Package 'dttr2'

July 10, 2020

Title Manipulate Date, POSIXct and hms Vectors

Version 0.3.0

Description Manipulates date ('Date'), date time ('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

URL https://github.com/poissonconsulting/dttr2

BugReports https://github.com/poissonconsulting/dttr2/issues

Depends R (>= 3.3)

Imports chk, hms, lifecycle

Suggests covr, rlang, testthat

RdMacros lifecycle

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 7.1.1.9000

NeedsCompilation no

Author Joe Thorley [aut, cre] (<https://orcid.org/0000-0002-7683-4592>), Poisson Consulting [cph, fnd]

Maintainer Joe Thorley <joe@poissonconsulting.ca>

Repository CRAN

Date/Publication 2020-07-10 10:30:06 UTC

R topics documented:

check_tz *Check Time Zone*

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)
```

chk_time *Check Time*

Description

Checks if scalar hms object using `vld_time()`.

Usage

```
chk_time(x, x_name = NULL)
```

Arguments

`x`	The object to check.
`x_name`	A string of the name of object x or NULL.

Value

NULL, invisibly. Called for the side effect of throwing an error if the condition is not met.

Examples

```
chk_time(hms::as_hms("10:00:10"))
try(chk_time(1))
```

---

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(), ...)
```

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

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

dtt_adjust_units *Adjust Units*

Description

Adjust Units

Usage

```
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

```
dtt_adjust_units(60, to = "minutes")
```

dtt_aggregate *Aggregates*

Description

Aggregates a date/time vector

Usage

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

## S3 method for class 'Date'
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

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

---

dtt_complete *Complete*

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")))
```

dtt_completed *Completed*

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

dtt_date *Date*

Description

Coerces vectors to floored Date vectors.

Usage

```
dtt_date(x, ...)

dtt_date(x) <- value

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

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

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

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

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

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

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

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

dtt_set_date(x, value)

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

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

Arguments

x	A vector.

...	Unused.
value	A date vector.

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
- Date: Set date values for a Date vector
- POSIXct: Set date values for a POSIXct vector
- Date: Set time values for a Date vector
- POSIXct: Set date values for a POSIXct 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_add_time *Date Add Time*

Description

Adds times to Dates vector and sets timezone in a single function.

Usage

```
dtt_date_add_time(x, time, tz = dtt_default_tz())
```

Arguments

x	A Date vector.
time	A hms vector.
tz	A string of the timezone.

Value

A POSIXct vector.

Examples

```
dtt_date_add_time(as.Date("2001-03-05"), hms::as_hms("06:07:08"), tz = "Etc/GMT+9")
```

`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(), ...)

## S3 method for class 'Date'
dtt_date_time(x, time = hms::as_hms("00:00:00"), tz = dtt_default_tz(), ...)

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

## S3 method for class 'hms'
dtt_date_time(x, date = dtt_date("1970-01-01"), tz = dtt_default_tz(), ...)
```

Arguments

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

Value

A floored POSIXct vector.

Methods (by class)

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

Examples

```
dtt_date_time(1L)
dtt_date_time(-1)
dtt_date_time(1, tz = "Etc/GMT+8")
dtt_date_time(as.Date("2000-01-02"))
dtt_date_time(as.Date("2000-01-02"), time = hms::as_hms("04:05:06"))
```

`dtt_day` *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.
...	Unused.
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

```
x <- 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
```

---

dtt_dayte *Dayte*

Description

Dayte

Usage

```
dtt_dayte(x, ...)

## S3 method for class 'Date'
dtt_dayte(x, start = 1L, ...)

## S3 method for class 'POSIXct'
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

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

---

dtt_dayte_time *Dayte Time*

Description

Dayte Time

Usage

```
dtt_dayte_time(x, ...)

## S3 method for class 'Date'
dtt_dayte_time(x, start = 1L, tz = dtt_default_tz(), ...)

## 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.
tz	A string of the time zone.

Value

A Date vector with the year set to year.

