## NAME

rpc.mountd - NFS mount daemon

## SYNOPSIS

**/usr/sbin/rpc.mountd** [*options*]

## DESCRIPTION

The **rpc.mountd** daemon implements the server side of the NFS MOUNT protocol, an NFS side protocol used by NFS version 2 [RFC1094] and NFS version 3 [RFC1813].

An NFS server maintains a table of local physical file systems that are accessible to NFS clients. Each file system in this table is referred to as an *exported file system*, or *export*, for short.

Each file system in the export table has an access control list. **rpc.mountd** uses these access control lists to determine whether an NFS client is permitted to access a given file system. For details on how to manage your NFS server's export table, see the exports(5) and exportfs(8) man pages.

### Mounting exported NFS File Systems

The NFS MOUNT protocol has several procedures. The most important of these are MNT (mount an export) and UMNT (unmount an export).

A MNT request has two arguments: an explicit argument that contains the pathname of the root directory of the export to be mounted, and an implicit argument that is the sender's IP address.

When receiving a MNT request from an NFS client, **rpc.mountd** checks both the pathname and the sender's IP address against its export table. If the sender is permitted to access the requested export, **rpc.mountd** returns an NFS file handle for the export's root directory to the client. The client can then use the root file handle and NFS LOOKUP requests to navigate the directory structure of the export.

### The rmtab File

The **rpc.mountd** daemon registers every successful MNT request by adding an entry to the */var/lib/nfs/rmtab* file. When receivng a UMNT request from an NFS client, **rpc.mountd** simply removes the matching entry from */var/lib/nfs/rmtab*, as long as the access control list for that export allows that sender to access the export.

Clients can discover the list of file systems an NFS server is currently exporting, or the list of other clients that have mounted its exports, by using the showmount(8) command. showmount(8) uses other procedures in the NFS MOUNT protocol to report information about the server's exported file systems.

Note, however, that there is little to guarantee that the contents of */var/lib/nfs/rmtab* are accurate. A client may continue accessing an export even after invoking UMNT. If the client reboots without sending a UMNT request, stale entries remain for that client in */var/lib/nfs/rmtab*.

## OPTIONS

**-d kind  or  --debug kind**

Turn on debugging. Valid kinds are: all, auth, call, general and parse.

**-F  or  --foreground**

Run in foreground (do not daemonize)

**-h  or  --help**

Display usage message.

**-o num  or  --descriptors num**

Set the limit of the number of open file descriptors to num. The default is to leave the limit unchanged.

**-N mountd-version  or  --no-nfs-version mountd-version**

This option can be used to request that **rpc.mountd** do not offer certain versions of NFS. The current version of **rpc.mountd** can support both NFS version 2, 3 and 4. If the either one of these version should not be offered, **rpc.mountd** must be invoked with the option **--no-nfs-version <vers>**.

**-n  or  --no-tcp**
> Don't advertise TCP for mount.

**-p num  or  -P num  or  --port num**
> Specifies the port number used for RPC listener sockets. If this option is not specified, **rpc.mountd** will try to consult */etc/services*, if gets port succeed, set the same port for all listener socket, otherwise chooses a random ephemeral port for each listener socket.
>
> This option can be used to fix the port value of **rpc.mountd**'s listeners when NFS MOUNT requests must traverse a firewall between clients and servers.

**-H  prog or  --ha-callout prog**
> Specify a high availability callout program. This program receives callouts for all MOUNT and UNMOUNT requests. This allows **rpc.mountd** to be used in a High Availability NFS (HA-NFS) environment.
>
> The callout program is run with 4 arguments. The first is **mount** or **unmount** depending on the reason for the callout. The second will be the name of the client performing the mount. The third will be the path that the client is mounting. The last is the number of concurrent mounts that we believe the client has of that path.
>
> This callout is not needed with 2.6 and later kernels. Instead, mount the nfsd filesystem on */proc/fs/nfsd*.

**-s, --state-directory-path** *directory*
> Specify a directory in which to place statd state information. If this option is not specified the default of */var/lib/nfs* is used.

**-r, --reverse-lookup**
> **rpc.mountd** tracks IP addresses in the *rmtab* file. When a DUMP request is made (by someone running **showmount -a**, for instance), it returns IP addresses instead of hostnames by default. This option causes **rpc.mountd** to perform a reverse lookup on each IP address and return that hostname instead. Enabling this can have a substantial negative effect on performance in some situations.

**-t N** or **--num-threads=N** or **--num-threads N**
> This option specifies the number of worker threads that rpc.mountd spawns. The default is 1 thread, which is probably enough. More threads are usually only needed for NFS servers which need to handle mount storms of hundreds of NFS mounts in a few seconds, or when your DNS server is slow or unreliable.

**-u  or  --no-udp**
> Don't advertise UDP for mounting

**-V version  or  --nfs-version version**
> This option can be used to request that **rpc.mountd** offer certain versions of NFS. The current version of **rpc.mountd** can support both NFS version 2 and the newer version 3.

**-v  or  --version**
> Print the version of **rpc.mountd** and exit.

**-g  or  --manage-gids**
> Accept requests from the kernel to map user id numbers into lists of group id numbers for use in access control. An NFS request will normally (except when using Kerberos or other cryptographic authentication) contains a user-id and a list of group-ids. Due to a limitation in the NFS protocol, at most 16 groups ids can be listed. If you use the **-g** flag, then the list of group ids received from the client will be replaced by a list of group ids determined by an appropriate lookup on the server. Note that the 'primary' group id is not affected so a **newgroup** command on the client will still be effective. This function requires a Linux Kernel with version at least 2.6.21.

## TCP_WRAPPERS SUPPORT

You can protect your **rpc.mountd** listeners using the **tcp_wrapper** library or iptables(8).

Note that the **tcp_wrapper** library supports only IPv4 networking.

Add the hostnames of NFS peers that are allowed to access **rpc.mountd** to */etc/hosts.allow*. Use the daemon name **mountd** even if the **rpc.mountd** binary has a different name.

Hostnames used in either access file will be ignored when they can not be resolved into IP addresses. For further information see the **tcpd(8)** and hosts_access(5) man pages.

### IPv6 and TI-RPC support

TI-RPC is a pre-requisite for supporting NFS on IPv6. If TI-RPC support is built into **rpc.mountd**, it attempts to start listeners on network transports marked 'visible' in */etc/netconfig*. As long as at least one network transport listener starts successfully, **rpc.mountd** will operate.

## FILES

| | |
|---|---|
| */etc/exports* | input file for **exportfs**, listing exports, export options, and access control lists |
| */var/lib/nfs/rmtab* | table of clients accessing server's exports |

## SEE ALSO

exportfs(8), exports(5), showmount(8), rpc.nfsd(8), **rpc.rquotad(8),** nfs(5), **tcpd(8),** hosts_access(5), iptables(8), netconfig(5)

RFC 1094 - "NFS: Network File System Protocol Specification"
RFC 1813 - "NFS Version 3 Protocol Specification"

## AUTHOR

Olaf Kirch, H. J. Lu, G. Allan Morris III, and a host of others.