NETAPP TECHNICAL REPORT

Export and Network Changes between Data ONTAP 7.0.5 and 7.2.4

Latesh Kumar KJ, NetApp
September 2008 | TR-3706

ABSTRACT

Major changes were introduced in NFS exports and mounts in Data ONTAP® 7.0.5, 7.2.1, and 7.2.4. The changes in Data ONTAP 7.0.5 are intended to secure the NFS environment. Increased customer expectation meant that the NFS code had to be redesigned in Data ONTAP 7.2.1 to introduce better access and matching logic, rule sharing, thread changes, and new access cache. This led to improved security as systems scaled up with a large number of NFS clients.

TABLE OF CONTENTS

1 INTRODUCTION

The Network File System (NFS) is a client/server application that lets a computer user view and optionally store and update files on a remote computer as though they were on the user's own computer. The user's system must have an NFS client, and the other computer must have the NFS server. Both also require TCP/IP to be installed, because the NFS server and client use TCP/IP as the program that sends the files and updates back and forth. (However, the User Datagram Protocol, UDP, which comes with TCP/IP, is used instead of TCP with earlier versions of NFS.)

NFS was developed by Sun™ Microsystems and has been designated as a file server standard. Its protocol uses the Remote Procedure Call (RPC) method of communication between computers. You can install NFS on Windows® and some other operating systems that use products like Sun's Solstice Network Client.

Using NFS, the user or a system administrator can mount all or a portion of a file system (which is a portion of the hierarchical tree in any file directory and subdirectory, including the one on your PC or Mac®). The portion of your file system that is mounted (designated as accessible) can be accessed with whatever privileges go with your access to each file (read-only or read-write).

2 ABOUT EXPORTS

The `/etc/exports` file contains the list of resources to be exported when Data ONTAP starts up, restarts, or receives the `exportfs -a` command (given that a valid NFS license exists). Maintaining the `/etc/exports` file enables resources to be available automatically upon system startup and eliminates the need to enter the `exportfs` command for each resource that you want to export.

In the `/etc/exports` file, you also specify the access restrictions and security service with which clients can mount each resource. At any time after system startup, you can override specifications in the `/etc/exports` file by using the `exportfs` command (until the system is restarted or another overriding `exportfs` command is issued).

INFORMATION IN /ETC/EXPORTS

The /`etc/exports` file specifies the following types of information:

- Complete path name of each resource to be exported.
- Targets to which each resource is exported. Targets are optional.
- Access restrictions with which each resource can be mounted. Access restrictions are optional.
- Security service that each resource uses. Security services are optional.

An export provides resources to targets, exportable resources:

- Volume
- Directory
- Qtree
- File

RULES FOR SPECIFYING AN EXPORT ENTRY

If the `/etc/exports` file entry contains only the path name of a resource, that resource can be mounted by all hosts as a read-write resource. If the `/etc/exports` file entry contains the path name of a resource followed by export options, the resource is exported with the specified access restrictions, which determine

which targets have permission to mount the resource. The type of access allowed for each target (for example, read-only or read-write)

NFS requests made by root on the client are handled in the following way. The requests can be denied, allowed, or allowed with a converted UID. In an `/etc/exports` file entry, the first access restriction option is preceded by a dash (-). Subsequent options are preceded by a comma (,). Multiple hosts associated with the same access restriction are separated by a colon (:).

TARGETS

Targets are destinations to which resources are exported.

- **Host:** Typically the UNIX® or Linux® host system connected to the storage system.
- **Netgroup:** A network-wide group of machines granted identical access to certain network resources for security and organizational reasons.
- **Subnet:** A physical grouping of connected network devices. Nodes on a subnet tend to be located in close physical proximity to each other on a LAN.
- **DNS Subdomain:** A subdomain is a domain that is part of a larger domain. A DNS hierarchy consists of the root-level domain at the top, underneath which are the top-level domains, followed by second-level domains, and finally the subdomains.

Identifies targets through:

- IP only
- Name-to-IP resolution
    - Local `/etc/hosts` file
    - Network Information Service (NIS)
    - Domain Name System (DNS)
- Group name resolution
    - Netgroups or `/etc/netgroup` file
    - Network Information Service (NIS)
    - Lightweight Directory Access Protocol (LDAP)

2.1 ACCESS RULES

Access control in NFS exists at the server as well as the client. The server uses permissions to limit read or write access to exported resources by clients. In addition, the server can limit access by using normal UNIX `rwx` controls.

The `/etc/exports` file contains the permissions assigned to exported resources. By default, `ro` or `rw` permissions are specified; read-write is the default permission. If `–rw=` is present, read-only is *not* the default for all other hosts not listed for the resource.

Also, when specifying permission levels with `ro` and `rw` options, the following are invalid combinations for `/etc/exports`:

- You cannot specify the `ro` option with the `ro=` option.
- You cannot specify the `rw` option with the `rw=` option.
- You cannot exclude an NFS client identifier from the `ro=` or `rw=` option and include the same NFS client identifier in the other option.

The order of precedence of the options is:

- The `ro` option takes precedence over the `rw` option.
- The `ro=` option takes precedence over the `rw` option.

- The `rw=` option takes precedence over the `ro` option.
- The `ro=` option takes precedence over the `rw=` option.
- A host name or IP address in the `ro=` or `rw` option takes precedence over a netgroup, subnet, or domain in the other option.
- Host names and IP addresses take precedence from left to right within an option.

