Service-type syntax

NetYCE currently supports a wide range of service-type commands. These commands total '327' which can be grouped in '19' object types that these commands manipulate.

Each object type has its internal alias-type that is used here as the alias-name for simplicity, but should be replaced with more descriptive names for any real-life applications.

Class name	Alias type
DOMAIN	<dom>
CLIENT	<clnt>
SITE	<sit>
SERVICE	<srv>
NODE	<node>
PORT	<port>
PORTS	<portlist>
LINK	<portlist>
SLOT	<slot>
SERVER	<server>
SUPERNET	<super>
SUBNET	<net>
DHCP	<dhcp>
ADDRESS	<addr>
IPV6_NET	<ipv6net>
IPV6_ADDR	<ipv6addr>
MPLS_VRF	<mpls_vrf>
VRF	<vrf>
CMDB	<cmdb>

Domain

The `DOMAIN` object uses the alias type `<dom>`

DOMAIN						
exec	*class*	*scope*	*match*	*value*	*alias*	*description*
DOMAIN as new alias						
ADD	DOMAIN	CURRENT	DOMAIN_NAME	value	<dom>	create new domain or locate existing. value is new domain name
LOCATE	DOMAIN	GLOBAL	DOMAIN_NAME	value	<dom>	find existing domain by name. value is domain name
DOMAIN as scope alias						
ASSIGN	DOMAIN	<dom>	-attribute-	value		assign the domain attribute a value. replace -attribute- with actual name.
DELETE	DOMAIN	<dom>	COMPLETE			delete Domain including custom parameters, networks and os_images. May not have YCE or CMDB nodes

DOMAIN						
DOMAIN alias in other contexts						
ADD	CMDB	<dom>	NODE	<node>	<cmdb>	Add a node to the CMDB using Domain alias. value = node-alias, nodename or fqdn
ADD	CMDB	<dom>	NODE	value	<cmdb>	Add a node to the CMDB using Domain alias. value = node-alias, nodename or fqdn
DOMAIN object in other contexts						
ASSIGN	CMDB	<cmdb>	DOMAIN	value		Assign a Domain to a cmdb-node/node-name. value = Domain
ASSIGN	CMDB	<node>	DOMAIN	value		Assign a Domain to a cmdb-node/node-name. value = Domain
DELETE	DOMAIN	GLOBAL	DOMAIN_NAME	value		delete Domain including custom parameters, networks and os_images. May not have YCE or CMDB nodes

Client

The `CLIENT` object uses the alias type `<clnt>`

CLIENT						
exec	*class*	*scope*	*match*	*value*	*alias*	*description*
CLIENT as new alias						
ADD	CLIENT	CURRENT	CLIENT_CODE	value	<clnt>	create new client of active Client_type. value is new client_code
LOCATE	CLIENT	CURRENT	CLIENT_CODE	value	<clnt>	find the client in the current client_type with specified client_code.
LOCATE	CLIENT	CURRENT	CURRENT	value	<clnt>	find the client in the current client_type with specified client_code.
CLIENT as scope alias						
ASSIGN	CLIENT	<clnt>	-attribute-	value		set the custom client attibute to the desired value
ASSIGN	CLIENT	<clnt>	PAR_GROUP	value		set the client custom attibute group to the desired value
DELETE	CLIENT	<clnt>	COMPLETE			delete client including sites, services, nodes, subnets, and topology
CLIENT as value alias						
ASSIGN	SERVER	<server>	CLIENT	<clnt>		assign the server to any client alias (move server)
CLIENT alias in other contexts						
ADD	SERVER	<clnt>	SERVER_KEY	value	<server>	create server with server_key on client alias

CLIENT						
ADD	SITE	<clnt>	SITE_CODE	value	<sit>	create new site for client. value is new site_code
ADD	SUPERNET	<clnt>	IP_PLAN	value	<super>	add new ip-supernet range to client from pool of free Supernets using numeric ip-plan ID. Or use API custom var “ip_supernet” to specify the supernet
ADD	SUPERNET	<clnt>	IP_SUPERNET	value	<super>	add new ip-supernet range to client. Format value as “<net-address>/<prefix>”. Set ip-plan using Assign-Supernet-Ip_plan or use API custom var “ip_plan”
LOCATE	NODE	<clnt>	NODE_NAME	value	<node>	find a node given its type in the alased client
LOCATE	NODE	<clnt>	NODE_POSITION	value	<node>	find a node given its position in the alased client
LOCATE	NODE	<clnt>	NODE_TYPE	value	<node>	find a node given its type in the alased client
LOCATE	SERVICE	<clnt>	SERVICE_CLASS	value	<srv>	find the service in the client-alias with the specified service-class
LOCATE	SERVICE	<clnt>	SERVICE_NAME	value	<srv>	find the service in the client-alias with the specified service-name
LOCATE	SERVICE	<clnt>	SERVICE_TYPE	value	<srv>	find the service in the client-alias with the specified service-type
LOCATE	SITE	<clnt>	SITE_CODE	value	<sit>	Locates the named site of the aliased client
LOCATE	SUPERNET	<clnt>	IP_PLAN	value	<super>	find the (first) supernet within the client matching the Ip_plan value
LOCATE	SUPERNET	<clnt>	IP_SUPERNET	value	<super>	find the supernet within the client matching the “<address>[/<prefix>]” value
LOCATE	SERVER	<clnt>	SERVER_NAME	value	<server>	find server with server_name on client alias
LOCATE	SERVER	<clnt>	SERVER_KEY	value	<server>	find server with server_key on client alias

Site

The SITE object uses the alias type <sit>

SITE						
exec	*class*	*scope*	*match*	*value*	*alias*	*description*
SITE as new alias						
ADD	SITE	<clnt>	SITE_CODE	value	<sit>	create new site for client. value is new site_code
LOCATE	SITE	<clnt>	SITE_CODE	value	<sit>	Locates the named site of the aliased client
LOCATE	SITE	GLOBAL	SITE_CODE	value	<sit>	Locates the named site globally

SITE						
SITE as scope alias						
ASSIGN	SITE	<sit>	-attribute-	value		set the custom site attibute to the desired value
ASSIGN	SITE	<sit>	PAR_GROUP	value		set the site custom attibute group to the desired value
DELETE	SITE	<sit>	COMPLETE			delete site including services, nodes, subnets and topology
SITE as value alias						
ASSIGN	SERVER	<server>	SITE	<sit>		assign the server to any site alias (move server)
SITE alias in other contexts						
ADD	SERVER	<sit>	SERVER_KEY	value	<server>	create server with server_key on site alias
ADD	SERVICE	<sit>	CURRENT	value	<srv>	creates a new service container on current site
ADD	SERVICE	SITE	SITE_CODE	<sit>	<srv>	creates a new service container on aliased site
LOCATE	SERVER	<sit>	SERVER_NAME	value	<server>	find server with server_name on site alias
LOCATE	NODE	<sit>	NODE_NAME	value	<node>	find a node given its name on the aliased site
LOCATE	NODE	<sit>	NODE_POSITION	value	<node>	find a node given its position on the aliased site
LOCATE	NODE	<sit>	NODE_TYPE	value	<node>	find a node given its type on the aliased site
LOCATE	SERVICE	<sit>	SERVICE_CLASS	value	<srv>	find the service on the site-alias with the specified service-class
LOCATE	SERVICE	<sit>	SERVICE_NAME	value	<srv>	find the service on the site-alias with the specified service-name
LOCATE	SERVICE	<sit>	SERVICE_TYPE	value	<srv>	find the service on the site-alias with the specified service-type
LOCATE	SERVER	<sit>	SERVER_KEY	value	<server>	find server with server_key on site alias

Service

The SERVICE object uses the alias type <srv>

SERVICE						
exec	*class*	*scope*	*match*	*value*	*alias*	*description*
SERVICE as new alias						
ADD	SERVICE	<sit>	CURRENT	value	<srv>	creates a new service container on current site
ADD	SERVICE	SITE	CURRENT	value	<srv>	creates a new service container on current site
ADD	SERVICE	SITE	SITE_CODE	<sit>	<srv>	creates a new service container on aliased site
LOCATE	SERVICE	CLIENT	CURRENT		<srv>	find the current selected service (tasks only)
LOCATE	SERVICE	CLIENT	SERVICE_CLASS	value	<srv>	find the service in the client with the specified service-class

SERVICE						
LOCATE	SERVICE	CLIENT	SERVICE_NAME	value	<srv>	find the service in the client with the specified service-name
LOCATE	SERVICE	CLIENT	SERVICE_TYPE	value	<srv>	find the service in the client with the specified service-type
LOCATE	SERVICE	<clnt>	SERVICE_CLASS	value	<srv>	find the service in the client-alias with the specified service-class
LOCATE	SERVICE	<clnt>	SERVICE_NAME	value	<srv>	find the service in the client-alias with the specified service-name
LOCATE	SERVICE	<clnt>	SERVICE_TYPE	value	<srv>	find the service in the client-alias with the specified service-type
LOCATE	SERVICE	GLOBAL	NODE	<node>	<srv>	find the service the node is part of
LOCATE	SERVICE	GLOBAL	NODE	value	<srv>	find the service the node is part of
LOCATE	SERVICE	GLOBAL	SERVICE_CLASS	value	<srv>	find the service globally with the specified service-class
LOCATE	SERVICE	GLOBAL	SERVICE_NAME	value	<srv>	find the service globally with the specified service-name
LOCATE	SERVICE	GLOBAL	SERVICE_TYPE	value	<srv>	find the service globally with the specified service-type
LOCATE	SERVICE	SERVICE	CURRENT		<srv>	find the current selected service (tasks only)
LOCATE	SERVICE	<sit>	SERVICE_CLASS	value	<srv>	find the service on the site-alias with the specified service-class
LOCATE	SERVICE	<sit>	SERVICE_NAME	value	<srv>	find the service on the site-alias with the specified service-name
LOCATE	SERVICE	<sit>	SERVICE_TYPE	value	<srv>	find the service on the site-alias with the specified service-type
LOCATE	SERVICE	SITE	SERVICE_CLASS	value	<srv>	find the service on the site with the specified service-class
LOCATE	SERVICE	SITE	SERVICE_NAME	value	<srv>	find the service on the site with the specified service-name
LOCATE	SERVICE	SITE	SERVICE_TYPE	value	<srv>	find the service on the site with the specified service-type
SERVICE as scope alias						
ASSIGN	SERVICE	<srv>	-attribute-	value		set the custom service attibute to the desired value
ASSIGN	SERVICE	<srv>	HIERARCHY_ID	value		set the service hierarchy-id to the desired value
ASSIGN	SERVICE	<srv>	PAR_GROUP	value		set the service custom attribute group to the desired value
ASSIGN	SERVICE	<srv>	SERVICE_CLASS	value		set the service class to the desired value
ASSIGN	SERVICE	<srv>	SERVICE_NAME	value		set the service name to the desired value
ASSIGN	SERVICE	<srv>	SERVICE_TYPE	value		set the service type to the desired value
DELETE	SERVICE	<srv>	COMPLETE			delete service including nodes, subnets and topology
SERVICE as value alias						
ASSIGN	SUBNET	<net>	SERVICE	<srv>		re-assign (move) the vlan/subnet to the service indicated
SERVICE alias in other contexts						
ADD	IPV6_NET	<srv>	CUSTOM	value	<ipv6net>	add new CUSTOM Ipv6-subnet. No IP-plan is used. Value is the net_name, the ipv6 address range must be assigned later
ADD	IPV6_NET	<srv>	NET_NAME	value	<ipv6net>	add an IPV6_NET by name. Value can be “ipv6net_name” or “ipv6_net_name - ipv6_plan_descr”
ADD	NODE	<srv>	NODE_TYPE	value	<node>	create new node in a service of specified node-type

