Incremental Journaling Backup Utility and Archiver

zpaq is a free and open source incremental, journaling command-line archiver for Windows, Linux and Mac OS/X. *Incremental* means that when you back up your hard drive, for example:

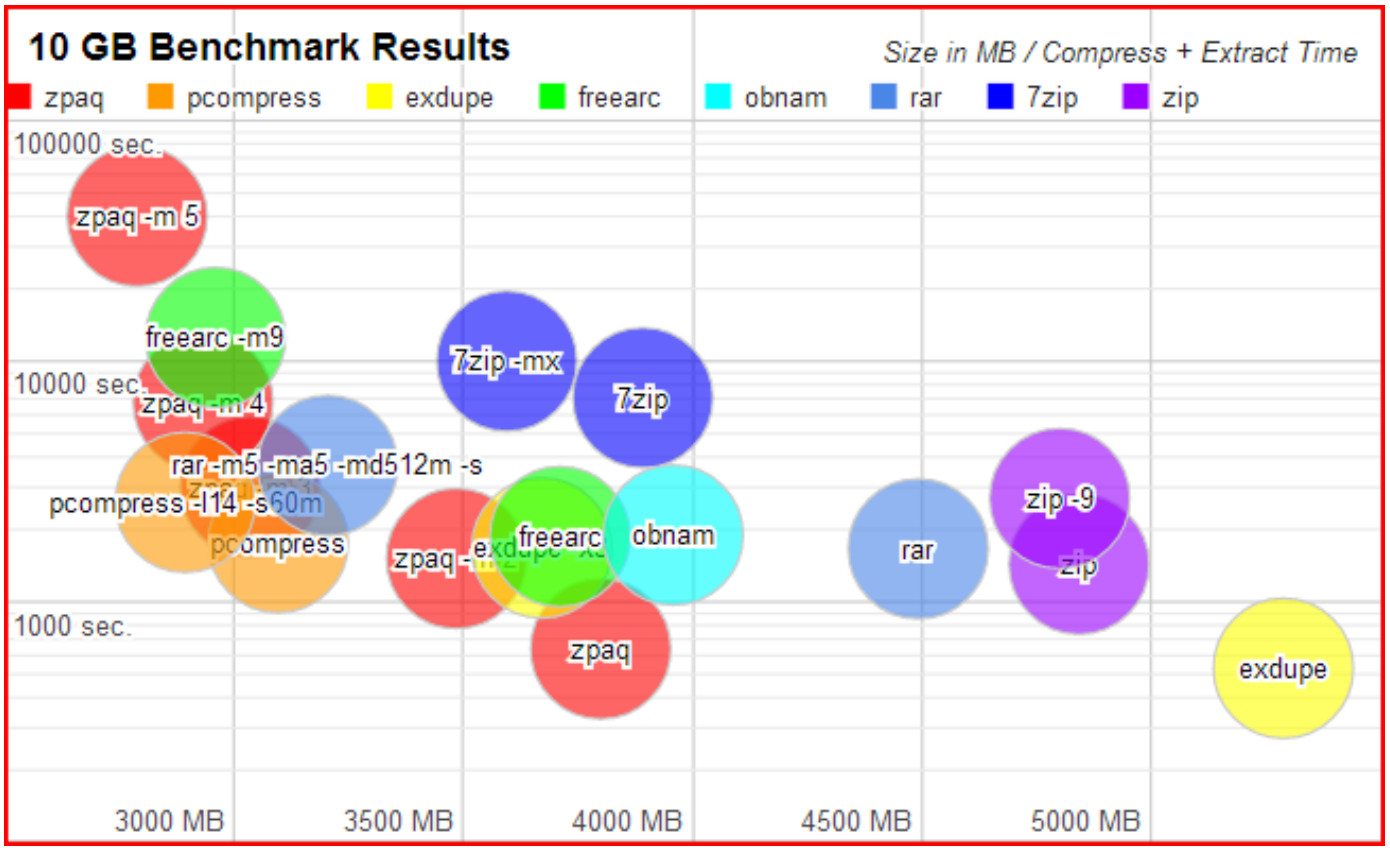
```
zpaq add e:\backup.zpaq c:\*
```

then only those files whose last-modified date or size has changed since the previous backup are added. For 100 GB of files, this typically takes about a minute, vs. an hour to create the first version. *Journaling* means that the archive is append-only. When you add files or directories to the archive, both the old and new versions are saved. You can recover old versions by specifying the date or version number, for example:

```
zpaq extract e:\backup.zpaq c:\Users\Bob -to tmp -until 2013-10-30
```

will extract all the files and directories in c:\Users\Bob as of the last backup on or before Oct. 30, 2013 and put them in a directory named tmp.

zpaq is faster and compresses better than most other popular archivers and backup programs, especially for realistic backups that have a lot of duplicate files and a lot of already compressed files.



Archive size vs. time to compress and extract 10 GB (79,431 files) to an external USB hard drive at default and maximum settings on a Dell Latitude E6510 laptop (Core i7 M620, 2+2 hyperthreads, 2.66 GHz, 4 GB, Ubuntu Linux, Wine 1.6). Data from 10 GB Benchmark (system 4).

