Contents

[Nimsoft\_robot\_status 1.1 2](#_Toc111802531)

[1 – Setup 3](#_Toc111802532)

[1.1 Extract zip file 3](#_Toc111802533)

[1.2 Nimsoft\_generic.dat 3](#_Toc111802534)

[1.3 Execution parameters 4](#_Toc111802535)

[1.3.1 Report output 4](#_Toc111802536)

[1.3.2 Include/Exclude 5](#_Toc111802537)

[1.3.3 Alarms 5](#_Toc111802538)

[1.3.4 Probe reporting 5](#_Toc111802539)

[2 Probe Output (Report and Alarm) 6](#_Toc111802540)

[2.1 Command Line output 6](#_Toc111802541)

[2.2 report\_nimsoft\_robot\_status\_ok.txt 7](#_Toc111802542)

[2.3 report\_nimsoft\_robot\_status\_nok.txt 8](#_Toc111802543)

[2.4 report\_nimsoft\_robot\_status\_hub.txt 9](#_Toc111802544)

[2.5 report\_nimsoft\_robot\_status\_subnet.txt 9](#_Toc111802545)

[3 - Tool Logic 10](#_Toc111802546)

[4 - Quick Start 10](#_Toc111802547)

[*5* - Nimsoft\_generic.dat (settings) 11](#_Toc111802548)

[5.1 SQL driver 14](#_Toc111802549)

[5.2 ODBC System DSN 15](#_Toc111802550)

[6 - Create Windows Perl environment 22](#_Toc111802551)

# Nimsoft\_robot\_status 1.1

This tool will list all you hubs, robots and probes and generate basic overview reports of your environment.

* Check hub status
* Check robot status
* Check probe status
* Generate Overview reports:
  + Status ok
  + Status nok
  + Subnet overview
  + Hub overview
* Generate optional alarms in case of hub, robot and/or probe problems
* Exclude in regex format hub, robot and/or probes

# – Setup

This tool was written in Perl and is packaged with the Perl source and with the compiled Perl. If you don’t have Perl installed on your hub, you can use the compiled version.

## Extract zip file

* Export: **nimsoft\_robot\_status\_1.1.zip** into a directory of your choice

**Note:** the zip file version number can be different that noted here

## Nimsoft\_generic.dat

The tool zip file contains nimsoft\_generic.dax to avoid overwriting your possible previous custom settings.

To avoid that you have to repeat all security and server setup parameters for each tool, we created a unique parameter file that contains all important information:

* Uim server
* Sql server
* Userid’s
* Password in encrypted format

If you use already a tool that use this nimsoft\_generic.dat file, you can copy that already customized file in the directory where you copied this tool (and go directly to next section)

If this is your first tool that uses nimsoft\_generic.dat, perform the steps explained in chapter 7.

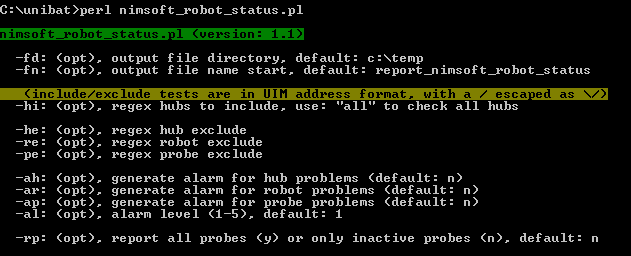
**Note1**: the file nimsoft\_generic.pm is used to decode the nimsoft\_generic.dat file

**Note2**: if you use multiple tools that use nimsoft\_generic.dat, and you want that all tools use the same version of nimsoft\_generic.dat, you can copy the file to your perl/lib directory

## Execution parameters

This tool has no configuration file, all parameters are command line options.

If you execute the tool without parameters it will display the possible execution options:



### Report output

* **-fd:** File Directory: directory where the report files are to be created, default: c:\temp
* **-**fn: File Name: file name start that will be used for the reports, default: report\_nimsoft\_robot\_status. This will result in reports:
  + report\_nimsoft\_robot\_status\_ok
  + report\_nimsoft\_robot\_status\_nok
  + report\_nimsoft\_robot\_status\_hub
  + report\_nimsoft\_robot\_status\_subnet

### Include/Exclude

Optionally, you can use several include/exclude options to avoid known problems from your reports.

