NAME

wimlib-imagex-mount, wimlib-imagex-mountrw, wimlib-imagex-unmount − Mount and unmount an image from a WIM archive

SYNOPSIS

wimlib-imagex mount *WIMFILE* [*IMAGE*] *DIRECTORY* [*OPTION*...]
wimlib-imagex mountrw *WIMFILE* [*IMAGE*] *DIRECTORY* [*OPTION*...]
wimlib-imagex unmount *DIRECTORY* [--commit] [--check] [--rebuild]

DESCRIPTION

The **wimlib-imagex mount** and **wimlib-imagex mountrw** commands will mount the image in the Windows Imaging (WIM) file *WIMFILE* specified by *IMAGE* on the directory *DIRECTORY* using FUSE (Filesystem in Userspace). **wimlib-imagex mount** will mount the image read-only, while **wimlib-imagex mountrw** will mount the image read-write.

IMAGE may be a 1-based index of the image in the WIM to mount, or it may be the name of an image in the WIM. Use the **wimlib-imagex info** (1) command to see the available images in the WIM. *IMAGE* may be omitted if *WIMFILE* contains only one image.

The WIM image can be unmounted using the **wimlib-imagex unmount** command. Changes made to a WIM mounted read-write will be discarded unless the **--commit** flag is provided to **wimlib-imagex unmount**.

SPLIT WIMS

You may use **wimlib-imagex mount** to mount an image from a split WIM read-only. However, you may not mount an image from a split WIM read-write.

The *WIMFILE* argument is used to specify the first part of the split WIM, and the **--refs**="*GLOB*" option is used to provide a shell-style file glob that specifies the additional parts of the split WIM. *GLOB* is expected to be a single string on the command line, so *GLOB* must be quoted so that it is protected against shell expansion. *GLOB* must expand to all parts of the split WIM, except optionally the first part which may either omitted or included in the glob (but the first part MUST be specified as *WIMFILE* as well).

Here's an example. The names for the split WIMs usually go something like:

```
mywim.swm
mywim2.swm
mywim3.swm
mywim4.swm
mywim5.swm
```

To mount the first image of this split WIM to the directory "dir", we would do:

```
wimlib-imagex mount mywim.swm 1 dir --ref="mywim*.swm"
```

NOTES

If wimlib was configured using the **--without-fuse** flag, then the **wimlib-imagex mount**, **wimlib-imagex mountrw**, and **wimlib-imagex unmount** commands will not work. Also, these commands are not available in the Windows builds of wimlib.

You can mount multiple images from a WIM file read-only at the same time, but you can only mount one image at a time from a WIM read-write.

All files in the mounted WIM will be accessible regardless of whether there is a security descriptor in the WIM associated with the file or not. New files or directories created in a read-write mounted WIM will be created with no security descriptor. Although there is support for accessing named data streams (see the **--streams-interface** option), it is currently not possible to set or get DOS names, file attributes, or security descriptors in a mounted WIM.

By default, changes to a read-write WIM are made in-place by appending to the WIM. This is nice for big WIM files, since the entire file doesn't have to be rebuilt to make a small change. But, if you are making many changes to a read-write mounted WIM, especially deleting large files, it is suggested to provide the **--rebuild** option to **wimlib-imagex unmount** to force the WIM to be rebuilt, or else run **wimlib-imagex**

optimize on the WIM afterwards.

MOUNT OPTIONS

--check

When reading the WIM, verify its integrity if it contains an integrity table.

--streams-interface=*INTERFACE*

This option is inspired by the ntfs-3g filesystem driver (see **ntfs-3g** (8)). It controls how alternate data streams, or named data streams, in WIM files are made available.

If "none", it will be impossible to read or write the named data streams.

If "xattr" (default), named data streams will be accessible through extended file attributes, unless this support was disabled when compiling wimlib. The named data streams may be accessed through extended attributes named "user.*", where the * is the name of the named data stream. See **setfattr** (1) and **getfattr** (1).

If "windows", the named data streams will be accessible by specifying the filename, then a colon, then the name of the named data stream; for example, "myfile:mystream".

Please note that named data streams are a somewhat obscure NTFS feature that aren't actually used much, even though they complicate the WIM file format considerably. Normally, all you care about is the default or "unnamed" data stream.

--debug

Turn on debugging information printed by the FUSE library, and do not fork into the background.

--ref="*GLOB*"

File glob of additional split WIM parts that are part of the split WIM being mounted. This option is valid for **wimlib-imagex mount** but not **wimlib-imagex mountrw**. See **SPLIT_WIMS**.

--staging-dir=*DIR*

Store temporary staging files in a subdirectory of the directory *DIR*. Only valid for **wimlib-imagex mountrw**.

--unix-data

By default, **wimlib-imagex mount** and **wimlib-imagex mountrw** will ignore both Windows-style security descriptors (which may have been set either from Windows or by **wimlib-imagex capture** from a NTFS-volume) and UNIX-specific data (which is from using **wimlib-imagex capture** with the **--unix-data** flag). In this default mode, all files will simply be owned by the user running **wimlib-imagex** and will have mode 0777. (Note: they will still not be accessible to other users unless you also specify **--allow-other**.) If you instead provide the **--unix-data** flag, these default permissions will be overridden on a per-file basis with the UNIX-specific data when available, and in the case of **wimlib-imagex mountrw** it will be possible to change the UNIX permissions using the standard UNIX tools and functions.

--allow-other

Pass the **allow_other** option to the FUSE mount. See **mount.fuse** (8). Note: to do this is a non-root user, **user_allow_other** needs to be specified in /etc/fuse.conf (with the FUSE implementation on Linux, at least).

UNMOUNT OPTIONS

--commit

Update the WIM file with the changes that have been made. Has no effect if the mount is read-only.

--check

When writing *WIMFILE*, include an integrity table. Has no effect if the mount is read-only or if **--commit** was not specified.

--rebuild

Rebuild the entire WIM rather than appending any new data to the end of it. Rebuilding the WIM is slower, but will save a little bit of space that would otherwise be left as a hole in the WIM. Even

more space will be saved if the read-write mount resulted in streams being deleted from the WIM. Also see

--lazy Pass the **-z** option to **fusermount**, which performs a "lazy" unmount where the filesystem is detached immediately even if it is still busy. However, even with this option, **wimlib-imagex unmount** still waits for the filesystem to become unbusy; **--lazy** will only stop the unmount from immediately failing.

IMPLEMENTATION DETAILS

Since a WIM is an archive and not a filesystem, **wimlib-imagex mountrw** creates a temporary staging directory to contain files that are created or modified. This directory is located in the same directory as *WIMFILE* by default, but the location can be set using the **--staging-dir** option. When the filesystem is unmounted with **--commit**, the WIM is modified in-place (or rebuild completely with **--rebuild**), merging in the staging files as needed. Then, the temporary staging directory is deleted.

wimlib-imagex unmount runs in a separate process from the process that previously ran **wimlib-imagex mount**, and these two processes communicate using POSIX message queues. See *src/mount_image.c* in the sources for details. Note: As of wimlib v1.2.1, **wimlib-imagex unmount** correctly fails with an error within a reasonable amount of time (1 second) if the filesystem daemon is abnormally terminated.

SEE ALSO

wimlib-imagex(1)