

**NAME**

**zic** - timezone compiler

**SYNOPSIS**

**zic** [**--help**] [**--version**] [**-Dsv**] [**-b** *slim* | *fat*] [**-d** *directory*] [**-g** *gid*] [**-l** *localtime*] [**-L** *leapseconds*] [**-m** *mode*] [**-p** *posixrules*] [**-r** [*@lo*]/[*@hi*]] [**-R** *-@hi*] [**-t** *localtime-link*] [**-u** *uid*] [*filename ...*]

**DESCRIPTION**

The **zic** program reads text from the file(s) named on the command line and creates the timezone information format (TZif) files specified in this input. If a *filename* is "-", standard input is read.

The following options are available:

**--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. The default is **slim**, as software that mishandles 64-bit data typically mishandles timestamps after the year 2038 anyway. Also see the **-r** option for another way to alter output size.

**-D** Do not create directories.

**-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. The **zic** utility will act as if the input contained a link line of the form

Link	timezone	localtime
------	----------	-----------

If *timezone* is '-', any already-existing link is removed.

**-L** *filename*

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. The **zic** utility will act as if the input contained a link line of the form

```
Link    timezone        posixrules
```

If *timezone* is "" (the default), any already-existing link is removed.

Unless *timezone* is "", this option is obsolete and poorly supported.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 transitions are at standard time or Universal Time (UT) instead of local time.

**-r** [*@lo*][/*@hi*]

Limit the applicability of output files to timestamps in the range from *lo* (inclusive) to *hi* (exclusive), 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. The output files use UT offset 0 and abbreviation "-00" in place of the omitted timestamp data. For example, **-r -@0** omits data intended for negative timestamps (i.e., before the Epoch), and **-r -@0/@2147483648** outputs data intended only for nonnegative timestamps that fit into 31-bit signed integers. Although this option typically reduces the output file's size, the size can increase due to the need to represent the timestamp range boundaries, particularly if *hi* causes a TZif file to contain explicit entries for *pre-hi* transitions rather than concisely representing them with a proleptic TZ string. Also see the **-b slim** option for another way to shrink output size.

**-R -@hi**

Generate redundant trailing explicit transitions for timestamps that occur less than *hi* seconds since the Epoch, even though the transitions could be more concisely represented via the proleptic TZ string. This option does not affect the represented timestamps. Although it accommodates nonstandard TZif readers that ignore the proleptic TZ string, it increases the size of the altered output files.

**-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, something not supported by some older parsers, including **zic** itself through release 2022e.
- ⊕ A year that appears in a data file is outside the range of representable 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 versions 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 abbreviations 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 a proleptic TZ string. For example, as of 2023 this problem occurs for Morocco's daylight-saving rules, as these rules are based on predictions for when Ramadan will be observed, something that a proleptic TZ string cannot represent.
- ⊕ The output contains data that may not be handled properly by client code designed for older **zic** output formats. These compatibility issues affect only timestamps before 1970 or after the start of 2038.
- ⊕ The output contains a truncated leap second table, which can cause some older TZif readers to misbehave. This can occur if the **-L** option is used, and either an Expires line is present or the **-r** option is also used.
- ⊕ 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 reference client support at most 1200 transitions.
- ⊕ A time zone abbreviation has fewer than 3 or more than 6 characters. 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 contains 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 2048 bytes counting the newline, 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)

[https://pubs.opengroup.org/onlinepubs/9699919799/basedefs/V1\\_chap06.html](https://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 another by one or more white space characters. The white space characters 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 several contexts, and include month and weekday names and keywords such as "maximum", "only", "Rolling", and "Zone". A name can be abbreviated by omitting all but an initial prefix; any abbreviation must be unambiguous in context.

