

Package ‘rrd’

March 7, 2022

Type Package

Title Import Data from a RRD (Round Robin Database) File

Version 0.2.4

Description Makes it easy to import the data from a 'RRD' database (<<https://oss.oetiker.ch/rrdtool/>>) directly into R data structures. The resulting objects are 'tibble' objects or a list of 'tibble' objects, making it easy to manipulate the data. The package uses 'librrd' to import the numerical data in a 'RRD' database directly into R data structures without using intermediate formats.

URL <https://github.com/andrie/rrd/>, <https://andrie.github.io/rrd/>

BugReports <https://github.com/andrie/rrd/issues>

SystemRequirements librrd: 'librrd-dev' (DEB), 'rrdtool-devel' (RPM), 'rrdtool' (OSX), 'rrdtool' (CSW)

NeedsCompilation yes

OS_type unix

Imports assertthat, tibble

License MIT + file LICENSE

Suggests testthat, covr, ggplot2, spelling

RoxygenNote 7.1.2

Encoding UTF-8

Language en-US

Config/testthat/edition 3

Author Andrie de Vries [cre, cph],
Plamen Dimitrov [aut, cph],
Tobias Oetiker [cph] (rrdtool and rrd library,
<https://oss.oetiker.ch/rrdtool/>)

Maintainer Andrie de Vries <apdevries@gmail.com>

Repository CRAN

Date/Publication 2022-03-07 17:30:05 UTC

R topics documented:

rrd-package	2
describe_rrd	3
importRRD	4
read_rra	4
read_rrd	6

Index	7
--------------	----------

rrd-package	<i>rrd: Import Data from a RRD (Round Robin Database) File</i>
-------------	--

Description

Makes it easy to import the data from a 'RRD' database (<https://oss.oetiker.ch/rrdtool/>) directly into R data structures. The resulting objects are 'tibble' objects or a list of 'tibble' objects, making it easy to manipulate the data. The package uses 'librrd' to import the numerical data in a 'RRD' database directly into R data structures without using intermediate formats.

Details

Exposes the following functions:

- `describe_rrd()` to enumerate the archives included in a RRD file.
- `read_rrd()` to read an entire RRD file, including all the archives
- `read_rra()` to extract a single RRA (round robin archive) from an RRD file

For more information on RRDtool and the RRD format please refer to the official RRDtool [documentation](#) and [tutorials](#).

You can also read a more in-depth description of the package in an [R Views](#) blog post [Reading and analysing log files in the RRD database format](#).

Package history

Plamen Dimitrov wrote the original proof of concept of the package during a [Google Summer of Code 2014](#) project and wrote an accompanying blog post "[R Package for Working With RRD Files](#)".

Andrie de Vries became maintainer of the package early in 2018, and prepared the package for release to CRAN by adding documentation, examples and unit tests. At this time the API changed so resulting objects are tibble objects, making it easier to analyse data using tidyverse concepts. At this time he also published the "R Views" [blog post](#).

Author(s)

Maintainer: Andrie de Vries <apdevries@gmail.com> [copyright holder]

Authors:

- Plamen Dimitrov <dim.plamen@gmail.com> [copyright holder]

Other contributors:

- Tobias Oetiker (rrdtool and rrd library, <https://oss.oetiker.ch/rrdtool/>) [copyright holder]

References

<https://oss.oetiker.ch/rrdtool/doc/index.en.html>

See Also

Useful links:

- <https://github.com/andrie/rrd/>
- <https://andrie.github.io/rrd/>
- Report bugs at <https://github.com/andrie/rrd/issues>

describe_rrd

Describes content of a RRD file.

Description

Describes content of a RRD file.

Usage

```
describe_rrd(filename)
```

Arguments

filename File name

See Also

Other rrd functions: [read_rra\(\)](#), [read_rrd\(\)](#)

Examples

```
rrd_cpu_0 <- system.file("extdata/cpu-0.rrd", package = "rrd")
```

```
describe_rrd(rrd_cpu_0)
```

importRRD	<i>Deprecated functions.</i>
-----------	------------------------------

Description

Deprecated functions.

Deprecated functions.

Usage

```
importRRD(filename, cf = NULL, start = NULL, end = NULL, step = NULL)
```

```
importRRD(filename, cf = NULL, start = NULL, end = NULL, step = NULL)
```

Arguments

filename	File name
cf	The consolidation function that is applied to the data you want to fetch. Must be one of c("AVERAGE", "MIN", "MAX", "LAST")
start	start time
end	end time, defaults to the current system time
step	step

read_rra	<i>Imports the RRA data from an RRD database</i>
----------	--

Description

Finds the RRA (round robin array) that best matches the consolidation function and the step and imports all values (from all data stores) in that RRA that are between timestamp start and end. Note that start is not included in the result.

Returns a data.frame object having the timestamp and the data stores as columns. The data store names are retrieved from the RRD file and set as the corresponding column names. The timestamps are also used as row names.

Usage

```
read_rra(filename, cf, step, n_steps, start, end = Sys.time())
```

read_rrd	<i>Imports data from an RRD database</i>
----------	--

Description

Reads the metadata in the RRD and adjusts the parameters accordingly in order to expose all RRAs in their entirety.

Usage

```
read_rrd(filename)
```

Arguments

filename	File name
----------	-----------

Value

Returns a named list of data.frames. Each data frame corresponds to an RRA (see [read_rra\(\)](#)). The list has names constructed as "consolidation function" + "step" - e.g. "AVERAGE15".

References

<https://oss.oetiker.ch/rrdtool/doc/rrdfetch.en.html>

See Also

Other rrd functions: [describe_rrd\(\)](#), [read_rra\(\)](#)

Examples

```
rrd_cpu_0 <- system.file("extdata/cpu-0.rrd", package = "rrd")

describe_rrd(rrd_cpu_0)

cpu <- read_rrd(rrd_cpu_0)
names(cpu)
head(cpu[[1]])
tail(cpu[[1]])

tail(
  cpu$AVERAGE60$sys
)
```

Index

- * **librrd**
 - rrd-package, 2
- * **package**
 - rrd-package, 2
- * **rrd functions**
 - describe_rrd, 3
 - read_rra, 4
 - read_rrd, 6
- * **rrdtool**
 - rrd-package, 2
- * **rrd**
 - rrd-package, 2
 - _PACKAGE (rrd-package), 2

describe_rrd, 3, 5, 6

describe_rrd(), 2

importRRD, 4

read_rra, 3, 4, 6

read_rra(), 2, 6

read_rrd, 3, 5, 6

read_rrd(), 2

rrd (rrd-package), 2

rrd-package, 2