SERVICE						
ADD	SUBNET	<srv>	CUSTOM	value	<net>	add new CUSTOM subnet. No IP-plan is used
ADD	SUBNET	<srv>	NET_NAME	value	<net>	locate and add subnet from an IP-plan by name
ADD	SUBNET	<srv>	NET_NAME_ NOTFULL	value	<net>	locate and conditionally add subnet from an IP-plan by name should the existing be 'full'
ADD	SUBNET	<srv>	NET_NAME_ AT_ADDRESS	value	<net>	create an ip-plan based subnet in a service using net-name AND subnet-address. Value format: “<net_name> <net_address>”
ADD	SUBNET	<srv>	NET_NAME_ FROM_ADDRESS	value	<net>	create an ip-plan based subnet in a service using net-name AND subnet-address. Finds the first free subnet. Value format: “<net_name> <net_address>”
ADD	SUBNET	<srv>	FIRSTFREE_ FROM_SUBNET	<net>	<net>	create an ip-plan based subnet using existing subnet-alias. Finds the first free subnet
ADD	SUBNET	<srv>	FIRSTFREE_ SUBNET_OFFSET	<net> <offset>	<net>	create an ip-plan based subnet using existing subnet-alias. Finds the first free subnet starting from the provided offset
LOCATE	IPV6_NET	<srv>	NET_ADDRESS	value	<ipv6net>	find the IPV6_NET in the service using its Ipv6 network-address. Use the optional prefix to normalize the network-address: <Ipv6_address>/<prefix>
LOCATE	IPV6_NET	<srv>	VLAN_ID	value	<ipv6net>	find the IPV6_NET in the service using its Vlan_id. The subnet with the fewest assigned ports is selected.
LOCATE	IPV6_NET	<srv>	NET_DESCR	value	<ipv6net>	find the IPV6_NET in the service using its Net_description or Net_name. The subnet with the fewest assigned ports is selected.
LOCATE	IPV6_NET	<srv>	NET_NAME	value	<ipv6net>	find the IPV6_NET in the service using its Net_name. The subnet with the fewest assigned ports is selected.
LOCATE	NODE	<srv>	NODE_NAME	value	<node>	find a node given its type in its service
LOCATE	NODE	<srv>	NODE_POSITION	value	<node>	find a node given its position in its service
LOCATE	NODE	<srv>	NODE_TYPE	value	<node>	find a node given its type in its service
LOCATE	SUBNET	<srv>	NET_ADDRESS	value	<net>	find the subnet in the service by network-address. Value may include the /prefix
LOCATE	SUBNET	<srv>	NET_DESCR	value	<net>	locate subnet by description in service
LOCATE	SUBNET	<srv>	NET_NAME	value	<net>	locate subnet by name *or* description in service
LOCATE	SUBNET	<srv>	NET_NAME_ NOTFULL	value	<net>	find and conditionally add subnet from an IP-plan by name should the existing be 'full'
LOCATE	SUBNET	<srv>	VLAN_ID	value	<net>	find the subnet by vlan-id in the service
LOCATE	IPV6_NET	<srv>	IPV6_NET	value	<ipv6net>	find the IPV6_NET in the service using Net_name and Ipv6-plan. Type_value uses the format: <ipv6_subnet_name>-<ipv6_plan_description>

Node

The NODE object uses the alias type <node>

NODE						
exec	***class***	***scope***	***match***	***value***	***alias***	***description***

NODE						
NODE as new alias						
ADD	NODE	<srv>	NODE_TYPE	value	<node>	create new node in a service of specified node-type
LOCATE	NODE	CLIENT	NODE_NAME	value	<node>	find a node given its type in the client
LOCATE	NODE	CLIENT	NODE_POSITION	value	<node>	find a node given its position in the client
LOCATE	NODE	CLIENT	NODE_TYPE	value	<node>	find a node given its type in the client
LOCATE	NODE	<clnt>	NODE_NAME	value	<node>	find a node given its type in the alased client
LOCATE	NODE	<clnt>	NODE_POSITION	value	<node>	find a node given its position in the alased client
LOCATE	NODE	<clnt>	NODE_TYPE	value	<node>	find a node given its type in the alased client
LOCATE	NODE	GLOBAL	NODE_NAME	value	<node>	find node using its name. Supports wildcard
LOCATE	NODE	<portlist>	DOWNLINK_NODE	<node>	<node>	finds the downlink node relative to the “node” using the topology of the interfaces in “1portlist”
LOCATE	NODE	<portlist>	DOWNLINK_NODE	value	<node>	finds the downlink node relative to the “node” using the topology of the interfaces in “2portlist”
LOCATE	NODE	<portlist>	PEERLINK_NODE	<node>	<node>	finds the peer/interlink node relative to the “node” using the topology of the interfaces in “portlist”
LOCATE	NODE	<portlist>	PEERLINK_NODE	value	<node>	finds the peer/interlink node relative to the “node” using the topology of the interfaces in “portlist”
LOCATE	NODE	<portlist>	UPLINK_NODE	<node>	<node>	finds the uplink node relative to the “node” using the topology of the interfaces in “portlist”
LOCATE	NODE	<portlist>	UPLINK_NODE	value	<node>	finds the uplink node relative to the “node” using the topology of the interfaces in “portlist”
LOCATE	NODE	SERVICE	CURRENT		<node>	find the selected node in the current service (front-end non-“create” tasks only)
LOCATE	NODE	<sit>	NODE_NAME	value	<node>	find a node given its name on the aliased site
LOCATE	NODE	<sit>	NODE_POSITION	value	<node>	find a node given its position on the aliased site
LOCATE	NODE	<sit>	NODE_TYPE	value	<node>	find a node given its type on the aliased site
LOCATE	NODE	SITE	NODE_NAME	value	<node>	find a node given its name on the site
LOCATE	NODE	SITE	NODE_POSITION	value	<node>	find a node given its position on the site

NODE						
LOCATE	NODE	SITE	NODE_TYPE	value	<node>	find a node given its type on the site
LOCATE	NODE	<srv>	NODE_NAME	value	<node>	find a node given its type in its service
LOCATE	NODE	<srv>	NODE_POSITION	value	<node>	find a node given its position in its service
LOCATE	NODE	<srv>	NODE_TYPE	value	<node>	find a node given its type in its service
NODE as scope alias						
LOCATE	LINK	<node>	NODE	<node>	<portlist>	find the ports involved linking these two nodes
ASSIGN	NODE	<node>	-attribute-	value		set the custom node attribute to the desired value
ASSIGN	NODE	<node>	IPSEC_TUNNEL	value		assign node to pre-existing ipsec tunnel-id
ASSIGN	NODE	<node>	NODE_NAME	value		rename existing node to specified name
ASSIGN	NODE	<node>	NODE_POSITION	value		assign a node to a position in the reference-model
ASSIGN	NODE	<node>	PAR_GROUP	value		set the node custom attribute group to the desired value
ASSIGN	NODE	<node>	REDUNDANT	value		rename existing node to specified name
ASSIGN	NODE	<node>	TEMPLATE	value		change Template and ports. Keeps obsolete topo/subnets, keeps port-details
ASSIGN	NODE	<node>	TEMPLATE_ FORCE	value		change Template and ports. Drops obsolete topo/subnets, updates port-details
ASSIGN	NODE	<node>	VRF	<vrf>		make node a member of the MPLS vrf specified
DELETE	NODE	<node>	COMPLETE			delete node including topology
DELETE	NODE	<node>	VRF	<vrf>		remove node from vrf
NODE as value alias						
ADD	CMDB	<dom>	NODE	<node>	<cmdb>	Add a node to the CMDB using Domain alias. value = node-alias, nodename or fqdn
LOCATE	LINK	<node>	NODE	<node>	<portlist>	find the ports involved linking these two nodes
LOCATE	NODE	<portlist>	DOWNLINK_NODE	<node>	<node>	finds the downlink node relative to the “node” using the topology of the interfaces in “1portlist”
LOCATE	NODE	<portlist>	PEERLINK_NODE	<node>	<node>	finds the peer/interlink node relative to the “node” using the topology of the interfaces in “portlist”

NODE						
LOCATE	NODE	<portlist>	UPLINK_NODE	<node>	<node>	finds the uplink node relative to the “node” using the topology of the interfaces in “portlist”
LOCATE	PORT	<portlist>	NODE	<node>	<port>	find first port in list matching the node
LOCATE	PORTS	<portlist>	NODE	<node>	<portlist>	find ports in list matching the node
LOCATE	SERVICE	GLOBAL	NODE	<node>	<srv>	find the service the node is part of
LOCATE	CMDB	CMDB	NODE	<node>	<cmdb>	Find a node in the CMDB by node name. value = node-alias, nodename or fqdn
ASSIGN	VRF	<vrf>	NODE	<node>		make node a member of the MPLS vrf specified
DELETE	VRF	<vrf>	NODE	<node>		remove node from vrf
DELETE	CMDB	CMDB	NODE	<node>		Delete a node from the CMDB by node name. value = node-alias, nodename or fqdn
NODE alias in other contexts						
ADD	PORT	<node>	ATM	value	<port>	value Interface-name: [type][slot/][module/]port. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORT	<node>	ETHERNET	value	<port>	value Interface-name: [type][slot/][module/]port. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORT	<node>	FAST_ETHERNET	value	<port>	value Interface-name: [type][slot/][module/]port. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORT	<node>	FOURTYGIGABIT_ ETHERNET	value	<port>	value Interface-name: [type][slot/][module/]port. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORT	<node>	GIGABIT_ ETHERNET	value	<port>	value Interface-name: [type][slot/][module/]port. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORT	<node>	HUNDREDGIGABIT_ ETHERNET	value	<port>	value Interface-name: [type][slot/][module/]port. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command

