A free Macintosh Port of Info-ZIP's

Zip and UnZip

By Dirk Haase
www.haase-online.de/dirk/maczip
Please find my email address on my homepage.

==

Abstract:

MacZip is a cross-platform compatible tool that includes both Zip (for compression) and UnZip (for extraction).

Zip is a compression and file packaging utility for Unix, VMS, MSDOS, OS/2, Windows 9x, Windows NT, Atari, Macintosh, Amiga, Acorn RISC OS, and other systems.

UnZip unpacks zip archives. The Zip and UnZip programs can process archives produced by PKZIP, and PKZIP and PKUNZIP can work with archives produced by zip. Zip version 2.2 is compatible with PKZIP 2.04.

If you are new to MacZip please read first the file "ReadMe.1st".

License:

See the accompanying file LICENSE, version 2000-Mar-25 or later for terms of use. If, for some reason, both of these files are missing, the Info-ZIP license also may be found at: ftp://ftp.freesoftware.com/pub/infozip/license.html

Requirements

MacZip requires at least System 7 and a Macintosh with a minimum of a Motorola 68020 or PowerPC 601 processor. Other configurations may work but it is not tested at all.

The application is distributed as a fat binary with both regular 68K and native PowerPC versions included.

Installation

Move the executable(s) somewhere--for example, drag it (or them) to your Applications folder. For easy access, make an alias in the Launcher Control Panel or directly on your desktop. The GUI is very simple. It was not my intention to make a full-blown GUI, however I think it is comfortable enough to use it as regular tool.

This port supports also Apple-event. So you can install it in your WWW-Browser as a helper app.

For more Info about the contents of this package, take a look into the "macos/Contents" (or :macos:Contents) file. Some notes on how to rebuild the Macintosh applications can be found in INSTALL.

Usage:

Basically there are four ways to start MacZip:

a) Drag'n Drop
b) using the Dialog box (Menu: File -> Zip/Unzip):

Please read the file "ReadMe.1st" for the description of the items a and b.

c) Using the Command line (Menu: File->Command Line):
The Zip & UnZip tools are command line tools. So the behavior is exactly the same like the Zip & UnZip tools on Unix or Windows/DOS. This means, if you want to zip some files, you have to write a command line like this: "zip [switches] path_to_zip_archive path_to_files_folders"

- Go to "File" and select "Command Line" and the "MacZip Entry box" Dialog Box appears.

An example:

a: your zip may be created at
Macintosh HD:applications:archive.zip

b: your files may be found at
Macintosh HD:somewhere:my_folder_to_archive:*

Note: At the end of the path there must be a filename or a wild card ! (see Footnote: 1 wild card, 2 Mac path names)

So the command line should look like (one line!):

```
zip "Macintosh HD:applications:archive.zip" "Macintosh HD:somewhere:my_folder_to_archive:*"
```

- Click on "Enter" to start the task.

Since you can not set a default folder you have to enter always a full qualified path names. Full-qualified path names are path names including the Volume name ! (see Footnote: 2 Mac path names)

d) Using Applescript:

There is only one additional event defined: "do_cmd". You can enter every valid command line. The first word must be "zip" or "unzip" to select the action (compress or extraction).

See sample Applescript:

```
tell application "MacZip (PPC)"
   activate
   with timeout of 90000 seconds
      do_cmd "zip -rjjN Volume:archive \"My Volume:*\" "
   end timeout
end tell
```

This script opens MacZip, brings it to the foreground on the Mac, starts the zip action with the command line: zip -rjjN Volume:archive "My Volume:*" .

A short introduction is also available online: http://www.sitec.net/maczip/How-To-Do/

It's possible to stop the run of Zip/Unzip with the well known shortcut [Command] + [.].

There are some Mac-specific switches available.

Zip Module:

```
-df    [MacOS] Include only data-fork of files zipped into
       the  archive.   Good for exporting files to foreign
       operating-systems.  Resource-forks will be  ignored
       at all.

-jj    [MacOS] record Fullpath (+ Volname).  The  complete
       path  including  volume  will be stored. By default
       the relative path will be stored.

-S     [MSDOS, OS/2, WIN32 and ATARI] Include  system  and
       hidden files.
       [MacOS]  Includes finder invisible files, which are
       ignored otherwise.
```

Unzip Module:

```
-E     [MacOS only] display contents of MacOS extra  field
```

during restore operation.

-i [MacOS only] ignore filenames stored in MacOS extra fields. Instead, the most compatible filename stored in the generic part of the entry's header is used.

