

Executable compression

By Contributors to Wikimedia projects

Published: 2005-04-22 · Archived: 2026-04-05 13:24:45 UTC

From Wikipedia, the free encyclopedia

"POPCOM" redirects here. For the music fair, see [Popkomm](#).

Executable compression is any means of [compressing](#) an [executable](#) file and combining the compressed data with decompression code into a single executable. When this compressed executable is executed, the decompression code recreates the original code from the compressed code before executing it. In most cases this happens transparently so the compressed executable can be used in exactly the same way as the original. Executable compressors are often referred to as **executable packers**, **runtime packers**, **software packers**, **software protectors**, or even "[polymorphic packers](#)" and "[obfuscating tools](#)".

A compressed executable can be considered a [self-extracting archive](#), where a compressed executable is packaged along with the relevant decompression code in an executable file. Some compressed executables can be decompressed to reconstruct the original program file without being directly executed. Two programs that can be used to do this are CUP386 and UNP.^{[*[citation needed](#)*]}

Most compressed executables decompress the original code in memory and most require slightly more memory to run (because they need to store the decompressor code, the compressed data and the decompressed code). Moreover, some compressed executables have additional requirements, such as those that write the decompressed executable to the file system before executing it.

Executable compression is not limited to binary executables, but can also be applied to scripts, such as [JavaScript](#). Because most scripting languages are designed to work on [human-readable code](#), which has a high [redundancy](#), compression can be very effective and as simple as replacing long names used to identify [variables](#) and [functions](#) with shorter versions and/or removing [white-space](#).

Advantages and disadvantages

[\[edit\]](#)

[Software](#) distributors use executable compression for a variety of reasons, primarily to reduce the [secondary storage](#) requirements of their software; as executable compressors are specifically designed to compress executable code, they often achieve better [compression ratio](#) than standard [data compression](#) facilities such as [gzip](#), [zip](#) or [bzip2](#).^{[*[citation needed](#)*]} This allows software distributors to stay within the constraints of their chosen distribution media (such as [CD-ROM](#), [DVD-ROM](#), or [floppy disk](#)), or to reduce the time and bandwidth customers require to access software distributed via the [Internet](#).

Executable compression is also frequently used to deter [reverse engineering](#) or to obfuscate the contents of the executable (for example, to hide the presence of [malware](#) from [antivirus scanners](#)) by proprietary methods of compression and/or added [encryption](#). Executable compression can be used to prevent direct [disassembly](#), mask [string literals](#) and modify signatures. Although this does not eliminate the chance of reverse engineering, it can make the process more costly.

A compressed executable requires less storage space in the file system, thus less time to transfer data from the file system into memory. On the other hand, it requires some time to decompress the data before execution begins. However, the speed of various storage media has not kept up with average processor speeds, so the storage is very often the bottleneck. Thus the compressed executable will load faster on most common systems. On modern desktop computers, this is rarely noticeable unless the executable is unusually big, so loading speed is not a primary reason for or against compressing an executable.

On operating systems which [page](#) executable images on demand from the disk, compressed executables make this process less efficient. The decompressor stub allocates a block of memory to hold the decompressed data, which stays allocated as long as the executable stays loaded, whether it is used or not, competing for memory resources with other applications all along. If the operating system uses a swap file, the decompressed data has to be written to it to free up the memory instead of simply discarding unused data blocks and reloading them from the executable image if needed again. This is usually not noticeable, but it becomes a problem when an executable is loaded more than once at the same time—the operating system cannot reuse data blocks it has already loaded, the data has to be decompressed into a new memory block, and will be swapped out independently if not used. The additional storage and time requirements mean that it has to be weighed carefully whether to compress executables which are typically run more than once at the same time.

Another disadvantage is that some utilities can no longer identify [run-time library](#) dependencies, as only the [statically linked](#) extractor stub is visible.

Also, some older [virus scanners](#) simply report all compressed executables as [viruses](#) because the decompressor stubs share some characteristics with those. Most modern virus scanners can unpack several different executable compression layers to check the actual executable inside, but some popular anti-virus and anti-malware scanners have had troubles with false positive alarms on compressed executables. In an attempt to solve the problem of [malware](#) obfuscated with the help of [runtime packers](#) the [IEEE Industry Connections Security Group](#) has introduced a [software taggant](#) system.

