Amount of Performance Data Storage NMIS8 Stores

NMIS8 is a Network Management System which performs performance management. NMIS8 collects SNMP data from routers, switches, firewalls, servers, and many more types of IT devices. It stores the collected data in a performance database, which is an RRD file using RRDTool.

This article will describe how much data NMIS8 stores and how to modify NMIS8 to store more data.

How much data does NMIS8 Store?

RRDTool is a round robin database, which stores data in a binary file which is effectively circular, you define how many elements of data you want to keep and keep feeding data and it will keep adding elements up to that many of entries, when it is full, the oldest entries are overwritten. To keep summarised data you define a summarisation scheme, so the data is rolled up into the next level. In NMIS8 we define how much data RRDTool keeps by using the NMIS8 modelling system. The files for the NMIS8 models are stored in /path/to/nmis8/models and specifically the default database definitions are kept in /path/to/nmis8/models/Common-database.nmis

By default NMIS8 polls for performance data every 5 minutes and data in the following scheme:

Data Summary	Days Kept For
raw data (5 minute polls)	8 days
30 minute averaged data	32 days
120 minute averaged data	189 days
daily averaged data	1890 days (5 years)

This summarisation scheme was chosen to get a good balance of data stored and disk space used. The file sizes created using this scheme are as follows:

RRD Type	File Size
Interface (ifInOctets and ifOutOcts)	581112 bytes
Packet data	1933120 bytes
HC Packet data (new)	2319408 bytes
MIB-2 IP data	1739976 bytes
NMIS Health data	3091984 bytes

What about statistical summarisation?

NMIS8 does not just keep the AVERAGES, it also keeps the MAXIMUM and MINIMUM for each data point, so in effect you have the range of data seen for a given summarisation, this is very important statistically, and many systems do not maintain this information. When doing analysis of performance data, it is important to use the averages (MEAN) as well as the MAXIMUM and MINIMUM (RANGE).

Can NMIS8 Store More Performance Data?

Yes, NMIS8 can be configured to store as much performance data as you have disk for. This is very easy to do when you first install an NMIS system, but can be done later as well. (If you are using the NMIS8 Virtual Machine please check out our instructions on Resizing NMIS VMs.) To store more data, first you need to determine how much more data you would like to store, then plug those numbers into the RRD Calculator Spreadsheet rrd_calc2.xls, attached for your convenience, this will tell you what you need to change in the NMIS8 model file /path/to/nmis8/models/Common-database.nmis

The default Common-database.nmis file contains this:

```
'db' => {
  'hbeat' => '900',
  'poll' => '300',
  'size' => {
      'reachability' => {
      'step_year' => '288',
      'rows_month' => '2268',
      'rows_year' => '1890',
      'step_day' => '1',
      'step_month' => '24',
      'step_week' => '6',
      'rows_day' => '2304',
      'rows_week' => '1536'
    },
    'interface' => {
      'step_year' => '288',
      'rows_month' => '2268',
      'rows_year' => '1890',
      'step_day' => '1',
      'step_month' => '24',
      'step_week' => '6',
      'rows_day' => '2304',
      'rows_week' => '1536'
    },
    'default' => {
      'step_year' => '288',
      'rows_month' => '2268',
      'rows_year' => '1890',
      'step_day' => '1',
      'step_month' => '24',
      'step_week' => '6',
      'rows_day' => '2304',
      'rows_week' => '1536'
    },
    'metrics' => {
      'step_year' => '288',
      'rows_month' => '2268',
      'rows_year' => '1890',
      'step_day' => '1',
      'step_month' => '24',
      'step_week' => '6',
      'rows_day' => '2304',
      'rows_week' => '1536'
    }
  }
},
```

This will need to be changed to suit your new scheme, you might like to only keep more interface data, or to change all of them to keep more data.

How would I configure NMIS8 to keep 32 days of raw data?

To configure NMIS8 to keep 32 days of raw data (5 minute) and 96 days of 30 minutes of interface data you would make the following change to the interface section.

