Table of Contents

NetYCE 7.0.5 Build_20181120

Release notes

Date: 2018-11-20

Enhancement

Log date filters

A date filter has been added to the Logs form. You can now sort both by month, and by day in order to narrow down your searches.

Task Log date filters

A date filter has been added to the Task Logs form. You can now sort both by month, and by day in order to narrow down your searches.

Custom Data Tables Persistent Search

Persistent search has also been added to the search field for the tables in the Custom Data tables list.

Relations and Scenarios Duplicate

Both the Scenario and Relation forms now have duplicate buttons. Note that NetYCE relations can also be duplicated, with the result being editable and deletable by modelers and managers.

Tibco Parse Default Actions

In .conf-files for the Connect Cramer form, you can create a list of actions to be executed whenever a value changes. In the past a value was matched by a variable name, and if there was no match the default actions would be selected. With this fix however, the default actions are always run on a change, and if there is a match with one of the variable names, its list of actions gets selected alongside it.

Topology positions update

In the past, all southern nodes in topologies were signified by the letter 'Z', of the Dutch 'Zuid'. We now changed this to the English 'S' of 'South'.

This impacts a number of different areas:

- The Ip_subnet table has been changed. Its topology attributes are now modified.
- Any templates, scenarios, relations and stored jobs mentioning these columns have been automatically modified
- Service types and node types mentioning these columns have been automatically modified
- However, any custom name, variable name or alias that contains an S or a Z has not been modified. These names are the responsibility of the engineer.

All standard files have been modified. If there is any other file that needs to be converted, you need to run the file system/z_to_s.pl, followed by the file names you want to convert. For example:

/opt/yce/system/z_to_s.pl etc/cr_fz_hfc.conf etc/fz_hfc.conf

New Topology form

The Topology form has been renewed. Like the ports forms, it now also has its own page. Functionality has remained mostly the same, and it now fully supports IPv6 addresses.

Job parameter handling

All Jobs use Scenarios to direct the generation of configuration files and their execution on the intended devices. Within the scenario variables can be used to control these configurations and the process flow of the scenario.

To this end, these variables are entered in their appropriate sections in the scenario. The section labeled [parameters] is used to define variables that are to remain constant for the job whereas the [scenario] section has variables that are more dynamic and exist only during the execution of the job.

If configurations are generated using templates that need the (external) input of scenario variables, these needed to be expressly included in the scenario command using a string of '-p “variable=value”' specifications. This behaviour has changed. Now all variables, being [parameter] or [scenario] defined, are now automatically exported to the configuration generator. The '-p' options are only useful when the name of the variable is different within the templates.

The exported variables also extend to the Relations. Any variable name in the scenario can therefore be used in the SQL-query of the relation.

The implementation allows the variables to used directly as <variable>, and this format can also be used to access the first element of any list-type variables. When a list-type variable is needed as a list, the variable can be referred to as <variable@SCN>. The @SCN is a reserved Relation name for the scenario.

Linux vendor-module

A new vendor module has been added to support Linux servers. This module is mostly intended for experimental use since all the module can do is login on a bash shell and execute non-interactive commands. However, command parsing templates are supported to extract variables.

The module needs a bash or bourne shell to function properly and requires the initial prompt not to exceed a single line.

CMDB service-types

To support Basic command jobs and OS-upgrades, Nodes need to be present in the NetYCE CMDB. A front-end tool for the CMDB is available to enter and maintain this data. However, there was no means to add nodes to the CMDB in bulk without having to create custom interfaces (like ODBC) or sql-style scripts.

Now a number of service-types have been added to 'ADD', 'LOCATE' and 'ASSIGN' CMDB nodes. This allows the creation of a few very simple service-types that can be used by the NetYCE CSV API. The CSV API is a wrapper for the regular API to execute service-types in bulk using a CSV format and has its own front-end tool.

See the Wiki article on 'service_types' for the CMDB.

