Table of Contents

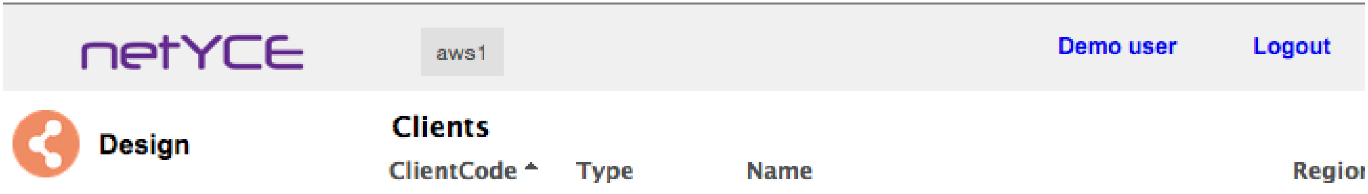
Sample Client_type setup

This article is intended as a “walk-thru” to get familiar with the initial steps of the NetYCE network modelling. It provides a step-by-step, click-by-click guide to create a new client-type an define site-types, service-classes and node-types for it.

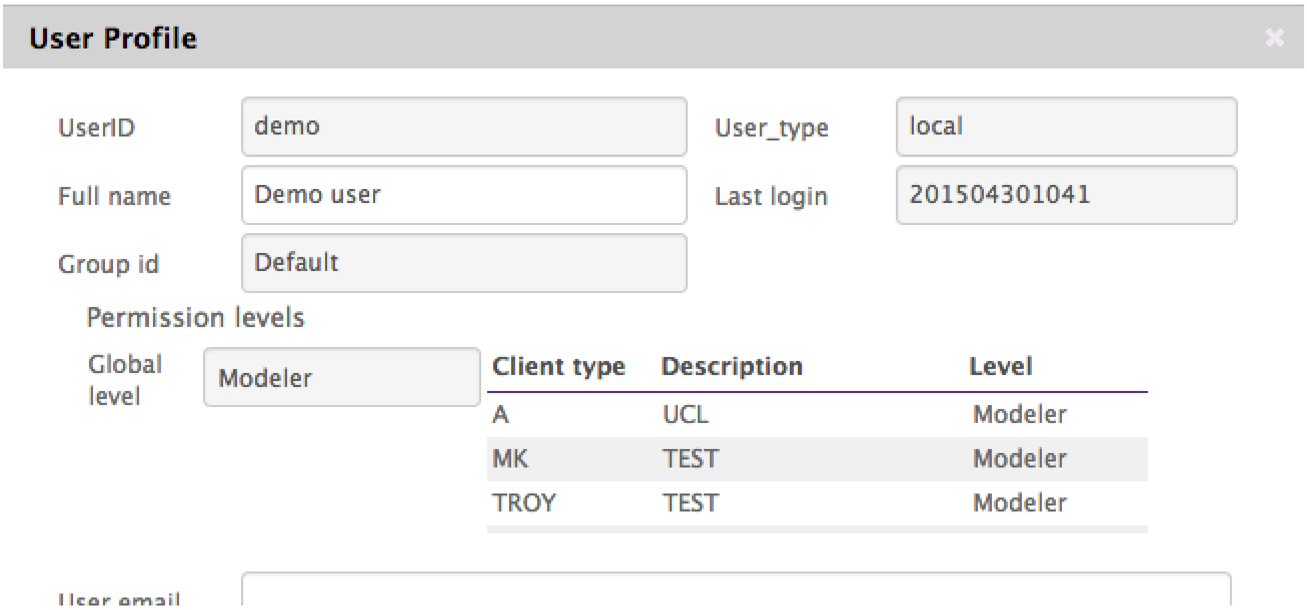
After each step in the modelling, the corresponding “build” phase equivalents are tried, in order to get familiar with the effects of the modelling on the realized network.

PREPARATION

Before starting out, ensure the required permissions are available to your account. After logging in, select your name from the header (in this case “Demo user”):



and review your profile for the appropriate permissions:



UserID	demo	User_type	local
Full name	Demo user	Last login	201504301041
Group id	Default		

Permission levels

Global level: Modeler

Client type	Description	Level
A	UCL	Modeler
MK	TEST	Modeler
TROY	TEST	Modeler

User email

`Modeler` Global level permissions are minimal. If `Manager` level is listed, you can also create users and modify permissions.

The Client-types and the permissions you have in them are listed in the table. Unlisted client-types are invisible to you. This guide expects you to have access to the “A” client-type (“UCL”).

In case you are assigned “Manager” permissions, you might wish to reduce the number of visible client-types and corresponding clients as outline below. Otherwise, skip to the next section.

Reduce visible Client-types

When setting up a new client_type, it is advisable to limit the number of shown client_types and clients. With too many (visible) client_types, locating the correct entries will become harder.
To modify select `Admin - Users`. From the `Groups` tab, select the the group of your user-id and remove the client_types from that group using the `Assigned Permissions` list and the > button.

