Package ‘dttr2’

February 5, 2020

Title Manipulate Dates, DateTimes and Times
Version 0.0.2
Description Manipulates date (Date), datetime (POSIXct) and time (hms) vectors.
   Date/times are considered discrete and are floored whenever encountered.
   Times are wrapped and time zones are maintained unless explicitly altered
   by the user.
License MIT + file LICENSE
Depends R (>= 3.4.0)
Imports checkr, hms
Suggests testthat, covr
URL https://github.com/poissonconsulting/dttr
BugReports https://github.com/poissonconsulting/dttr/issues
Encoding UTF-8
LazyData true
RoxygenNote 7.0.2
Language en-US
NeedsCompilation no
Author Joe Thorley [aut, cre] (<https://orcid.org/0000-0002-7683-4592>)
Maintainer Joe Thorley <joe@poissonconsulting.ca>
Repository CRAN
Date/Publication 2020-02-05 10:40:02 UTC

R topics documented:

check_tz .................................................. 2
dtt .......................................................... 3
dtt_add_units ............................................. 3
dtt_adjust_tz ............................................. 4
dtt_adjust_units ......................................... 5
check_tz

Description

Checks an object’s time zone as returned by dtt_tz().
Usage

check_tz(x, tz = dtt_tz(x), x_name = substitute(x), error = TRUE)

Arguments

x  The object to check.
tz  A string of the time zone to check that x’s matches.
x_name  A string of the name of the object.
error  A flag indicating whether to throw an informative error or immediately generate an informative message if the check fails.

Value

An invisible copy of x (if it doesn’t throw an error).

See Also

  dtt_tz

Examples

check_tz(Sys.time(), "UTC", error = FALSE)

---

dtt

**dtt Object**

Description

A dtt (short for date time) object is an object of class Date (date), POSIXct (datetime) or hms (time).

---

dtt_add_units

**Add Units**

Description

Add time units to a date time vector.
Usage

dtt_add_units(x, units, n = 1L)
dtt_add_years(x, n = 1L, ...)
dtt_add_months(x, n = 1L, ...)
dtt_add_days(x, n = 1L, ...)
dtt_add_hours(x, n = 1L, ...)
dtt_add_minutes(x, n = 1L, ...)
dtt_add_seconds(x, n = 1L, ...)

Arguments

x A date time vector.
units A string of the units.
n An integer of the number of units.
... Unused.

Value

The modified date time vector.

See Also

dtt_subtract_units()

Examples

dtt_add_units(as.Date("1999-12-31"), "days")

---

dtt_adjust_tz Adjust Time Zone

Description

Adjusts the time zone so that clock (but not the actual) time is altered for a date time vector. Equivalent to lubridate::with_tz().

Usage

dtt_adjust_tz(x, tz = dtt_default_tz(), ...)

## S3 method for class 'POSIXct'
dtt_adjust_tz(x, tz = dtt_default_tz(), ...)
**dtt_adjust_units**

**Arguments**

- **x**: A POSIXct vector.
- **tz**: A string of the time zone.
- **...**: Unused

**Value**

The date time vector with the new time zone and time.

**Methods (by class)**

- **POSIXct**: Adjust the time zone for a POSIXct vector

**See Also**

- `dtt_set_tz()`

**Examples**

```r
dtt_adjust_tz(as.POSIXct("1970-01-01", tz = "Etc/GMT+8"), tz = "UTC")
```

---

**dtt_adjust_units**

**Description**

Adjust Units

**Usage**

```r
dtt_adjust_units(x, from = "seconds", to = "seconds")
```

**Arguments**

- **x**: An integer or numeric vector
- **from**: A string of the original units.
- **to**: A string of the new units.

**Value**

A numeric vector.

**Examples**

```r
dtt_adjust_units(60, to = "minutes")
```
**dtt_aggregate**

**Description**

Aggregates a date/time vector

**Usage**

```r
dtt_aggregate(x, units, ...)
```

```r
## S3 method for class 'Date'

# S3 method for class 'Date'

# S3 method for class 'POSIXct'

dtt_aggregate(x, units = "days", ...)

# S3 method for class 'POSIXct'

dtt_aggregate(x, units = "seconds", ...)

# S3 method for class 'hms'

dtt_aggregate(x, units = "seconds", ...)
```

**Arguments**

- `x`: A date/time vector.
- `units`: A string of the units to aggregate by.
- `...`: Unused.