VRF custom attributes

The 'VRF' object now supports custom attributes like the 'Client', 'Site', 'Region', 'Domain, and 'Node' objects. The use of these custom attributes for the Vrf is supported in all use cases (API, Service-types, front-end) but is for the short-term future not visible in the front-end.

The Node VRF form is being re-developed to conform to the style and technologies of the current generation and will include the Custom attributes tab. This will be addressed in the next build.

New Template functions

Four new functions have been added for use in templates. These functions are all intended to retrieve an ip-address (ipv4 or ipv6) for a named node without having to rely on a relation.

The functions **Node_mgmt_ipv4** and **Node_mgmt_ipv6** simply provide the ipv4 or ipv6 management address of a node. It accepts any nodename in the YCE database as an argument.

The functions **Node_port_ipv4** and **Node_port_ipv6** allow you to retrieve ip-subnet/ipv6-subnet attributes from subnets assigned to a specific port from a node. Three mandatory arguments are required: the 'nodename', the 'portname' and the ip-subnet 'column name'. The portname can either be the internal Port_name or the vendor dependent full interface name.

The optional fourth argument 'filter' can be used to select the desired subnet if more than one subnet is assigned to the port. This 'filter' option behaves similar to the relation-syntax filtering

where a 'value' by itself will be compared against all columns or a 'column=value' filter will compare only the indicated column.

Examples:

```
[Node_port_ipv4(<hostname>,Gi01/00/03,Ip_parameter)]
[Node_port_ipv4(<hostname>,GigabitEthernet1/0/3,Ip_parameter)]

[Node_port_ipv4(<hostname>, 'Gi01/00/03', 'Net_address', Vrf_id = '3')]
[Node_port_ipv4(<hostname>, 'GigabitEthernet1/0/3', 'Net_address', Vrf_id
= '3')]

[Node_mgmt_ipv6(<hostname>)]
[Node_port_ipv6(<hostname>, 'Gi01/00/03', 'Net_address', Vrf_id = '3')]
```

New node vrf form

The node vrf form has been rebuilt in the new style, and it has gotten its own page. Functionality is the same as before.

New mpls vrf form

The mpls form has been rebuilt in the new style, and it has gotten its own page. You can now also select your vrf id from a list of free ids for your client type.

New Topo check report

A specialized report has been added to the Operate - Reports menu: 'Topo check'. This allows the modeler to report on all topology links where a mismatch exists between the assigned subnets at both ends. Depending on the design such a mismatch could be an issue or not.

The output lists topos with mismatches only and is sorted by Client_type, Hostname, Link-type. Per topology link the missing subnet details are listed.

Change

Command job run-time generation

The Command jobs include a “Commands” box where template-style commands can be entered. The CLI commands it generates will then be imported in the configuration of the target node.

This behaviour has been changed slightly. Instead of generating the CLI commands at the moment of scheduling, the commands are now generated at the moment of execution. This ensures the commands include all data changes between the time of scheduling and execution that a Relation will pick up.

The implementation of this change entails that the command section of the command-job is saved as a file with the name <jobid>.cmd. The “Cmd_exec” scenario command will use this file as the “template” to generate its commands for the indicated node. This method allows to use this “template” for other nodes as well if the scenario demands it.

Job log access for Engineers

Engineers were restricted to review only their own jobs in the Job Logs. This has now been changed so that these users, like the Modelers, can review all users jobs. The Operators still are restricted to their own jobs as before.

Command jobs

The preparation and execution of command jobs was extensively modified with the 7.0.5 release. One of these changes involved generating the command job configurations at the time of execution instead of job submission time.

The scenario command '**Cmd_exec**' was modified to generate the commands from the 'template' that are the commands entered. It would use this <jobid>.cmd as template to generate the commands using the node specified in the '**-n**' option and then execute the result on the same device. The '-f' option is used to name generated command file for reference purposes.

