|  |  |
| --- | --- |
| timezones {base} | R Documentation |

Time Zones

**Description**

Information about time zones in **R**. Sys.timezone returns the name of the current time zone.

**Usage**

Sys.timezone(location = TRUE)

OlsonNames(tzdir = NULL)

**Arguments**

|  |  |
| --- | --- |
| location | logical: defunct: ignored, with a warning for false values. |
| tzdir | The time-zone database to be used: the default is to try known locations until one is found. |

**Details**

Time zones are a system-specific topic, but these days almost all **R** platforms use similar underlying code, used by Linux, macOS, Solaris, AIX and FreeBSD, and installed with **R** on Windows. (Unfortunately there are many system-specific errors in the implementations.) It is possible to use the **R** sources' version of the code on Unix-alikes as well as on Windows: this is the default for macOS and recommended for Solaris.

It should be possible to set the current time zone via the environment variable TZ: see the section on ‘Time zone names’ for suitable values. Sys.timezone() will return the value of TZ if set initially (and on some OSes it is always set), otherwise it will try to retrieve from the OS a value which if set for TZ would give the initial time zone. (‘Initially’ means before any time-zone functions are used: if TZ is being set to override the OS setting or if the ‘try’ does not get this right, it should be set before the **R** process is started or (probably early enough) in file [.Rprofile](http://127.0.0.1:38355/help/library/base/help/.Rprofile)).

If TZ is set but invalid, most platforms default to UTC, the time zone colloquially known as GMT(see <https://en.wikipedia.org/wiki/Coordinated_Universal_Time>). (Some but not all platforms will give a warning for invalid values.) If it is unset or empty the *system* time zone is used (the one returned by Sys.timezone).

Time zones did not come into use until the second half of the nineteenth century and were not widely adopted until the twentieth, and *daylight saving time* (DST, also known as *summer time*) was first introduced in the early twentieth century, most widely in 1916. Over the last 100 years places have changed their affiliation between major time zones, have opted out of (or in to) DST in various years or adopted DST rule changes late or not at all. (The UK experimented with DST throughout 1971, only.)

A quite common system implementation of POSIXct is as signed 32-bit integers and so only goes back to the end of 1901: on such systems **R** assumes that dates prior to that are in the same time zone as they were in 1902. Most of the world had not adopted time zones by 1902 (so used local ‘mean time’ based on longitude) but for a few places there had been time-zone changes before then. 64-bit representations are becoming common; unfortunately on some 64-bit OSes (notably macOS) the database information is 32-bit and so only available for the range 1901–2038, and incompletely for the end years.

As from **R** 3.5.0, when a time zone location is first found in a session, its value is cached in object .sys.timezone in the base environment.

**Value**

Sys.timezone returns an OS-specific character string, possibly NA or an empty string (which on some OSes means UTC). This will be a location such as "Europe/London" if one can be ascertained.

A time zone region may be known by several names: for example "Europe/London" is also known as GB, GB-Eire, Europe/Belfast, Europe/Guernsey, Europe/Isle\_of\_Man and Europe/Jersey. A few regions are also known by a summary of their time zone, e.g. PST8PDTis an alias for America/Los\_Angeles.

OlsonNames returns a character vector, see the examples for typical cases. It may have an attribute "Version", something like "2017c".

**Time zone names**

Names "UTC" and its synonym "GMT" are accepted on all platforms.

Where OSes describe their valid time zones can be obscure. The help for the C function tzsetcan be helpful, but it can also be inaccurate. There is a cumbersome POSIX specification (listed under environment variable TZ at<http://pubs.opengroup.org/onlinepubs/9699919799/basedefs/V1_chap08.html#tag_08>), which is often at least partially supported, but there are other more user-friendly ways to specify time zones.

Almost all **R** platforms make use of a time-zone database originally compiled by Arthur David Olson and now managed by IANA, in which the preferred way to refer to a time zone is by a location (typically of a city), e.g., Europe/London, America/Los\_Angeles, Pacific/Easterwithin a ‘time zone region’. Some traditional designations are also allowed such as EST5EDT orGB. (Beware that some of these designations may not be what you expect: in particular EST is a time zone used in Canada *without* daylight saving time, and not EST5EDT nor (Australian) Eastern Standard Time.) The designation can also be an optional colon prepended to the path to a file giving complied zone information (and the examples above are all files in a system-specific location). See <http://www.twinsun.com/tz/tz-link.htm> for more details and references. By convention, regions with a unique time-zone history since 1970 have specific names in the database, but those with different earlier histories may not. Each time zone has one or two (the second for DST) *abbreviations* used when formatting times.

The abbreviations used have changed over the years: for example France used PMT (‘Paris Mean Time’) from 1891 to 1911 then WET/WEST up to 1940 and CET/CEST from 1946. (In almost all time zones the abbreviations have been stable since 1970.) The POSIX standard allows only one or two abbreviations per time zone, so you may see the current abbreviation(s) used for older times.

