

# Home NFS Setup HOWTO

This is a quick guide to setting up NFS in Slackware for use in a home LAN. The example used is for connection of a laptop computer with a desktop computer that also has an NTFS partition mounted on /music. It assumes that basic network connectivity has been established. This guide is largely cut and pasted from other more definitive documents.

From <http://nfs.sourceforge.net/nfs-howto/>

## 2.1. What is NFS?

The Network File System (NFS) was developed to allow machines to mount a disk partition on a remote machine as if it were a local disk. It allows for fast, seamless sharing of files across a network.

It also gives the potential for unwanted people to access your hard drive over the network (and thereby possibly read your email and delete all your files as well as break into your system) if you set it up incorrectly.

Setting up a secure NFS does require some additional work, but as good security habits start at home, these steps will also be presented.

For the purposes of this example:

- the desktop computer will be called DESKTOP and has an IP address 10.1.1.2
- the laptop computer will be called LAPTOP and has an IP address 10.1.1.3 using a wired connection
- or the laptop computer will be called LAPTOP-W and has an IP address 10.1.1.4 using a wireless connection

Both computers will be setup to act as NFS servers as well as clients.

## Access settings

### /etc/hosts

On the desktop computer /etc/hosts should have lines like:

10.1.1.3	LAPTOP.<your domain>	LAPTOP
10.1.1.4	LAPTOP-W.<your domain>	LAPTOP-W

On the laptop computer /etc/hosts should have a line like:

10.1.1.2	DESKTOP.<your domain>	DESKTOP
----------	-----------------------	---------

## **/etc/hosts.deny**

On both machines add:

```
portmap:ALL
lockd:ALL
mountd:ALL
rquotad:ALL
statd:ALL
```

## **/etc/hosts.allow**

On the desktop computer add:

```
# For NFS mount from LAN
portmap: 10.1.1.3 , 10.1.1.4
lockd: 10.1.1.3 , 10.1.1.4
rquotd: 10.1.1.3 , 10.1.1.4
mountd: 10.1.1.3 , 10.1.1.4
statd: 10.1.1.3 , 10.1.1.4
```

On the laptop computer add:

```
# For NFS mount from LAN
portmap: 10.1.1.2
lockd: 10.1.1.2
rquotd: 10.1.1.2
mountd: 10.1.1.2
statd: 10.1.1.2
```

The syntax in the above examples can be altered, e.g. 10.1.1. could be used to allow access from any machine on the 10.1.1.0/24 network.

## **/etc/exports**

On the desktop computer add:

```
# Allow export of root file system to LAPTOP with read/write and root access
/                LAPTOP(rw,no_root_squash,no_subtree_check) \
                 LAPTOP-W(rw,no_root_squash,no_subtree_check)
/music          LAPTOP(rw,no_root_squash,no_subtree_check,nohide) \
                 LAPTOP-W(rw,no_root_squash,no_subtree_check,nohide)
```

On the laptop computer add:

```
# Allow export of root file system to DESKTOP with read/write and root
access
/                DESKTOP(rw,no_root_squash,no_subtree_check)
```

Comments

- The `no_root_squash` option is very permissive. The `root_squash` option is much more secure.
- Allowing access to the entire root file system is very permissive. Restriction to a sub-directory is much more secure.
- The `nohide` option is required to show the contents of other mounted partitions.

## Daemon startup

### `/etc/rc.d/rc.nfsd`

Check that this file is executable on both computers

### `/etc/rc.d/rc.rpc`

Check that this file is executable on both computers. (Not strictly necessary as `/etc/rc.d/rc.nfsd` will run this, but will be important if you want the computer to work as an NFS client only).

## Binding ports

### a) Slackware versions up to 14.2

To use NFS through the firewall follow this guide that is quoted verbatim. Thanks rworkman!

