



for Mac OS X

User Guide

Index

1. Introduction to NTFS-3G.....	3
1.1 What is a file system driver?.....	3
1.2 How does it work?.....	3
1.3 Why two different builds?.....	4
2. Installation.....	5
2.1 Installing NTFS-3G.....	5
2.2 Uninstalling NTFS-3G.....	5
3. Using NTFS-3G.....	6
3.1 Mounting and unmounting NTFS volumes.....	6
3.2 Creating NTFS file systems.....	6
3.3 Checking and repairing NTFS file systems.....	7
4. Customizing NTFS-3G.....	8
4.1 Disabling NTFS-3G.....	8
4.2 Disabling the ublio caching layer.....	9
4.3 Disabling file name normalization.....	9
4.4 Enabling debug logging.....	10
5. Acknowledgments.....	11

1. Introduction to NTFS-3G

Thank you for downloading NTFS-3G, a free and open source read/write file system driver for the Windows file system NTFS. This document will explain what NTFS-3G is, how to install it and use it, and how to perform certain other operations.

This version of NTFS-3G has been adapted to Mac OS X, and can be used with both PowerPC Macs and the newer Intel Macs. It requires Mac OS X 10.4 (Tiger) or 10.5 (Leopard), and the free library *MacFUSE* in order to work.

MacFUSE can be downloaded from the following address: <http://code.google.com/p/macfuse/>

1.1 What is a file system driver?

A file system driver is the software component that interprets the on disk structures of the file system and presents a logical hierarchical view to the applications using it. It can be a part of the operating system, or come as a third party module, like NTFS-3G.

For instance, Mac OS X comes with drivers for the following physical file systems after a clean install:

- HFS / “Mac OS Original” (full support)
- HFS+ / “Mac OS Extended” (full support)
- HFSX / “Mac OS Extended (Case-sensitive)” (full support)
- Apple UFS / “UNIX File System” (full support)
- FAT / “MS-DOS File System” (full support)
- NTFS / “Windows NT Filesystem” (read only)
- ISO-9660 / “ISO 9660 File System” (full support)
- UDF / “Universal Disk Format” (full support)
- Mac OS X 10.5 only: ZFS / “ZFS File System” (read only)

The purpose of NTFS-3G is to extend not only Mac OS X, but many other operating systems too (Linux, FreeBSD, Solaris, ...) with the capabilities to write to NTFS partitions and having them appear as native parts of the operating system environment.

1.2 How does it work?

Mac OS X has a kind of plugin-based file system interface, where anyone can extend the operating system with new file system bundles, to make the operating system recognize, mount, create and repair other file systems than the built in types. This is what this NTFS-3G package uses to glue together the operating system independent NTFS-3G driver with Mac OS X.

Usually file systems drivers live in the kernel of the operating system, which makes writing them a risky business as they can easily crash the whole system if the developer makes mistakes. This is not the case with NTFS-3G. Instead, it lives in “user space”, where all other programs run, and is linked to the kernel through the file system bridge *MacFUSE* (a Mac version of FUSE, “Filesystem in Userspace”, originally created for the Linux kernel).

This approach makes NTFS-3G a lot more portable, and safer, since system stability won't be affected in the event of a file system driver crash. The only theoretical downside is that CPU usage will be slightly higher due to increased kernel/user space communication, which can result in slower read/write speeds. In practice, however, it has been shown that the speed of NTFS-3G can easily compare with kernel-based drivers (see: <http://www.ntfs-3g.org/performance.html>).

1.3 Why two different builds?

NTFS-3G for Mac OS X is currently released in two different versions for each new source release, the *stable* build and the *ublio* build. The reason for this is to make it possible for people to choose between data safety and performance.

NTFS-3G was originally written for Linux, which successfully, and with high performance provides caching for the hard disk devices that NTFS-3G accesses. Mac OS X and FreeBSD does not have that kind of efficient caching layer, so something similar must either be provided by NTFS-3G itself, or not at all. *Ublío* is the name of the caching layer provided by NTFS-3G through a patch. It was originally written by Csaba Henk for the FreeBSD port of NTFS-3G.

When only using internal drives, with ATA/SATA connections, you normally won't notice any difference in performance between the *ublio* build and the *stable* build. It is when using USB drives that the performance boost is most noticeable. When using a USB hard disk the transfer speed of the *ublio* build can be up to 20 times the speed of the *stable* build.

The downside of the *ublio* build is that you are more likely to lose data if you:

- Forget to unmount the drive prior to removing it from the system, for instance when unplugging an external hard disk. (It is always important to properly unmount all volumes on an external drive before pulling out the USB cable, but in this case it's even more so...)
- Have a power failure, system crash, or any other situation where the system does not shut down normally.

