NAME

wimoptimize − Optimize a WIM archive

SYNOPSIS

wimoptimize *WIMFILE* [*OPTION*...]

DESCRIPTION

wimoptimize, or equivalently **wimlib-imagex optimize**, rebuilds the standalone WIM archive *WIMFILE*. The new WIM is written to a temporary file, and it is renamed to the original file when it's ready. This will remove any holes that have been left in the WIM as a result of appending or deleting files or images, so the new WIM may be smaller than the old WIM.

By default, **wimoptimize** will reuse (not recompress) compressed data and will not change the solid or pipable status of the WIM. However, it can also perform recompression and/or convert between solid, non-solid, pipable, and non-pipable WIMs; see the options and examples below.

OPTIONS

--check
: Before optimizing the WIM, verify its integrity if it contains extra integrity information. Also include extra integrity information in the optimized WIM, even if it was not present before.

--nocheck
: Do not include extra integrity information in the optimized WIM, even if it was present before.

--recompress
: Recompress all data in the WIM while optimizing it. This will significantly increase the time needed to optimize the WIM, but it may result in a better compression ratio if wimlib can do a better job than the program that created the WIM --- which is likely the case if the WIM was Microsoft-created, as wimlib's compressors are slightly stronger.

--compress=*TYPE*[:*LEVEL*]
: Recompress the WIM using the specified compression type, and optionally the specified compression level for that compression type. This implies **--recompress**. See the documentation for this option to **wimcapture**(1) for more details.

--chunk-size=*SIZE*
: Set the WIM compression chunk size to *SIZE*. See the documentation for this option to **wimcapture**(1) for more details.

--solid
: Create a "solid" archive that compresses multiple files together. This usually results in a significantly better compression ratio but has disadvantages such as reduced compatibility. See the documentation for this option to **wimcapture**(1) for more details.

--solid-compress=*TYPE*[:*LEVEL*]
: Like **--compress**, but set the compression type used in solid resources. See the documentation for this option to **wimcapture**(1) for more details.

--solid-chunk-size=*SIZE*
: Like **--chunk-size**, but set the chunk size used in solid resources. See the documentation for this option to **wimcapture**(1) for more details.

--threads=*NUM_THREADS*
: Number of threads to use for compressing data. Default: autodetect (number of processors).

--pipable
: Rebuild the WIM so that it can be applied fully sequentially, including from a pipe. See **wimcapture**(1) for more details about creating pipable WIMs. By default, when neither **--pipable** or **--not-pipable** is specified, the optimized WIM will be pipable if and only if it was pipable before.

--not-pipable
: Rebuild the WIM in the non-pipable format.

--unsafe-compact

Compact the WIM in-place, without using a temporary file. Existing resources are shifted down to fill holes and new resources are appended as needed. The WIM is truncated to its final size, which may shrink the on-disk file. This is more efficient than a full rebuild, but it is only supported when no recompression is being done. More importantly, AN UNSAFE COMPACTION OPERATION CANNOT BE SAFELY INTERRUPTED! If the operation is interrupted, then the WIM will be corrupted, and it may be impossible (or at least very difficult) to recover any data from it. Users of this option are expected to know what they are doing and assume responsibility for any data corruption that may result.

NOTES

wimoptimize does not support split WIMs or delta WIMs. For such files, consider using **wimexport**(1) instead. Note that **wimoptimize** is roughly equivalent to:

wimexport *WIMFILE* all tmp.wim && mv tmp.wim *WIMFILE*

EXAMPLES

Rebuild 'install.wim':

wimoptimize install.wim

Rebuild and recompress 'install.wim':

wimoptimize install.wim --recompress

Rebuild and recompress 'install.wim' using LZX ("maximum") compression at a higher-than-default compression level. The compression chunk size remains unchanged. This command will be slow, but it might be useful for optimizing files for distribution. See *https://wimlib.net/compression.html* for some benchmark results.

wimoptimize install.wim --compress=LZX:100

Recompress 'install.wim' using solid-mode compression, then rename it to ´install.esd´. This will decrease the archive size significantly. (Also consider using 'wimexport install.wim all install.esd --solid'.):

wimoptimize install.wim --solid
mv install.wim install.esd

SEE ALSO

wimlib-imagex(1) **wimexport**(1) **wimverify**(1)