These regex include/exclude options will be matched against the UIM address format:

/domain/hub/robot/probe

Each exclude is displayed in the output so that you can check if your regex format is what you had in mind.

* **-hi:** Hub Include. Check only the hub that is matching this include regex
* **-he**: Hub Exclude. Exclude the hubs that are matching this regex.
* **-re**: Robot Exclude. Exclude robot(s) matching the regex.
* **-pe**: Probe Exclude. Exclude the probe(s) matching the regex

### Alarms

By default this tool will NOT generate alarms in your console.

* **-ah**: Alarm Hub. Create an alarm when hub problems are detected. Default: n.
* **-ar**: Alarm Robot. Create an alarm when robot problems are detected. Default: n.
* **-ap:** Alarm Probe. Create an alarm when probe problems are detected. Default: n
* **-al:** Alarm Level. Alarm level (1-5) used for the possible alarms. Default: 1.

### Probe reporting

* **-rp:** Report all Probes (y) or only the probes with problems (n). Default: n.

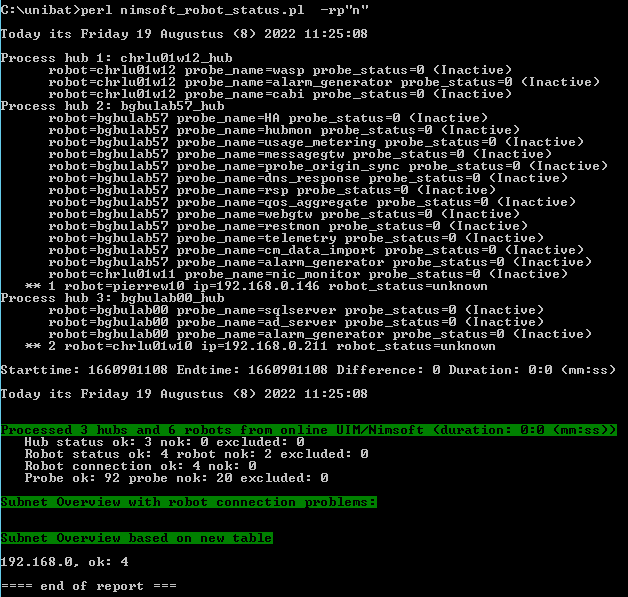
# Probe Output (Report and Alarm)

## Command Line output

When executing the tool from a command prompt it will write the results during execution. This was done so that you could follow what hubs he was processing.

This was also done so that you could have a feeling which hub was taking a long time to process. The tool uses internal UIM callbacks to obtain the hub, robot and probe information. This is executed with a retry: 1 and timeout: 10. If you have a hub where a lot of robots/probes are not responding this will influence the execution time of the tool.

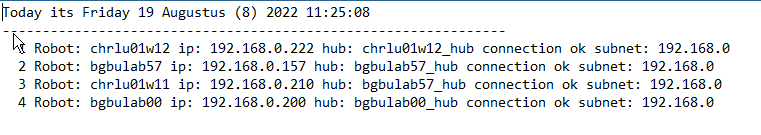
The command output will only display robots or probes with problems (=inactive):



Note1: it is normal that you HA hub will contain a lot of inactive probes.

Note2: the information displayed in this command output will also be generated in output files.

## report\_nimsoft\_robot\_status\_ok.txt



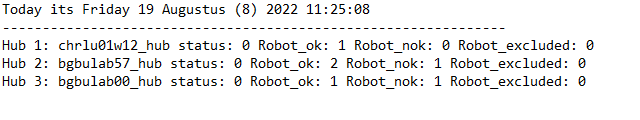
This file will contain an overview of all robots that responded to the callback: probe\_list.

## report\_nimsoft\_robot\_status\_nok.txt



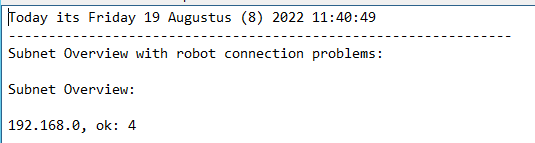
This file contains all robots and probes that have some sort of problem. This can be inactive or connection problem.

## report\_nimsoft\_robot\_status\_hub.txt



This file contains an overview of the processed hubs with some overview counters for each hub.

