NAME

libntfs-gnomevfs − Module for GNOME VFS that allows access to NTFS filesystems.

OVERVIEW

The GNOME virtual filesystem (VFS) provides universal access to different filesystems. The **libntfs-gnomevfs** module enables GNOME VFS aware clients to seamlessly utilize the NTFS library **libntfs**.

So you can access an NTFS filesystem without needing to use the NTFS utilities themselves (at least in theory anyway). In practice this is probably more useful for programs and programmers to make using **libntfs** easier, more generic, and to allow easier debugging of **libntfs**.

Examples

Prerequisites

To be able to follow these examples you will need to have installed the test utilities from the gnome-vfs-2.4.x package. The easiest way to do this is to download and compile the gnome-vfs-2 package, e.g. download from:

http://ftp.gnome.org/pub/GNOME/desktop/2.4/2.4.0/sources/gnome-vfs-2.4.0.tar.gz

Then run ./configure followed by make and make install (as root). This will install it into /usr/local so it should not conflict with your existing installation from rpm or deb packages which will be in /usr.

Note you may also need to add /usr/local/lib to /etc/ld.so.conf and then run ldconfig (as root) to let your system see the installed gnome-vfs-2.4.x libraries.

Then run ./configure followed by make and make install (as root) in the main **ntfsprogs** directory to build and install the **libntfs-gnomevfs** module and **libntfs** library which is used by the module.

Copying a file from an NTFS partition

To copy the file autoexec.bat from the main directory of an NTFS partition (/dev/hda1) to the /tmp directory on your system you could run:

/path/to/gnome-vfs-2.4.x/test/test-xfer file:///dev/hda1#libntfs:/autoexec.bat /tmp/autoexec.bat

To copy a file from a directory inside the NTFS partition you would just specify the full path. So for example to copy the file win.ini from the Windows directory you would run:

/path/to/gnome-vfs-2.4.x/test/test-xfer file:///dev/hda1#libntfs:/Windows/win.ini /tmp/win.ini

Shell access to an NTFS partition

For debugging it is most useful to be able to do various things to the NTFS partition while it is being operated upon by **libntfs**. This is achieved using the test-shell utility (from the gnome-vfs-2.4.x package) by running: /path/to/gnome-vfs-2.4.x/test/test-shell

This drops you into the GNOME VFS shell from where you can now cd into the NTFS partition (/dev/hda1) by typing: cd file:///dev/hda1#libntfs:/

You are now in the root directory of the NTFS partition. The first thing you will probably want to do is to type "ls" to display the directory contents.

You could then change directories using the "cd" command, e.g. to enter the Windows directory you would type: cd Windows

You can then open files, seek inside files, read from files (write is not enabled at present), etc thus exercising large portions of the NTFS library.

Use the "help" command while in the shell to see the available commands.

BUGS

No bugs are known but there are several limitations at the moment:

You cannot get information about files other than what the "ls" command in the test-shell can give you, i.e. the "info" command in the test-shell does not work.

Further access to the partition is read-only and hence you cannot write to files. This will be changed in the future once the module has had more wide testing.

There may be other limitations and possibly bugs. Please report any problems to the NTFS mailing list: linux-ntfs-dev@lists.sourceforge.net

AUTHORS

The **libntfs-gnomevfs** module was written by Jan Kratochvil. This man page was written by Anton Altaparmakov.

AVAILABILITY

The **ntfsprogs** package which contains the **libntfs-gnomevfs** module can be downloaded from http://linux-ntfs.sourceforge.net/downloads.html These manual pages can be viewed online at http://linux-ntfs.sourceforge.net/man/ntfsprogs.html

SEE ALSO

ntfsprogs(8)

NAME

mkntfs − create a NTFS 1.2 (Windows NT/2000/XP) file system

SYNOPSIS

mkntfs [**−s** *sector-size*] [**−c** *cluster-size*] [**−L** *volume-label*] [**−z** *mft-zone-multiplier*] [**−f** | **−Q**] [**-n**] [**−q**] [**−v**] [**−vv**] [**−C**] [**−F**] [**−I**] [**−V**] [**−l**] [**−h**] *device* [*number-of-sectors*]

DESCRIPTION

mkntfs is used to create a NTFS 1.2 (Windows NT 4.0) file system on a device (usually a disk partition). *device* is the special file corresponding to the device (e.g */dev/hdXX*). *number-of-sectors* is the number of blocks on the device. If omitted, **mkntfs** automagically figures the file system size.

OPTIONS

−s *sector-size*

Specify the size of sectors in bytes. Valid sector size values are 256, 512, 1024, 2048 and 4096 bytes per sector. If omitted, **mkntfs** *sector-size* is determined automatically and if that fails a default of 512 bytes per sector is used.

