Command-line Guide for Linux, Mac & Windows

File Archiving, File Management, Compression, Decompression, Extraction, Tar, Zip,

Contents

1. 7-Zip Versions
2. Command Line Syntax
    1. Syntax Conventions
3. Time Dependent Considerations
4. Command Detailed Reference
5. Switch Detailed Reference
6. Wildcard Parsers
7. List Files
8. Archive Formats and Parameters:
    1. 7z Archive Format Parameters
    2. ZIP Archive Format Parameters:
9. Switches
    1. Type of Archive Switch
    2. Password Switch
    3. Compression Method Switch
    4. Create Self-Extracting Archives Switch
    5. Update Options Switch
10. Exit Codes from 7-Zip
11. Archive Format Comparison
12. Archive Types in Detail
    1. 7z Archives
    2. BZIP2 Archives
    3. GZIP Archives
    4. ZIP Archives
13. Compression Methods in Detail
    1. BZIP2 Compression Method
    2. Deflate and Deflate64 Compression Methods
    3. GZip
    4. LZMA Compression Method
    5. PPMd Compression Method
    6. Compression Method Filters

See also: File Archiving and Compression, Accessing and Sharing Files, Network Access, Windows Terminal Servers

7-Zip Versions

7-Zip is an Archive and File Management utility available in command-line versions for Linux/Mac, ***"P7Zip"*** (7z.exe), as well as for Windows, ***"7za"*** (7za.exe). Although its interface is deceptively simple, the command-line versions of 7ZIP are highly customizable archiving programs when used with the command parameters and switchesdescribed below. Windows users who want to use the command-line version should generate a Help Desk ticket to install the standalone 7za.exe version.

To begin a session, open a terminal window. Invoke the version of 7Zip you are using by entering "**7z**" for P7Zip (7z.exe), or "**7za**" for 7Zip for Windows (7za.exe) to start either the P7-Zip or 7za application prior to entering commands. Other than this program invocation command, all commands, parameters and switches are identical for all command-line versions.

NOTE TO WINDOWS USERS: the following syntax examples begin by invoking the Linux command-line version, "7z". Please change the invocation to "**7za**" when applying these examples for use in 7-Zip for Windows.

Command Line Syntax

The general command line syntax begins by invoking the version of 7Zip you are using:

"**7z**" for P7Zip (*7z.exe*) users

or

"**7za**" for 7Zip for Windows (*7za.exe*) users

followed by the ***command*** and ***parameters***:

"command" "switches" "full_path_archive_name" "full_path_file_name"

Eg; ***7z* a -p *7Zip_Archive Test_file.txt*** creates a **7z** formatted archive named ***7Zip_Archive*** that is protected with a **p**assword , then **a**dds a file named ***test_file.txt*** to the archive.

Parameter and Switch Syntax: Use a separate **-m** switch for each parameter when adding them to the command line. For example: **7z a -t7z Encrypt.7z Test8.txt -mx=7 -mhe=on uses the**

File Names:

The command-line versions of 7Zip look to the directory where the 7Zip executable file is located to find the files you are managing. Therefore, you will need to begin a session either by moving your files into 7Zip's directory or by using the **full path** for the file names.

Eg; **7z a *"c:\Documents and Settings\JDoe\Desktop\archive_name.zip"* "c:\Documents and Settings\JDoe\Desktop*file_name.txt*"** creates a ZIP formatted archive and adds the specified text file to it. The full path must be enclosed in quotes as in the example above if any portion of it contains a space. Additionally, Windows users should use the *Set Sensitive Case mode switch* to "*insensitive*" (**-ssc-**).

Syntax Conventions

(archive format) - must be one of the supported archive formats. 7ZIP's native format, 7z, is the default. See *Creating an Archive* for detailed information on archive types. The archive type is denoted by the file type extension (eg., ".7z", ".zip", ".tar") you specify. If the archive doesn't exist, 7Zip will create it when you add the initial file(s).

<arguments> - The first argument will always be the command, followed by switches and filenames with their associated expressions - eg; ***"7z d archive.zip *.bak -r"***

[expressions] - optional, but only one expression can be specified - eg; **hc=[off | on]** is specified as ***"hc=on"*** to enable header compression in the command line; use one switch per expression in most cases, the -m switch.

{expression} - if used, replace with a user-defined string - eg; **{password}** is replaced by ***"myGreat!paSSphr4se"*** in the command line; must be combined with a switch or command.

The "***base_archive_name***" must be the first filename on the command line after the archive format and command.

Switches and other filenames can be in any order.

Wildcards or filenames with spaces must be quoted:

- "Dir\Program files*"
- Dir\"Program files"*

Switch options can be combined to save command line length. However, some switch options take optional string arguments, and therefore must be the last option in a combined argument token string because 7-Zip accepts the rest of the argument token as the optional argument.

Time Dependent Considerations

Transferring files to recipients in other time zones, Daylight Savings Time adjustments and relocating notebook computers to different time zones can cause problems with update commands that depend on the file's modification time. Therefore, employ only a file system and archive format that uses Coordinated Universal Time (UTC) if possible. For example, select the NTFS file system and choose one of the archive formats that support UTC like ***7z*** or ***ZIP*** **combined with the *-mtc switch***. However, in cases where you are restricted to using a file system that uses local time, use an archive format that supports local time as well; for example, use ***ZIP*** with ***FAT32***.

- **UTC file systems:** NTFS
- **UTC archive formats:** .zip with -mtc switch, 7z, tar, gzip2, iso, wim
- **Local time file systems:** FAT, FAT32
- **Local time archive formats:** rar, zip, cab

Command Detailed Reference

- Top of Page
- Current Section: Command Detailed Reference

Command	Description	Syntax	Example
a	Add	**" a "**	**7z a archive1.zip subdir** adds all files and subfolders from folder subdir to archive1.zip. The filenames in the archive will contain the subdir\ prefix. **7z a archive2.zip .\subdir*** adds all files and subfolders from folder subdir to archive2.zip. The filenames in the archive will not contain subdir\ prefix. **7z a Files.7z *.txt -r** adds all *.txt files from current folder and its subfolders to archive Files.7z.
b	Benchmark CPU and check RAM	**b [number_of_iterations] [-mmt{N}] [-md{N}] [-mm={Method}]**	See *Benchmark Command*, below
d	Delete		**7z d archive.zip *.bak -r** deletes *.bak files from archive archive.zip.
e	Extract	**" e "** **Possible Query Answers:** **Answer** / **Abbr.** / **Action** Yes / **y** / No / **n** /	7-Zip will prompt the user before overwriting existing files unless the user specifies the **-y** (Assume Yes on all queries) switch. If the user gives a **no** answer, 7-Zip will prompt for the