Note: For details of other permissions and options, see the online manual page for exports.

COMMON EXPORTFS SWITCHES

- Displays all current exports in memory:
    - `Exportfs`
- Adds exports to the `/etc/exports` file and memory:
    - `Exportfs –p [options] [path]`
- Reloads exports only from the `/etc/exports` file:
    - `Exportfs -r`
- Unexports all:
    - `Exportfs -uav`
- Unexports a specific export:
    - `Exportfs –u [path]`
- Unexports an export and removes it from `/etc/exports`:
    - `Exportfs –z [path]`
- Verifies the path to which a volume is exported:
    - `Exportfs –s [path name]`

2.2 ABOUT ACCESS RESTRICTIONS

When you export a resource, you can specify the access restrictions that govern how the resource can be mounted. If you export a resource without specifying access restrictions, it can be mounted read-write by all hosts.

You specify access restrictions by using export options in the `/etc/exports` file or with command line options for the `exportfs` command. (These options are identical using either method.) Access restriction options determine:

- Which hosts can mount the resource
- Whether the resource can be mounted read-write or read-only
- Whether the root user on the client can access the resource
- Whether files can be created with the SETUID bit
- The user ID (UID) of the user accessing the resource

When multiple restrictions are applied to the same resource, the most restrictive option always takes precedence.

2.3 UNDERSTANDING ACCESS RESTRICTIONS

The root option: This option determines the user ID for the root user on the client who sends the NFS requests after mounting the resource. The following list describes the effect of including or excluding a host from the `root` option:

- If you specify a host with the `root` option, the root user on that host keeps the root UID (0) when accessing the resource. Having a root UID does not guarantee read-write access. If the `ro` option is specified for a resource, it overrides the `root` option, making the resource a read-only resource.
- If no `root` option is specified for the resource, or if the host that is trying to access the resource is not specified with the `root` option, access by root on the client is either denied or is subject to modification by the `anon` option.

The rw option: This option gives read-write access to the specified hosts. If no hosts are specified, all hosts have read-write access.

The ro option: This option gives read-only access to the specified hosts. If no hosts are specified, all hosts have read-only access.

The anon option: This option determines the UID for the root user on the client. The following list describes the default UID specified by the `anon` option and the meanings of some special UIDs:

- By default, the `anon` option specifies a UID of 65534. That is, if you do not use the `root` and `anon` options for a resource, root users on all hosts access the resource by using the UID 65534.
- If the `anon` option specifies a UID of 65535, root access is disabled.
- If the `anon` option specifies a UID of 0, root access is granted to all hosts.

If a name is provided instead of a UID, that name is looked up according to the order specified in the `/etc/nsswitch.conf` file to determine the corresponding UID to be assigned by the `anon` option.

3 DATA ONTAP CHANGES BETWEEN 7.0.5 AND 7.2.4

Data ONTAP 7.2.4 introduces the following

The following commands have changed beginning with Data ONTAP 7.2.4:

- `disk remove_ownership`
- `ifstat`
- `sysconfig –a`

The following options were introduced in earlier Data ONTAP releases but not noted in previous Release Notes:

- `disk.maint_center.rec_allowed_entries`
- `disk.powercycle.enable`
- `disk.recovery_needed.count`
- `wafl.inconsistent.buf_limit`
- `wafl.inconsistent.ino_limit`
- `wafl.inconsistent.snap_limit`

Data ONTAP 7.2.1 introduces the following new command:

- `charmap`

The following command was modified in Data ONTAP 7.2 but not noted in previous Release Notes:

- `cifs shares [-t]`

The following commands have been modified in Data ONTAP 7.2.1:

- `aggr verify start aggr_name [-f plexnumber]`
- `vol verify start aggr_name [-f plexnumber]`

Data ONTAP 7.2.1 introduces the following new option:

- `ndmpd.tcpnodelay.enable`

The following option was introduced in an earlier Data ONTAP release, but documentation was not available until Data ONTAP 7.2.1:

- `cifs.show_dotfiles`

The following options have been modified in Data ONTAP 7.2.1:

- `ip.ping_throttle.drop_level`
- `iscsi.max_connections_per_session`

Data ONTAP 7.2.1 removes support for the following option:

- `nfs.export.allow_provisional_access`

4 FUNCTIONAL CHANGES BETWEEN DATA ONTAP 7.0.5 AND 7.2.4

4.1 EXPORT RULES

This document provides a functional specification that is modified in export-relation with the redesign. The changes happened between Data ONTAP 7.0.5 and 7.2.1 to 7.2.4. The general approach is to maintain existing functionality unchanged.

The basic job of the export code is to provide a means of defining what hosts are allowed to access the various NFS server exported trees and to limit the type of access that each host is allowed, based on an export specification for that export.

EXPORTING DIFFERENT LEVELS OF A DIRECTORY HIERARCHY

The way that Data ONTAP exports a resource is similar to that used by the SunOS™ 4.x operating system. However, unlike SunOS, Data ONTAP permits you to export a resource whose ancestor has already been exported. For example, you can export the `/vol/vol0/home/users` directory after you export the `/vol/vol0/home` directory.

When exporting different levels of a directory hierarchy, be careful not to give clients greater access permissions at a higher level in the hierarchy. For example, in the following `/etc/exports` file, the first entry specifies access restrictions for the `/vol/vol0` volume as read-only in Data ONTAP 7.0.5 and it specifies no access restrictions in 7.2.4.