A rule line has the form

```
Rule   NAME  FROM  TO    -    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 integer year can be supplied; the proleptic Gregorian calendar is assumed, with year 0 preceding year 1. Rules can describe times that are not representable as time values, with the unrepresentable times ignored; this allows rules to be portable among hosts with differing time value types.
- TO** Gives the final year in which the rule applies. The word **maximum** (or an abbreviation) means the indefinite future, and the word **only** (or an abbreviation) may be used to repeat the value of the *FROM* field.
- Is a reserved field and should always contain '-' for compatibility with older versions of **zic**. It was previously known as the *TYPE* field, which could contain values to allow a separate script to further restrict in which "types" of years the rule would apply.
- 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 include:
- 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 (breaking 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 local 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 indicator, 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 Ireland, 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, except

without suffix letters; 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, alternatively, a field in the same format as a rule-line *SAVE* field, giving the amount of time to be added to local standard time and whether the resulting time is standard or daylight saving. Standard time applies if this field is ‘’ or for timestamps occurring before any rule takes effect. 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’ shows where to put the time zone abbreviation’s variable part, which is taken from the *LETTER/S* field of the corresponding rule; any timestamps that precede the earliest rule use the *LETTER/S* of the earliest standard-time rule (which in this case must exist). Alternatively, a format can use the pair of characters ‘%z’ to stand for the UT offset in the form +- *hh*, +- *hhmm*, or +- *hhmms*, using the shortest form that does not lose information, 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 characters, ‘+’ and ‘-’. By convention, the time zone abbreviation ‘-00’ is a placeholder that means local time is unspecified.

**UNTIL** The time at which the UT offset or the rule(s) change for a location. 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. Continuation 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 standard time by default: that is, any of *L*’s timestamps preceding *L*’s earliest rule use the rule in effect after *L*’s first transition into standard 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 instant.

If a continuation line subtracts *N* seconds from the UT offset after a transition that would be interpreted to be later if using the continuation line’s UT offset and rules, the "until" time of the previous zone or

continuation line is interpreted according to the continuation line's UT offset and rules, and any rule that would otherwise take effect in the next *N* seconds is instead assumed to take effect simultaneously. For example:

```
# Rule  NAME  FROM  TO    -    IN    ON    AT    SAVE  LETTER/S
Rule    US    1967  2006  -    Oct   lastSun  2:00  0      S
Rule    US    1967  1973  -    Apr   lastSun  2:00  1:00  D
# Zone  NAME  STDOFF RULES  FORMAT  [UNTIL]
Zone    America/Menominee  -5:00  -    EST    1973 Apr 29 2:00
        -6:00  US    C%sT
```

Here, an incorrect reading would be there were two clock changes on 1973-04-29, the first from 02:00 EST (-05) to 01:00 CST (-06), and the second an hour later from 02:00 CST (-06) to 03:00 CDT (-05). However, **zic** interprets this more sensibly as a single transition from 02:00 CST (-05) to 02:00 CDT (-05).

A link line has the form

```
Link    TARGETLINK-NAME
```

For example:

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

The *TARGET* field should appear as the *NAME* field in some zone line or as the *LINK-NAME* field in some link 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. Links can chain together, although the behavior is unspecified if a chain of one or more links does not terminate in a Zone name. A link line can appear before the line that defines the link target. For example:

```
Link    Greenwich      G_M_T
Link    Etc/GMT Greenwich
Zone    Etc/GMT 0 - GMT
```

The two links are chained together, and G\_M\_T, Greenwich, and Etc/GMT all name the same zone.

Except for continuation lines, lines may appear in any order in the input. However, the behavior is unspecified if multiple zone or link lines define the same name.

The file that describes leap seconds can have leap lines and an expiration 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.

Rolling leap seconds would let one see Times Square ball drops where there'd be a "3... 2... 1... leap... Happy New Year" countdown, placing the leap second at midnight New York time rather than midnight UTC. Although stationary leap seconds are the common practice, rolling leap seconds can be useful in specialized applications like SMPTE timecodes that may prefer to put leap second discontinuities at the end of a local broadcast day. However, rolling leap seconds are not supported if the **-r** option is used.

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.

## EXTENDED EXAMPLE

Here is an extended example of **zic** input, intended to illustrate many of its features.

```
# Rule  NAME  FROM  TO    -   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 EU rules are for the European Union and for its predecessor organization, the European Communities. The timezone is named Europe/Zurich and it has the alias 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 offset was changed to 7<degree>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 completeness. 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, respectively. 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 compiled 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)