**Details**

The possible units values are 'seconds', 'minutes', 'hours', 'days', 'months' or 'years'.

**Value**

The floored date/time vector.

**Methods (by class)**

- Date: Aggregate a Date vector
- POSIXct: Aggregate a POSIXct vector
- hms: Aggregate a hms vector

**Examples**

```r
dtt_aggregate(as.Date(c("1992-01-01", "1991-02-02", "1991-03-03")), "years")
```
Description

Completes date/time vector.

Usage

dtt_complete(x, ...)

## S3 method for class 'Date'
dtt_complete(
  x,
  from = min(x),
  to = max(x),
  units = "days",
  unique = TRUE,
  sort = TRUE,
  ...
)

## S3 method for class 'POSIXct'
dtt_complete(
  x,
  from = min(x),
  to = max(x),
  units = "seconds",
  unique = TRUE,
  sort = TRUE,
  ...
)

## S3 method for class 'hms'
dtt_complete(
  x,
  from = min(x),
  to = max(x),
  units = "seconds",
  unique = TRUE,
  sort = TRUE,
  ...
)

Arguments

x A date/time vector.
... Unused.
from A date/time scalar of the start.
to A date/time vector of the end.
units A string of the time units.
unique A flag specifying whether to only return unique values.
sort A flag specifying whether to sort the vector.

Value
The completed date/time vector.

Methods (by class)
- Date: Complete a Date sequence vector
- POSIXct: Complete a POSIXct sequence vector
- hms: Complete a hms sequence vector

Examples

dtt_complete(as.Date(c("2001-01-01", "2001-01-03", "2001-01-01")))

<table>
<thead>
<tr>
<th>dtt_completed</th>
<th>Completed</th>
</tr>
</thead>
</table>

Description
Tests whether a date time is complete.

Usage
dtt_completed(x, ...)

## S3 method for class 'Date'
dtt_completed(x, units = "days", unique = TRUE, sorted = TRUE, ...)

## S3 method for class 'POSIXct'
dtt_completed(x, units = "seconds", unique = TRUE, sorted = TRUE, ...)

## S3 method for class 'hms'
dtt_completed(x, units = "seconds", unique = TRUE, sorted = TRUE, ...)
Arguments

- **x**: A date time vector
- **...**: Unused.
- **units**: A string of the units.
- **unique**: A flag indicating whether the values must be unique.
- **sorted**: A flag indicating whether the values must be sorted.

Value

A flag indicating whether complete.

Methods (by class)

- **Date**: Test if Date vector is complete
- **POSIXct**: Test if POSIXct vector is complete
- **hms**: Test if POSIXct vector is complete

Description

Coerces vectors to floored Date vectors.

Usage

```r
dtt_date(x, ...)
```

```r
## S3 method for class 'integer'
dtt_date(x, ...)
```

```r
## S3 method for class 'double'
dtt_date(x, ...)
```

```r
## S3 method for class 'character'
dtt_date(x, ...)
```

```r
## S3 method for class 'Date'
dtt_date(x, ...)
```

```r
## S3 method for class 'POSIXct'
dtt_date(x, ...)
```

```r
## S3 method for class 'hms'
dtt_date(x, ...)
```
Arguments

x A vector.

... Unused.

Value

A floored Date vector.

Methods (by class)

- integer: Coerce integer vector to a floored Date vector
- double: Coerce double vector to a floored Date vector
- character: Coerce character vector to a floored Date vector
- Date: Coerce Date vector to a floored Date vector
- POSIXct: Coerce POSIXct vector to a floored Date vector
- hms: Coerce hms vector to a floored Date vector

Examples

dtt_date(1L)
dtt_date(-1)
dtt_date("2000-01-01")
as.Date(as.POSIXct("2019-05-01", tz = "Etc/GMT-8"))
dtt_date(as.POSIXct("2019-05-01", tz = "Etc/GMT-8"))
dtt_date(hms::as.hms("23:59:59"))
dtt_date(hms::as.hms("24:00:00"))

dtt_date_time

Date Time

Description

Coerces vectors to floored POSIXct vectors.

Usage

dtt_date_time(x, ...)

