

## -m (Set compression Method) switch

Specifies the compression method.

### Syntax

```
-m<method_parameters>
```

The format for this switch depends on the archive type.

- [Zip](#)
- [GZip](#)
- [BZip2](#)
- [7z](#)
- [XZ](#)

## Zip

| Parameter                                   | Default   | Description   |
|---|-----------|---|
| <a href="#">x</a> =[0   1   3   5   7   9 ] | 5         | Sets level of compression.  |
| <a href="#">m</a> ={MethodID}               | Deflate   | Sets a method: Copy, Deflate, Deflate64, BZip2, LZMA, PPMd.                           |
| <a href="#">fb</a> ={NumFastBytes}          | 32        | Sets number of Fast Bytes for Deflate encoder.  |
| <a href="#">pass</a> ={NumPasses}           | 1         | Sets number of Passes for Deflate encoder.  |
| <a href="#">d</a> ={Size}[b k m]            | 900000    | Sets Dictionary size for BZip2  |
| <a href="#">mem</a> ={Size}[b k m]          | 24        | Sets size of used memory for PPMd.  |
| <a href="#">o</a> ={Size}                   | 8         | Sets model order for PPMd.  |
| <a href="#">mt</a> =[off   on   {N}]        | on        | Sets multithreading mode.   |
| <a href="#">em</a> ={EncryptionMethodID}    | ZipCrypto | Sets a encryption method: ZipCrypto, AES128, AES192, AES256                           |
| <a href="#">tc</a> =[off   on]              | on        | Stores NTFS timestamps for files: Modification time, Creation time, Last access time. |
| <a href="#">cl</a> =[off   on]              | off       | 7-Zip always uses local code page for file names.                                     |
| <a href="#">cu</a> =[off   on]              | off       | 7-Zip uses UTF-8 for file names that contain non-ASCII symbols.                       |

By default (if **cl** and **cu** switches are not specified), 7-Zip uses UTF-8 encoding only for file names that contain symbols unsupported by local code page.

### **x**=[0 | 1 | 3 | 5 | 7 | 9 ]

Sets level of compression. x=0 means Copy mode (no compression).

Deflate / Deflate64 settings:

| Level | NumFastBytes | NumPasses | Description |
|-------|--------------|-----------|-------------|
| 1     | 32           | 1         | Fastest     |
| 3     |              |           | Fast        |
| 5     |              |           | Normal      |
| 7     | 64           | 3         | Maximum     |

|   |     |    |       |
|---|-----|----|-------|
| 9 | 128 | 10 | Ultra |
|---|-----|----|-------|

x=1 and x=3 with Deflate method set fast mode for compression.

BZip2 settings:

| Level | Dictionary | NumPasses | Description |
|-------|------------|-----------|-------------|
| 1     | 100000     | 1         | Fastest     |
| 3     | 500000     |           | Fast        |
| 5     | 900000     |           | Normal      |
| 7     |            | 2         | Maximum     |
| 9     |            | 7         | Ultra       |

#### **fb={NumFastBytes}**

Sets the number of fast bytes for the Deflate/Deflate64 encoder. It can be in the range from 3 to 258 (257 for Deflate64). Usually, a big number gives a little bit better compression ratio and a slower compression process. A large fast bytes parameter can significantly increase the compression ratio for files which contain long identical sequences of bytes.

#### **pass={NumPasses}**

Sets number of passes for Deflate encoder. It can be in the range from 1 to 15 for Deflate and from 1 to 10 for BZip2. Usually, a big number gives a little bit better compression ratio and a slower compression process.

#### **d={Size}[b|k|m]**

Sets the Dictionary size for BZip2. You must specify the size in bytes, kilobytes, or megabytes. The maximum value for the Dictionary size is 900000b. If you do not specify any symbol from set [b|k|m], dictionary size will be calculated as DictionarySize = 2<sup>Size</sup> bytes.

#### **mem={Size}[b|k|m]**

Sets the size of memory used for PPMd. You must specify the size in bytes, kilobytes, or megabytes. The maximum value is 256 MB = 2<sup>28</sup> bytes. The default value is 24 (16MB). If you do not specify any symbol from the set [b|k|m], the memory size will be calculated as (2<sup>Size</sup>) bytes. PPMd uses the same amount of memory for compression and decompression.

