

Package: slurm (via r-universe)

May 20, 2026

Version 2026.3.20

License GPL-3

Title Running and Parsing Slurm Commands

Description User-friendly functions which parse output of command line programs used to query Slurm. Morris A. Jette and Tim Wickberg (2023) <[doi:10.1007/978-3-031-43943-8_1](https://doi.org/10.1007/978-3-031-43943-8_1)> describe Slurm in detail.

Imports data.table (>= 1.11.6), nc (>= 2019.11.22)

SystemRequirements Slurm command line programs (sacct, squeue)

Suggests testthat, ggplot2, R.utils

BugReports <https://github.com/tdhock/slurm/issues>

URL <https://github.com/tdhock/slurm>

Repository <https://tdhock.r-universe.dev>

Date/Publication 2026-03-21 02:51:27 UTC

RemoteUrl <https://github.com/tdhock/slurm>

RemoteRef HEAD

RemoteSha 10d08f9e67049dc8344aff52f1d6d342eedd8a3d

Contents

slurm-package	2
int.pattern	2
na.as.zero	2
sacct	3
sacct-2019-11-21	3
sacct-job13936577	4
sacct-multitype	4
sacct.pattern.list	4
sacct_fields	5
sacct_fread	5
sacct_lines	6

sacct_tasks	7
sjob	8
sjob_dt	8
sq.jobs	9

Index	10
--------------	-----------

slurm-package	<i>Running and Parsing SLURM Commands</i>
---------------	---

Description

User-friendly wrappers for SLURM commands.

Author(s)

Toby Dylan Hocking

int.pattern	<i>int pattern</i>
-------------	--------------------

Description

Match one or more digits and convert to integer.

Usage

"int.pattern"

na.as.zero	<i>na as zero</i>
------------	-------------------

Description

Convert empty string to zero and use as.integer otherwise.

Usage

na.as.zero(int.or.empty)

Arguments

int.or.empty int.or.empty

Value

Integer vector.

Author(s)

Toby Hocking <toby.hocking@r-project.org> [aut, cre]

sacct

sacct

Description

Run `sacct_lines` then `sacct_tasks`.

Usage

`sacct(...)`

Arguments

... ...

Value

Same as result of `sacct_tasks`.

Author(s)

Toby Hocking <toby.hocking@r-project.org> [aut, cre]

sacct-2019-11-21

Output of sacct for thousands of jobs

Description

The data come from running `sacct` on a SLURM system.

Format

This "csv" file results from `sacct -P` which uses the vertical bar | as the separator. There are five columns and six rows (including header).

sacct-job13936577 *Output of sacct for one job with several tasks*

Description

The data come from running sacct on a SLURM system.

Format

This "csv" file results from sacct -P which uses the vertical bar | as the separator. There are five columns and 709 rows (including header).

sacct-multitype *Output of sacct for three job with several tasks*

Description

The data come from running sacct on a SLURM system.

Format

This "csv" file results from sacct -P which uses the vertical bar | as the separator. There are five columns and six rows (including header).

sacct.pattern.list *sacct pattern list*

Description

Named list of patterns for parsing [sacct](#) fields.

Usage

"sacct.pattern.list"

sacct_fields	<i>sacct_fields</i>
--------------	---------------------

Description

Get current fields from [sacct](#).

Usage

```
sacct_fields(cmd = "sacct -e")
```

Arguments

cmd	command to run and capture output, default "sacct -e" to query slurm, can be changed to cat a file for testing.
-----	---

Value

character vector, items that can be used as format.fields argument to [sacct_lines](#)

Author(s)

Toby Hocking <toby.hocking@r-project.org> [aut, cre]

sacct_fread	<i>sacct_fread</i>
-------------	--------------------

Description

Run fread on the output of [sacct](#).

Usage

```
sacct_fread(...)
```

Arguments

... ...

Value

Data table with the same number of rows as the output of the [sacct](#) command, and additional columns that result from parsing the columns with [sacct.pattern.list](#). This table typically has more than one row per task, so typically this table is used as input to [sacct_tasks](#).

