

Discovery 6 - Migrate a CDM Device

Introduction

In this activity, you will learn how to migrate a device and its service data to a new NED ID. After completing this activity, you will be able to:

- Migrate a CDM device to a different NED.
- Update an existing service for it to work in a new NED.

Job Aids

The following job aid is available to help you complete the lab activities:

- This lab guide
- Student guide, for general explanations

Device Information

Device	Description	IP Address	Credentials
Student	Linux Ubuntu VM	10.10.20.50	developer, C1sco12345
R1 (dist-rtr01)	Virtual Router, IOS XE - Amsterdam-17.3.4	10.10.20.175	cisco, cisco
R2 (dist-rtr02)	Virtual Router, IOS XE - Amsterdam-17.3.4	10.10.20.176	cisco, cisco
R3 (internet-rtr01)	Virtual Router, IOS XE - Amsterdam-17.3.4	172.21.1.181	cisco, cisco
R4 (internet-rtr01)	Virtual Router, IOS XE - Amsterdam-17.3.4	172.21.1.181	cisco, cisco

Task 1: Migrate a NED

In this task, you will examine the implications of a NED migration.

Activity

Complete these steps:

Step 1 Connect to the NSO Linux server by clicking on the NSO server icon and open the Terminal by clicking the icon on the bottom bar.

```
developer@devbox:~$
```

Step 2 Go to NSO running instance directory and fetch the changes from the GitLab server.

```
developer@devbox:~$ cd nso-run
developer@devbox:~/nso-run$ git pull
Updating bdc559d..6479ed0
Fast-forward
 README.md                      | 4 ++--
 cisco-ios-cli-6.21.zip         | Bin 0 -> 60893878 bytes
 netflow.zip                   | Bin 0 -> 17278 bytes
 3 files changed, 2 insertions(+), 2 deletions(-)
 create mode 100644 cisco-ios-cli-6.21.zip
 create mode 100644 netflow.zip
```

Step 3 Unpack the **cisco-ios-cli-6.21.zip** NED to **packages** directory.

```
developer@devbox:~/nso-run$ unzip cisco-ios-cli-6.21.zip -d packages/
Archive:  cisco-ios-cli-6.21.zip
  creating: packages/cisco-ios-cli-6.21/
  creating: packages/cisco-ios-cli-6.21/load-dir/
   ...
  creating: packages/cisco-ios-cli-6.21/LICENSES.nedcom/
 inflating: packages/cisco-ios-cli-6.21/LICENSES.nedcom/APACHE_V2_LICENSE
 inflating: packages/cisco-ios-cli-6.21/LICENSES.nedcom/CUP_LICENSE
 inflating: packages/cisco-ios-cli-6.21/LICENSES.nedcom/README
 inflating: packages/cisco-ios-cli-6.21/build-meta-data.xml
 inflating: packages/cisco-ios-cli-6.21/package-meta-data.xml
developer@devbox:~/nso-run$
```

Step 4 Connect to the NSO CLI and issue **packages reload**.

```
developer@devbox:~$ ncs_cli -Cu admin

admin connected from 127.0.0.1 using console on student-vm
admin@ncs# packages reload
>>> System upgrade is starting.
>>> Sessions in configure mode must exit to operational mode.
>>> No configuration changes can be performed until upgrade has completed.
>>> System upgrade has completed successfully.
reload-result {
  package cisco-ios-cli-6.21
  result true
}
reload-result {
  package cisco-ios-cli-6.67
  result true
}
reload-result {
  package dns-snmp-service
  result true
}
```

```

}
admin@ncs#
System message at 2022-09-15 12:35:12...
  Subsystem stopped: ncs-dp-3-cisco-ios-cli-6.67:IOSDp
admin@ncs#
System message at 2022-09-15 12:35:12...
  Subsystem stopped: ncs-dp-2-cisco-ios-cli-6.21:IOSDp
admin@ncs#
System message at 2022-09-15 12:35:12...
  Subsystem started: ncs-dp-4-cisco-ios-cli-6.21:IOSDp
admin@ncs#
System message at 2022-09-15 12:35:12...
  Subsystem started: ncs-dp-5-cisco-ios-cli-6.67:IOSDp

