NAME

zdump − timezone dumper

SYNOPSIS

zdump [*option* ...] [*timezone* ...]

DESCRIPTION

The **zdump** program prints the current time in each *timezone* named on the command line.

OPTIONS

--version
Output version information and exit.

--help Output short usage message and exit.

-i Output a description of time intervals. For each *timezone* on the command line, output an interval-format description of the timezone. See “INTERVAL FORMAT” below.

-v Output a verbose description of time intervals. For each *timezone* on the command line, print the time at the lowest possible time value, the time one day after the lowest possible time value, the times both one second before and exactly at each detected time discontinuity, the time at one day less than the highest possible time value, and the time at the highest possible time value. Each line is followed by **isdst=***D* where *D* is positive, zero, or negative depending on whether the given time is daylight saving time, standard time, or an unknown time type, respectively. Each line is also followed by **gmtoff=***N* if the given local time is known to be *N* seconds east of Greenwich.

-V Like **-v**, except omit the times relative to the extreme time values. This generates output that is easier to compare to that of implementations with different time representations.

-c [*loyear***,**]*hiyear*
Cut off interval output at the given year(s). Cutoff times are computed using the proleptic Gregorian calendar with year 0 and with Universal Time (UT) ignoring leap seconds. Cutoffs are at the start of each year, where the lower-bound timestamp is exclusive and the upper is inclusive; for example, **-c 1970,2070** selects transitions after 1970-01-01 00:00:00 UTC and on or before 2070-01-01 00:00:00 UTC. The default cutoff is **-500,2500**.

-t [*lotime***,**]*hitime*
Cut off interval output at the given time(s), given in decimal seconds since 1970-01-01 00:00:00 Coordinated Universal Time (UTC). The *timezone* determines whether the count includes leap seconds. As with **-c**, the cutof f’s lower bound is exclusive and its upper bound is inclusive.

INTERVAL FORMAT

The interval format is a compact text representation that is intended to be both human- and machine-readable. It consists of an empty line, then a line “TZ=*string*” where *string* is a double-quoted string gi ving the timezone, a second line “**- - -** *interval*” describing the time interval before the first transition if any, and zero or more following lines “*date time interval*”, one line for each transition time and following interval. Fields are separated by single tabs.

Dates are in *yyyy*-*mm*-*dd* format and times are in 24-hour *hh*:*mm*:*ss* format where *hh*<24. Times are in local time immediately after the transition. A time interval description consists of a UT offset in signed ±*hhmmss* format, a time zone abbreviation, and an isdst flag. An abbreviation that equals the UT offset is omitted; other abbreviations are double-quoted strings unless they consist of one or more alphabetic characters. An isdst flag is omitted for standard time, and otherwise is a decimal integer that is unsigned and positive (typically 1) for daylight saving time and negative for unknown.

In times and in UT offsets with absolute value less than 100 hours, the seconds are omitted if they are zero, and the minutes are also omitted if they are also zero. Positive UT offsets are east of Greenwich. The UT offset **-00** denotes a UT placeholder in areas where the actual offset is unspecified; by convention, this occurs when the UT offset is zero and the time zone abbreviation begins with “**-**” or is “zzz”.

In double-quoted strings, escape sequences represent unusual characters. The escape sequences are \s for space, and \", \\, \f, \n, \r, \t, and \v with their usual meaning in the C programming language. E.g., the

double-quoted string “"CET\s\"\\"” represents the character sequence “CET "\”.

Here is an example of the output, with the leading empty line omitted. (This example is shown with tab stops set far enough apart so that the tabbed columns line up.)

```
TZ="Pacific/Honolulu"
-           -         -103126  LMT
1896-01-13  12:01:26  -1030    HST
1933-04-30  03        -0930    HDT  1
1933-05-21  11        -1030    HST
1942-02-09  03        -0930    HWT  1
1945-08-14  13:30     -0930    HPT  1
1945-09-30  01        -1030    HST
1947-06-08  02:30     -10      HST
```

Here, local time begins 10 hours, 31 minutes and 26 seconds west of UT, and is a standard time abbreviated LMT. Immediately after the first transition, the date is 1896-01-13 and the time is 12:01:26, and the following time interval is 10.5 hours west of UT, a standard time abbreviated HST. Immediately after the second transition, the date is 1933-04-30 and the time is 03:00:00 and the following time interval is 9.5 hours west of UT, is abbreviated HDT, and is daylight saving time. Immediately after the last transition the date is 1947-06-08 and the time is 02:30:00, and the following time interval is 10 hours west of UT, a standard time abbreviated HST.

Here are excerpts from another example:

```
TZ="Europe/Astrakhan"
-           -         +031212  LMT
1924-04-30  23:47:48  +03
1930-06-21  01        +04
1981-04-01  01        +05           1
1981-09-30  23        +04
...
2014-10-26  01        +03
2016-03-27  03        +04
```

This time zone is east of UT, so its UT offsets are positive. Also, many of its time zone abbreviations are omitted since they duplicate the text of the UT offset.

LIMITATIONS

Time discontinuities are found by sampling the results returned by localtime at twelve-hour intervals. This works in all real-world cases; one can construct artificial time zones for which this fails.

In the **-v** and **-V** output, “UT” denotes the value returned by **gmtime**(3), which uses UTC for modern timestamps and some other UT flavor for timestamps that predate the introduction of UTC. No attempt is currently made to have the output use “UTC” for newer and “UT” for older timestamps, partly because the exact date of the introduction of UTC is problematic.

SEE ALSO

tzfile(5), **zic**(8)