Feature comparison

	zpaq	pcompress	exdupe	freearc	obnam	rar	7zip	zip
Windows	W		W	W		W	W	W
Linux	L	L	L	L	L	L	L	L
Update	U		U	U	U	U	U	U
Incremental	I		I		I	I		I
Rollback	R				R			
Dedupe	D	D	D		D			
Encryption	E	E		E		E	E	E
GUI				G		G	G	G
Free	F	F	F	F	F		F	F
Open source	O	O	O	O	O		O	O
Specification	S							S

Download

zpaq.exe for Windows.

The latest version is zpaq v7.15, released Aug. 17, 2016. The download contain source code (zpaq.cpp, libzpaq.cpp, libzpaq.h), Windows executables (32 or 64 bit, XP or later), documentation (zpaq.pod), and a Makefile for compiling in Linux, BSD, or Mac OS/X. You may need unzip.exe to unzip from the Windows command line.

zpaq man page (HTML, latest version).

The ZPAQ archive format is described by a specification and reference decoder. A test case exercising all of the specification features should decompress to the Calgary corpus. The compression algorithm is described here.

The source code includes the libzpaq API providing compression and decompression services for applications in C++. Developers may be interested in the zpaqd development tool and sample configuration files found on the utilities page.

zpaq is written by Matt Mahoney and released to the public domain. It includes code from libdivsufsort 2.0 (C) Yuta Mori, 2003-2008, MIT license, public domain code for AES from libtomcrypt by Tom St Denis and public domain code for salsa20 by D. J. Bernstein.

Features

A zpaq archive can contain at most 4 billion files and at most 250 terabytes of data after deduplication and before compression.

zpaq is for user-level backups. Do not use it to back up the operating system or any software that requires a password to install. zpaq saves regular files and directories, last-modified dates (to the nearest second), and (optionally) Windows attributes or Linux permissions. It does not follow or save symbolic links or junctions. It unknowingly follows hard links. It does not save owner or group IDs, ACLs, extended attributes, the registry, or special file types like devices, sockets, or named pipes.

Open standard specification

The zpaq archive format is described by a precise specification and reference decoder (above). The format is not encumbered by any patents or pending patents in any country as far as I know. I have purposely published all past versions (below) to establish prior art so that no patents can be filed.

Backward and forward compatibility

All versions of zpaq can read archives produced by older versions back to version 1.00 (March 2009). To some extent, older versions can read archives produced by newer versions (forward compatibility) provided they don't use any unsupported features. These are as follows:

- v1.00 (Mar. 2009). Level 1 format. Streaming archives with at least one context model. Does not support deduplication or rollback.
- v5.00 (Aug. 2012). Level 2 format. Adds support for compression with pre/post processing with no context modeling (e.g. uncompressed or LZ77).
- v6.00 (Sept. 2012). Journaling format (dedupe and rollback).
- v6.44 (Jan. 2014). Encrypted archives.
- v6.47 (Jan. 2014). Multi-part archives. Older versions can read them if concatenated.

Many intermediate versions include compression improvements. This does not break forward compatibility because the decompression code is stored in the archive. The code is written in a sandboxed, virtual machine language called ZPAQL. On x86-32 and x86-64 processors, the ZPAQL code is translated to machine code and executed, so it is as fast as compression algorithms written in compiled languages like C or C++. On other hardware, the ZPAQL code is interpreted, which takes about twice as long.

For example, the following will create a streaming archive using BWT compression that can be extracted by all versions back to v1.00, even though most of these versions could not compress using BWT.

```
zpaq add archive.zpaq files -method s4.3ci1
```

Rollback

An archive is updated only by appending changes to it. You can roll back the archive to an earlier state by using the `-until` option to specify the date and time or version number where to stop reading.

When updating, `-until` will truncate the archive at that point before appending. So if you backed up some files you didn't mean to, then you can truncate the last update and repeat:

```
zpaq add backup c:\ -not c:\tmp -until -1
```

Transacted updates

Updates are committed by first appending a temporary header and then updating it when all of the compressed data and index changes are appended. If you interrupt zpaq (by typing Ctrl-C), then the partially appended data will be ignored and overwritten on the next update.

Deduplication

When adding files, zpaq uses a rolling hash function to split files into fragments with an average size of 64 KB along content-dependent boundaries. Then it computes the SHA-1 hash of the fragment and compares it with saved hashes from the current and previous versions. If it finds a match then the fragment is not stored.

Deduplication requires 1 MB of memory per GB of deduplicated but uncompressed archive data to update, and 0.5 MB per GB to list or extract.

Incremental update and restore

Files are added only if the date has changed since the last update. You can use the -force option to override, but in this case the file will be deduplicated and not saved unless the contents have really changed. This is slower than comparing dates but faster than compressing it again.

Extraction will not clobber existing files unless you give the -force option to allow overwrite. In this case, the file to be overwritten is compared with the stored hashes and not decompressed unless the size or contents is different.

Remote archive support

zpaq updates an archive by appending changes to it. To support remote backups without having to move huge files, zpaq can put the appended changes into a separate, numbered file that you would copy or move to remote storage. You can concatenate the parts to form a complete archive, or simply read them all at once by specifying a pattern in the archive name like "part???.zpaq". zpaq will then search for part001.zpaq, part002.zpaq, etc. and regard the concatenated sequence as a single archive.

