

How-To Create Custom Quick Actions in opConfig

Step 1. Create your Command Set

Before creating the quick action you will need to have your [Command Set](#) file created. opConfig lets you organize whatever commands you'd like it to run into an arbitrary number of groups which we call command sets. Let's create a basic command set using df to check the disc space on a Linux device for the purposes of this How-To however, these command sets can be as simple or as complicated as needed to complete the desired tasks.

Command sets are stored in the single file `conf/command_sets.nmis` in opConfig before 2.2.4, and since 2.2.4 opConfig supports this *and* individual command set files in `conf/command_sets.d/`. It is recommended that you use individual files (as that makes things easier to maintain and more robust, e.g. should one file have syntactical problems). For backwards compatibility the old single file is consulted first, then any individual ones. opConfig loads a command set when it's seen for the **first** time, and any subsequent occurrences of a clashing config set (i.e. same name) will be ignored but logged.

Navigate to the `/usr/local/omk/conf/command_sets.d` directory and create a new Command Set file. For this example I will be creating the file `df_test.nmis`.

```
cd /usr/local/omk/conf/command_sets.d
vi df_test.nmis
```

A command set definition consists of these types of information:

- meta-information like the command set's name (which can be used in CRON to determine when / how often this command set is run),
- optional control information like scheduling constraints and an expiration policy for the commands in that set,
- filters for selecting which devices/nodes the command set should apply to. These are discussed in more detail below.
- and finally the list of the actual commands to be run (which may also have certain modifiers and meta-data attached to them).

Below is an example of the Command Set file created for this How-To. This Command Set is running a simple df for Linux devices to display the amount of available disk space. You may copy this code to follow along or edit it to suite your needs:

```
%hash = (
  'DF_Test' => {
    'os_info' => {
      'os' => '/(Linux|CentOS|Ubuntu)/'
    },
    'scheduling_info' => {
      'run_commands_on_separate_connection' => 'false',
    },
    'commands' => [
      {
        'privileged' => 'false',
        'command' => 'df',
        'exec' => 'df',
        'tags' => ['Linux', 'DF', 'operations'],
      },
    ],
  },
);
```

Defining which nodes a command set applies to

This is normally done by filtering by OS info, but not necessarily limited to OS info only:

```
#filter using both regular expression and string equality checks:
'os_info' => {
  'version' => '/12.2|12.4|15.0/',
  'os' => 'IOS'
},
# or just the specific os family/type and that's all:
'os_info' => {
  'os' => '/(Linux|CentOS|Ubuntu)/'
},
```

The examples above specify that the node in question must have an `os_info` property set, with sub-properties `os` and `version` in the first example and just `os` in the second.

Plain string values indicate string comparison, but regular expressions can be given too. A node is only considered for this particular command set if all filtering expressions match. You can filter on [any node property](#), not just properties from `os_info` (but the default command sets only use `os_info`).

Prior to version 3.1.1, `opConfig` considered command sets without filter blocks as disabled; for these versions you may want a 'wildcard' filter matching anything whatsoever, which can be achieved by adding an `os_info` filter block with a 'match-everything' regular expression for `os`, e.g. `'os' => '/.*' /`.

From `opConfig` 3.1.1 onwards a command set without filter is interpreted as to apply to all nodes without any restriction.

More information on Command Sets can be found [HERE](#)

Step 2: Create your Quick Action

Quick Actions

Quick actions are templates for new Virtual Operator jobs, we have shipped four sample jobs but you can create your own. Clicking the quick action button will take you too a new virtual operator screen and fill out the specified fields. The default four jobs are defined in the file: `/usr/local/omk/lib/json/opConfig/table_schemas/opConfig_action-elements.json`

Create your own Quick Action

You can create your own Quick Actions by copying the `/usr/local/omk/lib/json/opConfig/table_schemas/opConfig_action-elements.json` file to `/usr/local/omk/conf/table_schemas/` (you may need to create the `table_schemas` folder if it does not exist) and then editing it. Available field options are described in the table below.

```
cp /usr/local/omk/lib/json/opConfig/table_schemas/opConfig_action-elements.json /usr/local/omk/conf/
table_schemas/
cd /usr/local/omk/conf/table_schemas
vi opConfig_action-elements.json
```

Once inside the `opConfig_action-elements.json` file, create a new section for the command set we created. Below is an example of the entry I added to the file:

```
{
  "name": "Linux Disc Space Test",
  "description": "Check Disc Space for Linux Devices",
  "nodes": ["odem"],
  "command_sets": ["DF_Test"],
  "buttonLabel": "Check Disc Space",
  "buttonClass": "btn-primary"
}
```

name	string	Name which is shown at the top of the quick action element
description	string	Text shown under the quick action name, useful to describe what the action does
command_sets	array of strings	Command set keys which you wish to be run
nodes	array of strings	Names of nodes which you wish the command sets to be run against
buttonLabel	string	Text of the run button
buttonClass	string	Css class applied to the button to colour it. <ul style="list-style-type: none">• btn-default• btn-primary (default, blue)• btn-success (green)• btn-warning (orange)• btn-danger (red)

Step 3: Confirm your Quick Action is working

After adding your Quick Action entry to the `opConfig_action-elements.json` file, navigate to `opConfig` and select 'Virtual Operator'. As shown in the screenshot below the newly created Quick Action 'Linux Disc Space Test' is now displaying. Click on the button to start the Quick Action.

opConfig 3.2.4 Views - Actions - Virtual Operator - Search Modules - System - Help - User: nmis

Home Virtual Operator Virtual Operator New Virtual Operator Job Filter Period

Troubleshoot High Bandwidth Usage...

