

# Basic and Advanced Thresholds in NMIS8

NMIS8 includes powerful capabilities for performance and operational thresholding, which greatly enhance network management capabilities. These thresholds result in alerts/events/notifications which NMIS can send when it sees a threshold breached. The thresholds have very granular controls which by default have been configured fairly broadly.

A list of standard thresholds NMIS includes is available in [NMIS Threshold Configuration](#)

A simple example of this is that you will likely need to be notified when the CPU is high on a Core device, vs an Access switch in Timbuktu. This is part of the NMIS idea of relevance of information in general, not all devices are created equal, some devices deserve better alerting, people of course are created equal.

If you want to just modify and add new thresholds read the first section "Using the NMIS GUI to Configure Thresholds", the second section "About Threshold Controls" goes into more details about thresholds.

As always thanks to our commercial customers who support everyone who use NMIS8, including the creation of documentation and these great features, if you are interested in commercial support, please email [contact@opmantek.com](mailto:contact@opmantek.com)

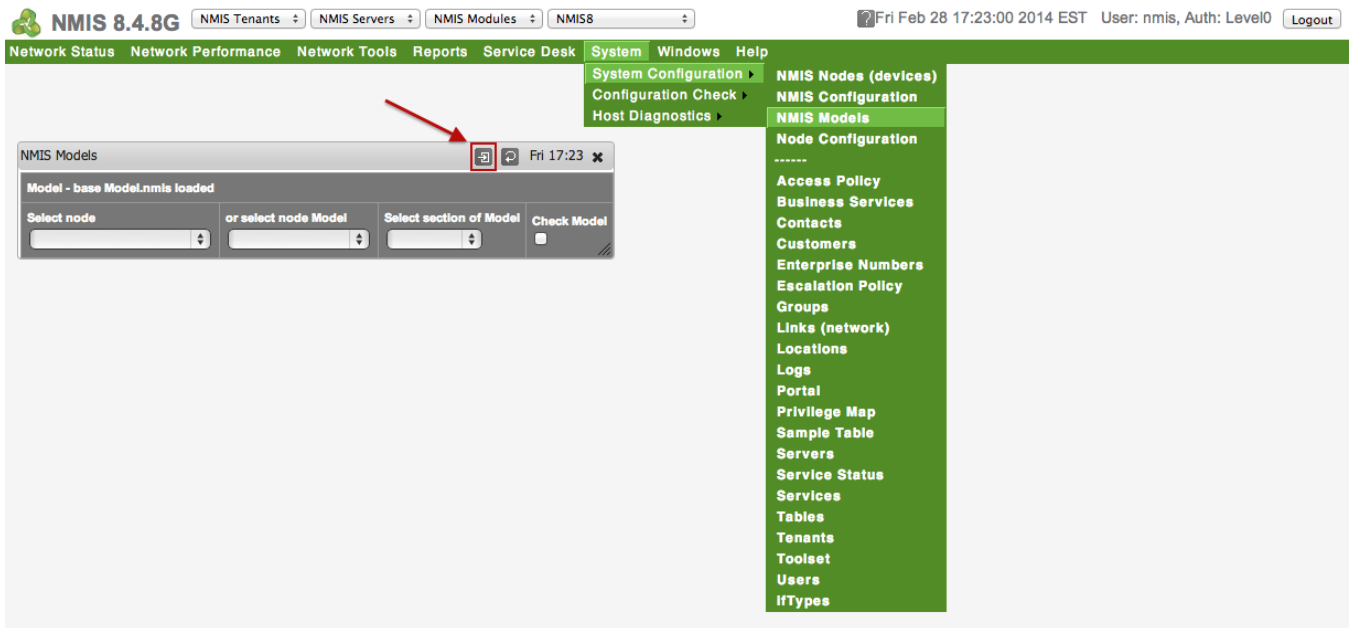
- [Using the NMIS GUI to Configure Thresholds](#)
  - [Access the NMIS Models GUI](#)
    - [NMIS 8.4 Example](#)
    - [NMIS 8.6 Example](#)
  - [Select the Section to Modify](#)
  - [Select the Threshold to Add a Controlled Select](#)
  - [Editing the Control](#)
  - [A New Control is Available](#)
  - [Changing a Threshold](#)
  - [Updated Threshold for Warning](#)
  - [Applying the Thresholds](#)
  - [Adding a New Select Block](#)
  - [Setting the Order and Thresholds](#)
  - [Adding a Control](#)
  - [New Threshold Select Added](#)
- [About Threshold Controls](#)
  - [Simple Thresholds](#)
  - [Have a set of thresholds for Core CPU](#)
  - [Advanced Thresholds with Controls](#)
  - [Advanced Control Options](#)
  - [Sample Controls](#)
  - [Threshold Dampening in NMIS](#)

