

NAME

wimlib-imagex-optimize – Optimize a WIM archive

SYNOPSIS

wimlib-imagex optimize *WIMFILE* [*OPTION...*]

DESCRIPTION

wimlib-imagex optimize will rebuild the stand-alone WIM *WIMFILE*. The new WIM is written to a temporary file, and it is renamed to the original file when it's ready. This action will remove any holes that have been left as a result of appending images, so the new WIM may be slightly smaller than the old WIM. In addition, some errors in the original WIM may be fixed by re-writing it (although most cannot). This command is also available as simply **wimoptimize** if the appropriate hard link or batch file has been installed.

OPTIONS**--check**

When reading *WIMFILE*, verify its integrity if an integrity table is present. In addition, include an integrity table in the optimized WIM. If this option is not specified, by default the integrity table (if present) is not checked, and an integrity table is included in the rebuilt WIM if and only if one was present in the original WIM.

--nocheck

Neither verify the integrity of *WIMFILE* using the integrity table, nor include an integrity table in the rebuilt WIM file.

--recompress

Recompress all compressed streams in *WIMFILE* when rebuilding it. This will increase the time needed to rebuild the WIM, unless the WIM is uncompressed, but it may result in a better compression ratio if wimlib can do a better job than the program that wrote the original file. A side effect of this is that every stream in the original WIM will be checksummed, so this can help verify that the WIM is intact (equivalent to applying all the images from it).

Note: as mentioned in the README, wimlib generally provides a slightly better XPRESS compression ratio than Microsoft's software, while it generally provides a slightly worse LZX compression ratio than Microsoft's software. So, you may not want to specify **--recompress** when optimizing a LZX-compressed WIM created on Windows with Microsoft's ImageX.

--compress-slow

With **--recompress**: Spend even more time compressing the data in order to achieve a higher compression ratio. Currently, this only has an effect with LZX ("maximum") compression. Depending on the data, compressing with this option will take around 10 to 15 times longer and produce a LZX-compressed WIM about 1% to 5% smaller than one produced with the default LZX compressor. Depending on the data, the resulting WIM may be approximately the same size (typically no more than 0.4% different) as a LZX-compressed WIM produced by WIMGAPI.

--threads=NUM_THREADS

Number of threads to use for compressing data. Default: autodetect (number of processors). This parameter is only meaningful when **--recompress** is also specified.

--pipable

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

--not-pipable

Rebuild the WIM in the non-pipable format. (This is the default if *WIMFILE* is not pipable.)

NOTES

wimlib-imagex optimize does not support split WIMs.

wimlib-imagex optimize is roughly equivalent to:

wimlib-imagex export *WIMFILE* all tmp.wim && mv tmp.wim *WIMFILE*

SEE ALSO

wimlib-imagex(1) **wimlib-imagex-export(1)**