NODE						
ADD	PORT	<node>	LOOPBACK	value	<port>	value Interface-name: [type][slot/][module/]port. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORT	<node>	MANAGEMENT	value	<port>	value Interface-name: [type][slot/][module/]port. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORT	<node>	PORT_CHANNEL	value	<port>	value Interface-name: [type][slot/][module/]port. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORT	<node>	PORT_CHANNEL_ NEW	value	<port>	value = lowest portchannel-id number to start search for new id
ADD	PORT	<node>	SERIAL	value	<port>	value Interface-name: [type][slot/][module/]port. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORT	<node>	TENGIGABIT_ ETHERNET	value	<port>	value Interface-name: [type][slot/][module/]port. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORT	<node>	TYPE_FROM_ NAME	value	<port>	value Interface-name: [type][slot/][module/]port. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORTS	<node>	ATM	value	<portlist>	value Interface-name: [type][slot/][module/]port[-port]. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORTS	<node>	ETHERNET	value	<portlist>	value Interface-name: [type][slot/][module/]port[-port]. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORTS	<node>	FAST_ETHERNET	value	<portlist>	value Interface-name: [type][slot/][module/]port[-port]. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command

NODE						
ADD	PORTS	<node>	FOURTYGIGABIT_ETHERNET	value	<portlist>	value Interface-name: [type][slot/][module/]port[-port]. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORTS	<node>	GIGABIT_ETHERNET	value	<portlist>	value Interface-name: [type][slot/][module/]port[-port]. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORTS	<node>	HUNDREDGIGABIT_ETHERNET	value	<portlist>	value Interface-name: [type][slot/][module/]port[-port]. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORTS	<node>	SERIAL	value	<portlist>	value Interface-name: [type][slot/][module/]port[-port]. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORTS	<node>	TENGIGABIT_ETHERNET	value	<portlist>	value Interface-name: [type][slot/][module/]port[-port]. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORTS	<node>	TYPE_FROM_NAME	value	<portlist>	value Interface-name: [type][slot/][module/]port[-port]. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	VRF	<node>	VRF_ID	value	<vrf>	create vrf on node using its id. Note: node must have management / loopback address assigned first!
ADD	VRF	<node>	VRF_NAME	value	<vrf>	create vrf on node using its name. Note: node must have management / loopback address assigned first!
ADD	VRF	<node>	MPLS_VRF	mpls_vrf	<vrf>	create vrf on node using a Mpls_vrf alias. Note: node must have management / loopback address assigned first!
LOCATE	LINK	<node>	PORT_TEMPLATE	value	<portlist>	find the ports involved in a link using a local port-template
LOCATE	VRF	<node>	VRF_NAME	value	<vrf>	find a vrf on the node using its name

NODE						
LOCATE	PORT	<node>	PORT_NAME	value	<port>	find first port of a node matching the (wildcard) port-name. Use internal or vendor name e.g.: Gi01/01/1*, GigabitEthernet/1/1*
LOCATE	PORT	<node>	PORT_TEMPLATE	value	<port>	find first port using this template. Wildcard optional
LOCATE	PORT	<node>	PORT_TEMPLATE_FIRST	value	<port>	find first port using this template. Wildcard optional
LOCATE	PORT	<node>	PORT_TEMPLATE_FIRSTNOTOPO	value	<port>	find first port without topology using this template. Wildcard optional
LOCATE	PORT	<node>	PORT_TEMPLATE_FIRSTSHUT	value	<port>	find first disabled port with this template. Wildcard optional
LOCATE	PORT	<node>	PORT_TEMPLATE_LAST	value	<port>	find last port using this template. Wildcard optional
LOCATE	PORT	<node>	PORT_TEMPLATE_LASTNOTOPO	value	<port>	find first port without topology using this template. Wildcard optional
LOCATE	PORT	<node>	PORT_TEMPLATE_LASTSHUT	value	<port>	find last disabled port with this template. Wildcard optional
LOCATE	PORTS	<node>	ATM	value	<portlist>	value format= slot[/module]/port[-port]. Use '*' for any slot, module or port
LOCATE	PORTS	<node>	ETHERNET	value	<portlist>	value format= slot[/module]/port[-port]. Use '*' for any slot, module or port
LOCATE	PORTS	<node>	FAST_ETHERNET	value	<portlist>	value format= slot[/module]/port[-port]. Use '*' for any slot, module or port
LOCATE	PORTS	<node>	FOURTYGIGABIT_ETHERNET	value	<portlist>	value format= slot[/module]/port[-port]. Use '*' for any slot, module or port
LOCATE	PORTS	<node>	GIGABIT_ETHERNET	value	<portlist>	value format= slot[/module]/port[-port]. Use '*' for any slot, module or port
LOCATE	PORTS	<node>	HUNDREDGIGABIT_ETHERNET	value	<portlist>	value format= slot[/module]/port[-port]. Use '*' for any slot, module or port
LOCATE	PORTS	<node>	LOOPBACK	value	<portlist>	value format= slot[/module]/port[-port]. Use '*' for any slot, module or port
LOCATE	PORTS	<node>	MANAGEMENT	value	<portlist>	value format= slot[/module]/port[-port]. Use '*' for any slot, module or port
LOCATE	PORTS	<node>	PORT_CHANNEL	value	<portlist>	value format= slot[/module]/port[-port]. Use '*' for any slot, module or port
LOCATE	PORTS	<node>	PORT_TEMPLATE_ALL	value	<portlist>	find all ports in list with this template

NODE						
LOCATE	PORTS	<node>	PORT_TEMPLATE_ALLNOTOPO	value	<portlist>	find all ports in list with this template without topology
LOCATE	PORTS	<node>	SERIAL	value	<portlist>	value format= slot[/module]/port[-port]. Use '*' for any slot, module or port
LOCATE	PORTS	<node>	TENGIGABIT_ETHERNET	value	<portlist>	value format= slot[/module]/port[-port]. Use '*' for any slot, module or port
LOCATE	SLOT	<node>	SLOT_ID	value	<slot>	find all interfaces using slot_id. Value format = slot[/module]. You can use '/' to indicate blank slot/module
LOCATE	VRF	<node>	VRF_ID	value	<vrf>	find a vrf on the node using its id
LOCATE	ADDRESS	<node>	MANAGEMENT		<addr>	Returns the management ip-address for the <node> as an <addr> alias
LOCATE	PORTS	<node>	PORT_NAME	value	<portlist>	find all ports of a node matching the (wildcard) port-name. Use internal or vendor name e.g.: Gi01/01/1*, GigabitEthernet/1/1*
LOCATE	PORTS	<node>	-attribute-	value	<portlist>	find all ports of a node matching the (wildcard) attribute value
LOCATE	PORT	<node>	-attribute-	value	<port>	find first port of a node matching the (wildcard) attribute value
ASSIGN	EVPN_VLAN	<node>	EVPN_CORE	<node>		assign eVPN to a core-node by node-alias
ASSIGN	EVPN_VLAN	<node>	EVPN_CORE	value		assign eVPN to a core-node by name
ASSIGN	CMDB	<node>	VENDOR	value		Assign a Vendor to a cmdb-alias/node-name. value = Vendor_type name
ASSIGN	CMDB	<node>	FQDN	value		Assign a fqdn (NodeName) to a cmdb-node/node-name. value = fqdn or ip-address
ASSIGN	CMDB	<node>	DOMAIN	value		Assign a Domain to a cmdb-node/node-name. value = Domain
ASSIGN	CMDB	<node>	-attribute-	value		Assign a value to any attribute of a cmdb-node/node-name. Replace -attribute for attribute name. value = attribute value
DELETE	SLOT	<node>	SLOT_ID	value		remove ports based on slot-id, delete any attached topology
NODE object in other contexts						
ADD	CMDB	<dom>	NODE	value	<cmdb>	Add a node to the CMDB using Domain alias. value = node-alias, nodename or fqdn
LOCATE	SERVICE	GLOBAL	NODE	value	<srv>	find the service the node is part of

NODE						
LOCATE	CMDB	CMDB	NODE	value	<cmdb>	Find a node in the CMDB by node name. value = node-alias, nodename or fqdn
DELETE	CMDB	CMDB	NODE	value		Delete a node from the CMDB by node name. value = node-alias, nodename or fqdn

Port

The PORT object uses the alias type `<port>`

PORT						
exec	***class***	***scope***	***match***	***value***	***alias***	***description***
PORT as new alias						
ADD	PORT	<node>	ATM	value	<port>	value Interface-name: [type][slot/][module/]port. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORT	<node>	ETHERNET	value	<port>	value Interface-name: [type][slot/][module/]port. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORT	<node>	FAST_ETHERNET	value	<port>	value Interface-name: [type][slot/][module/]port. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORT	<node>	FOURTYGIGABIT_ ETHERNET	value	<port>	value Interface-name: [type][slot/][module/]port. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORT	<node>	GIGABIT_ ETHERNET	value	<port>	value Interface-name: [type][slot/][module/]port. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command

PORT						
ADD	PORT	<node>	HUNDREDGIGABIT_ETHERNET	value	<port>	value Interface-name: [type][slot/][module/]port. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORT	<node>	LOOPBACK	value	<port>	value Interface-name: [type][slot/][module/]port. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORT	<node>	MANAGEMENT	value	<port>	value Interface-name: [type][slot/][module/]port. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORT	<node>	PORT_CHANNEL	value	<port>	value Interface-name: [type][slot/][module/]port. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORT	<node>	PORT_CHANNEL_NEW	value	<port>	value = lowest portchannel-id number to start search for new id
ADD	PORT	<node>	SERIAL	value	<port>	value Interface-name: [type][slot/][module/]port. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORT	<node>	TENGIGABIT_ETHERNET	value	<port>	value Interface-name: [type][slot/][module/]port. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORT	<node>	TYPE_FROM_NAME	value	<port>	value Interface-name: [type][slot/][module/]port. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command

PORT						
LOCATE	PORT	<node>	PORT_NAME	value	<port>	find first port of a node matching the (wildcard) port-name. Use internal or vendor name e.g.: Gi01/01/1*, GigabitEthernet/1/1*
LOCATE	PORT	<node>	PORT_TEMPLATE	value	<port>	find first port using this template. Wildcard optional
LOCATE	PORT	<node>	PORT_TEMPLATE_FIRST	value	<port>	find first port using this template. Wildcard optional
LOCATE	PORT	<node>	PORT_TEMPLATE_FIRSTNOTOPO	value	<port>	find first port without topology using this template. Wildcard optional
LOCATE	PORT	<node>	PORT_TEMPLATE_FIRSTSHUT	value	<port>	find first disabled port with this template. Wildcard optional
LOCATE	PORT	<node>	PORT_TEMPLATE_LAST	value	<port>	find last port using this template. Wildcard optional
LOCATE	PORT	<node>	PORT_TEMPLATE_LASTNOTOPO	value	<port>	find first port without topology using this template. Wildcard optional
LOCATE	PORT	<node>	PORT_TEMPLATE_LASTSHUT	value	<port>	find last disabled port with this template. Wildcard optional
LOCATE	PORT	<port>	AT_SUBNET	<net>	<port>	Locate the given port if assigned to the given subnet. Port and subnet are aliases
LOCATE	PORT	<portlist>	NODE	<node>	<port>	find first port in list matching the node
LOCATE	PORT	<portlist>	PORT_TEMPLATE_FIRST	value	<port>	find first port in list with this template
LOCATE	PORT	<portlist>	PORT_TEMPLATE_FIRSTNOTOPO	value	<port>	find first port in list with this template without topology
LOCATE	PORT	<portlist>	PORT_TEMPLATE_LAST	value	<port>	find last port in list with this template
LOCATE	PORT	<portlist>	PORT_TEMPLATE_LASTNOTOPO	value	<port>	find last port in list with this template without topology
LOCATE	PORT	<slot>	PORT_TEMPLATE_FIRST	value	<port>	find first port in list with this template
LOCATE	PORT	<slot>	PORT_TEMPLATE_FIRSTNOTOPO	value	<port>	find first port in list with this template without topology