## S3 method for class 'integer'
dtt_date_time(x, tz = dtt_default_tz(), ...)

## S3 method for class 'double'
dtt_date_time(x, tz = dtt_default_tz(), ...)

## S3 method for class 'character'
dtt_date_time(x, tz = dtt_default_tz(), ...)
```r
## S3 method for class 'Date'
\texttt{dtt\_date\_time(x, time = \texttt{hms::as.hms("00:00:00")}, tz = dtt\_default\_tz(), ...)}

## S3 method for class 'POSIXct'
\texttt{dtt\_date\_time(x, tz = dtt\_tz(x), ...)}

## S3 method for class 'hms'
\texttt{dtt\_date\_time(x, date = dtt\_date("1970-01-01"), tz = dtt\_default\_tz(), ...)}

### Arguments

- \textit{x}: A vector.
- \textit{...}: Unused.
- \textit{tz}: A string of the time zone.
- \textit{time}: A \texttt{hms} vector of the time.
- \textit{date}: A \texttt{Date} vector of the date.

### Value

A floored \texttt{POSIXct} vector.

### Methods (by class)

- \texttt{integer}: Coerce integer vector to a floored \texttt{POSIXct} vector
- \texttt{double}: Coerce double vector to a floored \texttt{POSIXct} vector
- \texttt{character}: Coerce character vector to a floored \texttt{POSIXct} vector
- \texttt{Date}: Coerce \texttt{Date} vector to a floored \texttt{POSIXct} vector
- \texttt{POSIXct}: Coerce \texttt{POSIXct} vector to a floored \texttt{POSIXct} vector
- \texttt{hms}: Coerce \texttt{hms} vector to a floored \texttt{POSIXct} vector

### Examples

\begin{verbatim}
\texttt{dtt\_date\_time(1L)}
\texttt{dtt\_date\_time(-1)}
\texttt{dtt\_date\_time(1, tz = "Etc/GMT+8")}
\texttt{dtt\_date\_time(as.Date("2000-01-02"))}
\texttt{dtt\_date\_time(as.Date("2000-01-02"), time = \texttt{hms::as.hms("04:05:06")})
\end{verbatim}
```
Get and Set Day Values

Description

Gets and sets day values for date/time vectors.

Usage

dtt_day(x, ...)
dtt_day(x) <- value

## S3 method for class 'Date'
dtt_day(x, ...)

## S3 method for class 'POSIXct'
dtt_day(x, ...)

## S3 replacement method for class 'Date'
dtt_day(x) <- value

## S3 replacement method for class 'POSIXct'
dtt_day(x) <- value

dtt_days(x, ...)
dtt_days(x) <- value

dtt_set_day(x, value)

Arguments

x A date/time vector.
...
value A integer vector of the day value(s).

Value

An integer vector (or the modified date/time vector).

Methods (by class)

- Date: Get integer vector of day values for a Date vector
- POSIXct: Get integer vector of day values for a POSIXct vector
- Date: Set day values for a Date vector
- POSIXct: Set day values for a POSIXct vector
### Examples

```r
dx <- as.Date("1990-01-02")
dtt_day(x)
dtt_day(x) <- 27L
x

x <- as.POSIXct("1990-01-02 23:40:51")
dtt_day(x)
dtt_day(x) <- 27L
x
```

### Description

Dayte

### Usage

```r
dtt_dayte(x, ...)
```

#### S3 method for class 'Date'

```r
dtt_dayte(x, start = 1L, ...)
```

#### S3 method for class 'POSIXct'

```r
dtt_dayte(x, start = 1L, ...)
```

### Arguments

- `x`: A date time vector.
- `...`: Unused.
- `start`: An integer scalar of the starting month or a Date scalar of the starting date.

### Value

A Date vector with the year set to year.
A Date vector of the daytes.

### Methods (by class)

- Date: Dayte a Date vector
- POSIXct: Dayte a POSIXct vector

### Examples

```r
dtt_dayte(as.Date(c("2001-01-01", "2015-12-13")))
```
### dtt_dayte_time

**Description**
Dayte Time

**Usage**

dtt_dayte_time(x, ...

```r
## S3 method for class 'POSIXct'
dtt_dayte_time(x, start = 1L, ...)
```

**Arguments**

- `x`: A date time vector.
- `...`: Unused.
- `start`: An integer scalar of the starting month or a Date scalar of the starting date.

**Value**
A Date vector with the year set to year.
A Date vector of the daytes.

**Methods (by class)**
- POSIXct: Dayte Time a POSIXct vector

**Examples**
dtt_dayte_time(as.POSIXct(c("2001-01-01 12:13:14", "2015-12-13"), tz = "Etc/GMT+10"))

### dtt_daytt

**Description**
Dayte Time

**Usage**

dtt_daytt(x, start = 1L)
**dtt_doy**

**Arguments**

- **x**: A Date or POSIXct vector.
- **start**: An integer vector specifying the start month of the year or a Date vector of the start date.