#### **o={Size}**

Sets the model order for PPMd. The size must be in the range [2,16]. The default value is 8.

#### **mt=[off | on | {N}]**

Sets multithread mode. If you have a multiprocessor or multicore system, you can get a speed increase with this switch. This option affects only compression (with any method) and decompression of BZip2 streams. Each thread in the multithread mode uses 32 MB of RAM for buffering. If you specify {N}, 7-Zip tries to use N threads.

## **GZip**

GZip uses the same parameters as Zip, but GZip compresses only with Deflate method. So GZip supports only the following parameters: x, fb, pass.

## **BZip2**

| Parameter             | Default | Description                              |
|-----------------------|---------|--|
| x=[1   3   5   7   9] | 5       | Sets level of compression.               |
| pass={NumPasses}      | 1       | Sets number of Passes for Bzip2 encoder. |
| d={Size}[b k m]       | 900000  | Sets Dictionary size for BZip2           |
| mt=[off   on   {N}]   | on      | Sets multithreading mode.                |

**x=[1 | 3 | 5 | 7 | 9 ]**

Sets level of compression

| Level | Dictionary | NumPasses | Description |
|-------|------------|-----------|-------------|
| 1     | 100000     | 1         | Fastest     |
| 3     | 500000     |           | Fast        |
| 5     | 900000     |           | Normal      |
| 7     |            | 2         | Maximum     |
| 9     |            | 7         | Ultra       |

**d={Size}[b|k|m]**

Sets the Dictionary size for BZip2. You must specify the size in bytes, kilobytes, or megabytes. The maximum value for the Dictionary size is 900000b. If you do not specify any symbol from set [b|k|m], dictionary size will be calculated as DictionarySize = 2<sup>Size</sup> bytes.

**pass={NumPasses}**

Sets the number of passes. It can be in the range from 1 to 10. The default value is 1 for normal mode, 2 for maximum mode and 7 for ultra mode. A bigger number can give a little bit better compression ratio and a slower compression process.

**mt=[off | on | {N}]**

Sets multithread mode. If you have a multiprocessor or multicore system, you can get a speed increase with this switch. If you specify {N}, for example mt=4, 7-Zip tries to use 4 threads.

## 7z

| Parameter  | Default | Description   |
|--|---------|---|
| <a href="#">x=[0   1   3   5   7   9 ]</a>                           | 5       | Sets level of compression.  |
| <a href="#">s=[off   on   [e] [{N}f] [{N}b   {N}k   {N}m   {N}g]</a> | on      | Sets solid mode.  |
| <a href="#">f=[off   on]</a>   | on      | Enables or disables compression filters for executable files.             |
| <a href="#">hc=[off   on]</a>  | on      | Enables or disables archive header compressing.                           |
| <a href="#">he=[off   on]</a>  | off     | Enables or disables archive header encryption.                            |
| <a href="#">b{C1}[s{S1}]:{C2}[s{S2}]</a>                             |         | Sets binding between coders.  |
| <a href="#">{N}={MethodID}[:param1] [:param2][..]</a>                | LZMA    | Sets a method: LZMA, LZMA2, PPMd, BZip2, Deflate, Delta, BCJ, BCJ2, Copy. |
| <a href="#">mt=[off   on   {N}]</a>                                  | on      | Sets multithreading mode.   |
| <a href="#">tc=[off   on]</a>  | off     | Stores file creation timestamps.  |

**x=[0 | 1 | 3 | 5 | 7 | 9 ]**

Sets level of compression

| Level | Method | Dictionary | FastBytes | MatchFinder | Filter | Description         |
|-------|--------|------------|-----------|-------------|--------|---------------------|
| 0     | Copy   |            |           |             |        | No compression.     |
| 1     | LZMA   | 64 KB      | 32        | HC4         | BCJ    | Fastest compressing |
| 3     | LZMA   | 1 MB       | 32        | HC4         | BCJ    | Fast compressing    |
| 5     | LZMA   | 16 MB      | 32        | BT4         | BCJ    | Normal compressing  |
| 7     | LZMA   | 32 MB      | 64        | BT4         | BCJ    | Maximum             |

|   |      |       |    |     |      |                   |
|---|------|-------|----|-----|------|-------------------|
|   |      |       |    |     |      | compressing       |
| 9 | LZMA | 64 MB | 64 | BT4 | BCJ2 | Ultra compressing |

Note: "x" works as "x=9".