PORT						
LOCATE	PORT	<slot>	PORT_TEMPLATE_LAST	value	<port>	find last port in list with this template
LOCATE	PORT	<slot>	PORT_TEMPLATE_LASTNOTOPO	value	<port>	find last port in list with this template without topology
LOCATE	PORT	<port>	AT_IPV6_NET	<ipv6net>	<port>	Locate the given port if assigned to the given Ipv6 subnet. Port and Ipv6 subnet are aliases
LOCATE	PORT	<portlist>	PORT_NAME	value	<port>	find first port within a portlist matching the (wildcard) name. Use internal or vendor name e.g.: Gi01/01/1*, GigabitEthernet/1/1*
LOCATE	PORT	<node>	-attribute-	value	<port>	find first port of a node matching the (wildcard) attribute value
LOCATE	PORT	<portlist>	-attribute-	value	<port>	find first port within a portlist matching the (wildcard) attrbute value
PORT as scope alias						
ADD	LINK	<port>	PORT	<port>	<portlist>	connect two ports using topology
LOCATE	PORT	<port>	AT_SUBNET	<net>	<port>	Locate the given port if assigned to the given subnet. Port and subnet are aliases
LOCATE	PORT	<port>	AT_IPV6_NET	<ipv6net>	<port>	Locate the given port if assigned to the given Ipv6 subnet. Port and Ipv6 subnet are aliases
ASSIGN	PORT	<port>	-attribute-	value		assign port attribute a value. replace -attribute- with actual name.
ASSIGN	PORT	<port>	PORT_CHANNEL	value		make the port a port-channel member of specified channel-id
ASSIGN	PORT	<port>	PORT_CHANNEL_OF	<port>		make the port a member of the channel-group of the other port
ASSIGN	PORT	<port>	PORT_DESCR	value		update the port_description. Defaults to template-description
ASSIGN	PORT	<port>	PORT_MODE	value		set the duplex mode of the port
ASSIGN	PORT	<port>	PORT_NAME_OF	<port>		move the port to the other port's location. Drops the target port along with topo/subs
ASSIGN	PORT	<port>	PORT_SHUT	value		enable or disable the port (shut=Y → disabled)

PORT						
ASSIGN	PORT	<port>	PORT_SPEED	value		set the port speed in Mbps
ASSIGN	PORT	<port>	PORT_TEMPLATE	value		set the port's port-template
ASSIGN	PORT	<port>	PORT_TEMPLATE_OF	<port>		set the port's port-template to that of the other port
ASSIGN	PORT	<port>	SUBNET	<net>		make the port a member of the vlan/subnet
ASSIGN	PORT	<port>	IPV6_NET	<ipv6net>		make the port a member of the Ipv6 vlan/subnet
DELETE	PORT	<port>	COMPLETE			delete port including topology
DELETE	PORT	<port>	SUBNET	<net>		remove subnet from port
DELETE	PORT	<port>	SUBNET_ALL			remove all subnet assignments from aliased port
DELETE	PORT	<port>	TOPOLOGY			remove topology from port
PORT as value alias						
ADD	LINK	<port>	PORT	<port>	<portlist>	connect two ports using topology
ASSIGN	ADDRESS	<addr>	PORT	<port>		set a single IP-address on a Management or Loopback interface
ASSIGN	IPV6_ADDR	<ipv6addr>	PORT	<port>		set a single IP-IPV6_ADDR on a Management or Loopback interface
ASSIGN	IPV6_NET	<ipv6net>	PORT	<port>		make the port a member of the Ipv6 vlan/subnet
ASSIGN	PORT	<port>	PORT_CHANNEL_OF	<port>		make the port a member of the channel-group of the other port
ASSIGN	PORT	<port>	PORT_NAME_OF	<port>		move the port to the other port's location. Drops the target port along with topo/subs
ASSIGN	PORT	<port>	PORT_TEMPLATE_OF	<port>		set the port's port-template to that of the other port
ASSIGN	SUBNET	<net>	PORT	<port>		make the port a member of the vlan/subnet
DELETE	IPV6_NET	<ipv6net>	PORT	<port>		remove IPV6_NET from port
DELETE	SUBNET	<net>	PORT	<port>		remove subnet from port
PORT alias in other contexts						

PORT						
LOCATE	SUBNET	<net>	AT_PORT	<port>	<net>	Locate the given subnet if assigned to the given port. Subnet and port are aliases
LOCATE	IPV6_NET	<ipv6net>	AT_PORT	<port>	<ipv6net>	Locate the given Ipv6 subnet if assigned to the given port. Ipv6 subnet and port are aliases
ASSIGN	PORTS	<portlist>	PORT_CHANNEL_OF	<port>		make the ports a member of the channel-group of the other port
ASSIGN	PORTS	<portlist>	PORT_TEMPLATE_OF	<port>		set the port port-template to that of the other port

Ports

The PORTS object uses the alias type `<portlist>`

PORTS						
exec	*class*	*scope*	*match*	*value*	*alias*	*description*
PORTS as new alias						
ADD	PORTS	<node>	ATM	value	<portlist>	value Interface-name: [type][slot/][module/]port[-port]. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORTS	<node>	ETHERNET	value	<portlist>	value Interface-name: [type][slot/][module/]port[-port]. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORTS	<node>	FAST_ETHERNET	value	<portlist>	value Interface-name: [type][slot/][module/]port[-port]. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORTS	<node>	FOURTYGIGABIT_ETHERNET	value	<portlist>	value Interface-name: [type][slot/][module/]port[-port]. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORTS	<node>	GIGABIT_ETHERNET	value	<portlist>	value Interface-name: [type][slot/][module/]port[-port]. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command

PORTS						
ADD	PORTS	<node>	HUNDREDGIGABIT_ETHERNET	value	<portlist>	value Interface-name: [type][slot/][module/]port[-port]. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORTS	<node>	SERIAL	value	<portlist>	value Interface-name: [type][slot/][module/]port[-port]. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORTS	<node>	TENGIGABIT_ETHERNET	value	<portlist>	value Interface-name: [type][slot/][module/]port[-port]. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORTS	<node>	TYPE_FROM_NAME	value	<portlist>	value Interface-name: [type][slot/][module/]port[-port]. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
LOCATE	PORTS	<node>	ATM	value	<portlist>	value format= slot[/module]/port[-port]. Use '*' for any slot, module or port
LOCATE	PORTS	<node>	ETHERNET	value	<portlist>	value format= slot[/module]/port[-port]. Use '*' for any slot, module or port
LOCATE	PORTS	<node>	FAST_ETHERNET	value	<portlist>	value format= slot[/module]/port[-port]. Use '*' for any slot, module or port
LOCATE	PORTS	<node>	FOURTYGIGABIT_ETHERNET	value	<portlist>	value format= slot[/module]/port[-port]. Use '*' for any slot, module or port
LOCATE	PORTS	<node>	GIGABIT_ETHERNET	value	<portlist>	value format= slot[/module]/port[-port]. Use '*' for any slot, module or port
LOCATE	PORTS	<node>	HUNDREDGIGABIT_ETHERNET	value	<portlist>	value format= slot[/module]/port[-port]. Use '*' for any slot, module or port
LOCATE	PORTS	<node>	LOOPBACK	value	<portlist>	value format= slot[/module]/port[-port]. Use '*' for any slot, module or port
LOCATE	PORTS	<node>	MANAGEMENT	value	<portlist>	value format= slot[/module]/port[-port]. Use '*' for any slot, module or port
LOCATE	PORTS	<node>	PORT_CHANNEL	value	<portlist>	value format= slot[/module]/port[-port]. Use '*' for any slot, module or port

PORTS						
LOCATE	PORTS	<node>	PORT_TEMPLATE_ALL	value	<portlist>	find all ports in list with this template
LOCATE	PORTS	<node>	PORT_TEMPLATE_ALLNOTOPO	value	<portlist>	find all ports in list with this template without topology
LOCATE	PORTS	<node>	SERIAL	value	<portlist>	value format= slot[/module]/port[-port]. Use '*' for any slot, module or port
LOCATE	PORTS	<node>	TENGIGABIT_ETHERNET	value	<portlist>	value format= slot[/module]/port[-port]. Use '*' for any slot, module or port
LOCATE	PORTS	<portlist>	AT_SUBNET	<net>	<portlist>	Locate ports in the given list if assigned to the given subnet. Ports and subnet are aliases
LOCATE	PORTS	<portlist>	NODE	<node>	<portlist>	find ports in list matching the node
LOCATE	PORTS	<portlist>	PORT_TEMPLATE_ALL	value	<portlist>	find all the ports in list with this template
LOCATE	PORTS	<portlist>	AT_IPV6_NET	<ipv6net>	<portlist>	Locate ports in the given list if assigned to the given Ipv6 subnet. Ports and Ipv6 subnet are aliases
LOCATE	PORTS	<node>	PORT_NAME	value	<portlist>	find all ports of a node matching the (wildcard) port-name. Use internal or vendor name e.g.: Gi01/01/1*, GigabitEthernet/1/1*
LOCATE	PORTS	<portlist>	PORT_NAME	value	<portlist>	find ports within a portlist matching the (wildcard) name. Use internal or vendor name e.g.: Gi01/01/1*, GigabitEthernet/1/1*
LOCATE	PORTS	<node>	-attribute-	value	<portlist>	find all ports of a node matching the (wildcard) attribute value
LOCATE	PORTS	<portlist>	-attribute-	value	<portlist>	find ports within a portlist matching the (wildcard) attrbute value
PORTS as scope alias						
LOCATE	PORTS	<portlist>	AT_SUBNET	<net>	<portlist>	Locate ports in the given list if assigned to the given subnet. Ports and subnet are aliases
LOCATE	PORTS	<portlist>	NODE	<node>	<portlist>	find ports in list matching the node
LOCATE	PORTS	<portlist>	PORT_TEMPLATE_ALL	value	<portlist>	find all the ports in list with this template
LOCATE	PORTS	<portlist>	AT_IPV6_NET	<ipv6net>	<portlist>	Locate ports in the given list if assigned to the given Ipv6 subnet. Ports and Ipv6 subnet are aliases
LOCATE	PORTS	<portlist>	PORT_NAME	value	<portlist>	find ports within a portlist matching the (wildcard) name. Use internal or vendor name e.g.: Gi01/01/1*, GigabitEthernet/1/1*

