NetYCE 7.2.0 Build_20200817

Release notes

Date: 2020-08-17

Featured

7.2.0 Release overview

The **7.2.0 release** has three main topics that are new or fully renewed. The main item of this release is the '**Compliance**' function. Compliance here refers to determining if a device configuration conforms to the policies defined for it. After every configuration change of a device, the compliance policies are executed and the appropriate reports and/or signalling is generated. Compliance policies are completely customizable and are unlimited in number. Each Policy consists of a number of rules which in turn use conditions.

Compliance relies on having access to the latest device configuration. This task befalls to the reworked NetYCE '**NCCM**' (Network Change and Configuration Management) that is now equipped to respond to syslog and Snmp-trap messages from devices to collect their configurations on demand.

Designed to be a high-performance and high-capacity Compliance function, it is released in two phases. The first phase was already very complete in all its capabilities on configuration validations. The three-tier definition levels of Policies, Rules and Conditions allow for multi-vendor policies that can cope with extensive condition-logic to deal with any design or specific constraints.

The second phase of the Compliance release is included in this version and entails additional rule types, group compliance and its own scheduler.

Another new feature worth mentioning are the '**System events**', the self-monitoring system that informs front-end users and NMS environment of changes in the NetYCE database availability and replication status. Additional system monitors will be added with the next releases.

Furthermore, a wide variety of enhancements to scenario functions and service-types were realized. More information is included below.

NCCM

The new Compliance is build to work alongside the rebuild NetYCE NCCM. NCCM is basically

providing a configuration backup function that is optimized to maintain a full history of all configuration changes. As before, an ultrafast reporting browser allows to find configuration changes between any two points in time.

Where the previous NCCM releases could either periodically poll all configurations or retrieve them as part of (scheduled) NetYCE jobs, the new NCCM engine integrates with a syslog and Snmp-trap receiver to get informed of a configuration change by the network devices directly. But it is also possible to feed the NCCM from an existing set of syslog receivers using forwarding. The NCCM events reader is fully configurable and will perform tasks like de-duplication and reverse DNS-resolving.

NCCM is available for an unlimited number of devices requiring no licenses as the NCCM works for all nodes defined in the CMDB. All that is required is that these nodes are assigned a supported NetYCE vendor-type and a 'domain' that defines the login credentials. Any modelled 'YCE'-nodes (created using service-types) are naturally also supported by the NCCM.

To set-up the periodic configuration retrieval for NCCM, the user creates 'Polling groups'. A polling group consists of either a static list of named nodes or a set of 'Node-groups'. Node-groups can use multiple rules using criteria to include or exclude nodes based on their attributes. These groups are dynamic in nature as they automatically find their members. Larger networks will likely benefit from dynamic groups as they will simplify maintenance for new or obsoleted devices.

Compliance

Compliance Phase-1

The Compliance front-end defaults with a dashboard view that graphically shows what percentage of the compliance-tested configurations are valid or not. As each compliance Rule is assigned a severity and the highest failing Rule severity determines the non-compliance rating, the graph uses different colours for each severity. A single glance at the dashboard thus reveals the volume, which policies, and the seriousness of any non-compliance issues. Severities can be fully customized but by default four severity levels are used.

For more details on the executed policies and their results the compliance reporting is used. Reports can be defined ad-hoc or saved for repeated usage. Results are available in the font-end for review and download, or can be executed and retrieved using API calls.

Each policy can be configured to send notifications on status changes using Syslog, Snmp-traps, Rest-API, emails or a combination of these. All notification types are configurable to suit the customer environment.

To facilitate using a test or acceptance environment to create new policies, they can be individually exported and imported into other NetYCE systems. Also HPNA policies can be directly imported.

As with NetYCE modelled nodes Compliance requires a license for each node. By default a license for one node is always active. These licences can be assigned individually to any YCE or CMDB node with NCCM configuration entries.

Compliance Phase-2

This release now also includes the Compliance phase-2 deliverables. Foremost amongst these is the ability to compare configuration sections between different nodes and determine compliance on found differences. Especially in situations where nodes operate in tandem or small groups with redundancy or failover capabilities. Being able to determine automatically if firewall rules or ACL's are identical within a group is a major requirement for many network owners.

