NMIS8 Developer Training

April 2013

Keith Sinclair, keiths@opmantek.com

NMIS Developer Training Agenda

Time	Session
30 mins	Using the NMIS API
30 mins	Customising the dashboard and adding widgets
5 mins	Custom Tables

- Look at model for Alcatel DSLAM, collect only on uplinks.
- A diagram showing the files which are part of the configuration.
  - Authentication system
  - Events, Escalations, Notifications

NMIS Development Topics

- Basic NMIS API's
- Importing and Exporting Nodes using API's
- Events, Escalations, Notifications
- NMIS Authentication and Authorisation system
- Device Models
- Customising the GUI

Opmantek Community Wiki

- Access all available documentation at the Opmantek Community Wiki. https://community.opmantek.com

- Register @ https://opmantek.com "Join Community" top right.

https://community.opmantek.com/display/NMIS/Home

Dashboard › NMIS › Home Browse Log In Search Confluence

Search

- Amount of Performance Data Storage NMIS8 Stores
- Default Credentials (Passwords) for NMIS8 VM
- Getting Started - Virtual Appliance Bundle & NMIS 8
- Logs, debugs and files which are useful when troubleshooting and resolving issues in NMIS
- Managing Servers and Services with NMIS8
- Moving the NMIS database to a new partition
- NMIS8 Configuration
- NMIS8 Virtual Machine
- NMIS8 VM Installation Guide
- NMIS 4.3.6e Installation Documentation
- NMIS 8 Installation Guide
- NMIS 8 Release Notes
- NMIS File Permissions
- NMIS Metrics, Reachability, Availability and Health
- Patching NMIS with 8.3.4G Update
- Patching NMIS with 8.3.6G Update
- Patching NMIS with 8.3.9G Update
- User Management in NMIS8
- Using SNMPv3 with NMIS for Secure Network Management

Home

Tools

Added by Community Admin, last edited by Keith Sinclair on Dec 20, 2012 (view change)

Welcome to the NMIS community page! Have you just downloaded NMIS (VM or source)? See the NMIS v8 Documentation section information to get you started and help you configure NMIS. If you are looking for information and documentation for one of our modules see the specific Module links below. Lastly, if you have just downloaded the VM Bundle with NMIS which includes all of our modules check out the **Getting Started - Virtual Appliance Bundle & NMIS 8** document to get you going.

NMIS v8 Documentation

Current NMIS v8 documentation is listed below.

- Getting Started - Virtual Appliance Bundle & NMIS 8
- NMIS8 Release Notes
- NMIS8 Installation guide - All you should need to know to get NMIS8 running on LInux
- NMIS8 VM Installation Guide - What you need to know to get NMIS8 VMware OVF ready for use on your VMware installation.
- NMIS8 Quick Start Guide
- NMIS8 Configuration Guide
- NMIS8 Virtual Machine - More information about the NMIS8 Virtual Machine.
- Managing Servers and Services with NMIS8
- NMIS Configuration Part 1 on the "show brain" blog
- NMIS Configuration Part 2 on the "show brain" blog
- Using SNMPv3 with NMIS for Secure Network Management

