Tools for Working with NMIS Models

- NMIS to Device Communications
  - Model Debug for an Update
  - Model Debug for a Collect
  - Example Model Debug Output
- Model Development Tool
  - Automated Base Model Detection and Creation
  - Discover What Existing Models Work on an NMIS Node
  - Check the Existing Models
    - Syntax Error in the Model File
    - Missing OID to Name definition in /usr/local/nmis8/mibs/nmis_mibs.oid
    - Variable Name for Storing in RRD longer than 19 Characters
    - Missing Graph file for the graphtype name
    - TODO - More Modelling Verifications
  - Rebuild the Model Schema
- Checking Models for specific nodes
- NIMS8 Model Tools - Dealing with Many Nodes and Many Models
  - Auditing your Nodes and Models
  - Which Models use which System Health Concepts
  - Releasing New Models to People
  - Patching Models with New Stuff
  - Comparing Two Directories of Models Contextually
  - Check the Syntax for all your Model files in a few Seconds
  - Show the model Structure

A core functionality of NMIS is the modelling system, this is an abstraction of how NMIS communicates with devices and allows a great deal of flexibility and control over how NMIS works.

There are several use tools and functions available with NMIS to help troubleshoot what is happening with device communication as well as when you are developing new models. This wiki provides details about what those tools are and how to use them.

In NMIS8 the model tool was called model_discovery.pl, but it did more than that, so it was renamed in NMIS9 to model_tool.pl (/usr/local/nmis9/admin/model_tool.pl)

This document will refer to using the modelling tool in NMIS9, along with some other useful tools in NMIS8 which still work, but are a little redundant with the new and improved model_tool.pl

NMIS to Device Communications

The NMIS CLI allows you to see what NMIS is doing when it communicates with a node, this output will tell you what data NMIS is requesting, which SNMP OID or WMI query is being used and what the device responds with.

Once you have added the node to NMIS you can run an update or collect and see what data is being requested:

Model Debug for an Update

NMIS 9:

Using the syntax below will result in a file created /tmp/update-PID-DATE.log

```
/usr/local/nmis9/bin/nmis-cli act=schedule job.priority=1 job.verbosity=3 job.output=/tmp/update job.force=true
job.type=update job.node=NMIS_NODE_NAME
```

NMIS 8:

```
/usr/local/nmis8/bin/nmis.pl type=update node=NMIS_NODE_NAME model=true force=true
```

Model Debug for a Collect

NMIS 9:

Using the syntax below will result in a file created /tmp/collect-PID-DATE.log

```
/usr/local/nmis9/bin/nmis-cli act=schedule job.priority=1 job.verbosity=3 job.output=/tmp/collect job.
force=true job.type=collect job.node=NMIS_NODE_NAME
```

NMIS 8:

```
/usr/local/nmis8/bin/nmis.pl type=collect node=NMIS_NODE_NAME model=true force=true
```

