NAME
>   wimlib-imagex-capture, wimlib-imagex-append – Create or append a WIM image

SYNOPSIS
>   **wimlib-imagex capture** *SOURCE WIMFILE* [*IMAGE_NAME* [*IMAGE_DESCRIPTION*]] [*OPTION*...]
>   **wimlib-imagex append** *SOURCE WIMFILE* [*IMAGE_NAME* [*IMAGE_DESCRIPTION*]] [*OPTION*...]

DESCRIPTION
>   The **wimlib-imagex capture** and **wimlib-imagex append** commands create a Windows Imaging (WIM)
>   image from a directory tree. The **wimlib-imagex capture** command creates a new WIM file containing the
>   captured image, while the **wimlib-imagex append** command appends the captured image to an existing
>   WIM file.
>
>   A WIM image is an independent directory tree in the WIM file. A WIM file may contain any number of
>   separate images. However, files are stored only one time in the entire WIM, regardless of how many
>   images the file appears in.
>
>   *SOURCE* specifies the location of the files to create the new WIM image from. If *SOURCE* is a directory,
>   the WIM image is captured from that directory. Alternatively, if the **--source-list** option is given,
>   *SOURCE* is interpreted as a file that itself provides a list of files and directories to include in the new WIM
>   image. Still alternatively, only on UNIX builds of wimlib, if *SOURCE* is a regular file or block device, it is
>   interpreted as an NTFS volume from which a WIM image is to be captured.
>
>   *IMAGE_NAME* and *IMAGE_DESCRIPTION* specify the name and description of the new image. If
>   *IMAGE_NAME* is not given, it is taken to be the same as the base name of *SOURCE*. If
>   *IMAGE_DESCRIPTION* is not given, no description is given to the new image.

NORMAL MODE (UNIX)
>   This section documents how files are captured from a directory on UNIX. See **WINDOWS VERSION** for
>   the corresponding documentation for the Windows version.
>
>   On UNIX, the "normal" image capture mode is entered when *SOURCE* specifies a directory. The WIM
>   image will be captured from the directory tree rooted at this directory. The directory may be on any type of
>   filesystem.
>
>   In this mode, the following information is captured from the directory tree:
>
>   •   The "normal" name and contents of each file and directory
>
>   •   File and directory creation, access, and modification timestamps to the nearest 100 nanoseconds, if
>   supported by the underlying filesystem
>
>   •   Hard links and symbolic links
>
>   However, in this mode, the following information is *not* captured from the directory tree:
>
>   •   UNIX file owners, groups, and modes. The resulting WIM image will contain no security information
>   (file permissions). (Exception: see the **--unix-data** option.)
>
>   •   Extended attributes.

NTFS MODE (UNIX)
>   This section documents how files are captured from an NTFS volume image on UNIX. See **WINDOWS
>   VERSION** for the corresponding documentation for the Windows version.
>
>   On UNIX, a special image capture mode is entered when *SOURCE* is a regular file or block device.
>   *SOURCE* is interpreted as an NTFS volume and opened using libntfs-3g. If successful, a WIM image is
>   captured containing the contents of the NTFS volume, including NTFS-specific data.
>
>   Please note that the NTFS image capture mode is *not* entered if *SOURCE* is a directory, even if an NTFS
>   filesystem is mounted on *SOURCE*. You must specify the NTFS volume itself (and it must be unmounted,
>   and you must have permission to read from it).
>
>   More specifically, in this mode, the following types of information are captured from the NTFS volume:

- All data streams of all files, including the un-named data stream as well as all named data streams.

- Reparse points, including symbolic links, junction points, and other reparse points.

- File and directory creation, access, and modification timestamps from NTFS inodes (these have a resolution of 100 nanoseconds).

- The security descriptor for each NTFS inode.

- File attribute flags.

- All names of all files, including names in the Win32 namespace, DOS namespace, Win32+DOS namespace, and POSIX namespace. This includes hard links.

WINDOWS VERSION

The Windows versions of **wimlib-imagex capture** and **wimlib-imagex append** act similarly to the corresponding commands of Microsoft's ImageX. For best results, the directory being captured should be on an NTFS volume and you should be running with Administrator privileges; however, non-NTFS filesystems and running without Administrator privileges are also supported.

On Windows, **wimlib-imagex capture** and **wimlib-imagex append** try to archive as much data as possible. This includes:

- All data streams of all files. This includes the default file contents, as well as named data streams if supported by the filesystem and if the version of Windows is Vista or later.

- Reparse points, including symbolic links, junction points, and other reparse points, if supported by the underlying filesystem. (Note: see **--rpfix** and **--norpfix** for documentation on how absolute symbolic links and junctions are captured.)

