

# minuimus.pl

Minuimus is a file optimiser utility script: You point it at a file, and it makes the file smaller without compromising the file contents. File optimisers are able to do this using a variety of file-specific optimisations, most of which involve decompressing data within a compressed file and recompressing it in a more efficient manner. The process could be compared directly to extracting a ZIP file, then recompressing it at the most demanding setting. It uses many of the same techniques used by other optimisers such as Papa's Best Optimizer.

The latest version is [Minuimus 4.1](#).

If running on Windows, the [bundled windows dependencies](#) will also be required.

Minuimus itself does not carry out all of this optimisation alone. It is dependent upon many other utilities for this, as well as some more specialised methods developed especially for it. Minuimus is a script which automates the process of calling upon all of these utilities, including the process of recursively processing container files and ensuring proper reassembly, detecting and handling the various errors that may occur, and of running some level of integrity checking upon the optimised files to prevent damage.

Minuimus's optimisations are, by default, completely transparent: Not a single pixel of an image will be changed in value, no audio or video will fall in quality. Even metadata is preserved unaltered. It also supports a number of forms of lossy optimisation, which must be explicitly enabled by command line option.

This optimisation isn't actually lossless, strictly speaking, because may delete some essentially-useless information within files such as edit history in PDFs. But it's pretty close. The important information is all preserved, if you use the default settings.

As Minuimus entirely automates the file optimisation process, nothing more is needed than to install the prerequisites and run a single command to point Minuimus at the files to be optimised. It will optimise those files it can, and skip over those it cannot.

Minuimus consists of a perl script and three supporting (optional) binaries which are used for the processing of PDF, WOFF and SWF files. These are written for use on Ubuntu Linux, but should be adaptable to other Linux distributions with little if any alteration. Minuimus will also work on Windows using Strawberry Perl, but has not been subjected to as much testing. These utilities are released under the GPL v3 license, as is Minuimus itself.

Minuimus goes beyond just calling on AdvanceCOMP. When faced with files which are zip containers - such as zip, epub, or docx - it will extract these files, recursively process all of the files contained within them, and put them back together. In this manner it can make e-books and office documents substantially smaller.

Use it to make your website faster, your game easier to distribute, or just to squeeze more pointless holiday photographs on to your computer. It's written on ubuntu linux, but should be adaptable to other distributions. This is a very capable program, as it is able to achieve an ongoing saving in storage and transmission with no cost beside a one-time need for some processor time.

The exact space saving achieved by this level of file optimisation is highly dependent upon the file being optimised. As is expected for any file optimiser, even after extensive testing, the results are too inconsistent to easily quantify. A collection of PDF files sampled from the-eye.eu was reduced in size to 90% of the input, while a half-terabyte sample taken from the archive.org 'computermagazine' collection was reduced with greater success to 78% of the input size. A collection of epub files from Project Gutenberg was reduced to a mere 95%, as these files are light on images, and ZIP files with no files inside which can be recursively optimised are reduced only very slightly, typically to around 97%.

# Supported file types

## Transparent optimisation:

- Images: JPEG TIFF PNG GIF, including animated PNG and GIF
- Documents: DOCX PPTX XLSX ODT ODS ODP EPUB PDF CBZ XPS
- Archives: ZIP 7Z GZ TGZ CAB
- Other: JAR WOFF FLAC SWF STL MP3

## Non-transparent conversion (Disabled unless explicitly specified):

- CBR->CBZ
- RAR->ZIP
- RAR->7z
- 7z->zpaq
- GIF->PNG
- PNG->WebP
- Legacy video format conversion to WebM
- MP3 to Opus, limited to >=256kbps MP3

