

# NMIS Troubleshooting

- [NMIS 9 improvements](#)
  - [Database as backend](#)
  - [Scheduler](#)
  - [Custom model folder](#)

When troubleshooting NMIS is important to know the differences between the nmis version 8 and 9. Even if they look the same, NMIS 9 has important improvements in the core that we should know when troubleshooting, specially if we are used to work with nmis8.

## NMIS 9 improvements

### Database as backend

NMIS 9 uses mongo db to save the data. But configuration files are saved in .nmis files, maintaining the same format as NMIS 8. The same as the RRD files.

So, it is important to know the difference if we want to handle this information. For example, Nodes. Nodes information used to be placed in conf/Nodes.nmis. Now, we would need to access database db.nodes to see the same data.

We also can use the [node admin tool](#), which can be used to import or export nodes.

In the support zip, the file Nodes.nmis will be generated, and can be imported using this tool. As an example:

```
/usr/local/nmis9/admin/node_admin.pl act=import_bulk nodes="/Nodes.nmis"
```

### Scheduler

NMIS 9 has a scheduler where we send the jobs to perform for the workers. We can set up priorities for the jobs, and it helps to organise all the jobs NMIS has to perform. It is important to be familiar with [the scheduler commands](#).

In NMIS 9, when we perform a job, like a collect or an update, instead of run it instantly, it will be sent to the scheduler, who will choose when the job will be run based on the jobs in the queue, the job type and the priority.

Example of a command in NMIS 8:

```
nmis8/bin/ ./nmis.pl type=collect node=MYNODE force=1 debug=1
```

The command output will be shown in the command line.

If we want to do the same in NMIS 9, we will use the nmis 9 cli tool:

```
nmis9/bin$ ./nmis-cli act=schedule job.type=collect job.node=MYNAME
```

But, in this case, we will see the output:

```
Job 5ec26b762b18131b08314c21 created for node Asgard-volla (db8dfa03-e800-476f-8f2a-2d4ba8116090) and type collect.
```

We would have to look for the logs jobs in the nmis.log file. But, we can choose to export the output to a file, and increase the job debug with job.file and job.verbosity:

```
nmis9/bin$ ./nmis-cli act=schedule job.type=collect job.node=MYNODE job.output=/tmp/mynode-collect job.verbosity=9
```

This is a resume table with some of this commands differences:

command	nmis 8	nmis 9
Tool	bin/nmis.pl	bin/nmis-cli

collect	<a href="#">./nmis.pl</a> type=collect node=MYNODE force=1	<code>./nmis-cli act=schedule job.type=collect job.node=MYNODE</code>
update	<a href="#">./nmis.pl</a> type=update node=MYNODE	<code>./nmis-cli act=schedule job.type=update job.node=MYNODE job.force=1</code>
Cleanup	<a href="#">./nmis.pl</a> type=purge	<code>./nmis-cli act=purge [simulate=t/f] [info=t/f]</code> <code>./nmis-cli act=dbcleanup [simulate=t/f] [info=t/f] [use_performance_query=t/f]</code>
Optional parameters	debug=9	<code>job.verbosity=9</code> <code>job.output=/path/file</code> Will log the output into a file

## Custom model folder

NMIS 9 has a models-custom folder to save customised models. It is really useful to work with an additional folder, and it is also really helpful for support related purposes.