From [http://rlworkman.net/howtos/NFS\\_Firewall\\_HOWTO](http://rlworkman.net/howtos/NFS_Firewall_HOWTO)

This document is intended to give you detailed steps for making NFS bind to user-specified ports instead of random ports assigned by the portmapper. This makes it *\*much\** easier to run a firewall on the NFS server, as you don't have to kludge something to find the NFS ports at each boot to open them with iptables.

NOTE: This was written for Slackware Linux, but the general ideas should apply on pretty much any distribution.

First, you'll want (it's not necessary, but handy to have for later) to make sure all of this is in `/etc/services`. I made sure "NFS" is in all of what I added or modified so that I can easily remove them (or just find them) if I need them later. Note that many of these are *\*local\** additions - they are not official IANA assignments.

<code>sunrpc</code>	<code>111/tcp</code>	<code># SUN Remote Procedure Call</code>
<code>sunrpc</code>	<code>111/udp</code>	<code># SUN Remote Procedure Call</code>
<code>nfsd</code>	<code>2049/tcp</code>	<code># NFS server daemon</code>
<code>nfsd</code>	<code>2049/udp</code>	<code># NFS server daemon</code>
<code>rpc.nfs-cb</code>	<code>32764/tcp</code>	<code># RPC nfs callback</code>
<code>rpc.nfs-cb</code>	<code>32764/udp</code>	<code># RPC nfs callback</code>
<code>status</code>	<code>32765/udp</code>	<code># NFS status (listen)</code>

```
status      32765/tcp      # NFS status (listen)
status      32766/udp      # NFS status (send)
status      32766/tcp      # NFS status (send)
mountd      32767/udp      # NFS mountd
mountd      32767/tcp      # NFS mountd
lockd       32768/udp      # NFS lock daemon/manager
lockd       32768/tcp      # NFS lock daemon/manager
rquotad     32769/udp      # NFS rquotad
rquotad     32769/tcp      # NFS rquotad
```

\*\*\*\*\*

Next, you'll need to modify your `/etc/rc.d/rc.nfsd` script accordingly:  
For other linux distributions, find the script that starts these  
daemons and add the needed flags.

```
# **** Make the quota daemon listen on port 32769
```

```
if [ -x /usr/sbin/rpc.rquotad ]; then
    echo " /usr/sbin/rpc.rquotad -p 32769"
    /usr/sbin/rpc.rquotad -p 32769
fi
```

```
# **** Make the mount daemon listen on port 32767
```

```
if [ -x /usr/sbin/rpc.mountd ]; then
    echo " /usr/sbin/rpc.mountd -p 32767"
    /usr/sbin/rpc.mountd -p 32767
fi
```

Now modify the `/etc/rc.d/rc.rpc` script (again, for other linux distros,  
find the script that starts this daemon and add the needed flags).  
On older versions (less than 11.0) of Slackware, `rpc.statd` is started  
in `rc.nfsd`, so look there instead.

```
# **** Have the portmap daemon chroot to /var/empty for increased security
# **** Make the status daemon listen on port 32765 and talk on port 32766
```

```
if [ -x /sbin/rpc.portmap -a -x /sbin/rpc.statd ]; then
    if ! ps axc | grep -q rpc.portmap ; then
        echo "Starting RPC portmapper: /sbin/rpc.portmap -t /var/empty"
        /sbin/rpc.portmap -t /var/empty
    fi
    if ! ps axc | grep -q rpc.statd ; then
        echo "Starting RPC NSM (Network Status Monitor): /sbin/rpc.statd -p
32765 -o 32766"
        /sbin/rpc.statd -p 32765 -o 32766
    fi
fi
```

```
# **** Note that you'll have to open port 32766 on the NFS clients
```

Now make the lock daemon listen on port 32768 only and set the nfs callback port to 32764.

