timezones: 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

'try' does not get this right, it should be set before the R process is started or (probably early enough) in file `.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. PST8PDT is 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

cumbersome POSIX specification (listed under environment variable TZ at http://pubs.opengroup.org/onlinepubs/9699919799/basedefs/V1_chap08.html#tag_ 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/Easter` within a 'time zone region'. Some traditional designations are also allowed such as `EST5EDT` or `GB`. (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 https://data.iana.org/time-zones/tz-link.html 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`.)

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

(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 command `timedatectl 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.

How the system time zone is found

This section is of background interest for users of a Unix-alike, but may help if an `NA` value is returned unexpectedly.

Commercial Unixen such as Solaris and AIX set TZ, so the value when `R` is started is used.

All other common platforms (Linux, macOS, *BSD) use similar schemes, either derived from `tzcode` (currently distributed from https://www.iana.org/time-zones) or independently coded (`glibc` , `musl-libc`). Such systems read the time-zone information from a file 'localtime', usually under '/etc' (but possibly under '/usr/local/etc' or '/usr/local/etc/zoneinfo'). As the usual Linux manual page for `localtime` says

'Because the time zone identifier is extracted from the symlink target name of '/etc/localtime', this file may not be a normal file or hardlink.'

Nevertheless, some Linux distributions (including the one from which that quote was taken) or sysadmins have chosen to copy a time-zone file to

Some Linux platforms provide two other mechanisms which are tried in turn before looking at '/etc/localtime'.

- 'Modern' Linux systems use `systemd` which provides mechanisms to set and retrieve the time zone (amongst other things). There is a command `timedatectl` to give details. (Unfortunately RHEL/Centos 6.x are not 'modern'.)
- Debian-derived systems since *ca* 2007 have supplied a file '/etc/timezone'. Its format is undocumented but it empirically it contains a single line of text naming the time zone.

In each case a sanity check is performed that the time-zone name is the name of a file in the time-zone database. (The systems probably use the time-zone file (symlinked to) '/etc/localtime', but the `Sys.timezone` code does not check that is the same as the named file in the database. This is deliberate as they may be from different dates.)

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`, `as.POSIXlt`.

https://en.wikipedia.org/wiki/Time_zone and https://data.iana.org/time-zones/tz-link.html for extensive sets of links.

https://data.iana.org/time-zones/theory.html for the 'rules' of the Olson/IANA database.

Examples

```
str(OlsonNames()) ## typically close to 600 hundred names,
## typically some acronyms/aliases such as "UTC", "NZ", "MET", "
## 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, countrie
table(lengths(Osl))# most are pairs, some triplets
str(Osl[lengths(Osl) >= 3])# "America" South and North ...
```

Functions

double: Double-Precision Vectors

dput: Write an Object to a File or Recreate it

drop: Drop Redundant Extent Information

droplevels: Drop Unused Levels from Factors

dump: Text Representations of R Objects

duplicated: Determine Duplicate Elements

dynload: Foreign Function Interface

eapply: Apply a Function Over Values in an Environment

eigen: Spectral Decomposition of a Matrix

encodeString: Encode Character Vector as for Printing

Encoding: Read or Set the Declared Encodings for a Character Vector

environment: Environment Access

EnvVar: Environment Variables

eval: Evaluate an (Unevaluated) Expression

exists: Is an Object Defined?

expression: Unevaluated Expressions

Extract: Extract or Replace Parts of an Object

Extract.data.frame: Extract or Replace Parts of a Data Frame

Extract.factor: Extract or Replace Parts of a Factor

Extremes: Maxima and Minima

extSoftVersion: Report Versions of Third-Party Software

factor: Factors

file.access: Ascertain File Accessibility

file.choose: Choose a File Interactively

file.info: Extract File Information

file.path: Construct Path to File

files: File Manipulation

files2: Manipulation of Directories and File Permissions

file.show: Display One or More Text Files

findInterval: Find Interval Numbers or Indices

find.package: Find Packages

force: Force Evaluation of

getLoadedDLLs: Get DLLs Loaded in Current Session

getNativeSymbolInfo: Obtain a Description of one or more Native (C/Fortran)...

gettext: Translate Text Messages

getwd: Get or Set Working Directory

gl: Generate Factor Levels

grep: Pattern Matching and Replacement

grepRaw: Pattern Matching for Raw Vectors

groupGeneric: S3 Group Generic Functions

grouping: Grouping Permutation

gzcon: (De)compress I/O Through Connections

hexmode: Display Numbers in Hexadecimal

Hyperbolic: Hyperbolic Functions

iconv: Convert Character Vector between Encodings

icuSetCollate: Setup Collation by ICU

identical: Test Objects for Exact Equality

identity: Identity Function

ifelse: Conditional Element Selection

interaction: Compute Factor Interactions

interactive: Is R Running Interactively?

