

# Package ‘dtr2’

September 19, 2021

**Title** Manipulate Date, POSIXct and hms Vectors

**Version** 0.4.0

**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/dtr2>

**BugReports** <https://github.com/poissonconsulting/dtr2/issues>

**Depends** R (>= 3.3)

**Imports** chk, hms, lifecycle

**Suggests** covr, rlang, testthat (>= 3.0.0)

**RdMacros** lifecycle

**Config/testthat/edition** 3

**Encoding** UTF-8

**Language** en-US

**RoxygenNote** 7.1.2

**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** 2021-09-19 04:40:09 UTC

## R topics documented:

check_tz . . . . .	3
chk_time . . . . .	4
dtt . . . . .	4

dtc_add_units . . . . .	5
dtc_adjust_tz . . . . .	6
dtc_adjust_units . . . . .	7
dtc_aggregate . . . . .	7
dtc_complete . . . . .	8
dtc_completed . . . . .	10
dtc_date . . . . .	11
dtc_date_add_time . . . . .	13
dtc_date_time . . . . .	13
dtc_day . . . . .	15
dtc_days_in_month . . . . .	16
dtc_days_in_year . . . . .	17
dtc_dayte . . . . .	17
dtc_dayte_time . . . . .	18
dtc_daytt . . . . .	19
dtc_day_decimal . . . . .	20
dtc_decade . . . . .	21
dtc_diff . . . . .	21
dtc_doy . . . . .	22
dtc_doy_decimal . . . . .	23
dtc_doy_to_date . . . . .	23
dtc_feb29_to_28 . . . . .	24
dtc_floor . . . . .	25
dtc_floored . . . . .	26
dtc_hours . . . . .	27
dtc_hour_decimal . . . . .	28
dtc_is_date . . . . .	29
dtc_is_date_time . . . . .	30
dtc_is_dtc . . . . .	30
dtc_leap_year . . . . .	31
dtc_minutes . . . . .	31
dtc_minute_decimal . . . . .	33
dtc_months . . . . .	34
dtc_month_decimal . . . . .	35
dtc_season . . . . .	36
dtc_seconds . . . . .	37
dtc_seq . . . . .	38
dtc_set_time . . . . .	39
dtc_set_tz . . . . .	41
dtc_study_year . . . . .	42
dtc_subtract_units . . . . .	42
dtc_sys_date . . . . .	43
dtc_sys_date_time . . . . .	44
dtc_sys_time . . . . .	45
dtc_sys_tz . . . . .	45
dtc_tz . . . . .	46
dtc_units . . . . .	47
dtc_units_per_unit . . . . .	48

`check_tz` 3

<code>dtw_wday</code> . . . . .	49
<code>dtw_wrap</code> . . . . .	49
<code>dtw_years</code> . . . . .	50
<code>dtw_year_decimal</code> . . . . .	51
<code>is_date_time</code> . . . . .	52
<code>NA_Date_</code> . . . . .	53
<code>NA_hms_</code> . . . . .	53
<code>NA_POSIXct_</code> . . . . .	53
<code>vld_time</code> . . . . .	54

**Index** 55

---

`check_tz`                      *Check Time Zone*

---

### Description

Checks an object's time zone as returned by `dtw_tz()`.

### Usage

```
check_tz(x, tz = dtw_tz(x), x_name = substitute(x), error = TRUE)
```

### Arguments

<code>x</code>	The object to check.
<code>tz</code>	A string of the time zone to check that <code>x</code> 's matches.
<code>x_name</code>	A string of the name of the object.
<code>error</code>	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

[dtw\\_tz\(\)](#)

Other check: [chk\\_time\(\)](#)

### 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.

### See Also

Other check: [check\\_tz\(\)](#)

### 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).

---

dtc_add_units	<i>Add Time Units</i>
---------------	-----------------------

---

### Description

Add time units to a date time vector.

### Usage

```
dtc_add_units(x, units, n = 1L)
```

```
dtc_add_years(x, n = 1L, ...)
```

```
dtc_add_months(x, n = 1L, ...)
```

```
dtc_add_days(x, n = 1L, ...)
```

```
dtc_add_hours(x, n = 1L, ...)
```

```
dtc_add_minutes(x, n = 1L, ...)
```

```
dtc_add_seconds(x, n = 1L, ...)
```

### Arguments

x	A date/time vector.
units	A string of the time units. The possible values are "secs", "minutes", "hours", "days" or "weeks".
n	An integer of the number of units.
...	Unused.