And, as network changes within a group of nodes will never be done simultaneously, the regular policy-check trigger of a changed configuration no longer suffices. For as one node in a group receives a change, it will immediately follow that it is out-of-sync with the others. The other nodes may catch up over the day, but it is at the end of the day that the policy of group compliance should be checked. This is the reason policies can now be scheduled to execute periodically using various types of intervals. Of course, scheduled policies can be setup for any policy, to function as a safeguard or when no syslog notifications are possible.

In addition to the configuration rules, policies now also support 'command rules'. These rules do not check configuration blocks but the responses to CLI commands. This rule-type allows policies to use state information in their compliance requirements.

Rules have multiple ways of selecting what part of the config they will validate for compliance. They either take the whole config, they can split up a config into logical blocks, determined by indentation, or they can select based on a starting and ending line.

Conditions are linked together through logic, consisting of hierarchical 'if', 'and' and 'or' logic. In conditions, there is support for regular expressions.

A special feature for F5 nodes is that we parse the config to search for any orphans: config blocks and sub-trees that are defined, but not used anywhere. We generate a report to the operator with these orphans, as a guide to help the operator clean them up.

Policies, along with its rules and conditions, can be imported from HPNA. HPNA Configuration and Software rules can be translated one-on-one. HPNA Diagnostics rules conversion is not supported. NetYCE policies can be exported and imported (single or in bulk) among NetYCE envionments.

Then there is the option to include NetYCE database variables of a device in the compliance rules. All condition lines have access to the node's context which can be used much like template parameters. This allows for conditions that check against values like the '<loopback>' ip-address of a node or its '<hostname>'. The variable support includes the relation syntax.

Finally we incorporated some feedback from users which resulted in bulk-import and -export of policies, form layout simplifications and a manual variant on the configuration block selection. We also managed to squeeze some bugs out.

Distributed NCCM / Compliance

Both NCCM and Compliance are designed to operate in a multi-server environment. Similar in setup for the distributed scheduling of command jobs, the NCCM and Compliance tasks can be directed to specific NetYCE servers for groups of nodes based on their attributes. The setup also allows for creating dedicated NCCM and Compliance servers if so desired.

System Events

NetYCE has been using redundant databases from the start to allow for hot-standby and failover database access. This allowed for transparent user-access during maintenance or partial outages. Since severe outages could cause replication conflicts requiring manual maintenance, it was felt that an external signalling of these situations was desired.

The new 'System events' function is essentially a self-monitoring system that can inform local and remote users on system status changes. The System events maintains a status for each monitoring function. Currently three monitors are active:

- Database-connection. Is a database connection available and if so, is it the primary or secondary server?
- Database-replication. When using two database servers are they in-sync or not?
- Debugging mode. A server can be set in debugging mode to gather extended logging or trace in detail. As it will impact performance, being notified of this status is relevant.

Notification of system-events to the local users is incorporated in the GUI. A separate header bar with a descriptive message will be displayed for each monitor. A colour coding is used to indicate its severity. A user can close each notification bar individually for the duration of the session (or event re-occurrence, or browser page-reload). The customization allows to change which user-roles are notified before or after login.

To signal external users or systems, notifications can be sent using Syslog, Snmp-traps, Rest-API or email. Its setup is very similar to the compliance signalling and can be fully customized.

System events can be manually cleared using the 'Admin - System status' tool. Please be aware that the monitors and the GUI only intermittently test the status. Notifications to set or clear a status can exhibit a delay of two to ten minutes.

Additional system monitors will be added in subsequent releases. Expect notifications for licensing issues, disk space availability and others.

Scenario functions

To extend the functionality of scenarios several new functions were added.

- **resched_job**. Schedule the running job to execute again. The resched_job command allows for recurring jobs. It is included in a scenario to reschedule the same job for a later day and/or time. Since the rescheduled job will execute the same scenario, this job will again be rescheduled for a later moment. In effect, the task will become a repeated job.
- **shortest_path**. Locate the devices that make up the shortest path between end points. Returns a variable list of all nodes making up the path including the start and end points. The shortest-path uses the topology (links between ports) of the nodes in the YCE database. Tracing a path is therefore limited to YCE nodes and the accuracy of the topology. For finding the shortest path the Dijkstra greedy algorithm is used. The model uses any kind of topology between ports of different nodes as a valid path and takes into account the modelling of the hierarchy in that uplink directions have a cost of '1', downlink directions a

cost of '3' and interconnections a cost of '2'. Thus resulting in a path where uplinks are preferred and core routes are not circumvented.

