



# Cisco NSO DLM サービスパック インストールレーション ガイド

バージョン 4.4.0

---

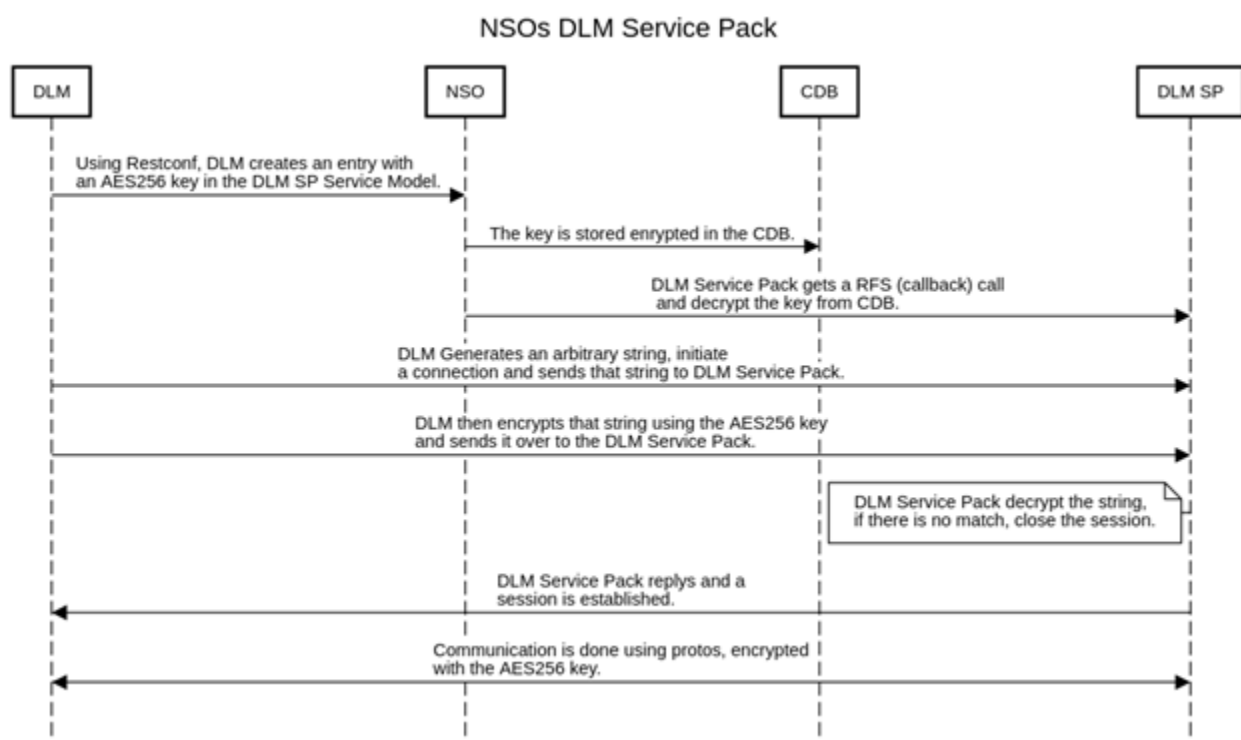
# 目次

概要 .....	2
Cisco NSO に必要な構成 .....	3
サービスパックのインストール.....	4
DLM サービスパック API.....	5
NSO 同期ポリシー .....	6
式の記述.....	11

## 概要

Device Lifecycle Management (DLM) 機能パックは、Cisco Crosswork のデバイス設定を Cisco NSO の設定と同期するために Cisco Network Services Orchestrator (NSO) インスタンスにインストールされます。Crosswork DLM と Cisco NSO の間の通信を効果的にチャネリングします。DLM 機能パックは、Cisco NSO からデバイス設定を抽出し、それらを Cisco Crosswork のデバイス設定と比較します。DLM 機能パックのインストール手順は、LSA 展開と非 LSA 展開のどちらの場合でも同じです。

次の図は、Cisco NSO 用 DLM 機能パックの機能の概要を示しています。



このドキュメントでは、DLM サービスパックを Cisco NSO 5.7.6 にインストールし、DLM で Cisco NSO 同期ポリシーを設定する方法について説明します。

## Cisco NSO に必要な構成

DLM が NSO への netconf 到達可能性テストを実行できるようにするには、NSO `ncs.conf` 構成ファイルに次の構成が含まれていることを確認してください。このファイルは通常、`/etc/ncs/` ディレクトリにあります。

```
<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>
```

サービスパックファイルをダウンロードする場所に応じて、`sshpass` ユーティリティがコンピュータ/Cisco NSO マシンにインストールされていることを確認します。たとえば、Ubuntu ディストリビューションの場合、コマンドは「`sudo apt-get install sshpass`」です。