### Value

The modified date time vector.

### See Also

[dtc\\_subtract\\_units\(\)](#)

Other add: [dtc\\_date\\_add\\_time\(\)](#)

### Examples

```
dtc_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\(\)](#)

Other tz: [dtt\\_set\\_tz\(\)](#), [dtt\\_sys\\_tz\(\)](#), [dtt\\_tz\(\)](#)

### Examples

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

---

dtc_adjust_units	<i>Adjust Units</i>
------------------	---------------------

---

**Description**

Adjust Units

**Usage**

```
dtc_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.

**See Also**

[dtc\\_add\\_units\(\)](#)

[dtc\\_subtract\\_units\(\)](#)

Other units: [dtc\\_units\\_per\\_unit\(\)](#), [dtc\\_units\(\)](#)

**Examples**

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

---

dtc_aggregate	<i>Aggregate Date/Time</i>
---------------	----------------------------

---

**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 time units. The possible values are "secs", "minutes", "hours", "days" or "weeks".
...	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**

```
dtc_complete(x, ...)

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

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

## S3 method for class 'hms'
dtc_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 scalar of the end.
units	A string of the time units. The possible values are "secs", "minutes", "hours", "days" or "weeks".
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

**See Also**

Other complete: [dtt\\_completed\(\)](#)

**Examples**

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

---

dtt_completed	<i>Completed</i>
---------------	------------------

---

**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 time units. The possible values are "secs", "minutes", "hours", "days" or "weeks".
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

**See Also**

Other complete: [dtt\\_complete\(\)](#)

---

dtt\_date

*Floor 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)
```

### 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

### See Also

Other floor: [dtt\\_date\\_time\(\)](#), [dtt\\_floored\(\)](#), [dtt\\_floor\(\)](#), [dtt\\_set\\_time\(\)](#)

### 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      *Add Time to Date*

---

**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 of the time.
tz	A string of the time zone.

**Value**

A POSIXct vector.

**See Also**

Other add: [dtt\\_add\\_units\(\)](#)

**Examples**

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

---

dtt\_date\_time      *Floor 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

### See Also

Other floor: [dtt\\_date\(\)](#), [dtt\\_floored\(\)](#), [dtt\\_floor\(\)](#), [dtt\\_set\\_time\(\)](#)

### 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"))
```

---

dtc\_day *Get and Set Day Values*

---

**Description**

Gets and sets day values for date/time vectors.

**Usage**

```
dtc_day(x, ...)  
  
dtc_day(x) <- value  
  
## S3 method for class 'Date'  
dtc_day(x, ...)  
  
## S3 method for class 'POSIXct'  
dtc_day(x, ...)  
  
## S3 replacement method for class 'Date'  
dtc_day(x) <- value  
  
## S3 replacement method for class 'POSIXct'  
dtc_day(x) <- value  
  
dtc_days(x, ...)  
  
dtc_days(x) <- value  
  
dtc_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

**See Also**

[dtt\\_day\\_decimal\(\)](#)

Other set date: [dtt\\_months\(\)](#), [dtt\\_years\(\)](#)

**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_days_in_month	<i>Days in the Month</i>
-------------------	--------------------------

---

**Description**

Days in the Month

**Usage**

```
dtt_days_in_month(x)
```

**Arguments**

x                    A Date or POSIXct vector.

**Value**

A integer vector of 28 to 31 indicating the days in the month.

**See Also**

Other days: [dtt\\_days\\_in\\_year\(\)](#), [dtt\\_doy\\_to\\_date\(\)](#), [dtt\\_doy\(\)](#)

**Examples**

```
dtt_days_in_month(as.Date(c("2000-02-11", "2001-02-01")))
```



---

dtt_days_in_year	<i>Days in the Year</i>
------------------	-------------------------

---

**Description**

Days in the Year

**Usage**

```
dtt_days_in_year(x)
```

**Arguments**

x                    A Date or POSIXct vector.

**Value**

A integer vector of 365 and 366 indicates the days of the year.

