

**NAME**

rrdmodify – Change the data source structure of an RRD

**SYNOPSIS**

```
rrdtool {modify} [-d <address >] [-s <newstep >] in-filename out-filename [ DEL:ds-name | DS:ds-spec ... ]
```

**DESCRIPTION**

The **modify** function copies an RRD file *in-filename* to *out-filename*, possibly removing or adding data sources and round-robin archives (RRAs) on the fly. When *in-filename* and *out-filename* denote the same file, this is (almost) identical to an in-place modification.

–d <address >

If given, **RRDTool** will try to connect to the caching daemon rrdcached at *address* to flush any data not yet written to the input RRD before reading it in. After writing the final file a FORGET command is sent to the daemon. This will make the daemon recognize the updated structure of the underlying RRD file.

–s <newstep >

Changes the step size of the RRD to *newstep*. There are restrictions on the possible *newstep* that can be given, namely: The current stepping of the *in-filename* must be a multiple of the *newstep*. This also means that currently the step size can only be reduced.

When giving this option, modifications to RRAs will be done AFTER adapting the step size, so any RRA specifications referring to counts of PDPs will already use the new stepping.

Note that changing the step size does NOT change the number of rows of any existing RRA by itself: Only the stepsize and assorted other meta values get changed (namely the PDP count of RRAs and possibly some CDP information).

For a reduced step size to make sense in practice, it is usually a good idea to add one or more new RRAs with a step size of 1, because all existing RRAs will afterwards definitely have a pdp step count of larger than one, effectively hiding the step smaller size.

*in-filename*

The name of the source **RRD** file you want to copy/modify.

*out-filename*

The name of the destination **RRD** file. If it names the same file as *in-filename*, the new RRD will effectively replace the old one.

**DEL:ds-name**

Every data source named with a DEL specification will be removed during the copy operation. The resulting RRD will miss both the definition and the data for that data source. Multiple DEL specifications are permitted.

**DS:ds-spec**

For every such data source definition (for the exact syntax see the create command), a new data source will be added to the output RRD. Multiple DS specifications are permitted.

**DELRA:index**

Removes the RRA with index *index*.

**RRA:rra-spec**

For every such archive definition (for the exact syntax see the create command), a new RRA will be added to the output RRD. Multiple RRA specifications are permitted.

**RRA#index:[+|=|<number>**

Adds/removes or sets the given number of rows for the RRA with index <index>.

–**daemon** *address*

If given, **RRDTool** will try to connect to the caching daemon rrdcached at *address* and will fail if the connection cannot be established. If the connection is successfully established the data for the

*in-filename* will be flushed before performing the copy/modify operation. Afterwards the *out-filename* will be forgotten by the cache daemon, so that the next access using the caching daemon will read the proper structure.

For a list of accepted formats, see the `-I` option in the `rrdcached` manual.

## ENVIRONMENT VARIABLES

The following environment variables may be used to change the behavior of `rrdtool update`:

### RRDCACHED\_ADDRESS

If this environment variable is set it will have the same effect as specifying the `--daemon` option on the command line. If both are present, the command line argument takes precedence.

## AUTHORS

Tobias Oetiker <[tobi@oetiker.ch](mailto:tobi@oetiker.ch)>, Peter Stamfest <[peter@stamfest.at](mailto:peter@stamfest.at)>