Opmantek Modules to Extend NMIS8

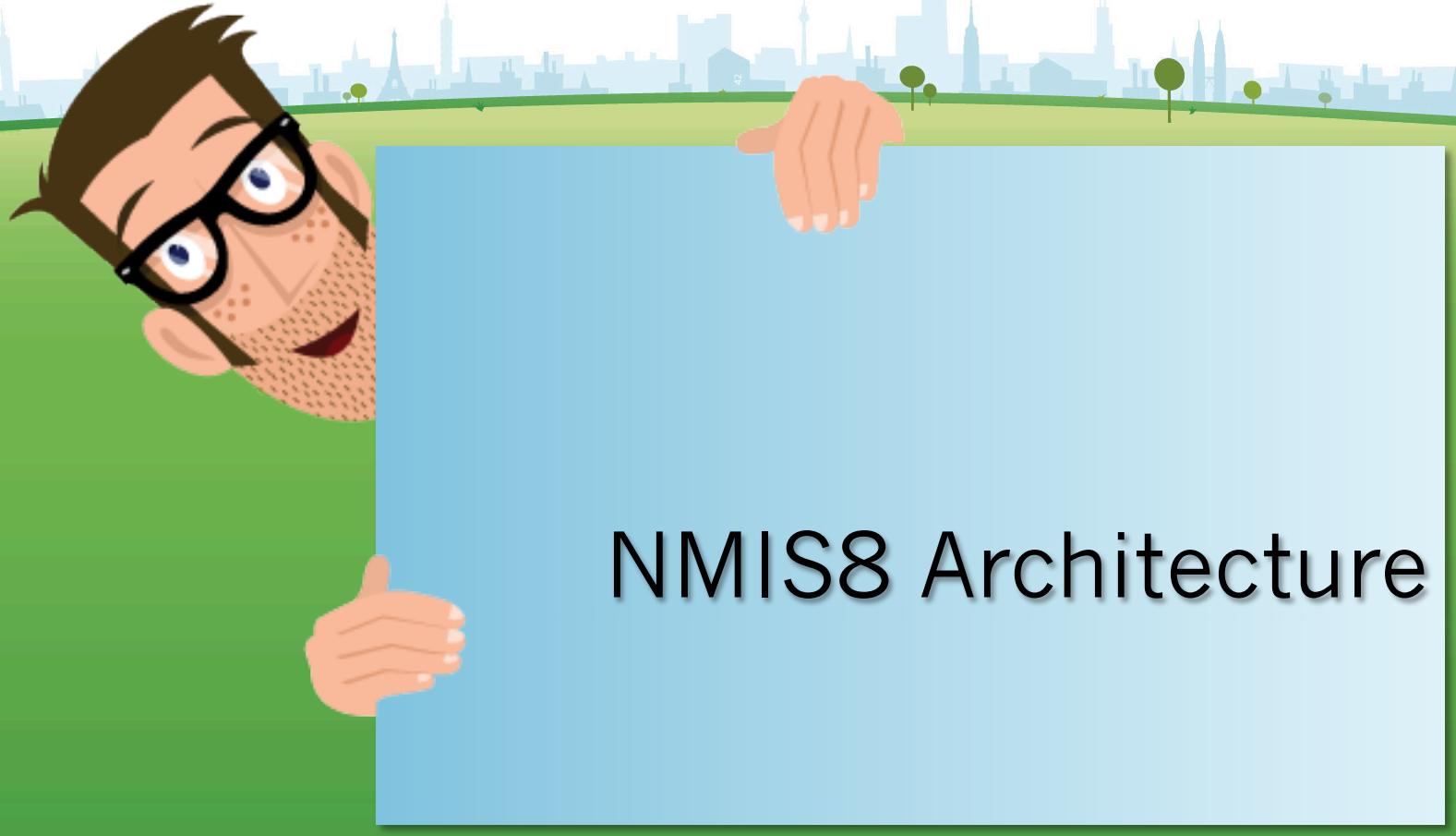
- opFlow - Application level visibility using NetFlow information, providing incredible visibility into bandwidth usage, by user and application.
- opReports - Expert reporting, with traffic lights and actions.
- opMaps - Geographical mapping of your Organisation, from NMIS information.
- opSLA - Enhanced IPSLA engine for 10000's of IPSLA probes on a single server.
- opHA - High availability for NMIS through Master/Slave and Multi-Master configurations.

NMIS in Depth

Why does NMIS do that, how does it work?

