

# Table of Contents

- Service-types** ..... 1
- Service-types form** ..... 1
- Service-types section ..... 2
- Adding or duplicating a service type ..... 2
- Deleting a service type ..... 2
- Finding a service type ..... 2
- Rows ..... 2
- Adding a row ..... 2
- Deleting a row ..... 3
- Resequence rows ..... 3
- Details of a row ..... 3
- Alias ..... 3
- Syntax** ..... 3
- Example ..... 4
- Node and information structure** ..... 4



# Service-types

Service-types are basically scripted tasks which perform an automated task on the network in the database. These tasks can be:

1. Add/remove devices to the network
2. Add/remove topology
3. Add/remove VLAN/IP-addresses
4. Add/remove/enable/disable interfaces
5. Add/remove/modify services
6. etc...

The service types can be executed through the NetYCE API interfaces or from the netYCE web-interface. Service types can create, modify or delete database objects, which can be located and used.

Please consult the article [Service-types overview](#) for an introduction to Service-types  
 A complete reference to all service type commands is listed in [Service-type syntax](#).

## Service-types form

The Service-types form is divided in 3 sections:

1. Service types
2. Rows within a selected service type
3. Details and editing of a selected row

Service types

Client type	Service class	Service type	Service task	Records
C3	C3_main_sans	C3_Cars_poi50/96	Creates	12
FP	Access Leaf	onboarding	assignInterface	8
FP	F1F2 Access Leaf	F1F2 Access Leaf	AssignNet Nexus 2232	108
FP	F1F2 Access Leaf	F1F2 Access Leaf	AssignNet Nexus 2240	66
FP	F1F2 Access Leaf	F1F2 Access Leaf	Add Po == 1000 Access	7
FP	F1F2 Access Leaf	F1F2 Access Leaf	Add Po == 1000 Trunk	7
FP	F1F2 Access Leaf	F1F2 Access Leaf	DELETE Access Leaf	22

Buttons: [New](#) [Duplicate](#) [Delete](#) [Search](#) [Refresh](#) [Export](#)

Seq	Exec	Class	Scope	Match	Value	Alias
1	LOCATE	SERVICE	SERVICE	CLIPRENT		<service>
2	LOCATE	NODE	<service>	NODE_POSITION	2A	<node>
3	LOCATE	NODE	<service>	NODE_POSITION	2B	<node>
4	ADD	PORT	<node>	PORT_CHANNEL_NEW	1000	<node.poi1000>
5	ADD	PORT	<node>	PORT_CHANNEL_NEW	1000	<node.poi1000>
6	ASSIGN	PORT	<node.poi1000>	PORT_TEMPLATE	FP_C3_Access_po	
7	ASSIGN	PORT	<node.poi1000>	PORT_TEMPLATE	FP_C3_Access_po	

Buttons: [New](#) [Delete](#) [Up](#) [Down](#) [Search](#)

Seq:  Exec:  Class:


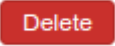
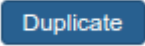
Scope:  Match:  Value:  Alias:

Note:

Description:

[Save](#)

## Service-types section


Within the Service types section, the user has an overview of the service types which belong to the Client types to which the user has access. Here a new service type can be added , an existing service type can be deleted  or duplicated .

### Adding or duplicating a service type

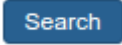
When adding or duplicating a service type the following data must be entered:

- Client type (drop down menu)
- Service class (drop down menu). This list is build from the available service classes of the client type.
- The name of the service type. This is chosen by the user.
- The name of the service task. Create is a reserved value

### Deleting a service type

When you want to delete a service type, simply select the service type to be deleted and then click the Delete  button and the OK button.






### Finding a service type

It is possible to search within the service types. Click the Search button  and type the selection criteria in the search bar. Press **[enter]** to commit.


## Rows

Within the second part of the window, the components of the selected service type are displayed.


Seq	Exec	Class	Scope	Match	Value	Alias
1	LOCATE	SERVICE	CLIENT	SERVICE_CLASS	UCL_core	<core.srv>
2	LOCATE	SERVICE	SITE	SERVICE_CLASS	UCL_aggr	<aggr.srv>
3	LOCATE	SERVICE	SERVICE	CURRENT		<acc.srv>
4	LOCATE	NODE	<acc.srv>	NODE_POSITION	NA	<node.na>
5	LOCATE	NODE	<acc.srv>	NODE_POSITION	NB	<node.nb>
6	LOCATE	NODE	<acc.srv>	NODE_POSITION	ZA	<node.za>
7	LOCATE	NODE	<acc.srv>	NODE_POSITION	ZB	<node.zb>



### Adding a row

When adding a new row, the new row will be added after the current selected row. To add a row click the  button below the section. A new row will be added.

## Deleting a row

In order to delete a row, click the  icon below the section. You will be asked to confirm, click *OK* to delete the row.

## Resequence rows

With the  and  buttons resequencing of the rows is possible. The aliases will not be updated!