		Always **a** Assume YES for ALL subsequent queries of the same class Skip **s** Assume NO for ALL subsequent queries of the same class Quit **q** Quit the program	file to be extracted to a new filename. Then a **no** answer skips that file; or, **yes** prompts for new filename. See *-y Switch* (Assume "Yes" on queries), below, for automated response.
l	List	**" l "**	**7z l archive.zip** lists all files from archive archive.zip.
t	Test	**" t "**	**7z t archive.zip *.doc -r** tests *.doc files in archive archive.zip.
u	Update	**" u "**	**7z u archive.zip *.doc** updates *.doc files to archive archive.zip.
x	eXtract with full paths	**" x "**	**7z x archive.zip** extracts all files from the archive archive.zip to the current directory. **7z x archive.zip -oc:\soft *.cpp -r** extracts all *.cpp files from the archive archive.zip to c:\soft folder.

Switch Detailed Reference

- Top of Page
- Previous Section: Command Detailed Reference
- Current Section: Switch Detailed Reference

Switch	Description	Commands/Switches to Use With	Syntax	Example
--	Stop switches parsing to allow file names starting with "-"	**All**	**--**	**7z t -- -ArchiveName.7z**
-ai	Include archive filenames	**a, d, e, l, t, u, x, -r**	**-ai[recurse_type]<file_ref>** <recurse_type>	**7z t -an -air!*.7z**tests *.7z archives in the

<table>
<tr><td></td><td></td><td></td><td>Specifies how wildcards and file names in this switch must be used. If this option is not given, recursion will be not used. For more details see specification of the -r (Recurse) switch.

<recurse_type> ::= r[- | o]</td><td>current directory and all it's subdirectories</td></tr>
<tr><td>-an</td><td>Disable parsing of archive_name</td><td>e, l, t, x, -ai, -ax</td><td>-an</td><td>7z t -an -ai!*.7z -ax!a*.7ztests all *.7z archives, except a*.7z archives</td></tr>
<tr><td>-ao</td><td>Overwrite mode</td><td>e, x, -y</td><td>-ao[a | s | t | u]

Switch — Description
-aoa — Overwrite All existing files without prompt.
-aos — Skip extracting of existing files.
-aou — auto rename extracting file (for example, name.txt will be renamed to name_1.txt).
-aot — auto rename existing file (for example, name.txt will be renamed to name_1.txt).</td><td>7z x test.zip -aoa extracts all files from test.zip archive and overwrites existing files without any prompt.</td></tr>
<tr><td>-ax</td><td>Exclude archive filenames</td><td>e, l, t, x, -ai, -an</td><td>-ax[<recurse_type>] <file_ref>

<recurse_type> ::= r[- | o]
<file_ref> ::= @{listfile} | !{wildcard}</td><td>7z t -an -ai!*.7z -ax!a*.7z tests all *.7z archives, except a*.7z archives</td></tr>
<tr><td>-i</td><td>Include filenames</td><td>a, d, e, l, t, u, x, -r, -x</td><td>-i[<recurse_type>]<file_ref>
<recurse_type> ::= r[- | o]
<file_ref> ::= @{listfile} | !{wildcard}

<recurse_type>
Specifies how wildcards and file names in this switch must be used. If this option is not given, then the global value, assigned by the -r (Recurse) switch will be used. For more details see specification of the -r (Recurse) switch.

<recurse_type> ::= r[- | o]</td><td>7z a -tzip src.zip *.txt -ir!DIR1*.cpp adds to src.zip archive all *.txt files from current directory and all *.cpp files from directory DIR1 and from all it's subdirectories</td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td>
<file_ref>
Specifies filenames and wildcards, or a list file, for files to be processed.
<file_ref> ::= @{listfile} | ! {wildcard}
<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td>{listfile}</td><td>Specifies name of list file. See List file description.</td></tr>
<tr><td>{wildcard}</td><td>Specifies wildcard or filename.</td></tr>
</table>
</td><td></td></tr>
<tr><td>-m</td><td>Set Compression Method</td><td>a, d, u, -t</td><td>-m<method_parameters></td><td>See Compression Method Switch, below</td></tr>
<tr><td>-o</td><td>Set Output directory</td><td>e, x</td><td>-o{dir_path}</td><td>7z x archive.zip -oc:\Doc-extracts all files from the archive.zip archive to the c:\Doc directory

7z x *.zip -o*-extracts all *.zip archives to subfolders with names of these archives</td></tr>
<tr><td>-p</td><td>Set Password</td><td>a, d, e, t, u, x</td><td>-p{password} See Password Switch, below for complete security precautions</td><td>7z a archive.7z -psecret -mhe *.txt compresses *.txt files to archive.7z using password "secret". Also it encrypts archive headers (-mhe switch), so filenames will be encrypted.</td></tr>
<tr><td>-r</td><td>Recurse subdirectories -Specifies the method of treating wildcards and filenames on the command line</td><td>a, d, e, l, t, u, x, -i, -x</td><td>-r[- | 0]
<table>
<tr><th>Switch</th><th>Description</th></tr>
<tr><td>-r</td><td>Enable recurse subdirectories.</td></tr>
<tr><td>-r-</td><td>Disable recurse subdirectories. This option is default for all commands.</td></tr>
</table>
</td><td>7z l archive.zip *.doc -r- lists all *.doc files that belong to the archived root directory in the archive.zip archive

