

CTW version 0.1 Manual

CTW version 0.1 is a prompt-oriented program to encode (compress) and decode (decompress) single files (especially text files), with the possibility to specify many different options. On this page you can find instructions on how to use it, starting with the [basic usage](#). The available [command line options](#) can be found after the basic usage. The meaning of the [statistics](#) that CTW generates, can be found at the bottom of this manual.

Basic usage

- To get a short explanation of the usage of the program: `ctw`
- To encode a file (e.g. `file.txt`) using the default options: `ctw e file.txt`
The default name for the output file is `file.txt.ctw`
It is also possible to specify a different output filename (e.g. `outfile.ctw`):
`ctw e file.txt outfile.ctw`
The program will output the settings and the encoding will start. When the encoding is finished, some CTW statistics will be shown.
- To decode a file (e.g. `file.txt.ctw`): `ctw d file.txt.ctw`
The default name for the output file is `file.txt.d`
It is also possible to specify a different output filename:
`ctw d file.txt.ctw outfile.txt`
- To show information about a CTW file (e.g. `file.ctw`): `ctw i file.ctw`
The settings that were used for encoding the CTW file are shown.

Command line options

There are different kind of command line options. First, there are options to specify parameters for the encoder/decoder. These options are only relevant for encoding, because a file can only be decoded correctly with exactly the same settings. In the decoding case, the settings that were used for encoding are used (which are stored in the header of the CTW file). The default values for all options are such, that usually a good performance is achieved.

<i>Option</i>	<i>Def.</i>	<i>Min.</i>	<i>Max.</i>	<i>Description</i>
<code>-dX</code>	6	1	12	Set the maximum depth of the context tree to X. In most cases, a higher value results in a slightly better compression rate, but a slower performance. With a larger tree depth, a larger tree array will be required (option <code>-nX</code>), because more nodes will be created.
<code>-tX</code>	32	1	32	Set the maximum number of tries for finding a node in the tree array. A higher value might result in a better compression, but a slower performance: there will be more tries before the search operation for a tree node will fail.

<i>Option</i>	<i>Def.</i>	<i>Min.</i>	<i>Max.</i>	<i>Description</i>
-nX	4M	1K	16M	Set the maximum number of nodes in the tree array, in which the tree nodes are stored. Values can be specified using 'M' for millions of nodes and 'K' for thousands of nodes (eg. -n4M means 4194304 nodes). The tree array requires 8 times X bytes of memory, so with the standard settings, 32 megabytes of memory will be allocated for the tree array. This setting is important for the compression performance of the encoder/decoder: a lot of nodes can't be allocated in a tree array that's too small, which results in a worse compression rate. The value must be a power of 2.
-fX	4M	512	16M	Set the maximum file buffer size; values can be specified using 'M' for megabytes and 'K' for kilobytes. If the maximum file buffer size is larger than the actual filesize, the actual file buffer size is equal to the actual filesize. If the (uncompressed) file does not fit in the file buffer entirely, the compression rate might be worse. The value must be a power of 2.
-bX	1K	1	16K	Set the maximum value of log beta. The internal CTW parameter log beta will be limited to this value. A smaller value might result in a better compression rate for non-stationary files, but the compression rate might be worse for other files. The value must be a power of 2.
-s	enabled			Disable strict unique path pruning. With strict pruning enabled, there are about 15% less nodes required because unique paths in the tree are more optimally pruned. This does not (significantly) affect the compression ratio.
-r	disabled			Enable weighting at the root nodes. If this is enabled, the weighting of the probabilities is also performed on the root nodes in the CTW tree. The compression ratio might be better or worse with root weighting.
-k	ZR			Use Krichevsky-Trofimov (KT) estimator instead of Zero-Redundancy (ZR) estimator. In which way the probabilities in the tree nodes are estimated, depends on the estimator that is used. In most cases, the Zero-Redundancy estimator performs better.

There are also some other options that do not affect the actual encoding/decoding process. These are:

<i>Option</i>	<i>Description</i>
-y	Force overwriting of existing files. If this switch is specified, the program will not ask the user if a file should be replaced or not and will overwrite existing files.
-lX	Enables logging to text file X. If this option is specified, the program will write some information to the file X: the filename and filesize of the file that was encoded/decoded, the used CTW settings, and the CTW statistics. If file X already exists, the information will be appended to the end of the file.

CTW statistics

After encoding or decoding, the CTW program shows some relevant statistics, which will be explained below:

<i>Statistic</i>	<i>Description</i>
#codebits	The number of bits that is used for the encoded file. Note: the filesize of the encoded file is not exactly equal to the number of codebits, because the file also contains a header.
#treenodes	The total number of tree nodes that is created in the tree array.
#failed	The number of times a tree node was not found in the tree array.
processing time	The total time it took to encode/decode the file (in seconds).
compression-rate	The average number of bits that was required to encode one byte of the uncompressed file (e.g. 8 bits/byte means no compression at all). This is the number of codebits (#codebits) divided by the original filesize.
decompression-rate	The average number of bytes that was decoded from one bit of the compressed file.