**Value**

A Date or POSIXct vector with the year for February 29th as 1972.

---

**dtt_doy**

*Day of the Year*

---

**Description**

Day of the Year

**Usage**

\[
\text{dtt\_doy}(x, \ldots)
\]

**Arguments**

- **x**: A Date or POSIXct vector.
- **\ldots**: Unused.

**Value**

A integer vector between 1 and 366 of the day of the year.

**Examples**

\[
\text{dtt\_doy(Sys.Date())}
\]

---

**dtt_doy\_to\_date**

*Day of the Year to Date*

---

**Description**

Day of the Year to Date

**Usage**

\[
\text{dtt\_doy\_to\_date}(x, \text{year} = 1972L)
\]
Arguments

x  An integer vector of the Day of the Year.

year  An integer scalar or vector of the year.

Value

A Date vector.

Examples

dtt_doy_to_date(3L)

dtt_feb29_to_28(as.Date("2004-02-29"))

Description

Converts Feb 29 to Feb 28th

Usage

dtt_feb29_to_28(x)

Arguments

x  A Date or POSIXct vector.

Value

The modified Date or POSIXct vector.

Examples

dtt_feb29_to_28(as.Date("2004-02-29"))
**dtt_floor**

**Floor**

Description

Floors a date/time vector

Usage

dtt_floor(x, units, ...)

## S3 method for class 'Date'
dtt_floor(x, units = "days", ...)

## S3 method for class 'POSIXct'
dtt_floor(x, units = "seconds", ...)

## S3 method for class 'hms'
dtt_floor(x, units = "seconds", ...)

Arguments

- **x** A date/time vector.
- **units** A string of the units to floor by.
- **...** Unused.

Value

The floored date/time vector.

Methods (by class)

- Date: Floor a Date vector
- POSIXct: Floor a POSIXct vector
- hms: Floor a hms vector

Examples

dtt_floor(hms::as.hms("23:59:59"), "hours")
Description

Test whether a date time vector is floored.

Usage

dtt_floored(x, ...)

## S3 method for class 'Date'
dtt_floored(x, units = "days", ...)

## S3 method for class 'POSIXct'
dtt_floored(x, units = "seconds", ...)

## S3 method for class 'hms'
dtt_floored(x, units = "seconds", ...)

Arguments

x           A Date, POSIXct or hms vector.
...          Unused.
units        A string of the time units to floor by.

Value

A flag indicating whether floored.

Methods (by class)

- Date: Test if Date vector is floored
- POSIXct: Test if POSIXct vector is floored
- hms: Test if hms vector is floored

Examples

dtt_floored(as.Date("2002-02-01"))
**dtt_hours**  
*Get and Set Hour Values*

**Description**

Gets and sets hour values for date/time vectors.

**Usage**

```r
dtt_hours(x, ...)  
dtt_hours(x) <- value  
dtt_hour(x, ...)  
dtt_hour(x) <- value  

## S3 method for class 'Date'  
dtt_hour(x, ...)  

## S3 method for class 'POSIXct'  
dtt_hour(x, ...)  

## S3 method for class 'hms'  
dtt_hour(x, ...)  

## S3 replacement method for class 'POSIXct'  
dtt_hour(x) <- value  

## S3 replacement method for class 'hms'  
dtt_hour(x) <- value  

dtt_set_hour(x, value)
```

**Arguments**

- **x**  
  A date/time vector.
- **...**  
  Unused.
- **value**  
  A integer vector of the hour value(s).

**Value**

An integer vector (or the modified date/time vector).
Methods (by class)

- Date: Get integer vector of hour values for a Date vector
- POSIXct: Get integer vector of hour values for a POSIXct vector
- hms: Get integer vector of hour values for a hms vector
- POSIXct: Set hour values for a POSIXct vector
- hms: Set hour values for a hms vector

Examples

```r
x <- as.POSIXct("1990-01-02 23:40:51")
dtt_hour(x)
dtt_hour(x) <- 01L
x

x <- hms::as.hms("23:40:51")
dtt_hour(x)
dtt_hour(x) <- 01L
x
```

---

dtt_is_date | Is Date
---

Description

Is Date

Usage

dtt_is_date(x)

Arguments

- x | An R object.