7z a -tzip archive.zip -r src*.cpp src*.h</td></tr>
</table>

			-ro Enable recurse subdirectories only for wildcard names.	adds all *.cpp and *.h files from directory src and all it's subdirectories to archive.zip archive
-scs	Set charset for list files	**a, u**	**-scs{UTF-8 | WIN | DOS}** **UTF-8** Unicode UTF-8 character set. **WIN** Default character set of Windows. **DOS** Default DOS (OEM) character set of Windows. Default charset is UTF-8.	**7z a archive.7z @listfile.txt -scsWIN** compresses files from listfile.txt list, that contains list of files in default character set of Windows.
-seml	Send archive by email	**a, u**	**-seml[.]** **[.]** Causes the archive to be deleted after attaching a copy of it to the email message.	**7z a archive.7z -seml a.txt** compresses the a.txt file and sends it in archive.7z by email.
-slp	Set Large Pages mode	a	**-slp[-]**	**7z a archive.7z -slp a.iso** compresses a.iso file with Large Pages mode switched on.
-slt	Show technical information	**l**	**-slt**	**7z l -slt archive.7z** shows detailed technical information for the files in archive.7z
-sfx	Create SFX archive	**a, d, u,**	**-sfx[{SFX_Module}]**	See *Creating Self-Extracting Archives*, below
-si	Read data from StdIn	**a, u**	**-si{file_name}** **{file_name}** Specifies a name that will be stored in the archive for the compressed data. If file_name is not specified, data will be stored without a name.	**7z a archive.gz -tgzip -siDoc2.txt < Doc.txt** compresses input stream from file Doc.txt to archive.gz archive using Doc2.txt file name. Note: The current version of 7-Zip does not support reading of archives from stdin
-so	Write data to StdOut	**a, e, u, x**	**-so**	**7z x archive.gz -so**

				> Doc.txt decompresses archive.gz archive to output stream and then redirects that stream to Doc.txt file **7z a dummy -tgzip -so Doc.txt > archive.gz** compresses the Doc.txt file to the 7-Zip standard output stream and writes that stream to archive.gz file
-ssc	Set Sensitive Case mode	**a, d, e, l, t, u, x**	**-scs[-]** **-ssc** Set case-sensitive mode. It's default for Posix/Linux systems **-ssc-** Set case-insensitive mode. It's default for Windows systems	**7z a archive.7z A*.txt -ssc -r** compresses all A*.txt files from current directory and all it's subdirectories. That command doesn't compress a*.txt files.
-ssw	Compress files open for writing	**a, u**	**-ssw**	**7z a archive.7z -ssw *.txt** compresses all *.txt files in current folder including files open for writing by another applications
-t	Type of archive	**a, d, e, l, t, u, x**	**-t{archive_type}**	See *Type of Archive Switch*, below
-u	Update options	**a, d, u**	**-u[-]<action_set>[! {new_archive_name}]** **<action_set>** ::= **<state_action>**... **<state_action>** ::= **<state> <action>** **<state>** ::= **p | q | r | x | y | z | w** **<action>** ::= **0 | 1 | 2 | 3**	See *Update Options Switch*, below
-v	Create Volumes	**a**	**-v{Size}[b | k | m | g]** **{Size}[b | k | m | g]** Specifies volume size in Bytes, Kilobytes (1 Kilobyte = 1024	**7z a a.7z *.txt -v10k -v15k -v2m** creates multi-volume a.7z archive. First volume will be 10 KB, second

			bytes), Megabytes (1 Megabyte = 1024 Kilobytes) or Gigabytes (1 Gigabyte = 1024 Megabytes). if you specify only {Size}, 7-zip will treat it as bytes. Multiple **-v** switches supported	will be 15 KB, and all others will be 2 MB. Complete the archiving task before using on copying volumes.
-w	Set Working directory	**a**, **d**, **u**	**-w[{dir_path}]** **{dir_path}** Specifies the destination directory path. It's not required that a path end with a backslash. If <*dir_path*> is not assigned, then 7-Zip will use the Windows temporary directory.	Sets the working directory for the temporary base archive. By default, 7-Zip builds a new base archive file in the same directory as the old base archive file. By specifying this switch, you can set the working directory where the temporary base archive file will be built. After the temporary base archive file is built, it is copied over the original archive; then, the temporary file is deleted. **7z a -tzip archive.zip *.cpp -wc:\temp** adds *.cpp files to the archive.zip archive, creating a temporary archive in c:\temp folder.
-y	Assume "Yes" on all Queries	**e**, **x**, **-ao**	**-y**	**7z x src.zip -y** extracts all files from src.zip archive. All overwrite queries will be suppressed and files on disk with same filenames as in archive will be overwritten.

Wildcard Parsers

Top of Page
Previous Section: Switch Detailed Reference
Current Section: Wildcard Parsers

- **"*"** means a sequence of arbitrary characters
- **"?"** means any character
- For example:

"*.txt" means all files with an extension of **".txt"**

"?a*" means all files with a second character of **"a"**

"*1*" means all names that contain the character **"1"**

"*.*.*" means all names that contain at least two **"."** characters

The default wildcard, **"*"**, will be used if there is no filename or wildcard in the command line.

Wildcards containing spaces must be placed in quotes. For example:

- "Dir\Program files*"
- Dir\"Program files"*

List Files

You can supply one or more filenames or wildcards for special list files (files containing lists of files). Each filename in such a list file must be separated by a new line symbol.

For list files, 7-Zip uses UTF-8 encoding by default and supports multiple lists files. Use **-scs** switch to change the encoding. For example, if the file "*listfile.txt*" contains the

following:

- My programs*.cpp
- Src*.cpp

then the command **7z a -tzip *archive.zip @listfile.txt*** adds to the archive named "*archive.zip*" all "*.*cpp*" files from the directories named "*My programs*" and "*Src*".

Archive Formats and Parameters:

Methods, Filters and Their Paramters:

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

(higher values correspond to higher compression ratios but slower speeds)

For example, **mf=HC4** and **mc=10000** can provide almost the same compression ratio as **mf=BT4**

7z Archive Format Parameters

- Top of Page
- Current Section: 7z Archive Format Parameters

Syntax: Use the **-t7z** switch after the "**a**" command or specify "***archive_name.7z***" to create a 7z archive. Use a separate **-m** switch for each parameter when adding the parameters below to the command line. For example: **7za a -t7z Encrypt.7z Test8.txt -mx=7 -mhe=on**