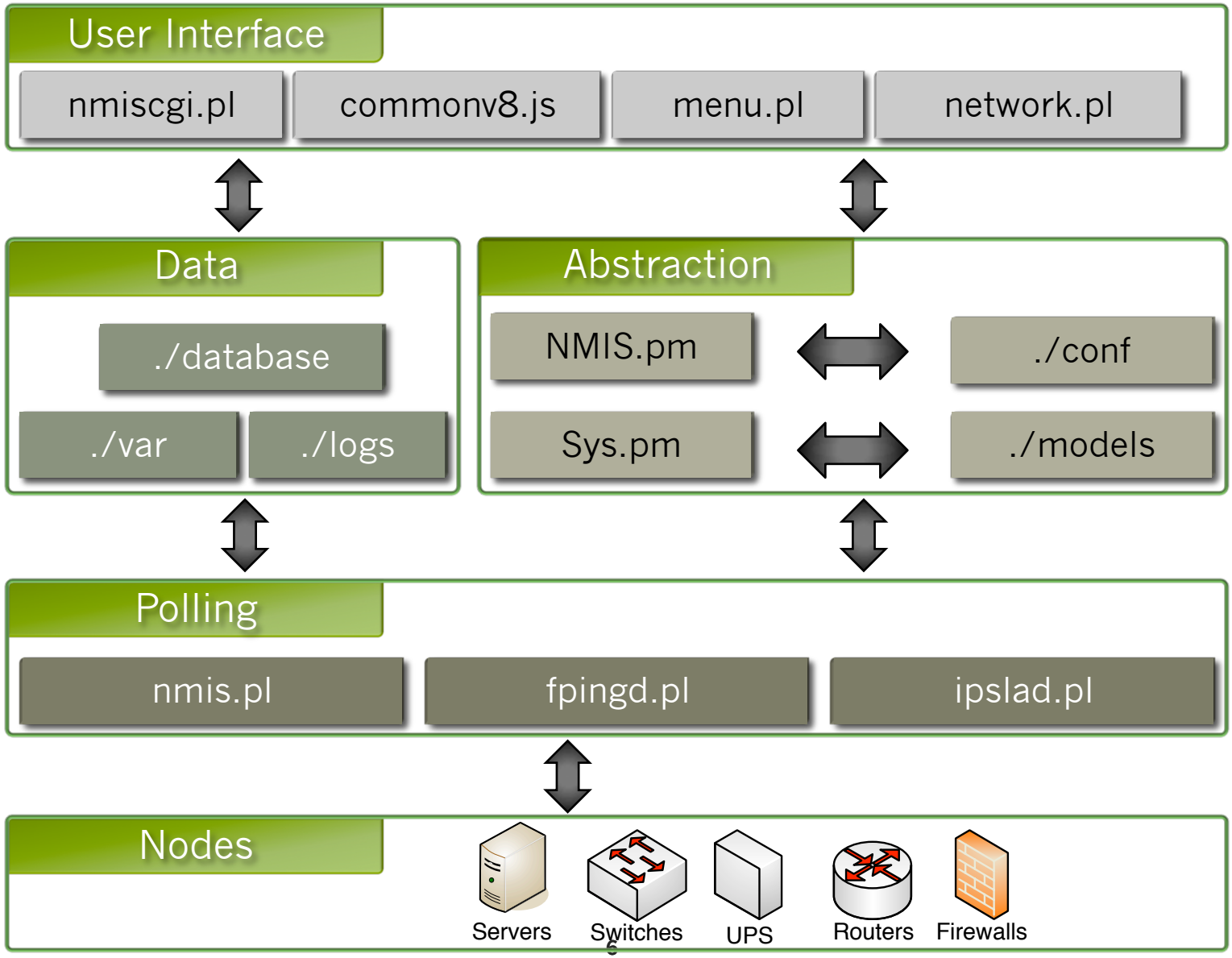
- NMIS Metrics, Reachability, Availability and Health
- Amount of Performance Data Storage NMIS8 Stores

NMIS Support



NMIS8 Architecture

NMIS8 Architecture



Using the NMIS8 API

Main API Calls – Working with the Node Table

```
use FindBin;
use lib "$FindBin::Bin/../lib";
use strict;
use func;
use NMIS;
my $C = loadConfTable();
my $LNT = loadLocalNodeTable();
foreach my $node (sort keys %{$LNT}) {
        print "Processing $node, group is $LNT->{$node}
{group} $LNT->{$node}{collect}\n”;
}
```