Executable compression used to be more popular when computers were limited to the storage capacity of [floppy disks](#), which were both slow and low capacity media, and small [hard drives](#); it allowed the computer to store more software in the same amount of space, without the inconvenience of having to manually unpack an archive file every time the user wanted to use the software. However, executable compression has become less popular because of increased storage capacity on computers. It has its use in the [demoscene](#) where demos have to stay within a size limit, e.g. [64k intro](#). Only very sophisticated compression formats, which add to load time, keep an executable small enough to enter these competitions.

List of executable packers

[\[edit\]](#)

CP/M and MSX-DOS executable

[\[edit\]](#)

Known executable compressors for [CP/M-80](#) / [MSX-DOS .COM files](#):

- PMexe (since 1990, written by Yoshihiko Mino, PMARC.COM+PMEXE.CPM, signature "-pms-")^[1]
- PopCom! (since 1992, written by Yoshihiko Mino, POPCOM.COM, signature "-pc1-")

Known executable compressors for [MS-DOS](#)-compatible executable files ([.COM](#) or [.EXE](#)):

- Realia Spacemaker (since 1982, written by [Robert B. K. Dewar](#), SM.COM, signature "MEMORY\$")^{[2][3][4][5][6][7][8][9]}
- [Microsoft](#) EXEPACK (since 1985, written by Reuben Borman, EXEPACK.EXE, LINK.EXE /E[XEPACK], signature "RB")^{[9][5][4][10]}
- LZEXE (since 1989, written by [Fabrice Bellard](#), LZEXE.EXE)^{[11][12]}
- [PKWare](#) PKLite (since 1990, written by [Phil Katz](#), PKLITE.EXE)^{[12][13]}
- DIET (since 1991, written by Teddy Matsumoto, DIET.EXE)^[12]
- TINYPROG (TINYPROG.EXE)
- RJS Software RJCUSH (since 1994, written by Roland Skinner, RJCUSH.EXE)
- XPA (since 1995, written by JauMing Tseng, XPA.EXE)
- Ibsen Software aPACK (since 1997, written by Jørgen Ibsen, APACK.EXE)
- [UPX](#) (since 1998, written by Markus F. X. J. Oberhumer and László Molnár)
- 32LiTE (since 1998, written by Oleg Prokhorov, 32LITE.EXE)
- Knowledge Dynamics LZW Compressor^[14]
- WWpack (since 1994, written by Piotr Warezak and Rafal Wierzbicki, WWPACK.EXE)
- 624 (only for .COM files smaller than 25 KB, uses [LZW](#))
- AINEXE
- AvPack
- ComPack
- HASP Envelope
- LGLZ
- PMWLITE
- ProPack
- UCExE
- WDOSX
- XE
- XPack

Known executable compressors under [OS/2](#):

- NeLite

- LxLite

Known executable compressors for [New Executables](#):

- PackWin
- PKWare PKLite (from version 2.01)
- WinLite

Portable Executable

[\[edit\]](#)

Known executable compressors for [Portable Executables](#):

Note: Clients in purple are no longer in development.

Name	Latest stable	Software license	x86-64 support
32Lite			
Alienyz	1.4 (17 August 2020)	Proprietary	No
ANDpakk2			
Armadillo	9.62 (7 June 2013)	Proprietary	Yes
ASPack	2.40 (7 December 2018)	Proprietary	Yes
ASPR (ASProtect)	2.78 (7 December 2018)	Proprietary	Yes
BeRoEXEPacker			
BIN-crypter			
BoxedApp Packer	3.3 (26 July 2015)	Proprietary	Yes
CExe	1.0b (20 July 2001)	GPL	No
Crinkler	2.3 (22 July 2020)	Zlib	Yes
dotBundle	1.3 (4 April 2013) ^[15]	Proprietary	Yes
Enigma Protector	6.60 (21 August 2019) ^[16]	Proprietary	Yes
Enigma Virtual Box	9.40 (10 October 2019) ^[16]	Proprietary	Yes
exe32pack			
EXE Bundle	3.11 (7 January 2011) ^[17]	Proprietary	?
EXECryptor			