Value

A flag indicating whether R is a Date vector.
**dtt_is_date_time**

**Is Date Time**

**Description**

Is Date Time

**Usage**

dtt_is_date_time(x)

**Arguments**

x An R object.

**Value**

A flag indicating whether R is a POSIXct vector.

---

**dtt_is_dtt**

**Is Date or DateTime Object**

**Description**

Is Date or DateTime Object

**Usage**

dtt_is_dtt(x)

**Arguments**

x An R object.

**Value**

A flag indicating whether R is a Date or POSIXct vector.
Get and Set Minute Values

**Description**

Gets and sets minute values for date/time vectors.

**Usage**

```r
# Usage examples

dtt_minutes(x, ...)

dtt_minutes(x) <- value

dtt_minute(x, ...)

dtt_minute(x) <- value

## S3 method for class 'Date'

dtt_minute(x, ...)

## S3 method for class 'POSIXct'

dtt_minute(x, ...)

## S3 method for class 'hms'

dtt_minute(x, ...)

## S3 replacement method for class 'POSIXct'

dtt_minute(x) <- value

## S3 replacement method for class 'hms'

dtt_minute(x) <- value

# Arguments

x       A date/time vector.
...
value   A integer vector of the minute value(s).
```

**Value**

An integer vector (or the modified date/time vector).
Methods (by class)

- Date: Get integer vector of minute values for a Date vector
- POSIXct: Get integer vector of minute values for a POSIXct vector
- hms: Get integer vector of minute values for a hms vector
- POSIXct: Set minute values for a POSIXct vector
- hms: Set minute values for a hms vector

Examples

```r
x <- as.POSIXct("1990-01-02 23:40:51")
dtt_minute(x)
dtt_minute(x) <- 27L
x

x <- hms::as.hms("23:40:51")
dtt_minute(x)
dtt_minute(x) <- 27L
x
```

---

**dtt_months**

Get and Set Month Values

Description

Gets and sets month values for date/time vectors.

Usage

```r
dtt_months(x, ...)
dtt_months(x) <- value
dtt_month(x, ...)
dtt_month(x) <- value
```

## S3 method for class 'Date'
dtt_month(x, ...)

## S3 method for class 'POSIXct'
dtt_month(x, ...)

## S3 replacement method for class 'Date'
dtt_month(x) <- value

## S3 replacement method for class 'POSIXct'
dtt_month(x) <- value

dtt_set_month(x, value)

Arguments

x | A date/time vector.
... | Unused.
value | A integer vector of the month value(s).

Value

An integer vector (or the modified date/time vector).

Methods (by class)

- Date: Get integer vector of month values for a Date vector
- POSIXct: Get integer vector of month values for a POSIXct vector
- Date: Set month values for a Date vector
- POSIXct: Set month values for a POSIXct vector

Examples

```r
x <- as.Date("1990-01-02")
dtt_month(x)
dtt_month(x) <- 11L
x
```

```r
x <- as.POSIXct("1990-01-02 23:40:51")
dtt_month(x)
dtt_month(x) <- 11L
x
```

dtt_season | Season

Description

Returns an ordered factor of the user specified seasons.

Usage

dtt_season(x, start = c(Spring = 3L, Summer = 6L, Autumn = 9L, Winter = 12L))

Arguments

x | A Date or POSIXct vector
start | A uniquely named integer vector of the first month of each season or a uniquely named Date vector of the first date of each season.
### dtt_seconds

**Details**

If the first month of the first season isn’t January (1L), then the last season is considered to wrap into the following year.

**Value**

An ordered factor of the seasons.

**Examples**

```r
dates <- as.Date(c("2001-01-01", "2001-02-28", "2012-09-01", "2012-12-01"))
dtt_season(dates)
dtt_season(dates, start = c(Monsoon = 2L, "Dry Period" = 6L))
dtt_season(dates, start = c(First = dtt_date("2000-01-01"), Second = dtt_date("2000-06-01")))
```

### dtt_seconds

*Get and Set Second Values*

**Description**

Gets and sets second values for date/time vectors.

