

NAME

mount_nfs - mount NFS file systems

SYNOPSIS

mount_nfs [-**23bcdiLINPsTU**] [-**a** *maxreadahead*] [-**D** *deadthresh*] [-**g** *maxgroups*] [-**I** *readdirsize*]
 [-**o** *options*] [-**R** *retrycnt*] [-**r** *readsize*] [-**t** *timeout*] [-**w** *writesize*] [-**x** *retrans*] *rhost:path node*

DESCRIPTION

The **mount_nfs** utility calls the `nmount(2)` system call to prepare and graft a remote NFS file system (*rhost:path*) on to the file system tree at the point *node*. This command is normally executed by `mount(8)`. For NFSv2 and NFSv3, it implements the mount protocol as described in RFC 1094, Appendix A and RFC 1813, Appendix I. For NFSv4, it uses the NFSv4 protocol as described in RFC 7530, RFC 5661 and RFC 7862.

By default, **mount_nfs** keeps retrying until the mount succeeds. This behaviour is intended for file systems listed in `fstab(5)` that are critical to the boot process. For non-critical file systems, the **bg** and **retrycnt** options provide mechanisms to prevent the boot process from hanging if the server is unavailable.

If the server becomes unresponsive while an NFS file system is mounted, any new or outstanding file operations on that file system will hang uninterruptibly until the server comes back. To modify this default behaviour, see the **intr** and **soft** options.

The options are:

- o** Options are specified with a **-o** flag followed by a comma separated string of options. See the `mount(8)` man page for possible options and their meanings. The following NFS specific options are also available:

acregmin=<*seconds*>

acregmax=<*seconds*>

acdirmin=<*seconds*>

acdirmax=<*seconds*>

When attributes of files are cached, a timeout calculated to determine whether a given cache entry has expired. These four values determine the upper and lower bounds of the timeouts for "directory" attributes and "regular" (ie: everything else). The default values are 3 -> 60 seconds for regular files, and 30 -> 60 seconds for directories. The algorithm

to calculate the timeout is based on the age of the file. The older the file, the longer the cache is considered valid, subject to the limits above.

actimeo=<*seconds*>

Set four cache timeouts above to specified value.

allgssname

This option can be used along with **-o gssname** to specify that all operations should use the host-based initiator credential. This may be used for clients that run system daemons that need to access files on the NFSv4 mounted volume.

bg If an initial attempt to contact the server fails, fork off a child to keep trying the mount in the background. Useful for `fstab(5)`, where the file system mount is not critical to multiuser operation.

bgnow

Like **bg**, fork off a child to keep trying the mount in the background, but do not attempt to mount in the foreground first. This eliminates a 60+ second timeout when the server is not responding. Useful for speeding up the boot process of a client when the server is likely to be unavailable. This is often the case for interdependent servers such as cross-mounted servers (each of two servers is an NFS client of the other) and for cluster nodes that must boot before the file servers.

deadthresh=<*value*>

Set the "dead server threshold" to the specified number of round trip timeout intervals before a "server not responding" message is displayed.

dumbtimer

Turn off the dynamic retransmit timeout estimator. This may be useful for UDP mounts that exhibit high retry rates, since it is possible that the dynamically estimated timeout interval is too short.

fg Same as not specifying **bg**.

gssname=<*service-principal-name*>

This option can be used with the KerberosV security flavors for NFSv4 mounts to specify the "service-principal-name" of a host-based entry in the default keytab file that is used for system operations. It allows the mount to be performed by "root" and avoids problems with cached credentials for the system operations expiring. The "service-principal-name" should be specified without instance or domain and is typically

"host", "nfs" or "root", although the form `<service>@<fqdn>` can also be used if the local system's `gethostname(3)` value does not match the host-based principal in the keytab.

hard Same as not specifying **soft**.

intr Make the mount interruptible, which implies that file system calls that are delayed due to an unresponsive server will fail with `EINTR` when a termination signal is posted for the process. To avoid leaving file locks in an indeterminate state on the NFS server, it is recommended that the **noLOCKD** option be used with this option.

maxgroups=<value>

Set the maximum size of the group list for the credentials to the specified value. This should be used for mounts on old servers that cannot handle a group list size of 16, as specified in RFC 1057. Try 8, if users in a lot of groups cannot get response from the mount point.

