



Cisco NSO DLM Service Pack

Installation Guide

Version 6.0.2

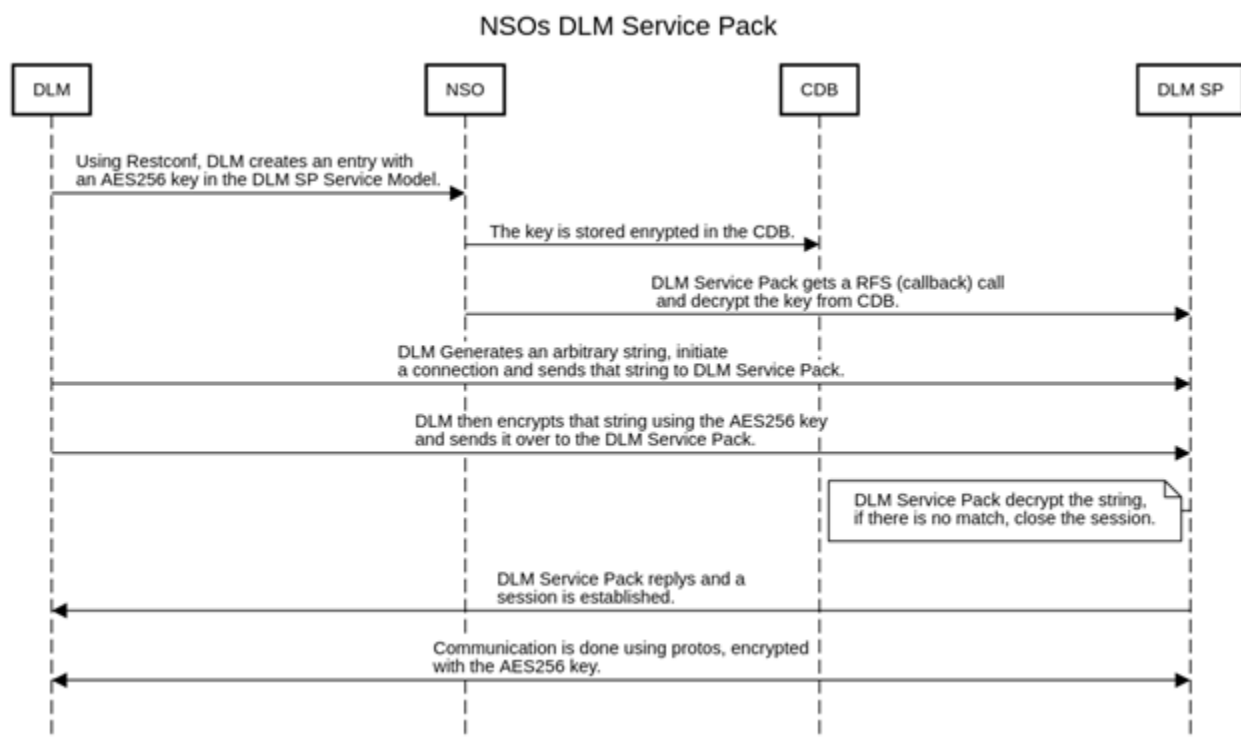
Contents

Overview	2
Cisco NSO Prerequisite Configurations.....	3
Installing the Service Pack.....	4
DLM Service Pack APIs	5
NSO Sync Policy.....	6
Writing Expression.....	12

Overview

The Device Lifecycle Management (DLM) function pack is installed on a Cisco Network Services Orchestrator (NSO) instance in order to synchronize device configurations in Cisco Crosswork with those in Cisco NSO. It effectively channels communication between the Crosswork DLM and Cisco NSO. The DLM function pack extracts device configurations from Cisco NSO and compares them with the device configurations in Cisco Crosswork. DLM function pack installation procedure remains same for LSA and non-LSA deployments.

The following diagram provides an overview of the function of the DLM function pack for Cisco NSO:



This document describes how to install the DLM service pack on Cisco NSO 6.1.9 and configure the Cisco NSO sync policy in DLM.

Cisco NSO Prerequisite Configurations

For the DLM to be able to perform the netconf reachability test to NSO, ensure that you have the following configuration in the NSO **ncs.conf** configuration file. The file is usually located in the **/etc/ncs/** directory.

```
<ssh>
  <algorithms>
    <kex>diffie-hellman-group14-sha1</kex>
    <mac>hmac-sha2-512,hmac-sha2-256,hmac-sha1</mac>
    <encryption>aes128-ctr,aes192-ctr,aes256-ctr</encryption>
  </algorithms>
</ssh>
```

Ensure that the **sshpas** utility is installed on your computer/Cisco NSO machine, depending on where you download the service pack files. For example, on Ubuntu distribution, the command is “**sudo apt-get install sshpass**”.

