opExport Installation Guide

- Prerequisites
  - MySQL Database / User
- Installation Steps
  - NMIS server
  - Additional Steps for the MySQL Database server
- Push the schema to all NMIS opExport Daemons
- Test SQL Server Connection
- Transferring Data
- Monitoring opExport Activity
- Error checking

Prerequisites

A working copy of NMIS that opExport can read and export the data from, and a running MySQL server.

Please note that there are some interoperability limitations present in opExport 1.6.7 and 1.6.8:

- opExport must be installed **BEFORE** any other opX products.
- opExport currently cannot coexist with other opX products on the MySQL server. (In client mode, i.e. on NMIS servers, coexistence works fine.)

MySQL Database / User

To create a new mysql database you will need to run a command like this

```
mysqladmin create opexport -u root -p
```

After creating a new database you will want to create a new user and grant them access to the new database:

```
mysql -u root -p
mysql> CREATE USER 'opexport'@'localhost' IDENTIFIED BY '42opexport42';
Query OK, 0 rows affected (0.00 sec)
mysql> GRANT ALL ON opexport.* to 'opexport'@'localhost';
Query OK, 0 rows affected (0.01 sec)
```

Installation Steps

NMIS server

- copy the tarball to the server, into your users home directory
- untar the file into a location of your choice, the default location is /usr/local/omk

```
# make sure the omkd daemon is not running
ps aux | grep opman
kill <pid_of_opmantek>

# start the install
cd /usr/local
tar xvf /tmp/opExport-Linux-x86_64-LIB-1.5.16.tar.gz
cd omk
cp install/omk-rotate.conf /etc/logrotate.d/
cp ./install/opCommon.nmis ./conf/
cp ./install/users.dat ./conf/
cp ./install/omkd.init.d /etc/init.d/omkd
chkconfig --add omkd

# make sure that "opExport" is listed in the omkd/load_applications configuration entry
# make sure the key is correct in opCommon, default is key123
vi ./conf/opCommon.nmis
:%s/key123/newkey123/
:wq!

service omkd start
# check to make sure it's running
ps aux | grep opman
root 16116 39.0 2.2 487632 283272 pts/0 R 05:39 6:47 opmantek.exe
root 21118 0.0 0.0 103224 844 pts/0 S+ 05:56 0:00 grep opma
```

- Now load up the GUI and **enter your license key and accept the EULA**:
- default username=nmis password=nm1888

```
http://your.server.names.or.ip:3000/omk/opExport/
```

- Now PUSH the schema to the NMIS server

Additional Steps for the MySQL Database server

- copy the schema files to the conf directory, this should **ONLY** be done on the MySQL server, not on the NMIS servers
- All of the above instructions should be done, then stop the daemon again

```
cd /usr/local/omk
cp ./install/schema_set.json ./conf/
cp ./install/export-* ./conf/

# enable the helper daemon
vi ./conf/opCommon.nmis
# search for this line:
'opexport_run_helper_daemon' => 'false', # required for MySQL servers, should be off for all others
# change it to true!
'opexport_run_helper_daemon' => 'true', # required for MySQL servers, should be off for all others
```

- Add the mysql connection options to conf/opCommon.nmis, the defaults are listed below, note: Do this ONLY on the server hosting the MySQL, the daemons on the NMIS servers do not use these options

```
'opexport' => {
 'opexport_sql_db_host' => 'localhost',
 'opexport_sql_db_port' => '3306',
 'opexport_sql_db_name' => 'opexport',
 'opexport_sql_db_user' => 'opexport',
 'opexport_sql_db_password' => '42opexport42',
 }
```

- RESTART the daemon

```
service omkd stop
ps aux | grep opman
# make sure it's not running

service omkd start

# verify the SQL database credentials by visiting
http://your.MYSQL.SERVER.NAME.or.ip:3000/omk/opExport/database_test_connect

# the output should say something like this:
{ 'message' => 'count of information_schema.tables: $VAR1 = [ [ \'47\' ] ]; ', 'success' => 'true' }
```

Push the schema to all NMIS opExport Daemons

The server that holds the MySQL Database holds the master copy of the schema, in order for an NMIS server to export it's daemon requires a schema to be pushed to it. NOTE: in this case the schema referred to is in opExport, not MySQL. opExport will use it's schema to create the MySQL tables for you.

Each NMIS server that will export it's data must be able to access the MySQL server with a simple host name (no dots), so you may need to edit /etc/hosts and add entries to allow this.

Additionally, the MySQL server requires the same for all the NMIS servers that will push to it, so again you may need to add them to it's /etc/hosts file.

Next, check that the MySQL server has a schema configured. From the dashboard loaded previously:

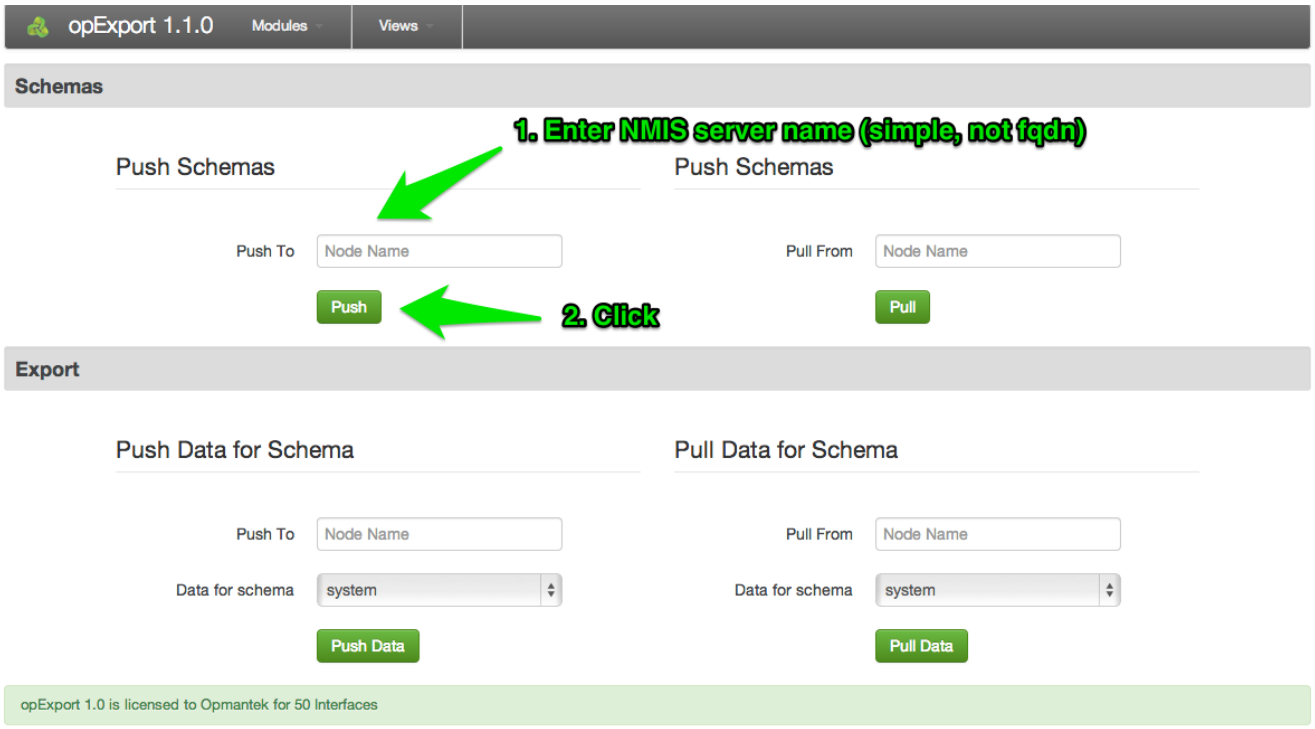
Or you can go directly to:

```
http://your.MySQL.server.name.or.ip:3000/omk/opExport/schemas?opexport_connection_key=key123
```

For each NMIS server you wish to have export you need to push the schema, in a web browser load the dashboard again:

```
http://your.MySQL.server.name.or.ip:3000/omk/opExport/?opexport_connection_key=key123
```

From here you can push the schema to each of your NMIS servers (which are running opExport). To push the schema fill out the form and submit it:



You can now verify that each of the NMIS servers have the schema:

```
http://your.NMIS.server.name1.or.ip:3000/omk/opExport/schemas?opexport_connection_key=key123
```

Test SQL Server Connection

The daemon on the SQL server can be tested to ensure it can connect to the database.

```
http://your.MySQL.server.name.or.ip:3000/omk/opExport/database_test_connect
```

This should return a message that looks like this:

```
{ 'message' => 'count of information_schema.tables: $VAR1 = [ [ \'32\' ] ]; ', 'success' => 'true' }
```

If it cannot connect you might see an error like this

```
'Connection to database not found, database handle is null
Access denied for user \'opexport\'@\'localhost\' (using password: YES)'
```

Transferring Data

There are two ways to transfer, the MySQL server can "pull" the data, or the NMIS server can "push" the data. You can load the dashboard from before (http://your.MySQL.server.name.or.ip:3000/omk/opExport/) to help you push/pull the data (the dashboard can also be loaded from the NMIS server)

Data to be pushed is available from these schemas:

diskIOTable
ciscoConfig
interface
interfaceStatus
nmisConfig
nodeStatus
services
storage
system

interfacePerformance
ipslaPerformance
systemPerformance
cbqosPerformance
upsPerformance

The drop-down on in the dashboard should allow you to choose which schema you would like to pull. If it is empty you the server does not have any schemas and requires them to be pushed to it from the opExport daemon on the MySQL server (see the installation instructions above for this).

To transfer data we ask the MySQL server to get the data from NMIS and save it:

```
#with curl
curl -s "http://localhost:3000/omk/opExport/request/pull/?
data_source=schema&node_source=NMIS_SERVER&data_source_name=system&opexport_connection_key=key123" &> /dev/null

 # or with wget:
/usr/bin/wget -qO- "http://localhost:3000/omk/opExport/request/pull/?
data_source=schema&node_source=NMIS_SERVER&data_source_name=interfaceStatus&opexport_connection_key=key123" &>
/dev/null
```

In CRON that might look something like this:

```
# On the MySQL Server: (so pulling, system data in this case)
*/5 * * * * /usr/bin/wget -qO- "http://localhost:3000/omk/opExport/request/pull/?
data_source=schema&node_source=NMIS_SERVER&data_source_name=system&opexport_connection_key=key123" &> /dev/null
```

NOTE: This is an example, you will want to think about which data you want and how often you want to update it. Several wget lines will likely be required.

Monitoring opExport Activity

Tailing the logs:

```
# on NMIS server
tail -f /usr/local/omk/log/opExport.log

# on MySQL you can also watch the helper daemon
tail -f /usr/local/omk/log/helper_daemon.log
```

Error checking

Logs are located in /usr/local/omk/log. All logs are helpful in finding errors. opDaemon.log will show when there has been a 500 error.

Also make sure that you have accepted the EULA on the machine, that the opexport_connection_key has been set correctly and that the daemon has been restarted after any config changes!