- **split**. Takes list(s) and a regex or wildcard separator to split the element strings into more elements; returns a list with the separated substring of each element.
- **grep**. Takes one or more lists and a condition, and runs that condition against a wildcard or a regex over the list. Returns all matching elements as a list variable.
- **match**. Takes one or more lists and a condition, and runs that condition against a wildcard or a regex over the list. Returns a list with the matching substring(s) of each element.
- **replace**. Replace searches for a specific string-value and replaces it with another string-value. The replace value can be empty. The match string is non case-sensitive.
- **calc**. Executes a numeric calculation

Enhancement

Table definitions

A new command line tool is now included in the distribution that can assist in database troubleshooting should it seem incompatible with the NetYCE code. This tool, **`ck_tabledef.pl`**, will verify that each database table is defined as the latest distribution expects it to be. Should update patches have failed or skipped and the incremental updates seem 'broken' this tool will report it. The tool can also be used to make the required alterations to ensure the table definitions are up-to-date.

Database versions

Over the years several database versions of MySQL/MariaDB have come into production. When restoring a database form one version to another incompatibilities could be experienced requiring a manual conversion. Unfortunately, when a NetYCE database was restored on a database engine of an older revision than the database, these conversions failed.

The restore tools were modified in two ways to deal with these version issues. First, attempting to restore a newer version database on an older database engine will be detected before the restore and will be denied. Secondly, when detecting that an older database is restored on a newer database engine will automatically perform the conversion needed.

Port update from Template

From the 'Ports' overview form of a node, two options exists to update the ports: 'Update from Node' and 'Update from Template'. These buttons are used to change the ports of the node using

the port definitions on another node or the node template respectively. The update has many selectable options on what items are changed or added.

Users found that one option was missing from the selection: the option NOT to remove the ports that were already defined but not present in the new layout. This option has been added to both update forms.

Port PoE attribute

All port definitions now have an additional attribute that sets the Power-over-Ethernet (PoE) profile. In the font-end, the form uses a drop-down menu from which to select the PoE profile name. These names can be defined in the Lookup using the class 'PoE_profiles'.

These profile names can be translated into configurations using templates in the regular fashion. When it is desired to parameterize the profiles, a set of Lookup variables can be created using the PoE profile-name as class. This setup allows for a fully customizable PoE profile design.

Dependency checks on delete

Template-, Relation- and Scenario-names can be used as references in many different sections of NetYCE. Template names are used in other Templates, in Scenarios or in Stored Jobs. Likewise can Relation and Scenario names be used in these contexts.

Potentially, the deletion of an assumed safe-to-delete Template can cause some configurations to fail or relations to produce expected results, effectively 'breaking' a work flow.

To prevent these unwelcome situations, any delete of Template, Scenario or Relation will now verify if any reference to its name is still in place. If it is, a popup will report on all references found and will deny the delete. This report includes details on where the reference is found including line number and the actual line of text where it is found.

Service-type Mpls_vrf

o new Service-type to add Mpls_vrf while finding next free Vrf_id o new service-type 'Add - Vrf - Mpls_vrf' o new service-type st_add_mplsvrf_new() for 'ADD - MPLS_VRF - VRF_ID_NEW'

Service-type custom subnet

o change service-type add_custom_subnet() to support custom subnet-plan vlan-id & vlan-tpl

Dynamic Node-groups caching

o improved the performance of node-groups by maintaining a cache of the calculated groups after a node update

Scenario 'Send-email' custom reports

o scenario function 'send_email' now also can send (scheduled) custom reports

Service-type duplicate rows

o A new button has been added to the service type form that allows you to duplicate service type records

Vendor-type MROTEK

o created MROTEK vendor module

Http proxy servers

o added support for configurable http proxies for the session-cookies

Scenario 'Csv_report' option

The scenario functions 'csv_report' and 'csv_file' support the new '-x' option that allows the generated report to be appended to the report-file instead of overwriting should it already exist.

Syntax highlighting

Syntax highlighting for Scenarios and Templates have been improved.

API call 'nccm_submit'

The configurations are normally retrieved from the nodes (jobs, nccm poll). But sometimes it could be desired to upload a configuration directly into the NCCM. For example when a node configuration cannot be retrieved directly and a NCCM report or Compliance check is required anyway.

The nccm_submit API call allows you to create an NCCM entry for a node as the 'latest' configuration. To submit a configuration the node must exist as either a CMDB node or as an YCE node.