**See Also**

Other days: [dtt\\_days\\_in\\_month\(\)](#), [dtt\\_doy\\_to\\_date\(\)](#), [dtt\\_doy\(\)](#)

**Examples**

```
dtt_days_in_year(as.Date(c("2000-10-11", "2001-01-01")))
```

---

dtt_dayte	<i>Dayte</i>
-----------	--------------

---

**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

**See Also**

Other dayte: [dtt\\_dayte\\_time\(\)](#), [dtt\\_daytt\(\)](#)

**Examples**

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

---

dtt_dayte_time	<i>Dayte Time</i>
----------------	-------------------

---

**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

**See Also**

Other dayte: [dtt\\_dayte\(\)](#), [dtt\\_daytt\(\)](#)

**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 of the starting month or a Date vector of the starting date.

**Value**

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

**See Also**

Other dayte: [dtt\\_dayte\\_time\(\)](#), [dtt\\_dayte\(\)](#)

---

dtt_day_decimal	<i>Get Decimal Day Values</i>
-----------------	-------------------------------

---

### Description

Gets decimal day values for date/time vectors.

### Usage

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

### Arguments

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

### Value

A numeric vector.

### Methods (by class)

- Date: Get numeric vector of decimal year values for a Date vector
- POSIXct: Get numeric vector of decimal year values for a POSIXct vector

### See Also

[dtt\\_day\(\)](#)

Other decimal: [dtt\\_doy\\_decimal\(\)](#), [dtt\\_hour\\_decimal\(\)](#), [dtt\\_minute\\_decimal\(\)](#), [dtt\\_month\\_decimal\(\)](#), [dtt\\_year\\_decimal\(\)](#)

### Examples

```
x <- as.POSIXct("1990-01-03 10:00:01")  
dtt_day_decimal(x)
```

---

dtt_decade	<i>Decade</i>
------------	---------------

---

**Description**

Decade

**Usage**

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

**Arguments**

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

**Value**

A integer vector of the decade.

**Methods (by class)**

- Date: Decade a Date vector

**Examples**

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

---

dtt_diff	<i>Time Difference</i>
----------	------------------------

---

**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 <code>dtt_date_time()</code> .
y	An object that can be coerced to a POSIXct using <code>dtt_date_time()</code> .
units	A string of the time 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	<i>Day of the Year</i>
---------	------------------------

---

**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.

**See Also**

[dtt\\_doy\\_decimal\(\)](#)

Other days: [dtt\\_days\\_in\\_month\(\)](#), [dtt\\_days\\_in\\_year\(\)](#), [dtt\\_doy\\_to\\_date\(\)](#)

**Examples**

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

---

dtt_doy_decimal	<i>Day of the Year Decimal</i>
-----------------	--------------------------------

---

**Description**

Day of the Year Decimal

**Usage**

```
dtt_doy_decimal(x, ...)
```

**Arguments**

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

**Value**

A numeric vector between 0 and 366 of the day of the year.

**See Also**

[dtt\\_doy\(\)](#)

Other decimal: [dtt\\_day\\_decimal\(\)](#), [dtt\\_hour\\_decimal\(\)](#), [dtt\\_minute\\_decimal\(\)](#), [dtt\\_month\\_decimal\(\)](#), [dtt\\_year\\_decimal\(\)](#)

**Examples**

```
dtt_doy_decimal(Sys.Date())
```

---

dtt_doy_to_date	<i>Day of the Year to Date</i>
-----------------	--------------------------------

---

**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.

**See Also**

Other days: [dtt\\_days\\_in\\_month\(\)](#), [dtt\\_days\\_in\\_year\(\)](#), [dtt\\_doy\(\)](#)

**Examples**

```
dtt_doy_to_date(3)
```

---

dtt_feb29_to_28	<i>Convert Feb 29 to Feb 28</i>
-----------------	---------------------------------

---

**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.

**See Also**

Other leap year: [dtt\\_leap\\_year\(\)](#)

**Examples**

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



---

dtt_floor	<i>Floor Date/Time</i>
-----------	------------------------

---

## 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 time units. The possible values are "secs", "minutes", "hours", "days" or "weeks".
...	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

## See Also

Other floor: [dtt\\_date\\_time\(\)](#), [dtt\\_date\(\)](#), [dtt\\_floored\(\)](#), [dtt\\_set\\_time\(\)](#)

## Examples

```
dtt_floor(hms::as_hms("23:59:59"), "hours")
```

---

`dtt_floored`*Test 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**

<code>x</code>	A Date, POSIXct or hms vector.
<code>...</code>	Unused.
<code>units</code>	A string of the time units. The possible values are "secs", "minutes", "hours", "days" or "weeks".