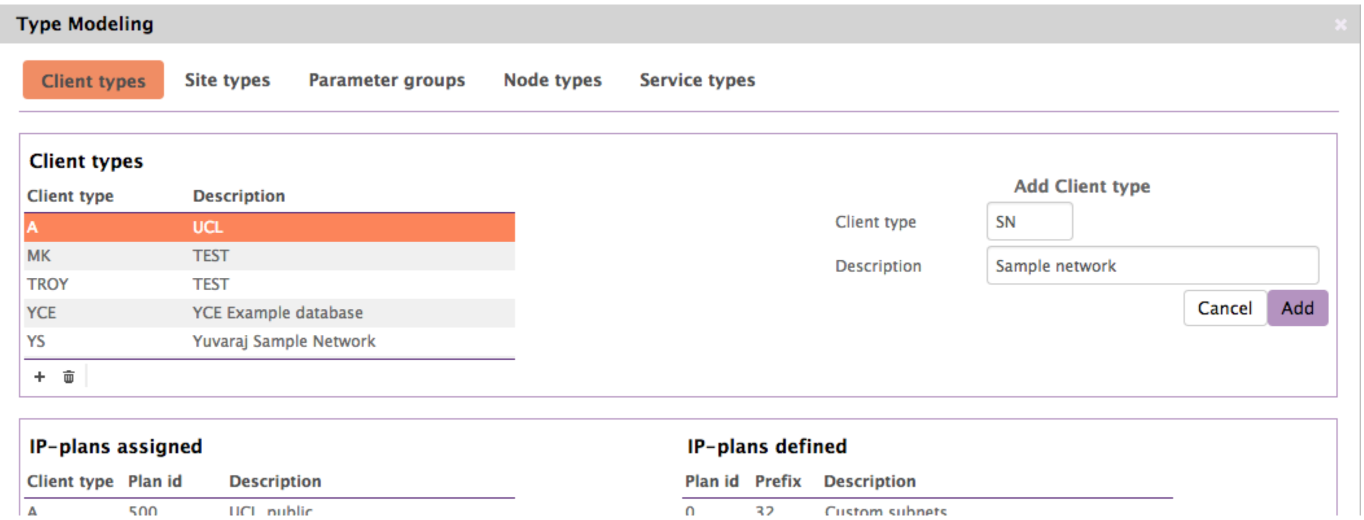
Adding or altering permissions for client_types uses the reverse selection, but make sure to select the appropriate permission level before clicking the < button.

I can be useful to leave at least ONE sample client_type (eg 'A') to compare results or setup with. I will use the 'A' (UCL) client_type as a reference in this article.

CLIENT_TYPE

Create new Client-type

Open the Modeling form: `Design - Modeling'`
Select the `Client_types` tab



Client types

Client type	Description
A	UCL
MK	TEST
TROY	TEST
YCE	YCE Example database
YS	Yuvaraj Sample Network

Add Client type

Client type	SN
Description	Sample network

IP-plans assigned

Client type	Plan id	Description
A	500	UCL public

IP-plans defined

Plan id	Prefix	Description
0	32	Custom subnets

Click the **+** icon under the Client types list

Choose an abbreviation for the new client_type.
This abbreviation will be referenced extensively since it denotes the network type or network architecture where customer networks ('Clients') are part of. Two or three capital letters are recommended.
I will use 'SN' in the remainder of this article. Type **SN** in the Client type box, and **`Sample Network`** in the Description box.
Click the 'Add' button.

Allow user-groups access to the Client type

Observe that your current group name is already added to the `Groups assigned` list. You can assign more groups from the `Groups available` list by selecting them and clicking the corresponding < button. These user groups will be allowed access to the "SN" models and the clients using it.

Select IP-plans for the Client type

Each client_type can be assigned multiple IP-plans. These IP-plans can be shared between client_types since they only describe how an IP-supernet allocated to a client will be sliced up in subnets. The Ip-plan will be defined in terms of IP-offsets (eg 0.0.0.0 and 0.0.2.0) and will be used as masks that overly the actual IP-range of the supernet.

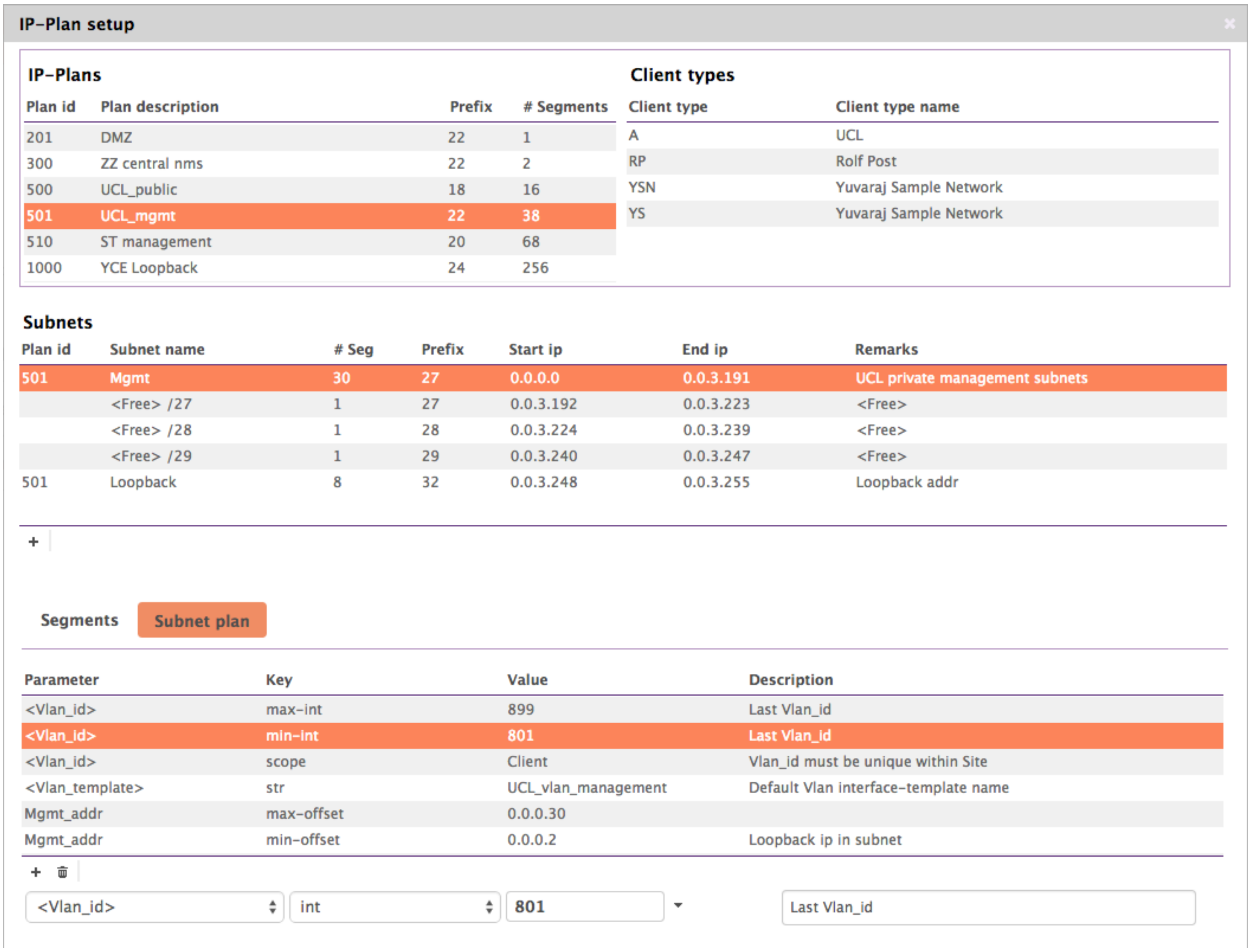
Many ip-plans can be made available to a client_type, depending on the desired IP-space allocation. If no ip-plans are assigned, all subnets created in the model will be have to be `Custom` subnets, meaning ‘manual’. When using IP-plans, the IP-subnet allocation and IP-address selections can be fully automated.