Keep (days)	Summarise (minutes)	Model Entry	Default Value	New Value
32 days	5 minutes	rows_day	2304	9216
96 days	30 minutes	rows_week	1536	4608

The entry for the Common-database.nmis model would look like this after the change:

```
'interface' => {
  'step_year' => '288',
  'rows_month' => '2268',
  'rows_year' => '1890',
  'step_day' => '1',
  'step_month' => '24',
  'step_week' => '6',
  'rows_day' => '9216',
  'rows_week' => '4608'
},
```

This change could be made to all the entries in the Common-database.nmis file if required.

NMIS8 will now create new RRD files with the new scheme defined in the model file. The resulting file sizes from the change above are:

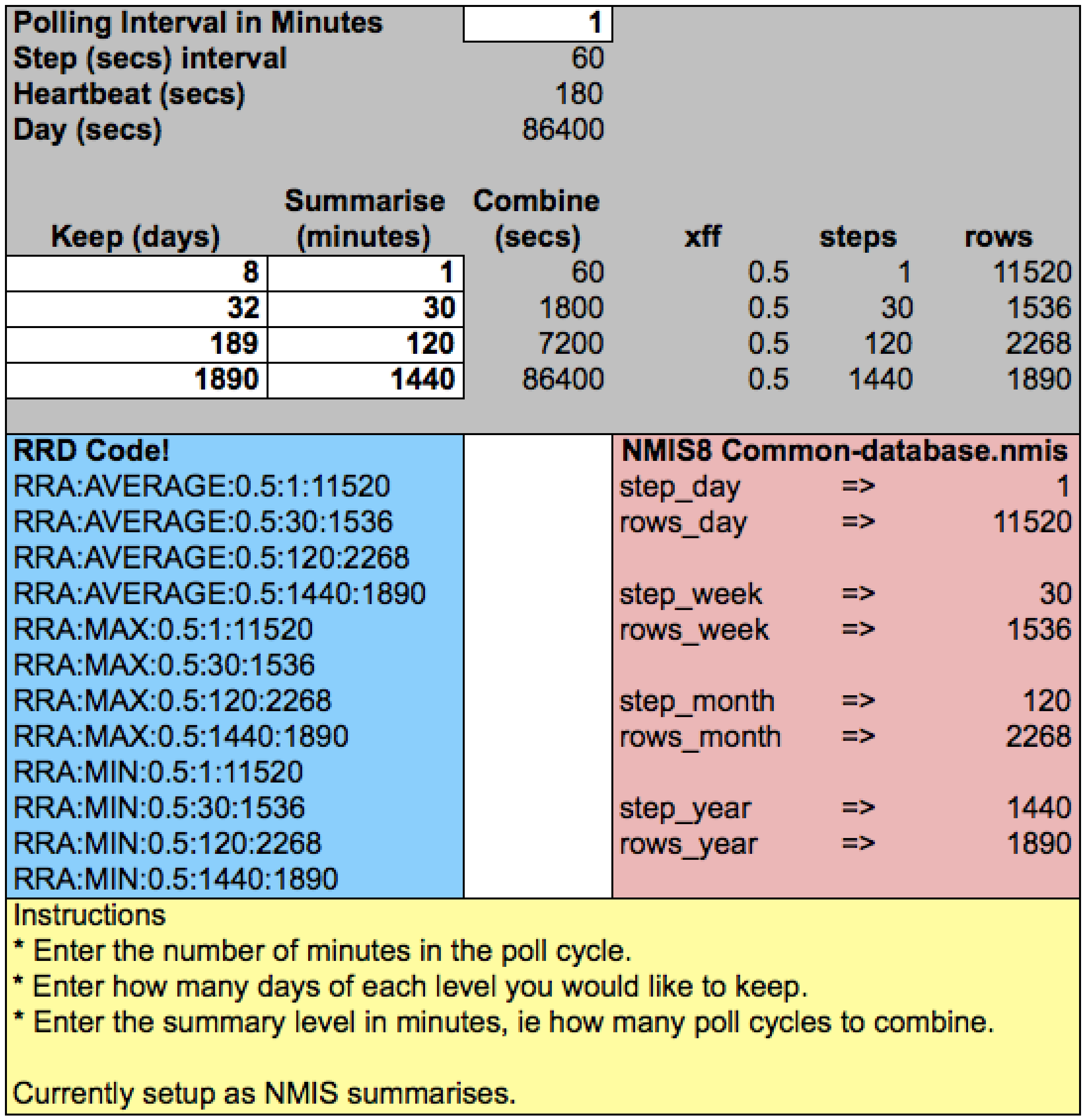
RRD Type	Old File Size	New File Size	% Increase
Interface (ifInOctets and ifOutOcts)	581112 bytes	1391364 bytes	239%
Packet data	1933120 bytes	4264508 bytes	220%