−c *cluster-size*

Specify the size of clusters in bytes. Valid cluster size values are powers of two, with at least 256, and at most 65536 bytes per cluster. If omitted, **mkntfs** *cluster-size* is determined by the volume size. The value is determined as follows:

Volume	**size**	**Default cluster**
0	- 512MB	512 bytes
512MB	- 1GB	1024 bytes
1GB	- 2GB	2048 bytes
2GB	+	4096 bytes

Note that the default cluster size is set to be at least equal to the sector size as a cluster cannot be smaller than a sector. Also, note that values greater than 4096 have the side effect that compression is disabled on the volume (due to limitations in the NTFS compression algorithm currently in use by Windows).

−L *volume-label*

Set the volume label for the filesystem.

−z *mft-zone-multiplier*

Set the MFT zone multiplier, which determines the size of the MFT zone to use on the volume. The MFT zone is the area at the beginning of the volume reserved for the master file table (MFT), which stores the on disk inodes (MFT records). It is noteworthy that small files are stored entirely within the inode; thus, if you expect to use the volume for storing large numbers of very small files, it is useful to set the zone multiplier to a higher value. Note, that the MFT zone is resized on the fly as required during operation of the NTFS driver but choosing a good value will reduce fragmentation. Valid values are 1, 2, 3 and 4. The values have the following meaning:

MFT zone multiplier	**MFT zone size (% of volume size)**
1	12.5% (default)
2	25.0%
3	37.5%
4	50.0%

−f Same as **−Q**.

−Q Perform quick format. This will skip both zeroing of the volume and bad sector checking.

−n Causes **mkntfs** to not actually create a filesystem, but display what it would do if it were to create a filesystem. All steps of the format are carried out except the actual writing to the device.

−q Quiet execution; only errors are written to stderr, no output to stdout occurs at all. Useful if **mkntfs** is run in a script.

−v Verbose execution.

−vv Really verbose execution; includes the verbose output from the **−v** option as well as additional output useful for debugging **mkntfs.**

−C Enable compression on the volume.

−F Force **mkntfs** to run, even if the specified *device* is not a block special device, or appears to be mounted.

−I Disable content indexing on the volume. (This is only meaningful on Windows 2000 and later. Windows NT 4.0 and earlier ignore this as they do not implement content indexing at all.)

−V Print the version number of **mkntfs** and exit.

−l Print the licensing information of **mkntfs** and exit.

−h Print the usage information of **mkntfs** and exit.

BUGS

mkntfs writes the backup boot sector to the last sector of the block *device* being formatted. However, current versions of the Linux kernel (all versions up to and including todays 2.4.18) either only report an even number of sectors when the sector size is below 1024 bytes, which is the case for most hard drives today (512 bytes sector size) or they return the correct number but accessing the last sector fails. Either way, this means that when a partition has an odd number of 512-byte sectors, the last sector is either not reported to us at all or it is not writable by us and hence the created NTFS volume will either have the backup boot sector placed one sector ahead of where it should be or it cannot be written at all. For this reason, **mkntfs** marks the NTFS volume dirty, so that when you reboot into Windows, check disk runs automatically and creates a copy of the backup boot sector in the correct location. This also has the benefit of catching any bugs in **mkntfs** as check disk would find any corrupt structures and repair them, as well as report them. - If you do see any problems reported, please report the messages to the author.

There may be other bugs. Please, report them to the author.

AUTHOR

This version of **mkntfs** has been written by Anton Altaparmakov <aia21@cantab.net> (if that fails, use <antona@users.sourceforge.net>).

AVAILABILITY

mkntfs is part of the ntfsprogs package and is available for download from http://sourceforge.net/project/showfiles.php?group_id=13956 in source (tar ball and rpm) and pre-compiled binary (i386 rpm and deb) form.

SEE ALSO

badblocks(8), **ntfsprogs**(8)

NAME

ntfscat − concatenate files and print them on the standard output

SYNOPSIS

ntfscat [*options*] **device file**

DESCRIPTION

ntfscat will read a file from an NTFS volume and display the contents on the standard output.

The case of the filename passed to **ntfscat** is ignored.

OPTIONS

Below is a summary of all the options that **ntfscat** accepts. All options have two equivalent names. The short name is preceded by − and the long name is preceded by −−. Any single letter options, that don't take an argument, can be combined into a single command, e.g. **−fv** is equivalent to **−f −v**. Long named options can be abbreviated to any unique prefix of their name.

−f
−−force
This will override some sensible defaults, such as not working with a mounted volume. Use this option with caution.

−h
−−help Show a list of options with a brief description of each one.

−q
−−quiet
Suppress some debug/warning/error messages.

−V
−−version
Show the version number, copyright and license **ntfscat**.

−v
−−vebose
Display more debug/warning/error messages.

EXAMPLES

Display the contents of a file in the root of an NTFS volume.

ntfscat /dev/hda1 boot.ini

Display the contents of a file in a subdirectory of an NTFS volume.

ntfscat /dev/hda1 /winnt/system32/drivers/etc/hosts

BUGS

ntfscat was written in a short time, to get something "out there". It needs a lot more work. If you find any bugs, please send an email to <linux-ntfs-dev@lists.sourceforge.net>

AUTHOR

ntfscat was written by Richard Russon (FlatCap) <ntfs@flatcap.org>
If you find this tool useful, make FlatCap happy and send him an email.