You can create a new IP-plan using the **+** in the corresponding IP-plan list, but let's skip this for now and review an existing one that we will use in this article: **`UCL_mgmt`** or plan_id `501`.
Scroll down in the IP-plan list and select the `501 - 22 - UCL_mgmt` entry. Then add to the SN client type using the **<** .

Review an existing IP-plan

Open the ip-plan setup form of this ip-plan by double-click on the entry (in the right-hand list, not the left-handed one).

The `UCL_mgmt` ip-plan maps onto a /22 supernet and has definitions for 38 subnets. The **`Subnets`** tab shows 30 segments are defined for `Mgmt` purposes, each using a /27. Also 8 subnets are defined for `Loopback` (/32 each of course). Some room for future allocations is present, denoted by the ‘<Free> subnets.



IP-Plan setup

IP-Plans

Plan id	Plan description	Prefix	# Segments
201	DMZ	22	1
300	ZZ central nms	22	2
500	UCL_public	18	16
501	UCL_mgmt	22	38
510	ST management	20	68
1000	YCE Loopback	24	256

Client types

Client type	Client type name
A	UCL
RP	Rolf Post
YSN	Yuvaraj Sample Network
YS	Yuvaraj Sample Network

Subnets

Plan id	Subnet name	# Seg	Prefix	Start ip	End ip	Remarks
501	Mgmt	30	27	0.0.0.0	0.0.3.191	UCL private management subnets
	<Free> /27	1	27	0.0.3.192	0.0.3.223	<Free>
	<Free> /28	1	28	0.0.3.224	0.0.3.239	<Free>
	<Free> /29	1	29	0.0.3.240	0.0.3.247	<Free>
501	Loopback	8	32	0.0.3.248	0.0.3.255	Loopback addr

Parameter	Key	Value	Description
<Vlan_id>	max-int	899	Last Vlan_id
<Vlan_id>	min-int	801	Last Vlan_id
<Vlan_id>	scope	Client	Vlan_id must be unique within Site
<Vlan_template>	str	UCL_vlan_management	Default Vlan interface-template name
Mgmt_addr	max-offset	0.0.0.30	
Mgmt_addr	min-offset	0.0.0.2	Loopback ip in subnet

<Vlan_id> | int | 801 | Last Vlan_id

Select the **Mgmt** subnet type and open the **Subnet plan** tab. The Subnet plan refers to a set of rules for the subnet(s) of this type. These rules include:

- rules for automatic ip-address calculation for later reference in the configuration templates;
- rules to locate a free vlan-id;
- rules to define a named ip-range to look for free addresses.

The named range **Mgmt_addr** will be used later in our example model.

Just take note but make no changes. Close the ip-plan setup form to get back to the Modeling form. Then close that as well. If a 'discard changes' popup shows up, ignore it and confirm.

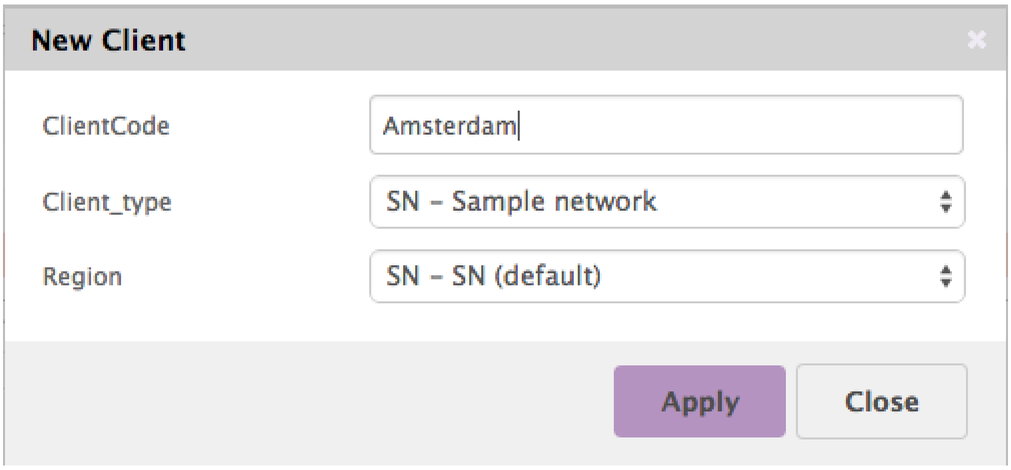
CLIENT

Although only the very minimal modelling is done so far, a new client of the new type can already be added. Select the '**Build**' menu to bring up the list of clients, their locations and devices.

