

# Package: hmstimer (via r-universe)

June 20, 2024

**Title** 'hms' Based Timer

**Version** 0.2.1

**Description** Tracks elapsed clock time using a `hms::hms()` scalar, which if running has an attribute named start that specifies the system time when the timer was started. The elapsed time is the value of the scalar plus the difference between the current system time and the system time when the timer was started.

**License** MIT + file LICENSE

**Depends** R (>= 3.4)

**Imports** hms

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

**Config/testthat/edition** 3

**Encoding** UTF-8

**Language** en-US

**RoxygenNote** 7.2.1

**NeedsCompilation** no

**Author** Joe Thorley [aut, cre]  
(<<https://orcid.org/0000-0002-7683-4592>>), Nadine Hussein [ctb]  
(<<https://orcid.org/0000-0003-4470-8361>>), Poisson Consulting  
[cph, fnd]

**Maintainer** Joe Thorley <joe@poissonconsulting.ca>

**Date/Publication** 2022-09-20 07:46:22 UTC

**Repository** <https://joethorley.r-universe.dev>

**RemoteUrl** <https://github.com/cran/hmstimer>

**RemoteRef** HEAD

**RemoteSha** 253d927a9e47b637a99ae2dbb6d72e17f60bfdea

## Contents

hms_timer . . . . .	2
tmr_ceiling . . . . .	3
tmr_elapsed . . . . .	3
tmr_floor . . . . .	4
tmr_format . . . . .	5
tmr_is_started . . . . .	6
tmr_is_stopped . . . . .	6
tmr_print . . . . .	7
tmr_reset . . . . .	8
tmr_round . . . . .	8
tmr_start . . . . .	9
tmr_stop . . . . .	10
tmr_timer . . . . .	11
<b>Index</b>	<b>12</b>

---

hms_timer	<i>hms Timer</i>
-----------	------------------

---

## Description

A hms Timer is a `hms::hms()` scalar which if running has an attribute named `start` that specifies the system time when the timer was started.

## Details

The elapsed time is the value of the scalar plus the difference between the current system time and the system time when the timer was started.

## Examples

```
str(tmr_timer())
str(tmr_timer(1.5, start = TRUE))

x <- tmr_timer(1, start = TRUE)
print(x)
Sys.sleep(0.1)
print(x)
print(tmr_elapsed(x))
print(x)
```

---

tmr_ceiling	<i>Ceiling hms Timer</i>
-------------	--------------------------

---

**Description**

Rounds a `hms_timer()` up to the nearest second.

**Usage**

```
tmr_ceiling(x)
```

**Arguments**

x                   A `hms_timer()`.

**Value**

A `hms_timer()`.

**See Also**

Other round: `tmr_floor()`, `tmr_format()`, `tmr_round()`

**Examples**

```
tmr_ceiling(tmr_timer(18.9))  
tmr_ceiling(tmr_timer(122.1))
```

---

tmr_elapsed	<i>Elapsed Time hms Timer</i>
-------------	-------------------------------

---

**Description**

Returns the elapsed time for a `hms_timer()` as a `hms_timer()`.

**Usage**

```
tmr_elapsed(x)
```

**Arguments**

x                   A `hms_timer()`.

### Details

The elapsed time is the value of the scalar plus the difference between the current system time and the system time when the timer was started.

If the original `hms_timer()` was running then the new `hms_timer()` is assigned an attribute named `start` of the current system time.

### Value

A `hms_timer()` of the elapsed time.

### See Also

Other start\_stop: `tmr_is_started()`, `tmr_is_stopped()`, `tmr_print()`, `tmr_reset()`, `tmr_start()`, `tmr_stop()`, `tmr_timer()`

### Examples

```
tmr <- tmr_start(tmr_timer())
print(tmr_elapsed(tmr))
Sys.sleep(0.01)
print(tmr_elapsed(tmr))
tmr <- tmr_stop(tmr)
print(tmr_elapsed(tmr))
Sys.sleep(0.01)
print(tmr_elapsed(tmr))
```

---

tmr\_floor

*Floor hms Timer*

---

### Description

Rounds a `hms_timer()` down to the nearest second.

### Usage

```
tmr_floor(x)
```

### Arguments

x                   A `hms_timer()`.