**s=[off | on | [e] [{N}f] [{N}b | {N}k | {N}m | {N}g]]**

Enables or disables solid mode. The default mode is s=on. In solid mode, files are grouped together. Usually, compressing in solid mode improves the compression ratio.

|                           |  |
|---------------------------|--|
| e                         | Use a separate solid block for each new file extension   |
| {N}f                      | Set the limit for number of files in one solid block     |
| {N}b   {N}k   {N}m   {N}g | Set a limit for the total size of a solid block in bytes |

These are the default limits for the solid block size:

| Compression Level | Solid block size |
|-------------------|------------------|
| Store             | 0 B              |
| Fastest           | 16 MB            |
| Fast              | 128 MB           |
| Normal            | 2 GB             |
| Maximum           | 4 GB             |
| Ultra             | 4 GB             |

Limitation of the solid block size usually decreases compression ratio but gives the following advantages:

- Decreases losses in case of future archive damage.
- Decreases extraction time of a group of files (or just one file), so long as the group doesn't contain the entire archive.

The updating of solid .7z archives can be slow, since it can require some recompression.

Example:

**s=100f10m**

set solid mode with 100 files & 10 MB limits per one solid block.

**f=[off | on]**

Enables or disables compression filters for executable files: dll, exe, ocx, sfx, sys. It uses BCJ2 filter in Ultra mode and BCJ filter in other modes. The default mode is f=on.

**hc=[off | on]**

Enables or disables archive header compressing. The default mode is hc=on. If archive header compressing is enabled, some parts of archive header will be compressed with LZMA method.

**he=[off | on]**

Enables or disables archive header encryption. The default mode is he=off.

**{N}**

Sets order of methods. It is used also to associate parameters with methods. Numbers must begin from 0. Methods that have smaller numbers will be used before others.

**b{C1}[s{S1}]:{C2}[s{S2}]**

Binds output stream S1 in coder C1 with input stream S2 in coder C2. If stream number is not specified, stream with number 0 will be used.

Usually coder has one input stream and one output stream. In 7z some coders can have multiple

input and output streams.

For example, [BCJ2](#) encoder has one input stream and four output streams.

### **mt=[off | on | {N}]**

Sets multithread mode. If you have a multiprocessor or multicore system, you can get a increase with this switch. 7-Zip supports multithread mode only for LZMA / LZMA2 compression and BZip2 compression / decompression. If you specify {N}, for example mt=4, 7-Zip tries to use 4 threads. LZMA compression uses only 2 threads.

### **{N}={MethodID}[:param1][:param2] ... [:paramN]**

Sets compression method. You can use any number of methods. The default method is LZMA.

Parameters must be in one of the following forms:

- {ParamName}={ParamValue}.
- {ParamName}{ParamValue}, if {ParamValue} is number and {ParamName} doesn't contain numbers.

Supported methods:

| MethodID              | Description                               |
|-----------------------|---|
| <a href="#">LZMA</a>  | LZ-based algorithm                        |
| <a href="#">LZMA2</a> | LZMA-based algorithm                      |
| <a href="#">PPMd</a>  | Dmitry Shkarin's PPMdH with small changes |
| BZip2                 | BWT algorithm                             |
| Deflate               | LZ+Huffman                                |
| Copy                  | No compression                            |

Supported filters:

| MethodID              | Description   |
|-----------------------|---|
| <a href="#">Delta</a> | Delta filter  |
| BCJ                   | converter for x86 executables                       |
| <a href="#">BCJ2</a>  | converter for x86 executables (version 2)           |
| ARM                   | converter for ARM (little endian) executables       |
| ARMT                  | converter for ARM Thumb (little endian) executables |
| IA64                  | converter for IA-64 executables                     |
| PPC                   | converter for PowerPC (big endian) executables      |
| SPARC                 | converter for SPARC executables                     |

Filters increase the compression ratio for some types of files. Filters must be used with one of the compression method (for example, BCJ + LZMA).