Main API Calls – Exporting Nodes as XML

```
my $LNT = loadLocalNodeTable();
open(XML,">nodes.xml") or die "Error with XML: $!\n";
print XML "<xml_nodes>\n";
foreach my $node (sort keys %{$LNT}) {
      print XML qq|
  <node>
    <name>$LNT->{$node}{name}</name>
    <group>$LNT->{$node}{group}</group>
  </node>
|;
}
print XML "</xml_nodes>\n";
close XML;
```

Main API Calls – Exporting Nodes as JSON

```
use JSON;
my $LNT = loadLocalNodeTable();
open(JSON,”>nodes.json”) or die “Error with JSON: $!\n”;
foreach my $node (sort keys %{$LNT}) {
      print JSON to_json( $LNT->{$node} );
}
close JSON;
```

OR

```
open(JSON,”>nodes.json”) or die “Error with JSON: $!\n”;
print JSON to_json( $LNT );
close JSON;
```

Main API Calls – Working with the Users Table

```
use FindBin;
use lib "$FindBin::Bin/../lib";
use strict;
use func;
use NMIS;
my $C = loadConfTable();
my $Users = loadUsersTable();
foreach my $user (sort keys %{$Users}) {
      print "$user, $Users->{$user}{groups} $Users-
>{$user}{privilege}\n";
}
```

Pushing NMIS data into a database

```
foreach my $node (sort keys %{$LNT}) {
        print "Processing $node, group is $LNT->{$node}
{group}\n”;

        my $stmt =<<EO_SQL;
INSERT INTO nodes(`node`, `group`)
VALUES ('$LNT->{$node}{name}', '$LNT->{$node}{group}’);
EO_SQL
        my $sth = $self->{_dbh}->prepare($stmt) or die
"ERROR prepare:\n$stmt\n$DBI::errstr";
        $sth->execute || warn "ERROR execute:\n$stmt\n
$DBI::errstr";

        $sth->finish();
}
```

Main API Calls

```
use FindBin;
use lib "$FindBin::Bin/../lib";
use strict;
use func;
use NMIS;
my $C = loadConfTable();
my $LNT = loadLocalNodeTable();
foreach my $node (sort keys %{$LNT}) {
        print "Processing $node, group is $LNT->{$node}{group}\n”;
        my $S = Sys::->new; # get system object
        $S->init(name=>$node,snmp=>'false');
        my $NI = $S->ndinfo;
        my $IF = $S->ifinfo;
        print "MATCH $NI->{system}{name} is $NI->{system}{sysDescr}\n";
        for my $ifIndex (keys %{$IF}) {
                print "$IF->{$ifIndex}{ifIndex}\t$IF->{$ifIndex}{ifDescr}\n";
        }
}
```

Node Table Entry

```
%hash = (
 'nmisdev' => {
    'cbqos' => 'none',
    'threshold' => 'true’,
    'model' => 'automatic',
    'roleType' => 'access',
    'webserver' => 'true',
    'community' => 'nmisGig8',
    'group' => 'DataCenter',
    'collect' => 'true',
    'calls' => 'false',
    'timezone' => '0',
    'depend' => 'N/A',
    'services' => 'http,http_server',
    'version' => 'snmpv2c',
    'name' => 'nmisdev',
    'active' => 'true’,
    'port' => '161',
    'host' => 'nmisdev',
    'rancid' => 'false',
    'netType' => 'lan',
    'ping' => 'true',
  },
);
```