How would I configure NMIS8 to poll every 1 minute and keep 8 days of raw data?

To configure NMIS8 to keep 8 days of raw data (with 1 minute polls) and 32 days of 30 minutes of interface data you would make the following change to all the sections.

Keep (days)	Summarise (minutes)	Model Entry	Default Value	New Value
8 days	1 minutes	rows_day	2304	11520
32 days	30 minutes	rows_week	1536	1536 (no change)

I can get these results using the RRD Calculator (which has been updated after ~12 years to include the NMIS8 steps/rows) this is the screen shot, I have changed the default of 5 minutes to 1 minute.



Polling Interval in Minutes	**1**
Step (secs) interval	60
Heartbeat (secs)	180
Day (secs)	86400

Keep (days)	**Summarise (minutes)**	**Combine (secs)**	**xff**	**steps**	**rows**
8	**1**	60	0.5	1	11520
32	**30**	1800	0.5	30	1536
189	**120**	7200	0.5	120	2268
1890	**1440**	86400	0.5	1440	1890

RRD Code!

```
RRA:AVERAGE:0.5:1:11520
RRA:AVERAGE:0.5:30:1536
RRA:AVERAGE:0.5:120:2268
RRA:AVERAGE:0.5:1440:1890
RRA:MAX:0.5:1:11520
RRA:MAX:0.5:30:1536
RRA:MAX:0.5:120:2268
RRA:MAX:0.5:1440:1890
RRA:MIN:0.5:1:11520
RRA:MIN:0.5:30:1536
RRA:MIN:0.5:120:2268
RRA:MIN:0.5:1440:1890
```

NMIS8 Common-database.nmis

```
step_day     =>      1
rows_day     =>  11520

step_week    =>     30
rows_week    =>   1536

step_month   =>    120
rows_month   =>   2268

step_year    =>   1440
rows_year    =>   1890
```

Instructions
* Enter the number of minutes in the poll cycle.
* Enter how many days of each level you would like to keep.
* Enter the summary level in minutes, ie how many poll cycles to combine.

Currently setup as NMIS summarises.

You will need to change the step and polling interface in the top of the file the defaults are hbeat (heartbeat) 900 and poll 300:

```
    'db' => {
      'hbeat' => '900',
      'poll' => '300',
      'size' => {
```

The new values from the spreadsheet are 180 and 60, so the change is:

```
    'db' => {
      'hbeat' => '180',
      'poll' => '60',
      'size' => {
```

The interface entry for the Common-database.nmis model would look like this after the change, the terms steps and rows are from RRDTool, using the year, month, etc is really a nominal name, these are the types and levels of summarisation, typically we summarise by day, week, month, year, but you can do others if needed:

```
 'interface' => {
   'step_year' => '288',
   'rows_month' => '2268',
   'rows_year' => '1890',
   'step_day' => '1',
   'step_month' => '24',
   'step_week' => '6',
   'rows_day' => '9216',
   'rows_week' => '4608'
 },
```

This change should be made to the following sections:

- reachability
- interface
- default
- metrics

After the change to 1 min data for 8 days, the RRD files are larger, an important consideration for overall disk usage.