Name	Latest stable	Software license	x86-64 support
EXE Stealth	4.14 (29 June 2011) ^[17]	Proprietary	?
eXPressor	1.8.0.1 (14 January 2010)	Proprietary	?
FSG	2.0 (24 May 2004) ^[18]	Freeware	No
kkrunchy src	0.23a4 (Unknown)	Public domain	No
MEW	1.1 (Unknown)	Freeware	No
MPRESS	2.19 (2 January 2012)	Freeware	Yes
MuCruncher			
NeoLite			
NsPack			
Obsidium	1.6 (11 April 2017) ^[19]	Proprietary	Yes
PECompact			
PEPack			
PESpin	1.33 (3 May 2011)	Freeware	Yes
Petite	2.4 (22 September 2016)	Freeware	No
PKLite32			
RLPack Basic	1.21 (31 October 2008)	GPL	No
Shrinker32			
Smart Packer Pro X	2.0.0.1 (3 June 2019)	Proprietary	Yes
Themida/WinLicense	3.2.5.0 (17 November 2025)	Proprietary	Yes
Upack			
UPX	5.0.1 (6 May 2025)	GPL	Yes
VMProtect	3.4 (3 August 2019)	Proprietary	Yes
WWPack32	1.20 (19 June 2000)		No
XComp/XPack	0.98 (18 February 2007)	Freeware	No
Yoda's Crypte			

Name	Latest stable	Software license	x86-64 support
YZPack			

Known executable compressors for [ELF](#) files:

- gzexe (uses a [shell script](#) stub and [gzip](#), works on most Unix-like systems)
- HASP Envelope
- [UPX](#)
- 624 (for Linux/386)^[20]

Known executable compressors for [CLI assembly](#) files:

- .NETZ
- NsPack
- Mpress
- HASP Envelope
- dotBundle
- Exepack.NET
- DotProtect:^[21] Commercial protector/packer for .net and mono. Features on-line verifications and "industry standard encryption".

Mac OS Classic applications

[\[edit\]](#)

Executable compressors for [Mac OS Classic](#) applications:

- Application VISE^[22]
- StuffIt InstallerMaker

Mach-O (Apple Mac OS X) files

[\[edit\]](#)

Known executable compressors for [Mach-O](#) (Apple Mac OS X) files:

- HASP Envelope
- [UPX](#)
- VMProtect

Commodore 64 and VIC-20

[\[edit\]](#)

Known executable compressors for executables on the [Commodore 64](#) and [VIC-20](#):

- PuCrunch^[23]
- Exomizer^[24]
- ByteBoozer^[25]

Known executable compressors for executables on the [Amiga](#) series:

- powerpacker^[26]
- Titanics cruncher^[26]
- imploder^[26]
- TNM cruncher
- Shrinkler^[27]
- PackFire^[28]

Known executable compressors for [Java](#):

[JAR](#) files:

- HASP Envelope
- pack200
- [ProGuard](#)

[WAR](#) files:

- HASP Envelope

There are two types of compression that can be applied to [JavaScript](#) scripts:

- Reduce the redundancy in the script (by removing comments, white space and shorten variable and functions names). This does not alter the behavior of the script.
- Compress the original script and create a new script that contains decompression code and compressed data. This is similar to binary executable compression.

Self-decompressing compressors

[\[edit\]](#)

These compress the original script and output a new script that has a decompressor and compressed data.

- JsSfx
- Packify

Redundancy reducing compressors

[\[edit\]](#)

These remove white space, remove comments, and shorten variable and function names but do not alter the behavior of the script.

- Packer
- [YUI compressor](#)^{[29][30]}
- [Shrinksafe](#)^[30]
- [JSMin](#)^[30]

- [Disk compression](#)
- [RAM compression](#)
- [Kolmogorov complexity](#)
- [Self-modifying code](#)
- [Self-relocating code](#)
- [Self-extracting archive](#)
- [Self-booting disk](#)