The second entry restricts access to the `/vol/vol0/home` directory to specified hosts. Because the first entry grants permission to any host to mount the `/vol/vol0` volume, hosts other than host1 and host2 can have access to the `/vol/vol0/home` directory by mounting the `/vol/vol0` volume.

```
/vol/vol0
```

```
/vol/vol0/home -rw=host1:host2
```

According to the export rules, any resource without a target is read-write; however, Data ONTAP 7.0.5 allows any host to mount the `/vol/vol0` volume default read only. The second entry allows only host1 and host2 to mount `/vol/vol0/home` in read and write with converted UID as “nfsnobody”.

INTERACTIONS AMONG OPTIONS

When Data ONTAP receives a mount request for a resource, it considers all options pertaining to the resource to determine applicable access restrictions.

The `/etc/exports` file contains the following entry for the `/vol/vol0` volume:

```
/vol/vol0 -ro=host1, root=host1, rw=host2
```

The `ro` and `root` options allow host1 to mount the `/vol/vol0` volume with root access, but host2 is not allowed to mount the resource as read-write in Data ONTAP 7.0.5. However, similar export entry in 7.2.4 allows host1 to mount `/vol/vol0` with `ro` and `root` permission and host2 has the read-write access.

ALL HOSTS TO MOUNT READ-WRITE AND READ-ONLY

In the `/etc/exports` file, enter the path name of the resource without any options:

```
/vol/vol0

/vol/vol0/home
```

The first export entry has read-only and the seond has read-write access with converted UID “nfsnobody” in Data ONTAP 7.0.5. However these two entries in 7.2.4 grant read-write access to any clients with converted UID “nfsnobody”.

FOR SPECIFIED HOSTS TO MOUNT READ-WRITE

To allow only certain hosts to mount the resource read-write and to prevent other hosts from mounting the resource, in the `/etc/exports` file, enter the path name of the resource with the `rw` option.

```
/vol/vol0           -rw=host1:host2

/vol/vol0/home      -rw=host3:host4
```

In Data ONTAP 7.0.5, host1 and host2 can mount the `/vol/vol0` directory as read-only; host3 and host4 are allowed to mount the `/vol/vol0/home` directory as read-write.

In Data ONTAP 7.2.4, host1, host2 and host3, host4 are allowed to mount `/vol/vol0` and `/vol/vol0/home` in read-write respectively.

FOR SPECIFIED HOSTS TO MOUNT READ-WRITE AND ALL OTHERS TO MOUNT READ-ONLY

To enable certain hosts to mount a resource read-write and all other hosts to mount the resource read-only, in the `/etc/exports` file, enter the path name of the resource with the `ro` and `rw` options:

```
/vol/vol0           -ro, rw=host1:host2

/vol/vol0/home      -ro, rw=host3:host4
```

In Data ONTAP 7.0.5, any hosts, including host1 and host2, are allowed to mount read-only; however, host3 and host4 are allowed to mount read-write and all others are granted read-only access.

In Data ONTAP 7.2.4, host1, host2 and host3, host4 are allowed to mount the resources `/vol/vol0` and `/vol/vol0/home` respectively; all other hosts are permitted to mount read-only.

FOR ALL USERS TO HAVE ROOT ACCESS

To give all clients root access, in the `/etc/exports` file, enter the path name of the resource with the `anon` option:

```
/vol/vol0     -anon=root

fas960c-svl03> exportfs -au
```

```
fas960c-svl03> exportfs -a                    Invalid option in Data ONTAP 7.0.5
exportfs [Line 1]: No such user root
exportfs [Line 1]: invalid option, /vol/vol0 not exported
                                              Valid option in Data ONTAP 7.2.4
```

Note: Pre-7.2.1 versions of NFS don't have user mapping; they have only UID and GID mapping. Therefore UID=0 is root, and anon=0 worked, but not anon=root. However, 7.2.1 and later do have user and UID/GID mapping. That is why it works in the latter release.

FOR ROOT USERS ON CLEINTS TO BE CONSIDERED AS "NOBODY"

In the `/etc/exports` file, enter the path name of the resource with the `anon` option.

```
/vol/vol0       -anon=65530
/vol/vol0/home -anon=65530
```

Note: The UID specified must not be 0.

In Data ONTAP 7.0.5, all root users on all clients are allowed to mount `/vol/vol0` as "nobody" read-only. However, `/vol/vol0/home` is permitted to read-write.

In Data ONTAP 7.2.4, all root users on any clients are allowed to mount `/vol/vol0` and `/vol/vol0/home` with UID "nobody" as read-write.

STANDARD AND NONSTANDARD ANON UID AND GID

In the `/etc/exports` file, enter the path name of the resource with the `anon` option.

Export rules (Anon)	Mount status
/vol/vol0/home -sec=sys,rw,anon=-0,nosuid	-rw-r--r-- 1 root root 0 Jun 29 2008 rootnegzero
/vol/vol0/home -sec=sys,rw,anon=100,nosuid	-rw-r--r-- 1 htt users 0 May 8 01:08 aix
/vol/vol0/home -sec=sys,rw,anon=-100,nosuid	-rw-r--r-- 1 4294967196 4294967196 0 Jun 29 2008 nn
/vol/vol0/home -sec=sys,rw,anon=-1000,nosuid	-rw-r--r-- 1 skeleton 1000 0 Jun 29 2008 1000

In Data ONTAP 7.0.5, any standard or nonstandard UID is allowed.