Command parse table anchors

Command parsing tables now also includes the use of anchors for multi-word matches. For example <brackets_match:)>

Template, Scenario, Relation delete confirmation

As an extension on the “Dependency checks on delete” enhancement, the confirmation popup behaviour was modified. When deleting Templates, Template revisions, Scenarios, and Relations now first check whether there are jobs, scenarios or templates that reference them. If there are, you won't be able to delete unless you have manager or system permissions.

The dependency report was extended for the Main- and Port-templates with the node names that still use them. To prevent useless long reports, the list of dependencies for each type is truncated at 10 entries at which point a count of additional dependencies is given.

The delete-override that is default for managers and system users can be modified by updating the Auth_permission table as described on the NetYCE Wiki.

Config parsing HP C7 & Cisco IOS vlan ranges

Vlan ranges in HP C7 and Cisco IOS configs (for example 22 25-30 32) are now expanded for config parsing, so you can parse all of them (therefore including 26,27,28,29), similar to Huawei and Ciena for which this function already existed.

New SiteCode form renewal

The look and feel of the New SiteCode form has been changed to be more consistent with the rest of the application. Functionally nothing changes.

Hostname Change form renewal

The look and feel of the Hostname change form has been changed to be more consistent with the rest of the application. Functionally nothing changes.

IP Servers

Support for Ip-Servers have been simplified and the controlling menu items removed. The 'Servers' are now part of the Client details form using a dedicated tab.

Where already configured, they can still be used as before using relations, but their creation no longer depend on “IP-plans” and “IP-server-plans”. Instead, in the Client 'Server' tab you can administrate servers and their attributes for the client and/or site in question.

Please see the wiki page Client - Servers for details.

Relation test tool

The “Design - Relation test” tool has been reworked to support CMDB as well as YCE-modeled

nodes. It now uses 'Node-groups' similar to many of the tools. But the tool now also supports direct entry of nodenames, sites or clients, including wildcard characters (* and ?).

Additionally, the relation test now allows you to provide overriding values for any <variable> used in the relation query. Especially relations used in port-templates or used in scenarios will benefit from this testing option. For more details see the Relation test page.

Change

Patch management

Updating NetYCE systems involve software code updates and changes to the databases. The latter uses patches that perform specific modifications to the database definitions or its data. To allow for upgrades that can jump any number of versions or build numbers, these patches are created to be incremental.

As skipping or failing to install a patch properly will cause the upgrade path to potentially be 'broken', it is essential that all patches are executed properly. After encountering several instances where this has happened, we modified the patch management to be more strict and also more persistent in ensuring the proper patches are incorporated.

When performing updates tests are executed to verify all patches, including those of older versions, are present. If any older patch failed or was skipped, it will be attempted again.

In addition to this change in the patch management a database definition verification tool is now included in the distribution.

Hot-deploy options

After completing an update or making changes to the setup, most of the NetYCE daemon processes running in the background are restarted. As these include the http-server (apache) and the back-end-servers (mojo and xch) that serve the GUI, the user that initiated the operation using the GUI would experience that the process never seemed to finish. Although the process did complete normally it was the restarting of these GUI servers that prevented the information reaching the user.

To prevent this situation, the method of restarting these three daemons was changed to allow the transactions to finish before a process would end and get relaunched. These so called 'hot-deploy' restarts avoid downtime and interrupted transactions.

Resized columns

o wider columns for Node name, Service-type, Service-name, Relation name, Port_descriptions, SiteVP (now 50 or 100 char)

Mpls_vrf Licenses

o added mpls-vrf licences plus enforcing

Sub-template indentation

When including sub-templates in templates, the lines generated by these sub-templates are substituted directly where the sub-template is called. The resulting configuration will include the lines, but literally. Any indentation (using leading spaces) would only result in the first line being offset by the spaces before the template name but not any other.

To allow the use of indented sub-templates, the code generator was modified to detect if a sub-template was included using indentation. If only spaces are used before the sub-template name, these spaces are used as a prefix for all lines the sub-template generates. When other characters are used before the sub-template, the substitution will be done in-line without prefixes.

The Command-job and View-config tools were modified to show the corresponding indentation levels.

Nodename type-ahead

Several forms allow to type a hostname that assists the user by listing matching nodenames based on what has been entered so far. Under some conditions the results were not updated or failed to include the desired name.

The behaviour of these fields has been modified to become more intuitive and reliable.

Fix

Scenario 'send_email' function

The scenario function to send emails, `send_email`, was recently extended to support attachments. The change also included a syntactic validation of the email addresses.

