Package ‘calendar’

February 11, 2019

Title Create, Read, Write, and Work with 'iCalander' Files, Calendars and Scheduling Data

Version 0.0.1

Description Provides function to create, read, write, and work with 'iCalander' files (which typically have '.ics' or '.ical' extensions), and the scheduling data, calendars and timelines of people, organisations and other entities that they represent. 'iCalendar' is an open standard for exchanging calendar and scheduling information between users and computers, described at <https://icalendar.org/>.

Depends R (>= 3.4.0)

License Apache License (>= 2.0)

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Suggests testthat, knitr, rmarkdown

Imports methods, tibble, lubridate

VignetteBuilder knitr

NeedsCompilation no

Author Robin Lovelace [aut, cre] (<https://orcid.org/0000-0001-5679-6536>), Layik Hama [aut] (<https://orcid.org/0000-0003-1912-4890>)

Maintainer Robin Lovelace <rob@0x@gmail.com>

Repository CRAN

Date/Publication 2019-02-11 13:53:13 UTC

R topics documented:

calendar ......................................................... 2
formats ......................................................... 3
holidays ......................................................... 3
ical .............................................................. 4
ical_example .................................................... 4
Description

iCalendar is an open standard for "exchanging calendar and scheduling information between users and computers". Files adhering to this standard are save as .ics files.

Details

The ical package is for interacting with such files
**Convenient datetime formats**

**Description**

Convenient datetime formats

**Usage**

formats

**Format**

An object of class list of length 8.

**Examples**

```r
formats <- list(
"ddmmyy"="%d%m%y",
"ddmmyyy"="%d%m%Y",
"ddmmyyy hh="="%d%m%Y %H",
"ddmmyyy hhmm"="%d%m%Y %H%M",
"yyyy-mm-dd"="%Y-%m-%d",
"yyyy-mm-dd hh:mm"="%Y-%m-%d %H:%M",
"yyyy-mm-dd hh"="%Y-%m-%d %H",
"yy-mm-dd"="%y-%m-%d"
)
# usethis::use_data(datetime_formats)
```

**Example ics file on English and Welsh holidays**

**Description**

This file is documented at https://www.gov.uk/government/publications/open-standards-for-government/exchange-of-calendar-events

**Usage**

holidays

**Format**

An object of class character of length 1.
Examples

# dataset was stored as follows:
u = "https://www.gov.uk/bank-holidays/england-and-wales.ics"
# download.file(u, "inst/extdata/england-and-wales.ics")
# holidays = "holidays"

ical Create object of class ical

Description

Create object of class ical

Usage

ical(x, ic_attributes = NULL)

Arguments

x Lines read-in in from an iCal file
ic_attributes Calendar attributes, e.g. as provided by ic_attributes_vec().

Examples

# ical from .ics characters:
class(ical_example)
ic <- ical(ical_example)
attributes(ic)
class(ic)
# ical from data frame:
ic_df <- data.frame(ic)
ic2 <- ical(ic_df)
class(ic2)
attributes(ic2)

ical_example Minimal example of raw ical data

Description

See https://calendar.google.com/calendar/ical/9sl1qu3qh2vdnq26bjgvtnos94

Usage

ical_example
ical_outlook

**Format**

An object of class character of length 22.

**Examples**

```r
# download.file("long_url", "inst/extdata/example.ics")
ical_example = readLines(system.file("extdata", "example.ics", package = "ical"))
# usethis::use_data(ical_example)
```

---

**ical_outlook Example of event data with multi-line description from Outlook**

---

**Description**

See here.

**Usage**

ical_outlook

**Format**

An object of class list of length 2.