**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

**See Also**

Other floor: [dtt\\_date\\_time\(\)](#), [dtt\\_date\(\)](#), [dtt\\_floor\(\)](#), [dtt\\_set\\_time\(\)](#)

**Examples**

```
dtt_floored(as.Date("2002-02-01"))
```

---

dtc_hours	<i>Get and Set Hour Values</i>
-----------	--------------------------------

---

**Description**

Gets and sets hour values for date/time vectors.

**Usage**

```
dtc_hours(x, ...)  
  
dtc_hours(x) <- value  
  
dtc_hour(x, ...)  
  
dtc_hour(x) <- value  
  
## S3 method for class 'Date'  
dtc_hour(x, ...)  
  
## S3 method for class 'POSIXct'  
dtc_hour(x, ...)  
  
## S3 method for class 'hms'  
dtc_hour(x, ...)  
  
## S3 replacement method for class 'POSIXct'  
dtc_hour(x) <- value  
  
## S3 replacement method for class 'hms'  
dtc_hour(x) <- value  
  
dtc_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

**See Also**

[dtt\\_hour\\_decimal\(\)](#)

Other set time: [dtt\\_minutes\(\)](#), [dtt\\_seconds\(\)](#)

**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\_hour\_decimal      *Get Decimal Hour Values*

---

**Description**

Gets decimal hour values for date/time vectors.

**Usage**

```
dtt_hour_decimal(x, ...)
```

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

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

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

**Arguments**

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

**Value**

A numeric vector.

**Methods (by class)**

- Date: Get numeric vector of decimal hour values for a Date vector
- POSIXct: Get numeric vector of decimal hour values for a POSIXct vector
- hms: Get numeric vector of decimal hour values for a hms vector

**See Also**

[dtt\\_hour\(\)](#)

Other decimal: [dtt\\_day\\_decimal\(\)](#), [dtt\\_doy\\_decimal\(\)](#), [dtt\\_minute\\_decimal\(\)](#), [dtt\\_month\\_decimal\(\)](#), [dtt\\_year\\_decimal\(\)](#)

**Examples**

```
x <- as.POSIXct("1990-01-02 23:40:51")
dtt_hour_decimal(x)
x <- hms::as_hms("23:40:51")
dtt_hour_decimal(x)
```

---

dtt_is_date	<i>Is Date</i>
-------------	----------------

---

**Description**

Is Date

**Usage**

```
dtt_is_date(x)
```

**Arguments**

x                    An R object.

**Value**

A flag indicating whether R is a Date vector.

**See Also**

Other is: [dtt\\_is\\_date\\_time\(\)](#), [is\\_date\\_time\(\)](#)

---

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.

**See Also**

Other is: [dtt\\_is\\_date\(\)](#), [is\\_date\\_time\(\)](#)

---

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_leap_year	<i>Test for Leap Year</i>
---------------	---------------------------

---

**Description**

Tests whether each year is a leap year.

**Usage**

```
dtt_leap_year(x)
```

**Arguments**

x                    A date/time vector.

**Value**

A logical vector indicating whether each year is a leap year.

**See Also**

Other leap year: [dtt\\_feb29\\_to\\_28\(\)](#)

**Examples**

```
dtt_leap_year(as.Date("1999-03-04", "2000-02-01"))
```

---

dtt_minutes	<i>Get and Set Minute Values</i>
-------------	----------------------------------

---

**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

### See Also

[dtt\\_minute\\_decimal\(\)](#)

Other set time: [dtt\\_hours\(\)](#), [dtt\\_seconds\(\)](#)

### 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\_minute\_decimal      *Get Decimal Minute Values*

---

### Description

Gets decimal minute values for date/time vectors.

### Usage

```
dtt_minute_decimal(x, ...)  
  