## Description of processing

- JPEG files initially processed by jpegoptim. For most JPEG files, this is as much as is possible. In some rare cases Minuimus may find a color JPEG which contains only grayscale values, in which case the empty color channels will be removed for a further reduction in size.
- PNG files are processed by optipng, followed by advpng - unless they are animated pngs, as advpng is not animation-safe. Those get advdef instead. Optionally pngout will be called if installed on non-animated PNGs, which sometimes saves another percent or so.
- GIF files are processed by gifsicle. If the file is less than 100KiB and flexigif is installed, it is then processed by flexigif.
- TIFF files are re-compressed on highest setting supported by imagemagick.
- JAR files are processed by advzip, as these files are too delicate to be safely manipulated beyond this. In particular, altering the files within would invalidate any signing.
- PDF files are processed by using qpdf to remove unused objects and object versions, ensure a consistent format and correct any minor errors. Following this pre-processing, all JPEG objects within the PDF are located and processed using jpegoptim. All objects compressed using DEFLATE are also identified and processed using a C helper binary, minuimus\_def\_helper - if this is not installed, DEFLATE processing is skipped. Certain unimportant internet metadata objects are deleted, but the main document metadata is not touched unless --discard-meta is specified. The processed PDF is then relinearised using qpdf. As a fail-safe against anything going wrong in this process, both original and compressed PDFs are rendered into bitmap format and compared - the original will only be overwritten if the optimised file rendered identically. This combination of qpdf, jpeg and DEFLATE processing makes Minuimus one of the most effective lossless PDF optimisation utilities available. Finally, if pdfsizeopt is installed, it will be called - as it includes some optimisations that minuimus lacks, and vice versa, the two tools together can outperform either alone.
- ZIP, DOCX, XLSX, ODT, ODS, ODP, EPUB and CBZ, the zip-derived archive formats, are processed by extracting to a temporary folder, processing all (non-archive) files within as independent files, then re-compressing into a ZIP file and compressing that with advzip. The 'mimetype' file is correctly placed as the first in the completed file using store-only compression in accordance with the EPUB OCF. Additionally a number of 'junk' files such as Thumb.db and .ds\_store are deleted from ZIP files.
- For CBZ files, any GIF, BMP, PCX or TIFF files within are converted to PNG (In an animation-safe manner) in addition to the standard ZIP processing. The capability to convert PNGs into WebP is present, but default disabled due to limited viewer support for WebP.
- GZ and TGZ files just get processed by advdef.
- 7Z archives are extracted and constituent files processed, then recompression attempted using both LZMA and PPMd compression algorithms on highest practical settings. Whichever of these produces

the smallest file is kept, unless the original file is smaller. Solid compression is not used.

- WOFF files (web fonts) are processed using a Zopfli-based recompressor.
- FLAC files are re-encoded using the highest possible profile-compliant settings - slightly better than the regular -9. Metadata is preserved. FLAC files are also examined to determine if they contain a mono audio track encoded as stereo. Such files are converted to true mono, which achieves a substantial space saving.
- CAB - the MS CAB, not Installshield CAB - will be repackaged if possible, but the savings from this are very small. Signed cabs are ignored.
- HTML, CSS and SVG files are also searched for any base64-encoded JPG, PNG or WOFF resources, and these resources optimised appropriately.
- SWF files will have their internal JPEG and PNG objects recompressed, and the outer DEFLATE wrapper run through Zopfli. However, as most SWF generation software is already focused on producing small files, savings are generally small.
- STL models in ASCII form will be converted to binary form. This makes them a lot smaller - but most STLs are already binary now. Optionally, explicit normals can be removed - this makes the files more compressible, though no longer strictly standard-compliant. I've yet to find a program that is affected by removing the vertex normals - they are really just a relic of the format's early origins.
- MP3 files will be repacked if that makes them smaller. Usually it doesn't - but some very old MP3s are poorly packed and might get a tiny bit smaller.

If the utility [Leanify](#) is installed, it will also be used to augment Minuimus's own processing where possible. As both programs can support a few tricks that the other cannot, they work better in conjunction than either alone on certain files. Most significantly, JPEG and formats containing JPEG.

Some further capabilities are disabled by default because they are not fully transparent, but can be enabled by command-line option:

- Conversion of PNG within CBZ files into WebP. The resulting files are more compact, but not supported by all CBZ-viewing software. All conversions animation-safe.
- Conversion of all GIFs to PNG, or of PNGs to WebP, or of legacy image formats (bmp, pcx, tiff) to PNG. These do chain: GIF-to-PNG plus PNG-to-WebP results in GIF-to-WebP.
- Conversion of CBR to CBZ and RAR to ZIP. This doesn't compact files directly, but does enable the use of the optimisations appropriate for the converted file type.
- Conversion of RAR or ZIP files into 7Z, if doing so would result in smaller overall size (Which it usually does), using the same optimisations used for 7Z files.
- Conversion of 7z to ZPAQ likewise. ZPAQ is perhaps the highest-ratio compression container currently available for general usage.

## Lossy multimedia operations

Additionally, minuimus has the option (disabled by default) of converting certain legacy multimedia files into modern codecs - reducing their size greatly in the process. This is, however, an inherently lossy operation. The compression will be abandoned if it results in an output file of larger size than the input.

- If video processing is enabled, video files using dated codecs such as DivX, XviD or MPEG1/2 will be converted to WebM using AV1 and Opus. The files will be compared in duration after encoding as a precaution against truncation due to damaged input files.
- If audio processing is enabled, MP3 files with a bitrate of 256kbps or above will be converted to 128kbps Opus. Both these bitrates are chosen as they are generally considered transparent for their respective codecs. The bitrate can be reduced with --audio-agg if processing speech only.
- Both of these also detect mono audio stored as stereo, and convert it to mono format for more efficient compression.

Minuimus.pl and associated helpers are released under the GPL licence, version three.