To make incremental backups with a local copy:

```
zpaq add "arc???" files
(copy arc001.zpaq)
zpaq add "arc???" files
(copy arc002.zpaq)
zpaq list "arc???"        (show contents)
zpaq extract "arc???"     (restore)
```

To back up without keeping a local copy of the archive, you keep a small local index (arc000.zpaq) as a copy of the remote archive minus the compressed file contents. zpaq maintains consistency between the index and archive.

```
zpaq add "arc???" files -index arc000.zpaq
(move arc001.zpaq)
zpaq add "arc???" files -index arc000.zpaq
(move arc002.zpaq)
zpaq list arc000          (show arc???.zpaq contents)
```

Encryption

Archives can be encrypted using AES-256 in CTR mode. A password must be given every time an encrypted archive is used. Keys are strengthened with Scrypt(N=16384, r=8, p=1) (requiring 208M

operations and 16 MB memory) to slow down brute force search for weak keys. Encrypted archives are prefixed with a 32 byte random salt, which also provides an 8 byte IV for the first half of the 16 byte AES counter. If a remote archive has a local index, then both are encrypted with the same key but different salts to generate independent keystreams. Encryption provides privacy but not authentication against tampering.

All of the encryption code (AES, Scrypt, SHA-1, SHA-256) is public domain and tested against published test vectors. The AES code is derived from libtomcrypt 1.17.

Multithreaded compression

zpaq has 5 compression levels. The default, -method 1, is the fastest. It is best for backups where you compress often and extract rarely. -method 2 compresses slower but decompresses as fast as -method 1. It is best for distributing files where you compress once and extract often. Methods 3, 4, and 5 are slower with better compression.

Fragments not removed by deduplication are packed into blocks for compression. Files are sorted by filename extension and then by decreasing size in order to group similar files together. The block size is 16 MB for method 1 and 64 MB for higher methods. You can change the block size to trade compression for memory usage.

Blocks are compressed or decompressed in parallel in separate threads. zpaq automatically detects the number of processor cores and uses all of them in the 64 bit version or at most 2 in the 32 bit version (which is limited to 2 GB memory). You can use the -threads option to change the number of threads. Resident memory per thread required to compress or decompress is approximately as follows. Virtual memory usage may be higher.

Method	Compress	Decompress	Algorithm
1	128 MB	32 MB	LZ77
2	450 MB	128 MB	LZ77
3	450 MB	400 MB	LZ77+CM or BWT
4	550 MB	550 MB	LZ77+CM, BWT or CM
5	850 MB	850 MB	CM

Method 1 uses LZ77, compressing by replacing duplicate strings with pointers to previous occurrences. Method 2 is the same but spends more time looking for better matches (using a suffix array instead of a hash table). Method 3 uses either BWT (context sorting) or LZ77 for long matches and an order 1 context model and arithmetic coding for literals depending on the file type. Method 4 either uses LZ77, BWT or a high order context model. Method 5 uses a complex, high order context mixing model with over 20 bit prediction components.

All methods except 5 test whether the data appears to be compressible or already compressed (random). Uncompressible data is simply stored.

An E8E9 filter is applied if x86 data (normally found in .exe and .dll files) is detected. The filter replaces x86 CALL and JMP relative addresses with absolute addresses to make the data more compressible.

Data analysis

zpaq has list options to make it easier to examine the contents of archives containing millions of files. For example, the following compares external dir1 to internal dir2 and lists only differences. Files are compared quickly by size and last modified date, or thoroughly by reading the file, computing its SHA-1 hashes and comparing with the hashes stored in the archive.

```
zpaq list backup dir1 -to dir2 -not =             (compare dates)
zpaq list backup dir1 -to dir2 -not = -force      (compare contents)
```

Other useful list options:

```
-only *.exe    List only files ending with .exe
-not *.exe     Don't list files matching a pattern.
-summary 20    List the 20 largest files and identify duplicates.
-all           Show all file versions.
-until 20      List contents as of the 20'th update
```

Error detection and recovery

zpaq archives are designed to minimize data loss if damaged. An archive is divided into blocks that can be decompressed independently. Each block begins with a 13 byte tag that can be found by scanning if the previous block is damaged. Each block ends with the SHA-1 hash of the uncompressed data, which is verified to detect errors. Blocks with hash mismatches or other errors are ignored with a warning without killing zpaq.

Each update contains 4 types of blocks.

- C - Update header: date, size of compressed data.
- D - Compressed data fragments, list of fragment sizes.
- H - List of fragment hashes and sizes, one per D block.
- I - Index updates: list of files updated or deleted. Each update includes the date, attributes, and list of fragments.

C blocks are used to skip over D blocks to read the index quickly. They are not needed to extract. If a D or H block is lost then so are any files that point to it. If an I block is lost, then so are any files in it. I blocks are small (16 KB) to minimize damage.

When extracting files, the D block is decompressed up to the last used fragment and those fragments are hashed and compared to the stored hashes in the H block.

The zpaq `-test -all` extract option will decompress internally and verify all of the fragment hashes without writing the output files.

Public Domain API

The source download includes *libzpaq*, a public domain application programming interface (API) in C++ that provides streaming compression and decompression services to and from files, strings, or arrays using built-in and custom compression algorithms. To use the code, you include `libzpaq.h` in your program and link to `libzpaq.cpp`. The API documentation is in `libzpaq.h`. The precise semantics is described in the ZPAQ specification.