-J [MacOS only] ignore MacOS extra fields. All Macintosh specific info is skipped. Data-fork and resource-fork are restored as separate files.

Select [File]->[Get Help on Zip/Unzip] for a complete list of switches.

Limitations / Problems:

---

- Aliases are not supported. I tried, but I got broken aliases This port will silently ignore all aliases. It's on my to-do list for future releases.

- Zip needs much memory to compress many files: You may need to increase the 'Preferred Size' in 'Get Info'. Values of 12 Megabytes or more are possible

- Unzip needs about 500 Kbytes of memory to unzip no matter how many files were compressed and expanded.

- and finally one big macintosh-related problem: This port has one weak point: It's based on path names. As you may be already know: Path names are not unique on a Mac ! The main reason is that an attempt to implement support exact saving of the MacOS specific internal file structures would require a throughout rewrite of major parts of shared code, probably sacrifying compatibility with other systems. I have no solution at the moment. The port will just warn you if you try zip from / to a volume which has a duplicate name. MacZip has problems to find the archive or the files. My (Big) recommendation: Name all your volumes with a unique name and MacZip will run without any problem.

Known Bugs:

- crypted files in a zip archive are sometimes corrupt: I get an error message: invalid compressed data to inflate. Appearance of this error is purely be chance: I did a small test: Unzipping an archive containing 3589 files 56 files fails to unzip, so about 1.5%. Root cause is completely unclear to me :(

I strongly recommend to test your archive (e.g. unzip -t archive).

Zip Programs / Macintosh Extra-Data:

A brief overview: Currently, as far as I know, there are 6 Zip programs available for the Macintosh platform. These programs build (of course) different variants of Zip files:

- **Info-ZIP's first Port of Zip**. Ported by Johnny Lee This Port is rather outdated and no longer supported (since 1992). 68K only. Only minimal Mac-info is stored (Creator/Type, Finder attributes). Creator/Type: '????' / '????' Until year 1998, only UnZip 5.32 survived.

- **ZipIt** by Tom Brown. This is Shareware and still supported I think. ZipIt has a nice GUI, but I found it can't handle large Zip files quite well. ZipIt compresses Macintosh files using the Mac Binary format. So, transferring files to other platforms is not so easy. Only minimal Mac-info is stored (Creator/Type, Finder attributes). Mac filenames are changed to a most compatible filename. Creator/Type: 'ZIP ' / 'ZIP '

- **PKZIP/mac v2.03/210d**. This is Shareware. This Zip implementation for the Mac can be found on ASI's web site (http://www.asizip.com/products/products.htm). The name of this program is misleading, it is NOT a product from PKWARE. ASI's last release version is v2.03, and they also offer a newer beta version PKZIP/mac 210d. But even the Beta version is rather outdated (1995). Only minimal Mac-info is stored (Creator/Type, Finder attributes). The Zipfile format looks like incompatible to other platforms. (More details about the compatibility issue can be found in proginfo/3rdparty.bug!). Type: 'PKz1' Mac filenames are restored without any change.

- **Aladdin DropZip** 1999, This is Shareware. Aladdin choosed

the format of ZipIt. Therefore, it has the some drawbacks
like ZipIt. Creator/Type: 'SITx' / 'ZIP '

- **SmartZip 1.0** 1999 - by Marco Bambini Vampire Software.
This is Shareware. SmartZip compresses Macintosh files using
the Mac Binary. Therefore, it has the same drawbacks like
ZipIt. Creator/Type: 'dZIP' / 'ZIP '

and finally:

- Info-ZIP's latest Port of Zip. **MacZip 1.0**. Ported by me :-)
It is supported (of course) and up to date. Full set of
macintosh info is stored: Creator/Type, Finder attributes,
Finder comments, MacOS 8.0 Folder settings, Icon/Folder
Positions ... Mac filenames are restored without any change.
Creator/Type: 'IZip' / 'ZIP '

Compatibility of my port; Extraction:

- Archives from Info-ZIP's first port (by Johnny Lee) are still compatible.
- Extraction of ZipIt archives is supported. This support is not complete: Filenames are correct but Directory names are sometimes mangled to a DOS compatible form. Segmented archives are not supported.
- PKZiP/mac archive files are extracted without resource-forks and without any Finder info. I have no information about that zip format.

Compatibility of my port; Compression:

- My port supports only the new Info-ZIP format (introduced with this port). Therefore archives created by MacZip 1.0 (March 1999) must be extracted with this version or later releases of Info-ZIP's UnZip to restore the complete set of Macintosh attributes.