## report\_nimsoft\_robot\_status\_subnet.txt



This subnet report ha s2 parts.

Part 1 will display all subnets with robots that have a connection problem. This could be in a new implementation that one subnet has no authorization to port 48000.

Part2 will display all encountered subnets with some totals.

# - Tool Logic

* Get hub address: nimGetVarStr(NIMV\_HUBADDR)
* Execute on hub address callback: **gethubs** (get list of known hubs)
* Loop in returned hublist table::
  + If hub is in hub\_exclude, bypass this hub + log
  + If hub is not active alarm (if activated)
  + Execute the hub callback: **getrobots** to obtain a list of robots under this hub
  + Check if robot is excluded, if yes process next robot
  + Execute callback: **probe\_list** (get list of probes)
  + Loop in returned hash:
    - If the probe is in the exclude (-pe), bypass this probe + log
    - If the probe is inactive create an alarm if activated via –ap +log

# - Quick Start

The easiest way to start with this tool is to run it with only one hub in the include parameter, this to check if the tool is running file.

Perl nimsoft\_robot\_status.pl –rp”n”

-rp”n”: report only the inactive probes.

Note: this tool needs at least 1 parameter, else the usage will be displayed.

# - Nimsoft\_generic.dat (settings)

All custom reporting tools use a common parameter file: **nimsoft\_generic.dat**

# -- UMP server & port & http/https

**uim\_server**=ump\_server\_name

**uim\_port**=80

**uim\_https**=http

# --- Nimsoft userid and crypted password (via nimsoft\_crypt.exe)

**uim\_user**=administrator

**uim\_password**=gWL/M/ij/

# --- Nimsoft domain, hub and robot to create address to the main hub

**uim\_domain**=xxx\_domain

**uim\_hub**=xxx\_hub

**uim\_robot**=xxx

# --- Wasp realm (used by Rest and UIMAPI) pre 20.4: NimJAAS from 20.4: wasp-engine

uim\_realm=wasp-engine

# --- SQL server, userid, crypted password (via nimsoft\_crypt.exe) and databasename

**sql\_server**=sql\_server\_name

**sql\_user**=sa

**sql\_password**=gWL/M/ij/

**sql\_db**=CA\_UIM

**sql\_type**=mssql

# - sql\_driver: "SQL Server" (=default) or a manual installed newer driver, example: "ODBC Driver 17 for SQL Server" (\* no quotes around driver name \*)

**sql\_driver**=SQL Server

# - in case you need to use a not current logged on Windows user

**sql\_dsn**=

# --- end of parameters ---

The 2 passwords are stored in an encrypted form. To generate this encrypted password, use:

nimsoft\_crypt.exe your\_password

As output, you will receive the string that you can copy as encrypted password in the above file.

**Note1**: this nimsoft\_generic.dat is common between multiple tools. It is possible that the tool you are working with is not using all variables in this file.

**Note2**: if you use "sql\_user=trusted" and use also "sql\_password=" we will connect to MSSQL via a trusted connection. (=your logged on userid)

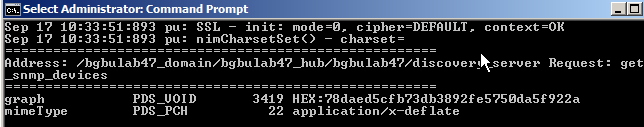
**Note3**: the uim\_robot must be defined in the format/case that UIM recognize them. (like it's displayed in IM)

**Note4:** sql\_driver: previously we used the hardcoded sql driver “SQL Server” that is by default installed in Windows systems. But when you want to connect to TLS 1.2 MSSQL you need to install (manually) a newer MSSQL driver (example: ODBC Driver 17 for SQL Server)

**Note5**: In case you receive a "communication error" while using a non-simulation execution of the tool, try to use the: /uim\_domain/uim\_hub/uim\_robot values like you use it in nimsoft\_generic.dat in the command:

pu -u administrator -p ??? /bgbulab47\_domain/bgbulab47\_hub/bgbulab47/discovery\_server get\_snmp\_devices

This commands must give a normal output like:



These names are case sensitive.

**Note6:** in case you want to use a user defined system DSN to perform the logon the MSSQL you can use the sql\_dsn parameter. Else keep this parameter blank.