```

Step 5 Add a new device named **R4** and perform a **sync-from**. Make sure that the device R4 is using an older version of the Cisco IOS NED – *cisco-ios-cli-6.21*.

```

admin@ncs# config
Entering configuration mode terminal
admin@ncs(config)# devices device R4
admin@ncs(config-device-R4)# address 10.10.20.181
admin@ncs(config-device-R4)# authgroup lab
admin@ncs(config-device-R4)# device-type cli ned-id cisco-ios-cli-
6.21 protocol telnet
admin@ncs(config-device-R4)# state admin-state unlocked
admin@ncs(config-device-R4)# commit
Commit complete.
admin@ncs(config-device-R4)# end
admin@ncs# devices device R4 sync-from
result true
admin@ncs#

```

Step 6 Exit the NSO CLI and go to running instance directory.

```

admin@ncs# exit
developer@devbox:~$ cd /home/developer/nso-run/

```

Step 7 Extract the *netflow* package to the */packages* directory.

```

developer@devbox:~/nso-run$ unzip netflow.zip -d packages/
Archive:  netflow.zip
  creating: packages/netflow/
  inflating: packages/netflow/package-meta-data.xml
  creating: packages/netflow/src/
  creating: packages/netflow/src/yang/
  inflating: packages/netflow/src/yang/netflow.yang
  inflating: packages/netflow/src/Makefile
  creating: packages/netflow/templates/
  inflating: packages/netflow/templates/netflow-template.xml
  creating: packages/netflow/test/
  creating: packages/netflow/test/internal/
  creating: packages/netflow/test/internal/lux/

```

```

creating: packages/netflow/test/internal/lux/basic/
inflating: packages/netflow/test/internal/lux/basic/run.lux
extracting: packages/netflow/test/internal/lux/basic/Makefile
inflating: packages/netflow/test/internal/lux/Makefile
inflating: packages/netflow/test/internal/Makefile
inflating: packages/netflow/test/Makefile
creating: packages/netflow/load-dir/
inflating: packages/netflow/load-dir/netflow.fxs
developer@devbox:~/nso-run$

```

Step 8 Navigate to *netflow* service model directory and open the *netflow.yang* file and inspect the YANG model.

This is how the contents of the file should look when you open it. You can see, that the package is used to model a service, that export NetFlow data to a specific destination.

```

developer@devbox:~/nso-run/packages/netflow/src$ cd packages/netflow/src/yang
developer@devbox:~/nso-run/packages/netflow/src$ cat netflow.yang
module netflow {
  namespace "http://cisco.com/example/netflow";
  prefix netflow;

  import ietf-inet-types {
    prefix inet;
  }
  import tailf-ncs {
    prefix ncs;
  }
  import tailf-common {
    prefix tailf;
  }
  augment /ncs:services {
    list netflow {
      description "Export NetFlow service";
      key device;

      uses ncs:service-data;
      ncs:servicepoint "netflow";

      leaf device {
        tailf:info "Device to export NetFlow data from";
        type leafref {
          path "/ncs:devices/ncs:device/ncs:name";
        }
      }

      leaf destination {
        tailf:info "Address to export NetFlow data to";
        type inet:ipv4-address;
      }
    }
  }
}

```

Step 9 Connect to the NSO CLI and reload the packages.

You will notice that the netflow package can't be reloaded now. The conflict occurs in the netflow package template, because the new NED apparently handles the netflow configuration differently than the old one.

```
developer@devbox:~/nso-run/packages/netflow/src/yang $ ncs_cli -Cu admin

admin connected from 127.0.0.1 using console on student-vm
admin@ncs# packages reload

>>> System upgrade is starting.
>>> Sessions in configure mode must exit to operational mode.
>>> No configuration changes can be performed until upgrade has completed.
>>> System upgrade has completed successfully.
reload-result {
  package cisco-ios-cli-6.21
  result true
}
reload-result {
  package cisco-ios-cli-6.67
  result true
}
reload-result {
  package dns-snmp-service
  result true
}
reload-result {
  package netflow
  result false
  info netflow-template.xml:12 the tag: version is different for ned-ids:
cisco-ios-cli-6.21:cisco-ios-cli-6.21, cisco-ios-cli-6.67:cisco-ios-cli-6.67
}
admin@ncs#
System message at 2022-09-15 15:38:45...
  Subsystem stopped: ncs-dp-15-cisco-ios-cli-6.21:IOSDp
admin@ncs#
System message at 2022-09-15 15:38:45...
  Subsystem started: ncs-dp-17-cisco-ios-cli-6.21:IOSDp
admin@ncs#
System message at 2022-09-15 15:38:45...
  Subsystem started: ncs-dp-18-cisco-ios-cli-6.67:IOSDp
admin@ncs# *** ALARM package-load-failure: netflow-template.xml:12 the tag:
version is different for ned-ids: cisco-ios-cli-6.21:cisco-ios-cli-6.21,
cisco-ios-cli-6.67:cisco-ios-cli-6.67
admin@ncs#
```