mntudp

Force the mount protocol to use UDP transport, even for TCP NFS mounts. (Necessary for some old BSD servers.)

nametimeo=<value>

Override the default of `NFS_DEFAULT_NAMETIMEO` for the timeout (in seconds) for positive name cache entries. If this is set to 0 it disables positive name caching for the mount point.

negnametimeo=<value>

Override the default of `NFS_DEFAULT_NEGNAMETIMEO` for the timeout (in seconds) for negative name cache entries. If this is set to 0 it disables negative name caching for the mount point.

nconnect=<value>

Specify the number of TCP connections (1-16) to be used for an NFS Version 4, minor version 1 or 2 mount. Multiple TCP connections can provide more client to server network bandwidth for certain network configurations such as:

- Multiple network interfaces that are aggregated together.
- A fast network interface that uses multiple queues.

The first TCP connection will be used for all RPCs that consist entirely of small RPC messages. The RPCs that can have large RPC messages (Read/Readdir/Write) are

distributed over the additional TCP connections in a round robin fashion. This option will result in more IP port#s being used. This option requires the **nfsv4** option. Note that for NFS servers such as AmazonEFS, where each new TCP connection can connect to a different cluster that maintains lock state separately, this option cannot be used.

nfsv2 Use the NFS Version 2 protocol (the default is to try version 3 first then version 2). Note that NFS version 2 has a file size limit of 2 gigabytes.

nfsv3 Use the NFS Version 3 protocol.

nfsv4 Use the NFS Version 4 protocol. This option will force the mount to use TCP transport. By default, the highest minor version of NFS Version 4 that is supported by the NFS Version 4 server will be used. See the **minorversion** option. Make sure that all your NFS Version 4 clients have unique values in */etc/hostid*.

minorversion=<value>

Use the specified minor version for a NFS Version 4 mount, overriding the default. The minor versions supported are 0, 1, and 2. This option is only meaningful when used with the **nfsv4** option.

oneopenown

Make a minor version 1 or 2 of the NFS Version 4 protocol mount use a single OpenOwner for all Opens. This may be useful for a server with a very low limit on OpenOwners, such as AmazonEFS. It may be required when an accumulation of NFS version 4 Opens occurs, as indicated by the "Opens" count displayed by *nfsstat(1)* with the **-c** and **-E** command-line options. A common case for an accumulation of Opens is a shared library within the NFS mount that is used by several processes, where at least one of these processes is always running. This option cannot be used for an NFS Version 4, minor version 0 mount. It may not work correctly when Delegations are being issued by a server, but note that the AmazonEFS server does not issued delegations at this time. This option is only meaningful when used with the **nfsv4** option.

pnfs Enable support for parallel NFS (pNFS) for minor version 1 or 2 of the NFS Version 4 protocol. This option is only meaningful when used with the **nfsv4** option.

noac Disable attribute caching.

noconn

For UDP mount points, do not do a *connect(2)*. This must be used if the server does not reply to requests from the standard NFS port number 2049 or replies to requests using a

different IP address (which can occur if the server is multi-homed). Setting the *vfs.nfs.nfs_ip_paranoia* sysctl to 0 will make this option the default.

nocto Normally, NFS clients maintain the close-to-open cache coherency. This works by flushing at close time and checking at open time. Checking at open time is implemented by getting attributes from the server and purging the data cache if they do not match attributes cached by the client.

This option disables checking at open time. It may improve performance for read-only mounts, but should only be used if the data on the server changes rarely. Be sure to understand the consequences before enabling this option.

noinet4, noinet6

Disables AF_INET or AF_INET6 connections. Useful for hosts that have both an A record and an AAAA record for the same name.

nolockd

Do *not* forward fcntl(2) locks over the wire via the NLM protocol for NFSv3 mounts or via the NFSv4 protocol for NFSv4 mounts. All locks will be local and not seen by the server and likewise not seen by other NFS clients for NFSv3 or NFSv4 mounts. This removes the need to run the rpcbind(8) service and the rpc.statd(8) and rpc.lockd(8) servers on the client for NFSv3 mounts. Note that this option will only be honored when performing the initial mount, it will be silently ignored if used while updating the mount options. Also, note that NFSv4 mounts do not use these daemons. The NFSv4 protocol handles locks, unless this option is specified.