## Using the NMIS GUI to Configure Thresholds

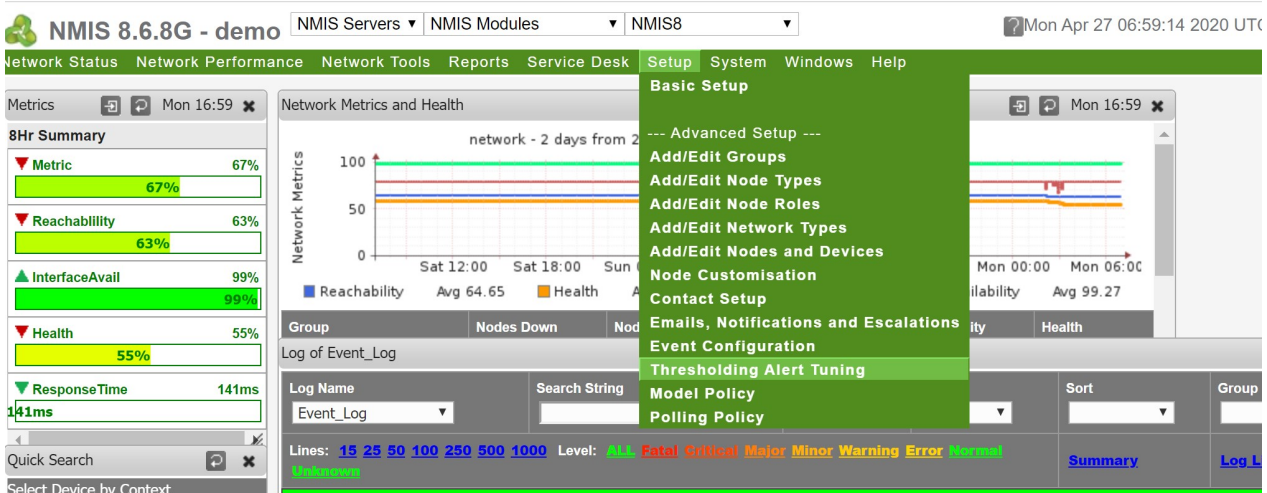
### Access the NMIS Models GUI

Access the GUI by clicking on the menu "System -> System Configuration -> NMIS Models", the "NMIS Models" widget will be displayed as shown below. Now this is a BIG form, so clicking on the POP OUT button will place it in its own browser TAB/Window. This is applicable in NMIS 8.4.8G and onwards.

### NMIS 8.4 Example



## NMIS 8.6 Example



## Select the Section to Modify

I have selected NMIS 8.6 Common-threshold.nmis to edit, we now just search for the "item" we want to change the threshold for.

Note: in older model formats e.g NMIS8.4 and before and in some older Modelling of devices you need to select the Nodes Model- file and the threshold section

Thresholding Alert Tuning

Select Model

Common-threshold

Select Section

threshold

Displaying Model Common-threshold, Section threshold

threshold	name							<a href="#">add</a>	
name		ApConnStatus							<a href="#">delete</a>
ApConnStatus			event	Proactive Ap Connection Status					<a href="#">edit</a>
ApConnStatus			item	ruckuCSN					<a href="#">edit</a>
ApConnStatus			select						<a href="#">add</a>
select				default					<a href="#">add</a> <a href="#">delete</a>
default					value				<a href="#">delete</a>
value						critical	0	<a href="#">edit</a>	
value						fatal	-1	<a href="#">edit</a>	
value						major	0	<a href="#">edit</a>	
value						minor	0	<a href="#">edit</a>	
value						warning	0	<a href="#">edit</a>	
ApConnStatus			title	Ap Connection Status: Disconnected					<a href="#">edit</a>
ApConnStatus			unit	blank					<a href="#">edit</a>
name		DiskQueueLength							<a href="#">delete</a>
DiskQueueLength			event	Proactive Disk Queue Length					<a href="#">edit</a>
DiskQueueLength			item	AvgDiskQLen					<a href="#">edit</a>
DiskQueueLength			select						<a href="#">add</a>
select				default					<a href="#">add</a> <a href="#">delete</a>
default					value				<a href="#">delete</a>