AVAILABILITY

ntfscat is part of the ntfsprogs package and is available from
http://linux−ntfs.sourceforge.net/downloads.html

SEE ALSO

ntfsls(8), **ntfsprogs**(8)

NAME

ntfsclone − Efficiently clone an NTFS filesystem

SYNOPSIS

ntfsclone [**−fhm**] **−o** [*FILE* | **−**] **device**
ntfsclone [**−fhm**] **−O** *FILE* **device**

DESCRIPTION

ntfsclone will efficiently clone (copy, save, backup, restore) an NTFS filesystem to a sparse file, device (partition) or standard output. It works at disk sector level and copies only the used data. Unused disk space becomes zero (cloning to sparse file), left unchanged (cloning to a disk/partition) or filled with zeros (cloning to standard output).

ntfsclone can be useful to make backups, an exact snapshot of an NTFS filesystem and restore it later on, or for developers to test NTFS read/write functionality, troubleshot/investigate users' issues using the clone without the risk of destroying the original filesystem.

The clone is an exact copy of the original NTFS filesystem from sector to sector thus it can be also mounted just like the original NTFS filesystem. For example if you clone to a file and the kernel has loopback device and NTFS support then the file can be mounted as

mount −t ntfs −o loop ntfsclone.img /mnt/ntfsclone

SPARSE FILES

A file is sparse if it has unallocated blocks (holes). The reported size of such files are always higher than the disk space consumed by them. The **du** command can tell the real disk space used by a sparse file. The holes are always read as zeros. All major Linux filesystem like, ext2, ext3, reiserfs, Reiser4, JFS and XFS, supports sparse files but for example the ISO 9600 CD-ROM filesystem doesn't.

HANDLING LARGE SPARSE FILES

As of today Linux provides inadequate support for managing (tar, cp, gzip, gunzip, bzip2, bunzip2, cat, etc) large sparse files. The only main Linux filesystem having support for efficient sparse file handling is XFS by the XFS_IOC_GETBMAPX **ioctl**. However none of the common utility supports it. This means when you tar, cp, gzip, bzip2, etc a large sparse file they will always read the entire file, even if you use the "sparse support" options.

bzip2 compresses large sparse files much better than **gzip** but it does so also much slower. Moreover neither of them handles large sparse files efficiently during uncompression from disk space usage point of view. A possible workaround is if you pipe the uncompressed stream through **cp**, for example this way,

bunzip2 −c image.bz2 | cp −−sparse=always /proc/self/fd/0 image

At present the most efficient way, both speed and space-wise, to compress and uncompress large sparse files by common tools is using **tar** with the options **−S** (handle sparse files "efficiently") and **−j** (filter the archive through bzip2). Altough **tar** still reads and analyses the entire file, it doesn't pass on the large data blocks having only zeros to filters and it also avoids writing large amount of zeros to the disk needlessly. But since **tar** can't create an archive from the standard input, you can't do this in-place by just reading **ntfsclone** standard output.

METADATA-ONLY CLONING

One of the features of **ntfsclone** is it can also save only the NTFS metadata using the option **−m** or **−−metadata** and the clone still will be mountable. In this case all non-metadata file content will be lost and reading them back will result always zeros.

The metadata-only image can be compressed very well, usually to not more than 1-3 MB thus it's relatively easy to transfer it for investigation to NTFS experts.

In this mode of ntfsclone, **NONE** of the user's data is saved, including the resident user's data embedded into metadata. All is filled with zeros. Moreover all the file timestamps, deleted and unused spaces

inside the metadata are filled with zeros. Thus this mode is inappropriate for example for forensic analyses.

Please note, filenames are not wiped out. They might contain sensitive information, so think twice before sending such an image to anybody.

OPTIONS

Below is a summary of all the options that **ntfsclone** accepts. All options have two equivalent names. The short name is preceded by − and the long name is preceded by −−. Any single letter options, that don't take an argument, can be combined into a single command, e.g. **−fm** is equivalent to **−f −m**.

−o, −−output *FILE*
Clone NTFS to the non-existent *FILE*. If *FILE* is '-' then clone to the standard output.

−O, −−overwrite *FILE*
Clone NTFS to *FILE*, overwriting if exists.

−m, −−metadata
Clone **ONLY METADATA** (for NTFS experts). Moreover only cloning to a file is allowed. You can't metadata-only clone to a device or standard output.

−f, −−force
Forces ntfsclone to proceed, overriding some safety checks. You can use this parameter multiply times if you want to overcome every single safety checks.

−h, −−help
Show a list of options with a brief description of each one.