<table>
<tr><th>Parameter</th><th>Default</th><th>Description</th></tr>
<tr><td rowspan="2">x=[0 | 1 | 3 | 5 | 7 | 9]</td><td rowspan="2">5</td><td>Sets level of compression.
<table>
<tr><th>Level</th><th>Method</th><th>Dictionary</th><th>FastBytes</th><th>MatchFinder</th><th>Filter</th><th>Description</th></tr>
<tr><td>0</td><td>Copy</td><td></td><td></td><td></td><td></td><td>No compression.</td></tr>
<tr><td>1</td><td>LZMA</td><td>64 KB</td><td>32</td><td>HC4</td><td>BCJ</td><td>Fastest compressing</td></tr>
<tr><td>3</td><td>LZMA</td><td>1 MB</td><td>32</td><td>HC4</td><td>BCJ</td><td>Fast compressing</td></tr>
<tr><td>5</td><td>LZMA</td><td>16 MB</td><td>32</td><td>BT4</td><td>BCJ</td><td>Normal compressing</td></tr>
<tr><td>7</td><td>LZMA</td><td>32 MB</td><td>64</td><td>BT4</td><td>BCJ</td><td>Maximum compressing</td></tr>
<tr><td>9</td><td>LZMA</td><td>64 MB</td><td>64</td><td>BT4</td><td>BCJ2</td><td>Ultra compressing</td></tr>
</table></td></tr>
</table>

s=[off | on | [e] [{N}f] [{N}b | {N}k | {N}m | {N}g]	**on**	Sets solid mode. The default mode is s=on. In solid mode, files are grouped together. Usually, compressing in solid mode improves the compression ratio. Use a larger block size for data types that are more efficiently compressed, like text. **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 with these default limits for the associated Compression Level: ▪ 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 current version of 7-Zip doesn't support updating of solid archives if it requires repacking solid blocks. Example: **s=100f10m** sets solid mode with 100 files and 10MB limits for one solid block.
f=[off | on]	**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]	**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]	**off**	Enables or disables archive header encryption. Default is **he=off**.
b{C1} [s{S1}]:{C2} [s{S2}]		Sets binding between coders. Binds the output stream S1 in coder C1 with input stream S2 in coder C2. If the stream number is not specified, stream 0 will be used. Usually the coder has one input stream and one output stream. In 7z some coders can have multiple input and output streams. Eg; the *BCJ2* encoder has one input stream and four output streams.
{N}= {MethodID} [:param1] [:param2] [..]	**LZMA**	Sets the compression method: LZMA, PPMd, BZip2, Deflate, BCJ, BCJ2, Copy. Where {N} is the order of the methods, also used to associate parameters with methods. Numbers must begin from 0, and are used in numerical order.
mt=[off | on | {N}]	**on**	Sets multithreading mode. If you have a multiprocessor or multicore system, you can get an increase with this switch. 7-Zip supports multithread mode only for LZMA compression and BZIP2 compression/decompression. If you specify **{N}**, for example **mt=4**, 7-Zip tries to use 4 threads. LZMA uses only 2 threads.
tc=[off | on]	**off**	Stores file creation timestamps.

7z Archive Type Supported Compression Methods: see the Choose a Compression Method section of *Creating an Archive*

- Top of Page
- Previous Section: 7z Archive Format Parameters

- Current Section: 7z Archive Compatible Filters

7z Archive Type Supported Filters:

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

Syntax: Use one -m switch for each parameter. For example: 7z a -t7z Archive.7z TestFile.txt -m

MethodID	Filter Description
BCJ2 (see parameters below)	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

ZIP Archive Format Parameters:

- Top of Page
- Current Section: ZIP, BZIP2, and GZIP Archive Parameters

Syntax: Use the **-tzip** switch after the "**a**" command or specify "***archive_name.zip***" to create a Zip archive.

Use a separate **-m** switch for each parameter when adding the parameters below to the command line. For example: **7z a -tzip Encrypt.zip Test8.txt -mx=7 -mm=Deflate64**

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 the local code page.

<table>
<tr><th>Parameter</th><th>Default</th><th>Description</th></tr>
<tr><td>x=[0 | 1 | 3 | 5 | 7 | 9]</td><td>5</td><td>Sets the level of compression.
<table>
<tr><th colspan="4">ZIP and GZIP Compression Parameters</th></tr>
<tr><th>Level</th><th>NumFastBytes</th><th>NumPasses</th><th>Description</th></tr>
<tr><td>1</td><td rowspan="3">32</td><td rowspan="3">1</td><td>Fastest</td></tr>
<tr><td>3</td><td>Fast</td></tr>
<tr><td>5</td><td>Normal</td></tr>
<tr><td>7</td><td>64</td><td>3</td><td>Maximum</td></tr>
<tr><td>9</td><td>128</td><td>10</td><td>Ultra</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Copy (No Compression)</td></tr>
</table>
<table>
<tr><th colspan="4">BZIP2 Compression Parameters</th></tr>
<tr><th>Level</th><th>NumFastBytes</th><th>NumPasses</th><th>Description</th></tr>
<tr><td>1</td><td>100000</td><td rowspan="2">1</td><td>Fastest</td></tr>
<tr><td>3</td><td>500000</td><td>Fast</td></tr>
<tr><td>5</td><td rowspan="3">900000</td><td rowspan="2">Normal</td></tr>
<tr><td>7</td><td>2</td><td>Maximum</td></tr>
<tr><td>9</td><td>7</td><td>Ultra</td></tr>
</table></td></tr>
<tr><td>m={MethodID} (see Method Parameters below)</td><td>Deflate</td><td>Sets a method: Copy, Deflate, Deflate64, BZip2, LZMA.

GZIP - Deflate method only</td></tr>
</table>

fb={NumFastBytes}	32	Sets number of Fast Bytes for Deflate encoder - Valid values: [3,258] for Deflate; [3,257] for Deflate64 A large fast bytes parameter can significantly increase the compression ratio for files which contain long identical sequences of bytes.
pass={NumPasses}	1	Sets number of Passes for Deflate encoder - Valid values: [1,15] for Deflate; [1,10] for BZIP2.
d={Size}[b|k|m]	900000	Sets Dictionary size for BZip2 - Specify size in bytes, KB, MB; max = 9×10^5 bytes No [b|k|m] parameter => DictionarySize = 2^{Size} bytes
mt=[off | on | {N}]	on	Sets multithreading 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.
em={EncryptionMethodID}	ZipCrypto	Sets a encryption method: ZipCrypto, AES128, AES192, AES256
tc=[off | on]	off	Stores NTFS timestamps for files: Modification time, Creation time, Last access time.
cl=[off | on]	off	7-Zip always uses local code page for file names.
cu=[off | on]	off	7-Zip uses UTF-8 for file names that contain non-ASCII symbols.