In Data ONTAP 7.2.4, `-ve` UIDs report errors on the system console; however, any UID that is being used with `anon` is carried and the same UID is created.

4.2 IMPLEMENTATION OF ROOT SQUASH ON DATA ONTAP

Some Linux NFS servers have an option called `no_squash`, which grants root access to anonymous users. Data ONTAP does not support `no_squash`. `Root_squash` is the mechanism for informing the server what user should be considered as the root user. To configure similar functionality on NetApp® systems, use `anon=0` in `/etc/exports`.

Caution: `anon=0` is dangerous. Do not use this configuration in a secure environment. The more secure option is to export with `root=ip address`; that is, `/vol/vol0/home -root=192.0.0.1`.

Using `sec=none` in the NFS exports treats all client requests as the anonymous user, which is what `all_squash` does.

```
/vol/vol0          -sec=none, anon=0
```

In Data ONTAP 7.0.5, `anon=0` treats any user as root user and allows to mount read write.

```
/vol/vol0          -sec=none, rw
```

In Data ONTAP 7.2.4, using `sec=none` in the exports treats all client requests and maps the UID as "nobody" with read-write access.

Both of these options can be set in the `/etc/exports` file on the system.

If the Linux clients do not work as expected after configuring `anon=0` on the system, you may be experiencing a known problem in RHEL4 update4 and older kernels, which is fixed in util-linux-2.12a-16.EL4.22. See the following Red Hat link for more information about this bug:

https://bugzilla.redhat.com/show_bug.cgi?id=187370

4.3 HOW TO USE THE ACTUAL OPTION

The `actual` parameter allows a server to alias the path names if storage has been moved and clients have not been updated to the new path. This parameter prevents the `/etc/exports` file from being removed, but it is removed from cache (memory). This is why the export became unmounted. The NFS shares must be exported (remounted) again for clients to connect. To remount, do an `exportfs -r` and then `exportfs`. The clients can then reconnect.

FilerView® does not permit "actual" NFS export paths without a `/vol or /etc` prefix in Data ONTAP versions 7.2.1, 7.2.1.1, 7.2.2, and 7.2.3. Users are allowed to remove the `/vol` when creating NFS exports for NFS clients view in FilerView prior to Data ONTAP 7.2.1. It was allowed to make `/(share name)` without being forced to user `/vol/(share name)` when using the path.

When upgrading from an earlier version of Data ONTAP to 7.2.2, the existing shares created by using `/(share name)` continue to work until any changes are made to the share. Existing shares that are migrated to the new version still work because of the mapping that was converted. The problem occurs after modifying existing shares.

If modified, these shares require you to use the forced `/vol/(share name)` or `/etc/(share name)`. If the `/vol` or `/etc` is removed from the `(share name` , then a notification in red appears at the top of the screen stating that you must use `/vol` or `/etc` to create the share. This feature was present in earlier releases of Data ONTAP but was not enforced. Modified share names can affect `-actual` paths.

As a best practice, modify the FSTAB on the hosts used for databases to look at absolute path `/vol/vol#/(directory)` instead of `-actual` path `/(share name)` when working with databases.

Suppose that a previous NFS server had data in `old:/old/server/info` and hundreds of NFS clients mounted this mount point. Data was then migrated to `filer:/vol/vol0/data`. Instead of reconfiguring hundreds of NFS clients to mount `filer:/vol/vol0/data`, the alternative is to the do following:

```
filer> exportfs -o access=nfsclients,actual=/vol/vol0/data /old/server/data
```

Here is how to determine the actual path to storage by using the `-s` option:

```
filer> exportfs -s /vol/vol9/vf19

/vol/vfilers/vf19


filer> exportfs -s /vol/vol9/vf19/src

/vol/vfilers/vf19/src
```

You can specify a symbollic virtual path of export entry into the `/etc/exports` file (which may or may not exist on the system, but which must start with `/vol/`) to *actually* export another path (the one specified in the `-actual` option). Here is a generic example of how to use the `-actual` option:

```
Filer> wrfile /etc/exports:

/vol/whatevernamehere/ -actual=/vol/vol2,sec=sys,rw
CTRL+C,

Filer> exportfs -a.
```

This exports the resource vol2, but clients can mount it with the name `/vol/whatevernamehere/`.

4.4 ACCESS CACHE

Access cache provides fast resolution in synchronous code paths of whether an access result is known or not; if it is known, determination of whether a client has read and/or write access; and whether or not root ID translation is in effect for that client. The access cache has the following effects:

- Rapid restart after crash and reboot or failover because of persistent cache
- Caching of positive, negative, and mixed positive and negative results
- Fast insert into the cache
- Bounded search time
- Continuous cache maintenance in the background
- Optimized handling of subnets

The exports cache is an independent module of the redesigned exports code. It is used during NFS operations to determine whether or not a client's access to an exported portion of the file system has recently been determined, and if so, what that result is . It is also used during mount operations. Clients are identified exclusively by their network addresses. The cache is a reservoir of access results. Individual results are accessed by a number of keys. Some keys are used to divide the system-wide access resolution cache into rule-specific caches. The remaining keys are used to locate individual cached access results within a rule-specific cache (rule cache). Ipv6 addresses are supported in IC.1 implementation.

