Rocky Enterprise Linux 9.2 Manual Pages on command 'tzfile.5'

C:\>man tzfile.5

TZFILE(5) Linux Programmer's Manual TZFILE(5)

NAME

tzfile - timezone information

DESCRIPTION

The timezone information files used by tzset(3) are typically found under a direc? tory with a name like /usr/share/zoneinfo. These files begin with a 44-byte header containing the following fields:

* The magic four-byte ASCII sequence ?TZif? identifies the file as a timezone in? formation file.
* A byte identifying the version of the file's format (as of 2017, either an ASCII NUL, or ?2?, or ?3?).
* Fifteen bytes containing zeros reserved for future use.
* Six four-byte integer values written in a standard byte order (the high-order byte of the value is written first). These values are, in order:

tzh_ttisgmtcnt

The number of UT/local indicators stored in the file.

tzh_ttisstdcnt

The number of standard/wall indicators stored in the file.

tzh_leapcnt

The number of leap seconds for which data entries are stored in the file.

tzh_timecnt

The number of transition times for which data entries are stored in the

file.

tzh_typecnt

The number of local time types for which data entries are stored in the file (must not be zero).

tzh_charcnt

The number of bytes of time zone abbreviation strings stored in the file.

The above header is followed by the following fields, whose lengths depend on the contents of the header:

* tzh_timecnt four-byte signed integer values sorted in ascending order. These values are written in standard byte order. Each is used as a transition time (as returned by time(2)) at which the rules for computing local time change.
* tzh_timecnt one-byte unsigned integer values; each one but the last tells which of the different types of local time types described in the file is associated with the time period starting with the same-indexed transition time and continu? ing up to but not including the next transition time. (The last time type is present only for consistency checking with the POSIX-style TZ string described below.) These values serve as indices into the next field.
* tzh_typecnt ttinfo entries, each defined as follows:

```
struct ttinfo {
    int32_t       tt_gmtoff;
    unsigned char  tt_isdst;
    unsigned char  tt_abbrind;
};
```

Each structure is written as a four-byte signed integer value for tt_gmtoff, in a standard byte order, followed by a one-byte value for tt_isdst and a one-byte value for tt_abbrind. In each structure, tt_gmtoff gives the number of seconds to be added to UT, tt_isdst tells whether tm_isdst should be set by localtime(3) and tt_abbrind serves as an index into the array of time zone abbreviation bytes that follow the ttinfo structure(s) in the file.

* tzh_leapcnt pairs of four-byte values, written in standard byte order; the first value of each pair gives the nonnegative time (as returned by time(2)) at which a leap second occurs; the second gives the total number of leap seconds to be ap? plied during the time period starting at the given time. The pairs of values are

sorted in ascending order by time. Each transition is for one leap second, ei?
ther positive or negative; transitions always separated by at least 28 days minus
1 second.

* tzh_ttisstdcnt standard/wall indicators, each stored as a one-byte value; they
tell whether the transition times associated with local time types were specified
as standard time or wall clock time, and are used when a timezone file is used in
handling POSIX-style timezone environment variables.

* tzh_ttisgmtcnt UT/local indicators, each stored as a one-byte value; they tell
whether the transition times associated with local time types were specified as
UT or local time, and are used when a timezone file is used in handling POSIX-
style timezone environment variables.

The localtime(3) function uses the first standard-time ttinfo structure in the file
(or simply the first ttinfo structure in the absence of a standard-time structure)
if either tzh_timecnt is zero or the time argument is less than the first transi?
tion time recorded in the file.

NOTES

This manual page documents <tzfile.h> in the glibc source archive, see timezone/tz?
file.h.

It seems that timezone uses tzfile internally, but glibc refuses to expose it to
userspace. This is most likely because the standardised functions are more useful
and portable, and actually documented by glibc. It may only be in glibc just to
support the non-glibc-maintained timezone data (which is maintained by some other
entity).

Version 2 format

For version-2-format timezone files, the above header and data are followed by a
second header and data, identical in format except that eight bytes are used for
each transition time or leap second time. (Leap second counts remain four bytes.)
After the second header and data comes a newline-enclosed, POSIX-TZ-environment-
variable-style string for use in handling instants after the last transition time
stored in the file or for all instants if the file has no transitions. The POSIX-
style TZ string is empty (i.e., nothing between the newlines) if there is no POSIX
representation for such instants. If nonempty, the POSIX-style TZ string must
agree with the local time type after the last transition time if present in the

eight-byte data; for example, given the string ?WET0WEST,M3.5.0,M10.5.0/3? then if a last transition time is in July, the transition's local time type must specify a daylight-saving time abbreviated ?WEST? that is one hour east of UT. Also, if there is at least one transition, time type 0 is associated with the time period from the indefinite past up to but not including the earliest transition time.

Version 3 format

For version-3-format timezone files, the POSIX-TZ-style string may use two minor extensions to the POSIX TZ format, as described in newtzset(3). First, the hours part of its transition times may be signed and range from -167 through 167 instead of the POSIX-required unsigned values from 0 through 24. Second, DST is in effect all year if it starts January 1 at 00:00 and ends December 31 at 24:00 plus the difference between daylight saving and standard time.

Future changes to the format may append more data.

SEE ALSO

time(2), localtime(3), tzset(3), tzselect(8), zdump(8), zic(8)

COLOPHON

This page is part of release 5.05 of the Linux man-pages project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at https://www.kernel.org/doc/man-pages/.

2019-03-06 TZFILE(5)