NAME

exports **-** define remote mount points for NFS mount requests

SYNOPSIS

exports

DESCRIPTION

The **exports** file specifies remote mount points for the NFS mount protocol per the NFS server specification; see *Network File System Protocol Specification,* RFC1094, Appendix A and *NFS: Network File System Version 3 Specification,* Appendix I.

Each line in the file (other than comment lines that begin with a #) specifies the mount point(s) and export flags within one local server file system or the NFSv4 tree root for one or more hosts. A long line may be split over several lines by ending all but the last line with a backslash (‘\’). A host may be specified only once for each local file or the NFSv4 tree root on the server and there may be only one default entry for each server file system that applies to all other hosts. The latter exports the file system to the "world" and should be used only when the file system contains public information.

In a mount entry, the first field(s) specify the directory path(s) within a server file system that can be mounted on by the corresponding client(s). There are three forms of this specification. The first is to list all mount points as absolute directory paths separated by whitespace. This list of directory paths should be considered an "administrative control", since it is only enforced by the mountd(8) daemon and not the kernel. As such, it only applies to NFSv2 and NFSv3 mounts and only with respect to the client’s use of the mount protocol. The second is to specify the pathname of the root of the file system followed by the **-alldirs** flag; this form allows the host(s) to mount at any point within the file system, including regular files if the **-r** option is used on mountd(8). Because NFSv4 does not use the mount protocol, the "administrative controls" are not applied and all directories within this server file system are mountable via NFSv4 even if the **-alldirs** flag has not been specified. The third form has the string ‘‘V4:’’ followed by a single absolute path name, to specify the NFSv4 tree root. This line does not export any file system, but simply marks where the root of the server’s directory tree is for NFSv4 clients. The exported file systems for NFSv4 are specified via the other lines in the **exports** file in the same way as for NFSv2 and NFSv3. The pathnames must not have any symbolic links in them and should not have any "." or ".." components. Mount points for a file system may appear on multiple lines each with different sets of hosts and export options.

The second component of a line specifies how the file system is to be exported to the host set. The option flags specify whether the file system is exported read-only or read-write and how the client UID is mapped to user credentials on the server. For the NFSv4 tree root, the only options that can be specified in this section are ones related to security: **-sec**, **-tls**, **-tlscert** and **-tlscertuser**.

Export options are specified as follows:

-maproot=**user** The credential of the specified user is used for remote access by root. The credential includes all the groups to which the user is a member on the local machine (see id(1)). The user may be specified by name or number. The user string may be quoted, or use backslash escaping.

-maproot=**user:group1:group2:...** The colon separated list is used to specify the precise credential to be used for remote access by root. The elements of the list may be either names or numbers. Note that user: should be used to distinguish a credential containing no groups from a complete credential for that user. The group names may be quoted, or use backslash escaping.

-mapall=**user** or **-mapall**=**user:group1:group2:...** specifies a mapping for all client UIDs (including root) using the same semantics as **-maproot**.

The option **-r** is a synonym for **-maproot** in an effort to be backward compatible with older export file formats.

In the absence of **-maproot** and **-mapall** options, remote accesses by root will result in using a credential of 65534:65533. All other users will be mapped to their remote credential. If a **-maproot** option is given, remote access by root will be mapped to that credential instead of 65534:65533. If a **-mapall** option is given, all users (including root) will be mapped to that credential in place of their own.

-sec=**flavor1:flavor2...** specifies a colon separated list of acceptable security flavors to be used for remote access. Supported security flavors are sys, krb5, krb5i and krb5p. If multiple flavors are listed, they should be ordered with the most preferred flavor first. If this option is not present, the default security flavor list of just sys is used.

The **-ro** option specifies that the file system should be exported read-only (default read/write). The option **-o** is a synonym for **-ro** in an effort to be backward compatible with older export file formats.

WebNFS exports strictly according to the spec (RFC 2054 and RFC 2055) can be done with the **-public** flag. However, this flag in itself allows r/w access to all files in the file system, not requiring reserved ports and not remapping UIDs. It is only provided to conform to the spec, and should normally not be used. For a WebNFS export, use the **-webnfs** flag, which implies **-public**, **-mapall**=**nobody** and **-ro**. Note that only one file system can be WebNFS exported on a server.

A **-index**=*file* option can be used to specify a file whose handle will be returned if a directory is looked up using the public filehandle (WebNFS). This is to mimic the behavior of URLs. If no **-index** option is specified, a directory filehandle will be returned as usual. The **-index** option only makes sense in combination with the **-public** or **-webnfs** flags.