The complete set of keys for lookup is:

- vFiler™ ID
- Client address
- Export point
- Access security flavor

The access results cached for each combination of keys includes whether each of the following is positive, negative, in progress, delayed, or unknown:

- Read access
- Write access
- Root access

Read and write access determine whether a client has read and/or write access to the exported data. Root access specifies whether an incoming root credential from a particular client is left as it is or is translated to `uid anon`. These properties have no meaning to the cache, which merely stores them for use by the caller in access determinations.

HOW ACCESS CACHE WORKS

When an NFS client attempts to access an exported share, Data ONTAP must grant or deny access to the NFS client. The following algorithm is used to determine access to a client:

- A first pass evaluation in no-wait context matches client against IP addresses in the export rule, and host names specified in the export rule that have been resolved to IP addresses, by the forward mapping mechanism.
- In the second pass evaluation, reverse name lookup is done on the NFS client's IP and then compared with host information in the export rules.
- Finally, a third evaluation for group membership includes netgroup checks and subdomain membership checks.

Access Cache Prior to Data ONTAP 7.2.4

- There is one access cache per export rule.
- Access permissions are cumulative.
- The access cache hit path is stored in 256 hash chains.
- Multiple access caches exist for different security flavors, even for the same export.
- All access cache information is stored by IP address.
- All name resolutions happen on the mount assist thread.
- Access cache information is not persistent during system reboots, cluster takeovers, and giveback operations.

Access Cache in Data ONTAP 7.2.4

- Each client request is stored in a data structure radix tree.
- Cache data corresponding to subnets and IP addresses can be stored.
- Facilitates queuing of request in case of cache miss.
- Uses export thread pool for processing cache misses.
- Access permissions are cumulative; when all requested attributes are found, cache lookup is done and the client request is redriven.

Advantages of Access Cache

- The cache contents are pushed to disk every 15 minutes.
- Access cache nodes handle each request separately.
- Upon reboot or failover, cached information is restored and inserted back in the cache.
- Lookup time is reduced.

BOUNCING EXPORT ENTRIES DIFFERENTLY THAN SPECIFIED IN /ETC/EXPORTS

To export entries that are not in the `/etc/exports` file, or to export entries with access restrictions that are different from those specified in the `/etc/exports` file, you make use of the `exportfs` command with switch `-c`:

```
exportfs -c host path_name [ro|rw|root]
```

- `Host` is the dotted IP address of the client to check in the access cache.
- `Path_name` is the path name to a resource specified in the `/etc/exports` file.
- `[ro | rw | root]` checks for read-only (`ro`), read-write (`rw`), or `root` permissions. If this parameter is not provided, the `exportfs -c` command checks to see if the client has mount permissions to export.

Remember that the `-actual` parameter allows server-side aliasing of path names if the storage has been moved and clients have not yet been updated to the new path. To check the access path, use `-c`:

```
> exportfs -c 192.168.208.51 /vol/vfilers/vf19
exportfs: 192.168.208.51 has mount access to /vol/vfilers/vf19
```

Note: RPC requests come across the wire with IP addresses and not machine names, so to test the algorithm to determine access, it is best to pass in the IP address, not the host name.

Every NFS request that comes across the wire results in an access check. To reduce name resolution calls, an access cache is created to store a client's access rights for an export. However, if the export rule is reloaded, the access cache is reinitialized.

Table 1) Access cache entries.

Option	Description

Nfs.export.neg.timeout	Specifies how long a negated entry lives.
Nfs.export.pos.timeout	Specifies how long a normal entry lives. (If a client's IP is changed and there is still a long time before an expunging occurs, the cache entry can be flushed from the access cache.)
Exportfs –f	If a flush is tried while an access cache entry is being used, the following message appears: exportfs: Cache in use, please try again If this message persists and the cache has to be flushed for new connections, then reloading the exports forces the cache to be reloaded.
Exportfs –a	To export all file system paths specified in the /etc/exports file, enter the exportfs -a command.

4.5 BASIC NFS VERSUS NFS WITH NIS

Network Information Service (NIS) provides a simple network lookup service consisting of databases and processes. It was formerly known as Sun Yellow Pages (YP). The functionality remains the same. Its purpose is to provide information that has to be known throughout the network, to all machines on the network. Information likely to be distributed by NIS is:

- Login names, passwords, home directories (`/etc/passwd`)
- Group information (`/etc/group`)
- Host names and IP numbers (`/etc/hosts`)
- Netgroups

With Data ONTAP 7.1 and later, the storage system is capable of becoming an NIS slave. Like Domain Name System (DNS), NIS enables you to centrally maintain host information. NIS provides two methods for system host name resolution:

- Using a makefile master on the NIS server, which creates an `/etc/hosts` file and copies it to the storage system's default volume for local host name lookup
- Using host map, maintained as a database on the NIS server, which the storage system queries in a host lookup request across the network

Network groups can also be stored in a network information services, such as LDAP, NIS, or NIS+ (in NIS compatibility mode only).

- `/etc/nsswitch.conf file`
  Determines the order in which authentication occurs

```
fas960c-svl03*> rdfile /etc/nsswitch.conf
#Auto-generated by setup Tue Apr  1 00:23:53 GMT 2008
hosts: files        nis     dns     /etc/hosts  /etc/nsswitch.conf
/etc/resolv.conf
passwd: files       nis     ldap
netgroup: files     nis  ldap
group: files        nis     ldap
shadow: files       nis
```

Both Network File System (NFS) and Network Information System (NIS) are client/server applications, which means that they sit at the top layer of the protocol stack and use External Data Representation (XDR) and Remote Procedure Calls (RPC) services.