## **LZMA**

LZMA is an algorithm based on Lempel-Ziv algorithm. It provides very fast decompression (about 10-20 times faster than compression). Memory requirements for compression and decompression also are different (see [d={Size}\[b|k|m\]](#) switch for details).

| Parameter                       | Default | Description               |
|---------------------------------|---------|---------------------------|
| <a href="#">a=[0 1]</a>         | 1       | Sets compressing mode     |
| <a href="#">d={Size}[b k m]</a> | 24      | Sets Dictionary size      |
| <a href="#">mf={MF_ID}</a>      | bt4     | Sets Match Finder         |
| <a href="#">fb={N}</a>          | 32      | Sets number of Fast Bytes |

|               |    |  |
|---------------|----|--|
| <b>mc={N}</b> | 32 | Sets Number of Cycles for Match Finder       |
| <b>lc={N}</b> | 3  | Sets number of Literal Context bits - [0, 8] |
| <b>lp={N}</b> | 0  | Sets number of Literal Pos bits - [0, 4]     |
| <b>pb={N}</b> | 2  | Set number of Pos Bits - [0, 4]              |

**a={0|1}**

Sets compression mode: 0 = fast, 1 = normal. Default value is 1.

**d={Size}[b|k|m]**

Sets Dictionary size for LZMA. You must specify the size in bytes, kilobytes, or megabytes. The maximum value for dictionary size is 1 GB =  $2^{30}$  bytes. Default values for LZMA are 24 (16 MB) in normal mode, 25 (32 MB) in maximum mode (-mx=7) and 26 (64 MB) in ultra mode (-mx=9). If you do not specify any symbol from the set [b|k|m], the dictionary size will be calculated as DictionarySize =  $2^{\text{Size}}$  bytes. For decompressing a file compressed by LZMA method with dictionary size N, you need about N bytes of memory (RAM) available.

**mf={MF\_ID}**

Sets Match Finder for LZMA. Default method is bt4. Algorithms from hc\* group don't provide a good compression ratio, but they often work pretty fast in combination with fast mode (a=0). Memory requirements depend on dictionary size (parameter "d" in table below).

| MF_ID | Dictionary        | Memory Usage |        | Description |                 |
|-------|-------------------|--------------|--------|-------------|-----------------|
| bt2   |                   | 9.5 * d      | + 4 MB | Binary Tree | 2 bytes hashing |
| bt3   |                   | 11.5 * d     |        |             | 3 bytes hashing |
| bt4   | 64 KB ... 48 MB   | 11.5 * d     |        |             | 4 bytes hashing |
|       | 64 MB ... 1024 MB | 10.5 * d     |        |             |                 |
| hc4   | 64 KB ... 48 MB   | 7.5 * d      |        | Hash Chain  |                 |
|       | 64 MB ... 1024 MB | 6.5 * d      |        |             |                 |

Note: Your operation system also needs some amount of physical memory for internal purposes. So keep at least 32MB of physical memory unused.

**fb={N}**

Sets number of fast bytes for LZMA. It can be in the range from 5 to 273. The default value is 32 for normal mode and 64 for maximum and ultra modes. Usually, a big number gives a little bit better compression ratio and slower compression process.

**mc={N}**

Sets number of cycles (passes) for match finder. It can be in range from 0 to 1000000000. Default value is  $(16 + \text{number\_of\_fast\_bytes} / 2)$  for BT\* match finders and  $(8 + \text{number\_of\_fast\_bytes} / 4)$  for HC4 match finder. If you specify mc=0, LZMA will use default value. Usually, a big number gives a little bit better compression ratio and slower compression process. For example, mf=HC4 and mc=10000 can provide almost the same compression ratio as mf=BT4.

**lc={N}**

Sets the number of literal context bits (high bits of previous literal). It can be in range from 0 to 8. Default value is 3. Sometimes lc=4 gives gain for big files.

**lp={N}**

Sets the number of literal pos bits (low bits of current position for literals). It can be in the range from 0 to 4. The default value is 0. The lp switch is intended for periodical data when the period is equal to  $2^{\text{value}}$  (where lp=value). For example, for 32-bit (4 bytes) periodical data you can use lp=2. Often it's better to set lc=0, if you change lp switch.

**pb={N}**

Sets the number of pos bits (low bits of current position). It can be in the range from 0 to 4. The default value is 2. The pb switch is intended for periodical data when the period is equal  $2^{\text{value}}$  (where  $\text{lp}=\text{value}$ ).