In the simplest case, the application provides an error handling function and derived implementations of two abstract classes, Reader and Writer, specifying the input and output byte

streams. For example, to compress from stdin to stdout (assuming binary I/O as in Linux):

```
#include "libzpaq.h"
#include <stdio.h>
#include <stdlib.h>

void libzpaq::error(const char* msg) {  // print message and exit
  fprintf(stderr, "Oops: %s\n", msg);
  exit(1);
}

class In: public libzpaq::Reader {
public:
  int get() {return getchar();}  // returns byte 0..255 or -1 at EOF
} in;

class Out: public libzpaq::Writer {
public:
  void put(int c) {putchar(c);}  // writes 1 byte 0..255
} out;

int main() {
  libzpaq::compress(&in, &out, "1");  // -method 1
}
```

To decompress:

```
libzpaq::decompress(&in, &out);
```

There are also functions for reading and writing block and segment headers and for passing specialized methods or ZPAQL code to the compressor, as documented in libzpaq.h. The ZPAQ utilities page contains sample compression algorithms written in ZPAQL and a tool *zpaqd* for running, testing, and debugging ZPAQL.

History

All versions of the software and documentation can be downloaded below. The major development steps were:

- Feb. 15, 2009: zpaq 0.01, First of 9 experimental, mutually incompatible versions.
- Mar. 12, 2009: zpaq 1.00. First level 1 standard conforming archiver using interpreted ZPAQL for forward and backward compatibility.
- Sept. 29, 2010: libzpaq 1.00. First version of API providing compression services to applications in C++.
- Nov. 5, 2010: libzpaq 2.01. If an external C++ compiler is available then zpaq will translate ZPAQL to C++ and recompile itself to improve speed.
- Jan. 26, 2011: pzpaq 0.01. First multi-threaded version (later renamed zp, then merged back into zpaq).
- Nov. 13, 2011: libzpaq/zpaq 4.00. First version with JIT-accelerated ZPAQL for x86/64, eliminating need for external C++ compiler.
- Feb. 1, 2012: libzpaq 5.00. Level 2 standard allowed high speed compression without a context model (pre/post processing only).

- Sept. 26, 2012: zpaq 6.00. Journaling format to support deduplication, fast indexing, update recovery, and storing multiple versions of files and directories.
- June 11, 2013: zpaq 6.27. Moved developer tools into zpaqd.
- Dec. 20, 2013: zpaq 6.43. Adds AES encryption.
- Nov. 22, 2014: zpaq 6.56. Supports remote multi-part archives with a local index.

zpaq versions 7.00 and older are licensed under GPL v3. The SHA-1 code used in versions prior to libzpaq 1.00 is derived from RFC 3174, which is copyright (C) 2001, The Internet Society. Please see this document for the full license.

zpaq v0.01 Open source (C++) and Win32 executables, Feb. 15, 2009.
zpaq v0.02 adds E8E9 transform. Fully supports post-processing. Not compatible with v0.01. Feb. 19, 2009.
zpaq v0.03 modifies MIX, MIX2, IMIX to fix poor compression on large files. Not compatible with v0.02. Feb. 19, 2009.
zpaq v0.04 modifies train() and squash() for improved compression. Not compatible with v0.03. Feb. 21, 2009.
zpaq v0.05 modifies probability representation and mixer weights to prevent mixer overflow and to improve compression for highly redundant data. Not compatible with v0.04. Feb. 26, 2009.
zpaq v0.06 adds SHA1 checksums, replaces IMIX2 with ISSE. Not compatible with v0.05. Feb. 27, 2009.
zpaq v0.07 improves ISSE and bit-history state table. Not compatible with v0.06. Feb. 28, 2009.
zpaq v0.08 adds LZP transform and minor improvements. Not compatible with v0.07. Mar. 8, 2009.
zpaq v0.09 removes counters from ISSE and ICM to improve speed. Not compatible with v0.09. Mar. 9, 2009.
zpaq v1.00 (first level 1 compliant version) includes unzpaq1 candidate reference decoder. Simplified bit history tables. Not compatible with earlier versions. Mar. 12, 2009.
fast.cfg written Apr. 26, 2010. Now part of libzpaq distribution.
unzpaq 1.01 updates reference decoder comments and help message and fixes some VS2005 compiler issues. Compatible with 1.00. Apr. 27, 2009.
unzpaq 1.02 and zpaq 1.02 closes extracted files immediately after decompression instead of when program exits. Fixes g++ 4.4 warnings. Compatible with 1.00 and 1.01. June 14, 2009.
unzpaq 1.03 and zpaq 1.03 has a default compression mode (mid.cfg), supports compressing files in segments to separate blocks and extracting them as suggested in part 7 of the spec. Does not store paths by default. Does not extract to absolute paths by default. Some minor improvements. Sept. 7, 2009 (added zpaq.exe Sept. 8, 2009).
zpaq103b adds zpaqsfx 1.03, a stub for creating self extracting archives. No changes to zpaq or unzpaq. Sept. 14, 2009.
zpaq104 can list and extract from self extracting archives without running them. Added progress meter. zpaqsfx.exe stub is slightly smaller. unzpaq unchanged. Sept. 18, 2009.
zpaq105 removes built in x and p preprocessors and makes them separate programs called from config files with compile time postprocessor testing. Adds if-else-endif and do-while to ZPAQL. Many small changes. Sept. 28, 2009.
zpaq106 adds "ta" to append locater tags to allow ZPAQ streams to be detected when embedded in arbitrary data. zpaq1.pdf revision 1 adds this recommendation. unzpaq106.cpp implements it. Sept. 29, 2009.
zpaqsfx 1.06 self extracting archive stub is now separate from the ZPAQ distribution. Updated Sept. 29, 2009, posted Oct. 26, 2009. (Replaced by zpsfx in libzpaq 2.01)
zpipe v1.00, a simple streaming compressor, Sept. 30, 2009. Linux patch added Jan. 18, 2010.

