Package 'icalutils'

May 27, 2024

Type Package

Title Read, Process and Create iCalendar Data and Files

Version 0.2-0

Date 2023-12-21

Maintainer Enrico Schumann <es@enricoschumann.net>

Description Utilities for handling iCalendar data, in particular reading and creating/writing iCalendar files. The package is written in pure R, with no dependencies (package 'base64enc' is only needed if calendar files have attachments). The package maps iCalendar dates and date-times to R's native 'Date' and 'POSIXct' classes. Recurrance rules for repeated items (such as birthdays) can be expanded into actual occurance sets. Timezones are supported.

License GPL-3

URL http://enricoschumann.net/R/packages/icalutils,
https://git.sr.ht/~enricoschumann/icalutils,
https://github.com/enricoschumann/icalutils

Suggests base64enc, tinytest

NeedsCompilation no

Author Enrico Schumann [aut, cre] (<https://orcid.org/0000-0001-7601-6576>), Unicode, Inc. [dtc, cph]

Contents

icalutils *Read and Create iCalendar Data*

Description

Read and create iCalendar data

Usage

```
read_icalendar(file, strict.eol = TRUE, use.OlsonNames = TRUE,
               uid.names = FALSE, keep.source = TRUE,
               components = c("VEVENT", "VTODO", "VJOURNAL",
                              "VFREEBUSY", "VTIMEZONE"),
               ...)

## S3 method for class 'icalendar'
as.data.frame(x, row.names = NULL, optional = FALSE,
              adjust.allday = TRUE,
              recur.expand = FALSE,
              recur.until = NULL,
              recur.count = NULL,
              use.OlsonNames = TRUE,
              all.timestamps.POSIXct = TRUE,
              all.timestamps.Date = FALSE,
              timestamps.tz = "",
              components = c("VEVENT", "VTODO"),
              ...)

rrule(dtstart, dtend, freq,
      until = NULL, count = NULL,
      interval = 1,
      bysecond = NULL, byminute = NULL, byhour = NULL,
      byday = NULL, bymonthday = NULL, byyearday = NULL,
      byweekno = NULL, bymonth = NULL, bysetpos = NULL,
      wkst = NULL, rdate = NULL, exdate = NULL, text = NULL)

vevent(dtstart, summary, dtend = NULL,
       all.day = inherits(dtstart, "Date"),
       description = NULL, uid = NULL, categories = NULL,
       ...,
       vcalendar = TRUE, file, fold = TRUE)

to_vevent(x, ...)

save_attachments(file, out.dir, strict.eol = TRUE)
```

Arguments

file	character. The file to be read (read_icalendar, save_attachments) or written (vevent).
out.dir	character: the directory where to store attachments
strict.eol	logical. RFC 5545 says that the end of a line should be signalled by CRLF, and when strict.eol is TRUE, such line ends are required. (This also means that when long lines are unfolded, the function looks for CRLF followed by a space character.) iCalendar files that use a simple LF may be read with strict.eol set to FALSE.
adjust.allday	logical. If TRUE, the end day is moved one day back. If an all-day event ends on a given day, its end is stored as the next day in the icalendar format.
recur.expand	logical
recur.until	a date or datetime
recur.count	an integer
use.OlsonNames	logical. Map timezone names in file to names returned by OlsonNames?
uid.names	logical: use UID values as names for the resulting list? Careful with Outlook: UIDs are often insanely long.
all.timestamps.POSIXct	logical. To recover the original Date, use as.Date(..., tz = "<tz>").
all.timestamps.Date	logical
timestamps.tz	character. Applies only if return.class is data.frame. This is used for local time: timestamps that have no timezone information attached. Default is "", which means to use the current timezone.
...	additional arguments
x	a list with class attribute vevent or, for to_event an object for which a coercion method exists
row.names	see as.data.frame
optional	see as.data.frame
dtstart	Date or POSIXct
dtend	Date or POSIXct
freq	a character string: SECONDLY, MINUTELY, HOURLY, DAILY, WEEKLY, MONTHLY, or YEARLY
until	Date or POSIXct
count	integer
interval	integer. Default is 1.
bysecond	a vector of integers between 0 and 60 (though 60 will be changed to 59, to comply with the POSIX standard)
byminute	a vector of integers
byhour	a vector of integers