Example Model Debug Output

You will see below oid=biglongnumber, e.g. oid=1.3.6.1.4.1.9.9.43.1.1.1.0 and you will see oid=name, e.g. oid=processorRam, what is important to note is that SNMP ONLY uses OID's in dotted decimal notation form e.g 1.3.6.1.4.1.9.9.43.1.1.1.0, when NMIS sends

```
keiths@lodur:/usr/local/nmis8$ /usr/local/nmis8/bin/nmis.pl type=collect node=asgard model=true force=true
MODEL loadInfo asgard class=system:
  :  oid=1.3.6.1.4.1.9.9.43.1.1.1.0 name=configLastChanged value=2046
  :  oid=1.3.6.1.4.1.9.9.43.1.1.3.0 name=bootConfigLastChanged value=0
  :  oid=1.3.6.1.4.1.9.9.43.1.1.2.0 name=configLastSaved value=38597471
MODEL loadInfo asgard class=system:
  :  oid=1.3.6.1.4.1.9.9.25.1.1.1.2.5 name=softwareVersion value=12.4(25f)
  :  oid=processorRam name=processorRam value=99.61 Mb
  :  oid=chassisVersion name=chassisVer value=7.0
  :  oid=1.3.6.1.4.1.9.9.42.1.1.1.0 name=rttMonApplVersion value=2.2.0 Round Trip Time MIB
  :  oid=chassisId name=serialNum value=FHK11472788
  :  oid=1.3.6.1.4.1.9.9.25.1.1.1.2.2 name=softwareImage value=C1841-ADVENTERPRISEK9-M
  :  oid=1.3.6.1.4.1.9.9.42.1.1.13.0 name=rttMonApplResponder value=enabled
MODEL loadInfo asgard class=system:
  :  oid=sysContact name=sysContact value=default
  :  oid=sysDescr name=sysDescr value=Cisco IOS Software, 1841 Software (C1841-ADVENTERPRISEK9-M), Version 12.4
(25f), RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2011 by Cisco Systems, Inc.
Compiled Tue 16-Aug-11 06:21 by prod_rel_team
  :  oid=cmSystemInstalledModem name=InstalledModems value=0
  :  oid=snmpEngineTime name=sysUpTime value=44764700
  :  oid=sysObjectID name=sysObjectID value=1.3.6.1.4.1.9.1.620
  :  oid=ifNumber name=ifNumber value=10
  :  oid=sysName name=sysName value=asgard.localdomain
  :  oid=sysLocation name=sysLocation value=default
--etc--
```

Model Development Tool

The model discovery tool is new and is there to make things easier. It supports several useful functions when development new models or working with existing ones.

model_discovery.pl was released in NMIS8.7.0G and NMIS 9.2.2 but was an early version, check with Opmantek Support or grab a more recent version from GitHub: model_tool.pl

Automated Base Model Detection and Creation

This tool started a little more generically but it can now generate a base model automatically. Based on what it discovers, when starting with a new device type to become and NMIS node this is where you should start, it will take a default model and add all the things it can find which are already modelled, this would include things from the standard MIBs. First add your new thing into NMIS and get SNMP working, NMIS will use the Default model for this node which means SNMP is working and you can get more!

For example where ASGARD is the name of your node in NMIS.

```
/usr/local/nmis9/admin/model_tool.pl node=asgard model=CiscoRouter-auto common_exclude="Windows|^tcp"
```

A file called /usr/local/nmis9/models-custom/Model-CiscoRouter-auto.nmis would be created, you can now edit the node in NMIS and change model from auto to be CiscoRouter-auto and use this model.

You probably want to add some additional graphs to the nodegraph element but that can come later.

Discover What Existing Models Work on an NMIS Node

What existing modelling is there which could be useful. For example where ASGARD is the name of your node in NMIS.

```
/usr/local/nmis9/admin/model_tool.pl node=asgard file=/tmp/ASGARD.txt errors=false
```

The resulting file is a TAB delimited CSV and looks like below, it will help you understand which other existing modelling might work on this node :