Cisco NSO マシンに管理者ユーザーとしてログインし、次を実行します。

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

## サービスパックのインストール

DLM サービスパックをインストールするには、次の手順を実行します。

1. DLM サービスパックの .tar.gz ファイルを NSO マシンにコピーし、root ユーザーを使用して **/var/opt/ncs/packages** ディレクトリに配置します。
2. sudo 権限を持ち、ncsadmin グループに含まれるユーザー（root ユーザーを除くすべてのユーザー）で NSO にログインします。
3. 「**ncs\_cli -u admin**」を使用して NSO プロンプトを開始します。NCS プロンプトで、「**admin@ncs> request packages reload**」を実行します。

**注：** DLM サービスパックのインストール後に他のサービスパックをインストールする場合、手順 2 は必要ありません。

## DLM サービスバック API

DLM は、NSO でアクションを呼び出すことができるように、次の API を公開します。

### Request Sync

Cisco NSO ポリシーに従って、DLM に Cisco NSO との同期を実行するように要求します。

メソッド : POST

エンドポイント : `/crosswork/inventory/v1/nso/sync`

ペイロード : 空のオブジェクト。「{}」など

### Fetch-Ssh-Keys

NSO デバイスへの関連付けに従って、ノードの `fetch-ssh-keys` をスケジュールするように DLM に要求します。

メソッド : POST

エンドポイント : `/crosswork/inventory/v1/nso/fetch-ssh-keys`

ペイロード : GetNodes API と同様の RobotNodeGetReq プロトコル

### 接続

NSO デバイスへの関連付けに従って、ノードの接続をスケジュールするように DLM に要求します。

メソッド : POST

エンドポイント : `/crosswork/inventory/v1/nso/connect`

ペイロード : GetNodes API と同様の RobotNodeGetReq プロトコル

### Sync-From

NSO デバイスへの関連付けに従って、ノードの `sync-from` をスケジュールするように DLM に要求します。

メソッド : POST

エンドポイント : `/crosswork/inventory/v1/nso/sync-from`

ペイロード : GetNodes API と同様の RobotNodeGetReq プロトコル

### Sync-To

NSO デバイスへの関連付けごとにノードの `sync-to` をスケジュールするように DLM に要求します。

メソッド : POST

エンドポイント : `/crosswork/inventory/v1/nso/sync-to`

ペイロード : GetNodes API と同様の RobotNodeGetReq プロトコル

### Check-Sync

ノードとその NSO デバイスへの関連付けの `check-sync` をスケジュールするように DLM に要求します。

メソッド : POST

エンドポイント : `/crosswork/inventory/v1/nso/check-sync`

ペイロード : GetNodes API と同様の RobotNodeGetReq プロトコル

### Service Pack Sync Policy

実行可能なさまざまなアクションに関する DLM 同期手順のガイダンスとルールを含む NSO 同期ポリシーを更新します。

メソッド : PUT

エンドポイント : /crosswork/inventory/v1/nso/policy

ペイロード : NsoPolicy プロトコル説明については、次のセクションを参照してください。

## NSO 同期ポリシー

DLM 同期先/同期元ポリシーを使用すると、実行時に、プロバイダー、デバイスの範囲、および同期プロセスの方向をフィルタリングできます。

デフォルトでは、DLM のすべてのデバイスは Cisco NSO に対してオンボードされます（その逆も同様）。Cisco NSO 同期ポリシーでは、デバイスの完全なリストをフィルタリングする式を定義することにより、特定のサブセットのデバイスをオンボードできます。また、Cisco NSO で使用されるドライバ (NED) を微調整することもできます（デフォルトの NED でない場合）。

以下は、すぐに使用できるデフォルトのポリシーです。

```
{
  "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": [
        ]
      }
    }
  }
}
```

LSA 展開におけるポリシーの例を次に示します。

```
{
  "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
}
}

```

## 例 1

次のポリシー例では、ソフトウェアバージョンプレフィックスが 6.23.xxx であるすべての IOS-XR デバイスに cisco-iosxr-cli-7.29 NED を設定します。

```

{
  "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-7.29",
          "rule": "productinfo.softwaretype='IOS XR' and
productinfo.softwareversion='6.23.*'"
        }
      ]
    }
  }
}

```

## 例 2

次の例では、Juniper デバイスがオンボードされず、「east」で始まるホスト名を持つデバイスのみがオンボードされるように定義しています。

```

{
  "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-7.29",
        "rule": "productinfo.softwaretype='IOS' and
productinfo.softwareversion='6.23.*'"
      }
    ]
  }
}
}

```

## 名前

DLM は、更新中に指定された名前を「デフォルト」で上書きします。

## Providers\_criteria

Crosswork で定義されているどの NSO プロバイダーを同期プロセスに含めるかを決定できます。Crosswork 4.1 の場合、NSO プロバイダーは 1 つしかないため、常に「\*」にする必要があります。