Switches

- Top of Page
- Previous Section: ZIP, BZIP2, and GZIP Archive Parameters
- Current Section: Command-line Switches

Type of Archive Switch

Syntax: -t{archive_type}

{archive_type} Specifies the type of archive: 7z, zip, gzip, bzip2, tar

If the **-t{archive_type}** switch is not specified, 7-Zip uses extension of archive to detect the type of archive.

Note: **gzip** or **bzip2** formats support only one file per archive. If you want to compress more than one file to these formats, create a **tar** archive first, and then compress it with your selected format.

-t Switch Examples:

7z a -tzip archive.zip *.txt

adds all *.txt files from current directory to zip archive archive.zip.

7z t -t7z.split archive.7z.001

tests all files in archive.7z.001. It also checks that archive is multivolume .7z archive.

7z x -tiso archive.iso

extracts files from archive.iso open as ISO archive.

7z x -tudf archive.iso

extracts files from archive.iso open as UDF archive.

Commands that can be used with this switch

a (Add), **d** (Delete), **e** (Extract), **l** (List), **t** (Test), **u** (Update), **x** (Extract with full paths)

Password Switch

- Top of Page
- Current Section: Password Switch

Syntax: -p{*password*}

-for maximum security, don't enter the password inside the switch argument; just enter "-p" to activate the switch then the program will prompt you to enter a password with echoing turned off. In this way your command-line file can't be searched for the password.

-p Switch Examples:

- ***7z a -psecret archive.7z -mhe *.txt***
- compresses ****.txt*** files to ***archive.7z*** using the password "***secret***. It also encrypts archive headers (***-mhe***), so filenames will be encrypted.
- **7z x archive.zip -psecret**
- extracts all files from ***archive.zip*** using the password "***secret***".

Commands that can be used with this switch:

a (Add), **d** (Delete), **e** (Extract), **t** (Test), **u** (Update), **x** (Extract with full paths)

Compression Method Switch

Syntax: -m<*method_parameters*> Where *method_parameters* follow the format for their Archive Type, below.

Parameters must be in one of the following forms:

- ***{ParamName}={ParamValue}***
- ***{ParamName}{ParamValue}***, where {*ParamName*} is a string and {*ParamValue*} is a number.

Compression Methods and Their Parameters

LZMA Compression Method Parameters:

Syntax: -m<*method_parameters*>

- Top of Page
- Current Section: LZMA Compression Method Parameters

Parameter	Default	Description
a=[0|1]	1	Sets Compressing Mode: 0 = fast, 1 = normal. Default value is 1
d={size}[b|k|m]	24	Sets Dictionary size: Specify size in bytes, KB, MB; max = 1GB (2^{30} bytes) Default: 24 (16MB) in Normal Mode, 25 (32MB) in Maximum Mode (-mx=7) and 26 (64MB) in Ultra Mode (-mx=9) No [b|k|m] parameter => DictionarySize = 2^{Size} bytes

<table>
<tr><td></td><td></td><td>N bytes of RAM are needed for decompressing a file compressed with LZMA Dictionary, size N.</td></tr>
<tr><td>mf={MF_ID}</td><td>bt4</td><td>Sets Match Finder:

Algorithms from hc* MF_ID group don't provide a good compression ratio but achieve good results in combination with

fast mode (a=0). Memory requirements depend on dictionary size, parameter "d", below:

<table>
<tr><th>MF_ID</th><th>RAM Requirements</th><th>Match Finder Description</th></tr>
<tr><td>bt2</td><td>d*9.5 + 4MB</td><td>Binary Tree with 2 bytes hashing</td></tr>
<tr><td>bt3</td><td>d*11.5 + 4MB</td><td>Binary Tree with 3 bytes hashing</td></tr>
<tr><td>bt4</td><td>d*11.5 + 4 MB</td><td>Binary Tree with 4 bytes hashing</td></tr>
<tr><td>hc4</td><td>d*7.5 + 4MB</td><td>Hash Chain with 4 bytes hashing</td></tr>
</table></td></tr>
<tr><td>fb={N}</td><td>32</td><td>Sets the number of Fast Bytes - Valid values: [5, 273]

Default: 32 in Normal Mode, 64 in Maximum and Ultra Modes</td></tr>
<tr><td>mc={N}</td><td>32 (mc=0)</td><td>Sets Number of Cycles for Match Finder - Valid values: [0, 10^9]

Default: BT* Match Finders - (16 + number_of_fast_bytes/2)

Default: HC4 Match Finder - (8 + number_of_fast_bytes/4)</td></tr>
<tr><td>lc={N}</td><td>3</td><td>Sets number of Literal Context bits (high bits of previous literal) - Valid values: [0, 8] Eg; lc=4 for larger files</td></tr>
<tr><td>lp={N}</td><td>0</td><td>Sets number of Literal Pos bits (low bits of current position for literals) - Valid values: [0, 4]

Use for periodic data where T=$2^{(lp)}$ Eg; for 32-bit (4 bytes) periodic data, use lp=2.

Often it's better to set lc=0, if you change the lp switch</td></tr>
<tr><td>pb={N}</td><td>2</td><td>Set number of Pos Bits (low bits of current position) - Valid values: [0, 4]

Use for periodic data where T=$2^{(lp)}$</td></tr>
</table>

PPMd Compression Method Parameters:

- Top of Page
- Previous Section: LZMA Compression Method Parameters
- Current Section: PPMd Compression Method Parameters

PPMd is a PPM-based algorithm based on Dmitry Shkarin's PPMdH source code. It provides a very good compression ratio for plain text files while maintaining the same speed and memory requirements for both compression and extraction.

Parameter	Default	Description
mem={Size}[b|k|m]	24	Sets size of memory used for the PPMd method

		Specify size in bytes, KB, MB; max = 2GB (2^{31})
o={Size}	6	Sets the model order - Valid values: [2,32]

BCJ2 Filter Parameters:

- Top of Page
- Previous Section: PPMd Compression Method Parameters
- Current Section: BCJ2 Filter Parameters

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 with BCJ2, 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.