Up to Slackware 11.0, this requires a kernel boot parameter (an append= line in lilo.conf) - a kernel stanza will look something like this:

```
image = /boot/vmlinuz-ide-2.4.37.11
append = "lockd.udpport=32768 lockd.tcpport=32768"
root = /dev/hda2
label = 2.4.37.11
read-only
```

After 11.0, but before Slackware 13.1, this requires setting module load options in a file in the /etc/modprobe.d/ directory - I create an aptly named file of /etc/modprobe.d/nfs.conf file and add the following lines:

```
options lockd nlm_udpport=32768 nlm_tcpport=32768
options nfs callback_tcpport=32764      # This is for NFSv4
```

In Slackware 13.1 and later, you will instead need to place the following in /etc/sysctl.conf:

```
fs.nfs.nlm_udpport=32768
fs.nfs.nlm_tcpport=32768
fs.nfs.nfs_callback_tcpport=32764
```

Finally, you'll need to reboot the machine since the lockd module probably will refuse to unload. Once it's rebooted, you can test to make sure all of the changes "took" with "rpcinfo -p" or "pmap\_dump" (rpcinfo is no longer present in Slackware 14.0 or later) -- as an example, here's a snippet of what I see here:

```
stora # rpcinfo -p
  program vers proto  port
  100000    2   tcp    111  portmapper
  100000    2   udp    111  portmapper
  100024    1   udp   32766  status
  100024    1   tcp   32766  status
  100227    3   tcp   2049
  100227    3   udp   2049
  100021    3   udp   32768  nlockmgr
  100021    3   tcp   32768  nlockmgr
  100005    3   udp   32767  mountd
  100005    3   tcp   32767  mountd
```

Copyright 2006-2011 Robby Workman, Tuscaloosa, Alabama, USA

```
#include <mit.h>  /* Standard MIT License */
You may mirror and/or otherwise use this file as you wish so long as you
retain attribution to the author.  If you have any questions, comments, or
suggestions for improvement, you may contact me at rworkman AT slackware.com
```

Note: Updated 20111126 for better consistency with the Debian NFS HOWTO (since the actual port assignments aren't important, we may as well recommend the same thing in both places); thanks to David Allen for the pointers and recommendation...

Note: Updated 20120820 to note pmap\_dump usage instead of rpcinfo in Slackware 14+; thanks to David Allen for the heads-up on that.

## b) Slackware version after 14.2

To use NFS through the firewall is now easier, as options for NFS can be set in `/etc/default/nfs` and `/etc/default/rpc`. It is simply necessary to uncomment the appropriate lines in `/etc/default/rpc`.

### `/etc/default/rpc`

```
# See also /etc/default/nfs

# Optional arguments passed to rpcbind. See rpcbind(8)
#RPCBIND_OPTS=""
#
# Optional arguments passed to rpc.statd. See rpc.statd(8)
#RPC_STATD_OPTS=""
# Optional hostname to start rpc.statd with.
#RPC_STATD_HOSTNAME="darkstar"
# Port rpc.statd should listen on.
RPC_STATD_PORT=32766
# Outgoing port rpc.statd should use.
RPC_STATD_OUTGOING_PORT=32765
#
# Optional options passed to rquotad. See rquotad(8)
#RPC_RQUOTAD_OPTS=""
# Optional port rquotad should listen on:
RPC_RQUOTAD_PORT=32769
#
# TCP port rpc.lockd should listen on:
LOCKD_TCP_PORT=32768
# UDP port rpc.lockd should listen on:
LOCKD_UDP_PORT=32768
#
# Optional arguments passed to rpc.mountd. See rpc.mountd(8)
#RPC_MOUNTD_OPTS=""
# Port rpc.mountd should listen on:
RPC_MOUNTD_PORT=32767
#
```

