Using Unique Identifiers (UUID) for NMIS Nodes

Prerequisites

NMIS version 8.3.19G or greater.

The perl library Data::UUID to be installed.

Unix Shell access to the NMIS server and suitable Unix privileges to edit the NMIS configuration files, usually a member of the group "nmis" or the root user.

Summary

Some NMIS users require a Unique Identifier for each node, so Opmantek has added this for those users who require it. To have NMIS add a UUID for each node, you can enable this to be added and exported as required. This capability uses Custom Tables to have NMIS add a UUID when adding a node.

A UUID might be useful or required when integrating NMIS with third party applications like HP Service Manager, or when using NMIS as your source of truth for integration with a CMDB or other database system.

A new module has been added in NMIS 8.3.19G called NMIS::UUID, this is required to support UUID's.

Adding UUID Support

Modify the file /usr/local/nmis8/conf/Table-Nodes.nmis and add the following code.

```
use NMIS::UUID;

--snip--

my $uuid;
if ( $C->{uuid_add_with_node} eq "true" ) {
 $uuid = getUUID();
}

--snip--

{ uuid => { header => 'UUID',display => 'header,readonly',value => ["$uuid"] }},
```

In context these changes look like this:

```
use NMIS;
use Auth;
use NMIS::UUID;
my $C = loadConfTable();
# variables used for the security mods
my $AU = Auth->new(conf => $C); # Auth::new will reap init values from NMIS::config
# Calling program needs to do auth, then set the ENVIRONMENT before this is called.
$AU->SetUser($ENV{'NMIS_USER'});
my @groups = ();
my $GT = loadGroupTable();
foreach (sort split(',',$C->{group_list})) { push @groups, $_ if $AU->InGroup($_); }
my @nodes = ();
my $LNT = loadLocalNodeTable(); # load from file or db
foreach (sort {lc($a) cmp lc($b)} keys %{$LNT}) { push @nodes, $_ if $AU->InGroup($LNT->{$_}{group}); }
my @models = ();
if ( opendir(MDL,$C->{'<nmis_models>'}) ) {
 @models = ('automatic',sort {uc($a) cmp uc($b)} (grep(s/^Model-(.*)\.nmis$/$1/,readdir MDL)));
} else {
 print Tr(td({class=>'error'},"Error on loading models names from directory $C->{'<nmis_models>'}"));
}
closedir(MDL);
my $uuid;
if ( $C->{uuid_add_with_node} eq "true" ) {
 $uuid = getUUID();
}
return (
 Nodes => [ # using an array for fixed order of fields
 { name => { header => 'Name',display => 'key,header,text',value => [""] }},
 { uuid => { header => 'UUID',display => 'header,readonly',value => ["$uuid"] }},
 { host => { header => 'Name/IP Address',display => 'header,text',value => [""] }},
--snip--
```

Not adding UUID when node added

You can use the configuration uuid_add_with_node set to true to enable a UUID to be added or disable it with false to allow the other methods to maintain the UUID.

```
'uuid_add_with_node' => 'true',
```

Including the UUID in the Export

The sample script /usr/local/nmis8/admin/export_nodes.pl has been modified to include the UUID in the export. You can review this code to see how it is added. This code also uses the method createNodeUUID to create UUID's which might be missing from the nodes file.

Adding and Auditing UUID for Nodes

A script has been included /usr/local/nmis8/admin/uuid_update_nodes.pl which will add any missing UUID's as well as maintain/update the file /usr/local/nmis8/conf/UUID.nmis which is a two way index for determining the UUID of devices and visa versa. The error below was introducted intentionally.

```
[keiths@nmisdev64 nmis8]$ admin/uuid_update_nodes.pl
0.00 Begin
ERROR: the improbable has happened, a UUID conflict has been found for 59A2847C-8D41-11E2-A990-F38D7588D2EB,
between YINYANG and FOOBAR
0.02 Begin
```