

**NAME**

date – print or set the system date and time

**SYNOPSIS**

**date** [*OPTION*]... [+*FORMAT*]

**date** [-u|--utc|--universal] [*MMDDhhmm*[[*CC*]*YY*][.ss]]

**DESCRIPTION**

Display the current time in the given *FORMAT*, or set the system date.

Mandatory arguments to long options are mandatory for short options too.

**-d, --date=STRING**

display time described by *STRING*, not 'now'

**--debug**

annotate the parsed date, and warn about questionable usage to stderr

**-f, --file=DATEFILE**

like **--date**; once for each line of *DATEFILE*

**-I[FMT], --iso-8601[=*FMT*]**

output date/time in ISO 8601 format. *FMT*='date' for date only (the default), 'hours', 'minutes', 'seconds', or 'ns' for date and time to the indicated precision. Example: 2006-08-14T02:34:56-06:00

**-R, --rfc-email**

output date and time in RFC 5322 format. Example: Mon, 14 Aug 2006 02:34:56 -0600

**--rfc-3339=*FMT***

output date/time in RFC 3339 format. *FMT*='date', 'seconds', or 'ns' for date and time to the indicated precision. Example: 2006-08-14 02:34:56-06:00

**-r, --reference=FILE**

display the last modification time of *FILE*

**-s, --set=STRING**

set time described by *STRING*

**-u, --utc, --universal**

print or set Coordinated Universal Time (UTC)

**--help** display this help and exit

**--version**

output version information and exit

*FORMAT* controls the output. Interpreted sequences are:

%%	a literal %
%a	locale's abbreviated weekday name (e.g., Sun)
%A	locale's full weekday name (e.g., Sunday)
%b	locale's abbreviated month name (e.g., Jan)
%B	locale's full month name (e.g., January)
%c	locale's date and time (e.g., Thu Mar 3 23:05:25 2005)
%C	century; like %Y, except omit last two digits (e.g., 20)
%d	day of month (e.g., 01)
%D	date; same as %m/%d/%y
%e	day of month, space padded; same as %_d

%F	full date; same as %Y-%m-%d
%g	last two digits of year of ISO week number (see %G)
%G	year of ISO week number (see %V); normally useful only with %V
%h	same as %b
%H	hour (00..23)
%I	hour (01..12)
%j	day of year (001..366)
%k	hour, space padded ( 0..23); same as %_H
%l	hour, space padded ( 1..12); same as %_I
%m	month (01..12)
%M	minute (00..59)
%n	a newline
%N	nanoseconds (000000000..999999999)
%p	locale's equivalent of either AM or PM; blank if not known
%P	like %p, but lower case
%q	quarter of year (1..4)
%r	locale's 12-hour clock time (e.g., 11:11:04 PM)
%R	24-hour hour and minute; same as %H:%M
%s	seconds since 1970-01-01 00:00:00 UTC
%S	second (00..60)
%t	a tab
%T	time; same as %H:%M:%S
%u	day of week (1..7); 1 is Monday
%U	week number of year, with Sunday as first day of week (00..53)
%V	ISO week number, with Monday as first day of week (01..53)
%w	day of week (0..6); 0 is Sunday
%W	week number of year, with Monday as first day of week (00..53)
%x	locale's date representation (e.g., 12/31/99)
%X	locale's time representation (e.g., 23:13:48)
%y	last two digits of year (00..99)
%Y	year
%Z	+hhmm numeric time zone (e.g., -0400)
%.Z	+hh:mm numeric time zone (e.g., -04:00)
%::Z	+hh:mm:ss numeric time zone (e.g., -04:00:00)
%:::Z	numeric time zone with : to necessary precision (e.g., -04, +05:30)
%Z	alphabetic time zone abbreviation (e.g., EDT)

By default, date pads numeric fields with zeroes. The following optional flags may follow '%':

– (hyphen) do not pad the field

- \_ (underscore) pad with spaces
- 0 (zero) pad with zeros
- ^ use upper case if possible
- # use opposite case if possible

After any flags comes an optional field width, as a decimal number; then an optional modifier, which is either E to use the locale's alternate representations if available, or O to use the locale's alternate numeric symbols if available.

## EXAMPLES

Convert seconds since the epoch (1970-01-01 UTC) to a date

```
$ date --date='@2147483647'
```

Show the time on the west coast of the US (use tzselect(1) to find TZ)

```
$ TZ='America/Los_Angeles' date
```

Show the local time for 9AM next Friday on the west coast of the US

```
$ date --date='TZ="America/Los_Angeles" 09:00 next Fri'
```

## DATE STRING

The `--date=STRING` is a mostly free format human readable date string such as "Sun, 29 Feb 2004 16:21:42 -0800" or "2004-02-29 16:21:42" or even "next Thursday". A date string may contain items indicating calendar date, time of day, time zone, day of week, relative time, relative date, and numbers. An empty string indicates the beginning of the day. The date string format is more complex than is easily documented here but is fully described in the info documentation.

## AUTHOR

Written by David MacKenzie.

## REPORTING BUGS

GNU coreutils online help: <<https://www.gnu.org/software/coreutils/>>

Report date translation bugs to <<https://translationproject.org/team/>>

## COPYRIGHT

Copyright © 2018 Free Software Foundation, Inc. License GPLv3+: GNU GPL version 3 or later <<https://gnu.org/licenses/gpl.html>>.

This is free software: you are free to change and redistribute it. There is NO WARRANTY, to the extent permitted by law.

## SEE ALSO

Full documentation at: <<https://www.gnu.org/software/coreutils/date>>  
or available locally via: `info '(coreutils) date invocation'`