## Select the Threshold to Add a Controlled Select

In this example we will add a select section, which will be controlled to only work on a specific device named "meatball" and the interface is "Dialer1". To do this we are going to hi-jack the threshold in the first position and edit the control. This is because these selects are executed in order and we need ours to be first, this is like an access list on a router or firewall. We will add another in position 10 when we are done. I will also update the default models to have gaps in the orders.....

threshold	util_out			<a href="#">delete</a>
threshold		event	Proactive Interface Output Utilisation	<a href="#">edit</a>
threshold		Item	outputUtil	<a href="#">edit</a>
threshold		select		<a href="#">add</a>
threshold		1		<a href="#">add delete</a>
threshold		control (if true)	\$ifSpeed <= 5000000 and \$ifSpeed >= 1000000	<a href="#">edit</a>
threshold		value		
threshold		critical	90	<a href="#">edit</a>
threshold		fatal	95	<a href="#">edit</a>
threshold		major	80	<a href="#">edit</a>
threshold		minor	70	<a href="#">edit</a>
threshold		warning	60	<a href="#">edit</a>
threshold		2		<a href="#">add delete</a>
threshold		control (if true)	\$ifSpeed == 10000000	<a href="#">edit</a>
threshold		value		

## Editing the Control

Now we just change the control to be the string "\$node eq \"meatball\" and \$ifDescr eq \"Dialer1\"" and click "Edit".

threshold

name

util\_out

select

1

control

\$node eq "meatball" and \$ifDescr eq "Dialer1"

Edit

Cancel

•

control

Format: expression

An operator test will be executed on this rule. If the result is true then the oid's are executed.

The next names of variable can be used, they are replaced at runtime:

\$node

\$nodeModel

\$nodeType

\$nodeVendor

\$sysObjectName

\$ifDescr

\$ifType

\$ifSpeed

\$installedModems

At the Node page the value of \$sysObjectName is presented under the name sysName.

example: \$sysObjectName =~ /7300|2620/

## A New Control is Available

Now we can tune the thresholds by clicking "edit" on the one we want to change,

threshold	util_out			<a href="#">delete</a>
threshold		event	Proactive Interface Output Utilisation	<a href="#">edit</a>
threshold		item	outputUtil	<a href="#">edit</a>
threshold		select		<a href="#">add</a>
threshold		1		<a href="#">add</a> <a href="#">delete</a>
threshold		control (if true)	\$node eq "meatball" and \$ifDescr eq "Dialer1"	<a href="#">edit</a>
threshold		value		
threshold		critical	90	<a href="#">edit</a>
threshold		fatal	95	<a href="#">edit</a>
threshold		major	80	<a href="#">edit</a>
threshold		minor	70	<a href="#">edit</a>
threshold		warning	60	<a href="#">edit</a>

## Changing a Threshold

Lets change the warning to be 65%, then click "Edit"

Edit of Model CiscoRouter

threshold				
	name			
		util_out		
			select	
			1	
				value
				warning
				65
				Edit Cancel

Updated Threshold for Warning

Now the warning level is set to 65%

threshold		util_out			<a href="#">delete</a>
threshold			event	Proactive Interface Output Utilisation	<a href="#">edit</a>
threshold			item	outputUtil	<a href="#">edit</a>
threshold			select		<a href="#">add</a>
threshold			1		<a href="#">add</a> <a href="#">delete</a>
threshold				control (if true) \$node eq "meatball" and \$ifDescr eq "Dialer1"	<a href="#">edit</a>
threshold				value	
threshold				critical 90	<a href="#">edit</a>
threshold				fatal 95	<a href="#">edit</a>
threshold				major 80	<a href="#">edit</a>
threshold				minor 70	<a href="#">edit</a>
threshold				warning 65	<a href="#">edit</a>