zpaq107 adds config file parameters and fixes some bugs. From now on the specification and reference decoder are not included unless they change. Oct. 2, 2009.
bwt_j2 is a config file (by Jan Ondrus) and preprocessor for BWT compression. Posted Oct. 7, 2009.
bwt_j3 is a bug fix for bwt_j2 to accept multiple files. Jan Ondrus, Oct 7, 2009.
exe_j1 is a config file and preprocessor for .exe and .dll files. It extends the E8E9 transform in exe.cfg to conditional jumps. Jan Ondrus, Oct. 7, 2009.
unzpaq108.cpp removes undefined behavior of ZPAQL shifts larger than 31 bits on non x86 hardware. Oct. 14, 2009.
zpaq108 generates optimized code that runs about twice as fast on systems with a C++ compiler installed. Oct. 14, 2009.
bmp_j4, configuration for .bmp files by Jan Ondrus, Oct. 14, 2009.
bwt_slowmode1 BWT compression based on BBB slow mode. Jan Ondrus, Oct. 15, 2009.
jpg_test2 JPEG config by Jan Ondrus, Oct. 20, 2009, posted Oct. 26, 2009.
zpaq109 Linux port and some cosmetic bug fixes. Oct. 21, 2009.
zpaq110 bug fix for Linux/g++ 4.4.1, Dec. 28, 2009.
zp v1.00 simple ZPAQ compatible archiver with 3 optimized compression levels, Apr. 26, 2010.
Added a license file to zpaq 1.10, zpipe 1.00, zpaqsfx 1.06, and zp 1.00 distributions on May 23, 2010. No software changes.
libzpaq 0.01, Sept. 27, 2010.
libzpaq 0.02, Sept. 28, 2010.
zpipe 2.00 updated to use libzpaq 0.02, Sept. 28, 2010.
libzpaq 1.00, Sept. 29, 2010. Package includes libzpaq, ZPAQ specification, reference decoder, zp, zpipe, and fast, mid, max config files.
libzpaq 1.01, Oct. 14, 2010. Updates libzpaq interface to use inheritance instead of templates, requiring changes to zp and zpipe. Now compiles faster.
libzpaq 1.02, Oct. 20, 2010. Adds zpsfx self extracting archive stub. Separates optimized models from libzpaq.cpp to libzpaqo.cpp.
libzpaq 2.00, Oct. 30, 2010. Ports zpaq to libzpaq, replacing zp.
libzpaq 2.01, Nov. 5, 2010. Added optimized self extracting archives. Simplified installation.
libzpaq 2.02, Nov. 13, 2010. zpaq shows compression component statistics. Libzpaq support added.
zpaq 2.03, Dec. 23, 2010, adds Linux support. The remaining code is split into libzpaq 2.02, zpipe 2.01, zpsfx 1.00, and configuration files min, fast, mid, and max.
zpaq 2.04, Dec.29, 2010, adds support for Visual C++, Borland, and Mars compilers in addition to g++. A Windows install script is added.
zpaq 2.05, Jan. 5, 2011. Fixed a bug in which zpaq crashed when decompressing an unnamed file (as created with zpipe or zpaq nc) without renaming. Separated zpaq.1.pod. (Updated corrupted install.sh on Jan. 13, 2011).
libzpaq 2.02a, Jan. 6, 2011. Updates the documentation.
pzpaq 0.01 parallel file compressor, Jan. 21, 2011.
pzpaq 0.02, Jan. 26, 2011, adds large file support (over 2 GB) to Windows.
pzpaq 0.03, Feb. 2, 2011, optimizes decompression for nonstandard compression levels by recompiling itself with g++ (like "zpaq ox").
pzpaq 0.04, Feb. 4, 2011, Windows version uses native threads and no longer requires pthreads-win32.
pzpaq 0.05, Feb. 10, 2011 removes -s option, puts temporary files in $TMPDIR or %TEMP%.
bwt v1, Mar. 16, 2011. 4 BWT based configurations.
unzp 1.00, May 10, 2011, a block level parallel decompresser optimized for fast, mid, max,