## S3 method for class 'Date'  
dtt_minute_decimal(x, ...)  
  
## S3 method for class 'POSIXct'  
dtt_minute_decimal(x, ...)  
  
## S3 method for class 'hms'  
dtt_minute_decimal(x, ...)
```

### Arguments

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

### Value

A numeric vector.

### Methods (by class)

- Date: Get numeric vector of decimal minute values for a Date vector
- POSIXct: Get numeric vector of decimal minute values for a POSIXct vector
- hms: Get numeric vector of decimal minute values for a hms vector

### See Also

[dtt\\_minute\(\)](#)

Other decimal: [dtt\\_day\\_decimal\(\)](#), [dtt\\_doy\\_decimal\(\)](#), [dtt\\_hour\\_decimal\(\)](#), [dtt\\_month\\_decimal\(\)](#), [dtt\\_year\\_decimal\(\)](#)

### Examples

```
x <- as.POSIXct("1990-01-02 23:40:51")  
dtt_minute_decimal(x)  
x <- hms::as_hms("23:40:51")  
dtt_minute_decimal(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**

<code>x</code>	A date/time vector.
<code>...</code>	Unused.
<code>value</code>	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

**See Also**[dtm\\_month\\_decimal\(\)](#)Other set date: [dtm\\_day\(\)](#), [dtm\\_years\(\)](#)**Examples**

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

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

---

dtm\_month\_decimal      *Get Decimal Month Values*

---

**Description**

Gets decimal month values for date/time vectors.

**Usage**

```
dtm_month_decimal(x, ...)
```

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

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

**Arguments**

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

**Value**

A numeric vector.

**Methods (by class)**

- Date: Get numeric vector of decimal year values for a Date vector
- POSIXct: Get numeric vector of decimal year values for a POSIXct vector

**See Also**

[dtt\\_month\(\)](#)

Other decimal: [dtt\\_day\\_decimal\(\)](#), [dtt\\_doy\\_decimal\(\)](#), [dtt\\_hour\\_decimal\(\)](#), [dtt\\_minute\\_decimal\(\)](#), [dtt\\_year\\_decimal\(\)](#)

**Examples**

```
x <- as.POSIXct("1990-01-03 10:00:01")
dtt_month_decimal(x)
```

---

dtt_season	<i>Season</i>
------------	---------------

---

**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**

<code>x</code>	A Date or POSIXct vector
<code>start</code>	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.
<code>first</code>	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")))
```

---

`dtc_seconds`*Get and Set Second Values*

---

**Description**

Gets and sets second values for date/time vectors.

**Usage**

```
dtc_seconds(x, ...)  
  
dtc_seconds(x) <- value  
  
dtc_second(x, ...)  
  
dtc_second(x) <- value  
  
## S3 method for class 'Date'  
dtc_second(x, ...)  
  
## S3 method for class 'POSIXct'  
dtc_second(x, ...)  
  
## S3 method for class 'hms'  
dtc_second(x, ...)  
  
## S3 replacement method for class 'POSIXct'  
dtc_second(x) <- value  
  
## S3 replacement method for class 'hms'  
dtc_second(x) <- value  
  
dtc_set_second(x, value)
```

**Arguments**

<code>x</code>	A date/time vector.
<code>...</code>	Unused.
<code>value</code>	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

**See Also**

Other set time: [dtt\\_hours\(\)](#), [dtt\\_minutes\(\)](#)

**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. The possible values are "secs", "minutes", "hours", "days" or "weeks".
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	<i>Floor Time</i>
--------------	-------------------

---

### 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, ...)

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

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

### 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
- Date: Set time values for a Date vector
- POSIXct: Set time values for a POSIXct vector



**See Also**

Other floor: [dtt\\_date\\_time\(\)](#), [dtt\\_date\(\)](#), [dtt\\_floored\(\)](#), [dtt\\_floor\(\)](#)

**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\(\)](#)

Other tz: [dtt\\_adjust\\_tz\(\)](#), [dtt\\_sys\\_tz\(\)](#), [dtt\\_tz\(\)](#)

**Examples**

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

---

dtt_study_year	<i>Study Year</i>
----------------	-------------------

---

**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	<i>Subtract Time Units</i>
--------------------	----------------------------

---

**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)
```

dtc\_subtract\_minutes(x, n = 1L)

dtc\_subtract\_seconds(x, n = 1L)

**Arguments**

- x                    A date/time vector.
- n                    An integer of the number of units.
- units                A string of the time units. The possible values are "secs", "minutes", "hours", "days" or "weeks".

**Value**

The modified date time vector.