Applying the Thresholds

The thresholds will be run on the next poll cycle or when they have been configured to run. You can run them on demand from the command line, with the following command (meatball is my router here, so change for your device).

```
/usr/local/nmis8/bin/nmis.pl type=threshold debug=true node=meatball
```

Adding a New Select Block

Now we need to add back the one we hi-jacked. So we add a new select block, click "add".

threshold		util_out		<a href="#">delete</a>
threshold		event	Proactive Interface Output Utilisation	<a href="#">edit</a>
threshold		Item	outputUtil	<a href="#">edit</a>
threshold		select		<a href="#">add</a>
threshold			1	<a href="#">add delete</a>
threshold		control (if true)	\$node eq "meatball" and \$ifDescr eq "Dialer1"	<a href="#">edit</a>
threshold		value		
threshold		critical	90	<a href="#">edit</a>
threshold		fatal	95	<a href="#">edit</a>
threshold		major	80	<a href="#">edit</a>
threshold		minor	70	<a href="#">edit</a>
threshold		warning	65	<a href="#">edit</a>

## Setting the Order and Thresholds

I am going to set this one as 10, it will happen after the others but before the default, complete all the details you require in the form and click "Edit"

threshold			
	name		
	util_out		
	select		
Add next part to Model CiscoRouter			
	order	10	<input type="text"/>
	fatal	95	<input type="text"/>
	critical	90	<input type="text"/>
	major	80	<input type="text"/>
	minor	70	<input type="text"/>
	warning	60	<input type="text"/>
<input type="button" value="Add"/> <input type="button" value="Cancel"/>			

- order  
Format: number  
Order of processing, starting at lowest number.
- fatal  
Format: number  
This number can be a normal value or percent, depending of the rules in stats. If the value of warning is higher then fatal then thresholds for higher being good and lower being bad.

## Adding a Control

We need to add a control field now, so on the new select section 10, click add

threshold		10		<a href="#">add</a> <a href="#">delete</a>
threshold		value		
threshold		critical	90	<a href="#">edit</a>
threshold		fatal	95	<a href="#">edit</a>
threshold		major	80	<a href="#">edit</a>
threshold		minor	70	<a href="#">edit</a>
threshold		warning	60	<a href="#">edit</a>

Complete the details in the control "\$ifSpeed <= 5000000 and \$ifSpeed >= 1000000" and click add.

threshold				
	name			
	util_out			
	select			
Add next part to Model CiscoRouter				
		control	\$ifSpeed <= 5000000 and \$ifSpeed >= 1000000	
		Add	Cancel	

• control  
Format: expression  
An operator test will be executed on this rule. If the result is true then the oid's are executed.  
The next names of variable can be used, they are replaced at runtime:

- \$node
- \$nodeModel
- \$nodeType
- \$nodeVendor
- \$sysObjectName
- \$ifDescr
- \$ifType
- \$ifSpeed
- \$installedModems

At the Node page the value of \$sysObjectName is presented under the name sysName.  
example: \$sysObjectName =~ /7300|2620/

## New Threshold Select Added

Now we have the old hi-jacked threshold created anew ready for low speed thresholding.

threshold		10		<a href="#">add</a> <a href="#">delete</a>
threshold		control (if true)	\$ifSpeed <= 5000000 and \$ifSpeed >= 1000000	<a href="#">edit</a>
threshold		value		
threshold		critical	90	<a href="#">edit</a>
threshold		fatal	95	<a href="#">edit</a>
threshold		major	80	<a href="#">edit</a>
threshold		minor	70	<a href="#">edit</a>
threshold		warning	60	<a href="#">edit</a>

## About Threshold Controls

### Simple Thresholds

In NMIS a simple threshold is defined by the following:



- the name
- the event name (which **must** begin with the phrase "Proactive" for correct event handling)
- a select (with a default and optionally more)
- default threshold values

In the file /usr/local/nmis8/models/Common-threshold.nmis this looks like this:

```
'cpu' => {
  'item' => 'avgBusy5min',
  'event' => 'Proactive CPU',
  'select' => {
    'default' => {
      'value' => {
        'critical' => '70',
        'fatal' => '80',
        'minor' => '50',
        'warning' => '40',
        'major' => '60'
      }
    }
  }
},
```

## Have a set of thresholds for Core CPU

BUT Core devices are more sensitive to CPU Load. So we want to use a different set of threshold values. Something like:

- 'critical' => '60',
- 'fatal' => '70',
- 'minor' => '40',
- 'warning' => '30',
- 'major' => '50'

But how to make these apply just to Core devices?