It now became apparent that this email validation was too strict resulting in an empty list of recipients with the error message “No email-address to send mail to”. This situation has been corrected.

Scenario 'st_exec' error handling

Service-types can be executed using the XCH API, the CSV API, Scenarios (jobs) and using the front-end. Especially when an error occurred using the API's or the Scenario was the error handling insufficient to abort the remainder of the actions. This situation has been corrected.

Service-types IPv6 names

When editing a Service-type where an IPv6 subnet is to be created, the help menu to choose from the available IPv6 subnets in the supernets would show the subnets of the IPv4 supernet plans, not of the IPv6 supernets. This has been corrected.

Service-type IPv6 errors

A running Service-type could crash (abort) when an IPv6 subnet is to be created. A missing or incorrect IPv6 value was often the cause. By adding validations and corresponding error-messages, the user is informed of the failed variable and the Service-type is halted gracefully.

Database auto-repair

To improve performance and resilience several LOGS and NCCM tables are maintained as weekly database tables which are combined into 'merger' tables to report on the full retention period. This mechanism was introduced tome time ago and proved very successful. However, this method also proved to be very sensitive to changes in the Mysql/MariaDB version.

To further enhance the resilience the various 'merge' tables are tested after the daily maintenance which will then trigger an auto-repair should it prove necessary. This repair entails a (week-)table by table export, table-creation and import to ensure compatibility with the running database engine.

CSV-report format

Downloading a CSV formatted report always showed the last field of the last record to be quoted. This was corrected.

Replication error

In a multi-server NetYCE environment two databases are used that provide failover and hot-standby redundancy using a master-master replication schema. In this setup, a database replication issue was traced back to a specific SQL statement that caused the replication mode to switch from statement-based to record-based. And although correctly processed, this mode caused a record to be duplicated where it should not.

The main consequence was that the Job-id's or Task-id's generated were no longer unique. The

individual logs were lost until the duplicate record was manually deleted.

The responsible SQL statements were re-developed to ensure the replication mode would remain statement-based.

Scenario 'relation'

The scenario 'relation' function can use variables from the scenario to pass on the the relation query for substitution. The variable substitution was modified to be more reliable.

IP-based server setup

A NetYCE VM (virtual machine) running on a (laptop) hypervisor like VirtualBox or VMware can be configured so its web front-end can be reached using its DNS-name or its IP-address. Due to security constraints enforced by browsers (cross-domain requests) the NetYCE configuration must be explicitly set to the one OR the other mode.

When using the less often used IP-based NetYCE server setup some (download) links found in the various tools would still appear to use the DNS-based setup. It was found that the http(s) server needed some modification to ensure it did not alter the IP-address into its dns name (referrer).

Scenario 'parse_cmd'

Several fixes were applied to make the scenario 'parse_cmd' function more reliable. Several fixes were applied to make the scenario 'parse_nccm', 'parse_run' functions more reliable. The most important is that config parsing now can produce logs in order to help track any error or issue.

OS-upgrade for CMDB-nodes

NetYCE supports two types of nodes, the YCE-nodes that are modelled and created using Service-types, and CMDB nodes that have minimal modelling. The OS-upgrade tool that previously supported both node types had stopped functioning for the CMDB nodes. This functionality has been restored.

Vendor-module 'HP_C5'

The NetYCE Vendor-module 'HP_C5' could fail (configuration) file transfers without indication to what caused it. This was resolved by adding error messages related to file-transfers.

Vendor-module all

NetYCE supports several file-transfer protocols. Only the traditional 'tftp' does not use any authentication. For the other protocols (ftp, sftp, scp) a NetYCE-only user account is used for the

authentication. To prevent this reserved user-account to be used out-of-context, the job logs will be censored for the authentication information for non-privileged users.

IP-adress change

When the IP-address of a system is changed due to DHCP or net_setup modifications, the system would not reconfigure itself properly. Even though the all configuration changes were processed and daemons properly restarted, the system would not update the database connection.

The issue was traced to a modification in the database monitoring for version 7.2.0 which has now been corrected. When executing net_setup, the IP-address change is now properly activated for all database access.

Crontab default

NetYCE uses the Unix/Linux chronograph for some periodic tasks like archiving by maintaining the 'crontab'. But the crontab is also used for executing the custom reports. A problem was located where the custom report entries (and possibly some custom entries) were removed form the crontab after a software update, halting the periodic report updates.

It was found that the `yce_setup` defaulted the crontab under most conditions. This situation has been corrected.