**Usage**

```r
dtt_seconds(x, ...)
dtt_seconds(x) <- value
dtt_second(x, ...)
dtt_second(x) <- value
```

```r
## S3 method for class 'Date'
dtt_second(x, ...)
```

```r
## S3 method for class 'POSIXct'
dtt_second(x, ...)
```

```r
## S3 method for class 'hms'
dtt_second(x, ...)
```

```r
## S3 replacement method for class 'POSIXct'
dtt_second(x) <- value
```

```r
## S3 replacement method for class 'hms'
dtt_second(x) <- value
```

```r
dtt_set_second(x, value)
```
Arguments

- **x**: A date/time vector.
- **...**: Unused.
- **value**: A integer vector of the second value(s).

Value

An integer vector (or the modified date/time vector).

Methods (by class)

- **Date**: Get integer vector of second values for a Date vector
- **POSIXct**: Get integer vector of second values for a POSIXct vector
- **hms**: Get integer vector of second values for a time vector
- **POSIXct**: Set second values for a POSIXct vector
- **hms**: Set second values for a hms vector

Examples

```r
x <- as.POSIXct("1990-01-02 23:40:51")
dtt_second(x)
dtt_second(x) <- 27L
x

x <- hms::as.hms("23:40:51")
dtt_second(x)
dtt_second(x) <- 27L
x
```

---

**dtt_seq**

Sequence

Description

Creates a date/time sequence vector. from and to are first floored and then a sequence is created by units. If length_out is defined then that number of units are added to from.

Usage

```r
dtt_seq(from, to, units, length_out = NULL, ...)
```

## S3 method for class 'Date'
```r
dtt_seq(from, to = from, units = "days", length_out = NULL, ...)
```

## S3 method for class 'POSIXct'
```r
dtt_seq(from, to = from, units = "seconds", length_out = NULL, ...)
```
dtt_set_tz

### dtt_set_tz

Sets the time zone for a date time vector without adjusting the clock time. Equivalent to `lubridate::force_tz()`.

#### Usage

```r
dtt_set_tz(x, tz = dtt_default_tz(), ...)
```

#### Examples

```r
dtt_seq(as.Date("2001-01-01"), as.Date("2001-01-05"))
```
Arguments

`x`  A date time vector.
`tz` A string of the new time zone.
`...` Unused.

Value

The date time vector with the new time zone.

Methods (by class)

- POSIXct: Set the time zone for a POSIXct vector

See Also

`dtt_adjust_tz()`

Examples

dtt_set_tz(as.POSIXct("1970-01-01", tz = "Etc/GMT+8"), tz = "UTC")

dtt_study_year(as.Date(c("2000-03-31", "2000-04-01", "2001-04-01")), start = 4L)
dtt_study_year(as.Date(c("2000-03-31", "2000-04-01", "2001-04-01")), start = 4L, full = FALSE)
**dtt_subtract_units**  

Subtract Units

Description

Subtract time units from a date time vector.

Usage

dtt_subtract_units(x, n = 1L, units = dtt_units(x))

dtt_subtract_years(x, n = 1L)

dtt_subtract_months(x, n = 1L)

dtt_subtract_days(x, n = 1L)

dtt_subtract_hours(x, n = 1L)

dtt_subtract_minutes(x, n = 1L)

dtt_subtract_seconds(x, n = 1L)

Arguments

- **x**: A date time vector.
- **n**: An integer of the number of units.
- **units**: A string of the units.

Value

The modified date time vector.

See Also

- `dtt_add_units`

Examples

dtt_subtract_units(as.Date("1999-12-31"), 2L, "days")
Get, Set or Reset Default Time Zone

**Description**

Get, Set or Reset Default Time Zone

**Usage**

```r
dtt_sys_tz()
dtt_set_sys_tz(tz = NULL)
dtt_reset_sys_tz()
dtt_default_tz()
dtt_set_default_tz(tz = NULL)
dtt_reset_default_tz()
```

**Arguments**

- `tz`: A string of the time zone.

**Value**

A string of the current or old time zone.

**Functions**

- `dtt_set_default_tz`: Set Default Time Zone
- `dtt_reset_default_tz`: Reset Default Time Zone

**Examples**

```r
## Not run:
dtt_default_tz()
old <- dtt_set_default_tz("Etc/GMT+8")
dtt_default_tz()
dtt_reset_default_tz()
dtt_default_tz()
dtt_set_default_tz(old)
dtt_default_tz()

## End(Not run)
```
dtt_time

---

**dtt_time**

**Time**

---

**Description**

Coerces vectors to floored (and wrapped) hms vectors.