RRD Type	Old File Size	New File Size	% Increase
Interface (ifInOctets and ifOutOcts)	581112 bytes	1244664 bytes	214%
Packet data	1933120 bytes	4973616 bytes	257%

Can I migrate existing NMIS8 RRD files to the new scheme

Yes and no, increasing the amount of data you store is not problem, changing the polling interval to 1 minute creates a logical problem. To change to 1 minute polling might be able to save the existing data but it will be 5 minute polling data, so the graphs would be wrong, you can delete all the existing files and start anew, and NMIS will just create them with the new values defined in the Common-database.nmis file.

To increase the data you can take existing NMIS8 RRD files and resize them using RRDTool Resize, the documentation for RRDTool resize is available on the RRDTool website http://oss.oetiker.ch/rrdtool/doc/rrdresize.en.html, in summary it would work like this.

If I wanted to change the amount of data being kept in an existing file, I would first determine the RRA Number for the entry to update, so I would run RRDTool Info on the file the output looks like this:

```
[root@nmisdev64 wanedge1]# /usr/local/rrdtool/bin/rrdtool info wanedge1-fastethernet0-1.rrd
filename = "wanedge1-fastethernet0-1.rrd"
rrd_version = "0003"
step = 300
last_update = 1355967616
header_size = 5256
ds[ifInOctets].index = 0
ds[ifInOctets].type = "COUNTER"
ds[ifInOctets].minimal_heartbeat = 900
ds[ifInOctets].min = 0.0000000000e+00
ds[ifInOctets].max = 1.0240000000e+06
ds[ifInOctets].last_ds = "9695172"
ds[ifInOctets].value = 1.2818317168e+02
ds[ifInOctets].unknown_sec = 0
ds[ifOperStatus].index = 1
ds[ifOperStatus].type = "GAUGE"
ds[ifOperStatus].minimal_heartbeat = 900
ds[ifOperStatus].min = 0.0000000000e+00
ds[ifOperStatus].max = 1.0000000000e+02
ds[ifOperStatus].last_ds = "100"
ds[ifOperStatus].value = 1.6507604000e+03
ds[ifOperStatus].unknown_sec = 0
ds[ifOutOctets].index = 2
ds[ifOutOctets].type = "COUNTER"
ds[ifOutOctets].minimal_heartbeat = 900
ds[ifOutOctets].min = 0.0000000000e+00
ds[ifOutOctets].max = 1.0240000000e+06
ds[ifOutOctets].last_ds = "17185853"
ds[ifOutOctets].value = 2.2330066758e+02
```

```
ds[ifOutOctets].unknown_sec = 0
rra[0].cf = "AVERAGE"
rra[0].rows = 2304
rra[0].cur_row = 537
rra[0].pdp_per_row = 1
rra[0].xff = 5.0000000000e-01
rra[0].cdp_prep[0].value = NaN
rra[0].cdp_prep[0].unknown_datapoints = 0
rra[0].cdp_prep[1].value = NaN
rra[0].cdp_prep[1].unknown_datapoints = 0
rra[0].cdp_prep[2].value = NaN
rra[0].cdp_prep[2].unknown_datapoints = 0
rra[1].cf = "AVERAGE"
rra[1].rows = 1536
rra[1].cur_row = 1325
rra[1].pdp_per_row = 6
rra[1].xff = 5.0000000000e-01
rra[1].cdp_prep[0].value = 1.5618120721e+01
rra[1].cdp_prep[0].unknown_datapoints = 0
rra[1].cdp_prep[1].value = 2.0000000000e+02
rra[1].cdp_prep[1].unknown_datapoints = 0
rra[1].cdp_prep[2].value = 2.7509214203e+01
rra[1].cdp_prep[2].unknown_datapoints = 0
rra[2].cf = "AVERAGE"
rra[2].rows = 2268
rra[2].cur_row = 1977
rra[2].pdp_per_row = 24
rra[2].xff = 5.0000000000e-01
rra[2].cdp_prep[0].value = 1.5614431658e+02
rra[2].cdp_prep[0].unknown_datapoints = 0
rra[2].cdp_prep[1].value = 2.0000000000e+03
rra[2].cdp_prep[1].unknown_datapoints = 0
rra[2].cdp_prep[2].value = 2.7504466927e+02
rra[2].cdp_prep[2].unknown_datapoints = 0
rra[3].cf = "AVERAGE"