**See Also**

[dtc\\_add\\_units\(\)](#)

**Examples**

dtc\_subtract\_units(as.Date("1999-12-31"), 2L, "days")

dtc\_sys\_date                    *Get System Date*

**Description**

Get System Date

**Usage**

dtc\_sys\_date(tz = dtc\_default\_tz())

**Arguments**

- tz                    A string of the time zone.

**Value**

A floored Date scalar.

**See Also**

Other sys: [dtc\\_sys\\_date\\_time\(\)](#), [dtc\\_sys\\_time\(\)](#)

**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	<i>Get System Date Time</i>
-------------------	-----------------------------

---

**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.

**See Also**

Other sys: [dtt\\_sys\\_date\(\)](#), [dtt\\_sys\\_time\(\)](#)

**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	<i>Get System Time</i>
--------------	------------------------

---

**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.

**See Also**

Other sys: [dtt\\_sys\\_date\\_time\(\)](#), [dtt\\_sys\\_date\(\)](#)

**Examples**

```
## Not run:  
dtt_sys_time()  
  
## End(Not run)
```

---

dtt_sys_tz	<i>Get, Set or Reset Default Time Zone</i>
------------	--

---

**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

### See Also

Other tz: [dtt\\_adjust\\_tz\(\)](#), [dtt\\_set\\_tz\(\)](#), [dtt\\_tz\(\)](#)

### 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.

**See Also**

Other tz: [dtc\\_adjust\\_tz\(\)](#), [dtc\\_set\\_tz\(\)](#), [dtc\\_sys\\_tz\(\)](#)

**Examples**

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

---

dtc\_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**

```
dtc_units(x, ...)

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

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

## S3 method for class 'hms'
dtc_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

**See Also**

Other units: [dtt\\_adjust\\_units\(\)](#), [dtt\\_units\\_per\\_unit\(\)](#)

**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	<i>Units per Unit</i>
--------------------	-----------------------

---

**Description**

Units per Unit

**Usage**

```
dtt_units_per_unit(units = "seconds", unit = "days")
```

**Arguments**

units	A string of the time units. The possible values are "secs", "minutes", "hours", "days" or "weeks".
unit	A string of the time unit.

**Value**

A number of the units per unit

**See Also**

Other units: [dtt\\_adjust\\_units\(\)](#), [dtt\\_units\(\)](#)

**Examples**

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



---

dtt_wday	<i>Get Week Day</i>
----------	---------------------

---

**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	<i>Wrap</i>
----------	-------------

---

**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

**See Also**

[dtt\\_year\\_decimal\(\)](#)

Other set date: [dtt\\_day\(\)](#), [dtt\\_months\(\)](#)

**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
```

---

dtt_year_decimal	<i>Get Decimal Year Values</i>
------------------	--------------------------------

---

**Description**

Gets decimal year values for date/time vectors.

**Usage**

```
dtt_year_decimal(x, ...)
```

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

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

**Arguments**

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

**Value**

A numeric vector.

**Methods (by class)**

- Date: Get numeric vector of decimal year values for a Date vector
- POSIXct: Get numeric vector of decimal year values for a POSIXct vector

**See Also**

[dtt\\_year\(\)](#)

Other decimal: [dtt\\_day\\_decimal\(\)](#), [dtt\\_doy\\_decimal\(\)](#), [dtt\\_hour\\_decimal\(\)](#), [dtt\\_minute\\_decimal\(\)](#), [dtt\\_month\\_decimal\(\)](#)

**Examples**

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

**See Also**

Other is: [dtt\\_is\\_date\\_time\(\)](#), [dtt\\_is\\_date\(\)](#)