All (customer) client networks are created and maintained to using the rules and definitions of the client-type they are created with. Many different client can be created using the same client-type.

In the '**Client**' list, click the **+** button.
A 'New Client' popup appears. First select the client_type: 'SN - Sample Network'
The automatically created Region 'SN - SN (default)' is already selected. Later additional regions can

be defined should that be required. Normally it isn't.

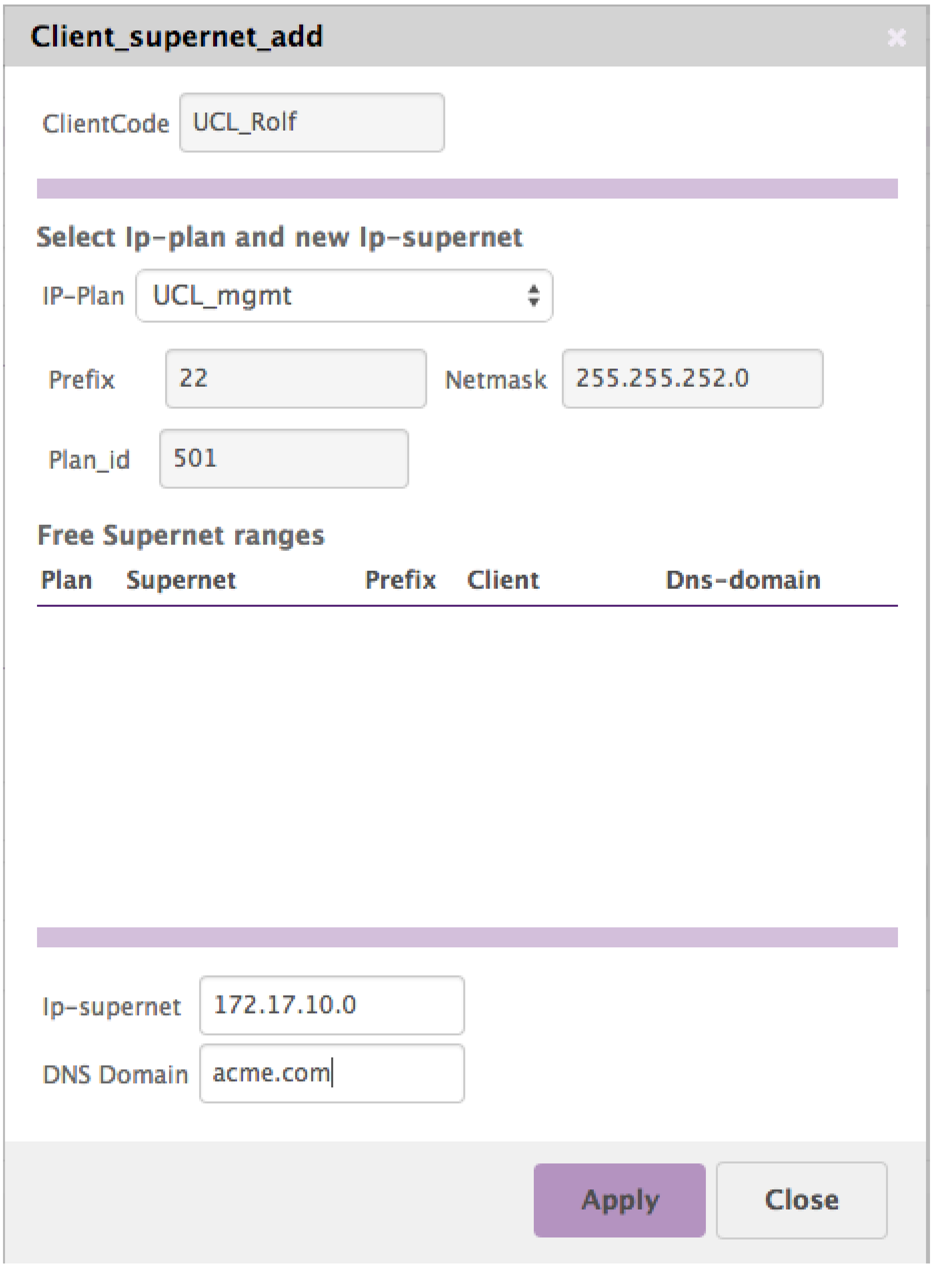


New Client

ClientCode	Amsterdam
Client_type	SN - Sample network
Region	SN - SN (default)

Now type the new Client code, the short reference name or number of this client. Choose **`Amsterdam`** and click `apply`.
The 'Client Details' forms opens automatically showing some attributes of this client. (Later this form is opened by double clicking the client in the 'Clients' list.)

Since we want to use an ip-plan for this client, we need to assign it a supernet. Click on the **+** under the '**Supernet**' table in this form.



Client_supernet_add

ClientCode: UCL_Rolf

Select Ip-plan and new Ip-supernet

IP-Plan: UCL_mgmt

Prefix: 22 Netmask: 255.255.252.0

Plan_id: 501

Free Supernet ranges

Plan	Supernet	Prefix	Client	Dns-domain

Ip-supernet: 172.17.10.0

DNS Domain: acme.com

Select the desired ip-plan from the ip-plan menu: **`UCL_mgmt`**, specifying a /22 prefix. The 'Free ranges' list shows available supernets matching the /22 - if any. These are supernets used earlier and freed or entries populated by an IPAM integration with NetYCE that was setup.

