Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'zic.8' command

$ man zic.8

ZIC(8) Linux System Administration ZIC(8)

NAME

zic - timezone compiler

SYNOPSIS

zic [option ...] [filename ...]

DESCRIPTION

The zic program reads text from the file(s) named on the command line and creates the time conversion information files specified in this in? put. If a filename is ?-?, standard input is read.

OPTIONS

--version

Output version information and exit.

--help Output short usage message and exit.

-b bloat

Output backward-compatibility data as specified by bloat. If bloat is fat, generate additional data entries that work around potential bugs or incompatibilities in older software, such as software that mishandles the 64-bit generated data. If bloat is slim, keep the output files small; this can help check for the bugs and incompatibilities. Although the default is currently fat, this is intended to change in future zic versions, as soft? ware that mishandles the 64-bit data typically mishandles time? stamps after the year 2038 anyway. Also see the -r option for

another way to shrink output size.

-d directory

Create time conversion information files in the named directory rather than in the standard directory named below.

-l timezone

Use timezone as local time. zic will act as if the input con? tained a link line of the form

Link timezone localtime

-L leapsecondfilename

Read leap second information from the file with the given name. If this option is not used, no leap second information appears in output files.

-p timezone

Use timezone's rules when handling nonstandard TZ strings like "EET-2EEST" that lack transition rules. zic will act as if the input contained a link line of the form

Link timezone posixrules

This feature is obsolete and poorly supported. Among other things it should not be used for timestamps after the year 2037, and it should not be combined with -b slim if timezone's transi? tions are at standard time or Universal Time (UT) instead of lo? cal time.

-r [@lo][/@hi]

Reduce the size of output files by limiting their applicability to timestamps in the range from lo (inclusive) to hi (exclu? sive), where lo and hi are possibly-signed decimal counts of seconds since the Epoch (1970-01-01 00:00:00 UTC). Omitted counts default to extreme values. For example, ?zic -r @0? omits data intended for negative timestamps (i.e., before the Epoch), and ?zic -r @0/@2147483648? outputs data intended only for nonnegative timestamps that fit into 31-bit signed integers. On platforms with GNU date, ?zic -r @$(date +%s)? omits data in? tended for past timestamps. Also see the -b slim option for an?

other way to shrink output size.

-t file

When creating local time information, put the configuration link in the named file rather than in the standard location.

-v Be more verbose, and complain about the following situations:

The input specifies a link to a link.

A year that appears in a data file is outside the range of rep? resentable years.

A time of 24:00 or more appears in the input. Pre-1998 versions of zic prohibit 24:00, and pre-2007 versions prohibit times greater than 24:00.

A rule goes past the start or end of the month. Pre-2004 ver? sions of zic prohibit this.

A time zone abbreviation uses a %z format. Pre-2015 versions of zic do not support this.

A timestamp contains fractional seconds. Pre-2018 versions of zic do not support this.

The input contains abbreviations that are mishandled by pre-2018 versions of zic due to a longstanding coding bug. These abbre? viations include ?L? for ?Link?, ?mi? for ?min?, ?Sa? for ?Sat?, and ?Su? for ?Sun?.

The output file does not contain all the information about the long-term future of a timezone, because the future cannot be summarized as an extended POSIX TZ string. For example, as of 2019 this problem occurs for Iran's daylight-saving rules for the predicted future, as these rules are based on the Iranian calendar, which cannot be represented.

The output contains data that may not be handled properly by client code designed for older zic output formats. These com? patibility issues affect only timestamps before 1970 or after the start of 2038.

The output file contains more than 1200 transitions, which may be mishandled by some clients. The current reference client

supports at most 2000 transitions; pre-2014 versions of the ref?
erence client support at most 1200 transitions.
A time zone abbreviation has fewer than 3 or more than 6 charac?
ters. POSIX requires at least 3, and requires implementations
to support at least 6.
An output file name contains a byte that is not an ASCII letter,
?-?, ?/?, or ?_?; or it contains a file name component that con?
tains more than 14 bytes or that starts with ?-?.

