Package 'slurm'

April 9, 2025

Version 2025.4.9

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> 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

NeedsCompilation no

Author Toby Hocking [aut, cre]

Maintainer Toby Hocking <toby.hocking@r-project.org>

Repository CRAN

Date/Publication 2025-04-09 10:30:02 UTC

Contents

Index   **10**

---

  slurm-package   *Running and Parsing SLURM Commands*

Description

User-friendly wrappers for SLURM commands.

Author(s)

Toby Dylan Hocking

---

  int.pattern   *int pattern*

Description

Match one or more digits and convert to integer.

Usage

```
"int.pattern"
```

---

  na.as.zero   *na as zero*

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   *sacct fields*

Description

Get current fields from [sacct](#).

Usage

```
sacct_fields()
```

Value

character vector.

Author(s)

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

`sacct_fread` *sacct fread*

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.

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")
```

Arguments

`args`	character string passed to [sacct](#) command line, e.g. -j123 for selecting job ID 123.
`format.fields`	character vector of field names to pass to [sacct](#) –format. Use [sacct_fields](#) to get all fields.
`delimiter`	passed as –delimiter.

Value

Same as [sacct_fread](#).

Author(s)

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

---

sacct_tasks	*sacct tasks*

Description

Summarize output from [sacct_fread](#).

Usage

```
sacct_tasks(match.dt)
```

Arguments

`match.dt`	match.dt

Value

data.table with one row per job/task.

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	*sjob*

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   *sjob dt*

Description

Get summary dt from [sacct](#) dt.

Usage

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

Arguments

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

Value

data.table summarizing State/ExitCode distribution over jobs

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]

---

task.pattern   *task pattern*

Description

Pattern for either one task or a range.

Usage

```
"task.pattern"
```

Index