Internal: Call an Internal Function

InternalMethods: Internal Generic Functions

invisible: Change the Print Mode to Invisible

is.finite: Finite, Infinite and NaN Numbers

is.function: Is an Object of Type (Primitive) Function?

is.language: Is an Object a Language Object?

is.object: Is an Object 'internally classed'?

ISOdatetime: Date-time Conversion Functions from Numeric Representations

isR: Are we using R, rather than S?

is.recursive: Is an Object Atomic or Recursive?

isS4: Test for an S4 object

is.single: Is an Object of Single Precision Type?

isSymmetric: Test if a Matrix or other Object is Symmetric (Hermitian)

is.unsorted: Test if an Object is Not Sorted

jitter: 'jitter' (Add Noise) to

Number of a Matrix

kronecker: Kronecker Products on Arrays

l10n_info: Localization Information

labels: Find Labels from Object

La_library: LAPACK Library

lapply: Apply a Function over a List or Vector

Last.value: Value of Last Evaluated Expression

La_version: LAPACK Version

lazyload: Lazy Load a Database of R Objects

length: Length of an Object

lengths: Lengths of List or Vector Elements

levels: Levels Attributes

libcurlVersion: Report Version of libcurl

libPaths: Search Paths for Packages

library: Loading/Attaching and Listing of Packages

library.dynam: Loading DLLs from Packages

license: The R License Terms

list: Lists - Generic and

Environment

list.files: List the Files in a Directory/Folder

load: Reload Saved Datasets

locales: Query or Set Aspects of the Locale

Log: Logarithms and Exponentials

Logic: Logical Operators

logical: Logical Vectors

LongVectors: Long Vectors

lower.tri: Lower and Upper Triangular Part of a Matrix

ls: List Objects

make.names: Make Syntactically Valid Names

make.unique: Make Character Strings Unique

mapply: Apply a Function to Multiple List or Vector Arguments

margin.table: Compute table margin

match: Value Matching

match.arg: Argument Verification Using Partial Matching

match.call: Argument Matching

match.fun: Extract a Function Specified by

nchar: Count the Number of Characters (or Bytes or Width)

nlevels: The Number of Levels of a Factor

noquote: Class for 'no quote' Printing of Character Strings

norm: Compute the Norm of a Matrix

normalizePath: Express File Paths in Canonical Form

notyet: Not Yet Implemented Functions and Unused Arguments

nrow: The Number of Rows/Columns of an Array

ns-dblcolon: Double Colon and Triple Colon Operators

ns-hooks: Hooks for Namespace Events

ns-internal: Namespace Internals

ns-load: Loading and Unloading Name Spaces

ns-reflect: Namespace Reflection Support

ns-topenv: Top Level Environment

NULL: The Null Object

numeric: Numeric Vectors

NumericConstants: Numeric Constants

octmode: Display Numbers in Octal

on.exit: Function Exit Code

Ops.Date: Operators on the Date Class

options: Options Settings

order: Ordering Permutation

outer: Outer Product of Arrays

Paren: Parentheses and Braces

parse: Parse R Expressions

paste: Concatenate Strings

path.expand: Expand File Paths

pcre_config: Report Configuration Options for PCRE

Platform: Platform Specific Variables

pmatch: Partial String Matching

polyroot: Find Zeros of a Real or Complex Polynomial

pos.to.env: Convert Positions in the Search Path to Environments

pretty: Pretty Breakpoints

print.dataframe: Printing Data Frames

print.default: Default Printing

prmatrix: Print Matrices, Old-style

proc.time: Running Time of R

prod: Product of Vector Elements

prop.table: Express Table Entries as Fraction of Marginal Table

pushBack: Push Text Back on to a Connection

qr: The QR Decomposition of a Matrix

qraux: Reconstruct the Q, R, or X Matrices from a QR Object

quit: Terminate an R Session

Quotes: Quotes

Random: Random Number Generation

Random-user: User-supplied Random Number Generation

range: Range of Values

rank: Sample Ranks

rapply: Recursively Apply a Function to a List

raw: Raw Vectors

rawConversion: Convert to or from Raw Vectors

RdUtils: Utilities for Processing Rd Files

readBin: Transfer Binary Data To and From Connections

readChar: Transfer Character Strings To and From Connections

readline: Read a Line from the Terminal

readLines: Read Text Lines from a Connection

readRDS: Serialization Interface for Single Objects

readRenviron: Set Environment Variables from a File

Recall: Recursive Calling

regex: Regular Expressions as used in R

reg.finalizer: Finalization of Objects

regmatches: Extract or Replace Matched Substrings

rep: Replicate Elements of Vectors and Lists

replace: Replace Values in a Vector

Reserved: Reserved Words in R

rev: Reverse Elements

rle: Run Length Encoding

rm: Remove Objects from a Specified Environment

Round: Rounding of Numbers

round.POSIXt: Round / Truncate Data-Time Objects

row: Row Indexes

row.names: Get and Set Row Names for Data Frames

rowsum: Give Column Sums of a Matrix or Data Frame, Based on a...