rra[3].rows = 1890
rra[3].cur_row = 1585
rra[3].pdp_per_row = 288
rra[3].xff = 5.0000000000e-01
rra[3].cdp_prep[0].value = 1.5614431658e+02
rra[3].cdp_prep[0].unknown_datapoints = 0
rra[3].cdp_prep[1].value = 2.0000000000e+03
rra[3].cdp_prep[1].unknown_datapoints = 0
rra[3].cdp_prep[2].value = 2.7504466927e+02
rra[3].cdp_prep[2].unknown_datapoints = 0
rra[4].cf = "MAX"
rra[4].rows = 2304
rra[4].cur_row = 611
rra[4].pdp_per_row = 1
rra[4].xff = 5.0000000000e-01
rra[4].cdp_prep[0].value = NaN
rra[4].cdp_prep[0].unknown_datapoints = 0
rra[4].cdp_prep[1].value = NaN
rra[4].cdp_prep[1].unknown_datapoints = 0
rra[4].cdp_prep[2].value = NaN
rra[4].cdp_prep[2].unknown_datapoints = 0
rra[5].cf = "MAX"
rra[5].rows = 1536
rra[5].cur_row = 1239
rra[5].pdp_per_row = 6
rra[5].xff = 5.0000000000e-01
rra[5].cdp_prep[0].value = 7.8502714737e+00
rra[5].cdp_prep[0].unknown_datapoints = 0
rra[5].cdp_prep[1].value = 1.0000000000e+02
rra[5].cdp_prep[1].unknown_datapoints = 0
rra[5].cdp_prep[2].value = 1.3968230743e+01
rra[5].cdp_prep[2].unknown_datapoints = 0
rra[6].cf = "MAX"
rra[6].rows = 2268
rra[6].cur_row = 251
rra[6].pdp_per_row = 24
```

```
rra[6].xff = 5.0000000000e-01
rra[6].cdp_prep[0].value = 7.9934094489e+00
rra[6].cdp_prep[0].unknown_datapoints = 0
rra[6].cdp_prep[1].value = 1.0000000000e+02
rra[6].cdp_prep[1].unknown_datapoints = 0
rra[6].cdp_prep[2].value = 1.4389061029e+01
rra[6].cdp_prep[2].unknown_datapoints = 0
rra[7].cf = "MAX"
rra[7].rows = 1890
rra[7].cur_row = 1714
rra[7].pdp_per_row = 288
rra[7].xff = 5.0000000000e-01
rra[7].cdp_prep[0].value = 7.9934094489e+00
rra[7].cdp_prep[0].unknown_datapoints = 0
rra[7].cdp_prep[1].value = 1.0000000000e+02
rra[7].cdp_prep[1].unknown_datapoints = 0
rra[7].cdp_prep[2].value = 1.4389061029e+01
rra[7].cdp_prep[2].unknown_datapoints = 0
rra[8].cf = "MIN"
rra[8].rows = 2304
rra[8].cur_row = 955
rra[8].pdp_per_row = 1
rra[8].xff = 5.0000000000e-01
rra[8].cdp_prep[0].value = NaN
rra[8].cdp_prep[0].unknown_datapoints = 0
rra[8].cdp_prep[1].value = NaN
rra[8].cdp_prep[1].unknown_datapoints = 0
rra[8].cdp_prep[2].value = NaN
rra[8].cdp_prep[2].unknown_datapoints = 0
rra[9].cf = "MIN"
rra[9].rows = 1536
rra[9].cur_row = 927
rra[9].pdp_per_row = 6
rra[9].xff = 5.0000000000e-01
rra[9].cdp_prep[0].value = 7.7678492474e+00
rra[9].cdp_prep[0].unknown_datapoints = 0
rra[9].cdp_prep[1].value = 1.0000000000e+02
rra[9].cdp_prep[1].unknown_datapoints = 0
rra[9].cdp_prep[2].value = 1.3540983460e+01
rra[9].cdp_prep[2].unknown_datapoints = 0
rra[10].cf = "MIN"
rra[10].rows = 2268
rra[10].cur_row = 720
rra[10].pdp_per_row = 24
rra[10].xff = 5.0000000000e-01
rra[10].cdp_prep[0].value = 7.6268952036e+00
rra[10].cdp_prep[0].unknown_datapoints = 0
rra[10].cdp_prep[1].value = 1.0000000000e+02
rra[10].cdp_prep[1].unknown_datapoints = 0
rra[10].cdp_prep[2].value = 1.2730238700e+01
rra[10].cdp_prep[2].unknown_datapoints = 0
rra[11].cf = "MIN"
rra[11].rows = 1890
rra[11].cur_row = 1641
rra[11].pdp_per_row = 288
rra[11].xff = 5.0000000000e-01
rra[11].cdp_prep[0].value = 7.6268952036e+00
rra[11].cdp_prep[0].unknown_datapoints = 0
rra[11].cdp_prep[1].value = 1.0000000000e+02
rra[11].cdp_prep[1].unknown_datapoints = 0
rra[11].cdp_prep[2].value = 1.2730238700e+01
rra[11].cdp_prep[2].unknown_datapoints = 0
```