EXAMPLES

Cloning (save, backup) an NTFS volume to a non-existent file

ntfsclone −−output ntfs.img /dev/hda1

Restoring a clone image to its original partition

ntfsclone −−overwrite /dev/hda1 ntfs.img

Space and speed-wise the most efficient way to compress a clone image

tar −cjSf ntfs.img.tar.bz2 ntfs.img

Uncompressing a **tar** archived clone image

tar −xjSf ntfs.img.tar.bz2

In-place compressing an NTFS volume. Note, gzip is faster usually at least 2-4 times but it creates also bigger compressed files.

ntfsclone −−output ntfs.img /dev/hda1 | bzip2 −c > ntfs.img.bz2

Restoring an NTFS volume from a compressed image

bunzip2 −c ntfs.img.bz2 | dd of=/dev/hda1

Backup an NTFS volume to a remote host, using **ssh** default compression.

ntfsclone −o − /dev/hda1 | ssh −C host 'bzip −c9 > ntfs.img.bz2'

Clone an NTFS volume to a remote host, using **ssh** default compression (type everything in one line).

**ntfsclone −o − /dev/hda1 | **
ssh −C host 'cat | cp −−sparse=always /proc/self/fd/0 ntfs.img'

Clone a remote NTFS volume to the local filesystem via **ssh** using a custom compression level (type everything in one line). Speed-wise the optimal compression level depends on your network, disk and CPU speed, saturation.

**ssh host 'ntfsclone −o − /dev/hda1 | gzip −2c' | **
gunzip −c | cp −−sparse=always /proc/self/fd/0 ntfs.img

Pack NTFS metadata for NTFS experts

ntfsclone −−metadata −−output ntfsmeta.img /dev/hda1
tar −cjSf ntfsmeta.img.tar.bz2 ntfsmeta.img

BUGS

This program has no known bugs. If you find one, please send an email to <linux-ntfs-dev@lists.sourceforge.net>.

Sometimes it might appear ntfsclone froze if the clone is on ReiserFS and even CTRL-C won't stop it. This is not a bug in ntfsclone, however it's due to ReiserFS being extremely inefficient creating large sparse files and not handling signals during this operation. This ReiserFS problem was improved in kernel 2.4.22. XFS, JFS and ext3 don't have this problem.

AUTHOR

ntfsclone was written by Szabolcs Szakacsits <szaka@sienet.hu>.

AVAILABILITY

ntfsclone is part of the ntfsprogs package and is available from
http://linux−ntfs.sourceforge.net/downloads.html

SEE ALSO

ntfsresize(8) **ntfsprogs**(8) **xfs_copy**(8) **debugreiserfs**(8) **e2image**(8)

NAME

ntfscluster − identify files in a specified region of an NTFS volume.

SYNOPSIS

ntfscluster [*options*] **device**

DESCRIPTION

ntfscluster has three modes of operation: *info*, *sector* and *cluster*.

Info

The default mode, *info* is currently not implemented. It will display general infomation about the NTFS volume when it is working.

Sector

The *sector* mode will display a list of files that have data in the specified range of sectors.

Cluster

The *cluster* mode will display a list of files that have data in the specified range of clusters. When the cluster size is one sector, this will be equivalent to the *sector* mode of operation.

OPTIONS

Below is a summary of all the options that **ntfscluster** accepts. All options have two equivalent names. The short name is preceded by − and the long name is preceded by −−. Any single letter options, that don't take an argument, can be combined into a single command, e.g. **−fv** is equivalent to **−f −v**. Long named options can be abbreviated to any unique prefix of their name.

−c *range*
−−cluster *range*
: Any files whose data is in this range of clusters will be displayed.

−f
−−force
: This will override some sensible defaults, such as not working with a mounted volume. Use this option with caution.

−−help Show a list of options with a brief description of each one.

−i
−−info This option is not yet implemented.

−q
−−quiet
: Reduce the amount of output to a minimum. Naturally, it doesn't make sense to combine this option with

−s
−−sector
: Any files whose data is in this range of sectors will be displayed.

−v
−−verbose
: Increase the amount of output that **ntfscluster** prints.

−V
−−version
: Show the version number, copyright and license **ntfscluster**.

EXAMPLES

Get some information about the volume /dev/hda1.

ntfscluster /dev/hda1

Look for files in the first 500 clusters of /dev/hda1.

ntfscluster -c 0-500 /dev/hda1

BUGS

The *info* mode isn't implemented yet. This program is quite limited, but it has no known bugs. If you find one, please send an email to <linux-ntfs-dev@lists.sourceforge.net>

AUTHOR

ntfscluster was written by Richard Russon (FlatCap) <ntfs@flatcap.org>
If you find this tool useful, make FlatCap happy and send him an email.

AVAILABILITY

ntfscluster is part of the ntfsprogs package and is available from
http://linux−ntfs.sourceforge.net/downloads.html

SEE ALSO

ntfsinfo(8), **ntfsprogs**(8)

NAME

ntfsfix − tool for fixing NTFS partitions altered by the Linux kernel NTFS driver.

SYNOPSIS

ntfsfix *device*

DESCRIPTION

This manual page documents briefly the **ntfsfix** command.

ntfsfix is a program that fixes NTFS partitions altered in any manner with the Linux NTFS driver. **ntfsfix** is **NOT** a Linux version of chkdsk. It only tries to leave the NTFS partition in a not-so-inconsistent state after the NTFS driver has written to it.

ntfsfix appeared because MS chkdsk is well known for its stupidity when fixing altered partitions. Because the main problems are journal files, **ntfsfix** aims to fix those issues.