In addition to NFS servers, NIS servers are typically used in large distributed networks.

A major problem in running a distributed computing environment is maintaining separate copies of common configuration files, such as the `passwd`, `group`, and `hosts` files. Ideally, the network should be consistent in its configuration so that users don't have to worry about where they have accounts or if they will be able to

find a new machine on the network. Preserving consistency, however, means that every change to one of these common files must be propagated to every host on the network, which is difficult and not scalable. The NIS addresses these problems. It is a distributed system that replaces copies of commonly replicated configuration files with a centralized management facility. Machines that are using NIS retrieve information from one centralized database that maintains updates and propagates changes to the rest of the network. Files that are generally the same on all hosts in a network, such as `/etc/passwd` and `/etc/hosts`, reside on the NIS database.

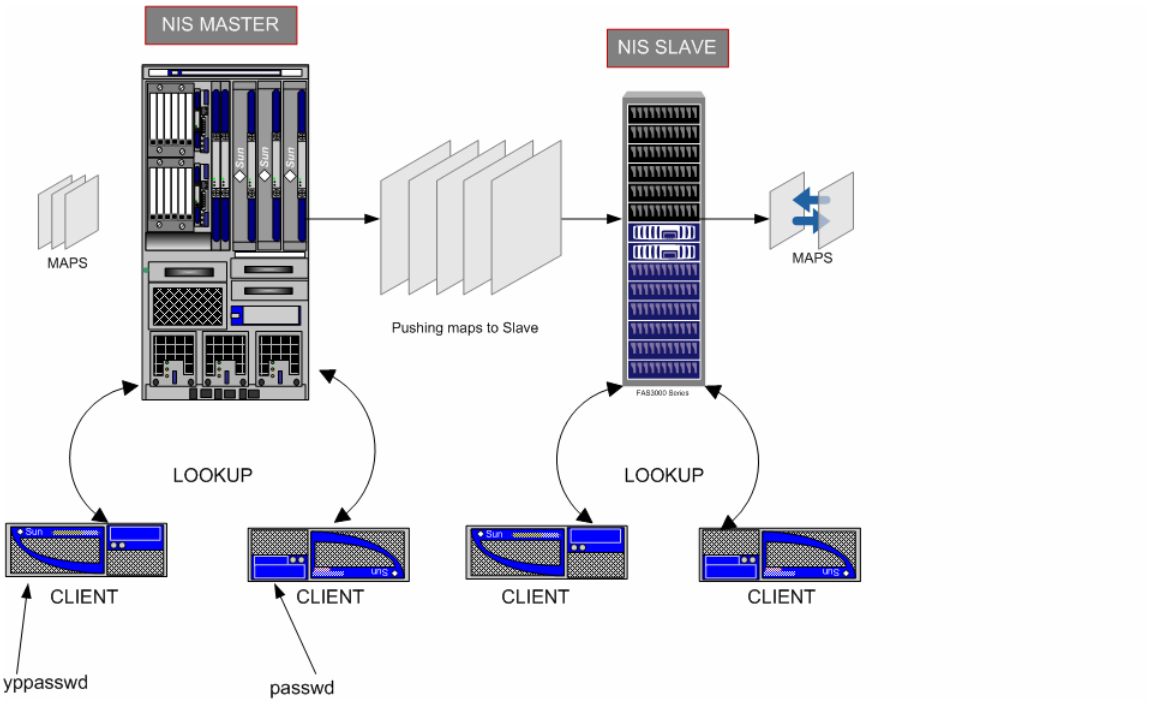

Figure 1) <<Add figure title.>>

Typically, NIS is a lookup service that NFS and mount depend on. It performs host lookup from the export maps, reverse lookups, and so on.

Network Information Service (NIS) is used to resolve:

- Host name to IP
- IP to host name
- Group

To configure NIS:

- `nis.enable` on (The default is off.)
- `nis.domainname domain_name`
- `nis.servers server_name_or_ip, server…`

Use NIS information to display configuration information about the system.

Note: The storage system works with an NIS+ server only if it is set to NIS compatibility mode. NIS+ does not netgroup.byhost maps and so on, so NIS+ lookup is a bottleneck, because the netgroup membership check involves multiple packet exchanges between the system and NIS+. Also, the system cannot act as an NIS slave to the NIS+ server; the `nis.slave` option has no effect.

Table 2) NIS daemon entries.

Daemon Name	Purpose
Portmap	The foundation RPC daemon upon which NIS runs.
yppasswdd	Lets users change their passwords on the NIS server from NIS clients.
Ypmatch key	Prints every value in the NIS map mapname whose key matches the keys again.
Ypwhich	If NIS is enabled, prints the name of the current NIS server.
Ypgroup	Displays the group file entries that have been locally cached from the NIS server.
Ypcat name	Prints all of the values in the NIS map that is provided.
Getent	Prints the unencrypted password.