Note: This port is completly unrelated to the shareware ZipIt. Even more, handling of special Macintosh attributes is incompatible with ZipIt. This port (MacZip) may be used to extract archives created by ZipIt, but make sure that you get the result as you expected.

Macintosh Files; File Forks:

All Macintosh files comprise two forks, known as the data fork and the resource fork. Unlike the bytes stored in the resource fork, the bytes in the data fork do not have to exhibit any particular internal structure. The application is responsible for interpreting the bytes in the data fork in whatever manner is appropriate. The bytes in the resource fork usually have a defined internal structure and contain data object like menus, dialog boxes, icons and pictures. Although all Macintosh files contain both a data fork and a resource fork, one or both of these forks may be empty.

MacZip stores data-forks and resource-forks separately. The Zipfile format does not allow to store two archive entries using exactly the same name. My solution is to modify the Path name of the resource-fork. All resource-fork names are prepended with a leading special directory named "XtraStuf.mac". So, when extracting on a Mac, you should never see this directory "XtraStuf.mac" on your *disk*.

On all foreign systems that support directories in filenames (e.g.: OS/2, Unix, DOS/Windows, VMS) you will get a directory "XtraStuf.mac" when extracting MacZip archives. You can delete the complete directory "XtraStuf.mac" since Mac resources do not make much sense outside the MacOS world.

Text encoding; Charsets of the Filenames:

The following information is only important if you plan to transfer archives across different platforms/language systems:

A typical Zip archive does not support different charsets. All filenames stored in the public area (= accessible by foreign systems other than MacOS) must be coded in the charset ISO-8859-1 (CP1252 in the Microsoft Windows world) or CP850 (DOSLatin1). The latter should only be used by Zip programs that mark the archive entries as "created under DOS". Apart from Macs, the commonly used platforms either support ISO-8859-1 directly, or are compatible with it. To achieve maximum compatibility, MacZip convert filenames from the Mac OS Roman character set to ISO-8859-1 and vice versa. But not every char of the charset MacRoman has their equivalent in ISO-8859-1. To make the mapping in most cases possible, I chose most similar chars or at least the MIDDLE DOT.

Mac OS Roman character set is used for at least the following Mac OS localizations: U.S., British, Canadian French, French, Swiss French, German, Swiss German, Italian, Swiss Italian, Dutch, Swedish, Norwegian, Danish, Finnish, Spanish, Catalan, Portuguese, Brazilian, and the default International system.

In all Mac OS encodings, character codes 0x00-0x7F are identical to ASCII, except that - in Mac OS Japanese, yen sign replaces reverse solidus - in Mac OS Arabic, Farsi, and Hebrew, some of the punctuation in this range is treated as having strong left-right directionality, although the corresponding Unicode characters have neutral directionality So, for best compatibility, confine filenames to the standard 7-bit ASCII character set.

If you generate a filename list of your archive (unzip -l), you will see the converted filenames. Your can also extract the archive with the switch '-i' (= ignore mac filenames), and test your result.

This MacZip port uses its own filename stored in the archive. At the moment, the filename will be not converted. However, I'm planning to add support for Unicode.

Currently, the following Mac OS encodings are NOT supported: Japanese, ChineseTrad, Korean, Arabic, Hebrew, Greek, Cyrillic, Devanagari, Gurmukhi, Gujarati, Oriya, Bengali, Tamil, Telugu Kannada, Malayalam, Sinhalese, Burmese, Khmer, Thai, Laotian, Georgian, Armenian, ChineseSimp, Tibetan, Mongolian, Ethiopic, Vietnamese, ExtArabic and finally: Symbol - this is the encoding for the font named "Symbol". Dingbats - this is the encoding for the font named "Zapf Dingbats". If you extract an archive coded with one of these charsets you will probably get filenames with funny characters.

These problems apply only to filenames and NOT to the file content. of course: The content of the files will NEVER be converted !!

File-/Creator Type:

This port uses the creator type 'IZip' and it is registered at Apple (since 08. March 1998). File types can not be registered any more. This port uses 'ZIP ' for Zip archive files. The creator 'IZip' type should be used for all future versions of MacZip.

Hints for proper restoration of file-time stamps:

UnZip requires the host computer to have proper time zone information in order to handle certain tasks correctly (see unzip.doc). To set the time zone on the Macintosh, go to the Map Control Panel and enter the correct number of hours (and, in a few locales, minutes) offset from Universal Time/Greenwich Mean Time. For example, the US Pacific time zone is -8 hours from UTC/GMT during standard (winter) time and -7 hours from UTC/GMT during Daylight Savings Time. The US Eastern time zone is -5 hours during the winter and -4 hours during the summer.