### Value

A `hms_timer()`.

### See Also

Other round: `tmr_ceiling()`, `tmr_format()`, `tmr_round()`

**Examples**

```
tmr_floor(tmr_timer(18.9))  
tmr_floor(tmr_timer(122.1))
```

---

tmr\_format

*Format hms Timer*

---

**Description**

Converts a [hms\\_timer\(\)](#) to a string of the clock time after rounding it to the number of digits.

**Usage**

```
tmr_format(x, digits = 3)
```

**Arguments**

x                   A [hms\\_timer\(\)](#).  
digits               A whole number of the number of decimal places.

**Details**

Negative values of digit are not permitted.

**Value**

A character string.

**See Also**

Other round: [tmr\\_ceiling\(\)](#), [tmr\\_floor\(\)](#), [tmr\\_round\(\)](#)

**Examples**

```
tmr_format(tmr_timer(61.66))  
tmr_format(tmr_timer(61.66), digits = 0)
```

---

tmr_is_started	<i>Is hms Timer Started</i>
----------------	-----------------------------

---

**Description**

Tests if a `hms_timer()` is started (as indicated by the presence of an attribute named start).

**Usage**

```
tmr_is_started(x)
```

**Arguments**

x            A `hms_timer()`.

**Value**

A flag (TRUE or FALSE).

**See Also**

Other start\_stop: `tmr_elapsed()`, `tmr_is_stopped()`, `tmr_print()`, `tmr_reset()`, `tmr_start()`, `tmr_stop()`, `tmr_timer()`

**Examples**

```
tmr <- tmr_timer(start = TRUE)
print(tmr_is_started(tmr))
tmr <- tmr_stop(tmr)
print(tmr_is_started(tmr))
```

---

tmr_is_stopped	<i>Is hms Timer Stopped</i>
----------------	-----------------------------

---

**Description**

Tests if a `hms_timer()` is stopped (as indicated by the absence of an attribute named start).

**Usage**

```
tmr_is_stopped(x)
```

**Arguments**

x            A `hms_timer()`.

**Value**

A flag.

**See Also**

Other start\_stop: [tmr\\_elapsed\(\)](#), [tmr\\_is\\_started\(\)](#), [tmr\\_print\(\)](#), [tmr\\_reset\(\)](#), [tmr\\_start\(\)](#), [tmr\\_stop\(\)](#), [tmr\\_timer\(\)](#)

**Examples**

```
tmr <- tmr_timer(start = TRUE)
print(tmr_is_stopped(tmr))
tmr <- tmr_stop(tmr)
print(tmr_is_stopped(tmr))
```

---

tmr\_print

*Print hms Timer*

---

**Description**

Returns the elapsed time for a [hms\\_timer\(\)](#) from the system time when the timer was started and the current system time as an hms time.

**Usage**

```
tmr_print(x)
```

**Arguments**

x            A [hms\\_timer\(\)](#).

**Details**

The elapsed time is the value of the scalar plus the difference between the current system time and the system time when the timer was started.

**Value**

A character string.

**See Also**

Other start\_stop: [tmr\\_elapsed\(\)](#), [tmr\\_is\\_started\(\)](#), [tmr\\_is\\_stopped\(\)](#), [tmr\\_reset\(\)](#), [tmr\\_start\(\)](#), [tmr\\_stop\(\)](#), [tmr\\_timer\(\)](#)

**Examples**

```
x <- tmr_start(tmr_timer())
tmr_print(x)
```

---

tmr_reset	<i>Reset hms Timer</i>
-----------	------------------------

---

**Description**

Resets a `hms_timer()` by creating a new one.

**Usage**

```
tmr_reset(x, seconds = 0)
```

**Arguments**

x	A <code>hms_timer()</code> .
seconds	A non-negative numeric scalar of the initial number of seconds.

**Value**

A `hms_timer()`.

**See Also**

Other start\_stop: `tmr_elapsed()`, `tmr_is_started()`, `tmr_is_stopped()`, `tmr_print()`, `tmr_start()`, `tmr_stop()`, `tmr_timer()`

**Examples**

```
tmr <- tmr_timer(10)
print(tmr)
tmr_reset(tmr)
```

---

tmr_round	<i>Round hms Timer</i>
-----------	------------------------

---

**Description**

Rounds a `hms_timer()` after updating it to the elapsed time.

