



# VTS Service Extension and Device Templates Migration

This appendix provides details on the considerations and pre-upgrade procedural steps required to identify the impacted service extension templates and device templates because of NED upgrade. Before you begin to pre-upgrade, make sure that the impacted templates to be fixed and migrated to the new NED version.

- [Pre-upgrade Considerations, on page 1](#)

## Pre-upgrade Considerations

The following steps provide pre-upgrade procedure to identify the impacted service extension templates and device templates migration.

### Step 1 To identify the impacted templates:

Download the `template-migration.tar` file from the CCO based on the Current and Target version of the VTS.

- After downloading the `.tar` file, transfer it to the VTC machine that has the customer CDB installed.
- Extract the `.tar` file through "`tar -xvf <filename>`", it will create the directory "`vts-launcher`" "`cd vts-launcher`" and "`ls -lt`" to get the list of files.
- Identify the file "`find-impacted-templates-<version1>-to-<version2>`" according to the migration path you are doing and login as an "`admin`" user to execute it.

**Note** If you have impacted templates you will have a directory created called "`templates`" that will contain a file per impacted template named "`<template name>.impacted.keys`". In each file there is a list of key paths that indicates an impacted key in that template.

- Enter "`cli`" mode through "`ncs_cli -u admin -C`" and for each template that was impacted (For example, has a file in the "`templates`" directory) execute the following CLI to export the template as json:

```
show running-config templates template <templatename> | display json | save ./templates/<template name>.json
```

Make sure that the `templates` directory has the permission set to `777` before running this command.

- Transfer the "`templates`" directory to your laptop or remote server and fix the templates by following next steps either manually (2a) or by using a dummy target VM (2b).

### Step 2 Fix the templates:

## 2a. Fix the templates manually:

You can fix templates manually. With the available list of impacted template files you can fix each one of the templates to be compatible with the target version. For example, after executing the script `./find-impacted-templates-252-262.sh` you can find a file named "sample1.impactd.keys" in *my templates* directory along with an already exported template called/named "sample1.json".

**Note** The "sample1.json" will contain old content of the template and exported using the "show" command.

When you open the "sample1.impactd.Keys" file, you can view the following entry:

```
config/nx:router/ospf{}/area{}/range{}/mask
```

The above entry indicates that in the upgrade-to version the **"mask"** attribute was either altered or deleted. To figure out the exact changes, open the target `<schema name>.txt` under the `vts-launchers/schemas` directory and seek the path up until the change. In this example, we seek the string `"config/nx:router/ospf{}/area{}/range"`.