Unlike `ypmatch`, `getent` doesn't provide an encrypted password when run on an NIS server, it just provides the user's entry in the `/etc/swd` file. On an NIS client, the results are identical, with both showing the encrypted password.

```
[root@ibmx335-svl02 ~]#getent passwd net
nisuser:x:504:100::/home/net:/bin/bash
```

4.6 DATA ONTAP ACTING AS NIS SLAVE

Data ONTAP 7.1 or later can be configured as an NIS slave. The slave can be turned on or off by using the following option: `options nis.slave.enable on | off`. Once the maps are downloaded by the slave, all NIS requests are serviced by using the downloaded maps; no NIS requests go to the NIS servers. If the slave is disabled, the system reverts back to the client behavior. The slave has two major functions:

- **Download the maps from the NIS master:** The NIS slave checks every 45 minutes with the master server for updates. If there are updates, they are downloaded.
- **Service YPPUSH requests**.

The following list describes Data ONTAP as a slave:

- The NIS slave does a periodic check to determine whether any maps have been updated. This check happens every 45 minutes. This fairly small time limit was chosen to keep current with the frequently changing `passwd` map.
- At the scheduled time interval, the NIS slave checks to determine whether any NIS map has updates, and if so pulls the updated map from the NIS server. During this period, when the NIS slave is determining whether a map update is required, all local NIS maps are inaccessible for NIS queries. In other words, even if only one map needs to be updated, all the other maps are effectively inaccessible for local NIS queries.
- The NIS slave is also able to accept YPPUSH notifications from the NIS master.
- If the NIS slave cannot satisfy a NIS query for any reason, all NIS queries get routed through the NIS client to the NIS server. So, during system boot, when the NIS slave hasn't been populated yet, the NIS client sends out NIS queries over the network to the NIS server. Note that the NIS maps are stored on the disk, so during any subsequent boots, the NIS slave downloads maps only if they have changed.
- NIS slave maps are cached on disk. Even if the NIS slave is enabled, NIS calls are blocking as fio operations to read from disk can block.
- NIS queries are serviced by the NIS database. Data ONTAP maintains an 8K NIS database in memory. Therefore, although the NIS slave maintains the maps on disk, it is possible that many of the NIS queries are from cache hits in the 8K NIS database cache or WAFL® file system buffers.
- The NIS slave is a new feature in the TsingTao release.

- No performance characterizations of the NIS slave have been done. The ramifications of the NIS slave lockdown when individual map update is being done are not well understood.
- All other NIS/YP requests are denied. If the system is configured as a slave on the NIS master, when the maps on the master are updated, the administrator has the option to notify all the slaves.
- The downloaded maps are stored under `/etc/yp/<NIS_domain_name>/`. For the slave to function correctly, there must be sufficient space on the root volume of the system to download maps. The amount of space needed depends on the size of the maps. It takes almost the same size as maps on the NIS server.

The maps are stored in a database file; you can verify the data in each of the map database files by using `db_dump185 –p <map_name>.`

```
 [root@ibmx335-svl13 etc]# ls -l  yp/file
-rw-r--r--  1 root root 16384 May  7 03:11 netgroup.byhost.db
-rw-r--r--  1 root root 16384 May  7 03:11 netgroup.db

 [root@ibmx335-svl13 file]# db_dump185 -p netgroup.byhost.db
format=print
type=btree
db_lorder=1234
db_pagesize=8192
HEADER=END
YP_DOMAIN_NAME
iop.eng.netapp.com
YP_LAST_MODIFIED
1209981194
YP_MAP_NAME
netgroup.byhost
YP_MASTER_NAME
localhost
ibmx335-svl02.iop.eng.netapp.com
all_hosts,untrusted_hosts
ibmx335-svl12.iop.eng.netapp.com
all_hosts,trusted_hosts

[root@ibmx335-svl13 file]# db_dump185 -p netgroup.db
format=print
type=btree
db_lorder=1234
db_pagesize=8192
HEADER=END
YP_DOMAIN_NAME
iop.eng.netapp.com
```

```
YP_LAST_MODIFIED
1209981194
YP_MAP_NAME
netgroup
YP_MASTER_NAME
localhost
all_hosts
trusted_hosts, untrusted_hosts
trusted_hosts
(ibmx335-svl12.iop.eng.netapp.com,,)
untrusted_hosts
(ibmx335-svl02.iop.eng.netapp.com,,)

[root@ibmx335-svl13 etc]# cat nsswitch.conf
#Auto-generated by setup Tue Apr 15 03:46:43 PDT 2008
hosts: files       nis     dns
passwd: files      nis     ldap
netgroup: files    nis  ldap
group: files       nis     ldap
shadow: files      nis

fas960c-svl03*> options nis
nis.domainname               iop.eng.netapp.com (value might be overwritten in
takeover)
nis.enable                   on         (value might be overwritten in
takeover)
nis.group_update.enable      on         (value might be overwritten in
takeover)
nis.group_update_schedule    24         (value might be overwritten in
takeover)
nis.netgroup.domain_search.enable on         (value might be overwritten in
takeover)
nis.netgroup.legacy_nisdomain_search.enable on
nis.servers                  172.17.39.83 (value might be overwritten in
takeover)

fas960c-svl03*> nis info
NIS domain is iop.eng.netapp.com

NIS group cache has been enabled
The group cache was last updated on Wed May  7 04:59:31 GMT 2008
IP Address     Type  State  Bound        Last Polled            Client calls
Became Active
```

```
--------------------------------------------------------------------------------
----------------

172.17.39.83     PREF  ALIVE   YES    Wed May  7 05:07:31 GMT 2008         0
Wed May  7 04:59:31 GMT 2008

NIS Performance Statistics:

Number of YP Lookups: 5

Total time spent in YP Lookups: 11 ms, 995 us

Number of network re-transmissions: 0

Minimum time spent in a YP Lookup: 0 ms, 0 us

Maximum time spent in a YP Lookup: 6 ms, 3 us

Average time spent in YP Lookups: 2 ms, 399 us

3 Most Recent Lookups:

 [0] Lookup time: 6 ms, 3 us     Number of network re-transmissions: 0

 [1] Lookup time: 5 ms, 3 us     Number of network re-transmissions: 0

 [2] Lookup time: 0 ms, 989 us   Number of network re-transmissions: 0
```