Step 10 Observe the differences between two NEDs by using the dry run of the migrate command.

```
admin@ncs# devices migrate device [ R4 ] old-ned-id cisco-ios-cli-6.21 new-
ned-id cisco-ios-cli-6.67 verbose dry-run
migrate-result {
  device R4
  result true
  modified-path {
    path /devices/device[name='R4']/config/ios:spd/headroom
    info leaf/leaf-list type has changed
  }
}
```

```

    }
    modified-path {
        path /devices/device[name='R4']/config/ios:mpls/traffic-
eng/reoptimize/timers/delay/cleanup
        info leaf/leaf-list type has changed from uint8 to uint16
    }
    modified-path {
        path /devices/device[name='R4']/config/ios:line/vty/transport/input
        info leaf/leaf-list type has changed
    }
    modified-path {
        path /devices/device[name='R4']/config/ios:ip/flow-export/destination
        info node type has changed from non-presence container to list
    }
    modified-path {
        path /devices/device[name='R4']/config/ios:ip/igmp/ssm-map/static
        info node type has changed from non-presence container to list
    }
    modified-path {
        path /devices/device[name='R4']/config/ios:ip/multicast/route-limit
        info node type has changed from leaf to non-presence container
    }
}
admin@ncs#

```

There are changes between the two device model versions. To find the changes that affect the Netflow configuration, you have to interpret the package reload error message from the previous step first:

```

netflow-template.xml:12 the tag: version is different for ned-ids: cisco-ios-
cli-6.21:cisco-ios-cli-6.21, cisco-ios-cli-6.67:cisco-ios-cli-6.67

```

The error message tells you that there is an error in the line 12 of the configuration template netflow-template.xml. This is related to **flow-export** configuration:

```

01 <?xml version="1.0"?>
02 <config-template xmlns="http://tail-f.com/ns/config/1.0"
    servicepoint="netflow">
03   <devices xmlns="http://tail-f.com/ns/ncs">
04     <device>
05       <name>{/device}</name>
06       <config>
07         <ip xmlns="urn:ios">
08           <flow-export>
09             <source>
10               <GigabitEthernet>4</GigabitEthernet>
11             </source>
12             <version>5</version>
13             <destination>
14               <ip>{/destination}</ip>
15             </destination>
16           </flow-export>
17         </ip>
18       </config>
19     </device>
20   </devices>
21 </config-template>

```

Run the **dry-run** of the **migrate** command again, with an additional condition "**begin flow-export**", which will display lines beginning with a line that includes "flow-export" statement.

```

admin@ncs# devices migrate device [ R4 ] old-ned-id cisco-ios-cli-6.21 new-
ned-id cisco-ios-cli-6.67 verbose dry-run | begin flow-export | more
    path /devices/device[name='R4']/config/ios:ip/flow-export/destination
    info node type has changed from non-presence container to list
  }
  modified-path {
    path /devices/device[name='R4']/config/ios:ip/igmp/ssm-map/static
    info node type has changed from non-presence container to list
  }
  modified-path {
    path /devices/device[name='R4']/config/ios:ip/multicast/route-limit
    info node type has changed from leaf to non-presence container
  }
}
admin@ncs#

```

The "destination" node has been changed from a non-presence container to a list. These changes are usually incompatible with the older configuration.

Step 11 Migrate the device to the new NED ID. This time without the *dry-run* parameter. In order to have the least amount of service downtime as possible, you should also use the *no-networking* parameter, which prevents any southbound communication to the devices. This means that the CDB and the devices will be out of sync for the time being, until the affected services are fixed, redeployed and the device configuration is synchronized.