FILES

Input files use the format described in this section; output files use
tzfile(5) format.
Input files should be text files, that is, they should be a series of
zero or more lines, each ending in a newline byte and containing at
most 511 bytes, and without any NUL bytes. The input text's encoding
is typically UTF-8 or ASCII; it should have a unibyte representation
for the POSIX Portable Character Set (PPCS) ?http://pubs.opengroup.org/
onlinepubs/9699919799/basedefs/V1_chap06.html? and the encoding's non-
unibyte characters should consist entirely of non-PPCS bytes. Non-PPCS
characters typically occur only in comments: although output file names
and time zone abbreviations can contain nearly any character, other
software will work better if these are limited to the restricted syntax
described under the -v option.
Input lines are made up of fields. Fields are separated from one an?
other by one or more white space characters. The white space charac?
ters are space, form feed, carriage return, newline, tab, and vertical
tab. Leading and trailing white space on input lines is ignored. An
unquoted sharp character (#) in the input introduces a comment which
extends to the end of the line the sharp character appears on. White
space characters and sharp characters may be enclosed in double quotes
(") if they're to be used as part of a field. Any line that is blank
(after comment stripping) is ignored. Nonblank lines are expected to
be of one of three types: rule lines, zone lines, and link lines.
Names must be in English and are case insensitive. They appear in sev?

eral contexts, and include month and weekday names and keywords such as maximum, only, Rolling, and Zone. A name can be abbreviated by omit? ting all but an initial prefix; any abbreviation must be unambiguous in context.

A rule line has the form

```
Rule  NAME  FROM  TO    TYPE  IN   ON       AT     SAVE   LETTER/S
```

For example:

```
Rule  US    1967  1973  -     Apr  lastSun  2:00w  1:00d  D
```

The fields that make up a rule line are:

NAME Gives the name of the rule set that contains this line. The name must start with a character that is neither an ASCII digit nor ?-? nor ?+?. To allow for future extensions, an unquoted name should not contain characters from the set ?!$%&'()*,/:;<=>?@[\]^`{|}~?.

FROM Gives the first year in which the rule applies. Any signed in? teger year can be supplied; the proleptic Gregorian calendar is assumed, with year 0 preceding year 1. The word minimum (or an abbreviation) means the indefinite past. The word maximum (or an abbreviation) means the indefinite future. Rules can de? scribe times that are not representable as time values, with the unrepresentable times ignored; this allows rules to be por? table among hosts with differing time value types.

TO Gives the final year in which the rule applies. In addition to minimum and maximum (as above), the word only (or an abbrevia? tion) may be used to repeat the value of the FROM field.

TYPE should be ?-? and is present for compatibility with older ver? sions of zic in which it could contain year types.

IN Names the month in which the rule takes effect. Month names may be abbreviated.

ON Gives the day on which the rule takes effect. Recognized forms include:

```
5        the fifth of the month
lastSun  the last Sunday in the month
```

lastMon the last Monday in the month

Sun>=8 first Sunday on or after the eighth

Sun<=25 last Sunday on or before the 25th

A weekday name (e.g., Sunday) or a weekday name preceded by ?last? (e.g., lastSunday) may be abbreviated or spelled out in full. There must be no white space characters within the ON field. The ?<=? and ?>=? constructs can result in a day in the neighboring month; for example, the IN-ON combination ?Oct Sun>=31? stands for the first Sunday on or after October 31, even if that Sunday occurs in November.

AT Gives the time of day at which the rule takes effect, relative to 00:00, the start of a calendar day. Recognized forms in? clude:

2 time in hours

2:00 time in hours and minutes

01:28:14 time in hours, minutes, and seconds

00:19:32.13 time with fractional seconds

12:00 midday, 12 hours after 00:00

15:00 3 PM, 15 hours after 00:00

24:00 end of day, 24 hours after 00:00

260:00 260 hours after 00:00

-2:30 2.5 hours before 00:00

- equivalent to 0

Although zic rounds times to the nearest integer second (break? ing ties to the even integer), the fractions may be useful to other applications requiring greater precision. The source format does not specify any maximum precision. Any of these forms may be followed by the letter w if the given time is lo? cal or ?wall clock? time, s if the given time is standard time without any adjustment for daylight saving, or u (or g or z) if the given time is universal time; in the absence of an indica? tor, local (wall clock) time is assumed. These forms ignore leap seconds; for example, if a leap second occurs at 00:59:60

local time, ?1:00? stands for 3601 seconds after local midnight instead of the usual 3600 seconds. The intent is that a rule line describes the instants when a clock/calendar set to the type of time specified in the AT field would show the specified date and time of day.

