

How to mount NTFS Windows XP filesystem on FreeBSD, NetBSD, OpenBSD

Author : admin



A friend of mine bring home a *Seagate External Hard Disk Drive* using USB 3 as a communication media. I needed to attach the hard disk to my FreeBSD router to transfer him some data, the External HDD is formatted to use NTFS as a main file partition and hence to make the file transfers I had to somehow mount the NTFS partition on the HDD.

FreeBSD and other BSDs, just like Linux does not have embedded NTFS file system mount support. In order to add an external write support for the plugged hdd NTFS I looked in the ports tree:

```
freebsd# cd /usr/ports
freebsd# make search name='ntfs'
Port: fusefs-ntfs-2010.10.2
Path: /usr/ports/sysutils/fusefs-ntfs
Info: Mount NTFS partitions (read/write) and disk images
Maint: ports@FreeBSD.org
B-deps: fusefs-libs-2.7.4 libiconv-1.13.1_1 libtool-2.4 libublio-20070103 pkg-config-0.25_1
R-deps: fusefs-kmod-0.3.9.p1.20080208_7 fusefs-libs-2.7.4 libiconv-1.13.1_1 libublio-20070103 pkg-config-0.25_1
WWW: http://www.tuxera.com/community/
```

```
Port: ntfsprogs-2.0.0_1
Path: /usr/ports/sysutils/ntfsprogs
Info: Utilities and library to manipulate NTFS partitions
Maint: ports@FreeBSD.org
B-deps: fusefs-libs-2.7.4 libiconv-1.13.1_1 libublio-20070103 pkg-config-0.25_1
```

R-deps: libublio-20070103 pkg-config-0.25_1

WWW: <http://www.linux-ntfs.org/>

```
freebsd# cd sysutils/fusefs-ntfs/
```

```
freebsd# ls
```

```
Makefile distinfo files/ pkg-descr pkg-plist
```

```
freebsd# cat pkg-descr
```

The ntfs-3g driver is an open source, freely available read/write NTFS driver, which provides safe and fast handling of the Windows XP, Windows Server 2003 and Windows 2000 filesystems. Almost the full POSIX filesystem functionality is supported, the major exceptions are changing the file ownerships and the access rights.

WWW: <http://www.tuxera.com/community/>

Using **ntfs-3g** I managed to succeed **mounting the NTFS on my old PC running FreeBSD ver. 7_2**

1. Installing fuserfs-ntfs support on BSD

Before I can use **ntfs-3g**, to mount the partition, I had to install **fuserfs-ntfs** bsd port, with:

```
freebsd# cd /usr/ports/sysutils/fusefs-ntfs
```

```
freebsd# make install clean
```

```
.....
```

I was curious if **ntfsprogs** provides other utilities to do the ntfs mount but whilst trying to install it I realized it is already installed as a dependency package to *fusefs-ntfs*.

fusefs-ntfs package provides a number of **utilities for creating, mounting, fixing and doing various manipulations with Microsoft NTFS filesystems**.

Here is a list of all the executable **utilities helpful in NTFS fs management**:

```
freebsd# pkg_info -L fusefs-ntfs\* | grep -E "/bin/|sbin/README"
```

```
/usr/local/bin/lowntfs-3g
```

```
/usr/local/bin/ntfs-3g
```

```
/usr/local/bin/ntfs-3g.probe
```

```
/usr/local/bin/ntfs-3g.secaudit
```

```
/usr/local/bin/ntfs-3g.usermap
```

```
/usr/local/bin/ntfscat
```

```
/usr/local/bin/ntfsccluster
```

```
/usr/local/bin/ntfscmp
```

```
/usr/local/bin/ntfsfix
```

```
/usr/local/bin/ntfsinfo
```

```
/usr/local/bin/ntfsls
```

```
/usr/local/sbin/mkntfs
```

```
/usr/local/sbin/ntfsclone
/usr/local/sbin/ntfscp
/usr/local/sbin/ntfslabel
/usr/local/sbin/ntfsresize
/usr/local/sbin/ntfsundelete
/usr/local/share/doc/ntfs-3g/README
/usr/local/share/doc/ntfs-3g/README.FreeBSD
```

The **README** and **README.FreeBSD** are wonderful, reading for those who want to get more in depth knowledge on using the up-listed utilities.

One utility, worthy to mention, I have used in the past is **ntfsfix**. *ntfsfix* resolve issues with NTFS partitions which were not unmounted on system shutdown (electricity outage), system hang up etc.