Note As an alternative approach to the CDM device migration, you can prepare and update the service in advance, in order to avoid going out of sync. This requires an additional device that already uses the newer version of the NED, which allows you to create the configuration template for the updated service.

```

admin@ncs(config)# exit
admin@ncs# devices migrate device [ R4 ] old-ned-id cisco-ios-cli-6.21 new-
ned-id cisco-ios-cli-6.67 no-networking
migrate-result {
  device R4
  result true
  modified-path {
    path /devices/device[name='R4']/config/ios:spd/headroom
    info leaf/leaf-list type has changed
  }
  ...
  modified-path {
    path /devices/device[name='R4']/config/ios:ip/flow-export/destination
    info node type has changed from non-presence container to list
  }
  modified-path {
    path /devices/device[name='R4']/config/ios:ip/igmp/ssm-map/static
    info node type has changed from non-presence container to list
  }
  modified-path {
    path /devices/device[name='R4']/config/ios:ip/multicast/route-limit
    info node type has changed from leaf to non-presence container
  }
}

```

```
admin@ncs#
```

Step 12 Verify that the R4 device is using the cisco-ios-cli-6.67 NED.

```
admin@ncs# admin@ncs# show running-config devices device R4 device-type
devices device R4
  device-type cli ned-id cisco-ios-cli-6.67
  device-type cli protocol telnet
!
admin@ncs#
```

Activity Verification

You have completed this task when you attain the following result:

- The R4 device is using the cisco-ios-cli-6.67 NED.

Task 2: Update the Service After NED Migration

In this task, you will update the existing service for it to work on both NED versions.

Activity

Complete these steps:

Step 1 Manually configure the configuration for NetFlow data exporting on device R4, which now uses a new NED. Because also the device's operating system was upgraded, the commands have changed. There is a different way of configuring the NetFlow data exporting. The destination now also requires a transport protocol and a destination port. Use the default 2055 port. Display the changes using the dry-run commit parameter.

```
admin@ncs# config
admin@ncs(config)# devices device R4 config
admin@ncs(config-config)# flow exporter NetFlow
admin@ncs(config-flow-exporter)# source GigabitEthernet 4
admin@ncs(config-flow-exporter)# destination 10.100.0.1
admin@ncs(config-flow-exporter)# export-protocol netflow-v5
admin@ncs(config-flow-exporter)# transport udp 2055
admin@ncs(config-config)# commit dry-run outformat xml
result-xml {
  local-node {
    data <devices xmlns="http://tail-f.com/ns/ncs">
      <device>
        <name>R4</name>
        <config>
          <flow xmlns="urn:ios">
            <exporter>
              <name>NetFlow</name>
```



```

        <destination>
          <address>10.100.0.1</address>
        </destination>
        <source>
          <GigabitEthernet>4</GigabitEthernet>
        </source>
        <transport>
          <udp>2055</udp>
        </transport>
        <export-protocol>netflow-v5</export-protocol>
      </exporter>
    </flow>
  </config>
</device>
</devices>
}
}
admin@ncs(config-config)#

```

Step 2 Do not commit the changes and exit the NSO CLI.

```

admin@ncs(config-config)# abort
admin@ncs# exit
developer@devbox:~/nso-run/packages/netflow/src/yang$

```

Step 3 You will now modify the XML template to support both versions of the NED. Since both versions use the "urn:ios" namespace, you will have to make the distinction as to which version of a NED to use for which part of the configuration. Open the XML template.

```

developer@devbox:~/nso-run/packages/netflow/src$ code ../../templates/netflow-
template.xml

```

Step 4 You can use the `<?if-ned-id?>` template processing instructions to set the configuration for a specific NED. Add an instruction for the old cisco-ios-cli-6.21 NED for the existing flow-export configuration and add another instruction for the new cisco-ios-cli-6.67 NED configuration, a combination of migrated changes and dry-run.

```

<?xml version="1.0"?>
<config-template xmlns="http://tail-f.com/ns/config/1.0"
servicepoint="netflow">
  <devices xmlns="http://tail-f.com/ns/ncs">
    <device>
      <name>{/device}</name>
      <config>
        <?if-ned-id cisco-ios-cli-6.21:cisco-ios-cli-6.21?>
          <ip xmlns="urn:ios">
            <flow-export>
              <source>
                <GigabitEthernet>4</GigabitEthernet>
              </source>
              <version>5</version>
              <destination>