From all of this data I am interested in the following six lines:

```
rra[0].rows = 2304
rra[1].rows = 1536
rra[4].rows = 2304
rra[5].rows = 1536
rra[8].rows = 2304
rra[9].rows = 1536
```

These entries define the storage of the day and week summaries for RRA entries AVERAGE, MIN and MAX.

The RRDTool resize command can only resize one RRA at a time, and works on adding rows, so we need to **add 6912 rows** to the daily value (9216 - 2304) and **add 3072 rows** to the weekly value (4608 - 1536). The following are the commands required to resize this RRD.

```
cp wanedge1-fastethernet0-1.rrd wanedge1-fastethernet0-1.rrd.bak
/usr/local/rrdtool/bin/rrdtool resize wanedge1-fastethernet0-1.rrd 0 GROW 6912
mv resize.rrd wanedge1-fastethernet0-1.rrd
/usr/local/rrdtool/bin/rrdtool resize wanedge1-fastethernet0-1.rrd 4 GROW 6912
mv resize.rrd wanedge1-fastethernet0-1.rrd
/usr/local/rrdtool/bin/rrdtool resize wanedge1-fastethernet0-1.rrd 8 GROW 6912
mv resize.rrd wanedge1-fastethernet0-1.rrd
/usr/local/rrdtool/bin/rrdtool resize wanedge1-fastethernet0-1.rrd 1 GROW 3072
mv resize.rrd wanedge1-fastethernet0-1.rrd
/usr/local/rrdtool/bin/rrdtool resize wanedge1-fastethernet0-1.rrd 5 GROW 3072
mv resize.rrd wanedge1-fastethernet0-1.rrd
/usr/local/rrdtool/bin/rrdtool resize wanedge1-fastethernet0-1.rrd 9 GROW 3072
mv resize.rrd wanedge1-fastethernet0-1.rrd
```

This takes a second or two to complete. This process can be scripted easily enough, and has been, this is included in the NMIS8 distribution in /path/to/nmis8/admin/rrd_resize.pl, the script will determine what RRA's need to be updated and then resize them accordingly.

Where are NMIS RRD Files Stored?

NMIS stores the RRD files in the folder /path/to/nmis8/database, then in various sub-directories depending on what is required. The script rrd_file_update.pl has the code to traverse a directory structure. NMIS can easily be configured to store RRD files in another location, this is controlled by the NMIS config item database_root which by default is set to '<nmis_data>/database'.

Conclusion

NMIS8 is a powerful network management system which can keep as much data as you have disk space to store, the defaults make sense for most organisations and provide a good balance between statistical granularity and disk space.