- File and directory creation, access, and modification timestamps to the nearest 100 nanoseconds.

- Security descriptors, if supported by the filesystem and **--no-acls** is not specified. Furthermore, unless **--strict-acls** is specified, the security descriptor for individual files or directories may be omitted or only partially captured if the user does not have permission to read them.

- File attributes, including hidden, sparse, compressed, encrypted, etc. Encrypted files will be stored in encrypted form rather than in plain text.

- DOS names (8.3) names of files; however, the failure to read them is not considered an error condition.

- Hard links, excluding directory hard links (which aren't supposed to exist anyway).

OPTIONS

--boot

> Specifies that the new image is to be made the bootable image of the WIM archive.

--check

> Include an integrity table in the new WIM file or the modified WIM file. If this option is not specified, no integrity table is included in *WIMFILE*, even if there was one before in the case of **wimlib-imagex append**.

--compress=*TYPE*

> Specifies the compression type for the new WIM file. This flag is only valid for **wimlib-imagex capture**, since the compression mode for **wimlib-imagex append** must be the same as that of the existing WIM. *TYPE* may be "none", "maximum", or "fast". By default, it is "fast".

> You may also specify the actual names of the compression algorithms, "XPRESS" and "LZX", instead of "fast" and "maximum", respectively.

--threads=*NUM_THREADS*

> Number of threads to use for compressing data. Default: autodetect (number of processors). Note: if creating or appending to an uncompressed WIM, additional threads will not be used, regardless of this parameter, since no compression needs to be done in this case.

--rebuild

> For **wimlib-imagex append**: rebuild the entire WIM rather than appending the 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. Also see **wimlib-imagex optimize**.

--flags=*EDITIONID*

> Specify a string to use in the <FLAGS> element of the XML data for the new image.

--verbose

> Print the names of files and directories as they are captured.

--dereference

> (UNIX only) Follow symbolic links and archive the files they point to, rather than archiving the links themselves.

--config=*FILE*

> Specifies a configuration file for capturing the new image. The configuration file specifies files that are to be treated specially during the image capture.

> The format of the configuration file is a number of sections containing path globs one per line, where each section begins with the tag [ExclusionList], [ExclusionException], [CompressionExclusionList], or [AlignmentList]. Currently, only the [ExclusionList] and [ExclusionException] sections are implemented. The [ExclusionList] section specifies a list of path globs to exclude from capture, while the [ExclusionException] section specifies a list of path globs to include in the capture even if the matched file or directory name also appears in the [ExclusionList].

> Relative globs with only one path component (e.g. *.mp3) match against a filename in any directory. Relative globs with multiple path components (e.g. dir/file), as well as absolute globs (e.g. /dir/file), are treated as paths starting at the root directory of capture, or the root of the NTFS volume for NTFS capture mode. If a directory is matched by a glob in the [ExclusionList], the entire directory tree rooted at that directory is excluded from the capture, unless **--dereference** is specified and there is another path into that directory through a symbolic link.

> For compatibility with Windows, the path separators in the globs may be either forward slashes or backslashes, and the line separators may be either UNIX-style or DOS-style. Globs with spaces in them must be quoted, and leading and trailing whitespace is not significant. Empty lines and lines beginning with ´#' or whitespace followed by '#' are ignored.

> Paths may not have drive letters in them, as they are all relative to the root of capture and not absolute external paths.

> If this option is not specified the following default configuration file is used:

>   [ExclusionList]
>   \$ntfs.log
>   \hiberfil.sys
>   \pagefile.sys
>   "\System Volume Information"
>   \RECYCLER
>   \Windows\CSC

--unix-data

> (UNIX only) Store the UNIX owner, group, and mode of all captured files. This is done by adding a special alternate data stream to each directory entry that contains this information. Please note that this flag is for convenience only, in case you want to use **wimlib-imagex** to archive files on UNIX. Microsoft's software will not understand this special information.

--no-acls

> Do not capture files' security descriptors. This option is available in the Windows version, as well as on UNIX in NTFS capture mode.

--strict-acls
>   (Windows only) Fail immediately if the full security descriptor of any file cannot be read. The
>   default behavior without this option is to first try omitting the SACL from the security descriptor,
>   then to try omitting the security descriptor entirely. The purpose of this is to capture as much data
>   as possible without always requiring Administrator privileges. However, if you desire that all secu-
>   rity descriptors be captured exactly, you may wish to provide this option, although the Administra-
>   tor should have permission to read everything anyway.