bwtrle1, bwt2 models with source level JIT for other models.
zp 1.01, May 12, 2011, a block level parallel compressor with 4 levels (bwtrle1, bwt2, mid, max). With unzp replaces pzpaq.
zp 1.02, May 16, 2011. Fixed -t option.
May 18, 2011. Undated zp.102.zip and unzp.100.zip with static x86-64 Linux binaries.
zp 1.03, May 26, 2011. Merges the compressor and decompresser unzp into one program.
wbpe 1.00, June 12, 2011. Dictionary preprocessor for text files.
wbpe 1.10, June 21, 2011.
zpaq 3.00, July 16, 2011. Combines features of zpaq v2.05 and zp v1.03. zp support is discontinued. Windows only.
zpaq 3.01, July 21, 2011. Adds 64 bit Linux support. Includes libzpaq 3.00.
bmp_j4a, July 21, 2011. Updated bmp_j4 .bmp configuration for zpaq v3.01.
libzpaq 3.00, July 28, 2011, from zpaq v3.01 but as a separate download.
libzpaq 4.00, Nov. 13, 2011. libzpaq.cpp, libzpaq.h, libzpaq.3.pod. Replaces source-level JIT with internal JIT for x86-32 and x86-64.
zpaq 4.00, Nov. 13, 2011. zpaq.cpp, zpaq.1.pod for use with libzpaq 4.00. Removes source generation, b and e commands and -j option.
calgarytest.zpaq, Nov. 13, 2011. Test case for ZPAQ compliance.
zpipe v2.01, Nov. 13, 2011. zpipe.exe linked to libzpaq v4.00. Source unchanged.
zpaq v4.01, Nov. 26, 2011. Source code adds incremental update and extraction.
zpaq v4.02, Nov. 28, 2011. Source code adds commands c, x output/, list hcomp/pcomp. Updated pi.cfg for this version.
libzpaq v4.01, Dec. 20, 2011. Fix for Mac OS (MAP_ANONYMOUS -> MAP_ANON).
zpaq v4.03, Dec. 21, 2011. Adds -n, -r, and -f options. Fixed bug in u (did not save filenames with no args).
lz1.zip, Dec. 29, 2011. LZ77 model.
ZPAQ level 2 standard, unzpaq200.cpp reference decoder, libzpaq 5.00 support, and calgarytest2.zpaq test case, Feb. 1, 2012. Level 2 allows the COMP section to be empty to store uncompressed (but possibly preprocessed) data to support faster compression models.
libzpaq 5.01, Feb. 2, 2012. Removed debugging code from libzpaq.cpp.
tiny_unzpaq.cpp v1.0, Mar. 21, 2012.
zpaq v4.04, Mar. 26, 2012. Fixed bug in r command that truncated output file.
zpsfx v1.01, Apr. 4, 2012. Self extractor modified by Klaus Post to create directories as needed.
zpaq v5.00, Aug. 27, 2012. Candidate release, primarily a small development tool. Updates libzpaq to v6.00a to include ZPAQL compiler.
zpaq v6.00, Sept.26, 2012. Candidate release. Adds journaling, incremental update, and deduplication to support large backups. Includes 4 compression levels (fast and slow LZ77, BWT, mid) plus all v5.00 features.
zpaq201.pdf, Sept. 28, 2012. Updated specification to describe streaming and journaling archive formats.
zpaq v6.01, Sept. 28, 2012. Adds -method 0, -list -force, improves -list -detailed, and bug fixes.
zpaq v6.02, Sept. 29, 2012. Speed and compression improvements. Adds -quiet option.
zpaq v6.03, Sept. 30, 2012. Saves and restores file attributes. Cleans up -list.
zpaq v6.04, Oct. 1, 2012. Compression and speed improvements by sorting by filename extension and storing uncompressible data. Adds -list -quiet.
bmp_j4b.zip, Oct. 1, 2012. Updated bmp_j4 .bmp model to work with zpaq v6.xx
zpaq v6.05, Oct. 2, 2012. Adds -list -history -summary, Linux port, bug fixes, and improved docs.
zpaq v6.06, Oct. 4, 2012. Simplifies -list and adds -compare.