PORTS						
LOCATE	PORTS	<portlist>	-attribute-	value	<portlist>	find ports within a portlist matching the (wildcard) attrbute value
ASSIGN	PORTS	<portlist>	-attribute-	value		assign port attribute a value. replace -attribute- with actual name.
ASSIGN	PORTS	<portlist>	PORT_CHANNEL	value		make the ports a port-channel member of specified channel-id
ASSIGN	PORTS	<portlist>	PORT_CHANNEL_OF	<port>		make the ports a member of the channel-group of the other port
ASSIGN	PORTS	<portlist>	PORT_MODE	value		set the duplex mode of the ports
ASSIGN	PORTS	<portlist>	PORT_SHUT	value		enable or disable the ports (shut=Y → disabled)
ASSIGN	PORTS	<portlist>	PORT_SPEED	value		set the ports speed in Mbps
ASSIGN	PORTS	<portlist>	PORT_TEMPLATE	value		set the ports port-template
ASSIGN	PORTS	<portlist>	PORT_TEMPLATE_OF	<port>		set the port port-template to that of the other port
ASSIGN	PORTS	<portlist>	SUBNET	<net>		make the ports a member of the vlan/subnet
ASSIGN	PORTS	<portlist>	IPV6_NET	<ipv6net>		make the ports a member of the Ipv6 vlan/subnet
DELETE	PORTS	<portlist>	COMPLETE			delete ports including topology
DELETE	PORTS	<portlist>	SUBNET	<net>		remove subnet from ports
DELETE	PORTS	<portlist>	SUBNET_ALL			remove all subnet assignments from aliased ports
DELETE	PORTS	<portlist>	TOPOLOGY			remove topology from ports
PORTS as value alias						
ASSIGN	IPV6_NET	<ipv6net>	PORTS	<portlist>		make the ports a member of the Ipv6 vlan/subnet
ASSIGN	SUBNET	<net>	PORTS	<portlist>		make the ports a member of the vlan/subnet
DELETE	IPV6_NET	<ipv6net>	PORTS	<portlist>		remove IPV6_NET from ports
DELETE	SUBNET	<net>	PORTS	<portlist>		remove subnet from ports
PORTS alias in other contexts						
ADD	LINK	<port>	PORT	<port>	<portlist>	connect two ports using topology
LOCATE	LINK	<node>	NODE	<node>	<portlist>	find the ports involved linking these two nodes
LOCATE	LINK	<node>	PORT_TEMPLATE	value	<portlist>	find the ports involved in a link using a local port-template
LOCATE	NODE	<portlist>	DOWNLINK_NODE	<node>	<node>	finds the downlink node relative to the “node” using the topology of the interfaces in “1portlist”
LOCATE	NODE	<portlist>	DOWNLINK_NODE	value	<node>	finds the downlink node relative to the “node” using the topology of the interfaces in “2portlist”

PORTS						
LOCATE	NODE	<portlist>	PEERLINK_NODE	<node>	<node>	finds the peer/interlink node relative to the “node” using the topology of the interfaces in “portlist”
LOCATE	NODE	<portlist>	PEERLINK_NODE	value	<node>	finds the peer/interlink node relative to the “node” using the topology of the interfaces in “portlist”
LOCATE	NODE	<portlist>	UPLINK_NODE	<node>	<node>	finds the uplink node relative to the “node” using the topology of the interfaces in “portlist”
LOCATE	NODE	<portlist>	UPLINK_NODE	value	<node>	finds the uplink node relative to the “node” using the topology of the interfaces in “portlist”
LOCATE	PORT	<portlist>	NODE	<node>	<port>	find first port in list matching the node
LOCATE	PORT	<portlist>	PORT_TEMPLATE_FIRST	value	<port>	find first port in list with this template
LOCATE	PORT	<portlist>	PORT_TEMPLATE_FIRSTNOTOPO	value	<port>	find first port in list with this template without topology
LOCATE	PORT	<portlist>	PORT_TEMPLATE_LAST	value	<port>	find last port in list with this template
LOCATE	PORT	<portlist>	PORT_TEMPLATE_LASTNOTOPO	value	<port>	find last port in list with this template without topology
LOCATE	PORT	<portlist>	PORT_NAME	value	<port>	find first port within a portlist matching the (wildcard) name. Use internal or vendor name e.g.: Gi01/01/1*, GigabitEthernet/1/1*
LOCATE	PORT	<portlist>	-attribute-	value	<port>	find first port within a portlist matching the (wildcard) attrbute value
ASSIGN	SUBNET	<net>	LINK	<portlist>		make the ports of the link a member of the vlan/subnet
ASSIGN	IPV6_NET	<ipv6net>	LINK	<portlist>		make the ports of the link a member of the Ipv6 vlan/subnet
ASSIGN	LINK	<portlist>	IPV6_NET	<ipv6net>		make the ports of the link a member of the Ipv6 vlan/subnet

Link

The `LINK` object uses the alias type `<portlist>`

LINK						
exec	***class***	***scope***	***match***	***value***	***alias***	***description***
LINK as new alias						
ADD	LINK	<port>	PORT	<port>	<portlist>	connect two ports using topology
LOCATE	LINK	<node>	NODE	<node>	<portlist>	find the ports involved linking these two nodes

LINK						
LOCATE	LINK	<node>	PORT_TEMPLATE	value	<portlist>	find the ports involved in a link using a local port-template
LINK as scope alias						
ASSIGN	LINK	<portlist>	IPV6_NET	<ipv6net>		make the ports of the link a member of the Ipv6 vlan/subnet
LINK as value alias						
ASSIGN	SUBNET	<net>	LINK	<portlist>		make the ports of the link a member of the vlan/subnet
ASSIGN	IPV6_NET	<ipv6net>	LINK	<portlist>		make the ports of the link a member of the Ipv6 vlan/subnet
LINK alias in other contexts						
ADD	PORTS	<node>	ATM	value	<portlist>	value Interface-name: [type][slot/][module/]port[-port]. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORTS	<node>	ETHERNET	value	<portlist>	value Interface-name: [type][slot/][module/]port[-port]. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORTS	<node>	FAST_ETHERNET	value	<portlist>	value Interface-name: [type][slot/][module/]port[-port]. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORTS	<node>	FOURTYGIGABIT_ETHERNET	value	<portlist>	value Interface-name: [type][slot/][module/]port[-port]. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORTS	<node>	GIGABIT_ETHERNET	value	<portlist>	value Interface-name: [type][slot/][module/]port[-port]. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORTS	<node>	HUNDREDGIGABIT_ETHERNET	value	<portlist>	value Interface-name: [type][slot/][module/]port[-port]. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command

LINK						
ADD	PORTS	<node>	SERIAL	value	<portlist>	value Interface-name: [type][slot/][module/]port[-port]. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORTS	<node>	TENGIGABIT_ETHERNET	value	<portlist>	value Interface-name: [type][slot/][module/]port[-port]. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
ADD	PORTS	<node>	TYPE_FROM_NAME	value	<portlist>	value Interface-name: [type][slot/][module/]port[-port]. Type may be vendor interface-name or NetYCE internal Port_class and overrides type in command
LOCATE	NODE	<portlist>	DOWNLINK_NODE	<node>	<node>	finds the downlink node relative to the “node” using the topology of the interfaces in “1portlist”
LOCATE	NODE	<portlist>	DOWNLINK_NODE	value	<node>	finds the downlink node relative to the “node” using the topology of the interfaces in “2portlist”
LOCATE	NODE	<portlist>	PEERLINK_NODE	<node>	<node>	finds the peer/interlink node relative to the “node” using the topology of the interfaces in “portlist”
LOCATE	NODE	<portlist>	PEERLINK_NODE	value	<node>	finds the peer/interlink node relative to the “node” using the topology of the interfaces in “portlist”
LOCATE	NODE	<portlist>	UPLINK_NODE	<node>	<node>	finds the uplink node relative to the “node” using the topology of the interfaces in “portlist”
LOCATE	NODE	<portlist>	UPLINK_NODE	value	<node>	finds the uplink node relative to the “node” using the topology of the interfaces in “portlist”
LOCATE	PORT	<portlist>	NODE	<node>	<port>	find first port in list matching the node
LOCATE	PORT	<portlist>	PORT_TEMPLATE_FIRST	value	<port>	find first port in list with this template
LOCATE	PORT	<portlist>	PORT_TEMPLATE_FIRSTNOTOPO	value	<port>	find first port in list with this template without topology
LOCATE	PORT	<portlist>	PORT_TEMPLATE_LAST	value	<port>	find last port in list with this template
LOCATE	PORT	<portlist>	PORT_TEMPLATE_LASTNOTOPO	value	<port>	find last port in list with this template without topology
LOCATE	PORTS	<node>	ATM	value	<portlist>	value format= slot[/module]/port[-port]. Use '*' for any slot, module or port

LINK						
LOCATE	PORTS	<node>	ETHERNET	value	<portlist>	value format= slot[/module]/port[-port]. Use '*' for any slot, module or port
LOCATE	PORTS	<node>	FAST_ETHERNET	value	<portlist>	value format= slot[/module]/port[-port]. Use '*' for any slot, module or port
LOCATE	PORTS	<node>	FOURTYGIGABIT_ ETHERNET	value	<portlist>	value format= slot[/module]/port[-port]. Use '*' for any slot, module or port
LOCATE	PORTS	<node>	GIGABIT_ ETHERNET	value	<portlist>	value format= slot[/module]/port[-port]. Use '*' for any slot, module or port
LOCATE	PORTS	<node>	HUNDREDGIGABIT_ ETHERNET	value	<portlist>	value format= slot[/module]/port[-port]. Use '*' for any slot, module or port
LOCATE	PORTS	<node>	LOOPBACK	value	<portlist>	value format= slot[/module]/port[-port]. Use '*' for any slot, module or port
LOCATE	PORTS	<node>	MANAGEMENT	value	<portlist>	value format= slot[/module]/port[-port]. Use '*' for any slot, module or port
LOCATE	PORTS	<node>	PORT_CHANNEL	value	<portlist>	value format= slot[/module]/port[-port]. Use '*' for any slot, module or port
LOCATE	PORTS	<node>	PORT_TEMPLATE_ ALL	value	<portlist>	find all ports in list with this template
LOCATE	PORTS	<node>	PORT_TEMPLATE_ ALLNOTOPO	value	<portlist>	find all ports in list with this template without topology
LOCATE	PORTS	<node>	SERIAL	value	<portlist>	value format= slot[/module]/port[-port]. Use '*' for any slot, module or port
LOCATE	PORTS	<node>	TENGIGABIT_ ETHERNET	value	<portlist>	value format= slot[/module]/port[-port]. Use '*' for any slot, module or port
LOCATE	PORTS	<portlist>	AT_SUBNET	<net>	<portlist>	Locate ports in the given list if assigned to the given subnet. Ports and subnet are aliases
LOCATE	PORTS	<portlist>	NODE	<node>	<portlist>	find ports in list matching the node
LOCATE	PORTS	<portlist>	PORT_TEMPLATE_ ALL	value	<portlist>	find all the ports in list with this template
LOCATE	PORTS	<portlist>	AT_IPV6_NET	<ipv6net>	<portlist>	Locate ports in the given list if assigned to the given Ipv6 subnet. Ports and Ipv6 subnet are aliases
LOCATE	PORTS	<node>	PORT_NAME	value	<portlist>	find all ports of a node matching the (wildcard) port-name. Use internal or vendor name e.g.: Gi01/01/1*, GigabitEthernet/1/1*