## LZMA2

LZMA2 is modified version of LZMA. it provides the following advantages over LZMA:

- Better compression ratio for data than can't be compressed. LZMA2 can store such blocks of data in uncompressed form. Also it decompresses such data faster.
- Better multithreading support. If you compress big file, LZMA2 can split that file to chunks and compress these chunks in multiple threads.

| Parameter                    | Default      | Description     |
|------------------------------|--------------|-----------------|
| <code>c={Size}[b k m]</code> | dictSize * 4 | Sets Chunk size |

If you don't specify ChunkSize, LZMA2 sets it to DictionarySize \* 4.

LZMA2 also supports all LZMA parameters, but  $\text{lp}+\text{lc}$  cannot be larger than 4.

LZMA2 uses: 1 thread for each chunk in x1 and x3 modes; and 2 threads for each chunk in x5, x7 and x9 modes. If LZMA2 is set to use only such number of threads required for one chunk, it doesn't split stream to chunks. So you can get different compression ratio for different number of threads. You can get the best compression ratio, when you use 1 or 2 threads.

## PPMd

PPMd is a PPM-based algorithm. This algorithm is mostly based on Dmitry Shkarin's PPMdH source code. PPMd provides very good compression ratio for plain text files. There is no difference between compression speed and decompression speed. Memory requirements for compression and decompression also are the same.

| Parameter                      | Default | Description                        |
|--------------------------------|---------|------------------------------------|
| <code>mem={Size}[b k m]</code> | 24      | Sets size of used memory for PPMd. |
| <code>o={Size}</code>          | 6       | Sets model order for PPMd.         |

### **mem={Size}[b|k|m]**

Sets the size of memory used for PPMd. You must specify the size in bytes, kilobytes, or megabytes. The maximum value is  $2\text{GB} = 2^{31}$  bytes. The default value is 24 (16MB). If you do not specify any symbol from the set `[b|k|m]`, the memory size will be calculated as  $(2^{\text{Size}})$  bytes. PPMd uses the same amount of memory for compression and decompression.

### **o={Size}**

Sets the model order for PPMd. The size must be in the range `[2,32]`. The default value is 6.

## BCJ2

BCJ2 is a Branch converter for 32-bit x86 executables (version 2). It converts some branch instructions for increasing further compression.

A BCJ2 encoder has one input stream and four output streams:

- s0: main stream. It requires further compression.
- s1: stream for converted CALL values. It requires further compression.
- s2: stream for converted JUMP values. It requires further compression.
- s3: service stream. It is already compressed.

If LZMA is used, the size of the dictionary for streams s1 and s2 can be much smaller (512 KB is enough for most cases) than the dictionary size for stream s0.

## Delta

It's possible to set delta offset in bytes. For example, to compress 16-bit stereo WAV files, you

can set "0=Delta:4". Default delta offset is 1.

## XZ

XZ supports only LZMA2 codec now. The switches are similar to switches for 7z format.

### Examples

```
7z a -tzip archive.zip *.jpg -mx0
```

adds \*.jpg files to archive.zip archive without compression.

```
7z a -t7z archive.7z *.exe *.dll -m0=BCJ -m1=LZMA:d=2l -ms -mmt
```

adds \*.exe and \*.dll files to solid archive archive.7z using LZMA method with 2 MB dictionary and BCJ converter. Compression will use multithreading optimization.

```
7z a -t7z archive.7z *.exe *.dll -m0=BCJ2 -m1=LZMA:d23 -m2=LZMA:d19 -m3=LZMA:d19  
-mb0:1 -mb0s1:2 -mb0s2:3
```

adds \*.exe and \*.dll files to archive archive.7z using BCJ2 converter, LZMA with 8 MB dictionary for main output stream (s0), and LZMA with 512 KB dictionary for s1 and s2 output streams of BCJ2.

```
7z a -t7z archive.7z *.txt -m0=PPMd
```

adds \*.txt files to archive archive.7z using PPMd method.

### Commands that can be used with this switch

[a \(Add\)](#), [d \(Delete\)](#), [u \(Update\)](#),

### See also

**Switches:** [-t \(set Type of archive\)](#),