The **-tls**, **-tlscert** and **-tlscertuser** export options are used to require the client to use TLS for the mount(s) per RFC NNNN. For NFS mounts using TLS to work, rpc.tlsservd(8) must be running on the server.

-tls requires that the client use TLS.
-tlscert requires that the client use TLS and provide a verifiable X.509 certificate during TLS handshake.
-tlscertuser requires that the client use TLS and provide a verifiable X.509 certificate. The otherName component of the certificate’s subjAltName must have a an OID of 1.3.6.1.4.1.2238.1.1.1 and a UTF8 string of the form "user@domain". "user@domain" will be translated to the credentials of the specified user in the same manner as nfsuserd(8), where "user" is normally a username is the server’s password database and "domain" is the DNS domain name for the server. All RPCs will be performed using these credentials instead of the ones in the RPC header in a manner similar to **-mapall**=**user**.

If none of these three flags are specified, TLS mounts are permitted but not required.

Specifying the **-quiet** option will inhibit some of the syslog diagnostics for bad lines in */etc/exports*. This can be useful to avoid annoying error messages for known possible problems (see *EXAMPLES* below).

The third component of a line specifies the host set to which the line applies. The set may be specified in three ways. The first way is to list the host name(s) separated by white space. (Standard Internet "dot" addresses may be used in place of names.) The second way is to specify a "netgroup" as defined in the *netgroup* file (see netgroup(5)). The third way is to specify an Internet subnetwork using a network and network mask that is defined as the set of all hosts with addresses within the subnetwork. This latter approach requires less overhead within the kernel and is recommended for cases where the export line refers to a large number of clients within an administrative subnet.

The first two cases are specified by simply listing the name(s) separated by whitespace. All names are checked to see if they are "netgroup" names first and are assumed to be hostnames otherwise. Using the full domain specification for a hostname can normally circumvent the problem of a host that has the same name as a netgroup. The third case is specified by the flag **-network**=**netname**[/*prefixlength*] and optionally **-mask**=**netmask**. The netmask may be specified either by attaching a *prefixlength* to the **-network** option, or by using a separate **-mask** option. If the mask is not specified, it will default to the historical mask for that network class (A, B, or C; see inet(4)). This usage is deprecated, and will elicit a warning log message. See the *EXAMPLES* section below.

Scoped IPv6 address must carry scope identifier as documented in inet6(4). For example, "fe80::%re2/10" is used to specify fe80::/10 on re2 interface.

For the third form which specifies the NFSv4 tree root, the directory path specifies the location within the server's file system tree which is the root of the NFSv4 tree. There can only be one NFSv4 root directory per server. As such, all entries of this form must specify the same directory path. For file systems other than ZFS, this location can be any directory and does not need to be within an exported file system. If it is not in an exported file system, a very limited set of operations are permitted, so that an NFSv4 client can traverse the tree to an exported file system. Although parts of the NFSv4 tree can be non-exported, the entire NFSv4 tree must consist of local file systems capable of being exported via NFS. All ZFS file systems in the subtree below the NFSv4 tree root must be exported. NFSv4 does not use the mount protocol and does permit clients to cross server mount point boundaries, although not all clients are capable of crossing the mount points.

The **-sec** option on these line(s) specifies what security flavors may be used for NFSv4 operations that do not use file handles. Since these operations (SetClientID, SetClientIDConfirm, Renew, DelegPurge and ReleaseLockOnwer) allocate/modify state in the server, it is possible to restrict some clients to the use of the krb5[ip] security flavors, via this option. See the *EXAMPLES* section below. This third form is meaningless for NFSv2 and NFSv3 and is ignored for them.

The mountd(8) utility can be made to re-read the **exports** file by sending it a hangup signal as follows:

```
/etc/rc.d/mountd reload
```

After sending the SIGHUP, check the syslogd(8) output to see whether mountd(8) logged any parsing errors in the **exports** file.

FILES

/etc/exports the default remote mount-point file

EXAMPLES

Given that */usr*, */u*, */a* and */u2* are local file system mount points, let's consider the following example:

```
/usr /usr/local -maproot=0:10 friends
/usr -maproot=daemon grumpy.cis.uoguelph.ca 131.104.48.16
/usr -ro -mapall=nobody
/u -maproot=bin: -network 131.104.48 -mask 255.255.255.0
/a -network 192.168.0/24
/a -network 3ffe:1ce1:1:fe80::/64
/u2 -maproot=root friends
/u2 -alldirs -network cis-net -mask cis-mask
/cdrom -alldirs,quiet,ro -network 192.168.33.0 -mask 255.255.255.0
/private -sec=krb5i
```

```
/secret -sec=krb5p
V4: /        -sec=krb5:krb5i:krb5p -network 131.104.48 -mask 255.255.255.0
V4: /        -sec=sys:krb5:krb5i:krb5p grumpy.cis.uoguelph.ca
```

The file systems rooted at */usr* and */usr/local* are exported to hosts within the "friends" network group with users mapped to their remote credentials and root mapped to UID 0 and group 10. They are exported read-write and the hosts in "friends".