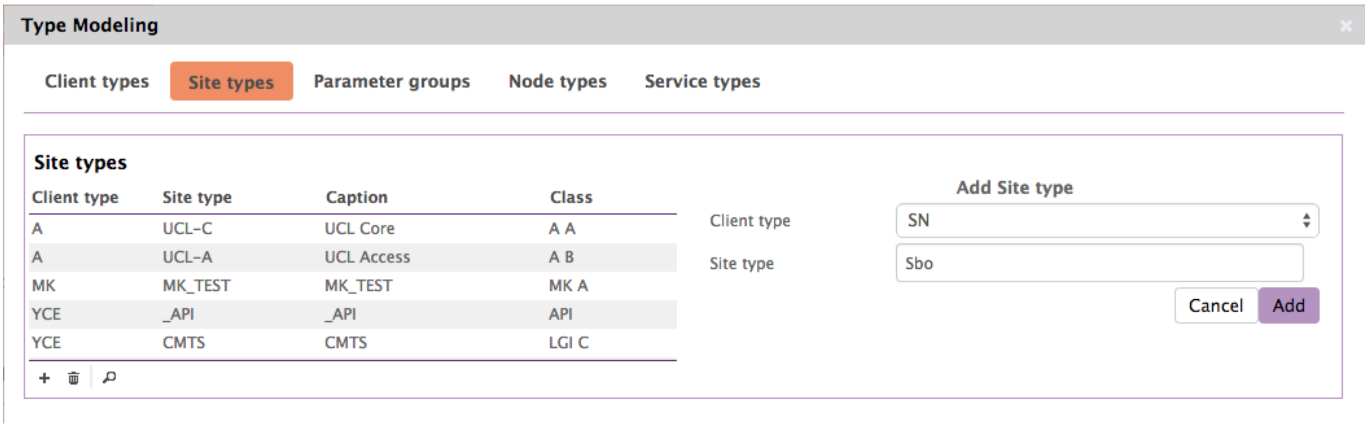
You can select one from the free list (if any) or type a new supernet ip-network address in the '**ip-supernet**' field. Choose a value to your liking. A mismatch in network address and prefix will automatically be corrected: entering 172.17.10.0 for a /22 will result in the supernet 172.17.8.0/22.

You also need to provide a DNS domain for the supernet (at least the first time). Use 'acme.com' or something else and hit `Apply`, then `close`. Also close the forms opened earlier.

The next step would be to add locations or 'Sites' to this 'Amsterdam' client. However, the required site-types are not defined yet.

SITE_TYPE

To add site types to the **SN** client type, open the **Modeling** form from the **Design** menu. Select the '**Site types**' tab.



Site types

Client type	Site type	Caption	Class
A	UCL-C	UCL Core	A A
A	UCL-A	UCL Access	A B
MK	MK_TEST	MK_TEST	MK A
YCE	_API	_API	API
YCE	CMTS	CMTS	LGI C

Add Site type

Client type: SN

Site type: Sbo

Click the **+** in the Site_types list and select **SN** from the '**Add site type**' menu.

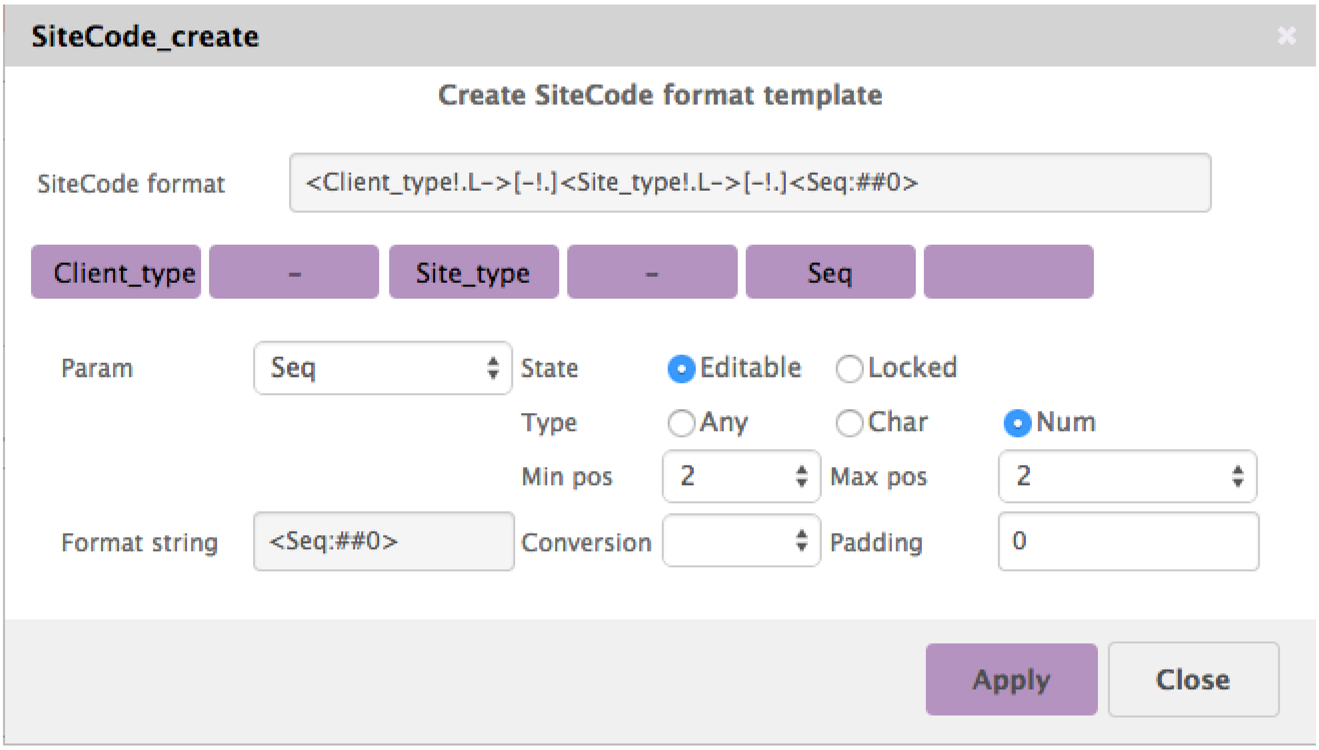
For the Site type name choose an abbreviation that describes to an operator what network architecture to expect. Usually these types already exist within the organization or the global network design. In our example we use **Sbo**, short for 'Sample Branch Office'.
Click the '**Add**' button. (ignore the 'site type exist' should one pop up).