**Usage**

```r
dtt_time(x, ...)
```

```r
## S3 method for class 'integer'
dtt_time(x, ...)
```

```r
## S3 method for class 'double'
dtt_time(x, ...)
```

```r
## S3 method for class 'character'
dtt_time(x, ...)
```

```r
## S3 method for class 'Date'
dtt_time(x, ...)
```

```r
## S3 method for class 'hms'
dtt_time(x, ...)
```

```r
## S3 method for class 'POSIXct'
dtt_time(x, ...)
```

```r
## S3 method for class 'POSIXlt'
dtt_time(x, ...)
```

**Arguments**

- **x**: A vector.
- **...**: Unused.

**Value**

A floored hms vector.

**Methods (by class)**

- **integer**: Coerce integer vector to a floored hms vector
- **double**: Coerce double vector to a floored hms vector
- **character**: Coerce character vector to a floored hms vector
• Date: Coerce Date vector to a floored hms vector
• hms: Coerce hms vector to a floored hms vector
• POSIXct: Coerce POSIXct vector to a floored hms vector
• POSIXlt: Coerce POSIXlt vector to a floored hms vector

Examples

dtt_time(1L)
dtt_time(1.999)
dtt_time(-0.001)
dtt_time(Sys.Date())
dtt_time(as.POSIXct("2001-01-01 02:30:40"))
dtt_time(as.POSIXct("2001-01-01 02:30:40", tz = "Etc/GMT-8"))

dtt_tz

Get, Set or Adjust Time Zone

Description

Gets, sets or the time zone for a date time vector.

Usage

dtt_tz(x, ...)

## S3 method for class 'POSIXct'
dtt_tz(x, ...)

Arguments

x A date time vector.
...

Value

A string of the time zone.

Methods (by class)

• POSIXct: Get the time zone for a POSIXct vector.

Examples

dtt_tz(as.POSIXct("1970-01-01", tz = "Etc/GMT+8"))
Description

Gets the smallest units for a date time vector. The possible values are 'seconds', 'minutes', 'hours', 'days', 'months' or 'years'.

Usage

dtt_units(x, ...)

## S3 method for class 'Date'
dtt_units(x, ...)

## S3 method for class 'POSIXct'
dtt_units(x, ...)

## S3 method for class 'hms'
dtt_units(x, ...)

Arguments

x          A Date, POSIXct or hms vector.

...        Unused.

Value

A string indicating the date time units.

Methods (by class)

- Date: Get time units for a Date vector
- POSIXct: Get time units for a POSIXct vector
- hms: Get time units for a hms vector

Examples

dtt_units(as.Date("2000-01-01"))
dtt_units(as.Date("2000-02-01"))
dtt_units(as.Date("2000-01-02"))
**dtt_units_per_unit**  *Units per Unit*

**Description**
Units per Unit

**Usage**
```r
dtt_units_per_unit(units = "seconds", unit = "days")
```

**Arguments**
- `units`: A string of the time units.
- `unit`: A string of the time unit.

**Value**
A number of the units per unit

**Examples**
```r
dtt_units_per_unit("hours")
```

---

**dtt_wrap**  *Wrap*

**Description**
Wrap

**Usage**
```r
dtt_wrap(x, ...)
```

**Arguments**
- `x`: A date/time vector.
- `...`: Unused.

**Examples**
```r
dtt_wrap(hms::as.hms("24:00:00"))
```
**dtt_years**

Get and Set Year Values

**Description**

Gets and sets year values for date/time vectors.

**Usage**

```r
dtt_years(x, ...)  
dtt_years(x) <- value  
dtt_set_year(x, value)  
dtt_year(x, ...)  
dtt_year(x) <- value  
```

```r  
## S3 method for class 'Date'  
dtt_year(x, ...)  
```

```r  
## S3 method for class 'POSIXct'  
dtt_year(x, ...)  
```

```r  
## S3 replacement method for class 'Date'  
dtt_year(x) <- value  
```

```r  
## S3 replacement method for class 'POSIXct'  
dtt_year(x) <- value  
```

**Arguments**

- **x**  
  A date/time vector.

- **...**  
  Unused.

- **value**  
  A integer vector of the year value(s).

**Value**

An integer vector (or the modified date/time vector).