---

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

- \* **NA**
    - NA\_POSIXct\_, 53
  - \* **add**
    - dtc\_add\_units, 5
    - dtc\_date\_add\_time, 13
  - \* **check**
    - check\_tz, 3
    - chk\_time, 4
  - \* **complete**
    - dtc\_complete, 8
    - dtc\_completed, 10
  - \* **datasets**
    - NA\_Date\_, 53
    - NA\_hms\_, 53
    - NA\_POSIXct\_, 53
  - \* **days**
    - dtc\_days\_in\_month, 16
    - dtc\_days\_in\_year, 17
    - dtc\_doy, 22
    - dtc\_doy\_to\_date, 23
  - \* **dayte**
    - dtc\_dayte, 17
    - dtc\_dayte\_time, 18
    - dtc\_daytt, 19
  - \* **decimal**
    - dtc\_day\_decimal, 20
    - dtc\_doy\_decimal, 23
    - dtc\_hour\_decimal, 28
    - dtc\_minute\_decimal, 33
    - dtc\_month\_decimal, 35
    - dtc\_year\_decimal, 51
  - \* **floor**
    - dtc\_date, 11
    - dtc\_date\_time, 13
    - dtc\_floor, 25
    - dtc\_floored, 26
    - dtc\_set\_time, 39
  - \* **is**
    - dtc\_is\_date, 29
    - dtc\_is\_date\_time, 30
    - is\_date\_time, 52
  - \* **leap year**
    - dtc\_feb29\_to\_28, 24
    - dtc\_leap\_year, 31
  - \* **set date**
    - dtc\_day, 15
    - dtc\_months, 34
    - dtc\_years, 50
  - \* **set time**
    - dtc\_hours, 27
    - dtc\_minutes, 31
    - dtc\_seconds, 37
  - \* **subtract**
    - dtc\_subtract\_units, 42
  - \* **sys**
    - dtc\_sys\_date, 43
    - dtc\_sys\_date\_time, 44
    - dtc\_sys\_time, 45
  - \* **tz**
    - dtc\_adjust\_tz, 6
    - dtc\_set\_tz, 41
    - dtc\_sys\_tz, 45
    - dtc\_tz, 46
  - \* **units**
    - dtc\_adjust\_units, 7
    - dtc\_units, 47
    - dtc\_units\_per\_unit, 48
- check\_tz, 3, 4  
chk\_time, 3, 4  
chk\_time(), 54
- dtc, 4  
dtc\_add\_days (dtc\_add\_units), 5  
dtc\_add\_hours (dtc\_add\_units), 5  
dtc\_add\_minutes (dtc\_add\_units), 5  
dtc\_add\_months (dtc\_add\_units), 5  
dtc\_add\_seconds (dtc\_add\_units), 5  
dtc\_add\_units, 5, 13