Running ntfsfix after mounting NTFS partitions read-write is recommended for reducing the chance of severe data loss when NT/W2K/XP tries to remount the affected partition(s).

In order to use **ntfsfix** you must unmount the NTFS partition, and run ntfsfix device, where device is the NTFS partition. After this, you can safely reboot into NT/W2K/XP. Please note that **ntfsfix** is not a chkdsk-like tool, and so is not guaranteed that it could fix all the alterations provoked by the NTFS driver.

AUTHOR

This manual page was written by David Martínez Moreno <ender@debian.org>, for Debian the GNU/Linux system (but may be used by others).

AVAILABILITY

ntfsfix is part of the ntfsprogs package and is available from http://linux-ntfs.sourceforge.net/.

SEE ALSO

mkntfs(8), **ntfsprogs**(8)

NAME

ntfsinfo − dump a file’s attributes

SYNOPSIS

ntfsinfo *-d device -i inode-number*

DESCRIPTION

ntfsinfo will dump the attributes of inode *inode-number*. Run ntfsinfo without arguments for a full list of options.

AUTHOR

ntfsinfo was written by Matthew J. Fanto (mattjf@uncompiled.com).

AVAILABILITY

ntfsinfo is part of the ntfsprogs package and is available from http://linux-ntfs.sourceforge.net/.

SEE ALSO

ntfsprogs(8)

NAME

ntfslabel − display/change the label on an ntfs file system

SYNOPSIS

ntfslabel *device* [*new-label*]

DESCRIPTION

ntfslabel will display or change the file system label on the ntfs file system located on *device*.

If the optional argument *new-label* is not present, **ntfslabel** will simply display the current file system label.

If the optional argument *new-label* is present, then **ntfslabel** will set the file system label to be *new-label*. NTFS file system labels can be at most 128 Unicode characters long; if *new-label* is longer than 128 Unicode characters, **ntfslabel** will truncate it and print a warning message.

It is also possible to set the file system label using the **−L** option of **mkntfs**(8) during creation of the file system.

AUTHOR

ntfslabel was written by Matthew J. Fanto (mattjf@uncompiled.com). This man page was written by Anton Altaparmakov (aia21@cantab.net).

AVAILABILITY

ntfslabel is part of the ntfsprogs package and is available from http://linux-ntfs.sourceforge.net/.

SEE ALSO

mkntfs(8), **ntfsprogs**(8)

NAME

ntfsls − list directory contents on an NTFS filesystem

SYNOPSIS

ntfsls [**−a** | **−−all**] [**−F** | **−−classify**] [**−f** | **−−force**] [**−h** | **−?** | **−−help**] [**−i** | **−−inode**] [**−l** | **−−long**] [**−p** | **−−path** *PATH*] [**−q** | **−−quiet**] [**−s** | **−−system**] [**−V** | **−−version**] [**−v** | **−−verbose**] [**−x** | **−−dos**] **−d** | **−−device** *DEVICE*

DESCRIPTION

ntfsls is used to list information about the files specified by the *PATH* option (the root directory by default). *DEVICE* is the special file corresponding to the device (e.g */dev/hdXX*).

OPTIONS

−a, −−all
Display all files. If this option is not specified file names in the POSIX namespace will not be displayed.

−F, −−classify
Append indicator (one of */=@|) to entries.

−f, −−force
Force execution. For example necessary to run on an NTFS partition stored in a normal file.

−h, −?, −−help
Print the usage information of **ntfsls** and exit.

−i, −−inode
Print inode number of each file. This is the MFT reference number in NTFS terminology.

−l, −−long
Use a long listing format.

−p, −−path
The directory whose contents to list or the file (including the path) about which to display information.

−q, −−quiet
Suppress some debug/warning/error messages.

−s, −−system
Unless this options is specified, all files beginning with a dollar sign character will not be listed as these files are usually system files.

−V, −−version
Print the version number of **ntfsls** and exit.

−v, −−verbose
Display more debug/warning/error messages.

−x, −−dos
Display short file names, i.e. files in the DOS namespace, instead of long file names, i.e. files in the WIN32 namespace.

−d, −−device *DEVICE*
The special file corresponding to the device that contains the NTFS partition to read. If you want to use an image of an NTFS partition stored on a normal file, you will also need to specify the **−f** or **−−force** options.

BUGS

We are not aware of any bugs. If you find a bug, please report it to <linux-ntfs-dev@lists.sourceforge.net>. Thank you.

AUTHOR

This version of **ntfsls** has been written by Lode Leroy <lode_leroy@hotmail.com> and enhanced by Anton Altaparmakov <aia21@cantab.net>. This man page has been written by Anton Altaparmakov.