node	nodeModel	Type	File	Path	Supported	SNMP_Object	SNMP_OID	OID_Used	result
ASGARD	CiscoRouter	system	Common-Cisco-netflow.nmis	Common/system/rrd/NetFlowStats/snmp/PktsExported	YES	1.3.6.1.4.1.9.9.387.1.4.4.0	1.3.6.1.4.1.9.9.387.1.4.4.0	1.3.6.1.4.1.9.9.387.1.4.4.0	307129
ASGARD	CiscoRouter	system	Model-CatalystIOS.nmis	Model/system/rrd/nodehealth/snmp/RouteNumber	YES	ipCidrRouteNumber	1.3.6.1.2.1.4.24.3	1.3.6.1.2.1.4.24.3	18
ASGARD	CiscoRouter	system	Common-mib2ip.nmis	Common/system/rrd/mib2ip/snmp/ipFragFails	YES	ipFragFails	1.3.6.1.2.1.4.18	1.3.6.1.2.1.4.18	0
ASGARD	CiscoRouter	system	Common-tcp.nmis	Common/system/rrd/tcp/snmp/tcpActiveOpens	YES	tcpActiveOpens	1.3.6.1.2.1.6.5	1.3.6.1.2.1.6.5	15892
ASGARD	CiscoRouter	system	Common-mib2ip.nmis	Common/system/rrd/mib2ip/snmp/ipForwDatagrams	YES	ipForwDatagrams	1.3.6.1.2.1.4.6	1.3.6.1.2.1.4.6	141405097
ASGARD	CiscoRouter	system	Model-CGESM.nmis	Model/system/rrd/nodehealth/snmp/avgBusy1	YES	avgBusy1	1.3.6.1.4.1.9.2.1.57	1.3.6.1.4.1.9.2.1.57	8
ASGARD	CiscoRouter	system	Model-CircuitMonitor.nmis	Model/system/sys/modem/snmp/InstalledModems	YES	cmSystemInstalledModem	1.3.6.1.4.1.9.9.47.1.1.1	1.3.6.1.4.1.9.9.47.1.1.1	0
ASGARD	CiscoRouter	system	Common-mib2ip.nmis	Common/system/rrd/mib2ip/snmp/ipReasmReqds	YES	ipReasmReqds	1.3.6.1.2.1.4.14	1.3.6.1.2.1.4.14	0
ASGARD	CiscoRouter	system	Model-ACME-Packet.nmis	Model/system/sys/standard/snmp/sysDescr	YES	sysDescr	1.3.6.1.2.1.1.1	1.3.6.1.2.1.1.1	Cisco IOS Software, 1841 Software (C1841-ADVENTERPRISEK9-M), Version 12.4(25f), RELEASE SOFTWARE (fc2)\nTechnical Support: http://www.cisco.com/techsupport\nCopyright (c) 1986-2011 by Cisco Systems, Inc.\nCompiled Tue 16-Aug-11 06:21 by prod_rel_team
ASGARD	CiscoRouter	system	Common-Cisco-netflow.nmis	Common/system/rrd/NetFlowStats/snmp/SampledPacket	YES	1.3.6.1.4.1.9.9.387.1.4.1.0	1.3.6.1.4.1.9.9.387.1.4.1.0	1.3.6.1.4.1.9.9.387.1.4.1.0	0
ASGARD	CiscoRouter	systemHealth	Common-Cisco-cbqos.nmis	Common/systemHealth/sys/Cisco_CBQoS	YES	cbQosIfType	1.3.6.1.4.1.9.9.166.1.1.1.1.2	1.3.6.1.4.1.9.9.166.1.1.1.1.2	$VAR1 = { '1.3.6.1.4.1.9.9.166.1.1.1.1.2.1085' => 1 };
ASGARD	CiscoRouter	system	Model-CGESM.nmis	Model/system/rrd/nodehealth/snmp/bufferElFree	YES	bufferElFree	1.3.6.1.4.1.9.2.1.9	1.3.6.1.4.1.9.2.1.9	1117
ASGARD	CiscoRouter	system	Common-mib2ip.nmis	Common/system/rrd/mib2ip/snmp/ipReasmOKs	YES	ipReasmOKs	1.3.6.1.2.1.4.15	1.3.6.1.2.1.4.15	0
ASGARD	CiscoRouter	system	Common-Cisco-system.nmis	Common/system/sys/extra/snmp/chassisVer	YES	chassisVersion	1.3.6.1.4.1.9.3.6.2	1.3.6.1.4.1.9.3.6.2	7

And what modelling will NOT work on this node.