zpaq v6.07, Oct. 7, 2012. Fixes porting issues with Mac OS/X and Visual C++.
lazy v1.00, Oct. 10, 2012. A fast LZ77 compressor/preprocessor and and config.
lazy2 v1.00, Oct. 31, 2012. lazy with an E8E9 filter (and 1 GB file size limit).
zpaq v6.16, Nov. 5, 2012. Better compression using lazy (-method 1) + e8e9 (all methods). Adds -test and -post.
zpaq v6.17, Dec. 13, 2012. Fixed display of international characters. libzpaq v6.17 has slightly faster SHA1. Has bugs. Do not use.
zpaq v6.18, Dec. 14, 2012. Bug fix.
zpaq v6.19, Jan. 23, 2013. Splits into zpaq (journaling archiver) and zpaqd (development tool). Adds methods 5-9. libzpaq v6.19 adds single pass compression checksums.
bmp_j4c, Jan. 24, 2013. Updated .bmp config file to work with new zpaq/zpaqd syntax.
zpaq v6.20, Feb. 1, 2013. Improved compression for methods 5 through 9. zpaq64.exe added Feb. 4, 2013.
zpaq v6.21, Feb. 6, 2013. Extract directories restores timestamps and attributes. Adds -until date. Lists alphabetically. Fixed docs. zpaq621-64.exe added Feb. 8, 2013.
zpaq v6.22, Feb. 13, 2013. -method supports custom algorithms. zpaqd and libzpaq fixes for Win64. Command line accepts international characters.
zpaq v6.23, Mar. 25, 2013. -method supports config files without preprocessors. zpaqd 6.23 speed improvements for g++ 4.7.0 and "ds" command. libzpaq 6.23 faster initialization.
zpaq v6.24, Apr. 12, 2013. Adds d (delete) command. Works with wildcards. zpaqd adds built-in configs 1..3.
zpaq v6.24a, Apr. 14, 2013. Recompile zpaq.exe, zpaqd.exe to get around compiler bug in 64 bit version of MinGW causing 32 bit zpaq to crash in WinXP.
zpaq v6.25, Apr. 19, 2013. libzpaq optimizations (3-5% faster) and bug fix for WinXP. No changes to zpaq or zpaqd except version number.
zpaq v6.26, May 9, 2013. Optimizations: zpaq improves grouping of incompressible files into blocks, faster StringBuffer. libzpaq JIT optimizes consecutive ZPAQL increments. zpaqd fixes compiler warning.
zpaq v6.27, May 13, 2013. Adds -all and -test options. Improved recovery of damaged archives. zpaqd updated to verify checksums when listing journaling archives.
zpaq v6.28, May 22, 2013. Changed zpaq -test to a command. Improved handling of damaged archives.
zpaq202.pdf, June 3, 2013. Level 2 revision 2 of spec adds a fragmentation recommendation for deduplication compatibility.
zpaq v6.29, June 6, 2013. Improved compression. Extended method 1 and 2 LZ77 parameters. Test command implements new 2.02 spec.
zpaq v6.30, June 7, 2013. Fixes bug in extracting read-only files. Adds -attr option.
zpaq v6.31, June 7, 2013. Changed -attr default to select all files.
zpaqd v6.27, June 11, 2013. From zpaq 6.27 but now a separate distribution.
zpaqd v6.32, June 19, 2013. Faster I/O when linked with libzpaq v6.32 (included).
zpaqd v6.33, June 20, 2013. libzpaq 6.33 bug fix and recompile to fix list command. No change to zpaqd 6.32 source.
zpaq v6.33, June 21, 2013. Improved compression, supports block sizes, streaming mode, -fragile option and compress to empty archive. Removed -attributes, -above. -method is 0..7
zpaq v6.34, June 25, 2013. Supports long LZ77 offsets. -method is 0..6. Default block size increased to 64 MB for 2..6.
zpaq v6.35, June 28, 2013. LZ77 look-ahead and other improvements. Better handling of nonexistent input files.

zpaq v6.36, July 5, 2013. LZ77 compression improvements. Memory options for special methods.
zpaq v6.37, July 15, 2013. Adds purge command.
zpaq v6.38, July 18, 2013. Fixes extraction bug in v6.28-6.37. Adds compare command.
zpaq v6.39, July 25, 2013. List command shows compression ratios. Fixes -method 0 compression in DEBUG mode.
zpaq v6.40, July 28, 2013. Adds -noattributes option. Windows version does not add reparse points.
zpaq v6.41, Aug. 2, 2013. Adds restore command. Fixed wildcard handling and extract -fragile.
zpaq v6.42, Sept. 26, 2013. Adds list -duplicates, faster updates, minor bug fixes.
zpaq v6.43, Dec. 20, 2013. Adds -key (encryption), show, sha1, sha256 commands. Updates libzpaq to v6.43 (adds AES, SHA256, Scrypt)
zpaq v6.44, Jan. 9, 2014. Adds encrypt command, removes restore, show, sha1, sha256, changes extract to skip existing files instead of error, changes purge syntax, prompt for passwords without echo, faster -method 5, some minor bug fixes.
zpaq v6.45, Jan. 12, 2014. Improves compression by sorting files by case insensitive extension, then by decreasing size rounded to 16K. Fixed VC++ compile error in 6.44.
zpaq203.pdf, Jan. 16, 2014. Level 2 revision 3 of the specification adds encryption.
zpaq v6.46, Jan. 17, 2014. Improved compare, added -fragment option. Fixed extracting streaming encrypted archives.
zpaq v6.47, Jan. 21, 2014. Adds snip command to support remote backups. Extends -since to extract and compare. Increased compression buffers for better core utilization.
zpaq v6.48, Jan. 23, 2014. Adds join command. Renames snip to split. Optimized decoder in libzpaq.cpp 6.48.
zpaq v6.49, Jan. 31, 2014. Adds progress indicator and other UI improvements. test can take filename args. libzpaq.cpp 6.49 and Makefile to fix Mac OS/X compiler warnings.
zpaq v6.50, Mar. 21, 2014. Reduced compression levels to -method 0..5 with better compression. Added -nodelete. Remove encrypt command, replaced with purge -all -newkey. Supports split archives directly, replacing split and join commands.
zpaq v6.51, May 7, 2014. Better method 2 compression using LZ77 look-ahead. Allows filtering by attributes and clearing Windows archive bit. libzpaq.h v6.51 allows empty Arrays.
zpaq v6.52, June 6, 2014. Fixes some usability issues. Improved memory use for extraction. libzpaq.cpp v6.52 fixes some harmless Mac OS/X compiler warnings.
zpaq v6.53, June 13, 2014. Smaller zpaq.exe and zpaq64.exe. Updated docs. No functional changes. Updated June 19, 2014 to remove UPX compression of zpaq.exe and zpaq64.exe because of false virus detection.
zpaq v6.54, June 16, 2014. Identical to updated 6.53 without UPX compression.
zpaq v6.55, July 24, 2014. Fixes cosmetic bugs in zpaq extract display.
zpaq204.pdf, Nov. 18, 2014. Adds a recommendation for multi-part archives with indexes for remote backups.
zpaq v6.56, Nov. 22, 2014. Adds indexes for multi-part archives, -method i, until -version, compare -with. Embeds divsufsort in zpaq.cpp. Separates docs to zpaq.pod.
zpaq v6.57, Nov. 25, 2014. Fixes excess memory usage bug introduced in v6.56.
zpaq v6.58, Dec. 10, 2014. Replaced purge with extract -to out.zpaq. Replaced compare with list -not =. Removed delete, extract -since, -not :attr, add -to. Added -only. Simplified test.
zpaq v6.59, Dec. 12, 2014. Added test -until, extract -to out.zpaq, list -to other.zpaq with renaming.
zpaq v6.60, Dec. 19, 2014. Requires "-method i" to update an index.
zpaq v7.00, Jan. 30, 2015. Adds compression methods to libzpaq. Removes -quiet, -fragment, -fragile, -since, -duplicates, -newkey. Allows add -to, extract -all, list files (compare).