## Provider\_policy[]

プロバイダー名で識別されるプロバイダーごとに、NSO との照合/同期時に DLM が実行するアクションを決定できます。

### provider\_policy[]/match

DLM は、Crosswork 内のデバイスを NSO 内のデバイスと照合し、IP アドレスを相互参照します。

### provider\_policy[]/matchRule

すべてを対象としない場合（「\*」など）、さまざまなノード属性を使用してデバイスのサブセットを定義する式。「式の記述」のセクションを参照してください。

### provider\_policy[]/onboardTo

DLM は、デバイスが NSO がない場合、デバイスを NSO にオンボードします。

### provider\_policy[]/onboardToRule

すべてを対象としない場合（「\*」など）、さまざまなノード属性を使用してデバイスのサブセットを定義する式。「式の記述」のセクションを参照してください。

**provider\_policy[]/onboardFrom**

DLM は、デバイスが Crosswork で見つからない場合、デバイスを NSO からオンボードします。

**provider\_policy[]/onboardFromRule**

すべてを対象としない場合（「\*」など）、さまざまなノード属性を使用してデバイスのサブセットを定義する式。「式の記述」のセクションを参照してください。

**provider\_policy[]/syncFrom**

DLM は、DLM が NSO でデバイスをオンボードした後、NSO デバイスで sync-from を実行します。

**provider\_policy[]/syncFromRule**

すべてを対象としない場合（「\*」など）、さまざまなノード属性を使用してデバイスのサブセットを定義する式。「式の記述」のセクションを参照してください。

**provider\_policy[]/checkSync**

デバイスがすでに NSO に存在する場合、DLM は NSO デバイスで check-sync を実行します。

**provider\_policy[]/checkSyncRule**

すべてを対象としない場合（「\*」など）、さまざまなノード属性を使用してデバイスのサブセットを定義する式。「式の記述」のセクションを参照してください。

**provider\_policy[]/Neds[]**

NSO は、デバイスごとに複数の NED を含める機能を提供します。デフォルトでは、DLM は NSO に存在する最新の CLI NED を使用します。DLM が特定のデバイスのグループに対して別の NED を使用するように設定する場合は、ここで次のように設定できます。

**provider\_policy[]/neds[]/ned**

NSO によって定義された NED パッケージ名。

**provider\_policy[]/neds[]/Rule**

すべてを対象としない場合（「\*」など）、さまざまなノード属性を使用してデバイスのサブセットを定義する式。「式の記述」のセクションを参照してください。

**lsa**

システムが LSA モデルであるかどうかを示すグローバル表示。

**policy/auto\_onboard\_rfs**

LSA モードで、DLM がデバイスを RFS に自動オンボードする必要があることを示します。

**policy/rfs\_spread\_method**

複数の RFS にデバイスを分散するために DLM が使用するメソッド。値には、ROUND\_ROBIN、CAPACITY、USER\_DEFINED が含まれます。

**policy/rfs\_spread\_value**

rfs\_spread\_method が CAPACITY の場合、各 RFS インスタンスに割り当てられるデバイスの数。

## 式の記述

Swagger ドキュメントで説明されているすべてのノード属性は、範囲設定とフィルタリングに使用できます。次の例は、ノードのさまざまな属性を参照する方法を示しています。

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

上の式は、ソフトウェアタイプが **IOS XR** で、ソフトウェアバージョンが **6.13** 以上で、ログイン情報プロファイルが **デバイス** に設定されているノードを対象としています。

このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。

リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動 / 変更されている場合がありますことをご了承ください。

あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。

シスコ コンタクトセンター



自社導入をご検討されているお客様へのお問い合わせ窓口です。

製品に関して | サービスに関して | 各種キャンペーンに関して | お見積依頼 | 一般的なご質問

お問い合わせ先

お電話での問い合わせ

平日 9:00 - 17:00

0120-092-255

お問い合わせウェブフォーム

[cisco.com/jp/go/vdc\\_callback](https://cisco.com/jp/go/vdc_callback)



©2023 Cisco Systems, Inc. All rights reserved.

Cisco, Cisco Systems, およびCisco Systemsロゴは、Cisco Systems, Inc. またはその関連会社の米国およびその他の一定の国における商標登録または商標です。本書類またはウェブサイトに掲載されているその他の商標はそれぞれの権利者の財産です。「パートナー」または「partner」という用語の使用はCiscoと他社との間のパートナーシップ関係を意味するものではありません。(1502R) この資料の記載内容は2023年03月現在のものです。この資料に記載された仕様は予告なく変更する場合があります。



シスコシステムズ合同会社

〒107-6227 東京都港区赤坂9-7-1 ミッドタウン・タワー

[cisco.com/jp](https://cisco.com/jp)