junctions in a NetApp FlexClone® volume or a Snapshot copy of a flexible volume cannot be traversed.

SnapMirror destination flexible volumes are not automatically junctioned into the namespace of the destination SVM. The administrator can recreate the namespace hierarchy in a SnapMirror destination as needed.

Junctions in a flexible volume that has been restored using SnapRestore® software can be traversed. If the Snapshot copy used for the volume SnapRestore contains a junction to a child flexible volume that has since been deleted, that junction cannot be traversed.

A junction that cannot be traversed appears to NFS clients as an empty read-only directory. CIFS clients receive an error if they attempt to access a junction that cannot be traversed.

For more information about backup and recovery, see the "Data Protection Guide," the "Data Protection Tape Backup and Recovery Guide," and TR-4015: "SnapMirror Configuration and Best Practices Guide for Clustered Data ONTAP 8.2."

## 4.4 LIF Configuration

An SVM's LIFs, like its flexible volumes, can be placed on any node in a cluster. This allows network access to scale along with storage capacity and performance. Namespace accessibility is not affected by the location of the physical port hosting a LIF.

## 4.5 Language Settings

When an SVM is created, its language setting is declared. Starting in Data ONTAP 8.2, volumes can have different language settings than the SVM in which they are contained.

By default, a volume inherits the language of the containing SVM. It is possible to specify a different language for the volume at creation time. It is also possible to change the language of an SVM; however, volume language cannot be changed. In Data ONTAP 8.2, SnapMirror copying between SVMs of different language types can occur, but the source and destination volumes must have the same language.

The default language for SVMs in Data ONTAP 8.2 is C.UTF-8. This setting provides a neutral, non-country-specific encoding. Unless technical requirements dictate the use of a different language encoding, consider using C.UTF-8.

A client must be able to traverse a parent volume to reach a child volume. If a client's language setting prevents it from representing the character set used by the parent volume, the child volume is not reachable by that client.

For in-depth guidance in selecting the appropriate SVM and volume languages, see TR-4160: "Secure Multi-Tenancy in Clustered Data ONTAP."

## 4.6 Consistency Groups

A *consistency group* is a consistent set of Snapshot copies taken at the same time on multiple volumes. When a consistency group is created, no volume can be ahead of another in terms of dependent writes.

For example, consider an application that is architected to recover from a consistent dataset. This application issues writes one after another. If any write appears in the dataset, all the writes issued before it should appear and none of the writes issued after it should appear.

A consistency group taken to back up an application reflects a precise point in that application's series of writes.

Consistency groups are available starting in clustered Data ONTAP 8.2. They can be created using NetApp ZAPI API calls. They can be used to back up parts of a namespace or an entire namespace. They are also used by NetApp SnapManager® products and are used natively by clustered Data ONTAP to back up an Infinite Volume.

# 5 Protocol-Specific Considerations

## 5.1 NFS

### Exports

Traditional NFSv3 servers are limited to a single local file system per NFS export. If a traditional NFSv3 server has a local file system mounted at `/export` and another mounted at `/export/data`, an export entry containing `/export` does not export `/export/data`. Two separate exports would be required.

In other words, a traditional NFSv3 server cannot export a single namespace that includes all of its local file systems. Each of its local file systems must be exported individually.

Because traditional NFSv3 servers export file systems individually, the exports and their access controls are defined in the same exports file. The name and location of this file depend on the NFS server implementation. A typical location is `/etc/exports`.

Clustered Data ONTAP does not require a separate NFS export for each volume. A namespace that includes every volume in the SVM is placed under a single NFS export. The export path is the same in every NFS-enabled SVM. It is always `/`.

When a UNIX client issues a `showmount -e` against a traditional NFSv3 server, one entry per exported file system is returned. In the example above, it returns `/export` and `/export/data`. A `showmount -e` command issued against an NFS-enabled SVM returns one entry: `/`.