**Note** For more examples of 2.6.2 upgrade, refer to the schema file - `nso4612.txt` file.

**Figure 1: Schema**

```

config/nx:router/ospf{}/area{}/filter-list/route-nap{}
config/nx:router/ospf{}/area{}/filter-list/route-nap{}/direction
config/nx:router/ospf{}/area{}/filter-list/route-nap{}/drection
config/nx:router/ospf{}/area{}/filter-list/route-nap{}/name
config/nx:router/ospf{}/area{}/nssa
config/nx:router/ospf{}/area{}/nssa/default-information-originate
config/nx:router/ospf{}/area{}/nssa/default-information-originate/route-nap
config/nx:router/ospf{}/area{}/nssa/no-redistribution
config/nx:router/ospf{}/area{}/nssa/no-summary
config/nx:router/ospf{}/area{}/stub
config/nx:router/ospf{}/area{}/stub/no-summary
config/nx:router/ospf{}/area{}/nssa-translate
config/nx:router/ospf{}/area{}/nssa-translate/nssa
config/nx:router/ospf{}/area{}/nssa-translate/nssa/translate
config/nx:router/ospf{}/area{}/nssa-translate/nssa/translate/type?
config/nx:router/ospf{}/area{}/range{}
config/nx:router/ospf{}/area{}/range{}/ip
config/nx:router/ospf{}/area{}/range{}/not-advertise
config/nx:router/ospf{}/area{}/range{}/cost
config/nx:router/ospf{}/area{}/sham-link{}
config/nx:router/ospf{}/area{}/sham-link{}/source
config/nx:router/ospf{}/area{}/sham-link{}/dest
config/nx:router/ospf{}/area{}/sham-link{}/cost
config/nx:router/ospf{}/area{}/virtual-link{}
config/nx:router/ospf{}/area{}/virtual-link{}/id
config/nx:router/ospf{}/area{}/virtual-link{}/authentication
config/nx:router/ospf{}/area{}/virtual-link{}/authentication-key
config/nx:router/ospf{}/area{}/virtual-link{}/authentication-key/auth-type
config/nx:router/ospf{}/area{}/virtual-link{}/authentication-key/auth-key
config/nx:router/ospf{}/area{}/virtual-link{}/dead-interval
config/nx:router/ospf{}/area{}/virtual-link{}/hello-interval
config/nx:router/ospf{}/area{}/virtual-link{}/message-digest-key{}
config/nx:router/ospf{}/area{}/virtual-link{}/message-digest-key{}/id
config/nx:router/ospf{}/area{}/virtual-link{}/message-digest-key{}/md5
config/nx:router/ospf{}/area{}/virtual-link{}/message-digest-key{}/md5/auth-type
config/nx:router/ospf{}/area{}/virtual-link{}/message-digest-key{}/md5/auth-key
config/nx:router/ospf{}/area{}/virtual-link{}/retransmit-interval
config/nx:router/ospf{}/area{}/virtual-link{}/transmit-delay
config/nx:router/ospf{}/auto-cost
config/nx:router/ospf{}/auto-cost/reference-bandwidth
config/nx:router/ospf{}/auto-cost/rate-uns
config/nx:router/ospf{}/bfd
config/nx:router/ospf{}/log-adjacency-changes
config/nx:router/ospf{}/log-adjacency-changes/detail
config/nx:router/ospf{}/max-lsa
config/nx:router/ospf{}/max-lsa/number
config/nx:router/ospf{}/max-lsa/threshold-value
config/nx:router/ospf{}/max-lsa/ignore-tlne
config/nx:router/ospf{}/max-lsa/ignore-count
config/nx:router/ospf{}/max-lsa/reset-tlne
config/nx:router/ospf{}/max-lsa/warning-only
config/nx:router/ospf{}/max-metric
config/nx:router/ospf{}/max-metric/router-lsa
config/nx:router/ospf{}/max-metric/external-lsa
config/nx:router/ospf{}/max-metric/external-lsa-value
config/nx:router/ospf{}/max-metric/include-stub
config/nx:router/ospf{}/max-metric/on-startup
config/nx:router/ospf{}/max-metric/on-startup-value

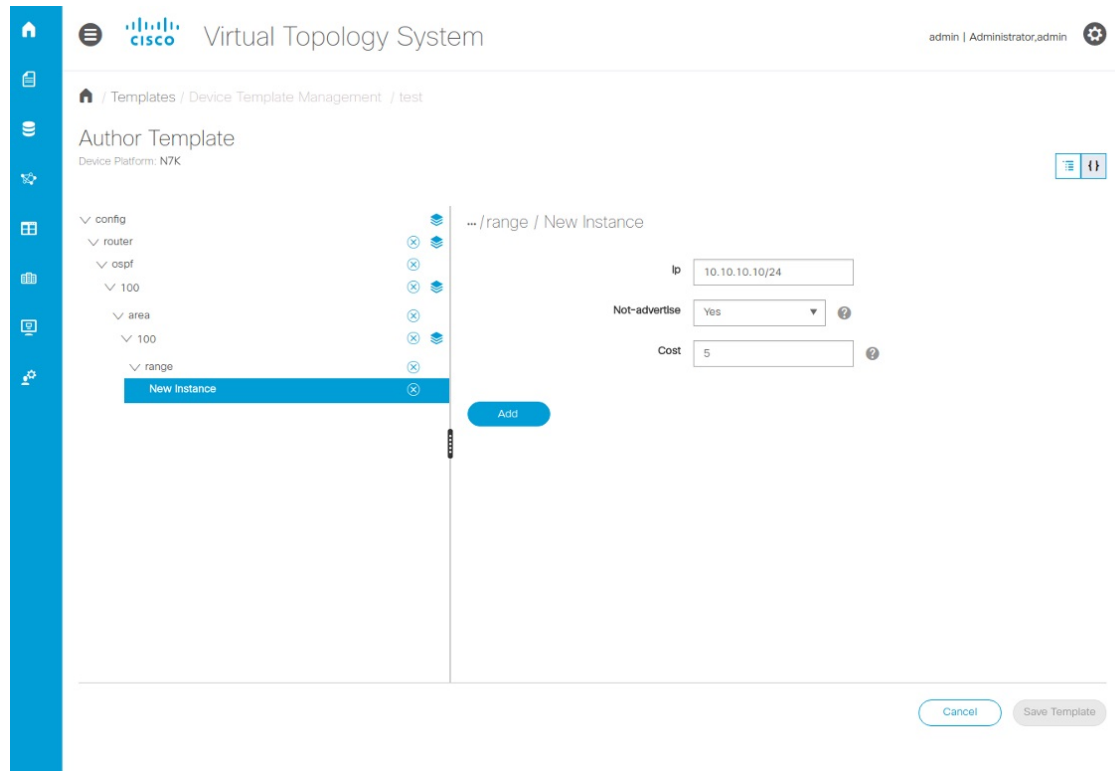
```