AVAILABILITY

ntfsls is part of the ntfsprogs package and is available for download from http://sourceforge.net/project/showfiles.php?group_id=13956 in source (tar ball and rpm) and pre-compiled binary (i386 rpm and deb) form.

SEE ALSO

ntfsprogs(8)

NAME

ntfstools − several tools for doing neat things with NTFS partitions

OVERVIEW

ntfsprogs is a suite of NTFS utilities based around a shared library. The tools are available for free and come with full source code.

TOOLS

mkntfs

mkntfs(8) : Format a partition using NTFS.

ntfscat

ntfscat(8) : Dump a file's contents to the standard output.

ntfsclone

ntfsclone(8) : Efficiently create/restore an image of an NTFS partition.

ntfscluster

ntfscluster(8) : Locate the owner of any given sector or cluster on an NTFS partition.

ntfsfix

ntfsfix(8) : Clear the LogFile of a partition to make Windows perform a thorough check next time it boots.

ntfsinfo

ntfsinfo(8) : Show some information about an NTFS partition or one of the files or directories within it.

ntfslabel

ntfslabel(8) : Show, or set, an NTFS partition's volume label.

ntfsls

ntfsls(8) : List information about files in a directory residing on an NTFS partition.

ntfsresize

ntfsresize(8) : Resize an NTFS partition without losing data.

ntfsundelete

ntfsundelete(8) : Recover deleted files from an NTFS partition.

AUTHORS

The tools have been written by Anton Altaparmakov, Richard Russon, Matthew Fanto, Szabolcs Szakacsits, and Lode Leroy.

AVAILABILITY

The **ntfsprogs** can be downloaded from http://linux-ntfs.sourceforge.net/downloads.html
These manual pages can be viewed online at http://linux-ntfs.sourceforge.net/man/ntfsprogs.html

SEE ALSO

libntfs-gnomevfs(8)

NAME

ntfsresize − resize an NTFS filesystem without data loss

SYNOPSIS

ntfsresize [**OPTIONS**] **--info** *device*
ntfsresize [**OPTIONS**] [**−−size** *size*[**k**|**M**|**G**]] *device*

DESCRIPTION

The **ntfsresize** program non-destructively resizes Windows XP/2000/NT4, Windows Server 2003 or Longhorn Beta NTFS filesystems. It can be used to shrink or enlarge any NTFS filesystem located on an unmounted *device* (usually a disk partition). The new filesystem will have *size* bytes. The *size* parameter may have one of the optional modifiers **k**, **M**, **G**, which means the *size* parameter is given in kilo-, mega- or gigabytes respectively. **ntfsresize** conforms to the SI, ATA, IEEE standards and the disk manufacturers by using k=10^3, M=10^6 and G=10^9.

If both **--info** and **--size** are omitted then the NTFS filesystem will be enlarged to the underlying device size.

The **ntfsresize** program doesn't manipulate the size of partitions. To do that you have to use a disk partitioning tool, for example **fdisk**(8).

IMPORTANT! Generally it's a good practice making regular backups of your valuable data, especially before using any partitioning tools. To do so for NTFS, you could use **ntfsclone**(8). It's also included in the **ntfsprogs**(8) package.

SHRINKAGE

If you wish to shrink an NTFS partition, first use **ntfsresize** to shrink the size of the filesystem. Then you may use **fdisk**(8) to shrink the size of the partition by deleting the partition and recreating it with the smaller size. But be careful, do not make the partition smaller than the new size of the NTFS filesystem otherwise you won't be able to boot and you might lose your data.

ENLARGEMENT

To enlarge an NTFS filesystem, first you must enlarge the size of the underlying partition. This can be done using **fdisk**(8) by deleting the partition and recreating it with a larger size. Make sure it will not overlap with an other existing partition. Then you may use **ntfsresize** to enlarge the size of the filesystem.

PARTITIONING

When recreating the partition by a disk partitioning tool, make sure you create it with the same starting disk cylinder (sector) and partition type as before. Otherwise you may lose your entire filesystem.

Also make sure you set the bootable flag for the partition if it existed before. Failing to do so you might not be able to boot your computer from the disk.

OPTIONS

Below is a summary of all the options that **ntfsresize** accepts. All options have two equivalent names. The short name is preceded by − and the long name is preceded by −−. Any single letter options, that don't take an argument, can be combined into a single command, e.g. **−fi** is equivalent to **−f −i**.

-i, --info

By using this option ntfsresize will determine the theoretically smallest shrunken filesystem size supported. Most of the time the result is the space already used on the filesystem. Ntfsresize will refuse shrinking to a smaller size than what you got by this option and depending on several factors it might be unable to shrink very close to this theoretical size. Although the integrity of your data should be never in risk, it's still strongly recommended to make a test run by using the **--no-action** option before real resizing.

Practically the smallest shrunken size generally is at around "used space" + (20-200 MB). Please also take into account that Windows might need about 50-100 MB free space left to boot safely.

This option never causes any changes to the filesystem, the partition is opened read-only.