byday	The weekdays (Monday, Tuesday, ...) on which an event takes place. Takes either a vector of two-letter weekday abbreviations (mo, tu, ...), or a list of two vectors.
bymonthday	a vector of integers
byyearday	a vector of integers
byweekno	a vector of integers
bymonth	a vector of integers
bysetpos	a vector of integers
wkst	character
rdate	Date or POSIXct
exdate	Date or POSIXct
text	the text of the RRULE as used in an iCalendar file
keep.source	logical
components	a character vector
summary	character
description	character
uid	character. If NULL, a UID is genereated as suggest at https://www.jwz.org/doc/mid.html .
all.day	logical
fold	logical. If TRUE (the default), long lines are wrapped as described in RFC 5545, section 3.1.
vcalendar	logical. If TRUE, the VEVENT component is wrapped into a VCALENDAR, i.e. is made into an iCalendar object
categories	a character vector of keywords

Details

read_icalendar reads a file and returns an object of class icalendar, which is a list of lists: each of these lists comprises one component (e.g. an event) of the icalendar.

The as.data.frame method returns a data.frame of a selection of properties of the events; see Value section below.

Terminology:

An iCalendar stream (typically, a file) consist of one or more iCalendar objects. Each iCalendar object has some properties (such as the version) and one or more components, such as events, TODOs or journal entries. (This implies that there cannot be empty iCalendar objects.) These components are again composed of properties, such as 'summary'. A property may have parameters: a 'summary', for instance, may have a parameter that tells its 'language'.

Dates and times:

iCalendar objects have three ways to describe timestamps: dates, datetimes with timezone information (which includes UTC times) and datetimes without timezone information. The functions in package **icalutils** map dates to Date and datetimes to POSIXct. For times without timezones, POSIXct with timezone UTC is used, and an attribute "localtime" with value TRUE is attached.

All-day events:

The iCalendar specification does not detail how to treat all-day events. Two implementations seem widespread: i) events on a single day define only `DTSTART`, but no end; ii) DTEND is set to the date after the final day of the event. For instance, an event that lasts from August, 1 to August, 3 has DTSTART of August, 1, and DTEND at August, 4 (as in August, 4, 00:00:00). If `adjust.allday` is `TRUE` (the default), the end will be set to August, 3 (i.e. DTEND becomes becomes inclusive).

Timezones:

Timezone names, which include specific names when daylight-saving time is in effect, are not portable and may vary between systems. Thus, the iCalender specification requires that timezone information (i.e. a mapping from a local time to a GMT offset) must be included in the file, as components of type `VTIMEZONE`.

In the current version of **icalutils**, information in such `VTIMEZONE` components is not used. Instead, **icalutils** uses R's (very good) timezone support (see `timezones`) and maps datetimes with a timezone parameter to the timezone with the same name in the Olson database. This clean and transparent approach will not work on Windows (of course not). Thus, Windows timezone names (such as the infamous *Romance time* for Paris) are mapped to Olson names via data provided by the Unicode consortium. See References.

Value

For `read_icalendar`, a `list` of lists: one list for every component of the iCalendar object. If coerced to a data.frame, the following columns are present. Additional columns may be added later; so columns should always be addressed by name, not by position.

`uid`	field value
`summary`	field value
`description`	field value, or `NA`
`location`	field value, or `NA`
`start`	field value
`end`	field value
`all.day`	logical
`recurring`	logical

Author(s)

Enrico Schumann

References

For the iCalendar standard see RFC 5545 `https://tools.ietf.org/html/rfc5545`. Additional RFCs, such as 7986, may become relevant in future versions.

The mappings from Windows timezone names to Olson names are taken from the Unicode Common Locale Data Repository (`http://cldr.unicode.org/`).

See Also

packages **ical** and **calendar**

Examples

```
    read_vevent("calendar.ics")

rrule(dtstart = as.Date("2019-01-01"),
      dtend =   as.Date("2019-01-05"),
      freq = "yearly", count = 5)
## $text
## [1] "FREQ=yearly;COUNT=5;INTERVAL=1"
##
## $recurrence_set
##      DTSTART      DTEND
## 1 2019-01-01 2019-01-05
## 2 2020-01-01 2020-01-05
## 3 2021-01-01 2021-01-05
## 4 2022-01-01 2022-01-05
## 5 2023-01-01 2023-01-05
```

Index