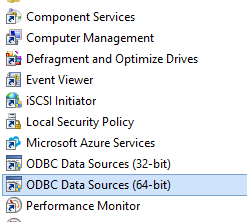
**Note7:** uimapi calls before 20.40 did use a realm: NimJAAS. But uimapi 20.4 introduced an undocumented change into realm: wasp-engine. This value is used by some of the utilities that use UIMAPI.

## SQL driver

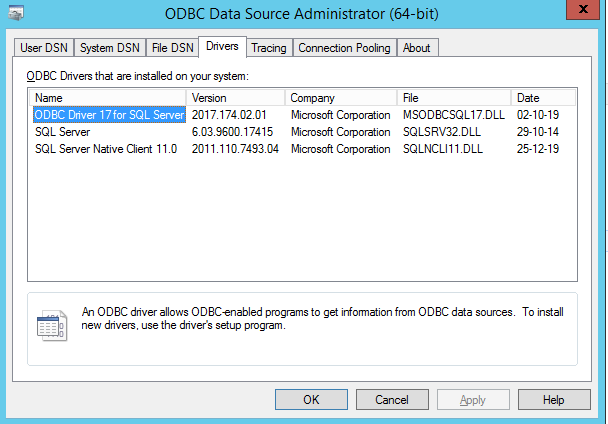
By default you will find the SQL driver: “SQL Server” installed on your Windows server.

But if you want/need to use TLS, it’s possible that you need a newer/updated version of the driver.

With the parameter: sql\_driver you can than enter the driver name, **without** surrounding quotes, that you installed.



Drivers can be found under: Administrative tools – ODBC Data Sources.



Under the tab: Drivers you can find the installed and available SQL drivers that you can use. You need to enter the “Name” as value of the parameter.

## ODBC System DSN

In nimsoft\_generic.dat you have several possibilities to define access to the CA\_UIM database.

The easiest way is to use an internal MSSQL user:

* sql\_user: sql userid
* sql\_password: password value encrypted with nimsoft\_crypt

If you want to access the CA\_UIM database with the current logged on user, you can use:

* sql\_user: trusted
* sql\_password:

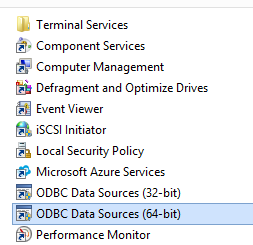
(the value kept empty)

If you want to use a Windows user to logon to SQL, not the current logged on user, you must use a pre-defined ODBC System DSN. The name you give to this System DSN is entered as value:

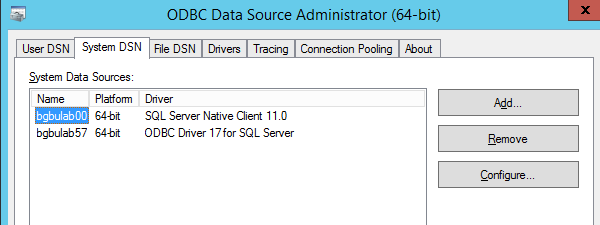
* sql\_user: windows user
* sql\_password: encrypted windows password
* sql\_dsn: name given to the ODBC System DSN

**Note1:** If you use one of the tools as probe or in a probe with a logon as Windows User, you **MUST** run the Nimsoft Service also with a Windows User (not needed to be the same Windows User)

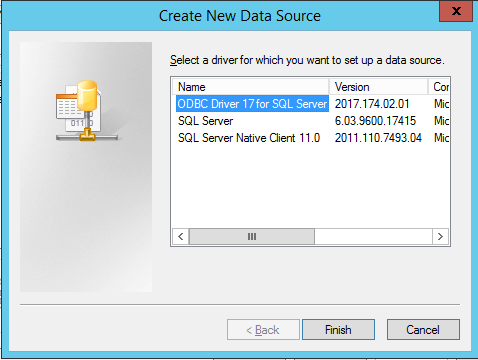
Under Administrative tools you can find:



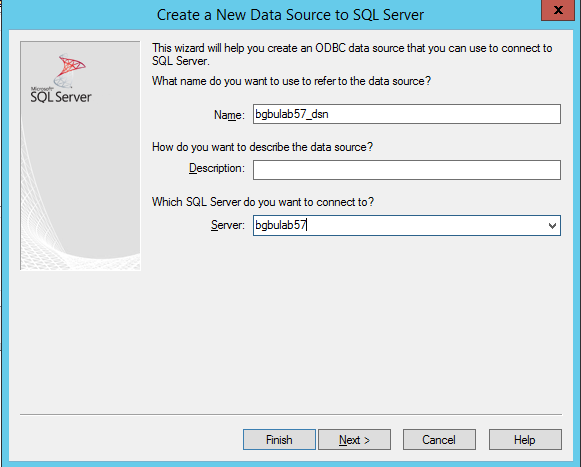
Select: ODBC Data Sources



Select the tab: System DSN - Add

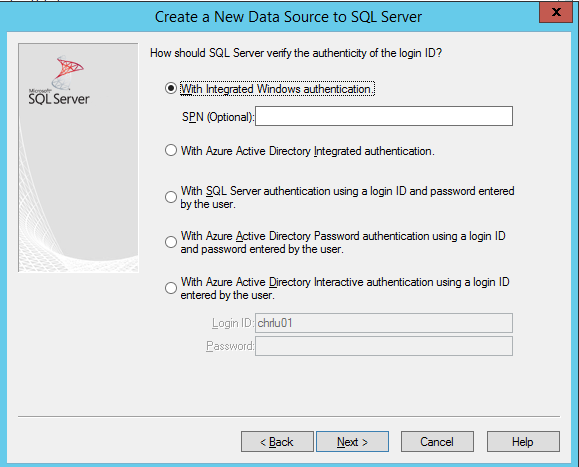


Select you ODBC driver and double click on that driver name:

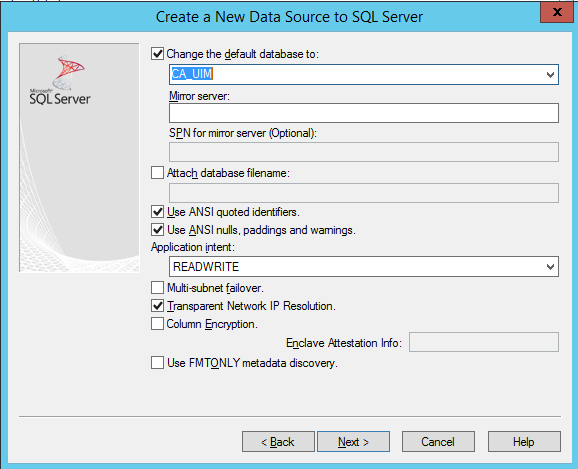


It’s the “name” that you will define, in this case: bgbulab57\_dsn, that you need to enter as value in the sql\_dsn parameter.

Select: next (and keep the defaults on that screen)

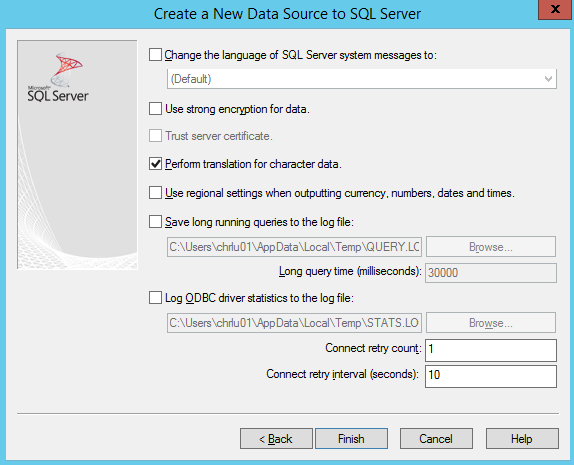


Select: next

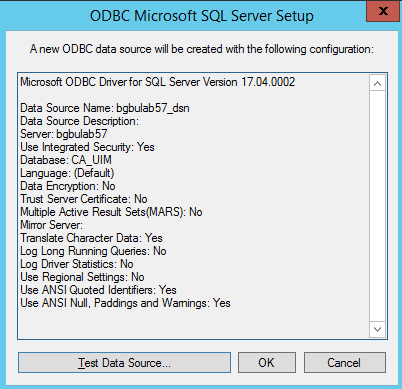


Change the default database to: **CA\_UIM**

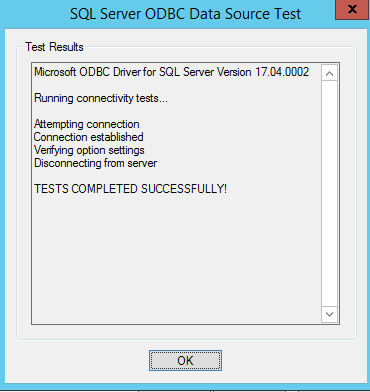
Select: next (and keep the defaults on the next screen)