Log in as admin user on the Cisco NSO machine and perform the following:

```
ncs_cli -u admin -C
config terminal
nacm groups group ncsadmin
user-name <your nso machine ssh user>
end
```

Installing the Service Pack

To install the DLM service pack:

1. Copy the DLM service pack.tar.gz file to the NSO machine and place it in the **/var/opt/ncs/packages** directory using the root user.
2. Log into NSO with a user that has sudo privileges and that is part of the **ncsadmin** group (any user except the root user).
3. Enter the NSO prompt using “**ncs_cli -u admin**”. At the ncs prompt, execute “**admin@ncs> request packages reload**”.

Note: If you install other service packs after installing the DLM service pack, then step 2 is not required.

DLM Service Pack APIs

DLM exposes the following APIs to allow you to invoke actions on NSO.

Request Sync

Request DLM to perform a sync with Cisco NSO per the Cisco NSO policy.

Method: POST

Endpoint: `/crosswork/inventory/v1/nso/sync`

Payload: empty object, e.g. "{}"

Fetch-Ssh-Keys

Request DLM to schedule a fetch-ssh-keys for the node per its association to the NSO Device.

Method: POST

Endpoint: `/crosswork/inventory/v1/nso/fetch-ssh-keys`

Payload: A RobotNodeGetReq proto, similar to the GetNodes API.

Connect

Request DLM to schedule a connect for the node per its association to the NSO device.

Method: POST

Endpoint: `/crosswork/inventory/v1/nso/connect`

Payload: A RobotNodeGetReq proto, similar to the GetNodes API.

Sync-From

Request DLM to schedule a sync-from for the node per its association to the NSO device.

Method: POST

Endpoint: `/crosswork/inventory/v1/nso/sync-from`

Payload: A RobotNodeGetReq proto, similar to the GetNodes API.

Sync-To

Request DLM to schedule a sync-to for the node per its association to the NSO device.

Method: POST

Endpoint: `/crosswork/inventory/v1/nso/sync-to`

Payload: A RobotNodeGetReq proto, similar to the GetNodes API.

Check-Sync

Request DLM to schedule a check-sync for the node and its association to the NSO device.

Method: POST

Endpoint: `/crosswork/inventory/v1/nso/check-sync`

Payload: A RobotNodeGetReq proto, similar to the GetNodes API.

Service Pack Sync Policy

Update the NSO sync policy that contains guidance and rules for the DLM sync procedure as to various actions it can take.

Method: PUT

Endpoint: `/crosswork/inventory/v1/nso/policy`

Payload: A NsoPolicy proto. Refer to the [NSO Sync Policy](#) section for explanation.

NSO Sync Policy

The DLM to/from sync policy allows you, at runtime, to filter the scope of the providers, devices, and direction of the sync process.

By default, all devices in DLM are onboarded to Cisco NSO and vice versa. The Cisco NSO sync policy allows you to onboard a specific subset of devices by defining an expression to filter the full list of devices. It also allows you to fine-tune the driver (NED) to be used on Cisco NSO (if not the default NED).

Following is the out-of-the-box default policy:

```
{
  "name": "default",
  "providers_criteria": "*",
  "provider_policy": {
    "<your provider name>": {
      "match": true,
      "matchRule": "*",
      "onboardTo": true,
      "onboardToRule": "",
      "onboardFrom": false,
      "onboardFromRule": "",
      "syncFrom": true,
      "syncFromRule": "*",
      "checkSync": true,
      "checkSyncRule": "",
      "neds": [
      ]
    }
  }
}
```

For LSA deployments, following is the example policy:

```
{
  "name": "default",
  "providers_criteria": "*",
  "provider_policy": {
    "<your provider name (CFS)>": {
      "match": true,
      "matchRule": "*",
      "onboardTo": true,
      "onboardToRule": "*",
      "onboardFrom": false,
```

```
    "onboardFromRule": "*",
    "syncFrom": true,
    "syncFromRule": "*",
    "checkSync": true,
    "checkSyncRule": "*",
    "neds": [
    ]
  },
  "<your provider name(RFS1)>": {
    "match": true,
    "matchRule": "*",
    "onboardTo": true,
    "onboardToRule": "*",
    "onboardFrom": false,
    "onboardFromRule": "*",
    "syncFrom": true,
    "syncFromRule": "*",
    "checkSync": true,
    "checkSyncRule": "*",
    "neds": [
    ]
  },
  "<your provider name(RFS2)>": {
    "match": true,
    "matchRule": "*",
    "onboardTo": true,
    "onboardToRule": "*",
    "onboardFrom": false,
    "onboardFromRule": "*",
    "syncFrom": true,
    "syncFromRule": "*",
    "checkSync": true,
    "checkSyncRule": "*",
    "neds": [
    ]
  }
},
"lsa": true,
"policy": {
```



```

    "auto_onboard_rfs": true,
    "rfs_spread_method": "ROUND_ROBIN",
    "rfs_spread_value": 150
  }
}