1. [^] [Gielen, Pierre; Taylor, Johnathan \(1997\) \[1993\]. Logan, Wolverine \(ed.\). "PMarc help manual". Retrieved 2019-02-22.](#) “[...] PMEXE.CPM [...] is a module [...] in combination with PMARC [...] used to make executable compressed COM files (just like LZEXE or PKLITE [...] type: PMARC <archive>.COM=PMEXE2.CPM <filename> [options] The archive-name must be .COM [...] not .PMA. The output file will have the extension .CPM. It's an [MSX-DOS](#) COM file [...] rename file [...] to run it [...]” {{cite web}} : CS1 maint: deprecated archival service ([link](#))
2. [^] ["Expert Report of Robert B. K. Dewar In Response To The Report Of Kenneth D. Crews". Cambridge University Press et al v. Patton et al, Filing 124, Supplemental Initial Disclosures by Cambridge University Press, Oxford University Press, Inc., Sage Publications, Inc. - Cambridge University Press, Oxford University Press, Inc., and Sage Publications, Inc. v. Mark P. Becker, Georgia State University President, et al, Civil Action No. 1:08-CV-1425-ODE \(Court document\). United States District Court For The Northern District Of Georgia, Atlanta Division. p. 18. Exhibit A. Archived from the original on 2018-05-01. Retrieved 2019-04-23.](#) “[...] SPACEMAKER and TERMULATOR, commodity software for IBM PC ([PC DOS](#) file compression utility and [VT-100](#) emulator), being marketed by Realia, Inc. [R.B.K. Dewar](#) (1982–1983), 8088 assembly language, 8,000 lines [...]”
3. [^] [Realia, Inc. \(January 1983\). "If you use DOS, you need this program". PC Magazine \(advertisement\). 2 \(9\). Ziff-Davis Publishing; 417. Retrieved 2019-04-22.](#) {{cite journal}} : CS1 maint: deprecated archival service ([link](#))
4. [^] [Jump up to: ^a ^b Dewar, Robert Berriedale Keith \(1984-03-13\). "DOS 3.1 ASMB \(Another Silly Microsoft Bug\)". info-ibmpc@USC-ISIB.ARPA. Archived from the original on 2018-05-01. Retrieved 2019-04-23.](#) “[...] The /E option of the linker should generate an EXE file which is logically equivalent to the uncompressed EXE file. The current version [...] results in AX being clobbered. AX on entry to an EXE file has a definite meaning (it indicates drive validity for the parameters), thus it should be passed through to the uncompressed image. Given this one very obvious violation of the interface rules, there may be others, I have not bothered to investigate further [...] I did write the Realia SpaceMaker program which does a similar sort of thing to the EXEPACK option (but needless to say does not have this particular [...]”

5. ^ [Jump up to: ^a ^b](#) Paul, Matthias R. (2002-10-07) [2000]. ["Re: masm .com \(PSP\) related trouble"](#). [Newsgroup: alt.lang.asm](#). Retrieved 2017-09-03. {{cite newsgroup}} : CS1 maint: deprecated archival service ([link](#))
6. ^ Necasek, Michal (2018-04-30). ["Realia SpaceMaker"](#). OS/2 Museum. [Archived](#) from the original on 2019-01-27. Retrieved 2019-02-22.
7. ^ Parsons, Jeff (2019-01-10). ["An Update on Early Norton Utilities"](#). PCjs. [Archived](#) from the original on 2019-01-29. Retrieved 2019-02-22.
8. ^ Necasek, Michal (2019-01-12). ["Yep, Norton Did It"](#). OS/2 Museum. [Archived](#) from the original on 2019-04-22. Retrieved 2019-04-22.
9. ^ [Jump up to: ^a ^b](#) Necasek, Michal (2018-03-23). ["EXEPACK and the A20-Gate"](#). OS/2 Museum. [Archived](#) from the original on 2018-11-13. Retrieved 2019-04-20.
10. ^ Miles, Ya'akov; Nather, Ed (1986-05-17) [1986-02-05, 1986-02-09]. ["Undocumented Microsoft LINK option: /E"](#). INFO-IBMPc mailing list. [Archived](#) from the original on 2018-05-01. Retrieved 2019-04-26.

"[Miles:] There exists an undocumented [...] switch to [Microsoft](#) LINK.EXE [...], which will cause an automatic compaction during binding. This process will eliminate storage for uninitialized arrays from the .EXE file produced by the linker [...] To use this feature, specify the /E option to the command line [...]

