NetYCE 7.0

November 2016

> This set of release notes is 'preliminary' and may change prior to the general release of version 7.0.
>
> Since Version 7.0 and version 7.1 tie together strongly, some of its intended feature list is incorporated here. Its status though, is even more 'preliminary'.
>
> Note: *Many of the items presented in this article are described in some detail in other Wiki articles. Links to these articles will be added as soon as they are moved to the 'public' Wiki.*

NetYCE version 7.0 offers an extensive range of improvements over the current version 6.3. Many of its subcomponents have been redesigned and reimplemented to allow for the new capabilities in this and forthcoming releases.

IPv6 modelling support

IPv6 ip-plans can be defined and used the in the network designs. The IPv6 plans are used to automatically allocate subnets and addresses for use in configurations. The ipv6 supernets, and their assignments are maintained in the integrated IPAM. This extends the IPv4 capabilities to IPv6. note: The NetYCE application and the device communication will support IPv6 in the next release.

Site supernets

IPv4 and IPv6 supernets can now be assigned to specific customer locations too. Previously the supernets were assigned to customers only. Supernets can also be shared amongst many Customers and Sites.

Custom attributes

Most NetYCE building blocks now support custom variables. Clients, Sites, Services, Nodes, Domains and Regions now allow for the assignment of a ‘group’ of custom variables. Custom variables can be created dynamically and without operational impact.

Improved front-end

The web-based front-end is being overhauled for a faster and more consistent user-experience. The form-based approach and menu structure is maintained so additional training is hardly required. This overhaul is ongoing so expect some forms to change with every update.

Multi-vendor templates

Template names can now be shared between various vendors. This allows engineers to create

templates that perform the same task using different vendor syntaxes. When submitting jobs using a mix of vendors, the generated code will automatically reflect the appropriate vendor.

Service type extensions

About 80 new automated actions can be used in the definition of the service-types. This brings the total of automation actions up to 210. The service type handling has been redesigned allowing improved error handling and seamless integration with the API.

Vendor module redesign

To support a more versatile set of network change jobs, several features needed to be added to the existing vendor modules and the scenarios. These features allow for:

- Jobs can connect to multiple nodes at the same time to perform orchestrated changes.
- Jobs that require both console-port commands and management-port commands
- Jobs that include vendor API actions
- Jobs that require feedback form the device (command responses, running configuration)

To achieve this, all vendor modules have been redesigned to use a data-driven state-machine (per device, per connection). This setup will allow for a more modular approach where vendor features can be included where available or disabled where undesired.

Device-console support

Out-of-band device management using the serial console port is now supported. The use of a Terminal-server providing the Lan to Serial access is required.

New vendor support

New vendor modules have been created for:

- F5,
- Checkpoint,
- PaloAlto,
- Cisco Wism,
- Ciena v6,
- Ciena v8

Scenario redesign

Again, to support multiple connections and improved interaction with the devices from within the job scenarios, the scenario capabilities have been redesigned. New commands have been added, but

more importantly, support for variables, conditionals and loops have been included.

VPN tab

The main ‘build’ form has been extended with a ‘VPN’ tab next to the ‘Sites’ tab. This data grid displays the various VPN types a customer uses or is part of. By selecting a VPN entry, the participating nodes or services are filtered in the corresponding grid. There are no forms available from this grid.

Node classes

The ‘Design’ component ‘Node types’ has been extended with ‘Node classes’. Each Node type is part of a Node class which groups several Node types of the same role within a Client type. The node_class will replace the now obsolete numeric ‘nodeType’ attribute.

Custom tables

Data in customer specific tables that are used in NMS integrations or provide additional information can now be maintained within the NetYCE front-end. The tool can be found in “Admin - Custom data”. Access requires ‘Manager’ privileges. Users with ‘System’ privileges will find that all NetYCE tables are available in this tool.

OS upgrades unavailable

The tool that performs automated OS verifications and upgrades in bulk is not available with version 7.0. Due to the redesign of the vendor modules must this tool be redeveloped. Its functionality will be restored in version 7.1.

Version 7.1 work in progress

Much of the work realized in version 7.0 is to facilitate the new capabilities in version 7.1. These capabilities will be incorporated in version 7.0.x as soon as they are available, although usually for familiarization and testing purposes.

Support for Redhat 7.x Linux OS

Modifications to NetYCE process management, installation and setup scripts
Design is in progress

Dual stack IPv4 and IPv6

Communicate with the devices using IPv4 and IPv6
Access NetYCE frontend and server using IPv4 and IPv6
Requires Redhat 7.x as OS.
Design is in progress

Distributed scheduler

Assign jobs to remote queues and/or balance assigned jobs over various servers
Design is in progress

Distributed scheduler view

View all scheduled jobs of all servers and manipulate them
Design is in progress

Parsing templates

Defines the variables in their sections the scenario should extract from the life data
Development is in progress

Config parsing

Vendor based processing of life or nccm configurations and variable extraction
All existing vendor modules will need to be extended with a configuration ‘pre-parser’
Development is in progress

Command parsing

Vendor based processing of life or nccm command responses and variable extraction
Design is in progress

Scenario extensions

Use the extracted variables from the config / command to drive the scenario
Development is in progress

OS upgrades

Redesign this tool to use new vendor modules and capabilities
Development is suspended

REST API

The NetYCE API traditionally used XML formatted requests. Now a second highly scalable REST API is added that will handle the same Json formatted requests.
Design is in progress

Port classes support

Vendor specific L1, L2, and L3 port types must be supported
Design is in progress

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

Permanent link:
https://wiki.netyce.com/doku.php?id=maintenance:releases:historical:releasenotes_701

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