



# NSO Scheduler

## NSO Feature Demo

Cisco Tail-f  
Stockholm, Sweden  
2017-11-06

NSO 4.5.2

# NSO Scheduler

## Context/background of why this feature/enhancement has been developed

In NSO today there are several places where operational data in lists may grow indefinitely if not maintained. One such example is the /devices/commit-queue/completed/queue-item list. To ease maintenance of pruning such data and other administration-though tasks in the NSO system we've added this feature.

## High level description of feature/enhancement

NSO now includes a native time-based job scheduler suitable for scheduling background work. Tasks can be scheduled to run at particular times or periodically at fixed times, dates, or intervals. It can typically be used to automate system maintenance or administration-though tasks.

## The feature/enhancement resolves the following problem(s)

The NSO scheduler can typically be used to automate system maintenance or administration-though tasks.

# Scheduling Periodic Work

A standard Vixie Cron expression is used to represent the periodicity in which the task should run. When the task is triggered, the configured action is invoked on the configured action node instance. The action is run as the user that configured the task. To schedule a task to run sync-from on 2 AM on the 1st every month we do:

```
admin(config)# scheduler task sync schedule "0 2 1 * *" action-name sync-from action-node /devices
```

If the action node instance is given as an XPath 1.0 expression, the expression is evaluated with the root as the context node, and the expression must return a node set. The action is then invoked on each node in this node set.

Optionally action parameters can be configured in XML format to be passed to the action during invocation.

```
admin(config-task-sync)# action-params "<device>ce0</device><device>ce1</device>"  
admin(config)# commit
```

Once the task has been configured you could view the next run times of the task:

```
admin(config)# scheduler task sync get-next-run-times display 3  
next-run-time [ "2017-11-01 02:00:00" "2017-12-01 02:00:00" "2018-01-01 02:00:00" ]
```

You could also see if the task is running or not:

```
admin# show scheduler task sync is-running  
is-running false
```

# Schedule Expressions

A standard Vixie Cron expression is a string comprising five fields separated by white space that represents a set of times. The following rules can be used to create an expression.

| Field        | Allowed values  | Allowed special characters |
|--------------|-----------------|----------------------------|
| Minutes      | 0-59            | *, - /                     |
| Hours        | 0-23            | *, - /                     |
| Day of month | 1-31            | *, - /                     |
| Month        | 1-12 or JAN-DEC | *, - /                     |
| Day of week  | 0-6 or SUN-SAT  | *, - /                     |

The following list describes the legal special characters and how you can use them in a Cron expression.

- **Star ( \* ).** Selects all values within a field. For example, \* in the minute field selects every minute.
- **Comma ( , ).** Commas are used to specify additional values. For example, using **MON,WED,FRI** in the day of week field.
- **Hyphen ( - ).** Hyphens define ranges. For example **1-5** in the day of week field indicates every day between Monday and Friday, inclusive.
- **Forward slash ( / ).** Slashes can be combined with ranges to specify increments. For example, **\*/5** in the minutes field indicates every 5 minutes.

# Scheduling Non-recurring Work

The scheduler can also be used to configure non-recurring tasks that will run at a particular time.

```
admin(config)# scheduler task my-compliance-report time 2017-11-01T02:00:00 action-name check-compliance action-node /reports
```

A non-recurring task will by default be removed when it has finished executing. It will be up to the action to raise an alarm if an error would occur. The task can also be kept in the task list by setting the **keep** leaf.

# Scheduling in a HA cluster

In a HA cluster a scheduled task will by default be run on the master HA node. By configuring the **ha-mode** leaf a task can be scheduled to run on nodes with a particular HA mode, for example a read-only action on the slave nodes. More specifically a task can be configured with the **ha-node-id** to only run on a certain node. These settings will not have any effect on a standalone node.

```
admin(config)# scheduler task my-compliance-report schedule "0 2 1 * *" ha-mode slave ha-node-id slave-node1 action-name check-compliance action-node /reports
```

Non-recurring work can only be scheduled to run on the master HA node.

# Troubleshooting

## History log

In order to find out whether a scheduled task has run successfully or not, the easiest way is to view the history log of the scheduler. It will display the last five runs of the scheduled task.

```
admin# show scheduler task sync history | notab
history 2017-11-01T02:00:00.55003+00:00
duration "0.15 sec"
succeeded true
history 2017-12-01T02:00:00.549939+00:00
duration "0.09 sec"
succeeded true
history 2017-01-01T02:00:00.550128+00:00
duration "0.01 sec"
succeeded false
info    "Resource device ce0 doesn't exist"
```

## Xpath/Devel log

- Detailed information from the XPath evaluator can be enabled and made available in the xpath log.
- Error information is written to the development log.

# Troubleshooting

## Suspending the scheduler

While investigating a failure with a scheduled task or performing maintenance on the system, like upgrading, it might be useful to suspend the scheduler temporarily.

```
admin# scheduler suspend
```

When ready the scheduler can be resumed.

```
admin# scheduler resume
```

# Default tasks

For new installations a default task is configured to run the 'prune-completed-queue-items' action under /devices/commit-queue/completed each day pruning all completed queue items that are older than a day.