When using the *ublio* build, you can actually disable *ublio* for some volumes, while leaving it on for others. This gives you the “best of both worlds”.

See section 4 (Customizing NTFS-3G) for details on how to disable *ublio* for specific volumes.

2. Installation

2.1 Installing NTFS-3G

Installing NTFS-3G is not more complicated than installing any other program, as it uses the standard Mac OS X package format to install itself.

Just double click the package NTFS-3G.pkg and follow instructions. You will have to provide administrator credentials to install it, since it needs to place files in protected system folders.

After you have installed the package, you may (in some cases when doing an upgrade) need to reboot your computer because some parts of Mac OS X tend to cache certain data from the file system bundle. In other cases, you may just need to unmount your NTFS file systems and remount them to activate the newly installed NTFS-3G version.

2.2 Uninstalling NTFS-3G

NTFS-3G can be easily uninstalled using the *Uninstall NTFS-3G...* button in the NTFS-3G preference pane, which should be available in *System Preferences* once you have installed the package. See section 4 for a description of the other features in the preference pane.

If this should fail for some reason, an uninstall script exists at the following location:

```
/System/Library/Filesystems/ntfs-3g.fs/Support/uninstall-ntfs-3g.sh
```

3. Using NTFS-3G

3.1 Mounting and unmounting NTFS volumes

This should be completely transparent to the user. As soon as NTFS-3G is installed, it overrides the internal NTFS driver and automatically mounts NTFS volumes inserted into the system using NTFS-3G. Any volume that previously got mounted with Apple's NTFS driver should get mounted with NTFS-3G without any required user intervention.

NTFS-3G can not mount volumes that are marked as *unclean*, i.e. have not been correctly unmounted previously, or contain file system errors. This can happen if you just “pull the plug” on a volume in Windows.

Windows has a feature called “Safe remove hardware” that should always be used when removing removable disks from a Windows system (regardless of using NTFS-3G to access them or not). If this feature is not used, NTFS-3G will not mount the media, because it might be in an inconsistent state. If NTFS-3G did mount partitions that are in an unclean state, data loss could possibly occur.

NTFS-3G will similarly not mount “hibernated” partitions, that contain a saved system state of a Windows instance as this can also lead to data loss.

If you still think it's safe to mount these partitions, it is possible to mount them manually with the *force* option and thus remove the “unclean” or “hibernated” flags. Consult <http://www.ntfs-3g.org> for details on command line syntax when doing a manual mount.

3.2 Creating NTFS file systems

NTFS-3G is fairly integrated with the operating system's utilities, such as the built in graphical tool *Disk Utility* and the command line version *diskutil*.

Disk Utility in Mac OS X 10.4 (Tiger) will not let the user choose third party file systems such as NTFS-3G for creation and formatting of volumes, but in Mac OS X 10.5 (Leopard) this has been fixed.

The command line utility *diskutil* works with NTFS-3G in both 10.4 and 10.5.

Note that NTFS file systems should **not** be used on hard drives that has been partitioned with *Apple Partition Map*, which is the default system partitioning scheme for PowerPC macs.

If you create an NTFS file system on a *Apple Partition Map* drive, no Windows computer will be able to read it.

3.2.1 Formatting an existing volume as NTFS

In Mac OS X 10.5 (Leopard) this is completely integrated in *Disk Utility*. You will see *Windows NT Filesystem (NTFS-3G)* as a file system option when erasing a volume.

To format an existing volume as NTFS from the Terminal (which is the only option in Mac OS X 10.4), you use the command *diskutil eraseVolume*, specifying the file system type *NTFS-3G*. Just typing the command prints a brief help text. For more detailed usage information, consult the *man* page (“man diskutil” in the Terminal).

Note: Setting the volume label of an NTFS volume to anything more advanced than 8-10 ASCII characters without spaces will not work due to internal limitations in *diskutil*, which apparently hard codes the limits of volume labels for the file system that it knows about. This is a Mac OS X issue, which only Apple could solve.

3.2.2 Repartitioning a disk with NTFS partitions

Similarly to formatting an existing volume, Disk Utility can be used for this purpose in Mac OS X 10.5, but not in 10.4, where the only option is using the command line utility *diskutil*.

The command line to use when repartitioning a disk is *diskutil eraseDisk*, and works similar to the previously mentioned *diskutil eraseVolume*. Please read the man page (“man diskutil”) for further information on how to use this utility.