SAVE Gives the amount of time to be added to local standard time when the rule is in effect, and whether the resulting time is standard or daylight saving. This field has the same format as the AT field except with a different set of suffix letters: s for standard time and d for daylight saving time. The suffix letter is typically omitted, and defaults to s if the offset is zero and to d otherwise. Negative offsets are allowed; in Ire? land, for example, daylight saving time is observed in winter and has a negative offset relative to Irish Standard Time. The offset is merely added to standard time; for example, zic does not distinguish a 10:30 standard time plus an 0:30 SAVE from a 10:00 standard time plus a 1:00 SAVE.

LETTER/S

Gives the ?variable part? (for example, the ?S? or ?D? in ?EST? or ?EDT?) of time zone abbreviations to be used when this rule is in effect. If this field is ?-?, the variable part is null.

A zone line has the form

Zone NAME STDOFF RULES FORMAT [UNTIL]

For example:

Zone Asia/Amman 2:00 Jordan EE%sT 2017 Oct 27 01:00

The fields that make up a zone line are:

NAME The name of the timezone. This is the name used in creating the time conversion information file for the timezone. It should not contain a file name component ?.? or ?..?; a file name component is a maximal substring that does not contain ?/?.

STDOFF

The amount of time to add to UT to get standard time, without any adjustment for daylight saving. This field has the same format

as the AT and SAVE fields of rule lines; begin the field with a minus sign if time must be subtracted from UT.

RULES The name of the rules that apply in the timezone or, alterna? tively, a field in the same format as a rule-line SAVE column, giving of the amount of time to be added to local standard time effect, and whether the resulting time is standard or daylight saving. If this field is - then standard time always applies. When an amount of time is given, only the sum of standard time and this amount matters.

FORMAT

The format for time zone abbreviations. The pair of characters %s is used to show where the ?variable part? of the time zone ab? breviation goes. Alternatively, a format can use the pair of characters %z to stand for the UT offset in the form ?hh, ?hhmm, or ?hhmmss, using the shortest form that does not lose informa? tion, where hh, mm, and ss are the hours, minutes, and seconds east (+) or west (?) of UT. Alternatively, a slash (/) separates standard and daylight abbreviations. To conform to POSIX, a time zone abbreviation should contain only alphanumeric ASCII charac? ters, ?+? and ?-?.

UNTIL The time at which the UT offset or the rule(s) change for a loca? tion. It takes the form of one to four fields YEAR [MONTH [DAY [TIME]]]. If this is specified, the time zone information is generated from the given UT offset and rule change until the time specified, which is interpreted using the rules in effect just before the transition. The month, day, and time of day have the same format as the IN, ON, and AT fields of a rule; trailing fields can be omitted, and default to the earliest possible value for the missing fields.

The next line must be a ?continuation? line; this has the same form as a zone line except that the string ?Zone? and the name are omitted, as the continuation line will place information starting at the time specified as the ?until? information in the

previous line in the file used by the previous line. Continua?
tion lines may contain ?until? information, just as zone lines
do, indicating that the next line is a further continuation.

If a zone changes at the same instant that a rule would otherwise take effect in the earlier zone or continuation line, the rule is ignored.

A zone or continuation line L with a named rule set starts with stan? dard time by default: that is, any of L's timestamps preceding L's ear? liest rule use the rule in effect after L's first transition into stan? dard time. In a single zone it is an error if two rules take effect at the same instant, or if two zone changes take effect at the same in? stant.

A link line has the form

```
Link  TARGET          LINK-NAME
```

For example:

```
Link  Europe/Istanbul  Asia/Istanbul
```

The TARGET field should appear as the NAME field in some zone line. The LINK-NAME field is used as an alternative name for that zone; it has the same syntax as a zone line's NAME field.

Except for continuation lines, lines may appear in any order in the in? put. However, the behavior is unspecified if multiple zone or link lines define the same name, or if the source of one link line is the target of another.