Even though clustered Data ONTAP does not require each volume to be exported individually, the administrator can configure access controls by using export policies. Export policies are analogous to the access specifications following the local file system path in a traditional `/etc/exports` file.

Clients can mount a specific location in an SVM's namespace by using a deep mount. With clustered Data ONTAP, clients also have the option to mount the top of an SVM's namespace, `/`, and browse to the appropriate location in the namespace.

## Deep Mounts

An NFS-enabled clustered Data ONTAP SVM, like many traditional NFS servers, allows a client to mount a path under a top-level export.

For example, if an NFS-enabled SVM has a volume junction path, qtree, or directory in `/users/jqpublic`, an NFS client can mount this subtree directly. The client does not need to mount the SVM's top-level export, `/`, then browse to `/users/jqpublic`.

Stated another way, if the SVM's host name is vs1, the UNIX NFS client mount command `mount vs1:/users/jqpublic /home` is legal, regardless of whether vs1's `/users/jqpublic` namespace location is a volume junction path, a qtree, or a regular directory.

## Access Controls

Junctions, a single namespace, and deep mounts eliminate the need for multiple NFS exports in a clustered Data ONTAP NFS SVM. However, these features do not provide the access control found in a traditional `/etc/exports` file. A clustered Data ONTAP NFS SVM uses export policies to enforce access control instead. Core differences between traditional NFS export rules and clustered Data ONTAP export policies follow.

Traditional NFS export access rules:

- Are defined in the exports file
- Are applied to an entire NFS export
- Cannot be reused for multiple exports
- Are used for NFS only
- Are checked on the initial NFS mount call

A clustered Data ONTAP export policy:

- Is defined using command-line or GUI tools
- Might differ across flexible volumes in the same namespace
- Can be reused by multiple flexible volumes or qtrees
- Can be used for both NFS and CIFS
- Is checked on junction and qtree traversal

## Free Space

Most UNIX and Linux® `df` implementations report the correct amount of free space, depending on which flexible volume corresponds to the path that is specified in the command. `df` does not report the aggregated free space available in descendant flexible volumes.

For example, suppose that an SVM, vs1, has 18MB free in its SVM root volume and 100GB free in a flexible volume junctioned under the path `/a_directory`. A client has vs1 mounted under `/mount/vs1`. A `df` command issued from that client reports 18MB free under `/mount/vs1`, but it reports 100GB free under `/mount/vs1/a_directory`.

Server administrators who account for free space by traversing client-side mounts should be aware of this behavior. If space accounting is required on the client side, individual flexible volumes can be mounted by using deep mounts.

Free space reporting is handled differently under NFS and CIFS. Free space reporting in CIFS is covered in section 5.2.

## 5.2   CIFS

An SVM serving NAS can provide CIFS access, NFS access, or both. In a properly configured SVM, files can be shared between NFS and CIFS clients.

To allow NFS and CIFS client access, you must set up both an NFS server and a CIFS server on the SVM. You must also configure name mapping between UNIX and Windows® users. Name mapping allows UNIX-style file permissions to be used by Windows clients and vice versa.

For guidance on configuring a CIFS server and setting up name mapping, see the "File Access and Protocols Management Guide."

### Shares

A single SVM might have multiple CIFS shares. A CIFS share can be defined at any directory or junction path in the namespace hierarchy. The storage administrator specifies a share name and a UNIX-style path to the top of the CIFS share when the share is created. Windows clients access the data in the share by using native Windows path conventions.

After the share has been created, check the security style and permission settings on the flexible volume housing the share and each of that flexible volume's namespace ancestors. If any of these flexible volumes use the UNIX or mixed security style, they must provide read and execute permissions to the users of the share.

### Access Controls

CIFS allows the definition of share-level permission settings in addition to export policies and file permissions. Three levels of access control can cause confusion when troubleshooting authentication problems. In cases in which the use of file-level and share-level permissions is sufficient, NetApp recommends permissive export policies.