3.2.3 Creating an NTFS disk image

Disk images can be created with NTFS file systems. In Mac OS X 10.5 this is completely integrated in Disk Utility, and you get to choose the *Windows NT Filesystem (NTFS-3G)* file system type when creating an empty disk image. In Mac OS X 10.4 you can use the command line utility *hdiutil*. For example “*hdiutil create -size 50M -fs NTFS-3G image.dmg*” creates a 50 MiB unpartitioned image containing an empty NTFS file system.

The only disk images that can contain NTFS file systems are plain read/write disk images (UDRW), unpartitioned, or partitioned with the Master Boot Record or GUID Partition Table partitioning schemes. However, *hdiutil*, which has only been written with Apple's file systems in mind, does not allow you to put partition systems with NTFS file systems on the disk image out of the box (at least in OS X 10.4). Instead you need to use *diskutil* to repartition the disk image after you have attached it to the system if you want to put a Master Boot Record or GUID Partition Table onto it.

3.3 Checking and repairing NTFS file systems

There is only limited support for checking and repairing NTFS file systems using free software at this time. The best way to repair a damaged NTFS file system is without doubt by using *chkdsk* in Windows.

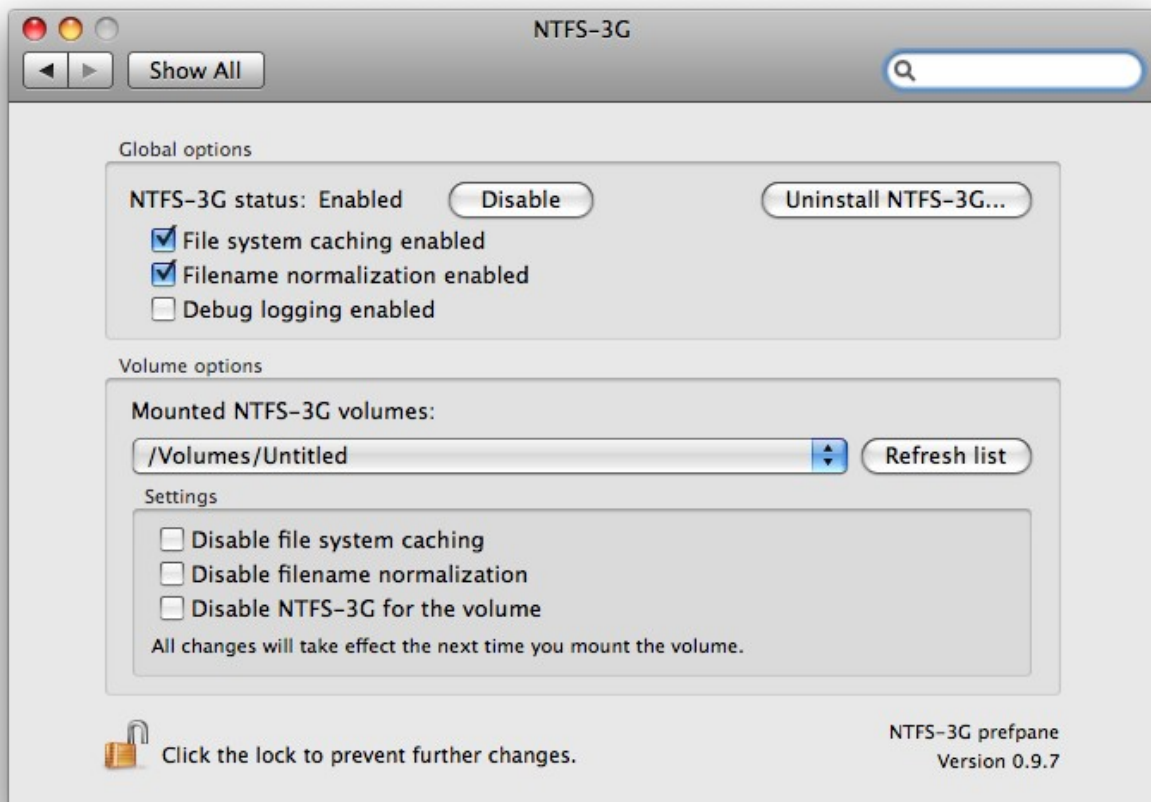
The NTFS-3G package for Mac OS X includes a utility called *ntfsfix* (part of the *ntfsprogs* collection), which can repair some fundamental inconsistencies of NTFS volumes.

This utility is currently **not** integrated with the file system bundle, and thus can't be invoked through Disk Utility, because of reports that it has corrupted some users' NTFS drives.

If you want to try it out despite these warnings, you will have to invoke it manually, from the Terminal. Type “man ntfsfix” for usage info.

