NAME

wimlib-imagex − Extract, create, modify, or mount a WIM (Windows Imaging Format) archive

SYNOPSIS

wimlib-imagex append *arguments...*
wimlib-imagex apply *arguments...*
wimlib-imagex capture *arguments...*
wimlib-imagex delete *arguments...*
wimlib-imagex dir *arguments...*
wimlib-imagex export *arguments...*
wimlib-imagex extract *arguments...*
wimlib-imagex info *arguments...*
wimlib-imagex join *arguments...*
wimlib-imagex mount *arguments...*
wimlib-imagex mountrw *arguments...*
wimlib-imagex optimize *arguments...*
wimlib-imagex split *arguments...*
wimlib-imagex unmount *arguments...*
wimlib-imagex update *arguments...*
wimlib-imagex verify *arguments...*

DESCRIPTION

wimlib-imagex deals with archives in the Windows Imaging Format (WIM). 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 Format (WIM) 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 released with Windows 8 features solid compression using the LZMS compression algorithm. Such files are also called "ESD files" and may carry the extension .esd instead of .wim. **wimlib-imagex** v1.6.0 and later supports these new files, unless 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: to save typing, 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**.

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
  (**wimlib-imagex info**)

- List the files in a WIM image
  (**wimlib-imagex dir**)
- Extract, or "apply", a full WIM image
  (**wimlib-imagex apply**)
- Extract files or directories from a WIM image
  (**wimlib-imagex extract**)
- Capture a WIM image and save it to a new WIM file
  (**wimlib-imagex capture**)
- Capture a WIM image and append it to an existing WIM file
  (**wimlib-imagex append**)
- Modify a WIM image by adding, deleting, or renaming files
  (**wimlib-imagex update**)
- (Linux only) Mount a WIM image read-only
  (**wimlib-imagex mount**)
- (Linux only) Mount a WIM image read-write
  (**wimlib-imagex mountrw**)
- Delete an image from a WIM file
  (**wimlib-imagex delete**)
- Export image(s) from a WIM file
  (**wimlib-imagex export**)
- Change the name or description of a WIM image
  (**wimlib-imagex info**)
- Change the bootable image index of a WIM file
  (**wimlib-imagex info**)
- Rebuild, and optionally recompress, a WIM file
  (**wimlib-imagex optimize**)
- Split a WIM file into multiple parts
  (**wimlib-imagex split**)
- Join a split WIM
  (**wimlib-imagex join**)
- Verify a WIM file
  (**wimlib-imagex verify**)

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, and most code is shared among all platforms. However, platform-specific features are supported when possible.
- On UNIX-like systems, integration with libntfs-3g allows capturing a WIM image directly from a block device containing an NTFS volume, or applying a WIM image directly to a block device containing an NTFS volume. This allows saving and restoring NTFS-specific data, such as security descriptors and named data streams, which would otherwise only be supported on Windows.
- 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.

- Support for WIM integrity tables. An integrity table is a list of SHA-1 message digests appended to the end of a WIM file which gives checksums over the WIM file itself. The **--check** option to several **wimlib-imagex** commands can be used to verify or add integrity tables.
- On UNIX-like systems, support for saving and restoring UNIX uids (user IDs), gids (group IDs), and modes to/from WIM images. This is a wimlib extension, but the Microsoft implementation ignores this extra metadata.
- Multithreaded compression. By default, data compression is multithreaded and will use all available processors.
- XPRESS, LZX, and LZMS decompression and compression. wimlib contains independent implementations of all these compression algorithms. Sometimes they can do better than the equivalent Microsoft implementations.
- "ESD file" support. As mentioned in **BACKGROUND INFORMATION**, "ESD files" use a new WIM format that features solid resources and LZMS compression. This support was first present in wimlib v1.6.0, but v1.7.0 and later have improved compatibility.
- On Linux, support for mounting WIM images with FUSE (Filesystem in UserSpacE).
- "Pipable" WIMs. This is a wimlib extension and is not compatible with the Microsoft implementation. A pipable WIM, created with **wimcapture** with the **--pipable** option, can be written to standard output or read from standard input. This can be used to pipe images to or from a server over the network to implement fast filesystem imaging and restore.
- 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.
- Fast incremental backups. 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 be read from disk. But even without this option, since the WIM format features single-instance files, a file identical to any already present in the WIM archive (in any image) will not be written, but rather a reference to the stored file will be used.
- 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 filter driver (WOF). 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.

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 ebiggers3@gmail.com. Feedback and suggestions are also welcome.

SEE ALSO

wimlib-imagex-append(1), **wimlib-imagex-apply**(1), **wimlib-imagex-capture**(1), **wimlib-imagex-delete**(1), **wimlib-imagex-dir**(1), **wimlib-imagex-export**(1), **wimlib-imagex-extract**(1), **wimlib-imagex-info**(1), **wimlib-imagex-join**(1), **wimlib-imagex-mount**(1), **wimlib-imagex-mountrw**(1), **wimlib-imagex-optimize**(1), **wimlib-imagex-split**(1), **wimlib-imagex-unmount**(1), **wimlib-imagex-update**(1), **wimlib-imagex-verify**(1),