The file system rooted at */usr* is exported to *131.104.48.16* and *grumpy.cis.uoguelph.ca* with users mapped to their remote credentials and root mapped to the user and groups associated with "daemon"; it is exported to the rest of the world as read-only with all users mapped to the user and groups associated with "nobody".

The file system rooted at */u* is exported to all hosts on the subnetwork *131.104.48* with root mapped to the UID for "bin" and with no group access.

The file system rooted at */u2* is exported to the hosts in "friends" with root mapped to UID and groups associated with "root"; it is exported to all hosts on network "cis-net" allowing mounts at any directory within /u2.

The file system rooted at */a* is exported to the network 192.168.0.0, with a netmask of 255.255.255.0. However, the netmask length in the entry for */a* is not specified through a **-mask** option, but through the /*prefix* notation.

The file system rooted at */a* is also exported to the IPv6 network 3ffe:1ce1:1:fe80:: address, using the upper 64 bits as the prefix. Note that, unlike with IPv4 network addresses, the specified network address must be complete, and not just contain the upper bits. With IPv6 addresses, the **-mask** option must not be used.

The file system rooted at */cdrom* will be exported read-only to the entire network 192.168.33.0/24, including all its subdirectories. Since */cdrom* is the conventional mountpoint for a CD-ROM device, this export will fail if no CD-ROM medium is currently mounted there since that line would then attempt to export a subdirectory of the root file system with the **-alldirs** option which is not allowed. The **-quiet** option will then suppress the error message for this condition that would normally be syslogged. As soon as an actual CD-ROM is going to be mounted, mount(8) will notify mountd(8) about this situation, and the */cdrom* file system will be exported as intended. Note that without using the **-alldirs** option, the export would always succeed. While there is no CD-ROM medium mounted under */cdrom*, it would export the (normally empty) directory */cdrom* of the root file system instead.

The file system rooted at */private* will be exported using Kerberos 5 authentication and will require integrity protected messages for all accesses. The file system rooted at */secret* will also be exported using Kerberos 5 authentication and all messages used to access it will be encrypted.

For the experimental server, the NFSv4 tree is rooted at ‘‘/’’, and any client within the 131.104.48 subnet is permitted to perform NFSv4 state operations on the server, so long as valid Kerberos credentials are provided. The machine grumpy.cis.uoguelph.ca is permitted to perform NFSv4 state operations on the server using AUTH_SYS credentials, as well as Kerberos ones.

In the following example some directories are exported as NFSv3 and NFSv4:

```
V4: /wingsdl/nfsv4
/wingsdl/nfsv4/usr-ports -maproot=root -network 172.16.0.0 -mask 255.255.0.0
/wingsdl/nfsv4/clasper   -maproot=root clasper
```

Only one V4: line is needed or allowed to declare where NFSv4 is rooted. The other lines declare specific exported directories with their absolute paths given in /etc/exports.

The exported directories’ paths are used for both v3 and v4. However, they are interpreted differently for v3 and v4. A client mount command for usr-ports would use the server-absolute name when using nfsv3:

```
mount server:/wingsdl/nfsv4/usr-ports /mnt/tmp
```

A mount command using NFSv4 would use the path relative to the NFSv4 root:

```
mount server:/usr-ports /mnt/tmp
```

This also differentiates which version you want if the client can do both v3 and v4. The former will only ever do a v3 mount and the latter will only ever do a v4 mount.

Note that due to different mount behavior between NFSv3 and NFSv4 a NFSv4 mount request for a directory that the client does not have permission for will succeed and read/write access will fail afterwards, whereas NFSv3 rejects the mount request.

SEE ALSO

nfsv4(4), netgroup(5), mountd(8), nfsd(8), rpc.tlsservd(8), showmount(8)

STANDARDS

The implementation is based on the following documents:

- *Network File System Protocol Specification, Appendix A, RFC 1094*.

- *NFS: Network File System Version 3, Appendix I, RFC 1813*.

- *Towards Remote Procedure Call Encryption By Default, RFC nnnn*.

BUGS

The export options are tied to the local mount points in the kernel and must be non-contradictory for any exported subdirectory of the local server mount point. It is recommended that all exported directories within the same server file system be specified on adjacent lines going down the tree. You cannot specify a hostname that is also the name of a netgroup. Specifying the full domain specification for a hostname can normally circumvent the problem.