Adding new Widgets

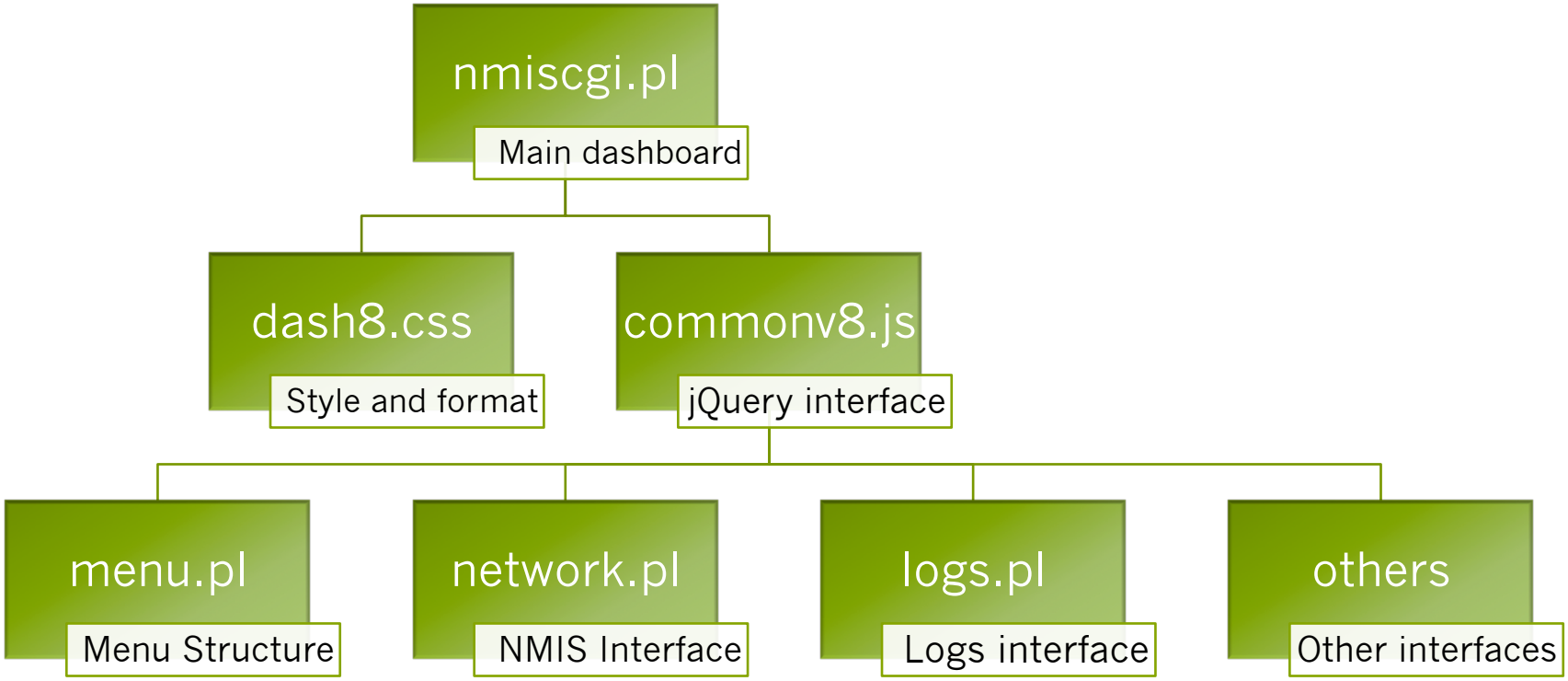
Components of NMIS8 UI

- <nmis8>/cgi-bin/nmiscgi.pl
- <nmis8>/cgi-bin/menu.pl
- <nmis8>/cgi-bin/network.pl
- <nmis8>/menu/js/commonv8.js

Libraries used by NMIS8 UI

- jQuery
- jdMenu

Relationship Between Components



Customising Dashboard

```
cd /usr/local/nmis8

cp cgi-bin/nmiscgi.pl cgi-bin/newdash.pl (e.g. bancopel.pl)

cp menu/js/commonv8.js menu/js/newdash.js (e.g. bancopel.js)



Edit newdash.pl, line 150

Before: {-type => 'text/javascript',-src => "$C->{'nmis_common'}"}

After: {-type => 'text/javascript',-src => "$C->{'<menu_url_base>'}/js/
newdash.js"}
```


Custom Tables

Custom Tables in NMIS

https://community.opmantek.com/display/NMIS/Custom+Tables+in+NMIS

- Added in NMIS 8.3.18G
- Allows adding arbitrary data to NMIS and integrate it as required.
- For example, adding information required to the Nodes table to allow it to be maintained in NMIS for the network and then export it to a CMDB or other target.


Review Parking Lot
Items


Conclusion/Summary