Note the Max/Sites/Client_type and Sites/Client. These can be configured to limit the number of this site type as per global architecture. Entering `0` is equivalent to 'no limit'.

The location naming convention can be defined if desired. Often the names of the network device is based on the location name (or site-code). It is the default behaviour of the node-types.

Click on the button '**Create Format**' to define the naming rules for a new location. If no naming convention applies, leave the format blank. In that case the operator can enter the site-code without restrictions.

In the Create Format popup, click on the unlabelled button (if no format exists yet) to allow the selection of the first element of the site-code. The naming rules allow up to seven elements.
For the example we will use a naming convention where the site code consists of: `<Client_type>-<Site_type>-<Sequence_number>`, all in lower-case and non editable by the operator save for the sequence number part.



SiteCode format: <Client_type!.L->[-!.]<Site_type!.L->[-!.]<Seq:##0>

Param: Seq

State: ● Editable ○ Locked

Type: ○ Any ○ Char ● Num

Min pos: 2

Max pos: 2

Format string: <Seq:##0>

Conversion:

Padding: 0

In the end the format template will look something similar to

```
<Client_type!.L->[-!.]<Site_type!.L->[-!.]<Seq:##0>
```

Click 'Apply' to save the site type definition so far, then close.

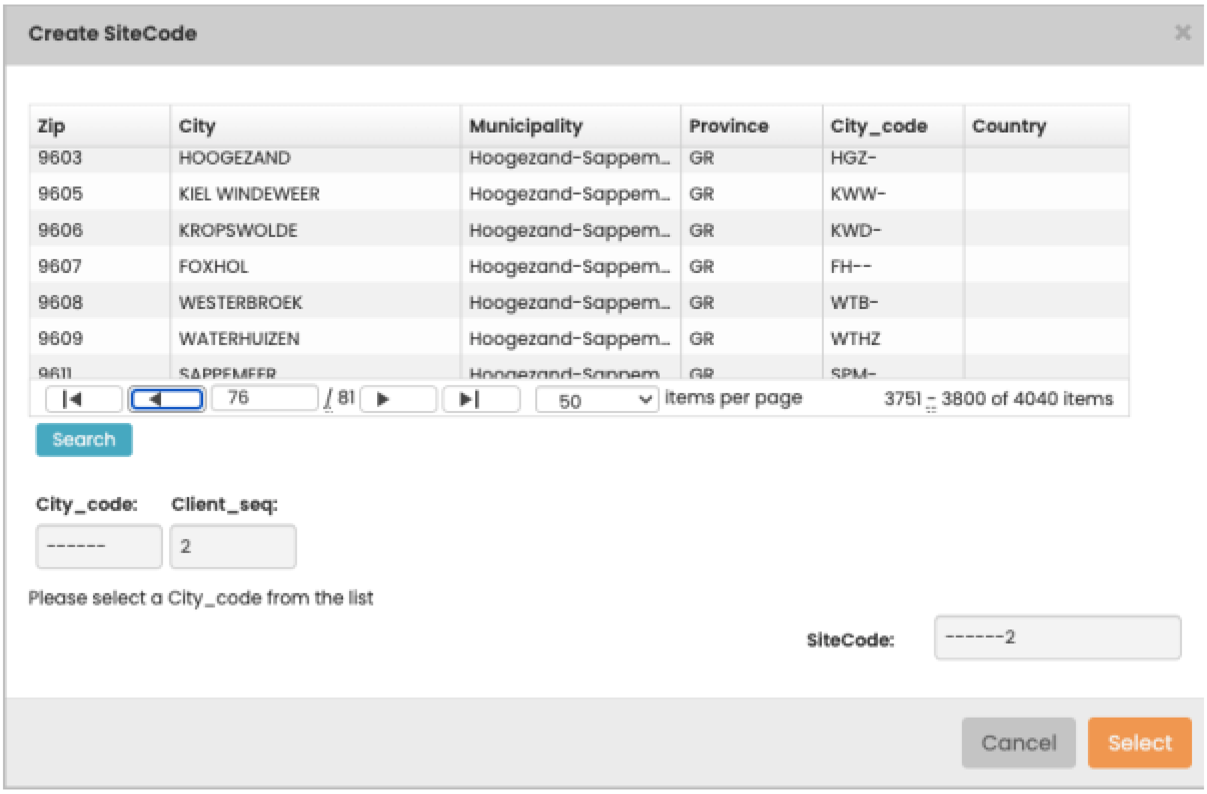
Before adding the remaining definitions to the site type, let's observe the result by adding a site to the ‘Amsterdam’ client.

ADD SITE

Add a new site to the **Amsterdam** client.

Open the **Build** menu. The ‘**Amsterdam**’ client is still selected if still working in the same session. Click the **+** icon under the ‘**Site**’ list of 'Amsterdam'.

Choose **SB0** as the Site type (the only choice so far), and click the ‘**SiteCode**’ button if enabled - meaning a site code format was used.



Zip	City	Municipality	Province	City_code	Country
9603	HOOGEZAND	Hoogezand-Sappem...	GR	HGZ-	
9605	KIEL WINDEWEER	Hoogezand-Sappem...	GR	KWW-	
9606	KROPSWOLDE	Hoogezand-Sappem...	GR	KWD-	
9607	FOXHOL	Hoogezand-Sappem...	GR	FH--	
9608	WESTERBROEK	Hoogezand-Sappem...	GR	WTB-	
9609	WATERHUIZEN	Hoogezand-Sappem...	GR	WTHZ	
9611	SAPPEMEER	Hoogezand-Sappem...	GR	SPM-	

76 / 81 50 items per page 3751 - 3800 of 4040 items

City_code: ------ Client_seq: 2

Please select a City_code from the list

SiteCode: ------2

At the bottom of the form the new site_code is shown. It should be **`sn-sbo-01`**. The form also shows selection options for zip-code and city based site codes although they are not used in the example. Click 'ok'.