-m Switch Examples

- Top of Page
- Current Section: Compression Method Switch 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=21 -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 multi-threading 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)

Create Self-Extracting Archives Switch

- Top of Page

Current Section: Self-Extracting Switch

Syntax:

-sfx[{SFX_Module}]

Specifies the SFX module that will be combined with the archive. This module must be placed in the same directory as the 7z.exe. If {SFX_Module} is not assigned, 7-Zip will use standard console SFX module 7zCon.sfx.

SFX_Module	Description
7z.sfx	Windows version.
7zCon.sfx	Console version.
7zS.sfx	Windows version for installers.
7zSD.sfx	Windows version for installers (uses MSVCRT.dll).

All SFX modules are uncompressed. You can use UPX program (http://upx.sourceforge.net) to compresss such modules. After compressing by the UPX program, the size of the sfx module will be reduced to 40-50% of its original size.

SFX modules for installers

SFX modules for installers are included in an external package (7z_extra). You can download these modules from www.7-zip.org. SFX modules for installers (7zS.sfx and 7zSD.sfx) allow you to create your own installation program. Such a module extracts the archive to the user's temp folder, and runs a specified program, and removes the temp files after the program finishes. A self-extracting archive for installers must be created as joining 3 files: SFX_Module, Installer_Config, 7z_Archive. In addition, an optional file, Installer_Config, is allowed. You can use the following command to create an installer self-extracting archive:

copy /b 7zS.sfx + config.txt + archive.7z archive.exe

An optimally small installation package size can be achieved, if the installation files are uncompressed before including them in the 7z archive.

-y switch for installer module specifies quiet mode extraction.

Installer Config file format

This config file contains commands for the Installer. The file begins with the string **;!@Install@!UTF-8!** and ends with **;!@InstallEnd@!**. The file must be written in UTF-8 encoding. The file contains any or all these string pairs:

ID_String="Value"

ID_String	Description
Title	Title for messages
BeginPrompt	Begin Prompt message
Progress	Value can be "yes" or "no". Default value is "yes".
RunProgram	Command for executing. Default value is "setup.exe". Substring **%%T** will be replaced with path to temporary folder, where files were extracted
Directory	Directory prefix for "RunProgram". Default value is ".\\"
ExecuteFile	Name of file for executing
ExecuteParameters	Parameters for "ExecuteFile"

Top of Page
Current Section: Self-Extracting Switch

You may omit any pair.

There are two ways to run a installation program: **RunProgram** and **ExecuteFile**. Use **RunProgram**, if you want to run a program from the .7z archive. Use **ExecuteFile**, if you want to open a document from the .7z archive, or if you want to execute a command from Windows.

If you use **RunProgram**, and if you specify empty directory prefix: **Directory**="", the system searches for the executable file in the following sequence:

1. The directory from which the application (installer) loaded.
2. The temporary folder, where files were extracted.
3. The Windows system directory.

Config file Examples

```
;!@Install@!UTF-8!
Title="7-Zip 4.00"
BeginPrompt="Do you want to install the 7-Zip 4.00?"
RunProgram="setup.exe"
;!@InstallEnd@!
```

```
;!@Install@!UTF-8!
Title="7-Zip 4.00"
BeginPrompt="Do you want to install the 7-Zip 4.00?"
ExecuteFile="7zip.msi"
;!@InstallEnd@!
```

```
;!@Install@!UTF-8!
Title="7-Zip 4.01 Update"
BeginPrompt="Do you want to install the 7-Zip 4.01 Update?"
ExecuteFile="msiexec.exe"
ExecuteParameters="/i 7zip.msi REINSTALL=ALL REINSTALLMODE=vomus"
;!@InstallEnd@!
```

-sfx Switch Examples

7z a -sfx a.exe *.txt

adds *.txt files to self extracting archive a.exe using the default console SFX module.

7z a -sfx7z.sfx a.exe *

adds all files to self extracting archive a.exe with module 7z.sfx using windows version of SFX mudule.

Commands that can be used with this switch

a (Add), **d** (Delete), **u** (Update)

Update Options Switch

- Top of Page
- Previous Section: Self-Extracting Switch
- Current Section: Update Options Switch

Syntax:

-u[-]<action_set>[!{new_archive_name}]

<action_set> ::= **<state_action>**...

<state_action> ::= **<state><action>**

<state> ::= **p | q | r | x | y | z | w**

<action> ::= **0 | 1 | 2 | 3**

Multiple update switches are supported. 7-Zip can create any number of new archives during one operation.

Time Dependent Considerations: see "Time Dependent Considerations", above, for possible time conflicts when creating updates.

Parameters:

If you don't specify a **!*{new_archive_name}*** option, then all options will refer to the main archive (the archive assigned on the command line after the 7z command). If you specify **!*{new_archive_name}*** option, then 7-Zip also will create a new archive with the specified name and all options will refer to that new archive.

" - " - Disables any updates in the base archive which is the archive assigned by "**base_archive_name**" on the command line. See Command line syntax for more details.

<action_set> - By default, the *action set* for each new archive is assigned as the *action set* of the main command. There are 3 different *action sets* for commands: **a** (Add), **d** (Delete), **u** (Update). You can overload any **<state_action>** pair.

{new_archive_name} - Specifies the path name of the new archive to be created. All options in this switch will refer to this new archive. If not assigned, then all options in this switch will refer to the base archive of the command.

<state> ::= p | q | r | x | y | z | w - Specifies the state of a particular file to be processed.

- Top of Page
- Current Section: Update Options Switch

For each unique filename there are 6 variants of state:

<state>	State condition	File on Disk	File in Archive
p	File exists in archive, but is not matched with wildcard.		Exists, but is not matched
q	File exists in archive, but doesn't exist on disk.	Doesn't exist	Exists
r	File doesn't exist in archive, but exists on disk.	Exists	Doesn't exist
x	File in archive is newer than the file on disk.	Older	Newer
y	File in archive is older than the file on disk.	Newer	Older
z	File in archive is same as the file on disk	Same	Same
w	What file is newer - can't be detected (times are the same, sizes are different)	?	?

<action> ::= 0 | 1 | 2 | 3 - Specifies the *action* for a given <*state*>. For each *state* you can specify one of the 3 variants of *actions*.

Actions:

<action>	Description
0	Ignore file (don't create item in new archive for this file)
1	Copy file (copy from old archive to new)
2	Compress (compress file from disk to new archive)
3	Create Anti-item (item that will delete file or directory during extracting). This feature is supported only in 7z format.