**Methods (by class)**

- Date: Get integer vector of year values for a Date vector
- POSIXct: Get integer vector of year values for a POSIXct vector
- Date: Set year values for a Date vector
- POSIXct: Set year values for a POSIXct vector
Examples

```r
x <- as.Date("1990-01-02")
dtt_year(x)
dtt_year(x) <- 11L
x

x <- as.POSIXct("1990-01-02 23:40:51")
dtt_year(x)
dtt_year(x) <- 2022L
x
```

<table>
<thead>
<tr>
<th>is.datetime</th>
<th>Is Date/Time</th>
</tr>
</thead>
</table>

Description

Tests whether an object is a Date, POSIXct, or hms vector.

Usage

```r
is.POSIXct(x)
is.Date(x)
is.hms(x)
```

Arguments

- `x` An object

Value

A flag indicating whether `x` inherits from Date, POSIXct or hms.

<table>
<thead>
<tr>
<th>NA_Date_</th>
<th>Missing Date</th>
</tr>
</thead>
</table>

Description

A missing Date object

Usage

```r
NA_Date_ |
```

Format

An object of class Date of length 1.
**NA_hms_**

**Description**
A missing hms object

**Usage**
NA_hms_

**Format**
An object of class hms (inherits from diff time) of length 1.

---

**NA_POSIXct_**

**Description**
A missing POSIXct object

**Usage**
NA_POSIXct_

**Format**
An object of class POSIXct (inherits from POSIXt) of length 1.
Index

*Topic datasets
  NA_Date_, 36
  NA_hms_, 37
  NA_POSIXct_, 37

check_tz, 2

dtt, 3
dtt_add_days (dtt_add_units), 3
dtt_add_hours (dtt_add_units), 3
dtt_add_minutes (dtt_add_units), 3
dtt_add_months (dtt_add_units), 3
dtt_add_seconds (dtt_add_units), 3
dtt_add_units, 3, 29

dtt_adjust_tz, 4, 28

dtt_adjust_units, 5

dtt_aggregate, 6

dtt_complete, 7

dtt_completed, 8

dtt_date, 9

dtt_date_time, 10

dtt_day, 12

dtt_day<- (dtt_day), 12

dtt_days (dtt_day), 12

dtt_days<- (dtt_day), 12

dtt_dayte, 13

dtt_dayte_time, 14

dtt_daytt, 14

dtt_default_tz (dtt_sys_tz), 30
dtt_doy, 15

dtt_doy_to_date, 15

dtt_feb29_to_28, 16

dtt_floor, 17

dtt_floored, 18

dtt_hour (dtt_hours), 19
dtt_hour<- (dtt_hours), 19

dtt_hours, 19

dtt_hours<- (dtt_hours), 19

dtt_is_date, 20

dtt_is_date_time, 21

dtt_is_dtt, 21

dtt_minute (dtt_minutes), 22
dtt_minute<- (dtt_minutes), 22

dtt_minutes, 22

dtt_minutes<- (dtt_minutes), 22

dtt_month (dtt_months), 23

dtt_month<- (dtt_months), 23

dtt_months, 23

dtt_months<- (dtt_months), 23

dtt_second (dtt_seconds), 25

dtt_second<- (dtt_seconds), 25

dtt_seconds, 25

dtt_seconds<- (dtt_seconds), 25

dtt_seq, 26

dtt_set_day (dtt_day), 12

dtt_set_default_tz (dtt_sys_tz), 30

dtt_set_hour (dtt_hours), 19

dtt_set_minute (dtt_minutes), 22

dtt_set_month (dtt_months), 23

dtt_set_second (dtt_seconds), 25

dtt_set_sys_tz (dtt_sys_tz), 30

dtt_set_tz, 5, 27

dtt_set_year (dtt_years), 35

dtt_study_year, 28

dtt_subtract_days (dtt_subtract_units), 29

dtt_subtract_hours
  (dtt_subtract_units), 29

dtt_subtract_minutes
  (dtt_subtract_units), 29

dtt_subtract_months
  (dtt_subtract_units), 29

dtt_subtract_seconds
  (dtt_subtract_units), 29

dtt_subtract_units, 4, 29
dtt_subtract_years
  (dtt_subtract_units), 29

dtt_sys_tz, 30
dtt_time, 31
dtt_tz, 3, 32
dtt_units, 33
dtt_units_per_unit, 34
dtt_wrap, 34
dtt_year (dtt_years), 35
dtt_year<- (dtt_years), 35
dtt_years, 35
dtt_years<- (dtt_years), 35

is.Date (is.datetime), 36
is.datetime, 36
is.hms (is.datetime), 36
is.POSIXct (is.datetime), 36

NA_Date_, 36
NA_hms_, 37
NA_POSIXct_, 37