Author(s)

Toby Hocking <toby.hocking@r-project.org> [aut, cre]

Examples

```
library(slurm)
sacct_fread(text="JobID|ExitCode|State|MaxRSS|Elapsed
18473217_1|0:0|RUNNING||00:03:47
18473217_1.extern|0:0|RUNNING||00:03:47")
```

sacct_lines

sacct lines

Description

Run `sacct` with args and parse output as a data.table

Usage

```
sacct_lines(args, format.fields = c("JobID",
  "ExitCode", "State",
  "MaxRSS", "Elapsed"),
  delimiter = "\t",
  out_file = getOption("slurm.sacct_lines.out_file"))
```

Arguments

args	character string passed to <code>sacct</code> command line, e.g. -j123 for selecting job ID 123.
format.fields	character vector of field names to pass to <code>sacct -format</code> . Use <code>sacct_fields</code> to get all fields.
delimiter	passed as <code>-delimiter</code> .
out_file	out_file

Value

Same as `sacct_fread`.

Author(s)

Toby Hocking <toby.hocking@r-project.org> [aut, cre]

sacct_tasks	<i>sacct tasks</i>
-------------	--------------------

Description

Use output of [sacct_fread](#) to compute a table with one row per task.

Usage

```
sacct_tasks(task.dt)
```

Arguments

```
task.dt      task.dt
```

Value

data.table with one row per job/task, typically used as input to [sjob_dt](#).

Author(s)

Toby Hocking <toby.hocking@r-project.org> [aut, cre]

Examples

```
library(slurm)
sacct.csv.gz <- system.file(
  "data", "sacct-job13936577.csv.gz", package="slurm", mustWork=TRUE)
if(requireNamespace("R.utils")){
  sacct.dt <- sacct_fread(sacct.csv.gz)
  task.dt <- sacct_tasks(sacct.dt)
  print(task.dt[State_batch != "COMPLETED"])
  if(require(ggplot2)){
    ggplot()+
      geom_point(aes(
        hours, megabytes, fill=State_batch),
        shape=21,
        data=task.dt)+
      scale_fill_manual(values=c(
        COMPLETED=NA,
        FAILED="red"))+
      scale_x_log10()+
      scale_y_log10()
  }
}
```

sjob	<i>sjob</i>
------	-------------

Description

Run [sacct](#) and summarize State/ExitCode values for given job IDS

Usage

```
sjob(job.id = sq.jobs(),
      tasks.width = 11,
      ...)
```

Arguments

job.id	job.id
tasks.width	tasks.width
...	...

Value

Data table from [sjob_dt](#).

Author(s)

Toby Hocking <toby.hocking@r-project.org> [aut, cre]

sjob_dt	<i>sjob dt</i>
---------	----------------

Description

Summarize task table from [sacct_tasks](#).

Usage

```
sjob_dt(time.dt, tasks.width = 11)
```

Arguments

time.dt	time.dt
tasks.width	tasks.width

Value

data.table with one row per unique value of job/State/ExitCode (and keyed on these columns).

Author(s)

Toby Hocking <toby.hocking@r-project.org> [aut, cre]

Examples

```
library(slurm)
sacct.csv.gz <- system.file(
  "data", "sacct-job13936577.csv.gz", package="slurm", mustWork=TRUE)
if(requireNamespace("R.utils")){
  task.dt <- sacct_fread(sacct.csv.gz)
  print(summary.dt <- sjob_dt(task.dt))
}
```

sq.jobs

sq.jobs

Description

get currently running jobs

Usage

```
sq.jobs(args = "-u $USER")
```

Arguments

args args

Value

character scalar: job1,job2,etc

Author(s)

Toby Hocking <toby.hocking@r-project.org> [aut, cre]

Index

* datasets

- sacct-2019-11-21, 3
- sacct-job13936577, 4
- sacct-multitype, 4

* package

- slurm-package, 2

int.pattern, 2

na.as.zero, 2

sacct, 3, 4–6, 8

sacct-2019-11-21, 3

sacct-job13936577, 4

sacct-multitype, 4

sacct.pattern.list, 4, 5

sacct_fields, 5, 6

sacct_fread, 5, 6, 7

sacct_lines, 3, 5, 6

sacct_tasks, 3, 5, 7, 8

sjob, 8

sjob_dt, 7, 8, 8

slurm (slurm-package), 2

slurm-package, 2

sq.jobs, 9