```

```

        <ip>{/destination}</ip>
      </destination>
    </flow-export>
  </ip>
  <?elif-ned-id cisco-ios-cli-6.67:cisco-ios-cli-6.67?>
    <flow xmlns="urn:ios">
      <exporter>
        <name>NetFlow</name>
        <destination>
          <address>{/destination}</address>
        </destination>
        <source>
          <GigabitEthernet>4</GigabitEthernet>
        </source>
        <transport>
          <udp>2055</udp>
        </transport>
        <export-protocol>netflow-v5</export-protocol>
      </exporter>
    </flow>
  <?end?>
</config>
</device>
</devices>
</config-template>

```

Step 5 Save the file.

Step 6 Enter NSO CLI and reload the packages.

No error should be present now for the netflow package.

```

developer@devbox:~/nso-run/packages/netflow/src $ ncs_cli -Cu admin

admin connected from 127.0.0.1 using console on student-vm
admin@ncs# packages reload
reload-result {
  package cisco-ios-cli-6.21
  result true
}
reload-result {
  package cisco-ios-cli-6.67
  result true
}
reload-result {
  package dns-snmp-service
  result true
}
reload-result {
  package netflow
  result true
}
admin@ncs#
System message at 2022-09-15 16:34:23...
  Subsystem stopped: ncs-dp-17-cisco-ios-cli-6.21:IOSDp

```

```
admin@ncs#  
System message at 2022-09-15 16:34:23...  
  Subsystem stopped: ncs-dp-18-cisco-ios-cli-6.67:IOSDp  
admin@ncs#  
System message at 2022-09-15 16:34:23...  
  Subsystem started: ncs-dp-20-cisco-ios-cli-6.21:IOSDp  
admin@ncs#  
System message at 2022-09-15 16:34:23...  
  Subsystem started: ncs-dp-21-cisco-ios-cli-6.67:IOSDp
```

Step 7 Synchronize the CDB configuration from the R4 device.

```
admin@ncs# devices device R4 sync-from  
result true  
admin@ncs#
```

Step 8 Deploy the *netflow* service to device R4. Use the address 10.100.0.1 as the destination.

```
admin@ncs# config  
Entering configuration mode terminal  
admin@ncs(config)# services netflow R4 destination 10.100.0.1
```

Step 9 Inspect a dry run of the commit for changes.

```
admin@ncs(config-netflow-R4)# commit dry-run  
cli {  
  local-node {  
    data devices {  
      device R4 {  
        config {  
          flow {  
+           exporter NetFlow {  
+             destination {  
+               address 10.100.0.1;  
+             }  
+             source {  
+               GigabitEthernet 4;  
+             }  
+             transport {  
+               udp 2055;  
+             }  
+             export-protocol netflow-v5;  
+           }  
        }  
      }  
    }  
  }  
  services {  
+    netflow R4 {  
+      destination 10.100.0.1;  
+    }  
  }  
}
```

```
}  
admin@ncs(config-netflow-R4)#
```

Step 10 Commit the changes and exit the NSO CLI.

```
admin@ncs(config-netflow-R4)# commit  
Commit complete.  
admin@ncs(config-netflow-R4)# end  
admin@ncs#
```

Step 11 Verify the configuration of the deployed netflow service.

```
admin@ncs# show running-config devices device R4 config flow | display xml  
<config xmlns="http://tail-f.com/ns/config/1.0">  
  <devices xmlns="http://tail-f.com/ns/ncs">  
    <device>  
      <name>R4</name>  
      <config>  
        <flow xmlns="urn:ios">  
          <exporter>  
            <name>NetFlow</name>  
            <destination>  
              <address>10.100.0.1</address>  
            </destination>  
            <source>  
              <GigabitEthernet>4</GigabitEthernet>  
            </source>  
            <transport>  
              <udp>2055</udp>  
            </transport>  
            <export-protocol>netflow-v5</export-protocol>  
          </exporter>  
        </flow>  
      </config>  
    </device>  
  </devices>  
</config>
```

Activity Verification

You have completed this task you attain the following result:

- The netflow service has been successfully deployed using the new cisco-ios-cli-6.67 NED.