-s, --size *size*[**k**|**M**|**G**]

Resize filesystem to *size*[**k**|**M**|**G**] bytes. The optional modifiers **k**, **M**, **G** mean the *size* parameter is given in kilo-, mega- or gigabytes respectively. Conforming to standards, k=10^3, M=10^6 and G=10^9. Use this option with **--no-action** first.

-f, --force

Forces ntfsresize to proceed with the resize operation if the filesystem is marked "dirty" for consistency check.

Please note, ntfsresize always marks the filesystem "dirty" before a real resize operation and it leaves that way for extra safety. Thus if NTFS was marked by ntfsresize then it's safe to use this option. If you need to resize several times without booting into Windows between each resizing steps then you must use this option.

-n, --no-action

Use this option to make a test run before doing the real resize operation. Volume will be opened read-only and **ntfsresize** displays what it would do if it were to resize the filesystem. Continue with the real resizing only if the test run passed.

-P, --no-progress-bar

Don't show progress bars.

-v, --verbose

More output.

-h, --help

Display help and exit.

EXIT CODES

The exit code is 0 on success, non-zero otherwise.

KNOWN ISSUES

No reliability problems are known or has been reported. If you need help please try the ntfsresize FAQ first (see below) and if you don't find your answer then send your question, comment or bug report to <linux-ntfs-dev@lists.sourceforge.net>. No subscription is needed but the mailing list is moderated and it can take some time to approve your post.

There are some very rarely met limitations at present: filesystems having bad sectors, highly fragmented Master File Table (MFT), relocation of the first MFT extent and resizing in the middle of some metadata in some cases aren't supported yet. These cases are detected and resizing is refused, restricted to a safe size or the closest safe size is displayed.

ntfsresize schedules an NTFS consistency check and after the first boot into Windows you must see **chkdsk** running on a blue background. This is intentional. Windows may force a quick reboot after the consistency check. Moreover after repartitioning your disk and depending on the hardware configuration, the Windows message **System Settings Change** may also appear. Just acknowledge it and reboot again.

AUTHOR

ntfsresize has been written by Szabolcs Szakacsits <szaka@sienet.hu>.

ACKNOWLEDGEMENT

Many thanks to Anton Altaparmakov and Richard Russon for libntfs, the excellent documentation and comments, to Gergely Madarasz, Dewey M. Sasser and Miguel Lastra and his colleagues at the University of Granada for their continuous and highly valuable help, furthermore to Erik Meade, Martin Fick, Sandro Hawke, Dave Croal, Lorrin Nelson, Geert Hendrickx, Robert Bjorkman and Richard Burdick for beta testing and to Theodore Ts'o whose **resize2fs**(8) man page formed the basis of this page.

AVAILABILITY

ntfsresize is part of the **ntfsprogs**(8) package and is available from http://linux-ntfs.sourceforge.net/ as source and precompiled binary. **ntfsresize** related news, example of usage, troubleshooting, statically linked binary and FAQ (frequently asked questions) is maintained at
http://mlf.linux.rulez.org/mlf/ezaz/ntfsresize.html

SEE ALSO

fdisk(8), **cfdisk**(8), **sfdisk**(8), **parted**(8), **mkntfs**(8), **ntfsclone**(8), **ntfsprogs**(8)

NAME

ntfsundelete − recover a deleted file from an NTFS volume.

SYNOPSIS

ntfsundelete [*options*] **device**

DESCRIPTION

ntfsundelete has three modes of operation: *scan*, *undelete* and *copy*.

Scan

The default mode, *scan* simply reads an NTFS Volume and looks for files that have been deleted. Then it will print a list giving the inode number, name and size.

Undelete

The *undelete* mode takes the inode and recovers as much of the data as possible. It saves the result to another location. Partly for safety, but mostly because NTFS write support isn't finished.

Copy

This is a wizard's option. It will save a portion of the MFT to a file. This probably only be useful when debugging *ntfsundelete*

Notes

ntfsundelete only ever **reads** from the NTFS Volume. **ntfsundelete** will never change the volume.

CAVEATS

Miracles

ntfsundelete cannot perform the impossible.

When a file is deleted the MFT Record is marked as not in use and the bitmap representing the disk usage is updated. If the power isn't turned off immediately, the free space, where the file used to live, may become overwritten. Worse, the MFT Record may be reused for another file. If this happens it is impossible to tell where the file was on disk.

Even if all the clusters of a file are not in use, there is no guarantee that they haven't been overwritten by some short−lived file.

Locale

In NTFS all the filenames are stored as Unicode. They will be converted into the current locale for display by **ntfsundelete**. The utility has successfully displayed some Chinese pictogram filenames and then correctly recovered them.

Extended MFT Records

In rare circumstances, a single MFT Record will not be large enough to hold the metadata describing a file (a file would have to be in hundreds of fragments for this to happen). In these cases one MFT record may hold the filename, but another will hold the information about the data. **ntfsundelete** will not try and piece together such records. It will simply show unnamed files with data.

Compressed and Encrypted Files

ntfsundelete cannot recover compressed or encrypted files. When scanning for them, it will display as being 0% recoverable.

