Options for `date`

The program accepts the following options. Also see Common options.

‘`-d` ***`datestr`***’

‘`--date=`***`datestr`***’

Display the date and time specified in *datestr* instead of the current date and time. *datestr* can be in almost any common format. It can contain month names, time zones, ‘`am`’ and ‘`pm`’, ‘`yesterday`’, etc. For example, `--date="2004-02-27 14:19:13.489392193 +0530"` specifies the instant of time that is 489,392,193 nanoseconds after February 27, 2004 at 2:19:13 PM in a time zone that is 5 hours and 30 minutes east of UTC.
Note: input currently must be in locale independent format. E.g., the LC_TIME=C below is needed to print back the correct date in many locales:

```
date -d "$(LC_TIME=C date)"
```

See Date input formats.

‘**`--debug`**’

Annotate the parsed date, display the effective time zone, and warn about potential misuse.

‘`-f` ***`datefile`***’

‘`--file=`***`datefile`***’

Parse each line in *datefile* as with `-d` and display the resulting date and time. If *datefile* is ‘`-`’, use standard input. This is useful when you have many dates to process, because the system overhead of starting up the `date` executable many times can be considerable.

‘`-I[`*`timespec`*`]`’

‘`--iso-8601[=`*`timespec`*`]`’

Display the date using an ISO 8601 format, ‘`%Y-%m-%d`’.

The argument *timespec* specifies the number of additional terms of the time to include. It can be one of the following:

‘`auto`’

Print just the date. This is the default if *timespec* is omitted.

‘`hours`’

Append the hour of the day to the date.

‘`minutes`’

Append the hours and minutes.

‘`seconds`’

Append the hours, minutes and seconds.

‘`ns`’

Append the hours, minutes, seconds and nanoseconds.

If showing any time terms, then include the time zone using the format ‘`%:z`’. This format is always suitable as input for the `--`

`date` (`-d`) and `--file` (`-f`) options, regardless of the current locale.

‘`-r` *`file`*’

‘`--reference=`*`file`*’

Display the date and time of the last modification of *file*, instead of the current date and time.

‘`-R`’

‘`--rfc-email`’

Display the date and time using the format ‘`%a, %d %b %Y %H:%M:%S %z`’, evaluated in the C locale so abbreviations are always in English. For example:

```
Fri, 09 Sep 2005 13:51:39 -0700
```

This format conforms to Internet RFCs 5322, 2822 and 822, the current and previous standards for Internet email. For compatibility with older versions of `date`, `--rfc-2822` and `--rfc-822` are aliases for `--rfc-email`.

‘`--rfc-3339=`*`timespec`*’

Display the date using a format specified by Internet RFC 3339. This is like `--iso-8601`, except that a space rather than a ‘`T`’ separates dates from times. This format is always suitable as input for the `--date` (`-d`) and `--file` (`-f`) options, regardless of the current locale.

The argument *timespec* specifies how much of the time to include. It can be one of the following:

‘`date`’

Print just the full-date, e.g., ‘`2005-09-14`’. This is equivalent to the format ‘`%Y-%m-%d`’.

‘`seconds`’

Print the full-date and full-time separated by a space, e.g., ‘`2005-09-14 00:56:06+05:30`’. The output ends with a numeric time-offset; here the ‘`+05:30`’ means that local time is five hours and thirty minutes east of UTC. This is equivalent to the format ‘`%Y-%m-%d %H:%M:%S%:z`’.

‘`ns`’

Like ‘`seconds`’, but also print nanoseconds, e.g., ‘`2005-09-14 00:56:06.998458565+05:30`’. This is equivalent to the format ‘`%Y-%m-%d %H:%M:%S.%N%:z`’.

‘`-s` *`datestr`*’

‘`--set=`*`datestr`*’

Set the date and time to *datestr*. See `-d` above. See also Setting the time.

‘`-u`’

‘`--utc`’

‘`--universal`’

Use Universal Time by operating as if the `TZ` environment variable were set to the string ‘`UTC0`’. UTC stands for Coordinated Universal Time, established in 1960. Universal Time is often called “Greenwich Mean Time” (GMT) for historical reasons. Typically, systems ignore leap seconds and thus implement an approximation to UTC rather than true UTC.