```

Example 1

The following example policy sets the cisco-iosxr-cli-7.52.2 NED for all IOS-XR devices with a software version prefix of 6.23.xxx:

```

{
  "name": "default",
  "providers_criteria": "*",
  "provider_policy": {
    "<your provider name>": {
      "match": true,
      "matchRule": "*",
      "onboardTo": true,
      "onboardToRule": "*",
      "onboardFrom": false,
      "onboardFromRule": "*",
      "syncFrom": true,
      "syncFromRule": "*",
      "checkSync": true,
      "checkSyncRule": "*",
      "neds": [
        {
          "ned": "cisco-iosxr-cli-7.52.2",
          "rule": "productinfo.softwaretype='IOS XR' and
productinfo.softwareversion='6.23.*'"
        }
      ]
    }
  }
}

```

Example 2

The following example defines that Juniper devices will not be onboarded and only devices with a hostname beginning with 'east' will be onboarded.

```

{
  "name": "default",

```

```

"providers_criteria": "*",
"provider_policy": {
  "<your provider name>": {
    "match": true,
    "matchRule": "*",
    "onboardTo": true,
    "onboardToRule": "productinfo.softwaretype!='JUNOS' and
hostname='East*'",
    "onboardFrom": false,
    "onboardFromRule": "*",
    "syncFrom": true,
    "syncFromRule": "*",
    "checkSync": true,
    "checkSyncRule": "*",
    "neds": [
      {
        "ned": "cisco-iosxr-cli-7.52.2",
        "rule": "productinfo.softwaretype='IOS XR' and
productinfo.softwareversion='6.23.*'"
      }
    ]
  }
}

```

Name

DLM will overwrite with “default” any name given during an update.

Providers_criteria

Allows you to decide which NSO providers, defined in Crosswork, should be considered for the sync process. For Crosswork 4.1, it should always be “*” as there is only one NSO provider.

Provider_policy[]

Per provider, identified by the provider’s name, you can decide what actions DLM will take when matching/syncing with NSO.

provider_policy[]/match

DLM matches the devices in Crosswork to the devices in NSO, cross referencing the IP address.

provider_policy[]/matchRule

If not all, e.g. “*”, an expression defining the subset of devices via the different node attributes. Refer to the [Writing Expression](#) section.

provider_policy[]/onboardTo

DLM onboards devices to NSO when those devices are missing on NSO.

provider_policy[]/onboardToRule

If not all, e.g. “*”, an expression defining the subset of devices via the different node attributes. Refer to the [Writing Expression](#) section.

provider_policy[]/onboardFrom

DLM onboards devices from NSO when the devices are missing on Crosswork.

provider_policy[]/onboardFromRule

If not all, e.g. “*”, an expression defining the subset of devices via the different node attributes. Refer to the [Writing Expression](#) section.

provider_policy[]/syncFrom

DLM performs a sync-from on the NSO device after DLM has onboarded the device on NSO.

provider_policy[]/syncFromRule

If not all, e.g. “*”, an expression defining the subset of devices via the different node attributes. Refer to the [Writing Expression](#) section.

provider_policy[]/checkSync

DLM performs a check-sync on an NSO device if the device already exists on NSO.

provider_policy[]/checkSyncRule

If not all, e.g. “*”, an expression defining the subset of devices via the different node attributes. Refer to the [Writing Expression](#) section.

provider_policy[]/Neds[]

NSO provides the ability to include several NEDs per device. By default, DLM will use the latest CLI NED that exists on NSO. If you want DLM to use a different NED for a group of devices, you can do so here by setting the following:

provider_policy[]/neds[]/ned

The NED package name, as defined by NSO.

provider_policy[]/neds[]/Rule

If not all, e.g. “*”, an expression defining the subset of devices via the different node attributes. Refer to the [Writing Expression](#) section.

lsa

A global indication if the system is in LSA model or not

policy/auto_onboard_rfs

In LSA mode, indicates DLM it should auto onboard devices to RFS.

policy/rfs_spread_method

Method to be used by DLM to spread devices across multiple RFS. Values include ROUND_ROBIN, CAPACITY, USER_DEFINED.

policy/rfs_spread_value

Number of devices to be assigned to each RFS instance if the rfs_spread_method is CAPACITY.

Writing Expression

All the node attributes described in the Swagger document can be used for scoping and filtering. The following example shows how to refer to the different attributes of the node:

```
productinfo.softwaretype='IOS XR' and productinfo.softwareversion>='6.13' and profile='devices'
```

The above expressions scope those nodes that have a software type of **IOS XR**, with a software version equal to or greater than **6.13** and a credentials profile set to **devices**.

The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on standards documentation, or language that is used by a referenced third-party product.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/c/en/us/about/legal/trademarks.html>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)