Since there is no `"/mask"` it is clear that the attribute was deleted. To identify the attribute that was added,

- Use a target VTC machine with the target version.
- Create that path as part of a new template.

You can view the targeted VTC version in the VTS GUI as shown in below figure:

Figure 2: GUI of the targeted VTC version:



As shown in the below image, remove the "mask" attribute and add it as ".xx" to the ip address and do that fix in the json file:

Figure 3: Before the fix json file:

```

Activities Terminal Mon 15:45
saichler@saichler: ~/vts252r/vts-launcher/templates
File Edit View Search Terminal Help
{
  "data": {
    "cisco-vts-templates:templates": {
      "template": [
        {
          "name": "sample1",
          "description": "sample1",
          "type": "cisco-vts-templates:device",
          "created-on": "2017-11-13T23:09:04.6+00:00",
          "modified-on": "2017-11-13T23:09:04.601+00:00",
          "device-template": {
            "platform": "cisco-vts-identities:N7K",
            "schema-revision-date": "2016-12-30",
            "schema-namespace": "http://tal1-f.com/ned/cisco-nx",
            "keypath-values": {
              "keypath-value": [
                {
                  "keypath": "config/nx/router/ospf{100}",
                  "value": "{\ospf{\:{\id\:\:100\,\:isolate\:\:true,\:default-metric\:\:5,\:distance\:\:7,\:maximum-paths\:\:14,\:bfd\:\:true,\:router-id\:\:10.10.10.1\:\:}"
                },
                {
                  "keypath": "config/nx/router/ospf{100}/area{100}",
                  "value": "{\:area{\:\:default-cost\:\:10,\:nssa-or-stub\:\:\:stub\,\:\:id\:\:\:100\:\:}"
                },
                {
                  "keypath": "config/nx/router/ospf{100}/area{100}/range{192.168.3.5 255.255.255.0}",
                  "value": "{\:range{\:\:ip\:\:192.168.3.5\,\:mask\:\:255.255.255.0\,\:cost\:\:50,\:advertise-choice\:\:advertise\:\:}"
                }
              ]
            }
          }
        }
      ]
    }
  }
}

```

Figure 4: After the fix json file:

```

File Edit View Search Terminal Help
{
  "data": {
    "cisco-vts-templates:templates": {
      "template": [
        {
          "name": "sample1",
          "description": "sample1",
          "type": "cisco-vts-templates:device",
          "created-on": "2017-11-13T23:09:04.6+00:00",
          "modified-on": "2017-11-13T23:09:04.601+00:00",
          "device-template": {
            "platform": "cisco-vts-identities:N7K",
            "schema-revision-date": "2016-12-30",
            "schema-namespace": "http://tall-f.com/ned/cisco-nx",
            "keypath-values": {
              "keypath-value": {
                "keypath": "config/nx:router/ospf{100}",
                "value": "{\ospf\":{\id\":100,\isolate\":true,\default-metric\":5,\distance\":7,\maximun-paths\":14,\bfd\":true,\router-id\":10.10.10.1}}",
                },
                "keypath": "config/nx:router/ospf{100}/area{100}",
                "value": "{\area\":{\default-cost\":10,\nssa-or-stub\":stub\",id\":100}}",
                },
                "keypath": "config/nx:router/ospf{100}/area{100}/range{192.168.3.5 255.255.255.0}",
                "value": "{\range\":{\ip\":192.168.3.5/24,\cost\":50,\advertise-choice\":advertise}}",
                }
              }
            }
          }
        }
      ]
    }
  }
}