4. Customizing NTFS-3G

Starting with version 2009.1.1, NTFS-3G includes a preference pane for managing its settings. It can be accessed by starting *System Preferences*, clicking the NTFS-3G icon and authorizing.



4.1 Disabling NTFS-3G

4.1.1 Globally

It is possible to disable NTFS-3G without uninstalling it. In the disabled state, NTFS-3G will ignore all NTFS volumes that are attached to the system, and pass the mount request on to Apple's NTFS driver (or any other installed NTFS driver with a later probe order).

This setting can be controlled with the Disable/Enable button next to *NTFS-3G status* in the preference pane.

The default setting is *enabled*.

4.1.2 For a volume

If you don't need NTFS-3G to be globally disabled, but only need it to ignore a specific volume when attached, you can select the volume in the drop down list in the *Volume options* section of the preference pane and tick the box *Disable NTFS-3G for the volume*.

This setting is governed by the control file:

```
.NTFS-3G/.ntfs-readonly
```

The file only needs to exist in the root of the volume. It should be an empty (0 bytes) file.

4.2 Disabling the *ublio* caching layer

Please note that these instructions only apply to the *ublio* build.

4.2.1 Globally

You can disable/enable the *ublio* caching layer when using the *ublio* build by unticking/ticking the check box *File system caching enabled* in the preference pane.

The default setting for the *ublio* build is *enabled*. The stable build doesn't have this option.

4.2.2 For a volume

To disable the *ublio* layer for one specific volume, select the volume in the *Volume options* section of the preference pane and tick the box *Disable file system caching*.

This setting is governed by the control file:

```
.NTFS-3G/.ntfs-noublio
```

When this file is present in the root of a volume when mounting, *ublio* caching is not activated, and the volume is mounted as if you were using the stable build.

4.3 Disabling file name normalization

NTFS-3G 2009.1.1 brings some changes to how certain international filenames are handled, especially characters that belong to western languages and korean. This is to increase Windows compatibility, but it may lead to problems when reading filenames with western accented or korean characters that have previously been created by Mac OS X / NTFS-3G.

If you experience this, you may want to disable file name normalization to bring back the old NTFS-3G behavior so you can convert your previously created file names to the Windows form.

4.3.1 Globally

Tick the check box *Disable file name normalization* in the preference pane to disable it completely for the entire computer.

The default setting is *enabled*.

4.3.2 For a volume

Select the volume in the drop down list in *Volume options* and tick the box *Disable file name normalization* in the *Settings* section below.

This setting is governed by the control file:

```
.NTFS-3G/.ntfs-nonfconv
```

When this file is present in the root of a volume when mounting, file name normalization is not activated, and the file names are returned as in the previous builds of NTFS-3G.

4.4 Enabling debug logging

If something doesn't seem to work properly in NTFS-3G, or you want to find out what happens when you attach an NTFS-3G volume for troubleshooting purposes, you can enable debug logging to get more information on what happens during the mount process.

Tick the checkbox called *Debug logging enabled* to enable this extended logging.

When debug logging is enabled, the utility script that helps NTFS-3G to mount volumes (and that does a lot of other important things too) writes extensive debug logging information to log files.

Normally, the script writes to the log file “/var/log/ntfs-3g_util.log”, but in case it can't access /var/log (if it's not run with superuser privileges) it falls back to “/var/tmp/<username>_ntfs-3g_util.log”, where <username> is substituted with the user name of the user running the script.

Typically, operations where you create a new NTFS file system are executed with the rights of the currently active user, in which case the log will be written to /var/tmp, while all other operations are executed by root, in which case the log file will be written to /var/log.

Note: The NTFS-3G utility script always tries to log errors, regardless of the setting of this preference value. Debug logging provides extremely fine grained logging suitable for bug reports.

The default setting is *disabled*.

5. Acknowledgments

This package would not exist if it were not for the following projects/individuals:

The [NTFS-3G](#) project, led by Szabolcs Szakacsits

The [MacFUSE](#) project, led by Amit Singh

Csaba Henk and Alejandro Pulver, who integrated FUSE and NTFS-3G into FreeBSD, thus also providing the foundation for NTFS-3G running on Mac OS X

The [FUSE: Filesystem in Userspace](#) project, led by Miklos Szeredi, which provides the foundation for user space file system development

The [Linux-NTFS](#) project, led by Anton Altaparmakov, which provides many important tools used in this package

[Paul Marks](#), who created the first NTFS-3G file system bundle for Mac OS X and documented a lot of aspects on how the DiskArbitration framework handles volumes

...and **lots** of other contributors and testers within the community.

Mac and Mac OS are trademarks of Apple Inc.

Windows is a trademark of Microsoft Corporation.

Other trademarks are the property of their respective owners.