## Advanced Thresholds with Controls

For example, different thresholds for core devices. Looking in Common-thresholds will give you some ideas, but you can add many "selects" and have properties like:

- \$name
- \$node
- \$host
- \$group
- \$roleType
- \$nodeModel
- \$nodeType
- \$nodeVendor
- \$sysDescr
- \$sysObjectName
- others for interface
- Almost unlimited possibilities.

So we can create a more specific threshold for core devices (NMIS has this already configured by default).

```

'cpu' => {
  'item' => 'avgBusy5min',
  'event' => 'Proactive CPU',
  'select' => {
    '10' => {
      'value' => {
        'critical' => '60',
        'fatal' => '70',
        'minor' => '40',
        'warning' => '30',
        'major' => '50'
      },
      'control' => '$roleType =~ /core/'
    },
    --snip--
    'default' => {
      'value' => {
        'critical' => '70',
        'fatal' => '80',
        'minor' => '50',
        'warning' => '40',
        'major' => '60'
      }
    }
  }
},
},

```

These are executed in the select order, and if no control is matched, then the default set is used.

## Advanced Control Options

The following are the available control options

### Node Properties

- \$name
- \$node
- \$host
- \$group
- \$roleType
- \$nodeModel
- \$nodeType
- \$nodeVendor
- \$sysDescr
- \$sysObjectName

### Indexed Objects like interfaces

- \$ifAlias
- \$Description
- \$ifDescr
- \$ifType
- \$ifSpeed
- \$ifMaxOctets
- \$maxBytes
- \$maxPackets
- \$entPhysicalDescr

### Newly added indexed objects in NMIS 8.6G

- \$hrStorageDescr
- \$hrStorageType
- \$hrStorageUnits (disk block size)
- \$hrStorageSize (disk size in blocks)
- \$hrStorageUsed (disk used in blocks)
- \$hrDiskSize (disk size in bytes, hrStorageSize \* hrStorageUnits)
- \$hrDiskUsed (disk used in bytes, hrStorageUsed \* hrStorageUnits)
- \$hrDiskFree (disk free in bytes)

## Sample Controls

The controls are little pieces of code which will be evaluated when needed, so you might want to do the following sorts of things

Result	Control
Apply the threshold to all devices in the group "Sales"	\$group eq "Sales"
Apply the threshold to all devices starting with the IP address 192.168	\$host =~ /192\.168/
Apply the threshold to all Cisco IOS devices	\$sysDescr =~ /Cisco IOS/
Use this threshold if the interface speed is between 1 and 5 megabits/second	\$ifSpeed <= 5000000 and \$ifSpeed >= 1000000
Use this threshold if the interface speed is 10 megabits	\$ifSpeed == 10000000
Use this threshold if the interface speed is 100 megabits	\$ifSpeed == 100000000
Use this threshold if the interface speed is 1 gigabits	\$ifSpeed == 1000000000
Use this threshold if the disk is larger than 100 gigabytes	\$hrDiskSize >= 104857600000

## Threshold Dampening in NMIS

We have a feature in NMIS called threshold dampening, NMIS will only clear a threshold event when it passes the threshold + the dampening factor, this is configured with the configuration options `threshold_rising_reset_dampening` and `threshold_falling_reset_dampening`.

We added this feature because many times the threshold would drop a little, and clear the threshold, and then rise again, causing a new threshold event. So we provided an threshold clearing dampening factor to prevent the flaps.

There are two choices here, reduce the damping setting to a lower level, e.g. 1% which should be enough or disable the feature. You can edit this in the GUI it is in the section globals, or modify the Configuration directly `nmis8/conf/Config.nmis`

To reduce it to 1%, change the config values to be:

```
threshold_falling_reset_dampening = 1.01
threshold_rising_reset_dampening = 0.99
```

To disable it completely:

```
threshold_falling_reset_dampening = 1.0
threshold_rising_reset_dampening = 1.0
```

- [Basic and Advanced Thresholds in NMIS8](#)