2. Start fusefs (ntfs) and configure it to auto load on system boot

Once **fuserfs-ntfs** is installed, if its necessary **ntfs** fs mounts to be permanently supported on the BSD system add **fusefs_enable="YES"** to **/etc/rc.conf** - (*the FreeBSD services auto load conf*).

```
freebsd# echo 'fusefs_enable="YES"' >> /etc/rc.conf
```

One note to make here is that you need to have also **dbus_enable="YES"** and **hald_enable="YES"** in **/etc/rc.conf**, not having this two in rc.conf will prevent fusefs to start properly. Do a quick grep to make sure this two variables are enabled:

Afterwards **fsusefs** load up script should be run:

```
freebsd# /usr/local/etc/rc.d/fusefs start
Starting fusefs.
```

Another alternative way to load ntfs support on the BSD host is to directly load **fuse.ko** kernel module:

```
freebsd# /sbin/kldload fuse.ko
```

3. Mounting the NTFS partition

In my case, the Seagate hard drive was detected as **da0**, where the NTFS partition was detected as **s1** (da0s1):

```
freebsd# dmesg|grep -i da0
da0 at umass-sim0 bus 0 target 0 lun 0
da0: Fixed Direct Access SCSI-4 device
da0: 40.000MB/s transfers
```

da0: 953869MB (1953525164 512 byte sectors: 255H 63S/T 121601C)br />GEOM_LABEL: Label for provider da0s1 is ntfs/Expansion Drive.
GEOM_LABEL: Label for provider da0s1 is ntfs/Expansion Drive.

Therefore further to mount it one can use **mount_ntfs** (to quickly mount in read only mode) or *ntfs-3g* for (read / write mode):

If you need to just quickly mount a disk drive to copy some data and umount it with no need for writing to the NTFS partition do;

```
freebsd# /sbin/mount_ntfs /dev/da0s1 /mnt/disk
```

Note that *mount_ntfs* command is a native BSD command and have nothing to do with **ntfs-3g**. Therefore using it to mount NTFS is not the same as mounting it via **ntfs-3g** cmd

```
freebsd# /usr/local/bin/ntfs-3g -o rw /dev/da0s1 /mnt/disk/
```

Something, I've noticed while using **ntfs-3g** is, it fails to properly exit even when the ntfs-3g shell execution is over:

```
freebsd# ps ax |grep -i ntfs|grep -v grep
18892 ?? Is 0:00.00 /usr/local/bin/ntfs-3g -o rw /dev/da0s1 /mnt/disk/
```

I dunno if this is some kind of ntfs-3g bug or feature specific to all versions of FreeBSD or it is something local to FBSD 7.2

Thought *ntfs-3g*, keeps appearing in process list, praise God as of time of writing NTFS support on FreeBSD proved to be stable.

Read / Write disk operations to the NTFS I tested it with works great. Just about 5 years ago I still remember write mode was still experimental. Now it seems NTFS mounts can be used with no hassle even on production machines.

4. Auto mounting NTFS partition on FreeBSD system boot

There are two approaches towards 'the problem' I can think of.
The better way to auto mount on boot (in my view) is through **/etc/fstab** use

If /etc/fstab + ntfs-3g is to be used, you will however change the default **/sbin/mount_ntfs** command to point to **/usr/local/bin/ntfs-3g**, i.e.:

```
freebsd# mv /sbin/mount_ntfs /sbin/mount_ntfs.orig
freebsd# ln -s /usr/local/bin/ntfs-3g /sbin/mount_ntfs
```

Then to mount `/dev/da0s1` via `/etc/fstab` add line:

```
/dev/ad0s1 /mnt/disk ntfs rw,late 0 0
```

To not bother with text editor run:

```
freebsd# echo '/dev/ad0s1 /mnt/disk ntfs rw,late 0 0' >> /etc/fstab
```

I've read in posts in freebsd forums, there is also a way to use **ntfs-3g** for mounting partitions, without substituting the original `bsd/sbin/mount_ntfs`, the exact commands suggested to be used with no need to prior **mv /sbin/mount_ntfs to /sbin/mount_ntfs.orig and link it to ntfs** is:

```
/dev/ad0s1 /disk ntfs rw,mountprog=/usr/local/bin/ntfs-3g,late 0 0
```

For any *other NTFS partitions*, for instance `/dev/ad0s2`, `/dev/ad2s1` etc. simply change the partition name and *mount points*.

The second alternative to adding the NTFS to auto mount is through `/etc/rc.local`. `/etc/rc.local` content will be executed very late in system boot. :

```
echo '/usr/local/bin/ntfs-3g -o rw /dev/da0s1' >> /etc/rc.local
```

One disadvantage of using `/etc/rc.local` for mounting the partition is the hanging `ntfs-3g` in proc list:

```
freebsd# ps ax |grep -i ntfs|grep -v grep
18892 ?? Is 0:00.00 /usr/local/bin/ntfs-3g -o rw /dev/da0s1 /mnt/disk/
```

Though, I haven't tested it yet. Using the same methodology should be perfectly working on **PC-BSD**, **DragonFlyBSD**, **NetBSD** and **OpenBSD**.

I will be glad if someone who runs any of the other BSDs can confirm, following this instructions works fine on these BSDs too.