# Firewall settings

## /etc/rc.d/rc.firewall

Here are some example lines to allow NFS:

```
## NFS uses TCP and UDP on ports 111, 2049, 32764-32769
# Accept TCP and UDP on port 111 from local LAN for portmap
$IPTABLES -A INPUT -i $EXTIF -p tcp -s $LOCAL_LAN --dport 111 -j ACCEPT
$IPTABLES -A INPUT -i $EXTIF -p udp -s $LOCAL_LAN --dport 111 -j ACCEPT
# Accept TCP and UDP on port 2049 from local LAN for nfsd
$IPTABLES -A INPUT -i $EXTIF -p tcp -s $LOCAL_LAN --dport 2049 -j ACCEPT
$IPTABLES -A INPUT -i $EXTIF -p udp -s $LOCAL_LAN --dport 2049 -j ACCEPT
# Accept TCP and UDP on port 32765 from local LAN for statd listen
# (set in /etc/rc.d/rc.rpc for Slackware <= 14.2 or /etc/default/rpc for
Slackware >= 14.2-current)
$IPTABLES -A INPUT -i $EXTIF -p tcp -s $LOCAL_LAN --dport 32765 -j ACCEPT
$IPTABLES -A INPUT -i $EXTIF -p udp -s $LOCAL_LAN --dport 32765 -j ACCEPT
# Accept TCP and UDP on port 32766 from local LAN for statd send
# (set in /etc/rc.d/rc.rpc for Slackware <= 14.2 or /etc/default/rpc for
Slackware >= 14.2-current)
$IPTABLES -A INPUT -i $EXTIF -p tcp -s $LOCAL_LAN --dport 32766 -j ACCEPT
$IPTABLES -A INPUT -i $EXTIF -p udp -s $LOCAL_LAN --dport 32766 -j ACCEPT
# Accept TCP and UDP on port 32767 from local LAN for mountd
# (set in /etc/rc.d/rc.nfsd for Slackware <= 14.2 or /etc/default/rpc for
Slackware >= 14.2-current)
$IPTABLES -A INPUT -i $EXTIF -p tcp -s $LOCAL_LAN --dport 32767 -j ACCEPT
$IPTABLES -A INPUT -i $EXTIF -p udp -s $LOCAL_LAN --dport 32767 -j ACCEPT
# Accept TCP and UDP on port 32768 from local LAN for lockd
# (set in /etc/sysctl.conf for Slackware <= 14.2 or /etc/default/rpc for
Slackware >= 14.2-current)
$IPTABLES -A INPUT -i $EXTIF -p tcp -s $LOCAL_LAN --dport 32768 -j ACCEPT
$IPTABLES -A INPUT -i $EXTIF -p udp -s $LOCAL_LAN --dport 32768 -j ACCEPT
# Accept TCP and UDP on port 32769 from local LAN for rquotad
# (set in /etc/rc.d/rc.nfsd for Slackware <= 14.2 or /etc/default/rpc for
Slackware >= 14.2-current)
$IPTABLES -A INPUT -i $EXTIF -p tcp -s $LOCAL_LAN --dport 32769 -j ACCEPT
$IPTABLES -A INPUT -i $EXTIF -p udp -s $LOCAL_LAN --dport 32769 -j ACCEPT
```

The work is done! Everything should now be in place. After rebooting both machines it should now be possible to:

- mount the root file system on the desktop computer on the /mnt/tmp directory on the laptop using

```
mount DESKTOP:/ /mnt/tmp
```

- mount the root file system on the laptop computer on the /mnt/tmp directory on the desktop using

```
mount LAPTOP:/ /mnt/tmp
```

## Sources

- <http://nfs.sourceforge.net/nfs-howto/>
- [http://rlworkman.net/howtos/NFS\\_Firewall\\_HOWTO](http://rlworkman.net/howtos/NFS_Firewall_HOWTO)

[howtos](#), [software](#), [nfs](#), [slackware 13.37](#), [slackware 14.0](#), [author allend](#)

From:

<https://docs.slackware.com/> - **SlackDocs**

Permanent link:

[https://docs.slackware.com/howtos:network\\_services:home\\_nfs\\_howto](https://docs.slackware.com/howtos:network_services:home_nfs_howto)

Last update: **2019/04/06 13:54 (UTC)**