4.7 EXPORTING RESOURCES BY USING NETGROUPS

When exporting a resource, if you do not want to list the names of individual hosts as targets, you can specify a predefined netgroup as the target. You can create netgroups in the `/etc/netgroup` file on the system or in your `netgroup.byhost` NIS map.

RELATIONSHIP BETWEEN /ETC/NETGROUP AND THE NETGROUP.BYHOST MAP

Information in the `/etc/netgroup` file is not linked to the `netgroup.byhost` NIS map. For example, the NIS master does not use the `/etc/netgroup` file as input for building the `netgroup.byhost` map. You can, however, copy the information from the NIS map to the `/etc/netgroup` file so that the system can use the `/etc/netgroup` file whenever the NIS server is not available.

RESTRICTIONS ON THE /ETC/NETGROUP FILE

Each line in the `/etc/netgroup` file can contain up to 4,096 characters. Make sure that you stay within this constraint when adding entries to this file.

FORMAT OF A NETGROUP ENTRY

Each entry in the `/etc/netgroup` file contains the netgroup name and a member list. You can specify either of the following items in the member list:

- Another netgroup name
- An entry in the form (*host_name*, *user_name*, *domain_name*)
    - *host_name* is a fully qualified name, if the host is not in the local domain.
    - *user_name* is ignored. Leave this field empty.
    - *domain_name* is the name of the domain in which the netgroup is valid. Use the local domain name or leave this field empty.

Note: Even if you leave a field empty, you must include the comma to keep the entry syntax correct.

EXAMPLE OF A NETGROUP FILE

The following `/etc/netgroup` file contains three netgroups:

- `trusted-hosts (host1,,) (host2,,)`
- `untrusted-hosts (host3,,) (host4,,) (host5,,)`
- `all-hosts trusted-hosts untrusted-hosts`

fas960c-svl03*> rdfile /etc/netgroup

trusted_hosts (ibmx335-svl12.iop.eng.netapp.com,,)

untrusted_hosts (ibmx335-svl02.iop.eng.netapp.com,,)

all_hosts trusted_hosts, untrusted_hosts

In an exports entry, you can specify the `trusted-hosts`, `untrusted-hosts`, or `all-hosts` netgroup as the export target with the `rw`, `ro`, and `root` options.

EXPORTING A RESOURCE TO A NETGROUP

To export a resource to a netgroup, complete the following steps:

```
/vol/vol0 -root=trusted-hosts,rw=trusted-hosts
```

```
/vol/vol1 -rw=untrusted-hosts
```

Result: In Data ONTAP 7.0.5, hosts that belong to the `trusted-hosts` netgroup can mount the `/vol/vol0` volume as read-write. The root user on any host that belongs to the `trusted-hosts` netgroup can access the `/vol/vol0` volume as root. Hosts that belong to the `untrusted-hosts` netgroup can mount `/vol/vol1`:

```
/vol/vol0 -root=trusted-hosts,rw=trusted-hosts
```

```
/vol/vol1 -rw=untrusted-hosts
```

Result: In Data ONTAP 7.2.4, hosts that belong to the `trusted-hosts` netgroup can mount the `/vol/vol0` volume as read-write. The root user on any host that belongs to the `trusted-hosts` netgroup can access the `/vol/vol0` volume as root. Hosts that belong to the untrusted-hosts netgroup can mount `/vol/vol1`.

Enter the following in the `/etc/exports` file:

```
path_name -rw=netgroup_name[:netgroup_name:...]
```

```
HAVING NETGROUPS as root=@
```

```
fas960c-svl03*> rdfile /etc/netgroup
trusted_hosts   (ibmx335-svl12.iop.eng.netapp.com,,)
untrusted_hosts (ibmx335-svl02.iop.eng.netapp.com,,)
all_hosts       trusted_hosts, untrusted_hosts
```

```
fas960c-svl03*> rdfile /etc/exports
/vol/vol0       -sec=sys,root=@,nosuid
/vol/vol1       -sec=sys,rw=trusted-hosts
```

In Data ONTAP 7.0.5, all hosts are allowed to mount `/vol/vol0` read only; hosts that belong to the `trusted-hosts` netgroup are allowed to access `/vol/vol1` read-only.

In Data ONTAP 7.2.4, any hosts are allowed to access `/vol/vol0` as “NFSNOBODY,” and `/vol/vol1` is accessible to trusted-hosts. Any other hosts trying to access `/vol/vol1` get

```
mount: mount to NFS server '172.17.44.42' failed: timed out (retrying).
mount: mount to NFS server '172.17.44.42' failed: timed out (retrying).
```