--rpfix, **--norpfix**
>   Set whether to fix targets of absolute symbolic links (reparse points in Windows terminology) or
>   not. When enabled (**--rpfix**), absolute symbolic links that point inside the directory tree being cap-
>   tured will be adjusted to be absolute relative to the root of the directory tree being captured. In
>   addition, absolute symbolic links that point outside the directory tree being captured will be ignored
>   and not be captured at all. When disabled (**--norpfix**), absolute symbolic links will be captured
>   exactly as is.
>
>   The default behavior for **wimlib-imagex capture** is equivalent to **--rpfix**. The default behavior for
>   **wimlib-imagex append** will be **--rpfix** if reparse point fixups have previously been done on *WIM-*
>   *FILE*, otherwise **--norpfix**.
>
>   In the case of a multi-source capture, (**--source-list** specified), passing **--norpfix** is recommended.
>   Otherwise, reparse point fixups will be disabled on all capture sources destined for non-root loca-
>   tions in the WIM image, while capture sources destined for the WIM root will get the default behav-
>   ior from the previous paragraph.

--source-list
>   **wimlib-imagex capture** and **wimlib-imagex append** support creating a WIM image from multiple
>   files or directories. When **--source-list** is specified, the *SOURCE* argument specifies the name of a
>   text file, each line of which is either 1 or 2 whitespace separated file paths. The first file path, the
>   source, specifies the path to a file or directory to capture into the WIM image. It may be either
>   absolute or relative to the current working directory. The second file path, if provided, is the target
>   and specifies the path in the WIM image that this file or directory will be saved as. Leading and
>   trailing slashes are ignored. "/" indicates that the directory is to become the root of the WIM image.
>   If not specified, the target string defaults to the same as the source string.
>
>   An example source list file is as follows:
>
>
>   # Make the WIM image from the 'winpe' directory
>   winpe   /
>
>
>   # Send the 'overlay' directory to '/overlay' in the WIM image
>   overlay /overlay
>
>
>   # Overlay a separate directory directly on the root of the WIM image.
>   # This is only legal if there are no conflicting files.
>   /data/stuff   /
>
>   Subdirectories in the WIM are created as needed. Multiple source directories may share the same
>   target, which implies an overlay; however, an error is issued if the same file appears in different
>   overlays to the same directory.
>
>   File paths containing whitespace may be quoted with either single quotes or double quotes. Quotes
>   may not be escaped.
>
>   Lines consisting only of whitespace and lines beginning with '#' preceded by optional whitespace
>   are ignored.
>
>   As a special case, if *SOURCE* is "-", the source list is read from standard input rather than an exter-
>   nal file.

The NTFS capture mode cannot be used with **--source-list**, as only capturing a full NTFS volume is supported.

NOTES

wimlib-imagex append does not support appending an image to a split WIM.

The different capture modes only specify the data that is captured and don't specify a special WIM format. A WIM file can contain images captured using different modes. However, all images in a WIM must have the same compression type, and **wimlib-imagex** always enforces this.

wimlib-imagex writes WIMs having the version number 0x10d00 and a compressed stream chunk size of 32768. The only WIMs I've seen that are different from this are some pre-Vista WIMs that had a different version number.

It is safe to abort an **wimlib-imagex append** command partway through; however, after doing this, it is recommended to run **wimlib-imagex optimize** to remove any data that was appended to the physical WIM file but not yet incorporated into the structure of the WIM, unless **--rebuild** was specified, in which case you should delete the temporary file left over.

SOURCE may be a symbolic link to a directory rather than a directory itself. However, additional symbolic links in subdirectories, or in additional source directories not destined for the WIM image root (with **--source-list**), are not dereferenced unless **--dereference** is specified.

With the UNIX version of **wimlib-imagex**, it is possible to create a WIM image containing files with names differing only in case, or files with names containing the characters ':', '*', '?', '"', '<', '>', '|', or '\', which are valid on POSIX-compliant filesystems but not Windows. Be warned that such files will not be extracted by default by the Windows version of **wimlib-imagex**, and Microsoft's ImageX might get confused by such names.

EXAMPLES

Create a new WIM 'mywim.wim' from the directory 'somedir', using LZX compression and including an integrity table:

        wimlib-imagex capture somedir mywim.wim --compress=maximum --check

Append an image to the WIM we just captured, but do it from an NTFS volume on the partition /dev/sda2 and name the image "Windows 7". You do not need to specify the compression type, because the WIM already is using LZX compression and this cannot be changed. You need to specify **--check** if you don't want the integrity table to be discarded.

        wimlib-imagex append /dev/sda2 mywim.wim --check "Windows 7"

SEE ALSO

wimlib-imagex(1)