For some time zones abbreviations are like -03 and +0845: this is done when there is no official abbreviation. (Negative values are behind (West of) UTC, as for the "%z" format for [strftime](http://127.0.0.1:38355/help/library/base/help/strftime).)

The function OlsonNames returns the time-zone names known to the currently selected Olson/IANA database. The system-specific location in the file system varies, e.g. ‘/usr/share/zoneinfo’ (Linux, macOS, FreeBSD), ‘/usr/share/lib/zoneinfo’ (Solaris, AIX), .... It is likely that there is a file named something like ‘zone.tab’ under that directory listing the locations known as time-zone names (but not for example EST5EDT). See also<https://en.wikipedia.org/wiki/Zone.tab>.

Where **R** was configured with option --with-internal-tzcode (the default on macOS and Windows: recommended on Solaris), the database at file.path(R.home("share"), "zoneinfo") is used by default: file ‘VERSION’ in that directory states the version. Environment variable TZDIR can be used to point to a different ‘zoneinfo’ database: this is also supported by the native services on some OSes, e.g. Linux using glibc except in secure modes.

Time zones given by name (*via* environment variable TZ, in tz arguments to functions such as [as.POSIXlt](http://127.0.0.1:38355/help/library/base/help/as.POSIXlt) and perhaps the system time zone) are loaded from the currently selected ‘zoneinfo’ database.

An attempt is made (once only per session) to map Windows' idea of the current time zone to a location, following a version of<http://unicode.org/repos/cldr/trunk/common/supplemental/windowsZones.xml> with additional values deduced from the Windows Registry and documentation. It can be overridden by setting the TZ environment variable before any date-times are used in the session.

Most platforms support time zones of the form Etc/GMT+n and Etc/GMT-n (possibly also without prefix Etc/), which assume a fixed offset from UTC (hence no DST). Contrary to some expectations (but consistent with names such as PST8PDT), negative offsets are times ahead of (east of) UTC, positive offsets are times behind (west of) UTC.

Immediately prior to the advent of legislated time zones, most people used time based on their longitude (or that of a nearby town), known as ‘Local Mean Time’ and abbreviated as LMT in the databases: in many countries that was codified with a specific name before the switch to a standard time. For example, Paris codified its LMT as ‘Paris Mean Time’ in 1891 (to be used throughout mainland France) and switched to GMT+0 in 1911.

Some systems (notably Linux) have a tzselect command which allows the interactive selection of a supported time zone name. On systems using systemd (notably Linux), the OS commandtimedatectl list-timezones will list all available time zone names.

**Warning**

There is a system-specific upper limit on the number of bytes in (abbreviated) time-zone names which can be as low as 6 (as required by POSIX). Some OSes allow the setting of time zones with names which exceed their limit, and that can crash the **R** session.

OlsonNames tries to find an Olson database in known locations. It might not succeed (when it returns an empty vector with a warning) and even if it does it might not locate the database used by the date-time code linked into **R**. Fortunately names are added rarely and most databases are pretty complete.

**Note**

Since 2007 there has been considerable disruption over changes to the timings of the DST transitions, aimed at energy conservation. These often have short notice and time-zone databases may not be up to date. (Morocco in 2013 announced a change to the end of DST at *a days* notice, and in 2015 North Korea gave imprecise information about a change a week in advance.)

On platforms with case-insensitive file systems, time zone names will be case-insensitive. They may or may not be on other platforms and so, for example, "gmt" is valid on some platforms and not on others.

Note that except where replaced, the operation of time zones is an OS service, and even where replaced a third-party database is used and can be updated (see the section on ‘Time zone names’). Incorrect results will never be an **R** issue, so please ensure that you have the courtesy not to blame **R** for them.

**See Also**

[Sys.time](http://127.0.0.1:38355/help/library/base/help/Sys.time), [as.POSIXlt](http://127.0.0.1:38355/help/library/base/help/as.POSIXlt).

<https://en.wikipedia.org/wiki/Time_zone> and <http://www.twinsun.com/tz/tz-link.htm> for extensive sets of links.

<https://data.iana.org/time-zones/theory.html> for the ‘rules’ of the Olson/IANA database.

**Examples**

Sys.timezone()

str(OlsonNames()) ## typically close to 600 hundred names,

## typically some acronyms/aliases such as "UTC", "NZ", "MET", "Eire", ..., but

## mostly pairs (and triplets) such as "Pacific/Auckland"

table(sl <- grepl("/", OlsonNames()))

OlsonNames()[ !sl ] # the simple ones

head(Osl <- strsplit(OlsonNames()[sl], "/"))

(tOS1 <- table(vapply(Osl, `[[`, "", 1))) # Continents, countries, ...

table(lengths(Osl))# most are pairs, some triplets

str(Osl[lengths(Osl) >= 3])# "America" South and North ...

[Package *base* version 3.5.0 [Index](http://127.0.0.1:38355/help/library/base/html/00Index.html)]