The scenario line would then be similar to this:

```
-- example
<node> = NL-ACME-CPE

Cmd_exec -n <node> -f <node>.cmd <verbose>
-- generates NL-ACME-CPE.cmd using NL-ACME-CPE data and executes it on
NL-ACME-CPE
```

However, this approach would prevent you from generating the commands from this 'template' when the execution must be performed on a different node than the node (or context) these commands were generated for. To resolve this issue, the option '**-o**' was added to the Cmd_exec. The '-o <node>' (origin-node) specifies the node name whose context is used to generate the commands from the command-job 'template'. These commands are then executed on the node specified with '-n' as usual.

When the -o is omitted, the '-n <node>' is providing the context from which the commands are generated. The Cmd_exec will NOT generate the named file if it already exists.

```
-- examples
<node> = NL-ACME-CPE
<code_node> = NL-AMS-CR02
```

```
Cmd_exec -o <node> -n NL-AMS-CR01 <verbose>
-- generates the file NL-ACME-CPE.cmd using NL-ACME-CPE data and executes
it on NL-AMS-CR01

Cmd_exec -o <node> -n <core_node> -f <core_node>.cmd <verbose>
-- generates the file NL-AMS-CR02.cmd using NL-ACME-CPE data and executes
it on NL-AMS-CR02
```

The change requires the engineer to explicitly define the context and the target nodes instead of relying on a command file that was implicitly generated at job-submission time. This approach also allows to re-use the command 'template' for multiple nodes within the scenario. The context and target nodes can be switched as appropriate.

```
-- example
<node> = NL-ACME-CPE
<code_node1> = NL-AMS-CR01
<code_node2> = NL-AMS-CR02

Cmd_exec -o <core_node1> -n <node> -f <node>_part1.cmd <verbose>
-- generates NL-ACME-CPE_part1.cmd using NL-AMS-CR01 data and executes it
on NL-ACME-CPE

Cmd_exec -o <code_node2> -n <node> -f <node>_part2.cmd <verbose>
-- generates NL-ACME-CPE_part2.cmd using NL-AMS-CR02 data and executes it
on NL-ACME-CPE
```

Node port display order

When viewing the ports of a node, the ports are virualized per slot in groups of ports per port-type (e.g. Fast_ethernet, Gigabit-ethernet, Loopback, Port_channel). The order of these port-types was often inconvenient since it was more or less random.

This behaviour is now changed and lists the port-types in a fixed order: Ethernet, Fast_ethernet, Gigabit_ethernet, TenGigabit_ethernet, etc. The order is controlled using the internal Port_class table where also the port-type name is mapped to the port-class. If required, the order can be modified to customer taste.

Node port display truncated

On nodes where the port-ids of a port-type exceed 150 will the clickable ports now be truncated. This is most often the case for Loopback or Port_channel type ports, but the behaviour is general. It will prevent extensive scrolling past unused port-id's and improves the rendering speed of the ports by the browser.

When the list of ports for a port-type is truncated, it is indicated by a [⇒] symbol indicating the list continues but is not shown. The full list will be available using the grid at the top of the form.

Fix

Custom Data Search Fix

The search for Custom Data has been upgraded. Before, you had to type very slowly in order to keep searching, or otherwise it would eat your input. This has been fixed now.

IPv6 Subnet Range Create Bug Fixed

The new fix in the IP Plan forms for IPv4 and IPv6 ranges and masks introduced a new bug that would not let you create more than one subnet range. This is now fixed.

IPv6 Subnet Range Delete Bug Fix

Previously you were unable to delete an IPv6 subnet range from an IPv6 Plan in the IPv6 plans form. This is now fixed.

Tibco Parse job names

There was a bug where in the Connect Cramer form, jobs would be scheduled with strange hash-names. This bug is now fixed.

Template Rev Authors Frontend

All template revisions now save their author as their “User_id”, while their “Full_name” is used in the forms and grids. Unless they don't have a “Full_name”, then their “User_id” is used.