If the 'SiteCode' button was disabled (because there was no format), type the new name: **`sb-sbo-01`**.

Click ‘Apply’. The ‘**Site Details**’ form then opens. It lists some default attributes. Complete as desired and close it. It can be reopened by double clicking in the Site list of the build menu.
A second site can be added in the same fashion. The site code is adjusted to remain unique.

To populate the new site with devices, subnets and topology, the **+** icon of the ‘**Nodes**’ list is used. Doing so at this stage of the site type definition, results in a lack of options and an image seems to be missing. This requires us to continue the modelling of the site.

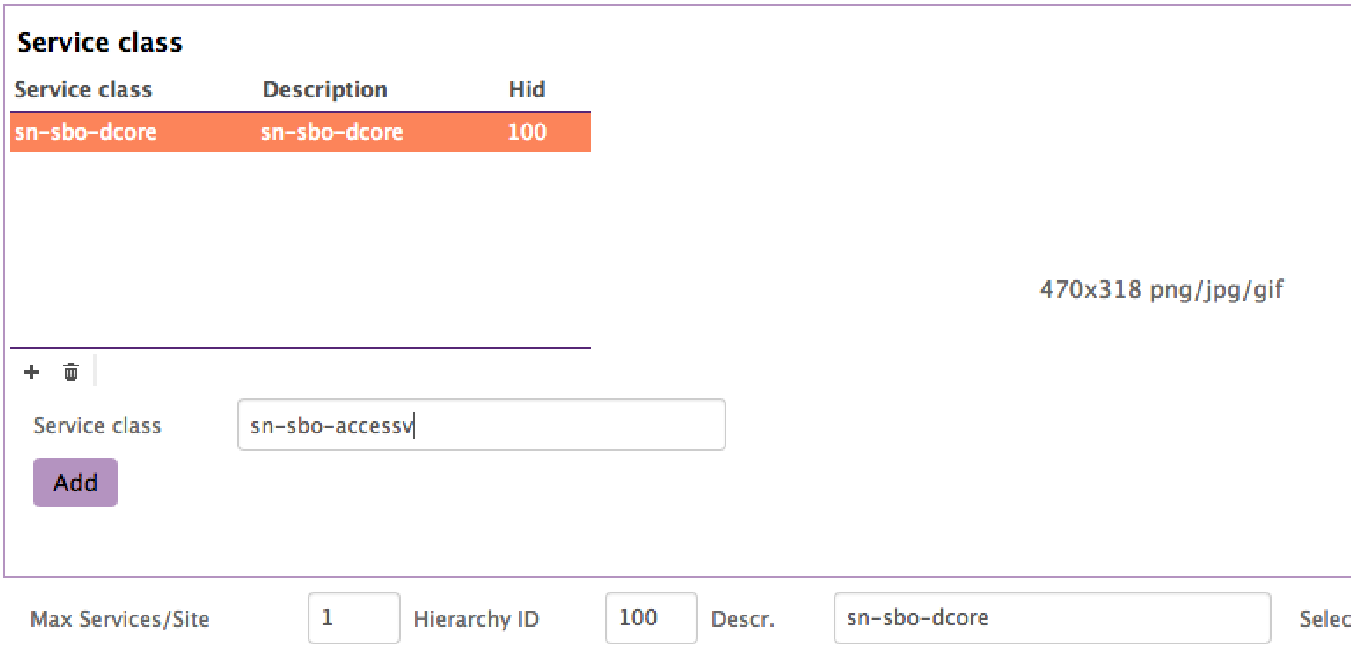
SITE TYPE: Adding Service classes

Open the ‘**Modeling**’ form from the ‘**Design**’ menu and select the '**Site Types**’ tab. Select the '**SBO**' site type again.

The ‘**Service class**’ list is currently still empty.
Now let’s assume the architecture of this site type demands the use of a redundant (dual) collapsed core and access switches that connect to each core. We call these functions ‘Service classes’ since they provide a redundant core and access services.

Click the **+** icon of the ‘Service class’ list.
Type the name for the service class, use a naming convention of your own choosing that is sufficiently flexible to allow easy identification. Use ‘**sn-sbo-dcore**’ and click ‘**Add**’.

Also add the class ‘**sn-sbo-accessv**’.



Service class

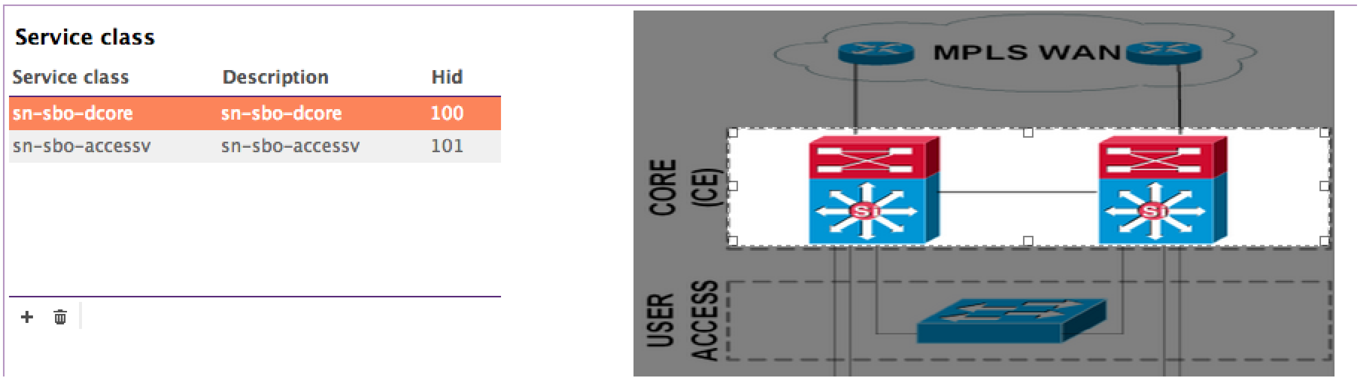
Service class	Description	Hid
sn-sbo-dcore	sn-sbo-dcore	100

Service class: sn-sbo-accessv

Max Services/Site: 1 Hierarchy ID: 100 Descr.: sn-sbo-dcore Selec

Note that each class created defaults with a ‘1’ for the Max Services/Site. That means the number of times this service can be added to the site. A ‘1’ for the core is just fine, but needs to be adjusted for the access layer. Its value could be determined by the number of access-to-core uplink ports or the naming convention. Update the access layer to a maximum of ‘5’.
Note: the hierarchy-id is a relative number indicating which service is ‘higher’ in the hierarchy. It is used to determine what is up or downlink with the lower numbers being ‘up’.