Export policies can be disabled for CIFS by using the `vserver cifs options modify -is-exportpolicy-enabled` command, available in the Advanced Privilege command set.

### SMB Automatic Node Referrals

Clustered Data ONTAP 8.2 can transparently redirect CIFS clients to a LIF on a node that houses the root of a CIFS share. For the remainder of the CIFS session, access is directed to this LIF. This functionality is called SMB automatic node referrals, or Auto-location.

SMB automatic node referrals leverage the Microsoft Distributed File System (DFS) referral protocol. Modern Windows clients support DFS natively, so no additional client-side code is required to use SMB automatic node referrals.

SMB automatic node referrals require NTLM authentication. If an environment requires exclusive use of Kerberos for CIFS authentication, SMB automatic node referrals cannot be used.

If the volume containing the CIFS share contains junctions to volumes on other nodes, NetApp does not recommend the use of Auto-location. This is to prevent the LIFs on the node that hosts the parent volume from being overused.

The Auto-location feature is disabled by default. SMB automatic node referrals can be enabled at the SVM level by using the `-is-referral-enabled` option to the SVM `cifs options modify` command in Advanced Privilege mode. For additional documentation, see the "Clustered Data ONTAP 8.2 File Access and Protocols Management Guide" and TR-4191: "Best Practices Guide for Clustered Data ONTAP 8.2 Windows File Services."

### Free Space

Windows Explorer computes the free space available in a CIFS share before attempting a copy operation. In clustered Data ONTAP 8.1, some SMB 2.0 copy operations fail if the volume containing the CIFS share is smaller than the file to be copied. This occurs even if the target of the copy operation is a volume junctioned beneath the CIFS share with sufficient space. This does not occur with clustered Data ONTAP 8.2.

# 6   Typical Configurations

Namespaces in clustered Data ONTAP can be constructed in a variety of ways. Which configuration is appropriate depends on the environment in which the SVM is deployed.

As a general recommendation, SVM root volumes should not contain user data. They should house junctions and directories exclusively. Maintaining separate data and SVM root volumes allows more flexibility to modify the namespace layout as business requirements change.

An SVM root volume that contains only junctions can be protected by using load-sharing mirrors that are read-only. User writes to the SVM root volume can also be prevented by using read-only export policies.

Permissions must allow namespace traversal to a child volume. Export policies must provide read access to any ancestor volumes, including the SVM root. Ancestor directories and volumes must also provide UNIX execute permissions to the users of a child volume.
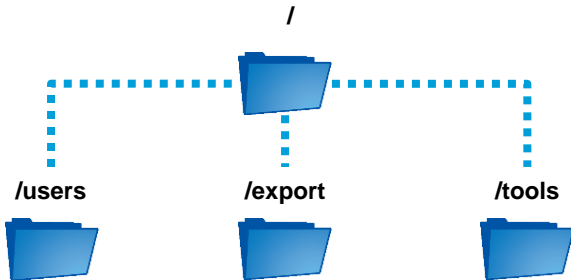
## 6.1   Flat Namespace

In a flat namespace, every flexible volume that contains user data is a child of the SVM root volume.

A flat namespace does not necessarily imply a flat directory structure in the SVM root volume. A junction can exist in the namespace at `/export/tools`, provided that `/export` is a directory in the SVM root volume, not a junction to another volume directly beneath the SVM root.

A flat namespace can provide enhanced availability. Load-sharing mirrors of the SVM root volume can protect reachability to the entire namespace.

Some applications that manage their own datastores do not require a namespace hierarchy. Those applications may be good candidates for a flat namespace.

**Figure 2) A flat SVM namespace. A dotted line represents a junction. Every volume in the namespace is junctioned directly beneath the SVM root volume.**



## 6.2   Nested Namespace

In a nested namespace, flexible volumes that house data can contain junctions as well. Just as in a flat namespace, the SVM root volume does not contain user data. However, the flexible volumes junctioned off the SVM root volume can contain junctions to descendant volumes, which can contain junctions to more descendant volumes, and so on.