A POSIXct vector of the dayte times.

Methods (by class)

- `Date`: Dayte Time a Date vector
- `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` *Dayte Time*

Description

Dayte Time

Usage

```
dtt_daytt(x, start = 1L)
```

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 dayte.

Value

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

dtt_diff *Time Difference*

Description

Gets the time difference in secs, minutes, hours, days or weeks. Uses difftime() but floors x and y first after coercing to POSIXct and adjusts the timezone of y to match that of x.

Usage

```
dtt_diff(x, y, units = "secs", as_difftime = FALSE)
```

Arguments

x	An object that can be coerced to a POSIXct using dtt_date_time().
y	An object that can be coerced to a POSIXct using dtt_date_time().
units	A string of the units. The possible values are "secs", "minutes", "hours", "days" or "weeks".
as_difftime	A flag specifying whether to return a difftime vector.

Value

A numeric vector of the time difference.

Examples

```
dtt_diff(as.Date(c("2001-01-02", "2000-12-31")), as.Date("2001-01-01"), "hours")
dtt_diff(as.Date("2001-01-02"), as.Date("2001-01-01"), "weeks")
```

dtt_doy *Day of the Year*

Description

Day of the Year

Usage

```
dtt_doy(x, ...)
```

Arguments

x	A Date or POSIXct vector.
...	Unused.

Value

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

Examples

```
dtt_doy(Sys.Date())
```

---

dtt_doy_to_date *Day of the Year to Date*

Description

Day of the Year to Date

Usage

```
dtt_doy_to_date(x, 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(3)
```

---

dtt_feb29_to_28 *Feb 29 to Feb 28*

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")
```

dtt_floored *Floored*

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

```
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

```
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.

dtt_minutes *Get and Set Minute Values*

Description

Gets and sets minute values for date/time vectors.

Usage

```
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

dtt_set_minute(x, value)
```

Arguments

x	A date/time vector.
...	Unused.
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

```
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

```
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

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

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

---

dtt_season *Season*

Description

Returns a factor of the user specified seasons.

Usage

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

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.
first	A string of the name of first season or NULL in which case the first season is that which includes Jan 1st.

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

A factor of the seasons.

Examples

```
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

```
dtt_seconds(x, ...)

dtt_seconds(x) <- value

dtt_second(x, ...)

dtt_second(x) <- value

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

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

## S3 method for class 'hms'
```

```
dtt_second(x, ...)

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

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

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

```
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

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

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

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

## S3 method for class 'hms'
dtt_seq(
  from,
  to = from,
  units = "seconds",
  length_out = NULL,
  wrap = TRUE,
  ...
)
```

Arguments

from	A date/time scalar of the start.
to	A date/time scalar of the end.
units	A string of the time units.
length_out	An integer of the number of units from from.
...	Unused
wrap	A flag specifying whether to wrap hms vectors from 23:59:59 to 00:00:00

Value

The date/time vector.

Methods (by class)

- Date: Create a Date sequence vector
- POSIXct: Create a POSIXct sequence vector
- hms: Create a hms sequence vector

Examples

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

`dtt_set_time` *Time*

Description

Coerces vectors to floored (and wrapped) hms vectors.

Usage

```
dtt_set_time(x, value)

dtt_time(x, ...)

dtt_time(x) <- value

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

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

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

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

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

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

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

Arguments

`x`	A vector.
`value`	A time 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_set_tz` *Set Time Zone*

Description

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

Usage

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

## S3 method for class 'POSIXct'
dtt_set_tz(x, tz = dtt_default_tz(), ...)
```

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 *Study Year*

Description

Study Year

Usage

```
dtt_study_year(x, start = 1L, full = TRUE)
```

Arguments

`x`	A Date or POSIXct vector.
`start`	An integer vector of the starting month or a Date vector of the starting date.
`full`	A flag specifying whether to return a character vector of the study years (or an integer vector of the first year)

Value

A character vector of the study year or an integer vector of the first year.

Examples