LINK						
LOCATE	PORTS	<portlist>	PORT_NAME	value	<portlist>	find ports within a portlist matching the (wildcard) name. Use internal or vendor name e.g.: Gi01/01/1*, GigabitEthernet/1/1*
LOCATE	PORT	<portlist>	PORT_NAME	value	<port>	find first port within a portlist matching the (wildcard) name. Use internal or vendor name e.g.: Gi01/01/1*, GigabitEthernet/1/1*
LOCATE	PORTS	<node>	-attribute-	value	<portlist>	find all ports of a node matching the (wildcard) attribute value
LOCATE	PORTS	<portlist>	-attribute-	value	<portlist>	find ports within a portlist matching the (wildcard) attrbute value
LOCATE	PORT	<portlist>	-attribute-	value	<port>	find first port within a portlist matching the (wildcard) attrbute value
ASSIGN	IPV6_NET	<ipv6net>	PORTS	<portlist>		make the ports a member of the Ipv6 vlan/subnet
ASSIGN	PORTS	<portlist>	-attribute-	value		assign port attribute a value. replace -attribute- with actual name.
ASSIGN	PORTS	<portlist>	PORT_CHANNEL	value		make the ports a port-channel member of specified channel-id
ASSIGN	PORTS	<portlist>	PORT_CHANNEL_OF	<port>		make the ports a member of the channel-group of the other port
ASSIGN	PORTS	<portlist>	PORT_MODE	value		set the duplex mode of the ports
ASSIGN	PORTS	<portlist>	PORT_SHUT	value		enable or disable the ports (shut=Y → disabled)
ASSIGN	PORTS	<portlist>	PORT_SPEED	value		set the ports speed in Mbps
ASSIGN	PORTS	<portlist>	PORT_TEMPLATE	value		set the ports port-template
ASSIGN	PORTS	<portlist>	PORT_TEMPLATE_OF	<port>		set the port port-template to that of the other port
ASSIGN	PORTS	<portlist>	SUBNET	<net>		make the ports a member of the vlan/subnet
ASSIGN	SUBNET	<net>	PORTS	<portlist>		make the ports a member of the vlan/subnet
ASSIGN	PORTS	<portlist>	IPV6_NET	<ipv6net>		make the ports a member of the Ipv6 vlan/subnet
DELETE	IPV6_NET	<ipv6net>	PORTS	<portlist>		remove IPV6_NET from ports
DELETE	PORTS	<portlist>	COMPLETE			delete ports including topology
DELETE	PORTS	<portlist>	SUBNET	<net>		remove subnet from ports
DELETE	PORTS	<portlist>	SUBNET_ALL			remove all subnet assignments from aliased ports
DELETE	PORTS	<portlist>	TOPOLOGY			remove topology from ports
DELETE	SUBNET	<net>	PORTS	<portlist>		remove subnet from ports

Slot

The SLOT object uses the alias type `<slot>`

SLOT						
exec	*class*	*scope*	*match*	*value*	*alias*	*description*
SLOT as new alias						
LOCATE	SLOT	<node>	SLOT_ID	value	<slot>	find all interfaces using slot_id. Value format = slot[/module]. You can use '/' to indicate blank slot/module
SLOT as scope alias						
ASSIGN	SLOT	<slot>	SLOT_ID	value		Move all interfaces to a new slot_id. Value format = slot[/module]. You can use '/' to indicate blank slot/module
SLOT alias in other contexts						
LOCATE	PORT	<slot>	PORT_TEMPLATE_FIRST	value	<port>	find first port in list with this template
LOCATE	PORT	<slot>	PORT_TEMPLATE_FIRSTNOTOPO	value	<port>	find first port in list with this template without topology
LOCATE	PORT	<slot>	PORT_TEMPLATE_LAST	value	<port>	find last port in list with this template
LOCATE	PORT	<slot>	PORT_TEMPLATE_LASTNOTOPO	value	<port>	find last port in list with this template without topology
SLOT object in other contexts						
DELETE	SLOT	<node>	SLOT_ID	value		remove ports based on slot-id, delete any attached topology

Server

The SERVER object uses the alias type `<server>`

SERVER						
exec	*class*	*scope*	*match*	*value*	*alias*	*description*
SERVER as new alias						
ADD	SERVER	<clnt>	SERVER_KEY	value	<server>	create server with server_key on client alias
ADD	SERVER	<sit>	SERVER_KEY	value	<server>	create server with server_key on site alias
LOCATE	SERVER	<sit>	SERVER_NAME	value	<server>	find server with server_name on site alias
LOCATE	SERVER	<sit>	SERVER_KEY	value	<server>	find server with server_key on site alias
LOCATE	SERVER	<clnt>	SERVER_NAME	value	<server>	find server with server_name on client alias
LOCATE	SERVER	<clnt>	SERVER_KEY	value	<server>	find server with server_key on client alias

SERVER						
SERVER as scope alias						
ASSIGN	SERVER	<server>	SERVER_NAME	value		set the server hostname to value
ASSIGN	SERVER	<server>	SERVER_STATUS	value		set the server status to value (numeric)
ASSIGN	SERVER	<server>	SERVER_NOTES	value		set the server description to value
ASSIGN	SERVER	<server>	CLIENT	<clnt>		assign the server to any client alias (move server)
ASSIGN	SERVER	<server>	SITE	<sit>		assign the server to any site alias (move server)
ASSIGN	SERVER	<server>	SERVER_ ADDRESS	value		set the server ipv4-address to value
ASSIGN	SERVER	<server>	SERVER_ IPV6_ADDR	value		set the server ipv6-address to value
DELETE	SERVER	<server>	COMPLETE			remove the server

Supernet

The SUPERNET object uses the alias type `<super>`

SUPERNET						
exec	***class***	***scope***	***match***	***value***	***alias***	***description***
SUPERNET as new alias						
ADD	SUPERNET	<clnt>	IP_PLAN	value	<super>	add new ip-supernet range to client from pool of free Supernets using numeric ip-plan ID. Or use API custom var “ip_supernet” to specify the supernet
ADD	SUPERNET	<clnt>	IP_SUPERNET	value	<super>	add new ip-supernet range to client. Format value as “<net-address>/<prefix>”. Set ip-plan using Assign-Supernet-Ip_plan or use API custom var “ip_plan”
LOCATE	SUPERNET	<clnt>	IP_PLAN	value	<super>	find the (first) supernet within the client matching the Ip_plan value
LOCATE	SUPERNET	<clnt>	IP_SUPERNET	value	<super>	find the supernet within the client matching the “<address>[/<prefix>]” value
SUPERNET as scope alias						
ASSIGN	SUPERNET	<super>	DNS_DOMAIN	value		assign the value to the supernet Dns_domain attribute
ASSIGN	SUPERNET	<super>	IP_PLAN	value		assign the numeric ip-plan ID to the supernet-alias. Supernet and plan prefixes must match. Cannot change a supernet plan-id which has active subnets
ASSIGN	SUPERNET	<super>	SITE_CODE	value		assign the value to the Supernet SiteCode attribute. SiteCode must exist in Client
DELETE	SUPERNET	<super>	COMPLETE			remove the supernet from the client after DELETING all its subnets. Place supernet in free pool

SUPERNET						
DELETE	SUPERNET	<super>	FREE			remove the supernet from the client IF there are no subnets left. Place supernet in free pool

Subnet

The SUBNET object uses the alias type `<net>`

SUBNET						
exec	***class***	***scope***	***match***	***value***	***alias***	***description***
SUBNET as new alias						
ADD	SUBNET	<srv>	CUSTOM	value	<net>	add new CUSTOM subnet. No IP-plan is used
ADD	SUBNET	<srv>	NET_NAME	value	<net>	locate and add subnet from an IP-plan by name
ADD	SUBNET	<srv>	NET_NAME_ NOTFULL	value	<net>	locate and conditionally add subnet from an IP-plan by name should the existing be 'full'
ADD	SUBNET	<srv>	NET_NAME_ AT_ADDRESS	value	<net>	create an ip-plan based subnet in a service using net-name AND subnet-address. Value format: “<net_name> <net_address>”
ADD	SUBNET	<srv>	NET_NAME_ FROM_ADDRESS	value	<net>	create an ip-plan based subnet in a service using net-name AND subnet-address. Finds the first free subnet. Value format: “<net_name> <net_address>”
ADD	SUBNET	<srv>	FIRSTFREE_ FROM_SUBNET	<net>	<net>	create an ip-plan based subnet using existing subnet-alias. Finds the first free subnet
ADD	SUBNET	<srv>	FIRSTFREE_ SUBNET_OFFSET	<net> <offset>	<net>	create an ip-plan based subnet using existing subnet-alias. Finds the first free subnet starting from the provided offset
LOCATE	SUBNET	<net>	AT_PORT	<port>	<net>	Locate the given subnet if assigned to the given port. Subnet and port are aliases