dtt\_add\_units(), 7, 43  
 dtt\_add\_years (dtt\_add\_units), 5  
 dtt\_adjust\_tz, 6, 41, 46, 47  
 dtt\_adjust\_tz(), 41  
 dtt\_adjust\_units, 7, 48  
 dtt\_aggregate, 7  
 dtt\_complete, 8, 11  
 dtt\_completed, 10, 10  
 dtt\_date, 11, 14, 25, 26, 41  
 dtt\_date<- (dtt\_date), 11  
 dtt\_date\_add\_time, 5, 13  
 dtt\_date\_time, 12, 13, 25, 26, 41  
 dtt\_day, 15, 35, 51  
 dtt\_day(), 20  
 dtt\_day<- (dtt\_day), 15  
 dtt\_day\_decimal, 20, 23, 29, 33, 36, 52  
 dtt\_day\_decimal(), 16  
 dtt\_days (dtt\_day), 15  
 dtt\_days<- (dtt\_day), 15  
 dtt\_days\_in\_month, 16, 17, 22, 24  
 dtt\_days\_in\_year, 16, 17, 22, 24  
 dtt\_dayte, 17, 19  
 dtt\_dayte\_time, 18, 18, 19  
 dtt\_daytt, 18, 19, 19  
 dtt\_decade, 21  
 dtt\_default\_tz (dtt\_sys\_tz), 45  
 dtt\_diff, 21  
 dtt\_doy, 16, 17, 22, 24  
 dtt\_doy(), 23  
 dtt\_doy\_decimal, 20, 23, 29, 33, 36, 52  
 dtt\_doy\_decimal(), 22  
 dtt\_doy\_to\_date, 16, 17, 22, 23  
 dtt\_feb29\_to\_28, 24, 31  
 dtt\_floor, 12, 14, 25, 26, 41  
 dtt\_floored, 12, 14, 25, 26, 41  
 dtt\_hour (dtt\_hours), 27  
 dtt\_hour(), 29  
 dtt\_hour<- (dtt\_hours), 27  
 dtt\_hour\_decimal, 20, 23, 28, 33, 36, 52  
 dtt\_hour\_decimal(), 28  
 dtt\_hours, 27, 32, 38  
 dtt\_hours<- (dtt\_hours), 27  
 dtt\_is\_date, 29, 30, 53  
 dtt\_is\_date\_time, 29, 30, 53  
 dtt\_is\_dtt, 30  
 dtt\_leap\_year, 24, 31  
 dtt\_minute (dtt\_minutes), 31  
 dtt\_minute(), 33  
 dtt\_minute<- (dtt\_minutes), 31  
 dtt\_minute\_decimal, 20, 23, 29, 33, 36, 52  
 dtt\_minute\_decimal(), 32  
 dtt\_minutes, 28, 31, 38  
 dtt\_minutes<- (dtt\_minutes), 31  
 dtt\_month (dtt\_months), 34  
 dtt\_month(), 36  
 dtt\_month<- (dtt\_months), 34  
 dtt\_month\_decimal, 20, 23, 29, 33, 35, 52  
 dtt\_month\_decimal(), 35  
 dtt\_months, 16, 34, 51  
 dtt\_months<- (dtt\_months), 34  
 dtt\_reset\_default\_tz (dtt\_sys\_tz), 45  
 dtt\_reset\_sys\_tz (dtt\_sys\_tz), 45  
 dtt\_season, 36  
 dtt\_second (dtt\_seconds), 37  
 dtt\_second<- (dtt\_seconds), 37  
 dtt\_seconds, 28, 32, 37  
 dtt\_seconds<- (dtt\_seconds), 37  
 dtt\_seq, 38  
 dtt\_set\_date (dtt\_date), 11  
 dtt\_set\_day (dtt\_day), 15  
 dtt\_set\_default\_tz (dtt\_sys\_tz), 45  
 dtt\_set\_hour (dtt\_hours), 27  
 dtt\_set\_minute (dtt\_minutes), 31  
 dtt\_set\_month (dtt\_months), 34  
 dtt\_set\_second (dtt\_seconds), 37  
 dtt\_set\_sys\_tz (dtt\_sys\_tz), 45  
 dtt\_set\_time, 12, 14, 25, 26, 39  
 dtt\_set\_tz, 6, 41, 46, 47  
 dtt\_set\_tz(), 6  
 dtt\_set\_year (dtt\_years), 50  
 dtt\_study\_year, 42  
 dtt\_subtract\_days (dtt\_subtract\_units),  
     42  
 dtt\_subtract\_hours  
     (dtt\_subtract\_units), 42  
 dtt\_subtract\_minutes  
     (dtt\_subtract\_units), 42  
 dtt\_subtract\_months  
     (dtt\_subtract\_units), 42  
 dtt\_subtract\_seconds  
     (dtt\_subtract\_units), 42  
 dtt\_subtract\_units, 42  
 dtt\_subtract\_units(), 5, 7  
 dtt\_subtract\_years  
     (dtt\_subtract\_units), 42  
 dtt\_sys\_date, 43, 44, 45



`dtc_sys_date_time`, 43, 44, 45  
`dtc_sys_time`, 43, 44, 45  
`dtc_sys_tz`, 6, 41, 45, 47  
`dtc_time` (`dtc_set_time`), 39  
`dtc_time`<- (`dtc_set_time`), 39  
`dtc_tz`, 6, 41, 46, 46  
`dtc_tz()`, 3  
`dtc_units`, 7, 47, 48  
`dtc_units_per_unit`, 7, 48, 48  
`dtc_wday`, 49  
`dtc_wrap`, 49  
`dtc_year` (`dtc_years`), 50  
`dtc_year()`, 52  
`dtc_year`<- (`dtc_years`), 50  
`dtc_year_decimal`, 20, 23, 29, 33, 36, 51  
`dtc_year_decimal()`, 51  
`dtc_years`, 16, 35, 50  
`dtc_years`<- (`dtc_years`), 50

`hms::hms`, 54

`is.Date` (`is_date_time`), 52  
`is.hms` (`is_date_time`), 52  
`is.POSIXct` (`is_date_time`), 52  
`is_date` (`is_date_time`), 52  
`is_date_time`, 29, 30, 52  
`is_time` (`is_date_time`), 52

`NA_Date_`, 53  
`NA_hms_`, 53  
`NA_POSIXct_`, 53

`vld_time`, 54  
`vld_time()`, 4