```
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")
```

dtt_sys_date *Get System Date*

Description

Get System Date

Usage

```
dtt_sys_date(tz = dtt_default_tz())
```

Arguments

tz	A string of the time zone.

Value

A floored Date scalar.

Examples

```
## Not run:
dtt_set_default_tz("Etc/GMT+12")
dtt_sys_date()
dtt_set_default_tz("Etc/GMT-12")
dtt_sys_date()
dtt_sys_date(tz = "Etc/GMT+12")

## End(Not run)
```

dtt_sys_date_time *Get System Date Time*

Description

Get System Date Time

Usage

```
dtt_sys_date_time(tz = dtt_default_tz())
```

Arguments

tz	A string of the time zone.

Value

A floored POSIXct scalar.

Examples

```
## Not run:
dtt_set_default_tz("UTC")
dtt_sys_date_time()
dtt_set_default_tz("Etc/GMT+8")
dtt_sys_date_time()
dtt_sys_date_time(tz = "UTC")

## End(Not run)
```

---

dtt_sys_time *Get System Time*

Description

Get System Time

Usage

```
dtt_sys_time(tz = dtt_default_tz())
```

Arguments

tz	A string of the time zone.

Value

A floored hms scalar.

Examples

```
## Not run:
dtt_sys_time()

## End(Not run)
```

dtt_sys_tz *Get, Set or Reset Default Time Zone*

Description

Get, Set or Reset Default Time Zone

Usage

```
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

```
## 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_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.
...	Unused.

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"))
```

dtt_units *Get Units*

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

```
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

```
dtt_units_per_unit("hours")
```

---

dtt_wday *Get Week Day*

Description

Gets the week days for the locale.

Usage

```
dtt_wday(x, abbr = FALSE, ...)

## Default S3 method:
dtt_wday(x, abbr = FALSE, ...)
```

Arguments

x	A date/time vector.
abbr	A flag specifying whether to abbreviate the week days.
...	Unused.

Value

An character vector of the week days.

Methods (by class)

- `default`: Get character vector of week days for a Date vector

Examples

```
x <- as.Date("1990-01-02")
dtt_wday(x)

x <- as.POSIXct("1990-01-02 23:40:51")
dtt_wday(x, abbr = TRUE)
```

dtt_wrap *Wrap*

Description

Wrap

Usage

```
dtt_wrap(x, ...)
```

Arguments

x	A date/time vector.
...	Unused.

Examples

```
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

```
dtt_years(x, ...)

dtt_years(x) <- value

dtt_set_year(x, value)

dtt_year(x, ...)

dtt_year(x) <- value

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

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

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

## 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

```
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
```

---

is_date_time *Is Date/Time*

Description

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

Usage

```
is.POSIXct(x)

is_date_time(x)

is.Date(x)

is_date(x)

is.hms(x)

is_time(x)
```

Arguments

x	An object

Value

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

NA_Date_ *Missing Date*

Description

A missing Date object

Usage

```
NA_Date_
```

Format

An object of class `Date` of length 1.

NA_hms_ *Missing hms*

Description

A missing hms object

Usage

```
NA_hms_
```

Format

An object of class `hms` (inherits from `difftime`) of length 1.

NA_POSIXct_ *Missing POSIXct*

Description

A missing POSIXct object

Usage

```
NA_POSIXct_
```

Format

An object of class `POSIXct` (inherits from `POSIXt`) of length 1.

vld_time *Validate Time*

Description

Validates that an object is scalar hms::hms object using `inherits(x,class) && length(x) == 1L && !anyNA(x)`.

Usage

```
vld_time(x)
```

Arguments

`x`	The object to check.

Value

A flag indicating whether the condition was met.

See Also

chk_time()

Examples

```
vld_time(1)
vld_time(hms::as_hms("10:12:59"))
```

Index