Discussion of Daylight Savings Time:

The setting in the Date & Time control panel for Daylight Savings time is a universal setting. That is, it assumes everybody in the world is observing Daylight Savings time when its check box is selected.

If other areas of the world are not observing Daylight Savings time when the check box is selected in the Date & Time control panel, then the Map control panel will be off by an hour for all areas that are not recognizing Daylight Savings time.

Conversely, if you set the Map control panel to an area that does not observe Daylight

Savings time and deselect/uncheck the check box for Daylight Savings time in the Date & Time control panel, then time in all areas celebrating Daylight Savings time will be off by an hour in the Map control panel.

Example:

> In the case of Hawaiians, sometimes they are three hours behind Pacific Standard Time (PST) and sometimes two hours behind Pacific Daylight Time (PDT). The Map control panel can only calculate differences between time zones relative to Greenwich Mean Time (GMT). Hawaii will always show up as three hours past the Pacific time zone and five hours past the Central time zone.
>
> When Hawaiians are not observing Daylight Savings time, but the rest of the country is, there is no combination of settings in Map and Date & Time control panels which will enable you to display Hawaiian local time correctly AND concurrently display the correct time in other places that do observe Daylight Savings time.
>
> The knowledge about which countries observe Daylight Savings time and which do not is not built into the Map control panel, so it does not allow for such a complex calculation.
>
> This same situation also occurs in other parts of the world besides Hawaii. Phoenix, Arizona is an example of an area of the U.S. which also does not observe Daylight Savings time.

Conclusion:
MacZip only knows the GMT and DST offsets of the current time, not for the time in question.

Projects & Packages:

A Note to version numbers: Version of MacZip is currently 1.03 and is based on the zip code version 2.3 and unzip code version 5.4. See About Box for current version and compiler build date.

Because of the amount of sources I splitted this port into several projects. See http://www.sitec.net/maczip for updates.

- core source parts:
  unzxxx.zip
  zipxxx.zip

These archives contains the main parts of the port. You can build libraries and a standalone App with Metrowerks standard console SIOUX. They contain only sources, no executables. These archives are exact copies of the standard Info-ZIP source distributions; they were only repackaged under MacOS using MacZip, with one minor addition: For those files that are stored in BinHex'ed format in the Info-ZIP reference source archives, unpacked version that are ready for use have been added.

- additional source part:

  MacZipxxx.zip: contains all the GUI stuff and the project files to build the main-app. Only sources of the GUI, no zip or unzip code. To build MacZip successfully you will need to also download the zip and unzip packages.

- executables:

  MacZipxxxnc.hqx: contains only executables and 'README.TXT', This version is without en-/decryption support !
  MacZipxxxc.hqx: contains only executables and 'README.TXT', This version supports en-/decryption !

- encryption sources:

  zcryptxx.zip: To build crypt versions of MacZip.
  download from ftp://ftp.icce.rug.nl/infozip/ (and subdirectories)

- documentation:

  MacZipDocu.zip: contains some further docus about the algorithm, limits, Info-ZIP's appnote and a How-to-do Webpage.

Credits:

Macstuff.c and recurse.c: All the functions are from More Files. More Files fixes many of the broken or underfunctional parts of the file system. Thanks to Jim Luther. (see morefiles.doc)

Footnotes:

1. wild card:

The '*' is a wild card and means 'all files'
Just in case you don't know wild cards:
'*' is a place holder for any character.
e.g.:
"this*" matches with "this_file" or "this_textfile" but it
doesn't match with "only_this_file" or "first_this_textfile"
"*this*" matches with "this_file" or "this_textfile" AND
matches with "only_this_file" or "first_this_textfile"

2. Mac pathnames:

The following characteristics of Macintosh pathnames should be noted:

A full pathname never begins with a colon, but must contain at least one colon. A partial pathname always begins with a colon separator except in the case where the file partial pathname is a simple file or directory name. Single trailing separator colons in full or partial pathnames are ignored except in the case of full pathnames to volumes. In full pathnames to volumes, the trailing separator colon is required. Consecutive separator colons can be used to ascend a level from a directory to its parent directory. Two consecutive separator colons will ascend one level, three consecutive separator colons will ascend two levels, and so on. Ascending can only occur from a directory; not a file.

Dirk Haase
==========