zpaq v7.01, Feb. 9. 2015. Licensed changed from GPL v3 to public domain. Restores -fragment option.
zpaq v7.02, Feb. 13, 2015. Adds -test option. libzpaq v7.02 fixes some bugs in handling of malformed archives.
zpaq v7.03, Mar. 11, 2015. Removed test cmd. Some bug fixes. Supports alternate data streams (Win64). libzpaq 7.03 increases decoder buffer size for better speed.
zpaq v7.04, Mar. 20, 2015. Fixed stall when compression runs out of memory. Fixed detecting number of CPUs in BSD.
zpaq v7.05, Apr. 17, 2015. Fixed -method 111. libzpaq.cpp v7.05 fixes a valgrind warning (LZ77 read past end of input).
zpaq v7.06, Mar. 16, 2016. Fixes handling of some corrupted archives. Conforms to new spec zpaq205.pdf. New man page, Makefile, COPYING.
zpaq v7.07, Mar. 18, 2016. Fixes v7.06 bug in creating multipart encrypted archives with incorrectly salted index.
zpaq206.pdf, Mar. 22, 2016 makes fragment list sizes in D blocks required and references unzpaq206.cpp, which is updated to read journaling and encrypted formats.
zpaqd v7.07, Mar. 26, 2016. Development tool adds encrypt command. Links to updated libzpaq 7.06.
zpaq v7.08, Mar. 30, 2016. Removes requirement for separate WinXP version. Fixes Intel and VS 2015 compiler errors. Removes multi-part archive support, -nodelete, add -test, and -key prompt. Updated to libzpaq v7.08 (smaller decoder buffer). Updated Makefile to link libzpaq.o statically.
zpaq v7.09, Apr. 5, 2016. Fixes bug in extracting streaming archive with empty first file name.
zpaq v7.10, Apr. 8, 2016. Adds multi-part archives, -index. Some UI changes. Updates libzpaq.h, zpaq.pod.
zpaq v7.11, Apr. 13, 2016. Adds -repack, -encrypt. Updates libzpaq.cpp, zpaq.pod
zpaq v7.12, Apr. 26, 2016. Faster extract. Removes -encrypt (combined with -repack). Updates libzpaq.h, zpaq.pod. Apr. 29, 2016: added zpaq-gcc481.exe for older machines.
zpaq v7.13, May 4, 2016. Adds support for sparse files in Windows.
zpaqd v7.08, June 21, 2016. Fixes double close in r command to output file.
zpaq v7.14, July 19, 2016. Faster backup to a network drive. Fixes Makefile for Mac.
zpaq 7.15 and zpaqd v7.15, Aug. 17, 2016. Fixes incorrect JIT assembly of "a+= 128" in libzpaq. Fixes "make install" on Mac.

Discussion about ZPAQ updates.

ZPAQ is intended to replace PAQ and its variants (PAQ8, PAQ9A, LPAQ, LPQ1, etc) with similar or better compression in a portable, standard format. Current versions of PAQ break archive compatibility with each compression improvement. ZPAQ is intended to fix that. I no longer maintain the older PAQ code.

Related projects

PPA packages by Anton Batenev.
Total Commander plug-in by Andrei Piasetski
MMN backup maker (uses zpaq64) by Marcelo Marchi Negreira
zpaq fork by Paul Harris
Open source ZPAQ GUI by thometal.
Zpaq Explorer GUI for browsing archives (list and extract only) by Luis León.

WinZPAQ GUI interface (discussion) for benchmark testing, by Sportman.
PeaZip by Giorgio Tani, supports ZPAQ streaming format, among many others. Does not support journaling format.
PAKKA GUI versioned extractor.
Github Repository maintained by Evan Nemerson and Yonggang Luo. Includes history and convert script
Debian packages, very old version.
Python subset to ZPAQL compiler (in Rust) and thesis by Kai Lüke.
bandizip will extract zpaq.

Contact

zpaq was written by Matt Mahoney, mattmahoneyfl (at) gmail (dot) com