noncontigwr

This mount option allows the NFS client to combine non-contiguous byte ranges being written such that the dirty byte range becomes a superset of the bytes that are dirty. This reduces the number of writes significantly for software builds. The merging of byte ranges is not done if the file has been file locked, since most applications modifying a file from multiple clients will use file locking. As such, this option could result in a corrupted file for the rare case of an application modifying the file from multiple clients concurrently without using file locking.

principal

For the RPCSEC_GSS security flavors, such as krb5, krb5i and krb5p, this option sets the name of the host based principal name expected by the server. This option overrides the default, which will be “nfs@<server-fqdn>” and should normally be sufficient.

noresvport

Do *not* use a reserved socket port number (see below).

port=<port_number>

Use specified port number for NFS requests. The default is to query the portmapper for the NFS port.

proto=<protocol>

Specify transport protocol version to use. Currently, they are:

udp - Use UDP over IPv4

tcp - Use TCP over IPv4

udp6 - Use UDP over IPv6

tcp6 - Use TCP over IPv6

rdirplus

Used with NFSV3 to specify that the **ReaddirPlus** RPC should be used. For NFSV4, setting this option has a similar effect, in that it will make the Readdir Operation get more attributes. This option reduces RPC traffic for cases such as "ls -l", but tends to flood the attribute and name caches with prefetched entries. Try this option and see whether performance improves or degrades. Probably most useful for client to server network interconnects with a large bandwidth times delay product.

readahead=<value>

Set the read-ahead count to the specified value. This may be in the range of 0 - 4, and determines how many blocks will be read ahead when a large file is being read sequentially. Trying a value greater than 1 for this is suggested for mounts with a large bandwidth * delay product.

readdirsize=<value>

Set the readdir read size to the specified value. The value should normally be a multiple of DIRBLKSIZ that is <= the read size for the mount.

resvport

Use a reserved socket port number. This flag is obsolete, and only retained for compatibility reasons. Reserved port numbers are used by default now. (For the rare case where the client has a trusted root account but untrustworthy users and the network cables are in secure areas this does help, but for normal desktop clients this does not apply.)

retrans=<value>

Set the retransmit timeout count for soft mounts to the specified value.

retrycnt=<count>

Set the mount retry count to the specified value. The default is a retry count of zero, which means to keep retrying forever. There is a 60 second delay between each attempt.

rsize=<value>

Set the read data size to the specified value. It should normally be a power of 2 greater than or equal to 1024. This should be used for UDP mounts when the "fragments dropped due to timeout" value is getting large while actively using a mount point. (Use `netstat(1)` with the `-s` option to see what the "fragments dropped due to timeout" value is.)

sec=<flavor>

This option specifies what security flavor should be used for the mount. Currently, they are:

krb5 - Use KerberosV authentication

krb5i - Use KerberosV authentication and
apply integrity checksums to RPCs

krb5p - Use KerberosV authentication and
encrypt the RPC data

sys - The default AUTH_SYS, which uses a
uid + gid list authenticator

soft A soft mount, which implies that file system calls will fail after *retrycnt* round trip timeout intervals.

syskrb5

This option specifies that a KerberosV NFSv4 minor version 1 or 2 mount uses AUTH_SYS for system operations. Using this option avoids the need for a KerberosV mount to have a host-based principal entry in the default keytab file (no **gssname** option) or a requirement for the user doing the mount to have a valid KerberosV ticket granting ticket (TGT) when the mount is done. This option is intended to be used with the **sec=krb5** and **tls** options and can only be used for NFSv4 mounts with minor version 1 or 2.

tcp Use TCP transport. This is the default option, as it provides for increased reliability on both LAN and WAN configurations compared to UDP. Some old NFS servers do not support this method; UDP mounts may be required for interoperability.

timeout=<value>

Set the initial retransmit timeout to the specified value, expressed in tenths of a second. May be useful for fine tuning UDP mounts over internetworks with high packet loss rates or an overloaded server. Try increasing the interval if `nfsstat(1)` shows high retransmit rates while the file system is active or reducing the value if there is a low retransmit rate but long response delay observed. (Normally, the **dumbtimer** option should be specified when using this option to manually tune the timeout interval.)