node	nodeModel	Type	File	Path	Supported	SNMP_Object	SNMP_OID	OID_Used	result
ASGARD	CiscoRouter	system	Model-Teldat-iM8.nmis	Model/system/rrd/modem/snmp/InstalledModem	NO	1.3.6.1.4.1.2007.1.2.1.5	1.3.6.1.4.1.2007.1.2.1.5	1.3.6.1.4.1.2007.1.2.1.5	noSuchObject
ASGARD	CiscoRouter	system	Model-PulseSecure.nmis	Model/system/sys/extra/snmp/iveTotalHits	NO	1.3.6.1.4.1.12532.14.0	1.3.6.1.4.1.12532.14.0	1.3.6.1.4.1.12532.14.0	noSuchObject
ASGARD	CiscoRouter	systemHealth	Model-AlcatelASAM.nmis	Model/systemHealth/sys/eqptBoard	NO	eqptSlotPlannedType	1.3.6.1.4.1.637.61.1.23.3.1.2	1.3.6.1.4.1.637.61.1.23.3.1.2	$VAR1 = undef;
ASGARD	CiscoRouter	systemHealth	Common-Cisco-memory.nmis	Common/systemHealth/sys/ciscoMemoryPool	NO	MemPoolName			$VAR1 = undef;
ASGARD	CiscoRouter	system	Model-HuaweiSwitchesCore.nmis	Model/system/rrd/nodehealth/snmp/hwMemoryDevFree	NO	1.3.6.1.4.1.2011.6.3.5.1.1.2.0.1.0	1.3.6.1.4.1.2011.6.3.5.1.1.2.0.1.0	1.3.6.1.4.1.2011.6.3.5.1.1.2.0.1.0	noSuchObject
ASGARD	CiscoRouter	system	Model-Eltek.nmis	Model/system/rrd/power/snmp/ACMains	NO	1.3.6.1.4.1.12148.9.7.1.11.0	1.3.6.1.4.1.12148.9.7.1.11.0	1.3.6.1.4.1.12148.9.7.1.11.0	noSuchObject
ASGARD	CiscoRouter	system	Model-CiscoASR.nmis	Model/system/rrd/nodehealth/snmp/avgBusy1	NO	cpmCPUTotal1minRev.2	1.3.6.1.4.1.9.9.109.1.1.1.7.2	1.3.6.1.4.1.9.9.109.1.1.1.7.2	noSuchInstance

ASGARD	CiscoRouter	system	Model-MGE-ups.nmis	Model/system/rrd/upsvolt/snmp/upsinputvolt3	NO	upsInputVoltage.3	1.3.6.1.2.1.33.1.3.3.1.3.3	1.3.6.1.2.1.33.1.3.3.1.3.3	noSuchObject
ASGARD	CiscoRouter	systemHealth	Common-lldp.nmis	Common/systemHealth/sys/lldpLocal	NO	lldpLocPortDesc	1.0.8802.1.1.2.1.3.7.1.4	1.0.8802.1.1.2.1.3.7.1.4	$VAR1 = undef;
ASGARD	CiscoRouter	systemHealth	Model-f5big_ip.nmis	Model/systemHealth/sys/F5_CPU	NO	sysPlaneStatEntry	.1.3.6.1.4.1.3375.2.1.12.2.2.1	.1.3.6.1.4.1.3375.2.1.12.2.2.1	$VAR1 = undef;

Check the Existing Models

Basic mode, check the model files and look for any errors, many things are checked for, more will be added.

```
/usr/local/nmis9/admin/model_tool.pl check=true
```

By default this will only check the models-default directory.

```
0.07 Load all the NMIS models from /usr/local/nmis9/models-default
```

To check model files in other directories, you need to add models_dir to the command, for example checking models-custom directory.

```
/usr/local/nmis9/admin/model_tool.pl check=true models_dir=models-custom
```

As a result, we can see this is now checking the models-custom directory

```
0.06 Load all the NMIS models from /usr/local/nmis9/models-custom
```

Debug is optional and debug=3 will reveal lots of cool information if you like data modelling. You should check keeping an eye on these errors, examples below.