The file that describes leap seconds can have leap lines and an expira? tion line. Leap lines have the following form:

```
Leap  YEAR  MONTH  DAY  HH:MM:SS  CORR  R/S
```

For example:

```
Leap  2016  Dec    31   23:59:60  +     S
```

The YEAR, MONTH, DAY, and HH:MM:SS fields tell when the leap second happened. The CORR field should be ?+? if a second was added or ?-? if a second was skipped. The R/S field should be (an abbreviation of) ?Stationary? if the leap second time given by the other fields should be interpreted as UTC or (an abbreviation of) ?Rolling? if the leap second time given by the other fields should be interpreted as local

(wall clock) time.

The expiration line, if present, has the form:

Expires YEAR MONTH DAY HH:MM:SS

For example:

Expires 2020 Dec 28 00:00:00

The YEAR, MONTH, DAY, and HH:MM:SS fields give the expiration timestamp in UTC for the leap second table; zic outputs this expiration timestamp by truncating the end of the output file to the timestamp. If there is no expiration line, zic also accepts a comment ?#expires E ...? where E is the expiration timestamp as a decimal integer count of seconds since the Epoch, not counting leap seconds. However, the ?#expires? comment is an obsolescent feature, and the leap second file should use an expi? ration line instead of relying on a comment.

EXTENDED EXAMPLE

Here is an extended example of zic input, intended to illustrate many of its features. In this example, the EU rules are for the European Union and for its predecessor organization, the European Communities.

```
# Rule  NAME  FROM  TO    TYPE  IN   ON       AT    SAVE  LETTER/S
Rule    Swiss 1941  1942  -     May  Mon>=1   1:00  1:00  S
Rule    Swiss 1941  1942  -     Oct  Mon>=1   2:00  0     -
Rule    EU    1977  1980  -     Apr  Sun>=1   1:00u 1:00  S
Rule    EU    1977  only  -     Sep  lastSun  1:00u 0     -
Rule    EU    1978  only  -     Oct  1        1:00u 0     -
Rule    EU    1979  1995  -     Sep  lastSun  1:00u 0     -
Rule    EU    1981  max   -     Mar  lastSun  1:00u 1:00  S
Rule    EU    1996  max   -     Oct  lastSun  1:00u 0     -
# Zone  NAME          STDOFF      RULES  FORMAT  [UNTIL]
Zone    Europe/Zurich  0:34:08    -      LMT     1853 Jul 16
                       0:29:45.50 -      BMT     1894 Jun
                       1:00       Swiss  CE%sT   1981
                       1:00       EU     CE%sT
Link    Europe/Zurich  Europe/Vaduz
```

In this example, the timezone is named Europe/Zurich but it has an

alias as Europe/Vaduz. This example says that Zurich was 34 minutes and 8 seconds east of UT until 1853-07-16 at 00:00, when the legal off? set was changed to 7?26?22.50?, which works out to 0:29:45.50; zic treats this by rounding it to 0:29:46. After 1894-06-01 at 00:00 the UT offset became one hour and Swiss daylight saving rules (defined with lines beginning with ?Rule Swiss?) apply. From 1981 to the present, EU daylight saving rules have applied, and the UTC offset has remained at one hour.

In 1941 and 1942, daylight saving time applied from the first Monday in May at 01:00 to the first Monday in October at 02:00. The pre-1981 EU daylight-saving rules have no effect here, but are included for com? pleteness. Since 1981, daylight saving has begun on the last Sunday in March at 01:00 UTC. Until 1995 it ended the last Sunday in September at 01:00 UTC, but this changed to the last Sunday in October starting in 1996.

For purposes of display, ?LMT? and ?BMT? were initially used, respec? tively. Since Swiss rules and later EU rules were applied, the time zone abbreviation has been CET for standard time and CEST for daylight saving time.

FILES

/etc/localtime

Default local timezone file.

/usr/share/zoneinfo

Default timezone information directory.

NOTES

For areas with more than two types of local time, you may need to use local standard time in the AT field of the earliest transition time's rule to ensure that the earliest transition time recorded in the com? piled file is correct.

If, for a particular timezone, a clock advance caused by the start of daylight saving coincides with and is equal to a clock retreat caused by a change in UT offset, zic produces a single transition to daylight saving at the new UT offset without any change in local (wall clock)

time. To get separate transitions use multiple zone continuation lines specifying transition instants using universal time.

SEE ALSO

tzfile(5), zdump(8)

COLOPHON

This page is part of release 5.10 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/.