Package 'iso8601'

December 20, 2024

Type Package

Title Working with ISO8601 Dates and Times

Version 0.1.2

Description Functions to parse strings with ISO8601 dates, times, and
date-times into R-objects. Additionally, there are functions to determine
the type of ISO8601 string and to standardise ISO8601 strings.

BugReports https://github.com/djvanderlaan/iso8601/issues

URL https://github.com/djvanderlaan/iso8601/tree/main/R-package

Depends R (>= 3.6.0)

Imports Rcpp, utils

Suggests methods

LinkingTo Rcpp

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation yes

Author Jan van der Laan [aut, cre] (<https://orcid.org/0000-0002-0693-1514>)

Maintainer Jan van der Laan <r@eoos.dds.nl>

Repository CRAN

Date/Publication 2024-12-20 15:10:02 UTC

Contents

---

iso8601standardise   *Standardise ISO8601 strings*

Description

Standardise ISO8601 strings

Usage

```
iso8601standardise(x, fillmissing = TRUE, toymd = TRUE, tozulu = TRUE)
```

Arguments

x   a character vector with valid ISO8601 date. time, or date-time strings.

fillmissing   recplace missing parts ot dates, and times with the minimum possible values. In
                    dates this is 1, in times this is 0.

toymd   convert dates in year-week-weekday format, or year-day format to year-month-
                    day format.

tozulu   convert times to GMT. Note that this implies fillmissing = TRUE for date-time
                    strings with time zone.

Value

Returns a character vector with the same length as x with the ISO8601 strings in a standardised
format. For input strings that are not a valid ISO8601 time, date or date-time string NA is returned
and a warning is issued.

Examples

```
''
x <- c("2024-01-01T12:34", "2024W011", "2024-123T13:00:00+01", "T1530", NA)

iso8601standardise(x)
iso8601standardise(x, fillmissing = FALSE)
iso8601standardise(x, toymd = FALSE)
iso8601standardise(x, tozulu = FALSE)
```

iso8601todataframe	*Split Date, Time and DateTime ISO8601 strings into their elements*

Description

Split Date, Time and DateTime ISO8601 strings into their elements

Usage

```
iso8601todataframe(
  x,
  transformdate = c("no", "toyearmonthday", "toyearday"),
  ndigitsyear = 4L
)
```

Arguments

x	character vector of date, time or date-time strings
transformdate	Transform the date to the given format. This also immplies that missing parts of the date are replaces by values of 1.
ndigitsyear	Number of digits used to encode the year. This should be an integer with values >= 4 with the same length as x or length one. When it is a vector with length greater than one, a different value is used for each element of x.

Value

Returns a data.frame with possibly the following columns:

type	Type of ISO 8601 string. A factor with the following possible values: 'Date', 'Time', 'Datetime', 'Duration', 'Interval', 'RepeatingInterval'.
year	Year.
month	Month.
day	Day of the month.
week	Week
weekday	Day of the week
yearday	Day of the year.
hour	Hour of the day.
minutes	Minutes
seconds	Seconds
tzoffsethours	The time zone offset. An offset of +1 means 1 hour to the east of GMT.
tzoffsetminutes	The additional offset in minutes (between 0 and 59).

Columns corresponding to elements that do not occur in any of the strings are omitted from the resulting data.frame.

Examples

```
iso8601todataframe(c("2014-W01-1", "2041-02-12T12+00", NA, "T22"))

iso8601todataframe(c("2014-W01-1", "2041-02-12T12+00", NA, "T22"),
  transformdate = "toyearmonthday")
```

---

iso8601todate   *Convert date in ISO8601 format into R Date object*

Description

Convert date in ISO8601 format into R Date object

Usage

```
iso8601todate(x, ndigitsyear = 4L)
```

Arguments

x   character vector of ISO8601 date string. When the string also contains a time part, this is ignored.

ndigitsyear   Number of digits used to encode the year. This should be an integer with values >= 4 with the same length as x or length one. When it is a vector with length greater than one, a different value is used for each element of x.

Details

Dates are accepted in any of the formats supported by ISO8601.

Value

Returns a [Date](#) object with the same length as x.

Examples

```
iso8601todate(c("2024-01-01", "2024-W01-1", "2024001",
  "2024-01-01T12:30"))

iso8601todate("2019-08-17")
iso8601todate("2019-08")
iso8601todate("2019")
iso8601todate("20190817")
iso8601todate("2019-W33-6")
iso8601todate("2019-W33")
iso8601todate("2019W336")
iso8601todate("2019W33")
iso8601todate("2019-229")
```

```
iso8601todate("2019229")
iso8601todate("\u22120009-123")
iso8601todate("-0009")
iso8601todate("+002019-229", ndigitsyear = 6)
```

---

iso8601todatetime   *Convert date-time in ISO8601 format into R POSIXct object*

Description

Convert date-time in ISO8601 format into R POSIXct object

Usage

```
iso8601todatetime(x, ndigitsyear = 4L)
```

Arguments

x   character vector of date-time objects

ndigitsyear   Number of digits used to encode the year. This should be an integer with values
                 >= 4 with the same length as x or length one. When it is a vector with length
                 greater than one, a different value is used for each element of x.