SUBNET						
LOCATE	SUBNET	SERVICE	CURRENT		<net>	find the current selected subnet in the service (tasks only)
LOCATE	SUBNET	<srv>	NET_ADDRESS	value	<net>	find the subnet in the service by network-address. Value may include the /prefix
LOCATE	SUBNET	<srv>	NET_DESCR	value	<net>	locate subnet by description in service
LOCATE	SUBNET	<srv>	NET_NAME	value	<net>	locate subnet by name *or* description in service
LOCATE	SUBNET	<srv>	NET_NAME_ NOTFULL	value	<net>	find and conditionally add subnet from an IP-plan by name should the existing be 'full'
LOCATE	SUBNET	<srv>	VLAN_ID	value	<net>	find the subnet by vlan-id in the service
SUBNET as scope alias						
LOCATE	SUBNET	<net>	AT_PORT	<port>	<net>	Locate the given subnet if assigned to the given port. Subnet and port are aliases
ASSIGN	SUBNET	<net>	-attribute-	value		assign the subnet attribute a value. replace -attribute- with actual name.
ASSIGN	SUBNET	<net>	ADDR_RESERVATI ON	value		set the subnets address reservation count to value
ASSIGN	SUBNET	<net>	ADDR_RESERVATION_ DEC	value		decrease the subnets address reservation count to value
ASSIGN	SUBNET	<net>	ADDR_RESERVATION_ INC	value		increase the subnets address reservation count to value
ASSIGN	SUBNET	<net>	LINK	<portlist>		make the ports of the link a member of the vlan/subnet
ASSIGN	SUBNET	<net>	MULTICAST	value		set multicast true/false for the subnet
ASSIGN	SUBNET	<net>	NET_DESCR	value		assign this description to the subnet
ASSIGN	SUBNET	<net>	PORT	<port>		make the port a member of the vlan/subnet
ASSIGN	SUBNET	<net>	PORTS	<portlist>		make the ports a member of the vlan/subnet
ASSIGN	SUBNET	<net>	SERVICE	<srv>		re-assign (move) the vlan/subnet to the service indicated

SUBNET						
ASSIGN	SUBNET	<net>	VLAN_ID	value		assign this vlan-id to the subnet
ASSIGN	SUBNET	<net>	VLAN_ID_ INCLIENT	value		find and assign the first free vlan-id in client starting at type-value
ASSIGN	SUBNET	<net>	VLAN_ID_ INSERVICE	value		find and assign the first free vlan-id in the service starting at type-value
ASSIGN	SUBNET	<net>	VLAN_ID_ INSITE	value		find and assign the first free vlan-id in site starting at type-value
ASSIGN	SUBNET	<net>	VLAN_TEMPLATE	value		Assigns the subnet a (layer3) Vlan-template
ASSIGN	SUBNET	<net>	VRF	<vrf>		make the vlan/subnet a member of the VRF
DELETE	SUBNET	<net>	COMPLETE			delete subnet, remove from all ports
DELETE	SUBNET	<net>	PORT	<port>		remove subnet from port
DELETE	SUBNET	<net>	PORTS	<portlist>		remove subnet from ports
DELETE	SUBNET	<net>	VRF	<vrf>		remove subnet from vrf
SUBNET as value alias						
ADD	SUBNET	<srv>	FIRSTFREE_ FROM_SUBNET	<net>	<net>	create an ip-plan based subnet using existing subnet-alias. Finds the first free subnet
ADD	SUBNET	<srv>	FIRSTFREE_ SUBNET_OFFSET	<net> <offset>	<net>	create an ip-plan based subnet using existing subnet-alias. Finds the first free subnet starting from the provided offset
ASSIGN	PORT	<port>	SUBNET	<net>		make the port a member of the vlan/subnet
ASSIGN	PORTS	<portlist>	SUBNET	<net>		make the ports a member of the vlan/subnet
ASSIGN	VRF	<vrf>	SUBNET	<net>		make the vlan/subnet a member of the VRF
DELETE	PORT	<port>	SUBNET	<net>		remove subnet from port
DELETE	PORTS	<portlist>	SUBNET	<net>		remove subnet from ports
DELETE	VRF	<vrf>	SUBNET	<net>		remove subnet from vrf
SUBNET alias in other contexts						
ADD	DHCP	<net>	TYPE	value	<dhcp>	create a dhcp data-set for a subnet. Type can be off / relay / scope / dual

SUBNET						
LOCATE	ADDRESS	<net>	ADDRESS_ FIRSTFREE	value	<addr>	find the first free address in a named range of a subnet
LOCATE	ADDRESS	<net>	ADDRESS_ FIRSTFREE	value	<addr> <offset>	find the first free address in a named range of a subnet starting from an offset
LOCATE	ADDRESS	<net>	ADDRESS_ LASTFREE	value	<addr>	find the last free address in a named range of a subnet
LOCATE	ADDRESS	<net>	PICK	value	<addr>	Select specific address in the subnet if free. Value may be offset or absolute. Negative offsets allowed
LOCATE	ADDRESS	<net>	PICK_FORCED	value	<addr>	Select specific address in the subnet. Value may be offset or absolute. Negative offsets allowed
LOCATE	DHCP	<net>	CURRENT		<dhcp>	find the dhcp data-set for the subnet
LOCATE	PORT	<port>	AT_SUBNET	<net>	<port>	Locate the given port if assigned to the given subnet. Port and subnet are aliases
LOCATE	PORTS	<portlist>	AT_SUBNET	<net>	<portlist>	Locate ports in the given list if assigned to the given subnet. Ports and subnet are aliases
ASSIGN	IPV6_NET	<ipv6net>	VLAN_ID_OF	<net>		assign the VLAN_ID of the IPV4-net to the IPV6_NET

Dhcp

The DHCP object uses the alias type <dhcp>

DHCP						
exec	*class*	*scope*	*match*	*value*	*alias*	*description*
DHCP as new alias						
ADD	DHCP	<net>	TYPE	value	<dhcp>	create a dhcp data-set for a subnet. Type can be off / relay / scope / dual
LOCATE	DHCP	<net>	CURRENT		<dhcp>	find the dhcp data-set for the subnet
DHCP as scope alias						
ASSIGN	DHCP	<dhcp>	DHCP_DOMAIN	value		define the Dhcp domain name option value
ASSIGN	DHCP	<dhcp>	DNS_SERVERS	value		define the Dns-servers option value. Enter a list of ip-addresses
ASSIGN	DHCP	<dhcp>	LEASE_TIME	value		define the Lease-time option value. Normally in minutes

DHCP						
ASSIGN	DHCP	<dhcp>	NTP_SERVERS	value		define the Ntp-servers option value. Enter a list of ip-addresses
ASSIGN	DHCP	<dhcp>	OPTION1	value		define the spare option1 value.
ASSIGN	DHCP	<dhcp>	OPTION2	value		define the spare option2 value.
ASSIGN	DHCP	<dhcp>	RELAY_GROUP	value		define the Dhcp-relay-group option value.
ASSIGN	DHCP	<dhcp>	RELAY_SERVERS	value		define the Dhcp-relay-servers option value. Enter a list of ip-addresses
ASSIGN	DHCP	<dhcp>	SCOPE	value		specify scope-start and scope-end as ip-offsets (0.0.0.50) or numeric-offsets. Scope-end is optional. Separate using space, comma or ' - '
ASSIGN	DHCP	<dhcp>	SCOPE_ EXCLUDED	value		list of ip-ddresses in the scope to exclude. Separate using space, comma or ' - '
ASSIGN	DHCP	<dhcp>	TFTP_SERVERS	value		define the Tftp-servers option value. Enter a list of ip-addresses
ASSIGN	DHCP	<dhcp>	TYPE	value		type can be off / relay / scope / dual
ASSIGN	DHCP	<dhcp>	WINS_SERVERS	value		define the Wins-servers option value. Enter a list of ip-addresses
DELETE	DHCP	<dhcp>	COMPLETE			remove the Dhcp data-set from the subnet

Address

The ADDRESS object uses the alias type <addr>

ADDRESS						
exec	*class*	*scope*	*match*	*value*	*alias*	*description*
ADDRESS as new alias						
LOCATE	ADDRESS	<net>	ADDRESS_ FIRSTFREE	value	<addr>	find the first free address in a named range of a subnet
LOCATE	ADDRESS	<net>	ADDRESS_ FIRSTFREE	value	<addr> <offset>	find the first free address in a named range of a subnet starting from an offset
LOCATE	ADDRESS	<net>	ADDRESS_ LASTFREE	value	<addr>	find the last free address in a named range of a subnet
LOCATE	ADDRESS	<net>	PICK	value	<addr>	Select specific address in the subnet if free. Value may be offset or absolute. Negative offsets allowed
LOCATE	ADDRESS	<net>	PICK_FORCED	value	<addr>	Select specific address in the subnet. Value may be offset or absolute. Negative offsets allowed
LOCATE	ADDRESS	<node>	MANAGEMENT		<addr>	Returns the management ip-address for the <node> as an <addr> alias
ADDRESS as scope alias						
ASSIGN	ADDRESS	<addr>	PORT	<port>		set a single IP-address on a Management or Loopback interface

Ipv6_net

The IPV6_NET object uses the alias type <ipv6net>

IPV6_NET						
exec	*class*	*scope*	*match*	*value*	*alias*	*description*
IPV6_NET as new alias						
ADD	IPV6_NET	<srv>	CUSTOM	value	<ipv6net>	add new CUSTOM Ipv6-subnet. No IP-plan is used. Value is the net_name, the ipv6 address range must be assigned later
ADD	IPV6_NET	<srv>	NET_NAME	value	<ipv6net>	add an IPV6_NET by name. Value can be “ipv6net_name” or “ipv6_net_name - ipv6_plan_descr”
LOCATE	IPV6_NET	<srv>	NET_ADDRESS	value	<ipv6net>	find the IPV6_NET in the service using its Ipv6 network-address. Use the optional prefix to normalize the network-address: <Ipv6_address>/<prefix>
LOCATE	IPV6_NET	<srv>	VLAN_ID	value	<ipv6net>	find the IPV6_NET in the service using its Vlan_id. The subnet with the fewest assigned ports is selected.
LOCATE	IPV6_NET	<srv>	NET_DESCR	value	<ipv6net>	find the IPV6_NET in the service using its Net_description or Net_name. The subnet with the fewest assigned ports is selected.
LOCATE	IPV6_NET	<srv>	NET_NAME	value	<ipv6net>	find the IPV6_NET in the service using its Net_name. The subnet with the fewest assigned ports is selected.
LOCATE	IPV6_NET	SERVICE	CURRENT		<ipv6net>	find the current selected IPV6_NET in the service (GUI tasks only)
LOCATE	IPV6_NET	<srv>	IPV6_NET	value	<ipv6net>	find the IPV6_NET in the service using Net_name and Ipv6-plan. Type_value uses the format: <ipv6_subnet_name>-<ipv6_plan_description>
LOCATE	IPV6_NET	<ipv6net>	AT_PORT	<port>	<ipv6net>	Locate the given Ipv6 subnet if assigned to the given port. Ipv6 subnet and port are aliases
IPV6_NET as scope alias						
LOCATE	IPV6_NET	<ipv6net>	AT_PORT	<port>	<ipv6net>	Locate the given Ipv6 subnet if assigned to the given port. Ipv6 subnet and port are aliases
ASSIGN	IPV6_NET	<ipv6net>	PORTS	<portlist>		make the ports a member of the Ipv6 vlan/subnet
ASSIGN	IPV6_NET	<ipv6net>	NET_DESCR	value		change the Ipv6 subnet description to “value”. Value defaults to Ipv6_subnet_name
ASSIGN	IPV6_NET	<ipv6net>	NET_DESCR_OF	<ipv6net>		change the Ipv6 subnet description to the description found in subnet-alias of value
ASSIGN	IPV6_NET	<ipv6net>	PORT	<port>		make the port a member of the Ipv6 vlan/subnet
ASSIGN	IPV6_NET	<ipv6net>	VLAN_ID_OF	<net>		assign the VLAN_ID of the IPV4-net to the IPV6_NET
ASSIGN	IPV6_NET	<ipv6net>	VLAN_ID	value		assign this vlan-id to the IPV6_NET
ASSIGN	IPV6_NET	<ipv6net>	VLAN_ID_ INCLIENT	value		find and assign the first free vlan-id in client starting at type-value
ASSIGN	IPV6_NET	<ipv6net>	VLAN_ID_ INSERVICE	value		find and assign the first free vlan-id in the service starting at type-value
ASSIGN	IPV6_NET	<ipv6net>	VLAN_ID_ INSITE	value		find and assign the first free vlan-id in site starting at type-value
ASSIGN	IPV6_NET	<ipv6net>	VLAN_TEMPLATE	value		Assigns the IPV6_NET a (layer3) Vlan-template
ASSIGN	IPV6_NET	<ipv6net>	VRF	<vrf>		
ASSIGN	IPV6_NET	<ipv6net>	-attribute-	value		assign an Ipv6 subnet attribute a value. replace -attribute- with actual name.

