# Package: jot (via r-universe)

October 15, 2026

**Title** Jot Down Values for Later

**Version** 0.0.5

**Description** Reproducible work requires a record of where every statistic originated. When writing reports, some data is too big to load in the same environment and some statistics take a while to compute. This package offers a way to keep notes on statistics, simple functions, and small objects. Notepads can be locked to avoid accidental updates. Notepads keep track of who added the notes and when the notes were added. A simple text representation is used to allow for clear version histories.

**License** MIT + file LICENSE

**Config/Needs/website** christopherkenny/ctktemplate

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.2

**Imports** cli, yaml

**URL** http://christophertkenny.com/jot/, https://github.com/christopherkenny/jot

**BugReports** https://github.com/christopherkenny/jot/issues

**Repository** https://christopherkenny.r-universe.dev

**Date/Publication** 2026-05-01 16:15:13 UTC

**RemoteUrl** https://github.com/christopherkenny/jot

**RemoteRef** HEAD

**RemoteSha** 8854249029b26ab0e8e0e1501ed73878daa8f304

## Contents


get_jot_user . . . . . 2
get_sys_time . . . . . 2

jot . . . . . 3
jot_erase . . . . . 4
jot_list . . . . . 5
jot_lock . . . . . 5
jot_new_pad . . . . . 6
jot_open . . . . . 7
jot_read . . . . . 8
jot_skim . . . . . 8

**Index** **10**

---

`get_jot_user` *Get Current User*

---

## Description

Get Current User

## Usage

```
get_jot_user(user = "")
```

## Arguments

| | |
|---|---|
| `user` | user to override get with. Default is `''`, where it checks options, then system info. |

## Value

character

## Examples

```
get_jot_user()
get_jot_user(NA_character_)
```

---

`get_sys_time` *Get System Time as Integer*

---

## Description

Get System Time as Integer

## Usage

```
get_sys_time()
```

## Value

integer

## Examples

```
get_sys_time()
```

---

`jot` *Jot Down a Note*

---

## Description

Jot Down a Note

## Usage

```
jot(
  note,
  name,
  overwrite = FALSE,
  user = get_jot_user(),
  time = get_sys_time(),
  pad = jot_active()
)

write_jot(
  note,
  name,
  overwrite = FALSE,
  user = get_jot_user(),
  time = get_sys_time(),
  pad = jot_active()
)
```

## Arguments

| | |
|---|---|
| `note` | data to save. required. |
| `name` | character, name of data. required. |
| `overwrite` | logical, default is FALSE. If a note of the same name exists, should it be overwritten? |
| `user` | name of user. default is `get_jot_user()`. |
| `time` | system time. default is `get_sys_time()`. |
| `pad` | path to file of pad to write to. default is `jot_active()`. |

## Value

path to pad, invisibly

## Examples

```
tf <- tempfile()
jot_new_pad(tf)
jot_activate(tf)
jot(3, 'example')
```

---

`jot_erase` *Erase a Note*

---

## Description

Erase a Note

## Usage

```
jot_erase(name, pad = jot_active())
```

## Arguments

| | |
|---|---|
| `name` | character, name of note to erase. required. |
| `pad` | path to file of pad to write to. default is `jot_active()`. |

## Value

path to pad, invisibly

## Examples

```
tf <- tempfile()
jot_new_pad(tf)
jot(3, 'example')
jot_erase('example')
```

## jot_list *List out names of values in a jot notepad*

### Description

List out names of values in a jot notepad

### Usage

```
jot_list(pad = jot_active())
```

### Arguments

| | |
|---|---|
| pad | path to notepad to open |

### Value

vlist of names of values in notepad

### Examples

```
tf <- tempfile()
jot_new_pad(tf)
jot_activate(tf)
jot('ex_val', 'ex')
jot_list()
```

## jot_lock *Lock or Unlock a jot Notepad*

### Description

- `jot_lock()` locks a notepad so that `jot` will not write to it.
- `jot_unlock()` unlocks a notepad so that `jot` can write to it.

### Usage

```
jot_lock(pad = jot_active())

jot_unlock(pad = jot_active())
```

### Arguments

| | |
|---|---|
| pad | path to file to open |

## Value

path to pad, invisibly

## Examples

```
tf <- tempfile()
jot_new_pad(tf)
jot_activate(tf)
jot_lock()
jot_unlock()
```

---

`jot_new_pad` *Create new jot Notepad*

---

## Description

Create new jot Notepad

## Usage

```
jot_new_pad(pad = ".jot.yaml", title)

jot_new(pad = ".jot.yaml", title)
```

## Arguments

| | |
|---|---|
| `pad` | path to file to store new notepad. |
| `title` | what to call the notepad. If not supplied, uses the name of the file. |

## Value

path to file, invisibly

## Examples

```
tf <- tempfile()
jot_new_pad(tf)
```

jot_open *Open and Close jot Notepads*

## Description

- `jot_open()` and `jot_activate()` opens an existing notepad and sets it to active.
- `jot_current()` and `jot_active()` return the path to the current notepad.
- `jot_close()` and `jot_deactivate()` closes the open notepad, if there is one.

## Usage

```
jot_open(pad)

jot_activate(pad)

jot_active()

jot_current()

jot_close()

jot_deactivate()
```

## Arguments

| | |
|---|---|
| pad | path to file to open |

## Value

path to pad, invisibly

## Examples

```
tf <- tempfile()
jot_new_pad(tf)
jot_open(tf)
jot_active()
jot_current()
jot_close()
jot_active()
```

## jot_read *Read a jot notepad*

### Description

Read a jot notepad

### Usage

```
jot_read(name, pad = jot_active())

read_jot(name, pad = jot_active())
```

### Arguments

| | |
|---|---|
| name | name of noteto read |
| pad | path to file to open |

### Value

value specified or all values

### Examples

```
tf <- tempfile()
jot_new_pad(tf)
jot_activate(tf)
jot(note = 'ex_note', name = 'ex_name')
jot_read()
jot_read('ex_name')
```

## jot_skim *Read in all values from a jot notepad*

### Description

Read in all values from a jot notepad

### Usage

```
jot_skim(pad = jot_active())
```

### Arguments

| | |
|---|---|
| pad | path to file to open |

## Value

value specified or list of all values

## Examples

```
tf <- tempfile()
jot_new_pad(tf)
jot_activate(tf)
jot('ex_val', 'ex')
jot_skim()
```

# Index


∗ **helpers**
    get_jot_user, 2
    get_sys_time, 2
∗ **lock**
    jot_lock, 5
∗ **pad**
    jot_new_pad, 6
    jot_open, 7
∗ **read**
    jot_list, 5
    jot_read, 8
    jot_skim, 8
∗ **write**
    jot, 3
    jot_erase, 4

get_jot_user, 2
get_sys_time, 2

jot, 3
jot_activate *(*jot_open*)*, 7
jot_active *(*jot_open*)*, 7
jot_close *(*jot_open*)*, 7
jot_current *(*jot_open*)*, 7
jot_deactivate *(*jot_open*)*, 7
jot_erase, 4
jot_list, 5
jot_lock, 5
jot_new *(*jot_new_pad*)*, 6
jot_new_pad, 6
jot_open, 7
jot_read, 8
jot_skim, 8
jot_unlock *(*jot_lock*)*, 5

read_jot *(*jot_read*)*, 8

write_jot *(*jot*)*, 3