NAME

wimlib-imagex − Extract, create, modify, or mount a WIM archive

SYNOPSIS

wimlib-imagex append *arguments...* (or **wimappend** *arguments...*)
wimlib-imagex apply *arguments...* (or **wimapply** *arguments...*)
wimlib-imagex capture *arguments...* (or **wimcapture** *arguments...*)
wimlib-imagex delete *arguments...* (or **wimdelete** *arguments...*)
wimlib-imagex dir *arguments...* (or **wimdir** *arguments...*)
wimlib-imagex export *arguments...* (or **wimexport** *arguments...*)
wimlib-imagex extract *arguments...* (or **wimextract** *arguments...*)
wimlib-imagex info *arguments...* (or **wiminfo** *arguments...*)
wimlib-imagex join *arguments...* (or **wimjoin** *arguments...*)
wimlib-imagex mount *arguments...* (or **wimmount** *arguments...*)
wimlib-imagex mountrw *arguments...* (or **wimmountrw** *arguments...*)
wimlib-imagex optimize *arguments...* (or **wimoptimize** *arguments...*)
wimlib-imagex split *arguments...* (or **wimsplit** *arguments...*)
wimlib-imagex unmount *arguments...* (or **wimunmount** *arguments...*)
wimlib-imagex update *arguments...* (or **wimupdate** *arguments...*)
wimlib-imagex verify *arguments...* (or **wimverify** *arguments...*)

DESCRIPTION

wimlib-imagex deals with archive files in the Windows Imaging (WIM) format. Its interface is similar to Microsoft's ImageX, but **wimlib-imagex** is cross-platform and has useful improvements and extensions.

To do its work, **wimlib-imagex** uses **wimlib**, an open source C library that provides interfaces for manipulating WIM archives. wimlib is completely independent from the equivalent Microsoft implementation (WIMGAPI, or wimgapi.dll). You can use wimlib in your own programs, although for command-line use **wimlib-imagex** already provides access to most of wimlib's functionality.

BACKGROUND INFORMATION

The Windows Imaging (WIM) format was designed by Microsoft primarily for archiving Windows filesystems, such as NTFS. However, it can be used on other platforms as well, with some limitations. A WIM archive contains one or more images, each of which is a logically independent directory tree. Images are indexed starting from 1, and each may also have a name. File data is stored as content-addressable "blobs" that are deduplicated across the entire archive. Data may be compressed using one of several compression algorithms.

An update of the WIM format which Microsoft released with Windows 8 uses solid-mode LZMS compression to achieve a better compression ratio. Such files are also called "ESD files" and may have the .esd extension instead of .wim. wimlib fully supports these files except when they are encrypted.

COMMANDS

wimlib-imagex accepts one of a number of commands (listed above in **SYNOPSYS**), and additional arguments depending on the specific command. Although **wimlib-imagex** will print usage information with **--help** or if you invoke it incorrectly, the full documentation for each **wimlib-imagex** command can be found in the appropriate manual page.

Note: if appropriate hard links or batch files have been installed, a command **wimlib-imagex** *COMMAND* can also be accessed as simply **wim***COMMAND*; for example, **wimapply** for **wimlib-imagex apply**. For brevity the documentation uses the shorter names.

GENERAL FEATURES

The following are some of the general features, or use cases, currently supported by **wimlib-imagex**, and pointers to the relevant commands:

- Display information about a WIM file (**wiminfo**)
- List the files in a WIM image (**wimdir**)

- Extract, or "apply", a full WIM image (**wimapply**)
- Extract files or directories from a WIM image (**wimextract**)
- Capture a WIM image and save it to a new WIM file (**wimcapture**)
- Capture a WIM image and append it to an existing WIM file (**wimappend**)
- Modify a WIM image by adding, deleting, or renaming files (**wimupdate**)
- (Linux only) Mount a WIM image read-only (**wimmount**)
- (Linux only) Mount a WIM image read-write (**wimmountrw**)
- Delete an image from a WIM file (**wimdelete**)
- Export image(s) from a WIM file (**wimexport**)
- Change the name or description of a WIM image (**wiminfo**)
- Change the bootable image index of a WIM file (**wiminfo**)
- Rebuild, and optionally recompress, a WIM file (**wimoptimize**)
- Split a WIM file into multiple parts (**wimsplit**)
- Join a split WIM (**wimjoin**)
- Verify the validity and integrity of a WIM file (**wimverify**)

DETAILED FEATURES

This section presents some of the interesting features of **wimlib-imagex** in more detail.