**Examples**

```r
# ical_outlook_raw <- readLines(long_url)
# ical_outlook_list <- ic_list(ical_outlook_raw)
# ical_outlook <- ical_outlook_list[1:2]
# ical_outlook[[2]][c(1:38)] <- gsub("a|e|i|o|f|l|t|n|b", "a", ical_outlook[[2]][c(1:38)])
# ical_outlook[[2]] <- ical_outlook[[2]][c(1, 5, 35, 36:55)]
# usethis::use_data(ical_outlook)
```

---

**ic_attributes_vec Extract attributes from ical text**

---

**Description**

Extract attributes from ical text

**Usage**

```r
ic_attributes_vec(x = NULL, ic_attributes = c(BEGIN = "VCALENDAR",
   PRODID = "ATFutures/calendar", VERSION = "2.0", CALSCALE = "GREGORIAN",
   METHOD = "PUBLISH"))
```
Arguments

- **x**
  - Lines read-in in from an iCal file
- **ic_attributes**
  - Calendar attributes, e.g. as provided by `ic_attributes_vec()`.

Examples

```r
ic_attributes_vec() # default attributes (can be changed)
ic_attributes_vec(ical_example)
```

---

**ic_bind_list**

*Bind list of named vectors of variable length into data frame*

Description

Based on: https://stackoverflow.com/questions/17308551/

Usage

`ic_bind_list(x)`

Arguments

- **x**
  - list of named vectors

Examples

```r
ic_bind_list(list(ic_vector(ical_example)))
ic_file <- system.file("extdata", "england-and-wales.ics", package = "ical")
ic_raw = readLines(ic_file)
x <- lapply(ic_list(ic_raw), function(x) {
ic_vector(x)
})
ic_df <- ic_bind_list(x)
head(ic_df)
x <- lapply(ical_outlook, function(x) {
ic_vector(x)
})
ic_bind_list(x)
```
ic_character

Convert ical object to character strings of attributes

Description

Convert ical object to character strings of attributes

Usage

ic_character(ic)

Arguments

ic  object of class ical

Examples

ic <- ical(ical_example)
ic_character(ic)
identical(ical_example, ic_character(ic))

ic_char_datetime

Convert datetime object to character string

Description

Convert datetime object to character string

Usage

ic_char_datetime(x, zulu = FALSE)

Arguments

x  datetime object
zulu  is the datetime in Zulu time? FALSE by default, which means the calendar’s current timezone is used.

Examples

x <- ic_datetime("20180809T160000")
ic_char_datetime(x) == "20180809T160000"
x <- ic_datetime("20180809T160000Z")
ic_char_datetime(x, zulu = TRUE) == "20180809T160000Z"
ic_char_date(as.Date("1985-12-26"))
ic_char_event  
*Convert ical object to character strings of events*

**Description**
Convert ical object to character strings of events

**Usage**
```r
ic_char_event(ic)
```

**Arguments**
- `ic` object of class ical

**Examples**
```r
ic <- ical(ical_example)
ic_char_event(ic)
ic_char_event(ic[1, , ])
```

---

ic_dataframe  
*Convert iCal lines of text into a data frame*

**Description**
Returns a data frame

**Usage**
```r
ic_dataframe(x)
```

**Arguments**
- `x` Lines read-in in from an iCal file

**Examples**
```r
ic_dataframe(ical_example)
ic_dataframe(ical_outlook)
ics_file <- system.file("extdata", "england-and-wales.ics", package = "calendar")
x = readLines(ics_file)
x_df = ic_dataframe(x)
head(x_df)
x = data.frame(x_df)
x_df2 = ic_dataframe(x)
identical(x, x_df2)
```
ic_date

Convert ical date into R date

Description
Convert ical date into R date

Usage
ic_date(x)

Arguments
x
Lines read-in in from an iCal file

Examples
ic_date("20120103")

ic_datetime

Convert ical datetime into R datetime Z at the end of an ical stamp stands of Zulu time
which is UTC = GMT https://greenwichmeantime.com/info/zulu/

Description
Convert ical datetime into R datetime Z at the end of an ical stamp stands of Zulu time
which is UTC = GMT https://greenwichmeantime.com/info/zulu/

Usage
ic_datetime(x)

Arguments
x
Lines read-in in from an iCal file

Examples
ic_datetime("20180809T160000Z")
ic_date("20120103")
ic_event  
Create ical object from properties_core inputs

Description

Create an ical event using either POSIXct type or character type with format parameters.