IPV6_NET						
ASSIGN	IPV6_NET	<ipv6net>	LINK	<portlist>		make the ports of the link a member of the Ipv6 vlan/subnet
DELETE	IPV6_NET	<ipv6net>	COMPLETE			delete IPV6_NET, remove from all ports
DELETE	IPV6_NET	<ipv6net>	PORT	<port>		remove IPV6_NET from port
DELETE	IPV6_NET	<ipv6net>	PORTS	<portlist>		remove IPV6_NET from ports
DELETE	IPV6_NET	<ipv6net>	VRF	<vrf>		remove IPV6_NET from vrf
IPV6_NET as value alias						
ASSIGN	IPV6_NET	<ipv6net>	NET_DESCR_OF	<ipv6net>		change the Ipv6 subnet description to the description found in subnet-alias of value
ASSIGN	PORT	<port>	IPV6_NET	<ipv6net>		make the port a member of the Ipv6 vlan/subnet
ASSIGN	PORTS	<portlist>	IPV6_NET	<ipv6net>		make the ports a member of the Ipv6 vlan/subnet
ASSIGN	LINK	<portlist>	IPV6_NET	<ipv6net>		make the ports of the link a member of the Ipv6 vlan/subnet
IPV6_NET alias in other contexts						
LOCATE	IPV6_ADDR	<ipv6net>	IPV6_ADDR_FIRSTFREE	value	<ipv6addr>	find the first free IPV6_ADDR in a named range of a IPV6_NET
LOCATE	IPV6_ADDR	<ipv6net>	IPV6_ADDR_LASTFREE	value	<ipv6addr>	find the last free IPV6_ADDR in a named range of a IPV6_NET
LOCATE	IPV6_ADDR	<ipv6net>	RANDOM	value	<ipv6addr>	find the first free IPV6_ADDR in a named range of a IPV6_NET
LOCATE	PORT	<port>	AT_IPV6_NET	<ipv6net>	<port>	Locate the given port if assigned to the given Ipv6 subnet. Port and Ipv6 subnet are aliases
LOCATE	PORTS	<portlist>	AT_IPV6_NET	<ipv6net>	<portlist>	Locate ports in the given list if assigned to the given Ipv6 subnet. Ports and Ipv6 subnet are aliases

Ipv6_addr

The IPV6_ADDR object uses the alias type <ipv6addr>

IPV6_ADDR						
exec	*class*	*scope*	*match*	*value*	*alias*	*description*
IPV6_ADDR as new alias						
LOCATE	IPV6_ADDR	<ipv6net>	IPV6_ADDR_FIRSTFREE	value	<ipv6addr>	find the first free IPV6_ADDR in a named range of a IPV6_NET
LOCATE	IPV6_ADDR	<ipv6net>	IPV6_ADDR_LASTFREE	value	<ipv6addr>	find the last free IPV6_ADDR in a named range of a IPV6_NET
LOCATE	IPV6_ADDR	<ipv6net>	RANDOM	value	<ipv6addr>	find the first free IPV6_ADDR in a named range of a IPV6_NET
IPV6_ADDR as scope alias						
ASSIGN	IPV6_ADDR	<ipv6addr>	PORT	<port>		set a single IP-IPV6_ADDR on a Management or Loopback interface

Mpls_vrf

The MPLS_VRF object uses the alias type <mpls_vrf>

MPLS_VRF						
exec	*class*	*scope*	*match*	*value*	*alias*	*description*
MPLS_VRF as new alias						
ADD	MPLS_VRF	CURRENT	VRF_ID	value	mpls_vrf	create or locate new mpls-vrf for this client_type. value is new vrf_id
ADD	MPLS_VRF	CURRENT	VRF_NAME	value	mpls_vrf	create or locate new mpls-vrf for this client_type. value is new vrf_name
ADD	MPLS_VRF	GLOBAL	VRF_ID_NEW	value	mpls_vrf	create new mpls-vrf for this client_type. value is lowest vrf-id to start search for new id
LOCATE	MPLS_VRF	CURRENT	VRF_NAME	value	mpls_vrf	locate mpls-vrf for this client_type. value is new vrf_name
LOCATE	MPLS_VRF	CURRENT	VRF_ID	value	mpls_vrf	locate mpls-vrf for this client_type. value is new vrf_id
MPLS_VRF as scope alias						
ASSIGN	MPLS_VRF	mpls_vrf	-attribute-	value		set the mpls-vrf attribute to the desired value
DELETE	MPLS_VRF	mpls_vrf	COMPLETE			delete mpls_vrf including its node assignments
MPLS_VRF as value alias						
ADD	VRF	<node>	MPLS_VRF	mpls_vrf	<vrf>	create vrf on node using a Mpls_vrf alias. Note: node must have management / loopback address assigned first!

Vrf

The VRF object uses the alias type `<vrf>`

VRF						
exec	*class*	*scope*	*match*	*value*	*alias*	*description*
VRF as new alias						
ADD	VRF	<node>	VRF_ID	value	<vrf>	create vrf on node using its id. Note: node must have management / loopback address assigned first!
ADD	VRF	<node>	VRF_NAME	value	<vrf>	create vrf on node using its name. Note: node must have management / loopback address assigned first!
ADD	VRF	<node>	MPLS_VRF	mpls_vrf	<vrf>	create vrf on node using a Mpls_vrf alias. Note: node must have management / loopback address assigned first!
LOCATE	VRF	SITE	VRF_ID	value	<vrf>	find a vrf on the site using its id
LOCATE	VRF	<node>	VRF_NAME	value	<vrf>	find a vrf on the node using its name
LOCATE	VRF	<node>	VRF_ID	value	<vrf>	find a vrf on the node using its id
LOCATE	VRF	SITE	VRF_NAME	value	<vrf>	find a vrf on the site using its name
VRF as scope alias						

VRF						
ASSIGN	VRF	<vrf>	-attribute-	value		Set the Node_vrf attribute given tin Type_match to the value specified. Vrf_name cannot be changed.
ASSIGN	VRF	<vrf>	NODE	<node>		make node a member of the MPLS vrf specified
ASSIGN	VRF	<vrf>	SUBNET	<net>		make the vlan/subnet a member of the VRF
ASSIGN	VRF	<vrf>	VRF_TEMPLATE	value		Value is Vrf template name
DELETE	VRF	<vrf>	NODE	<node>		remove node from vrf
DELETE	VRF	<vrf>	SUBNET	<net>		remove subnet from vrf
VRF as value alias						
ASSIGN	IPV6_NET	<ipv6net>	VRF	<vrf>		
ASSIGN	NODE	<node>	VRF	<vrf>		make node a member of the MPLS vrf specified
ASSIGN	SUBNET	<net>	VRF	<vrf>		make the vlan/subnet a member of the VRF
DELETE	IPV6_NET	<ipv6net>	VRF	<vrf>		remove IPV6_NET from vrf
DELETE	NODE	<node>	VRF	<vrf>		remove node from vrf
DELETE	SUBNET	<net>	VRF	<vrf>		remove subnet from vrf

Cmdb

The CMDB object uses the alias type <cmdb>

CMDB						
exec	***class***	***scope***	***match***	***value***	***alias***	***description***
CMDB as new alias						
ADD	CMDB	<dom>	NODE	<node>	<cmdb>	Add a node to the CMDB using Domain alias. value = node-alias, nodename or fqdn
ADD	CMDB	<dom>	NODE	value	<cmdb>	Add a node to the CMDB using Domain alias. value = node-alias, nodename or fqdn
LOCATE	CMDB	CMDB	NODE	value	<cmdb>	Find a node in the CMDB by node name. value = node-alias, nodename or fqdn
LOCATE	CMDB	CMDB	NODE	<node>	<cmdb>	Find a node in the CMDB by node name. value = node-alias, nodename or fqdn
CMDB as scope alias						
LOCATE	CMDB	CMDB	NODE	value	<cmdb>	Find a node in the CMDB by node name. value = node-alias, nodename or fqdn
LOCATE	CMDB	CMDB	NODE	<node>	<cmdb>	Find a node in the CMDB by node name. value = node-alias, nodename or fqdn
ASSIGN	CMDB	<cmdb>	VENDOR	value		Assign a Vendor to a cmdb-alias/node-name. value = Vendor_type name
ASSIGN	CMDB	<cmdb>	FQDN	value		Assign a fqdn (NodeName) to a cmdb node alias. value = fqdn or ip-address
ASSIGN	CMDB	<cmdb>	DOMAIN	value		Assign a Domain to a cmdb-node/node-name. value = Domain

CMDB						
ASSIGN	CMDB	<cmdb>	-attribute-	value		Assign a value to any attribute of a cmdb-node/node-name. Replace -attribute for attribute name. value = attribute value
DELETE	CMDB	CMDB	NODE	<node>		Delete a node from the CMDB by node name. value = node-alias, nodename or fqdn
DELETE	CMDB	CMDB	NODE	value		Delete a node from the CMDB by node name. value = node-alias, nodename or fqdn
CMDB object in other contexts						
ASSIGN	CMDB	<node>	VENDOR	value		Assign a Vendor to a cmdb-alias/node-name. value = Vendor_type name
ASSIGN	CMDB	<node>	FQDN	value		Assign a fqdn (NodeName) to a cmdb-node/node-name. value = fqdn or ip-address
ASSIGN	CMDB	<node>	DOMAIN	value		Assign a Domain to a cmdb-node/node-name. value = Domain
ASSIGN	CMDB	<node>	-attribute-	value		Assign a value to any attribute of a cmdb-node/node-name. Replace -attribute for attribute name. value = attribute value