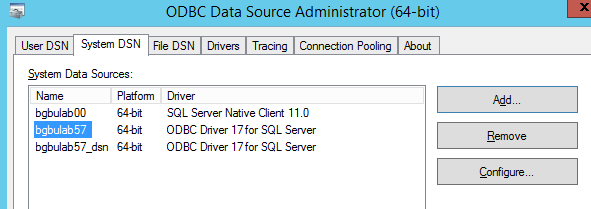
Select finish:



Select “test data source”:



Your new System DSN is now defined:



# - Create Windows Perl environment

This tool contains the Perl source and compiled Perl.

If you want to create a Perl environment that can run this Perl source & optionally compile the source yourself, you can follow the documented steps.

Once the Perl environment is created you can compile the Perl source:

pp -C -o c:\unibat\queuecheck.exe c:\unibat\queuecheck.pl

pp -C -o c:\unibat\queuecheck\_mysql.exe -l="C:\strawberry\c\bin\libmysql\_\_.dll" c:\unibat\queuecheck.pl

**Note1**: you must execute this PP command from a command prompt with as directory where you placed:

* nimsoft\_check\_package\_version.pl
* nimsoft\_generic.pm

**Note2:** the -l option is needed to include all dll modules to be able to execute the compiled module on an external server without Perl installed.

* + download from: http://strawberryperl.com/releases.html the file:
    - strawberry-perl-5.14.2.1-64bit.msi (it's a **must** that you download version 5.14.2)

**Note:** with UIM 20.40 a newer Perl SDK was released that support also Perl 5.32 (strawberry-perl-5.32.0.1-64bit.msi)

* + install the msi in: c:\Strawberry64 (as an example)
  + verify that the following directories are in the system path:
    - C:\strawberry64\perl\bin
    - C:\strawberry64\perl\site\bin
    - C:\strawberry64\c\bin
  + refresh the command prompt or reboot the activate the path
  + deploy the probe: SDK\_Perl to the main UIM server (this creates: C:\Program Files (x86)\Nimsoft\perllib)

**Note:** UIM package: SDK\_PERL version 20.40\_HF (or higher) can be used with Perl 5.32

* + copy the directories under: C:\Program Files (x86)\Nimsoft\perllib to C:\strawberry64\perl\lib
  + now you are ready to install additional packages via cpan (from command prompt): (you need internet access because these modules are get directly from cpan):
    - * cpan install Crypt::RC4
      * cpan install XML::Simple
      * cpan install HTTP::Request
      * cpan install DBD::ODBC (here you will receive some messages, but it's normal)
      * cpan install MIME::Base64
      * cpan install Time::Piece
      * cpan install Time::Seconds
      * cpan install LWP::UserAgent
      * (cpan install -f WWW::Mechanize) (generates messages and can take a long time to complete)
      * cpan install PAR::Packer
      * cpan install MIME::Lite
      * cpan install Data::GUID
      * cpan install Sys::HostAddr
      * (cpan install XML::LibXML)
      * cpan –fi Statistics::LineFit
      * cpan install Math::Spline
      * cpan install DBD::CSV
      * (cpan install DBD::Chart)
      * cpan install Net::SNMP
      * pip http://strawberryperl.com/package/kmx/perl-modules-patched/Crypt-OpenSSL-Random-0.04\_patched.tar.gz
      * (cpan -fi Net::SSL::ExpireDate) (generates a lot of messages and can take some time)
      * cpan install Win32::Console
      * cpan install Term::ANSIColor
      * cpan install Win32::Console::ANSI

Note: the lines between () are not needed to run the probe or utility but are packages that are also installed on the original Perl directory received from CA services. (and can be needed if you run other customized reports/tools/probes)