## Details of a row

<b>Seq</b>	<input type="text" value="3"/>	<b>Exec:</b>	<input type="text" value="LOCATE"/>	<b>Class:</b>	<input type="text" value="SERVICE"/>
<b>Scope:</b>	<input type="text" value="SERVICE"/>	<b>Match:</b>	<input type="text" value="CURRENT"/>	<b>Value:</b>	<input type="text"/>
				<b>Alias</b>	<input type="text" value="&lt;acc.srv&gt;"/>
<b>Note</b>	<input type="text" value="find the current selected service (tasks only)"/>				
<b>Description</b>	<input type="text"/>				
	<input type="text"/>				

When selecting a row, the details will be displayed in the third section of the form. After each updated value, the next field will be updated to ensure the accurate syntax. The description field can be used as comment. It is also possible to add an empty row and use this as a comment row. In this case only the description field will be set.

## Alias

The **alias** is being used only internally within a Service type script to use the result of one row as a variable in another row. It is advisable to use distinctive names for aliases as well as surround them by `<hooks>` to identify them visually as such.

Please refer to [Service-types overview](#) for some guidelines on Alias names.

## Syntax

The definition of a Service type is done with a specific syntax, which is guided by the visual editor. The **Add** and **Locate** can be assigned **aliases**, to be used as a reference to the object in the commands following it.

In general, everything written in UPPERCASE are syntax keywords, everything written in lowercase are **aliases** or **values**.

The article on [Service type syntax](#) offers a summary of the service-type commands available. They are grouped around the various object types that NetYCE supports.

### Example

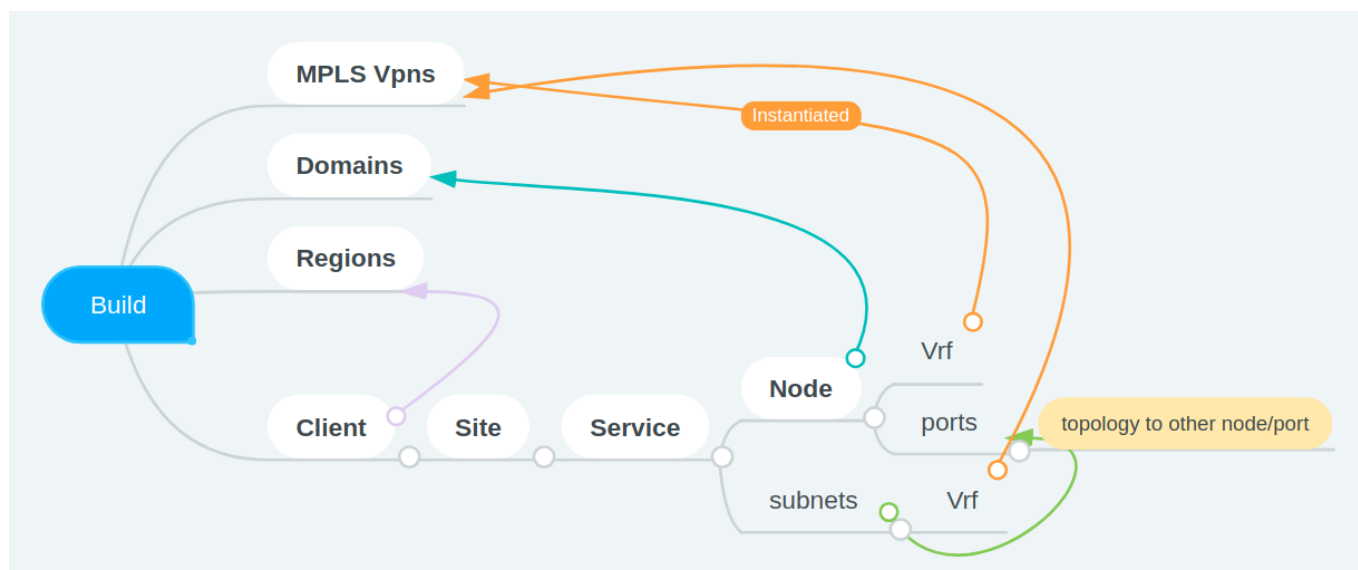
The following example will add a node to a newly created service with a IP address assigned to interface Loopback0.

Sequence	Exec	Class	Scope	Match	Value	Alias
1	ADD	SERVICE	SITE	CURRENT		<srv>
2	ADD	NODE	<srv>	NODE_TYPE	LAN_ACCESS	<node>
3	ADD	SUBNET	<srv>	NET_NAME	Loopback	
4	LOCATE	ADDRESS	ADDRESS_FIRSTFREE	Lo0	<ip>	
5	LOCATE	PORT	<node>	PORT_TEMPLATE_FIRST	Loopback0	<port>
6	ASSIGN	ADDRESS	<ip>	PORT	<port>	

A whole set of example Service types is included in every distribution. You can copy and adapt them to suit your own architecture.

## Node and information structure

The below diagram shows how some of the database information for Nodes, Services, subnets, etc. is structured. This could help you design your service types and help you understand how each of the pieces tie together.



From:

<http://wiki.netyce.com/> - **Knowledge base**

Permanent link:

<http://wiki.netyce.com/doku.php/menu:design:servicetypes>

Last update: **2020/06/04 12:12**