The Recovered File's Size and Date

To recover a file **ntfsundelete** has to read the file's metadata. Unfortunately, this isn't always intact. When a file is deleted, the metadata can be left in an inconsistant state. e.g. the file size may be zero; the dates of the file may be set to the time it was deleted, or random.
To be safe **ntfsundelete** will pick the largest file size it finds and write that to disk. It will also try and set the file's date to the last modified date. This date may be the correct last modified date, or something unexpected.

OPTIONS

Below is a summary of all the options that **ntfsundelete** accepts. All options have two equivalent names. The short name is preceded by − and the long name is preceded by −−. Any single letter options, that don't take an argument, can be combined into a single command, e.g. **−fv** is equivalent to **−f −v**. Long named options can be abbreviated to any unique prefix of their name.

−b *num*
−−byte *num*
If any clusters of the file cannot be recovered, the missing parts will be filled with this byte. The default is zeros.

−C
−−case When scanning an NTFS volume, any filename matching (using the **−−match** option) is case−insensitive. This option makes the maching case−sensitive.

−c *range*
−−copy *range*
This wizard's option will write a block of MFT FILE records to a file. The default file is *mft* which will be created in the current directory. This option can be combined with the **−−output** and **−−destination** options.

−d *dir*
−−destination *dir*
This option controls where to put the output file of the **−−undelete** and **−−copy** options.

−f
−−force
This will override some sensible defaults, such as not overwriting an existing file. Use this option with caution.

−h
−−help Show a list of options with a brief description of each one.

−m *pattern*
−−match *pattern*
Filter the output of the **−−scan** option, by only looking for matching filenames. The pattern can include the wildcards '?', match exactly one character or '*', match zero or more characters. By default the matching is case−insensitive. To make the search case sensitive, use the **−−case** option.

−o *file*
−−output *file*
Use this option to set name of output file that **−−undelete** or **−−copy** will create.

−p *num*
−−percentage *num*
Filter the output of the **−−scan** option, by only matching files with a certain amount of recoverable content. **Please read the caveats section for more details.**

−q
−−quiet
Reduce the amount of output to a minimum. Naturally, it doesn't make sense to combine this option with **−−scan**.

−s
−−scan Search through an NTFS volume and print a list of files that could be recovered. This is the default action of **ntfsundelete**. This list can be filtered by filename, size, percentage recoverable or last modification time, using the **−−match**, **−−size**, **−−percent** and **−−time** options, respectively.

The output of scan will be:

```
Inode  Flags  %age    Date      Size  Filename
 6038  FN..    93%  2002-07-17  26629  thesis.doc
```

Flag	**Description**
F/D	File/Directory
N/R	(Non-)Resident data stream
C/E	Compressed/Encrypted data stream
!	Missing attributes

The percentage field shows how much of the file can potentially be recovered.

−S *range*
−−size *range*
Filter the output of the **−−scan** option, by looking for a particular range of file sizes. The range may be specified as two numbers separated by a '−'. The sizes may be abbreviated using the suffixes k, m, g, t, for kilobytes, megabytes, gigabytes and terabytes respectively.

−t *since*
−−time *since*
Filter the output of the **−−scan** option. Only match files that have been altered since this time. The time must be given as number using a suffix of d, w, m, y for days, weeks, months or years ago.

−u *num*
−−undelete *num*
Recover the file with this inode number. This option can be combined with **−−output**, **−−destination**, and **−−byte**.

When the file is recovered it will be given its original name, unless the **−−output** option is used.

−v
−−verbose
Increase the amount of output that **ntfsundelete** prints.

−V
−−version
Show the version number, copyright and license **ntfsundelete**.

EXAMPLES

Look for deleted files on /dev/hda1.

ntfsundelete /dev/hda1

Look for deleted documents on /dev/hda1.

ntfsundelete /dev/hda1 -s −m '*.doc'

Look for deleted files between 5000 and 6000000 bytes, with at least 90% of the data recoverable, on /dev/hda1.

ntfsundelete /dev/hda1 −S 5k−6m −p 90

Look for deleted files altered in the last two days

ntfsundelete /dev/hda1 −t 2d

Undelete inode number 3689, call the file 'work.doc' and put it in the user's home directory.

ntfsundelete /dev/hda1 −u 3689 −o work.doc −d ~

Save MFT Records 3689 to 3690 to a file 'debug'

ntfsundelete /dev/hda1 −c 3689−3690 −o debug

BUGS

There are some small limitations to this program, but currently no known bugs. If you find one, please send an email to <linux-ntfs-dev@lists.sourceforge.net>

AUTHOR

ntfsundelete was written by Richard Russon (FlatCap) <ntfs@flatcap.org>
If you find this tool useful, make FlatCap happy and send him an email.

AVAILABILITY

ntfsundelete is part of the ntfsprogs package and is available from http://linux−ntfs.sourceforge.net/downloads.html

SEE ALSO

ntfsinfo(8), **ntfsprogs**(8)