Developing Device Models for NMIS

Related Material

Please checkout the Device Modelling Checklist

- Tools for Working with NMIS Models
- Modelling MIBS that use Indexes using the systemHealth section
- Tutorial: NMIS8, create a new model
- Tutorial: NMIS8, adding a new threshold

Table of Contents

Developing Device Models for NMIS

Checkout the handy training we developed for NMIS users, this is available here:

- NMIS8 Device Modelling Training
- NMIS8 Device Modelling Training (PDF of a PowerPoint presentation)
- NMIS8 Device Modelling Training Webcast

- Tutorial: NMIS8, adding a new threshold
- Tutorial: NMIS8 adding a new model
- Tutorial: NMIS8 adding a new model (continuation)

Checklist for Model Development

The training covers this in details, but here is a quick summary of what needs to be updated and changed when adding new device support with modelling

- Update nmis8/models/Model.nmis
- New model file created and updated
  - update the nodeVendor, nodeModel, nodeType fields (among others)
- Any new resolvable OID's added to nmis_oids.nmis, e.g. the sysObjectID names.
- Update Common-heading.nmis
- Update Common-database.nmis
- Create any needed graphs
- Update Common-threshold.nmis
- Update Common-stats.nmis

Creating Thresholds with NMIS Modelling

Please refer to the training material above for an overview of creating the thresholds, the information below should assist in understanding the relationships between the different sections and moving parts.

Main collection is /systemHealth/rrd/**hrProcessorLoad**

Common-database entry called **hrProcessorLoad**

Common-heading entry for the /systemHealth/rrd/hrProcessorLoad/graphtype => **hrprocload**

To create a threshold, statistical data is needed, this is extracted using Common-stats and this is based on the rrd name, so for this hrProcessorLoad collection, the Common-stats entry would be **hrProcessorLoad**

To compare the data to the threshold, there needs to be a Common-threshold section, this is named after the /systemHealth/rrd/hrProcessorLoad /threshold => **arCpuLoad**

So that the threshold system can match data from the stats results, we use the item to match the data, so NMIS is going to look for the item in the resulting stats data, in this case the item name is **hrProcessorLoad,** which matches the print statement: 'PRINT:hrProcessorLoad:AVERAGE:**hrProcessorLoad**=% 1.0f' in the Common-stats section.

```
### Model-AristaSwich.nmis
%hash = (
  'systemHealth' => {
    'rrd' => {
      'hrProcessorLoad' => {
        'graphtype' => 'hrprocload',
        'indexed' => 'true',
        'threshold' => 'arCpuLoad'
        'snmp' => {
          'hrProcessorLoad' => {
            'oid' => 'hrProcessorLoad',
            'option' => 'gauge,0:U'
          }
        }
      }

### Common-database.nmis
  'hrProcessorLoad' => '/nodes/$node/health/hrProcessorLoad-$index.rrd',

### Common-heading.nmis
  'hrprocload' => 'Processor Load',

### Common-stats.nmis
%hash = (
  'stats' => {
    'type' => {
      'hrProcessorLoad' => [
        'DEF:hrProcessorLoad=$database:hrProcessorLoad:AVERAGE',
        'PRINT:hrProcessorLoad:AVERAGE:hrProcessorLoad=%1.0f',
      ]

### Common-threshold.nmis
%hash = (
  'threshold' => {
    'name' => {
      'arCpuLoad' => {
        'item' => 'hrProcessorLoad',
        'event' => 'Proactive CPU Utilisation',
        'select' => {
          'default => {
            'value' => {
              'fatal' => '90',
              'critical' => '80',
              'major' => '70',
              'minor' => '60',
              'warning' => '50'
            }
          }
        }
      },
```