Usage

```r
ic_event(uid = ic_guid(),
         start_time = as.POSIXct(round.POSIXt(Sys.time(), units = "hours")),
         end_time = 1, format = "%Y-%m-%d %H:%M",
         summary = "ical event", more_properties = FALSE,
         event_properties = calendar::properties)
```

Arguments

- `uid`: the unique id of the event, by default generated by `ic_uid()`
- `start_time`: start time, by default the start time plus one hour
- `end_time`: a number representing the number of hours after `start_time` or an datetime object (of class POSIXct) when the event ends. By default set to 1, meaning 1 hour after `start_time`.
- `format`: required if `start_time` and `end_time` are vectors and are not of datetime format
- `summary`: short outline of the event
- `more_properties`: add placeholder columns for properties in addition to properties_core, by default FALSE
- `event_properties`: named vector of additional properties to include. By default These include names stored in the data object properties (a packaged dataset), containing NAs to be subsequently populated.

Format

In case of `start_time` and `end_time` being character values, a format must be provided.

Value

object of class ics
Examples

```
ic_event()
s <- lubridate::ymd_h("2019-01-01 00")
ic_event(start_time = s, end_time = 3)
# also accepts this format by default, thanks to lubridate::ymd_h:
ic_event(start_time = "2019-01-01 00")
ic_event(more_properties = TRUE)
ic_event(start_time = "18-10-12", end_time = "18-10-13", format = calendar::formats$\"yy-mm-dd\")
```

---

ic_extract  
Extract contents of iCal fields

Description

Return formatted data from iCal fields

Usage

```
ic_extract(x, pattern)
```

Arguments

- **x**: Lines read-in from an iCal file
- **pattern**: A text string to search from (an ical field)

Examples

```
ic_extract(ical_example, "DTSTART")
```

---

ic_extract_raw  
Extract raw contents of iCal fields

Description

Return info from iCal files, return raw unprocessed text.

Usage

```
ic_extract_raw(x, pattern)
```

Arguments

- **x**: Lines read-in from an iCal file
- **pattern**: A text string to search from (an ical field)

Examples

```
pattern = "TSTAMP"
ic_extract_raw(ical_example, pattern)
```
**ic_find**  
*Find contents of iCal fields*

**Description**
Find contents of iCal fields

**Usage**

```python
ic_find(x, pattern)
```

**Arguments**
- `x`  
  Lines read-in in from an iCal file
- `pattern`  
  A text string to search from (an ical field)

**Examples**

```python
pattern = "DTSTAMP"
ic_find(ical_example, pattern)
```

---

**ic_guid**  
*Get an ical GUID*

**Description**
Provided without any testing. Slight improvement from SO question

**Usage**

```python
ic_guid()
```

**Examples**

```python
ic_guid()
```
ic_list

Convert raw ical text into a list of items

Description

This function breaks-up the iCalendar object into a list. By default it breaks it into events, where the number of events is the number of BEGIN:VEVENT event initiation lines (assumes all events start and end with :VEVENT), as per the specification (see ic_spec()).

Usage

ic_list(x, pattern = "BEGIN:VEVENT", include_pattern = FALSE)

Arguments

x
Lines read-in from an iCal file

pattern
A text string to search from (an ical field)

include_pattern
should the pattern be included in the output? FALSE by default.

Examples

ic_list(ical_example)
ics_file <- system.file("extdata", "england-and-wales.ics", package = "ical")
x = readLines(ics_file)
ics_list = ic_list(x)
ics_list[1:2]
ic_list(x, include_pattern = TRUE)

ic_read

Read ics file

Description

Read ics file

Usage

ic_read(file)

Arguments

file
ics file to read

Value

object of class ics
Examples

```r
f <- system.file("extdata", "england-and-wales.ics", package = "calendar")
ics_df = ic_read(f)
head(ics_df)
```

---

**ic_spec**

*View or download the ical specification*

**Description**

This function shows the spec underlying iCal files.