Details

Date-time strings with a time-zone are converted to UTC. If all date-time strings have a time zone
the returned object will have it's display time zone set to GMT, otherwise the time zone is set to
local time. It is assumed that date-times without time zone are in local time.

Missing parts of the time are set to 0.

Value

Returns a [POSIXct](#) object with the same length as x. The object additionally has a 'timezone'
attribute which is a character vector with the same length as x. This vector contains the original
time zone of the ISO8601 date-time.

Examples

```
x <- iso8601todatetime(c("2024-01-01T12:30", "2024-W01-1T12:30Z",
  "2024-001T12:30+01"))
x
attr(x, "timezone")

# Examples of full date-times
iso8601todatetime("2019-08-17T16:15:14Z")
iso8601todatetime("2019-08-17T161514Z")
iso8601todatetime("2019-08-17T16:15Z")
```

```
iso8601todatetime("2019-08-17T1615")
iso8601todatetime("2019-08-17T16Z")
iso8601todatetime("+002019-08-17T16:15:14Z", ndigitsyear = 6)

# Fractional times
iso8601todatetime("2019-08-17T16:15:14,00Z")
iso8601todatetime("2019-08-17T16:15:14.00Z")
iso8601todatetime("2019-08-17T161514.00Z")
iso8601todatetime("2019-08-17T161514,00Z")
iso8601todatetime("2019-08-17T16:15.24Z")
iso8601todatetime("2019-08-17T16:15,24Z")
iso8601todatetime("2019-08-17T1615.24Z")
iso8601todatetime("2019-08-17T1615,24Z")
iso8601todatetime("2019-08-17T16.2539Z")
iso8601todatetime("2019-08-17T16,2539Z")

# When extended format T can be omitted
iso8601todatetime("2019-08-17 16:15:14Z")
iso8601todatetime("2019-08-17 16:15:14,00Z")
iso8601todatetime("2019-08-17 16:15:14.00Z")
iso8601todatetime("2019-08-17 16:15Z")
iso8601todatetime("2019-08-17 16:15Z")
iso8601todatetime("2019-08-17 16:15.24Z")
iso8601todatetime("2019-08-17 16:15,24Z")

# Time zones
iso8601todatetime("2019-08-17T16:15:14Z")
iso8601todatetime("2019-08-17T16:15:14+01:00")
iso8601todatetime("2019-08-17T16:15:14\u00b100:00")
iso8601todatetime("2019-08-17T16:15:14-01")
iso8601todatetime("2019-08-17T16:15:14\u221200:00")
iso8601todatetime("2019-08-17T16:15:14")
```

---

iso8601totime   *Convert time in ISO8601 format into R POSIXct object*

Description

Convert time in ISO8601 format into R POSIXct object

Usage

```
iso8601totime(x)
```

Arguments

x   character vector of time strings

Details

Times are acceptes in any of the formats supported by ISO8601.

Value

Returns a Time object, which is a subclass of POSIXct object. Time is represented as a time on 1970-01-01. The only difference between the Time object and the POSIXct object is the formatting, which leaves out the date part.

Examples

```
iso8601totime(c("12:30:12", "T12", "T000910"))

iso8601totime("T16:15:14")
iso8601totime("T16:15:14,00")
iso8601totime("T16:15:14.00")
iso8601totime("T161514")
iso8601totime("T161514.00")
iso8601totime("T161514,00")
iso8601totime("T16:15:14,00")
iso8601totime("T16:15:14.00")
iso8601totime("T161514.00")
iso8601totime("T161514,00")
iso8601totime("T16:15.24")
iso8601totime("T16:15,24")
iso8601totime("T1615.24")
iso8601totime("T1615,24")
iso8601totime("T16.2539")
iso8601totime("T16,2539")

# T can be omitted
iso8601totime("16:15:14")
iso8601totime("16:15:14,00")
iso8601totime("16:15:14.00")
iso8601totime("16:15:14,00")
iso8601totime("16:15:14.00")
iso8601totime("16:15.24")
iso8601totime("16:15,24")
iso8601totime("161514")
iso8601totime("161514,00")
iso8601totime("161514.00")
iso8601totime("1615")
iso8601totime("1615")
iso8601totime("1615.24")
iso8601totime("1615,24")
```

---

iso8601type   *Determine the type of ISO8601 strings*

Description

Determine the type of ISO8601 strings

Usage

```
iso8601type(x)
```

Arguments

x   character vector with ISO8601 strings

Value

Character vector with the same length as 'x' indicating the type of object each element in 'x' is. Elements of this string are:

Y	Year.
M	Month, or minutes in the time part. When followed by a '.' in the time part these are fractional minutes.
D	Day, this can be dau of the month, day of the week or day of the year.
T	Marks the start of the time part.
H	Hour. When followed by a '.' these are fractional hours.
M	Month.
S	Seconds. When followed by a '.' these are fractional seconds.
Z	Time is in GMT/Zulu time.
±Z	The time zone is indicated using an offset from GMT.
P	A period. These are not parsed further.
I	An interval. These are not parsed further.
R	A repeating interval. These are not parsed further.

Examples

```
iso8601type(c("T12", "2045-W05-1T13.5", "2012-12-05", "13:25"))
```

Index