

# opConfig CLI tool

opConfig has some features only available through the command line tool, located at <omk\_dir>/bin/opconfig-cli.exe.

Information on what commands are supported is printed when no options are specified. All options support debug=true for debug output and debug=9 for verbose output.

opConfig cli has to be run as root.

Listed below are the possible options for the act=X argument:

- [Discover](#)
- [Test\\_connect](#)
- [run\\_command\\_sets](#), [Running commands on devices](#)
- [get\\_command\\_output](#), [Get the last output from a command for a node](#)
- [find\\_command\\_revisions](#), [Get the revisions of a command](#)
- [diff\\_command\\_outputs](#), [Diff two revisions](#)
- [purge options](#)
- [Config sets](#)
- [Queue](#)
- [list\\_policies](#), [import\\_policy](#), [export\\_policy](#); [update\\_config\\_status](#), [export\\_config\\_status](#) and [check\\_compliance](#)
- [Credential sets](#)
- [Refresh OS Info](#)
- [Create Compliance Report](#)

## Discover

Tests a single given node (argument node=nodeX) and attempts to determine its Transport and Credential Set configuration. The node must already have a Personality set.

```
./opconfig-cli.pl act=discover node=NAME debug=1
```

## Test\_connect

opconfig-cli can be used to test connections to help debug situations that don't make any sense.

An example of how it can be used:

```
bin/opconfig-cli.pl act=test_connect host=192.168.88.254 transport=Telnet personality=ios username=testuser  
password=testpass
```

The options for transport and personality are given above.

It is also possible to test an existing connection from the connections.nmis file by specifying node=node\_name , if any options are specified on the command line along with the node they will override the settings loaded from the connections.nmis file.

command="some command" can also be specified to test the output of a specific command.

## run\_command\_sets, Running commands on devices

This command will run all applicable command sets for all nodes (by default).

Options:

- nodes=node1,node2,etc -- only command sets that apply to the specified nodes will be run.
- names=command\_set1,command\_set2,etc -- only run the specified command sets (and of course only for nodes matching the command sets' criteria)
- tags=tag1,tag2,etc -- The candidate commands are filtered by the tags specified. If one or more tags given on the command line is present in the list of a command's tags, then the command will be run.

```
./opconfig-cli.pl act=run_command_sets node=NODE_NAME debug=9 names=Check_Disk_Usage_home
```

## get\_command\_output, Get the last output from a command for a node

Requires node=node\_name command="command name" and returns output collected from the last run of this command

```
./opconfig-cli.pl act=get_command_output node=NODE_NAME command="show configuration"
```

## find\_command\_revisions, Get the revisions of a command

Requires the node name and returns a list of the commands a revisions for a node.

```
./opconfig-cli.pl act=find_command_revisions node=NODENAME
```

Expected output:

Node	Command	Revision
mynode	top -n 3 -d 5 -M -b	1
mynode	top -a -n 1 -M	1
mynode	vmstat 1 10	2
mynode	pvdisplay	1
mynode	ps -ef	1
mynode	lvdisplay	1
mynode	df	2
mynode	chkconfig	1
mynode	disk usage /home	0

## diff\_command\_outputs, Diff two revisions

Shows the diff from the output of 2 revisions of stored output (does not run them, only queries). The command line would look similar to get\_command\_output with the edition of revision\_1= and revision\_2=

```
./opconfig-cli.pl act=diff_command_outputs node=NODENAME command="disk usage /home" revision_1=2 revision_2=3
```

## purge options

By default opConfig does not remove old revisions from the database; if you run many non-change-detecting commands (or encounter frequent changes in your commands' outputs) then this will likely make the GUI unwieldy (e.g. the revision drop down menus will become very large). The revisions can be removed with the command line option:

```
./opconfig-cli.pl act=purge_revisions node=NODE_NAME debug=9
```

There is also an option to remove indices:

```
./opconfig-cli.pl act=purge_indices
```

More information about purge opConfig data can be found [here](#).

## Config sets

As mentioned above opConfig lets you organise whatever commands you'd like it to run into an arbitrary number of groups which we call command sets.

There are several cli options to manage config sets:

```
./opconfig-cli.pl act=list_configsets
./opconfig-cli.pl act=import_configset file=/tmp/configset_test1.json
./opconfig-cli.pl act=export_configset name=test1
./opconfig-cli.pl act=disable_configset name=test1
./opconfig-cli.pl act=enable_configset name=test1 revision=1
./opconfig-cli.pl act=push_configset name=test1 node=NODE_NAME
```

You can read more information of the config sets in the [user manual](#).

## Queue

The opConfig queue has jobs scheduled to be run in order. There are some cli options to check the queue list of the scheduled jobs or remove a job from the queue:

```
./opconfig-cli.pl act=list_queue with_ids=1
./opconfig-cli.pl act=remove_queued id=5e42a04b2b181345a5069093
```

## list\_policies, import\_policy, export\_policy; update\_config\_status, export\_config\_status and check\_compliance

These operations are documented on the separate page about [Compliance Management](#).

```
./opconfig-cli.pl act=export_policy name=cisco file=/tmp/policies2.txt
./opconfig-cli.pl act=import_policy name=cisco file=/tmp/policies2.txt
./opconfig-cli.pl act=list_policies
./opconfig-cli.pl act=export_config_status node=NODENAME file=/tmp/configstatus.txt
./opconfig-cli.pl act=update_config_status node=NODENAME force=1
./opconfig-cli.pl act=check_compliance name=cisco node=NODENAME debug=1
```

## Credential sets

Credential sets can also be managed from opConfig version 4.3.2.

All the operations are defined in the [following guide](#).

## Refresh OS Info

As part of Discovering OS Info, you can refresh the information of OS's for nodes in the "Node and IP information", "Node reports" and Node pages with the opConfig CLI tool:

```
# Refresh OS Info for all nodes
./opconfig-cli.pl act=refresh_os_info

# Refresh OS Info for a particular node
./opconfig-cli.pl act=refresh_os_info node=node_name

# Refresh OS Info for only newly imported/created nodes
./opconfig-cli.pl act=refresh_os_info only_new=y
```

## Create Compliance Report

Generate a compliance report, in csv format, which shows compliance to [Compliance Policies](#).

```
# Command to create compliance report

./opconfig-cli.pl act=create_compliance_report file=/tmp/all_nodes_compliance.csv
```