- Multi-platform support. **wimlib-imagex** is supported on both UNIX-like systems (mainly Linux, but also FreeBSD, Mac OS X, etc.) and Windows. Most code is shared among all platforms, but platform-specific features are still supported when possible.
- XPRESS, LZX, and LZMS compression and decompression. wimlib contains advanced implementations of all these compression algorithms. These have been improved over time and now usually outperform and outcompress their Microsoft equivalents, while remaining fully compatible.
- Solid-mode compression, or "ESD file", support. "ESD files" are an updated WIM format that uses solid LZMS compression to achieve a better compression ratio.
- Multithreaded compression. By default, wimlib's data compression is multithreaded and will use all available processors.
- On UNIX-like systems, integration with libntfs-3g allows capturing a WIM image directly from an NTFS volume, or applying a WIM image directly to an NTFS volume. This allows saving and restoring NTFS-specific data and metadata, such as security descriptors and named data streams, which would otherwise only be supported on Windows.
- On UNIX-like systems, optional support for saving and restoring standard UNIX file permissions (owner/group/mode), UNIX special files, and extended attributes. (This is a wimlib extension; Microsoft's WIM software ignores this extra information.)
- On Linux, support for mounting WIM images with FUSE (Filesystem in UserSpacE), both readonly and read-write.
- Split WIMs. A split WIM is a WIM archive split into multiple parts. **wimsplit** can create a split WIM from a standalone WIM, and **wimjoin** can create a standalone WIM from a split WIM.
- Delta WIMs. A delta WIM contains image metadata but excludes file data already present in another WIM file. A delta WIM can be created using **wimcapture** with the **--delta-from** option.
- "Pipable" WIMs. As a wimlib extension (not compatible with the Microsoft implementation), **wimcapture** supports capturing a WIM file to standard output in a special "pipable" format which can later be applied by sending it to **wimapply** on standard input. Among other things, this can be used to pipe images to or from a server over the network to implement fast filesystem imaging and restore.

- Support for WIM integrity tables. Although file data in WIM archives is always checksummed, there can also be an extra set of checksums (an "integrity table") associated with the WIM file itself to provide extra integrity assurance. The **--check** option to several **wimlib-imagex** commands can be used to verify or add these extra checksums.
- Fast incremental backups. Because WIM archives use content-addressible file data, the contents of files are automatically deduplicated. In addition, using the **--update-of** option of **wimcapture** or **wimappend**, you can optimize an image capture so that files that are unmodified based on timestamps are not even read from disk.
- Windows-specific image metadata support. When capturing an image of a Windows operating system, wimlib will automatically populate XML metadata fields such as the Windows OS version details by scanning well-known system files.
- WIMBoot support. On Windows 8.1 and later, files can be "externally backed" by a WIM archive with the help of Microsoft's Windows Overlay Filesystem (WOF) filter driver. With the **--wimboot** option, **wimapply** will extract "pointer files" to the WIM archive rather than the files themselves.
- VSS snapshot support. On Windows, **wimcapture** or **wimappend** with the **--snapshot** option will automatically create a temporary VSS snapshot and capture the image from it. This can be used to image a "live" Windows system.
- Long path support on Windows. **wimlib-imagex** can capture and apply files with paths exceeding the MAX_PATH (260 character) limitation of the Win32 subsystem.
- Non-Administrator support on Windows. You can run **wimlib-imagex** without Administrator rights, subject to some limitations.

COMMON OPTIONS

The following options work for all **wimlib-imagex** commands:

--help Display the help, then exit.

--version
Display the version and legal information, then exit.

--quiet
Suppress informational and progress messages.

CASE SENSITIVITY

By default, the case sensitivity of **wimlib-imagex** differs somewhat between UNIX-like systems and Windows. WIM images may (but usually do not) have multiple files with the same case-insensitive name. Internally, wimlib stores filenames as case-sensitive, but on Windows paths actually provided by the user for use in a WIM image (e.g. for extracting, adding, renaming, or deleting files) will by default be treated as case-insensitive in order to get the "expected" behavior. This differs from the default behavior on UNIX-like systems, where such paths will be treated as case-sensitive.

Note that with case insensitivity, a path component may in general be ambiguous due to multiple files or directories having the same case-insensitive name. In such cases, if there is a file or directory with an exactly matching name, it is chosen; otherwise, one of the case-insensitively matching file or directories is chosen arbitrarily.

The default case sensitivity of **wimlib-imagex** can be overridden by explicitly setting the environmental variable **WIMLIB_IMAGEX_IGNORE_CASE** to 1, in which case such paths will be treated case insensitively, or 0, in which such paths will be treated case sensitively.

Regardless of these settings, options and non-path arguments must be specified in lower case.

LICENSE

REPORTING BUGS

Report bugs to *https://wimlib.net/forums/*. Feedback and suggestions are also welcome.

SEE ALSO

wimappend(1), **wimapply**(1), **wimcapture**(1), **wimdelete**(1), **wimdir**(1), **wimexport**(1), **wimextract**(1), **wiminfo**(1), **wimjoin**(1), **wimmount**(1), **wimmountrw**(1), **wimoptimize**(1), **wimsplit**(1), **wimunmount**(1), **wimupdate**(1), **wimverify**(1),