NetApp recommends nested namespaces for certain environments that are not well suited to a flat namespace; for example,  an SVM that contains groups of volumes with different language settings. Each language has a volume junctioned directly beneath the SVM root volume. Any additional volumes that share the same language are descendants of the language-specific top-level volume.

Another example is a namespace that reflects a hierarchy. For example, a namespace might be constructed to reflect the elements of a project undergoing development. A namespace might also be constructed to reflect the groups and subgroups inside an organization. In many cases, the administrator may want to divide these groups by SVM, rather than using a namespace. See TR-4160, "Secure Multi-Tenancy in Clustered Data ONTAP," for more information.

In a nested namespace, ancestor volumes must be traversed to reach a child volume. Access to a child volume requires the availability of its ancestors. For this reason, the namespace should be limited to the minimum depth required by the environment.

**Figure 3) A nested SVM namespace. A dotted line represents a junction. The volumes located in /export/data, /tools/beta, and /tools/production are junctioned off data volumes, not the SVM root volume.**
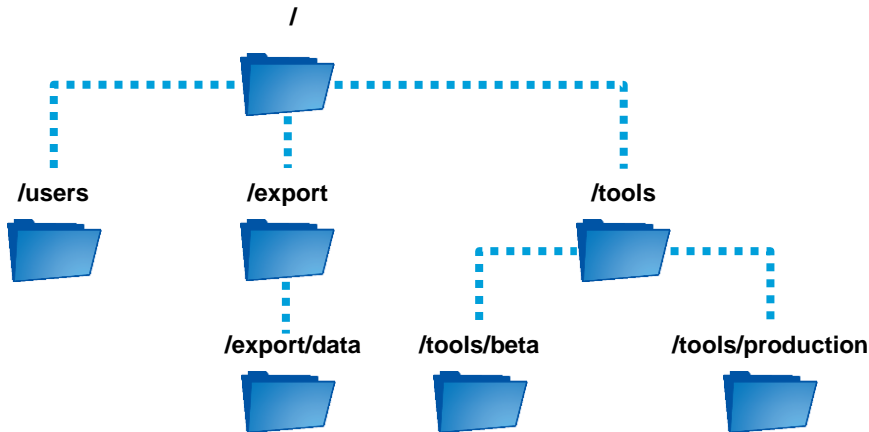
## 6.3 Infinite Volume Namespace

Infinite Volume is a cost-effective large container that can grow to PBs of storage and billions of files. An Infinite Volume is made up of constituents that reside in many cluster nodes. Infinite Volume is used for large-scale unstructured storage repositories.

An SVM that contains an Infinite Volume can contain only one Infinite Volume and no flexible volumes other than the SVM root. The Infinite Volume must be junctioned to a top level directly beneath the SVM root. The default junction path for an Infinite Volume is `/NS`.

For more information about Infinite Volume, see "TR-4178: Infinite Volume Deployment and Implementation Guide."

# 7 Additional Reading and Resources

- "File Access and Protocols Management Guide"
- "Logical Storage Management Guide"
- "Data Protection Guide"
- "Data Protection Tape Backup and Recovery Guide"
- TR-3965: "NetApp Thin Provisioning Deployment and Implementation Guide"
- TR-4037: "Introduction to NetApp Infinite Volume"
- TR-4178: "Infinite Volume Deployment and Implementation Guide"
- TR-4015: "SnapMirror Configuration and Best Practices Guide for Clustered Data ONTAP 8.2"
- TR-4067: "Clustered Data ONTAP NFS Implementation Guide"
- TR-4191: "Best Practices Guide for Clustered Data ONTAP 8.2 Windows File Services"

Refer to the Interoperability Matrix Tool (IMT) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.