NAME

   smount, sumount − Mount or unmount a Shore file system as an NFS file system

SYNOPSIS

   `smount [-rv] [-o options] host:path /path`

   sumount [-vflrm] [-p port] [host:]path

   sumount [-vflrm] [-p port] host:path path

DESCRIPTION

   The Shore server is capable of acting as an NFS (Sun Network File System) server, making a Shore file system appear to be a Unix file system. A Shore file system (or any subtree of its directory hierarchy) may be ''mounted'' into the namespace of any system that supports the client side of the NFS protocol (which includes just about any flavor of Unix these days, as well as many non-Unix systems). To allow Shore to act as an NFS server on a host that also has a standard NFS server, Shore listens on port 2999 (or any other port specified in the configuration file) for NFS requests rather than the standard port of 2049. The standard Unix utility `mount(8)` on most systems has an option that allows you to specify a non-standard port for NFS requests. Unfortunately, this option only controls NFS requests *per se ;* NFS **mount** and **unmount** requests are still sent to the ''standard'' port. The Shore distribution comes with patches for the some versions of `mount(8)` and `umount(8)` to add an option to control the port used for **mount** and **unmount** requests, but for copyright reasons, only patches are included, not the entire programs, so they are only useful to users who have access to the source programs. Moreover, these programs vary substantially among flavors of Unix, so they must be separately patched for each system.

   Shore also includes the programs **smount** and **sumount**. These programs have the advantage that they can be freely distributed under the same terms as the rest of Shore, and since they are considerably simpler than they vendor-supplied versions, they should be easier to port to new systems. However, they lack many of the bells and whistles of the vendor-supplied versions, notably the ability to mount file system types other than NFS or to mount more than one file system at a time.

IMPORTANT NOTE

   On many systems, these commands may only be executed by the super-user, since the `mount(2)` and `unmount(2)` system calls are privileged.

MOUNT

   There must be exactly two non-option arguments. The first argument must have the form

   `host:remote_path`

   where *host* is the name of a host where a Shore server is running (''localhost'' is always acceptable) and *remote_path* is a pathname relative to the database (that is, relative to the root directory of the root volume) served by that server.

   The second argument is the absolute pathname

   `local_path`

   of a node (usually an empty directory) on the local file system. If the request is successful, *local_path* becomes an alias for the node named by *remote_path* in the database served by the indicated server.

MOUNT OPTIONS

   -r   Mount the specified database as a read-only file system.

   -v   Verbose. Indicate exactly what is being done.

   -o options
         Specify a variety of system-specific options, such as buffer sizes and timeouts. The exact options available depend on the host operating system. See `mount(8)` for a description. In general, all options relevant to NFS are supported, as well as one additional option: *mport=nnnn*, where *nnnn* is

the port number (in decimal) where mount and unmount request packets should be sent. The default is compiled into the program (currently 2999).

UMOUNT

Either one or two non-option arguments may be supplied. If two arguments are supplied, they should match the arguments to a corresponding call of **smount.** If only one argument is given, it is interpreted as matching the first or second argument of **smount,** depending on whether it contains a colon. The other argument is reconstructed, if possible, from data stored by **smount.** For example, under SunOS, `mount(8)` (and hence **smount**) records its parameters in `/etc/mtab`.

Sh_umount performs three operations (in this order): it uses a local system call to remove the mount point, it sends an NFS unmount request to the server to tell it that the mount association is removed, and it updates `/etc/mtab` to remove the information. Normally, if any operation fails, subsequent ones are skipped as well. (Note: The standard /etc/umount will probably work ok except that the second step (sending a message to the mountd server) will send the message to the wrong place. Since most mountd servers do a slipshod job of keeping track of who's got what mounted anyhow, you may not notice any difference).

UMOUNT OPTIONS

-v   Verbose. Indicate exactly what is being done.

-f   Force the unmount, even if `/etc/mtab` does not indicate any matching mount. With this option, both arguments must be specified.

-i   Ignore errors; try to perform all three actions even if some of them fail.

-l   Do **not** execute the (local) `unmount(2)` system call.

-r   Do **not** send the NFS unmount request to the remote Shore server.

-m   Do **not** update `/etc/mtab` (or whatever).

-p port

Specify a port to use in contacting the server. If this option is omitted, **sumount** uses the value specified in the corresponding **smount,** if that value was recored in `/etc/mtab` , and otherwise, a compiled-in default (2999).

BUGS

These programs only handle file systems of type NFS. In particular, they cannot do local mounts or mounts of other file system types supported on various systems.

Many features commonly provided by the standard `/etc/mount` are not supported. In particular, the **-a** option (mount all file system listed in a standard place, such as `/etc/fstab`) is not supported , nor is any other usage that attempts to mount more than one file system in one call.

AUTHOR

Marvin Solomon (solomon@cs.wisc.edu)

VERSION

This manual page applies to Version 1.1 of the Shore software.

SPONSORSHIP

The Shore project is sponsored by the Advanced Research Project Agency, ARPA order number 018 (formerly 8230), monitored by the U.S. Army Research Laboratory under contract DAAB07-91-C-Q518.

COPYRIGHT