Find the top talkers in the core network

Nodes: [View](#) [Refresh](#) [Command Sets](#) [Troubleshoot_High_Bandwidth_IOS](#)

[Troubleshoot](#)

Troubleshoot Linux Issues

Look for common problems in Linux

Nodes: [View](#) [Refresh](#) [Command Sets](#) [Linux_Highly](#)

[What's Wrong](#)

Restart Opmantek

Just restart the daemon

Nodes: [View](#) [Refresh](#) [Command Sets](#) [Restart_Opmantek_Daemon](#)

[restart opmtek](#)

Linux Disc Space Test

Check Disc Space for Linux Devices

Nodes: [View](#) [Refresh](#) [Command Sets](#) [Linux_Disk_Space](#)

[Check Disc Space](#)

Troubleshoot Interface Issues on IOS

Looking for issues with interfaces on Cisco IOS

Nodes: [View](#) [Refresh](#) [Command Sets](#) [IOS_INTERFACE_ISSUES](#)

[Troubleshoot](#)

Check Systems

Check these devices now.

Nodes: [View](#) [Refresh](#) [Command Sets](#) [IOS_Highly](#) [Linux_Highly](#)

[Check](#)

Simulate Network Outage

Change the routes to simulate a broken network.

Nodes: [View](#) [Refresh](#) [Command Sets](#) [Simulate_Network_Outage](#)

[Create Kaca](#)

Network Troubleshoot

Look for common problems in Linux

Nodes: [View](#) [Refresh](#) [Command Sets](#) [Network_Troubleshoot](#)

[Verify network connectivity](#)

Why is odem slow?

Run a quick disk and cpu check

Nodes: [View](#) [Refresh](#) [Command Sets](#) [Test_Linux_CPU](#) [Test_Linux_Disk_IO](#)

[Test Thor](#)

Fix Network Outage

FIX the routes to simulate a broken network.

Nodes: [View](#) [Refresh](#) [Command Sets](#) [Fix_Network_Outage](#)

[Calm the Kaca](#)

Network Scan

Look for common problems in Linux

Nodes: [View](#) [Refresh](#) [Command Sets](#) [Network_Scan](#)

[Network Scan](#)

Restart MongoDB

Just restart the daemon

Nodes: [View](#) [Refresh](#) [Command Sets](#) [Restart_MongoDB_Daemon](#)

[restart mongod](#)

Generate Traffic

Download a file on odem to generate network traffic

Nodes: [View](#) [Refresh](#) [Command Sets](#) [Generate_Traffic](#)

[make some noise](#)

After clicking the Quick Action button the page below will auto fill the nodes provided in the opConfig_action-elements.json file as well as the Command Sets we created in the df_test.nmis file. Schedule a time for the Quick Action to run, give it a name and a description then click 'Schedule'.

opConfig 3.2.4 Views - Actions - Virtual Operator - Search Modules - System - Help - User: nmis

Home New Virtual Operator Job New Virtual Operator Job Filter Period

Select Scope

Nodes: Nodes which the job will run on

Command Sets: Command sets which will be run on the selected nodes

Tags: Tags to select command sets - optional

Time

Schedule:

Job Details

Name: Details:

[Schedule](#)

Once the job has been completed you can view the results and details of the Quick Command. To view the command output click on the command in the Commands table as indicated by the arrow in the screenshot below.

opConfig 3.2.4 Views - Actions - Virtual Operator - Search Modules - System - Help - User: nmis

Home Schedules Virtual Operator Schedule Virtual Operator Schedule Filter Period

Job Details

Name: DF_Test - odem

Status: Finished

User: nmis

Scheduled: 2020-04-27T19:32:43

Started: 2020-04-27T19:32:43

Ended: 2020-04-27T19:32:46

Duration: 3 Seconds

Commands

Derived	Condition	Nodes	Command	revision	Detected At	Command Set
		odem	df	4181	2020-04-27T19:32:43	DF_Test

Showing 1 to 1 of 1 entries

Operations

Date	Node	Status	Event Type	Operations	Details
2020-04-27T19:32:46	odem	OK	Command Execution Completed	df	Command execution completed successfully

Showing 1 to 1 of 1 entries

The page below displays the command output as well as the Command Summary and the Most Recent Revisions. That's it, that's all there is to it! Your Quick Action is working!

opConfig 3.2.4

ViewsActionsVirtual OperatorSearch

ModulesSystemHelpUser: nmls

HomeJob ResultCommand Output

Command Output

Compare RevisionsCompare Command OutputsRaw OutputRun Command Now

Filter80

Filter Command Outputs

Nodeodem

Commanddf

Revision4181

Filter

Diff Detect Changes

Disable

Command Summary

JobDF_Test - odem

Revision4181

Nodeodem

Host192.168.88.15

Commanddf

Command SetDF_Test

Created at2020-04-27T19:32:43

Updated at2020-04-27T19:32:43

Last Attempt at2020-04-27T19:32:43

odem OS Summary

OSLinux

Version2.6.32-754.17.1.el6.x86_64

Major2.6

Image

Most Recent Revisions

Date/Time	Revision
2020-04-27T19:32:43	4181
2020-04-27T19:02:46	4180
2020-04-27T19:00:10	4179
2020-04-27T18:19:43	4178
2020-04-27T18:02:45	4177

Command Output

filesystem

36-blocks

Used Available Use% Mounted on

/dev/mapper/vg_msi64-lv_root

16382888

5287652

32%

/

/tmp

14778836

0

0%

/dev/shm

/dev/sda1

487652

183468

37%

/boot

/dev/mapper/vg_msi64_data-lv_data

41149748

15334816

33%

/data

/dev/mapper/vg_msi64-lv_var

28027268

7867668

28%

/var