**Usage**

```
tmr_round(x, digits = 0)
```

**Arguments**

x	A <code>hms_timer()</code> .
digits	A whole number of the number of decimal places.



**Details**

Negative values of digit are permitted.

**Value**

A `hms_timer()`.

**See Also**

Other round: `tmr_ceiling()`, `tmr_floor()`, `tmr_format()`

**Examples**

```
tmr_round(tmr_timer(18.9))
tmr_round(tmr_timer(18.9), 1)
tmr_round(tmr_timer(18.9), -1)
tmr_round(tmr_timer(121), -2) # 121 is rounded to 100 seconds
```

---

tmr\_start

*Start hms Timer*


---

**Description**

Starts a `hms_timer()` by adding an attribute named start of the current system time.

**Usage**

```
tmr_start(x)
```

**Arguments**

x                   A `hms_timer()`.

**Details**

If the `hms_timer()` is already started, the function simply issues a warning and returns the original object.

**Value**

A started `hms_timer()`.

**See Also**

Other start\_stop: `tmr_elapsed()`, `tmr_is_started()`, `tmr_is_stopped()`, `tmr_print()`, `tmr_reset()`, `tmr_stop()`, `tmr_timer()`

## Examples

```
tmr <- tmr_start(tmr_timer())
print(tmr_elapsed(tmr))
Sys.sleep(0.01)
print(tmr_elapsed(tmr))
```

---

tmr\_stop

*Stop hms Timer*

---

## Description

Stops a `hms_timer()` after updating it to the elapsed time.

## Usage

```
tmr_stop(x)
```

## Arguments

x                   A `hms_timer()`.

## Details

If the `hms_timer()` is already stopped, the function simply issues a warning and returns the original object.

## Value

A stopped `hms_timer()`.

## See Also

Other start\_stop: `tmr_elapsed()`, `tmr_is_started()`, `tmr_is_stopped()`, `tmr_print()`, `tmr_reset()`, `tmr_start()`, `tmr_timer()`

## Examples

```
tmr <- tmr_stop(tmr_timer(start = TRUE))
print(tmr_elapsed(tmr))
Sys.sleep(0.01)
print(tmr_elapsed(tmr))
```

---

tmr_timer	<i>Create hms Timer</i>
-----------	-------------------------

---

**Description**

Creates a `hms_timer()`.

**Usage**

```
tmr_timer(seconds = 0, start = FALSE)
```

**Arguments**

seconds	A non-negative numeric scalar of the initial number of seconds.
start	A flag indicating whether to start the timer.

**Value**

A `hms_timer()`.

**See Also**

Other start\_stop: `tmr_elapsed()`, `tmr_is_started()`, `tmr_is_stopped()`, `tmr_print()`, `tmr_reset()`, `tmr_start()`, `tmr_stop()`

**Examples**

```
tmr <- tmr_timer()
print(tmr)
class(tmr)
```

# Index

## \* **round**

- tmr\_ceiling, 3
- tmr\_floor, 4
- tmr\_format, 5
- tmr\_round, 8

## \* **start\_stop**

- tmr\_elapsed, 3
- tmr\_is\_started, 6
- tmr\_is\_stopped, 6
- tmr\_print, 7
- tmr\_reset, 8
- tmr\_start, 9
- tmr\_stop, 10
- tmr\_timer, 11

hms::hms(), 2

hms\_timer, 2

hms\_timer(), 3–11

tmr\_ceiling, 3, 4, 5, 9

tmr\_elapsed, 3, 6–11

tmr\_floor, 3, 4, 5, 9

tmr\_format, 3, 4, 5, 9

tmr\_is\_started, 4, 6, 7–11

tmr\_is\_stopped, 4, 6, 6, 7–11

tmr\_print, 4, 6, 7, 7, 8–11

tmr\_reset, 4, 6, 7, 8, 9–11

tmr\_round, 3–5, 8

tmr\_start, 4, 6–8, 9, 10, 11

tmr\_stop, 4, 6–9, 10, 11

tmr\_timer, 4, 6–10, 11