A graphical image can be uploaded to show the operator the reference architecture of the site. By dragging a rectangle over appropriate sections of this image, the modeller can create ‘buttons’ on this image that correspond to the creation of the service classes he defined. It is way operators can ignore the service class names and concentrate on the familiar drawings.



Service class

Service class	Description	Hid
sn-sbo-dcore	sn-sbo-dcore	100
sn-sbo-accessv	sn-sbo-accessv	101

The result can be observed when attempting to add a node to the **Amsterdam** - **sn-sbo-01** site. The two services are now listed, and possibly clickable graph too. However, clicking the **+** for each of these classes will produce no selectable ’**Service type**’. These will need to defined first.

Service types define the creation and usage of network components like devices (nodes), subnets, topology, relations and assignments. In the same way we had to define an ip-plan and supernets to create subnets, we need hardware, templates and blueprints as components to create nodes and configurations.

HARDWARE

Open the ‘Hardware' form from the ‘Design' menu.
Here we can define what hardware types we are allowed to use in our designs (service types).
Assume we standardised the entire (SN) network on c3560 and c4503 from Cisco.
Click the ‘+’ of the ‘Models’ list and select/enter the following values:
Vendor type: Cisco_IOS, Model_name: c3560-24, Hw_model: c3560-enterprise, Hw_type: Switch. The remainder of attributes are optional at this stage.
Also add the 4503 using Vendor type: Cisco_IOS, Model_name: c3560-24, Hw_modelc3560-enterprise, Hw_type: Router.

After these are created you will see no (default) template is associated with the models. We need to add these (main) templates before we can properly use the hardware.

MAIN TEMPLATE

Open the ‘Templates’ form from the ‘Build’ menu.
No entries should be visible at this time since the templates are always unique for the client_type. Therefore only those templates are shown of the client_type where the client belongs to. This way a template change will never affect other networks.

In the ‘Main templates’ tab, add the two templates for the hardware we added. Start with:

- Template: sbo-dcore-4503, Type: main, Vendor_type: Cisco_IOS, Hw_model: c4503, Node_class: Core

This opens the 'Template revision’ form where the actual template text can be added. For now is it sufficient to put some remarks in the text:

```
! sbo-dcore-4503 main template
!
```

Make sure the ‘Status’ field is set to ‘production’ before clicking the ‘ok’. The default is ‘planned’. Each templates (main, sub or port templates) has a set of revisions of which only one has the ‘production’ status. After changing a ‘planned’ template to ‘production’ level, the existing production revision becomes ‘historical’. Configurations can be tested while using ‘planned’ template revisions, but they can never be pushed to a device.

Also add two more main templates using the 3560’s, on for each role they are used in:

- Template: sbo-dcore-3560-24, Type: main, Vendor_type: Cisco_IOS, Hw_model: c3560-24, Node_class: Core
- Template: sbo-accesv-3560-24, Type: main, Vendor_type: Cisco_IOS, Hw_model: c3560-24, Node_class: Access

The template texts for these are equivalent.

NODE TYPES

To use the hardware and templates together as a standard component in service types, node types are created where default and custom attributes are assigned. Although these can be created an attribute at the time, it is quicker to duplicate an existing and modify a few specifics.

Open the ‘Modeling’ form from the ‘Design’ menu, Select the 'Node types’ tab. Select any of the existing node types from the list and click the ‘Duplicate’ action.
Then select SN for the new Client type and type the Node type name: `sn-sbo-dcore-4503`.
Click the ‘Duplicate’ button.

In the ‘Parameter’ list of the new node type a few alterations need to be made to parameter values:

- Hostname: `Sequenced_nodename(“<site_code>-“,”001”,”002”)`
- Node_fqdn: `Concat(“<Hostname>”,”.acme.com”)`
- Template: `sbo-dcore-4503`

The arguments used with the function sequenced_nodename() here will create the node name for the new device based on the site code where a free sequence number is appended between 001 and 002.

The same steps can be repeated to create the node types `sn-sbo-dcore-3560-24`

- Hostname: `Sequenced_nodename(“<site_code>-“,”001”,”002”)`
- Node_fqdn: `Concat(“<Hostname>”,”.acme.com”)`
- Template: `sbo-dcore-3560-24`

and `sn-sbo-accessv-3560-24`

- Hostname: `Sequenced_nodename(“<site_code>-“,”a01”,”a99”)`
- Node_fqdn: `Concat(“<Hostname>”,”.acme.com”)`
- Template: `sbo-accessv-3560-24`

Note: More experienced modellers will probably create just two node types, one the the core, one for the access. They will then use the service types to modify the template (and implicitly the hardware too). Being a framework, the NetYCE modelling usually allows for several methods to achieve an objective.

SERVICE TYPES

The Service types will bring everything we have build so far together.

> to be added

PORT TEMPLATES

> to be added

BLUEPRINT SWITCHPORTS

> to be added

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

Permanent link:
https://labs-wiki.netyce.com/doku.php/guides:user:sample_client_type_setup

Last update: **2022/04/29 08:51**