[Nather:] The option does not exist in MS Link versions 3.00 and 3.01 [Miles:] By comparing the sizes of the (packed) files generated from LINK ver 3.02 and the /E option with the size of the .EXE file manually packed with [...] EXEPACK, I have come to the conclusion that LINK ver 3.02 option /E generates EXACTLY the same size file as manually running EXEPACK on a regular .EXE file output by LINK [...]"
11. ^ [Bellard, Fabrice](#) (2003-02-09). ["LZEXE home page"](#). bellard.org. [Archived](#) from the original on 2019-03-24. Retrieved 2019-03-18.
12. ^ [Jump up to: ^a ^b ^c](#) Salomon, David (2000) [1998]. ["Chapter 3.22: EXE Compressors"](#). *Data Compression: The Complete Reference* (2 ed.). [Springer-Verlag](#). p. 212. [doi:10.1007/978-3-642-86092-8](#). [ISBN 978-3-540-78086-1](#). [S2CID 35889155](#). [Archived](#) from the original on 2022-04-07. Retrieved 2019-04-26.
13. ^ Paul, Matthias R. (2002-04-11). ["Re: \[fd-dev\] ANNOUNCE: CuteMouse 2.0 alpha 1"](#). freedos-dev. [Archived](#) from the original on 2020-02-21. Retrieved 2020-02-21.

"[...] > no one packer may pack combos like .SYS+.COM or .SYS+.EXE. [...] There are packers for .COM or .EXE and others for .SYS, but I too have not seen a packer which supports [both in one](#). [...] possibility to [combine a program/TSR and device driver](#) in .EXE files [...] and a program/TSR.COM and device driver into a .COM program [...] It might also be possible to add another self-made stub to the file, after it has already been compressed [...] all the compressed [DR-DOS](#) device drivers use a similar technique to let the normal PKLITE .COM decompressor work with .SYS files (meanwhile PKLITE supports a similar feature for .SYS files itself). [...]" (NB. PKLITE 1.50 (1995) and higher gained the capability to compress device drivers, but not combined COM+SYS drivers.)
14. ^ ["Google Code Archive - Long-term storage for Google Code Project Hosting"](#). [Archived](#) from the original on 2021-12-03. Retrieved 2022-04-07.
15. ^ ["DotBundle - Download an evaluation version"](#). [Archived](#) from [the original](#) on 2013-08-21. Retrieved 2013-05-06.
16. ^ [Jump up to: ^a ^b](#) ["Software Protection, Software Licensing, Software Virtualization"](#). [Archived](#) from the original on 2018-03-13. Retrieved 2018-03-12.

17. ^ [Jump up to: ^a ^b](#) "[WebtoolMaster Software News](#)". *Archived* from the original on 2020-07-20. Retrieved 2012-01-15.
18. ^ "[Archived copy](#)". www.xtreeme.prv.pl. Archived from [the original](#) on 2004-05-25. Retrieved 2022-01-15.

{{cite web}} : CS1 maint: archived copy as title (link)
19. ^ "[Download | Obsidium Software Protection System](#)". *Archived* from the original on 2019-02-07. Retrieved 2017-04-18.
20. ^ "[624](#)".
21. ^ DotProtect http://site.yvansoftware.be/dotpacker1_0 Archived 22 January 2011 at the [Wayback Machine](#)
22. ^ Kiene, Steve; Mark, Dave (1999). "[A Chat With Steve Kiene](#)". *MacTech*. Vol. 15, no. 4. *Archived* from the original on 2017-12-10. Retrieved 2017-12-10.
23. ^ "[Lossless Data Compression Program: Hybrid LZ77 RLE](#)". www.cs.tut.fi. Archived from [the original](#) on 2014-07-30. Retrieved 2022-01-15.
24. ^ web.comhem.se/~u13114991/exo/
25. ^ "[ByteBoozer \(PC\)](#)". *Archived* from the original on 2022-08-15. Retrieved 2022-04-07.
26. ^ [Jump up to: ^a ^b ^c](#) "[Crunchers to download](#)".
27. ^ "[Askeksa/Shrinkler](#)". [GitHub](#). 2021-09-25. *Archived* from the original on 2022-04-07. Retrieved 2022-04-07.
28. ^ "[PackFire v1.2k by Neural](#)". *Archived* from the original on 2024-06-30. Retrieved 2022-04-07.
29. ^ Nizamettin Gok and Nitin Khanna. "[Building Hybrid Android Apps with Java and JavaScript](#)". Publisher: O'Reilly Media. 2013. p. 32. quote: "YUI Compressor is ... is one of the most popular JavaScript minifier tools"
30. ^ [Jump up to: ^a ^b ^c](#) Nicholas C. Zakas "[Professional JavaScript for Web Developers](#)". 2011. p. 768-769. quote: "Crunchers: An important part of the JavaScript build process ... JSMIn ... Dojo ShrinkSafe ... and ... YUI compressor ..."

Source: http://en.wikipedia.org/wiki/Executable_compression