```

After the fix as shown in the above image, the sample1.json file is ready to be used during the migration process.

Follow the above steps and migrate all the impacted templates in the *templates* directory according to the new model.

To automate the 2a section, you can refer to the sample conversion script which will convert the impacted templates to the format compatible to VTS 2.6.2.

**Note** This conversation script is not supported and you need to identify the impacted Keypaths and modify and fix/update the script accordingly.

#### Steps to perform:

a. Once you identify the templates which are impacted using template migration script, you can copy this 'Sample\_Conversion2521\_262.py' script to the *vts-launcher* directory.

b. Download both the packages *pexpect* and *ptyprocess* and install them as listed below:

Copy the below two packages to */home/admin* in VTS 2.5.2 VM.

- *ptyprocess-0.5.2.tar.gz*
- *pexpect-4.4.0.tar.gz*

c. Login to VTS 2.5.2 VM as admin and switch to root by running **sudo su -**.

• Extract the packages:

```
# tar -zxvf ptyprocess-0.5.2.tar.gz
# tar -zxvf pexpect-4.4.0.tar.gz
```

• Install *ptyprocess* first:

Navigate to */home/admin/ptyprocess-0.5.2* directory, run **chmod 777 \***, and run:

```
# python3 setup.py install
```

• Install *pexpect*:

Navigate to */home/admin/pexpect-4.4.0* directory, run **chmod 777 \***, and run:

```
# python3 setup.py install
```

Once done, go back to the `/vts-launcher` directory and run the conversion script. At the Username/Password prompt, enter the UI login credentials.

```
VTS-252-To-262/vts-launcher# python3 Sample_Conversion2521_262.py

Enter VTS Username: admin
Enter VTS Password:
```

Here, impacted templates will get converted to the VTS 2.6.2 compatible format and are stored in the `templates-for-migration/` directory, which is located in the `cd /vts-launcher/./templates-for-migration` directory.

- d. Copy the `templates-for-migration` directory to `/home/admin` and follow the upgrade process steps by providing proper permissions and proceeding with the upgrade process.

### 2b. To fix the templates by creating a dummy target VM:

If there is a target VTC VM available, recreate the impacted templates with the same name and export as json files by running the following command from the NCS CLI:

```
- show running-config templates template <template name> | display json | save ./tmp/<template name>.json
```

**For VTS 2.6.2 as it supports multi sites, when you create a template in VTS 2.6.2 you can view this template under the specific site.**

If you are exporting the templates from VTS 2.6.2, use the command:

```
% show running-config vts-service sites site 111C3493-DFC6-4D54-8E0C-02470CA25111 templates template N7K-RS | display json | save /tmp/N7K-RS.json
```

where default-site-id is `111C3493-DFC6-4D54-8E0C-02470CA25111`

These json files can be used for template migration in the next step by copying to `/home/admin/templates-for-migration` in the VTC VM instead of fixing the existing templates.

## Step 3 Copy the migrated templates to the VTC 2.5.2 machine

- a) Create a directory `/home/admin/templates-for-migration` in the VTC VM and copy the migrated json files to the directory. During VTS upgrade, templates in the json files will get updated in the CDB. Make sure that `templates-for-migration` directory with the contents have the correct ownership and permission before starting the upgrade.

**Note** Template migration has to be completed and the migrated templates needs to be copied back to the VTC VM prior to taking the backup.

- b) Run the following commands to set the right permission:

```
chown -R admin:vts-admin /home/admin/templates-for-migration
chmod -R 777 /home/admin/templates-for-migration
```

Only migrated json files which need be imported back to the CDB should be present in the `templates-for-migration` directory.