**Usage**

```r
ic_spec(mode = "view",
spec_url = "https://tools.ietf.org/rfc/rfc5545.txt")
```

**Arguments**

- `mode` character string specifying whether to look at the spec (`mode = "view"`, the default) or read it in (`mode = "read"`)
- `spec_url` the location of the latest version of the spec, from tools.ietf.org/rfc/rfc5545.txt by default

---

**ic_vector**

*Return a named vector from raw iCal text*

**Description**

This is designed to be a helper function for creating data frames ical lists.

**Usage**

```r
ic_vector(x, pattern = "[A-Z]-?[A-Z]"
```

**Arguments**

- `x` Lines read-in in from an iCal file
- `pattern` A text string to search from (an ical field)

**Examples**

```r
x = ical_example[18:19]
ic_vector(x)
```
ic_write

Write ics file

Usage

ic_write(ic, file)

Arguments

ic  object of class ical
file  ics file to write

Examples

ic <- ical(ical_example)
ic_write(ic, file.path(tempdir(), "ic.ics"))
f <- system.file("extdata", "example.ics", package = "calendar")
identical(readLines(file.path(tempdir(), "ic.ics")), readLines(f))
f <- system.file("extdata", "england-and-wales.ics", package = "calendar")
ics_df <- ic_read(f)
ic_write(ics_df, file.path(tempdir(), "ic.ics"))
# test similarity between files with diff tool like meld - from shell:
# meld ic.ics instOextdataOengland-and-wales.ics

properties

The key 'properties' that are allowed in ical files

Description

The key 'properties' that are allowed in ical files

Usage

properties

Format

An object of class character of length 54.
Properties

Examples

```r
# doc = ic_spec(mode = "read")
# key_locations = grepl(pattern = " \| [A-Z]", x = doc)
# summary(key_locations)
# doc_key = doc[key_locations]
# # regexplain::regex_gadget(text = doc_key) # explore
# key_properties = grepl(pattern = "3.7|3.8", x = doc)
# doc_properties = doc[key_locations & key_properties]
# # regexplain::regex_gadget(text = doc_properties) # explore
# properties = regmatches(doc_properties, regexpr('[A-Z][A-Z]+', doc_properties))
# usethis::use_data(properties)
```

---

**properties_core**

The key 'properties' that are allowed in ical files

---

Description

The key 'properties' that are allowed in ical files

Usage

properties_core

Format

An object of class list of length 4.

Examples

```r
properties_core = list(
    "UID"="UID",
    "DTSTART"="DTSTART",
    "DTEND"="DTEND",
    "SUMMARY"="SUMMARY"
)
```

---

**properties_ical**

ical default VCALENDAR properties in one line vectors.

---

Description

ical default VCALENDAR properties in one line vectors.

Usage

properties_ical
Format

An object of class character of length 5.

Examples

```r
# properties_ical <- ical_example[1:5]
# properties_ical
# properties_ical[2] <- "PRODID:-//ATFutures/ical //EN"
# use this::use_data(properties_ical)
```
Index

*Topic datasets
  formats, 3
  holidays, 3
  ical_example, 4
  ical_outlook, 5
  properties, 15
  properties_core, 16
  properties_ical, 16

calendar, 2
calendar-package (calendar), 2

formats, 3

holidays, 3

ic_attributes_vec, 5
ic_bind_list, 6
ic_char_date(ic_char_datetime), 7
ic_char_datetime, 7
ic_char_event, 8
ic_character, 7
ic_dataframe, 8
ic_date, 9
ic_datetime, 9
ic_event, 10
ic_extract, 11
ic_extract_raw, 11
ic_find, 12
ic_guid, 12
ic_list, 13
ic_read, 13
ic_spec, 14
ic_spec(), 13
ic_vector, 14
ic_write, 15
ical, 4
ical_example, 4
ical_outlook, 5

properties, 15