Syntax Error in the Model File

You will get the compile error first with the details and the name of the file after it.

```
String found where operator expected at (eval 219) line 41, near "}
      'threshold'"
        (Missing operator before
      'threshold'?)
  MODEL ERROR: Could not load Model-TELDATL1.nmis
```

Missing OID to Name definition in /usr/local/nmis8/mibs/nmis_mibs.oid

```
MODEL ERROR: with Model OID Model-POWERALERT-ups.nmis :: Model/system/sys/ups/snmp/upsoutputsrc UpsOutSource
```

Variable Name for Storing in RRD longer than 19 Characters

```
MODEL ERROR: RRD variable HbsAirLinkCompressedStatic found longer than 19
```

Missing Graph file for the graphtype name

```
MODEL ERROR: missing file for graph type cmtscpu: /usr/local/nmis8/models/Graph-cmtscpu.nmis
MODEL ERROR: missing file for graph type power: /usr/local/nmis8/models/Graph-power.nmis
```

TODO - More Modelling Verifications

- Foreach Vendor in Models.nmis, ensure that a model exists, ensure that the vendor matches the Enterprise name.

Rebuild the Model Schema

Run this to build the model schema automatically. This will make the file /usr/local/nmis8/conf/Model-Schema.nmis

```
/usr/local/nmis9/admin/model_tool.pl models_dir=models-default make_schema=1 check=true
```

Checking Models for specific nodes

There are some errors just found when loading a model for a specific device. So, a new argument can be passed to check for those specific errors:

```
/usr/local/nmis9/admin/model_tool.pl nodes=true
```

Released in NMIS 9.2.4.

NIMS8 Model Tools - Dealing with Many Nodes and Many Models

Auditing your Nodes and Models

The Model Audit tool is what you need a trivial tool which produces a TEXT TAB delimited output you can load into your favourite spreadsheet, so you can see which NMIS nodes are using which models and adjust accordingly.

```
/usr/local/nmis8/admin/model_audit.pl
```

Which Models use which System Health Concepts

The Model Summary tool is what you need a trivial tool which produces a TEXT TAB delimited output you can load into your favourite spreadsheet, so you can see which NMIS Models are polling for which things.

```
/usr/local/nmis8/admin/model_summary.pl
```

Releasing New Models to People

Sometimes you need to get all the models into ZIP file so you can release them to a customer.

```
[keiths@thor build]$ /usr/local/nmis8/build/make_models_release
Usage: make_models_release [-s] [-z target.zip] [model1.nmis graph2.nmis...]

checks (and with -s prints) the dependencies of all listed model/graph files,
and zips them all up if -z is given.

note: if not run out of the models-install dir, then the FIRST model file's
directory will be assumed to hold ALL relevant files!
```

Patching Models with New Stuff

A handy template for patching model files to do something new, instead of shipping around model files.

```
/usr/local/nmis8/admin/install_cisco_model_dev.pl
```

Comparing Two Directories of Models Contextually

Model files are structured files (code is not really that structured), so that you can compare what has changed in your models-install to your models folders we made a handy tool.

```
/usr/local/nmis8/admin/compare_models.pl
```

Check the Syntax for all your Model files in a few Seconds

Run it, can't hurt.

```
/usr/local/nmis8/admin/modelcheck.pl

--snip--
 Processing /usr/local/nmis8/models/Common-Windows-interface.nmis
  Processing /usr/local/nmis8/models/Model-TELDATL1.nmis
String found where operator expected at (eval 483) line 38, near "}
      'calls'"
        (Missing operator before
      'calls'?)
  Processing /usr/local/nmis8/models/Graph-lockstat.nmis
--snip--
```

Show the model Structure

Useful for some people

```
/usr/local/nmis8/admin/show_model_structure.pl
```