



Cisco Network Services Orchestrator DLM Service Pack 1.0 Installation Guide

First Published: April 19, 2021

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. YOU MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California. NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

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: www.cisco.com/go/trademarks. 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)

© 2021 Cisco Systems, Inc. All rights reserved.

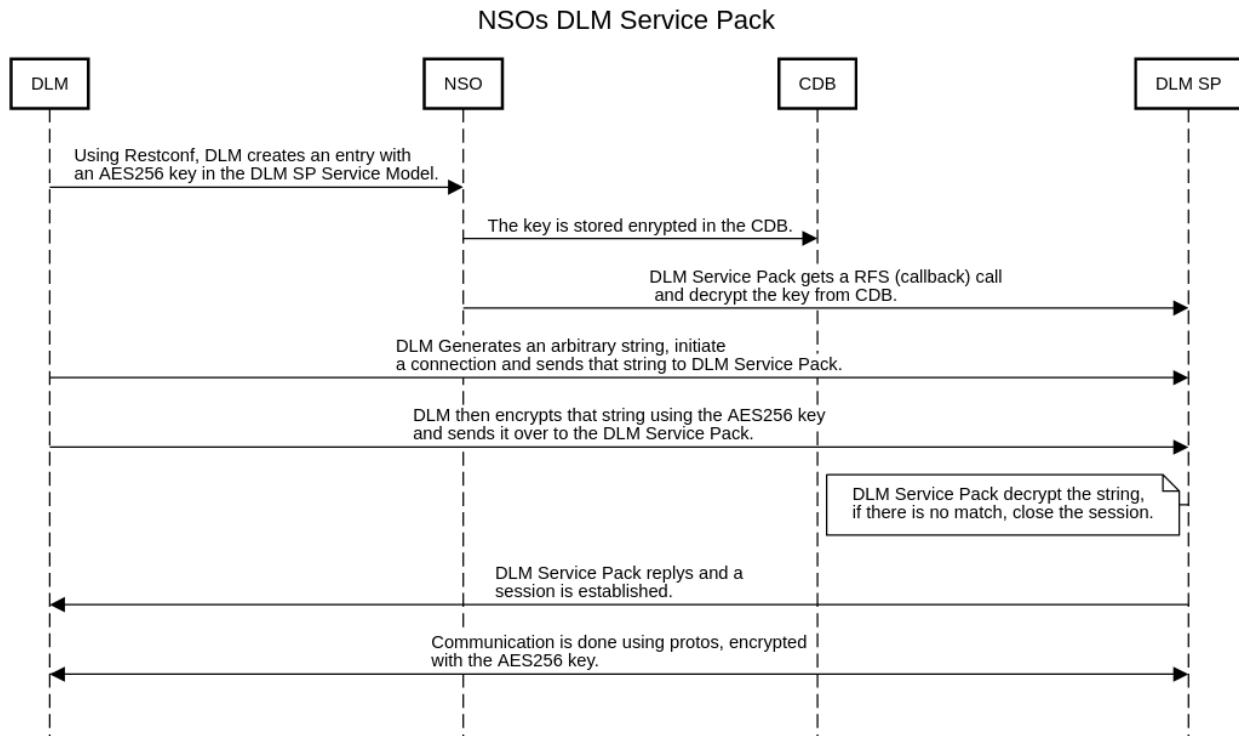
Contents

Introduction	4
Overview of NSO Service Pack.....	4
NSO Pre-Requisite Configurations.....	4
Manual Install.....	5
DLM Service Pack APIs.....	5
Request Sync.....	5
Fetch-Ssh-Keys.....	5
Connect.....	5
Sync-From.....	5
Sync-To.....	6
Check-Sync.....	6
Service Pack Sync Policy.....	6
NSO Sync Policy.....	6
Name	7
Providers_criteria.....	7
Provider_policy[]	7
Writing Expression	8

Introduction

Starting from NSO version 5.4, the Restful API was removed. After evaluating the various options, the best, fastest and most reliable solution was to develop a Java based service pack to service DLM needs with NSO. This document describes how to install the DLM Service Pack and how to configure the NSO Sync Policy in DLM.

Overview of NSO Service Pack



NSO Pre-Requisite Configurations

In NSO `ncs.conf` configuration file (usually located at `/etc/ncs/` directory), please make sure you have the following config. This is for DLM to be able to do the Netconf reachability test to NSO.

```

<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>
  
```

Make sure the `sshpas` utility is installed on your Laptop/NSO machine, depending on to where you downloaded the service pack files. On Ubuntu distribution, for example, the command is “`sudo apt-get install sshpass`”.

Login to your NSO admin user on the NSO machine and do the following:

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

Manual Install

1. Copy the dlm service pack (.tar.gz) file to /var/opt/ncs/packages on the NSO VM.
2. As the admin user, issue the following commands within the NSO CLI console:

```
ncs_cli -u admin -C
machine request packages reload
```

DLM Service Pack APIs

DLM is exposing the below APIs to allow the user to invoke some actions on NSO.

Request Sync

Request DLM to perform a sync with NSO per the 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 per 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, containing guidance & rules for the DLM sync procedure of various actions it can take.

Method: PUT

Endpoint: </crosswork/inventory/v1/nso/policy>

Payload: A NSOPolicy proto. Please see the next section for explanation.

NSO Sync Policy

The DLM to/from NSO Sync policy allows the user, in runtime, to filter & set the scope of the providers, devices & direction of the sync process.

Here 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": [
        ]
    }
}
}

```

Name

Always be default, DLM will overwrite with “default” any name given during update.

Providers_criteria

Allows the user to decide which NSO providers, defined in CW, should be considered for the sync process. For future use, in CW 4.0 should always be “*” as there is only one NSO provider.

Provider_policy[]

Per provider, identified by the provider name, you can decide of the different actions that DLM should take when matching/synching with NSO.

provider_policy[]/match

Should DLM match devices in CW to the devices in NSO, cross referencing the IP.

provider_policy[]/matchRule

If not all, e.g. “*”, an expression defining the subset of devices via the different nodes attributes. Please refer to the Writing Expression section.

provider_policy[]/onboardTo

Should DLM onboard 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 nodes attributes. Please refer to the Writing Expression section.

provider_policy[]/onboardFrom

Should DLM onboard devices from NSO when those devices are missing on CW.

provider_policy[]/onboardFromRule

If not all, e.g. `**`, an expression defining the subset of devices via the different nodes attributes. Please refer to the Writing Expression section.

provider_policy[]/syncFrom

Should DLM perform a sync-from on NSO device when DLM freshly onboarded the device on NSO.

provider_policy[]/syncFromRule

If not all, e.g. `**`, an expression defining the subset of devices via the different nodes attributes. Please refer to the Writing Expression section.

provider_policy[]/checkSync

Should DLM perform a check-sync on an NSO device if the device was already existing on NSO.

provider_policy[]/checkSyncRule

If not all, e.g. `**`, an expression defining the subset of devices via the different nodes attributes. Please 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 exist on NSO. If you wish for 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 nodes attributes. Please refer to the Writing Expression section.

Writing Expression

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

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

The above expression scope those nodes that have a software type of IOS XR and their software version is equal or greater than 6.13 and their credentials profile is set to devices.

