

# Package ‘rrd’

July 5, 2018

**Type** Package

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

**Version** 0.2.1

**Date** 2018-07-02

**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.

**SystemRequirements** librrd-dev, rrdtool

**OS\_type** unix

**RoxygenNote** 6.0.1

**NeedsCompilation** yes

**Imports** assertthat, tibble

**License** MIT + file LICENSE

**Suggests** testthat, covr

**Author** Andrie de Vries [cre, cph],  
Plamen Dimitrov [aut, cph]

**Maintainer** Andrie de Vries <[apdevries@gmail.com](mailto:apdevries@gmail.com)>

**Repository** CRAN

**Date/Publication** 2018-07-05 08:30:06 UTC

## R topics documented:

rrd-package	2
describe_rrd	2
importRRD	3
read_rra	4
read_rrd	5

<b>Index</b>	<b>7</b>
--------------	----------

---

rrd-package

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

---

## Description

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](#).

## References

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

---

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 <code>c("AVERAGE", "MIN", "MAX", "LAST")</code>
start	start time
end	end time, defaults to the current system time
step	step

---

`read_rra`*Imports the RRA data from an RRD database*

---

### 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())
```

### Arguments

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

### Details

The `filename`, `cf` (consolidation function) and `step` arguments uniquely identify an RRA array in the RRD file.

The arguments `start` and `end` define the time-slice to be retrieved. Note that `start` is not included in the result. Refer to the documentation for [rrdfetch](#) for more information.

The returned `data.frame` has the timestamp and the data stores as separate columns. The names of the data sources are extracted from the RRD file and set as column names. The timestamps are also used as row names.

### See Also

Other rrd functions: [describe\\_rrd](#), [read\\_rrd](#)

## Examples

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

# Note that the default end time is the current time (Sys.time())
# However, since the sample data is historic, specify the end time

start_time <- as.POSIXct("2018-05-01") # timestamp with data in example
end_time <- as.POSIXct("2018-05-02")  # timestamp with data in example

# read archive by specifying start time
avg_60 <- read_rra(rrd_cpu_0, cf = "AVERAGE", step = 60L,
                  start = start_time,
                  end = end_time)

names(avg_60)
head(avg_60)
tail(avg_60)

# read archive by specifying number of rows to retrieve
avg_60 <- read_rra(rrd_cpu_0, cf = "AVERAGE", step = 60L,
                  n_steps = 5,
                  end = end_time)

names(avg_60)
avg_60
```

---

read\_rrd

*Imports data from an RRD database*

---

## 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

\*Topic **librrd**

rrd-package, 2

\*Topic **package**

rrd-package, 2

\*Topic **rrdtool**

rrd-package, 2

\*Topic **rrd**

rrd-package, 2

describe\_rrd, 2, 4, 6

describe\_rrd(), 2

importRRD, 3

read\_rra, 3, 4, 6

read\_rra(), 2, 5

read\_rrd, 3, 4, 5

read\_rrd(), 2

rrd (rrd-package), 2

rrd-package, 2