

Mounting NFS shares on Mac OS X

I've recently decided to give [Mac OS X](#) a try. For the past week or so I've been spending a good few hours a day working in **Snow Leopard** installed on a MacBook Pro borrowed from a friend.

While [Mac OS](#) is unlike any Unix-like operating system I've managed so far, there are certainly some of similarities. I can honestly say that I'm enjoying the Mac Book Pro so far, and hope to discover most of the differences compared to my previous Unix-like desktop which is Ubuntu 9.10.

Mounting NFS on MAC OS X

One thing which I noticed immediately was that out of the box it was impossible to mount any NFS shares from my Ubuntu NAS server. Any attempt to [mount](#) a remote filesystem would give me an error like this:

```
mbp:~ root# mount nasbox:/try /mnt
mount_nfs: /mnt: Operation not permitted
```

This error was happening for a relatively simple NFS share on the Ubuntu box:

```
nasbox# cat /etc/exports
/try          (rw)
```

... so I started looking around and realised that the reason for this strange problem is quite simple.

Mac OS X uses non-standard port for outgoing NFS connections

That's right! Apparently, [Mac OS X](#) uses a non-privileged port (**2049**) for TCP and UDP connections serving the NFS transport.

What this means is that most likely attempts to mount remote filesystems will fail, because most NFS servers don't really like connections from insecure ports.

There are two ways to approach this problem:

1. Fix it on the client side (probably makes more sense)
2. Fix it on the NFS server side (if you manage both systems)

Using reserved NFS port number on Mac OS X

There's a mount option supported by the `mount_nfs` command, which allows you to force the NFS client connections to originate from a privileged port. This will magically make your attempts to mount remote filesystems successful. The option is called `resvport`.

First we double-check that default mounts still don't work:

```
mbp:~ root# mount nasbox:/try /mnt
mount_nfs: /mnt: Operation not permitted
```

... and now let's use the `resvport` option:

```
mbp:~ root# mount -o resvport nasbox:/try /mnt
```

... and make sure we're actually looking at a mounted filesystem:

```
mbp:~ root# df -h /mnt
Filesystem      Size   Used  Avail Capacity  Mounted on
nasbox:/try     61Gi   56Gi   2.6Gi    96%      /mnt
```

Allowing connections from non-privileged ports on NFS server

Like I said, if you manage both the Mac OS based client and the NFS server, perhaps it makes more sense to relax the

default NFS server security and allow the connections from non-privileged ports.

Just to remind you about the validity of such a decision, the option to allow non-privileged connections is called `insecure`.

Here's how you use it:

```
nasbox# cat /etc/exports
/try          (rw,insecure)
```

After making this change to the `/etc/exports` file, you'll have to restart your NFS server. On my Ubuntu NAS box, it's done like this:

```
nasbox# /etc/init.d/nfs-kernel-server restart
* Stopping NFS kernel daemon
...done.
* Unexporting directories for NFS kernel daemon...
...done.
* Exporting directories for NFS kernel daemon...
...done.
* Starting NFS kernel daemon
...done.
```

We know are ready to attempt the default mount of the same filesystem on the **Mac OS X** client:

```
mbp:~ root# mount nasbox:/try /mnt
```

That's it! I won't promise any **Mac OS** posts just yet, but if there is enough interest – I'd love to do the research and to post all the discoveries on the [Unix Tutorial](#) pages.

See Also

- [MacOS](#)
- [Helios 4 – NAS Storage](#)
- [MacOS Catalina](#)
- [MacOS Commands](#)
- [ISO to USB in MacOS](#)

- [Create bootable USB in MacOS](#)