The NCCMD Daemon

The yce_nccmd daemon is responsible for handling both the NCCM and Compliance. It works in cycles of 5 minutes, spawning child processes who handle the bulk of its work and finishing when done. It supports multiple servers, so you can have multiple daemons running on different servers and they will not interfere (though only one master daemon can run per server).

This article gives an overview of the actions that the daemon performs, per cycle

Startup

The daemon will be automatically started by the skulker. You can also manually start it if it isn't already running with the following command:

```
/opt/yce/bin/yce_nccmd.pl
```

A -v flag can be given to make it verbose and log its output to the screen. A -d flag will daemonize this process.

Parameters

The daemon uses a number of variables that can be set in a variety of ways:

- **Poll interval**: Determines the number of seconds between each daemon cycle. Set in the NCCM Lookup under the *Nccm_poll_interval* variable. Can be set per server.
- **NCCM Max children**: Determines the maximum number of nccm child processes that can be started by the daemon. Set in the NCCM Lookup under the *Nccm_max_children* variable. Can be set per server.
- **Compliance Max Children**: Determines the maximum number of compliance child processes that can be started by the daemon. Set in the NCCM Lookup under the *Cmpl_max_children* variable. Can be set per server.
- **Node limit**: Determines the maximum number of nodes to be grabbed for each cycle. Is currently hard coded to 1200.

Cleanup

The first thing the daemon does is clean up any potential leftover data from previous cycles that might have crashed, and killing any stray children that haven't been killed yet.

Check nccm node groups

Evaluating node groups can be a time-intensive process that can be quite a burden on the GUI, so they are evaluated here. The daemon evaluates all polling groups, and creates nccm schedules for any newly created or added nodes. It also removes any schedules for nodes that are removed from polling groups and are deleted.

NOTE that if a node appears in multiple polling groups, its minimum values are selected for its schedule time and max retries.

Check compliance policies

When a policy changes, its nodes need to be re-evaluated to see if they still are compliant. This process determines which policies have changed, and it schedules them all for a compliance check.

Check command rules

When a command rule is changed, it needs to be re-run to be able to provide the most recent data to check for compliance. On a command rule change a flag is set, the daemon then looks at this flag, and runs the relevant commands on the nodes linked to it, and stores the results in the Cmpl_cmd_reply table.

Check compliance policy schedules

When a policy schedule updates, a flag is set. The daemon then looks at this flag, and updates the schedule times for all policy schedules that have been changed.

Note that if a schedule for a node is set, but the node has already been scheduled for an earlier time (for example, through the API or resulting a config change), the earliest of those two times is taken, so no overwriting takes place.

Get NCCM nodes

The daemon then gathers all nodes to be scheduled for NCCM. This is based on their Schedule_time. Any node that has a schedule of either somewhere in the past, or up to 5 minutes into the future is selected, capped at the node limit. The earliest schedules go first.

Run NCCM on the nodes

These nodes are then split over a number of children, who proceed to poll them for NCCM. After this nccm poll, regardless of whether the nccm poll was successful, any command rules that are linked to this node through its node groups, linked to policies, are run and stored.

Depending on whether they have the option set to run compliance upon a config change, the child then checks if any changes have been made to this node. If yes, then it schedules this node for compliance.

If the daemon cannot poll every single node within the 5-minute time limit, any nodes it did not get to are released, back into the queue and can be picked up by the next cycle, or a different daemon.

Check compliance license

At this point, the nodes scheduled for compliance are checked if the license allows for them. If not, it will revoke the license from nodes until the numbers do match, starting with the ones with the schedule time furthest into the future.

Get compliance nodes

The daemon then gathers all nodes to be checked for compliance. For that, it looks at their Schedule time (in the Cmpl_node table) and grabs all that are either in the past, 5 minutes into the future, sipping those that have their schedule time set to zero (those nodes will not be picked up). Again, capped at the node limit, prioritising the nodes with the earliest schedule time.

Check nodes for compliance

These nodes are spread out over a number of child processes, in the same way as with the NCCM. Their configs are taken from the NCCM and checked against all policies that are linked to them, through the node groups assigned to these policies.

All rules that match the node's vendor type are evaluated and result in either compliant or not compliant. If any of the rules fail, this means that its policy is not compliant on this node. The priority of the policy's rules that is the highest amongst the ones that failed is returned.

The daemon then checks if the policy in question has a scheduled policy linked to it. If there is, and it is active, then the next schedule time is calculated and set. Otherwise the next schedule time is set to zero. Note that scheduled policies are only available at Compliance Phase 2.

Any nodes that aren't checked within the time limit of 5 minutes are released again, and can be picked up during the next cycle, or by a different daemon.

From:
https://wiki.netyce.com/ - **Technical documentation**

Permanent link:
https://wiki.netyce.com/doku.php?id=guides:reference:compliance:nccmd_daemon

Last update: **2024/07/03 12:31**