Any update command (such as **a** (Add), **d** (Delete), **u** (Update)) can be assigned with variants of **Actions**.

Variants of Actions for commands that use the update switch (a, d, u):

command \ <state>	p	q	r	x	y	z	w
d (Delete)	1	0	0	0	0	0	0
a (Add)	1	1	2	2	2	2	2
u (Update)	1	1	2	1	2	1	2
Freshen	1	1	0	1	2	1	2

Synchronize	1	0	2	1	2	1	2

-u Switch Examples

- Top of Page
- Current Section: Update Options Switch

7z u c:\1\exist.7z -u- -up0q3x2z0!c:\1\update.7z *

creates a new archive update.7z and writes to this archive all files from current directory which differ from files in exist.7z archive. exist.7z archive will not be changed.

7z u c:\1\exist.7z -up0q3x2z0!c:\1\update.7z * -ms=off

creates a new archive update.7z and writes to this archive all files from the current directory which differ from files in exist.7z archive.

Note: The current version of 7-Zip cannot change an archive created with the solid option switched on. To update a .7z archive, you must create and update the archive in non-solid mod (**-ms=off** switch).

Commands that can be used with this switch:

a (Add), **d** (Delete), **u** (Update),

Exit Codes from 7-Zip

7-Zip returns the following exit codes:

Code	Meaning
0	No error
1	Warning (Non fatal error(s)). For example, one or more files were locked by some other application, so they were not compressed.
2	Fatal error
7	Command line error
8	Not enough memory for operation
255	User stopped the process

- Top of Page
- Current Section: Archive Format Comparison

Archive Format Comparison

7z (7ZIP's native format), ***TAr*** and ***ZIP*** formats are available with various compression methods. See *Type of Archive Switch* for additional information**.**

Specification	7z Archive Format	TAr Archive Format	ZIP Archive Format
Compression Ratio	100%	n/a	140% (with LZMA method)
Compression Levels	Store, Fastest, Fast, Normal, Maximum, Ultra	Store	Store, Fastest, Fast, Normal, Maximum, Ultra
Compression	LZMA (default), PPMd,	BZIP2, GZIP	Deflate (default), Deflate64,

Methods	BZIP2		BZIP2, LZMA, PPMd
Dictionary Size	64KB, 1MB, 2MB, 3MB, 4MB, 6MB, 8MB, 12MB, 16MB, 24MB, 32MB, 48MB, 64MB	Specify in parameters	32KB
Word Size	8, 12,16, 24, 32, 48, 64, 96, 128, 192, 256, 273	Specify in parameters	8, 12,16, 24, 32, 48, 64, 96, 128, 192, 256, 258
Solid Block Size	Non-solid, 1MB, 2MB, 4MB, 8MB, 16MB, 32MB, 64MB, 128MB, 256MB, 512MB, 1GB, 2GB, 4GB, 8GB, 16GB, 32GB, 64GB, Solid	Not supported	Not supported
CPU Threads	1/2, 2/2	Not supported	1/2, 2/2, 3/2, 4/2
Split to volumes, bytes	4480M - DVD, 700M - CD, 650M - CD, 145,7664 - 3.5" floppy	4480M - DVD, 700M - CD, 650M - CD, 145,7664 - 3.5" floppy	4480M - DVD, 700M - CD, 650M - CD, 145,7664 - 3.5" floppy
Update Mode	Add and replace files, Update and Add Files, Freshen Existing Files, Synchronize Files	Add and replace files, Update and Add Files, Freshen Existing Files, Synchronize Files	Add and replace files, Update and Add Files, Freshen Existing Files, Synchronize Files
Options	Create SFX archive, Compress Shared Files	Compress Shared Files	Compress Shared Files
Encryption	AES-256	Not supported	AES-256 or ZIPCrypto

Archive Types in Detail

Top of Page
Current Section: 7z Archives

7z Archives

Parameter	Default	Description
x=[0 | 1 | 3 | 5 | 7 | 9]	5	Sets level of compression.
s=[off | on | [e] [{N}f] [{N}b | {N}k | {N}m | {N}g]	on	Sets solid mode.
f=[off | on]	on	Enables or disables compression filters for executable files.
hc=[off | on]	on	Enables or disables archive header compressing.
he=[off | on]	off	Enables or disables archive header encryption.
b{C1}		Sets binding beetwen coders.

[s{S1}]:{C2}[s{S2}]		
{N}={MethodID}[:param1][:param2][..]	LZMA	Sets a method: LZMA, PPMd, BZip2, Deflate, BCJ, BCJ2, Copy.
mt=[off | on | {N}]	on	Sets multithreading mode.
tc=[off | on]	off	Stores file creation timestamps.

Compression Level Parameter for 7z Archives: x=[0 | 1 | 3 | 5 | 7 | 9] Sets the 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

Solid Mode Parameter for 7z Archives: 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

- Top of Page
- Current Section: 7z Archives

Solid Block Size:

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 current version of 7-Zip doesn't support updating of solid archives, if it requires repacking solid blocks. Eg:

s=100f10m sets ***solid mode*** with 100 files & 10 MB limits per one solid block.

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

f=[off | on] **Parameter for 7z Archives:**

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] **Parameter for 7z Archives:**

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] **Parameter for 7z Archives:**

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

{N} **Parameter for 7z Archives:**

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}] **Parameter for 7z Archives:**

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.

Usally 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}] **Parameter for 7z Archives:**

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 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.

- Top of Page
- Current Section: 7z Archives

{N}={MethodID}[:param1][:param2] ... [:paramN] **Parameter for 7z Archives:**

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 for 7z Archives:

MethodID	Description
LZMA	Based on the LZ algorithm, provides fast compression, very fast decompression and high compression ratios
PPMd	Provides high speeds and compression ratios especially with plain text files
BZip2	Uses the versatile BWT algorithm
Deflate	LZ+Huffman algorithm
Copy	No compression

Supported filters for 7z Archives: Filters increase the compression ratio for some types of files. Filters must be used with one of the compression method (for example, BCJ + LZMA).

