NAME

share_nfs – make local NFS filesystems available for mounting by remote systems

SYNOPSIS

share [–**F nfs**] [–**o** *specific_options*] [–**d** *description*] *pathname*

DESCRIPTION

The **share** command makes local filesystems available for mounting by remote systems.

If no argument is specified, then **share** displays all filesystems currently shared, including NFS filesystems and filesystems shared through other distributed file system packages.

OPTIONS

–**o** *specific_options*
: Specify options in a comma-separated list of keywords and attribute-value-assertions for interpretation by the file-system-type-specific command.

  If *specific_options* is not specified, then by default sharing will be read-write to all clients.

  specific_options can be any combination of the following:

  rw Sharing will be read-write to all clients.

  rw=*client*[**:***client*]. . .
  : Sharing will be read-write to the listed clients; overrides the **ro** suboption for the clients specified. Netgroup names may be used in place of client names unless the list is used to override an **ro** option.

  ro Sharing will be read-only to all clients.

  ro=*client*[**:***client*]. . .
  : Sharing will be read-only to the listed clients; overrides the **rw** suboption for the clients specified. Netgroup names may be used in place of client names unless the list is used to override an **rw** option.

  anon=*uid*
  : Set *uid* to be the effective user ID of unauthenticated users (**AUTH_DES** or **AUTH_KERB** authentication), or to be root if **AUTH_UNIX** authentication is used. By default, unknown users are given the effective user ID **UID_NOBODY**. If *uid* is set to –**1**, access is denied.

  root=*host*[**:***host*] . . .
  : Only root users from the specified hosts will have root access. By default, no host has root access.

  secure Clients must use the **AUTH_DES** authentication of RPC to be authenticated. **AUTH_UNIX** authentication is the default. See the **anon=***uid* option (above) for information about how unauthenticated requests are handled.

  nosuid By default, clients are allowed to create files on the shared file system with the setuid or setgid mode enabled. Specifying **nosuid** causes the server file system to silently ignore any attempt to enable the setuid or setgid mode bits.

aclok Allows the NFS server to do access control for NFS Version 2 clients (running SunOS 2.4 or earlier). When **aclok** is set on the server, maximal access is given to all clients. For example, with **aclok** set, if anyone has read permissions, then everyone does. If **aclok** is not set, minimal access is given to all clients.

kerberos
Clients must use the **AUTH_KERB** authentication of RPC to be authenticated. **AUTH_UNIX** authentication is the default. See the **anon=***uid* option (above) for information about how unauthenticated requests are handled.

–d *description* Provide a comment that describes the filesystem to be shared.

pathname Specify the pathname of the filesystem to be shared.

FILES

/etc/dfs/fstypes list of system types, NFS by default
/etc/dfs/sharetab system record of shared file systems

SEE ALSO

unshare(1M), **share**(1M)

NOTES

The command will fail if both **ro** and **rw** are specified for the same client name. If the same client name exists in both the **ro=** and **rw=** lists, the **rw** will override the **ro**, giving read/write access to the client specified.

ro=, **rw=**, and **root=** are guaranteed to work over UDP but may not work over other transport providers.

If a filesystem is shared with an **ro=** list and a **root=** list, any host that is on the **root=** list will be given only read-only access, regardless of whether that host is specified in the **ro=** list, unless **rw** is declared as the default, or the host is mentioned in an **rw=** list. The same is true if the filesystem is shared with **ro** as the default. For example, the following **share** commands will give read-only permissions to **hostb**:

share –F nfs –o ro=hosta,root=hostb /var

share –F nfs –o ro,root=hostb /var

The following will give read/write permissions to **hostb**:

share –F nfs –o ro=hosta,rw=hostb,root=hostb /var

share –F nfs –o root=hostb /var

If the filesystem being shared is a symbolic link to a valid pathname, the canonical path (the path which the symbolic link follows) will be shared.

For example, if **/export/foo** is a symbolic link to **/export/bar** (**/export/foo -> /export/bar**), the following **share** command will result in **/export/bar** as the shared pathname (and not **/export/foo**).

example# share -F nfs /export/foo

Note that an NFS mount of **server:/export/foo** will result in **server:/export/bar** really being mounted.

This line will share the **/disk** file system read-only at boot time.

share –F nfs –o ro /disk

Note that the same command entered from the command line will not share the **/disk** file system unless there is at least one file system entry in the **/etc/dfs/dfstab** file. The **mountd**(1M) and **nfsd**(1M) daemons only run if there is a file system entry in **/etc/dfs/dfstab** when starting or rebooting the system.