Job scheduling failure

In environments where multiple NetYCE servers are used to schedule job, the scheduling could fail with program errors. This would happen if those servers could not be resolved by the submitting server using the short server name. Using different DNS domains for the various servers is the most likely scenario.

Similar errors when submitting a job will occurr when a NetYCE server was setup to use ip-adresses only because it has no DNS entry.

When submitting jobs, the NetYCE server uses API calls to itself and the other servers to connect to the scheduler and to submit the job. If the ip-resolving failed for these API calls, these errors would occurr.

These cases have now been corrected by ensuring the full-qualified server name is used to resolve, or skip the resolving if the ip-address was to be used directly.

Context functions - argument handling

Context functions are a means to create Relations that are not strictly based on a SQL statement but consist of code that executes SQL to build more complex output. Sometimes it is far more efficient to create some code to produce the desired output. Or it just takes a lot less time to write a bit of code than to develop a very hard to understand SQL statement.

The file `/opt/yce/bin/context_functions.pl` is intended for those situations and can be considered a plug-in function og the NetYCE configuration generator.

Since the functions created in this plug-in are called from the regular Relations form to assign them their relation name, they also need to receive their parameter arguments in that call.

A problem has been fixed in dealing with these parameters: whitespace between arguments or quoted arguments caused the function to work on arguments with their whitespace and quotes still attached, usually resulting in no output. The argument handling has been revised to correct this issue.

Config diff

The scenario command 'config_diff' failed to retrieve the last configuration to compare the current version against. The previous NetYCE implementation stored the last config in the 'Node_config' table along with all responses to 'show' commands.

With the implementation of NCCM the configurations were stored in tables in the separate NCCM database. The 'config_diff' command was not updated however to use this NCCM environment.

This has now been corrected.

Engineer permissions

The default permissions for 'engineer' level users were found to be inadequate to perform their intended tasks. Several front-end and API functions previously restricted to 'modeler' level users have not been entended to include engineers.

These permissions now allow engineers to create, edit and delete clients, sites and nodes using both the front-end and the API. This relates to these objects but also their sub components like custom attributes, ports, templates and vrfs.

Scenario variable resolving

With the redesigned command job and scenario handling of version 7.0.5 the variables handling was modified. The command generator now has access to all scenario variables which in turn

extended to the relations.

With this behaviour in place we believed the scenario would no longer require variable substitution from the node. This assumption proved incorrect. Although scenario facilities exist to access relations for a node, existing scenarios in production were incompatible.

As a result the original scenario behaviour has been re-implemented. At time of execution, the entire scenario is parsed for variables and relations from the <node> context and substituted. This restores the functionality of the existing scenarios.

Node_log support

The Vendor modules were for the 7.0.5 release updated to behave more consistently and allow better control over error conditions. This was partly realized by creating more common functions for all vendors to use. Most of the existing functions were ported to these new vendor modules, but not all. The function to store the results of 'show' commands proved to be amongst these, although not intentionally.

To correct this situation, the function has been re-implemented. CLI commands starting with vendor specific key words like 'show' or 'display' are detected and their results stored in the YCE Node_log table for later examination or reporting.

- the 'show'-type commands must be fully typed, abbreviations are not supported
- the 'show'-type commands are executed as any other, only afterwards is the command evaluated as a storable result and saved in the database.
- the 'show'-type commands are expected to be executed in the appropriate mode. The vendor module will not magically switch between enabled and disabled modes.

Most vendors have the commands for 'show' or 'display' and 'ping', 'dir', 'do' recognized.

Duplicate node slots

A bug where romoved nodes did not clean up their node-slots caused duplicate node-slots when these nodes were re-created. Altough the problemof not removing the slots was fixed, the existing duplicates would not be removed until a port was added or removed from the node.

A patch will be executed when updating the NetYCE system to remove these duplicate slots.