sample: Random Samples and Permutations

save: Save R Objects

scale: Scaling and Centering of Matrix-like Objects

scan: Read Data Values

search: Give Search Path for R Objects

seek: Functions to Reposition Connections

seq: Sequence Generation

seq.Date: Generate Regular Sequences of Dates

seq.POSIXt: Generate Regular Sequences of Times

Serialization Interface

sets: Set Operations

setTimeLimit: Set CPU and/or Elapsed Time Limits

showConnections: Display Connections

shQuote: Quote Strings for Use in OS Shells

sign: Sign Function

sink: Send R Output to a File

slice.index: Slice Indexes in an Array

slotOp: Extract or Replace A Slot

socketSelect: Wait on Socket Connections

solve: Solve a System of Equations

sort: Sorting or Ordering Vectors

source: Read R Code from a File, a Connection or Expressions

Special: Special Functions of Mathematics

split: Divide into Groups and Reassemble

sprintf: Use C-style String Formatting Commands

sQuote: Quote Text

srcfile: References to Source Files and Code

Dispatching S4 Methods

startsWith: Does String Start or End With Another String?

Startup: Initialization at Start of an R Session

stop: Stop Function Execution

stopifnot: Ensure the Truth of R Expressions

strptime: Date-time Conversion Functions to and from Character

strrep: Repeat the Elements of a Character Vector

strsplit: Split the Elements of a Character Vector

strtoi: Convert Strings to Integers

strtrim: Trim Character Strings to Specified Display Widths

structure: Attribute Specification

strwrap: Wrap Character Strings to Format Paragraphs

subset: Subsetting Vectors, Matrices and Data Frames

substitute: Substituting and Quoting Expressions

substr: Substrings of a Character Vector

sum: Sum of Vector Elements

svd: Singular Value Decomposition of a Matrix

sweep: Sweep out Array Summaries

switch: Select One of a List of Alternatives

Syntax: Operator Syntax and Precedence

Sys.getenv: Get Environment Variables

Sys.getpid: Get the Process ID of the R Session

Sys.glob: Wildcard Expansion on File Paths

Sys.info: Extract System and User Information

Sys.localeconv: Find Details of the Numerical and Monetary Representations in...

sys.parent: Functions to Access the Function Call Stack

Sys.readlink: Read File Symbolic Links

Sys.setenv: Set or Unset Environment Variables

Sys.setFileTime: Set File Time

Sys.sleep: Suspend Execution for a Time Interval

sys.source: Parse and Evaluate Expressions from a File

trace: Interactive Tracing and Debugging of Calls to a Function or...

traceback: Get and Print Call Stacks

tracemem: Trace Copying of Objects

transform: Transform an Object, for Example a Data Frame

Trig: Trigonometric Functions

trimws: Remove Leading/Trailing Whitespace

try: Try an Expression Allowing Error Recovery

typeof: The Type of an Object

unique: Extract Unique Elements

unix/Signals: Interrupting Execution of R

unlink: Delete Files and Directories

unlist: Flatten Lists

unname: Remove 'names' or 'dimnames'

UseMethod: Class Methods

userhooks: Functions to Get and Set Hooks for Load, Attach, Detach and...

utf8Conversion: Convert Integer Vectors to or from UTF-8 encoded

Encoded

vector: Vectors

Vectorize: Vectorize a Scalar Function

Version: Version Information

warning: Warning Messages

warnings: Print Warning Messages

weekday.POSIXt: Extract Parts of a POSIXt or Date Object

which: Which indices are TRUE?

which.min: Where is the Min() or Max() or first TRUE or FALSE ?

windows/shell: Invoke a System Command, using a Shell

windows/shell.exec: Open a File or URL using Windows File Associations

with: Evaluate an Expression in a Data Environment

withVisible: Return both a Value and its Visibility

write: Write Data to a File

writeLines: Write Lines to a Connection

xtfrm: Auxiliary Function for Sorting and Ranking

zapsmall: Rounding of

Characteristics of the Machine

zpackages: Listing of Packages

zScript: Scripting Language Interface

zutils: Miscellaneous Internal/Programming Utilities