timeo=<value>

Alias for **timeout**.

tls This option specifies that the connection to the server must use TLS per RFC 9289. TLS is only supported for TCP connections and the `rpc.tlsclntd(8)` daemon must be running for an NFS over TCP connection to use TLS.

tlscertname=<name>

This option specifies the name of an alternate certificate to be presented to the NFS server during TLS handshake. The default certificate file names are "cert.pem" and "certkey.pem". When this option is specified, *name* replaces "cert" in the above file names. For example, if the value of *name* is specified as "other" the certificate file names to be used will be "other.pem" and "otherkey.pem". These files are stored in `/etc/rpc.tlsclntd` by default. This option is only meaningful when used with the **tls** option and the `rpc.tlsclntd(8)` is running with the **-m** command line flag set.

udp Use UDP transport.

vers=<vers_number>

Use the specified version number for NFS requests. See the **nfsv2**, **nfsv3**, and **nfsv4** options for details.

wcommitsize=<value>

Set the maximum pending write commit size to the specified value. This determines the maximum amount of pending write data that the NFS client is willing to cache for each file.

wsiz=<value>

Set the write data size to the specified value. Ditto the comments w.r.t. the **rsiz** option, but using the "fragments dropped due to timeout" value on the server instead of the client. Note that both the **rsiz** and **wsiz** options should only be used as a last ditch effort at improving performance when mounting servers that do not support TCP mounts.

IMPLEMENTATION NOTES

When neither the **rsize** nor **wsize** options are specified, the I/O size will be set to the largest value supported by both the NFS client and server. The largest value supported by the NFS client is defined by the tunable **vfs.maxbcachebuf** which can be set to a power of two up to **kern.maxphys**.

The `nfsstat(1)` command with the **-m** command line option will show what **mount_nfs** option settings are actually in use for the mount.

COMPATIBILITY

The following command line flags are equivalent to **-o** named options and are supported for compatibility with older installations.

- 2** Same as **-o nfsv2**
- 3** Same as **-o nfsv3**
- D** Same as **-o deadthresh**
- I** Same as **-o readdirsize=<value>**
- L** Same as **-o nolockd**
- N** Same as **-o noresvport**
- P** Use a reserved socket port number. This flag is obsolete, and only retained for compatibility reasons. (For the rare case where the client has a trusted root account but untrustworthy users and the network cables are in secure areas this does help, but for normal desktop clients this does not apply.)
- R** Same as **-o retrycnt=<value>**
- T** Same as **-o tcp**
- U** Same as **-o mntudp**
- a** Same as **-o readahead=<value>**
- b** Same as **-o bg**
- c** Same as **-o noconn**

- d** Same as **-o dumbtimer**
- g** Same as **-o maxgroups**
- i** Same as **-o intr**
- l** Same as **-o rdirplus**
- r** Same as **-o rsize=<value>**
- s** Same as **-o soft**
- t** Same as **-o retransmit=<value>**
- w** Same as **-o wsize=<value>**
- x** Same as **-o retrans=<value>**

The following **-o** named options are equivalent to other **-o** named options and are supported for compatibility with other operating systems (e.g., Linux, Solaris, and OSX) to ease usage of autofs(5) support.

- o vers=2**
Same as **-o nfsv2**
- o vers=3**
Same as **-o nfsv3**
- o vers=4**
Same as **-o nfsv4**

SEE ALSO

nfsstat(1), nmount(2), unmount(2), lagg(4), nfsv4(4), fstab(5), gssd(8), mount(8), nfsd(8), nfsiod(8), rpc.tlscIntd(8), showmount(8)

HISTORY

A version of the **mount_nfs** utility appeared in 4.4BSD.

BUGS

Since NFSv4 performs open/lock operations that have their ordering strictly enforced by the server, the

options **intr** and **soft** cannot be safely used. For NFSv4 minor version 1 or 2 mounts, the ordering is done via session slots and the NFSv4 client now handles broken session slots fairly well. As such, if the **noLOCKD** option is used along with **intr** and/or **soft**, an NFSv4 minor version 1 or 2 mount should work fairly well, although still not completely correctly. For NFSv4 minor version 0 mounts, **hard** mounts without the **intr** mount option is strongly recommended.