MethodID	Description
BCJ	converter for x86 executables

BCJ2	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

BZIP2 Archives

- Top of Page
- Previous Section: 7z Archives
- Current Section: BZIP2 Archives

Parameter	Default	Description
x=[1 | 3 | 5 | 7 | 9]	5	Sets level of compression.
pass={NumPasses}	1	Sets number of Passes for Bzip2 encoder.
mt=[off | on | {N}]	on	Sets multithreading mode.

Compression Level Parameter for BZIP2 Archives: x=[1 | 3 | 5 | 7 | 9] Sets the level of compression:

Level	NumPasses	Description
5	1	Default compression method.
7	2	Maximum compression method.
9	7	Ultra compression method.

pass={NumPasses} Parameter for BZIP2 Archives:

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 but a slower compression process.

mt=[off | on | {N}] Parameter for BZIP2 Archives:

Sets ***multi-thread 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.

GZIP Archives

GZIP uses the same parameters as ***ZIP***, but ***GZIP*** compresses only with ***Deflate*** method.

ZIP Archives

- Top of Page
- Current Section: ZIP Archives

Parameter	Default	Description
x=[0 | 1 | 3 | 5 | 7 | 9]	5	Sets level of compression.
m={MethodID}	Deflate	Sets a method: Copy, Deflate, Deflate64, BZip2, LZMA.
fb={NumFastBytes}	32	Sets number of Fast Bytes for Deflate encoder.
pass={NumPasses}	1	Sets number of Passes for Deflate encoder.
d={Size}[b|k|m]	900000	Sets Dictionary size for BZip2
mt=[off | on | {N}]	on	Sets multithreading mode.
em={EncryptionMethodID}	ZipCrypto	Sets a encryption method: ZipCrypto, AES128, AES192, AES256
tc=[off | on]	off	Stores NTFS timestamps for files: Modification time, Creation time, Last access time.

cl=[off | on]	off	7-Zip always uses local code page for file names.
cu=[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 the local code page.

Compression Level Parameter for *ZIP* Archives: x=[0 | 1 | 3 | 5 | 7 | 9] Sets level of compression. **x=0** means ***Copy mode*** (no compression).

ZIP Archive Available Compression Methods

1. Deflate / Deflate64 settings for ***ZIP*** Archives: **x=1** and **x=3** with ***Deflate*** method set ***fast mode*** for compression.

- Top of Page
- Current Section: ZIP Archive Compression Methods

Level	NumFastBytes	NumPasses	Description
1	32	1	Fastest
3			Fast
5			Normal
7	64	3	Maximum
9	128	10	Ultra

2. BZip2 settings for ***ZIP*** Archives:

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

fb={NumFastBytes} Parameter **for *ZIP* Archives using BZip2:**

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} Parameter **for *ZIP* Archives using BZip2:**

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.

- Top of Page
- Current Section: ZIP Archives

d={Size}[b|k|m] Parameter **for *ZIP* Archives using BZip2**

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^Size bytes.

mt=[off | on | {N}] Parameter **for *ZIP* Archives using BZip2:**

Sets ***multi-thread 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***.

Compression Methods in Detail

- Top of Page
- Current Section: BZIP2 Compression Method

BZIP2 Compression Method

BZIP2 uses the BWT algorithm for compression providing fast speeds and relatively good compression ratios.

Parameter	Default	Description
x=[1 | 3 | 5 | 7 | 9]	5	Sets level of compression.
pass={NumPasses}	1	Sets number of Passes for Bzip2 encoder.
mt=[off | on | {N}]	on	Sets multithreading mode.

x=[1 | 3 | 5 | 7 | 9] Sets the level of compression

Level	NumPasses	Description
5	1	Default compression method.
7	2	Maximum compression method.
9	7	Ultra compression method.

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.

- Top of Page
- Current Section: BZIP2 Compression Method

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.

Deflate and Deflate64 Compression Methods

Deflate employs the LZ77 algorithm providing relatively quick speeds and moderate compression ratios.

Deflate64 increases the dictionary size for Deflate and achieves better compression.

Deflate/Deflate64 Settings:

Level	NumFastBytes	NumPasses	Description
1	32	1	Fastest
3			Fast
5			Normal
7	64	3	Maximum
9	128	10	Ultra

GZip

GZip uses the same parameters as Zip, but GZip compresses only with Deflate method in the 7Zip Windows graphic user interface version. Command line version uses GZip method or Deflate.

LZMA Compression Method

LZMA is the default and general compression method of **7z** format. The main features of the **LZMA** method:

Top of Page
Current Section: LZMA Compression Method

- High compression ratio
- Variable dictionary size (up to 4 GB)
- Compression speed: about 1 MB/s on 2 GHz CPU
- Decompression speed: about 10-20 MB/s on 2 GHz CPU
- Small memory requirement for decompression (depends from dictionary size)
- Small code size for decompression: about 5 KB
- Supports multi-threading and P4's hyper-threading

LZMA is based on Lempel-Ziv algorithm that 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=[0|1]	1	Sets compressing mode
d={Size}[b|k|m]	24	Sets Dictionary size
mf={MF_ID}	bt4	Sets Match Finder
fb={N}	32	Sets number of Fast Bytes
mc={N}	32	Sets Number of Cycles for Match Finder
lc={N}	3	Sets number of Literal Context bits - [0, 8]
lp={N}	0	Sets number of Literal Pos bits - [0, 4]
pb={N}	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^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).

- Top of Page
- Current Section: LZMA Compression Method

MF_ID	Memory	Description
bt2	d*9.5 + 4 MB	Binary Tree with 2 bytes hashing.
bt3	d*11.5 + 4 MB	Binary Tree with 3 bytes hashing.
bt4	d*11.5 + 4 MB	Binary Tree with 4 bytes hashing.
hc4	d*7.5 + 4 MB	Hash Chain with 4 bytes hashing.

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 + number_of_fast_bytes / 2) for BT* match finders and (8 + 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^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^value (where lp=value).

PPMd Compression Method

PPMd is a PPM-based algorithm and provides a very good compression ratio and speed for plain text files. Compression/decompression speeds and memory requirements are identical.

- Top of Page
- Current Section: PPMd Compression Method

Parameter	Default	Description
mem={Size}[b|k|m]	24	Sets size of used memory for PPMd.

o={Size}	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 2GB = 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^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.

Compression Method Filters

- Top of Page
- Current Section: Compression Method Filters

MethodID	Description
BCJ	converter for x86 executables
BCJ2	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