



Cisco Crosswork Workflow Manager 1.0 Get Started guide

First Published: 2023-06-01

Last Modified: 2023-06-14

Americas Headquarters

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
<http://www.cisco.com>
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883



CHAPTER 1

What is CWM?

Cisco Crosswork Workflow Manager is a tool that simplifies and automates complex network operations and other business processes. It provides a centralized platform for creating, managing, and executing workflows, allowing for manual operator intervention during workflow execution while ensuring repeatability and fault-tolerance. Workflows are defined using a standardized Domain Specific Language based on the Serverless Workflow specification, enabling workflow designers to express complex business processes, dependencies, and decision logic in a unified and readable format.

- [CWM documentation overview, on page 1](#)

CWM documentation overview

Cisco Crosswork Workflow Manager Documentation comprises five guides that help you learn and use the system:

- Operator guide
- Get Started guide
- Administrator Guide
- Workflow Creator guide
- Adapter Developer guide



CHAPTER 2

Core concepts

The section contains the following topics:

- [Core concepts, on page 3](#)

Core concepts

This section defines the main concepts and components used in the Crosswork Workflow Manager (CWM) that help understand how the platform works and its features. You can familiarize yourself with them to ease the very first steps with CWM.

- **Activity:** An activity is a function that executes a single, well-defined action against the target, outside system (whether it's an app or solution). Activities are defined in adapters and allow communication with an outside system.
- **Adapter:** Adapters are responsible for communication with external services, like other applications, systems or environments. The adapters define and expose activities that are consumed by workflow definitions. Every adapter can be associated with the worker that will execute the adapter activities.
- **Adapter SDK:** The adapter SDK automatically generates the structure for the required adapter's components. Developers can further define activities that are needed and then extend the integrations with the client environment.
- **Event:** Events are created during workflow execution based on the occurrences defined in the workflow definition. All events that happened during workflow execution are recorded in the **Job Event Log** table in CWM UI.
- **Execution engine:** CWM has an internal worker called the execution engine. It enables the execution of workflow definition. This worker is not visible in the CWM UI.
- **Job:** A job represents the single execution of a particular workflow definition. To be able to run a job in Crosswork Workflow Manager, you need first to add your workflow definition to the CWM. Running a new job instantiates a workflow definition stored in CWM. Before starting a job run, you enter the initial start data (Job variables). It means that your workflow executions are isolated and may use different data than other executions of the same workflow definition.
- **Workflow:** Workflows help you capture, organize and automate processes with repeatable actions performed in a specified order. In the context of CWM, documentation differentiates between:
 - **workflow definition:** piece of code written in JSON or YAML, based on the Serverless Workflow Specification and vendor-neutral, domain-specific language.

- **workflow execution (job):** single execution of a workflow definition.
- **Workflow engine:** The workflow engine manages the way how your workflow definitions are interpreted and conducted. It receives events, schedules tasks, and manages the execution of workflows.
- **Worker:** Workers carry out the workflow and are responsible for executing the workflow definition code, relevant adapter code and activities defined in the workflow definition. Depending on your needs and scale, you can have multiple workers for every workflow definition. Your worker can be associated with one adapter and its activities or with multiple ones.
- **CWM UI:** The CWM UI is a graphical user interface of CWM that allows users to interact with the system and gives access to its core functionalities.



CHAPTER 3

Run example workflow using Cisco NSO adapter

This section contains the following topics:

- [Run example workflow using Cisco NSO adapter, on page 5](#)

Run example workflow using Cisco NSO adapter

This quick start uses a locally installed [Cisco Crosswork Network Service Orchestrator](#) application and the CWM with the Cisco NSO adapter to show you a basic use case scenario for creating and running a successful workflow. It will guide you through how to install an adapter, create a worker for the workflow execution and run the created workflow to quickly get tangible results in Cisco NSO.

Workflow overview

The purpose of the example workflow is to automatically create a VPN service for two NSO devices.

First, we point to the devices in the data input and then try to perform the NSO `check-sync` operation on them. Then, depending on the result:

- if not in sync, we push a device to perform a `sync-from`, and only then try to create a VPN for it;
- if in sync, we don't perform `sync-from` but directly create a VPN for the device.

If all the steps are executed successfully, the execution engine reports workflow execution completion and displays the final data input. The results are visible in NSO too. If the engine encounters errors while performing a step, it uses the specified `retry` policy. In case errors persist beyond the retry limits, the engine ends the execution with a **Failed** status.

Go through the sections below to learn the details of how data input, functions, states, actions, and data filters are defined. If you want to know how the sausage is made, you can read the [Create workflow](#) chapter in the [Workflow Creator](#) guide.

Prerequisites

- Cisco NSO 6.0 installed. If you don't have it, follow the [installation instructions](#).
- CWM installed using OVA. Go to [CWM Administrator guide for instructions](#).

Step 1: Install NSO adapter

To interact with Cisco NSO, CWM needs a dedicated Cisco NSO adapter. Here's how you install it using the CWM API:

Upload NSO adapter file

-
- Step 1** Get the latest NSO adapter installation file from the CWM Software Package.
 - Step 2** Go to the CWM User Interface in a browser, and log in using credentials generated upon the installation of CWM. If the application is hosted locally, then the default address is `localhost:8443`.
 - Step 3** From the navigation menu on the left, click the **swagger** icon.
 - Step 4** In the **adapters** section, click the `POST /adapter` endpoint to expand it. Inside the endpoint, click **Try it out**.
 - Step 5** In the subsection that appears, click **Choose File**, select the NSO installation file and click **Upload**, then click **Execute**.
If the server response code is `201`, the adapter file is successfully uploaded into the CWM database.
-

Deploy Cisco NSO adapter

Before you can use the adapter, you need to deploy the uploaded adapter file using the CWM API:

-
- Step 1** In the CWM API **adapters** section, click the `GET /adapter` endpoint to expand it. Inside the endpoint, click **Try it out** and **Execute**.
 - Step 2** From the server response body, copy the value of the `id` field. It should be `cisco.nso.v1.0.0` or similar, depending on your adapter version.
 - Step 3** In the CWM API **adapters** section, click the `POST /adapter/{adapterId}/deploy` endpoint to expand it.
 - Step 4** Inside the endpoint, click **Try it out**. Paste the adapter id into the **Adapter ID** field.
 - Step 5** In the **createWorker** field, set the `createWorker` parameter to `true`. This will create a worker with the same name as the adapter id.
 - Step 6** Click **Execute**.

If the server response code is `201`, the adapter plugin is successfully installed and you're good to proceed.

Note If you want to create a worker manually, follow the instructions in the CWM Administrator Guide. Remember that in this case, you will need to update the workflow definition with your created worker name.

Step 2: Create secret and resource

To define the resources and secrets to be passed in securely to the Cisco NSO adapter, you need to create a secret and resource in the CWM API. Here's how to do it:

Create secret

Step 1 In the CWM API **secrets** section, click the `POST /secret` endpoint to expand it.

Step 2 Inside the endpoint, click **Try it out**, and paste the following data into the **Secret to add** field:

```
{
  "secret": {
    "username": "admin",
    "password": "admin"
  },
  "secretId": "NSOSecret",
  "secretType": "basicAuth"
}
```

Step 3 Click **Execute**.

If the server response code is 201, the secret is successfully created and you can start creating your resource.

Create resource

Step 1 In the CWM API **resources** section, click the `POST /resource` endpoint to expand it.

Step 2 Inside the endpoint, click **Try it out**, and paste the data into the **Resource input** field (modify if needed to fit your installation):

```
{
  "resource": {
    "scheme": "http",
    "host": "127.0.0.1",
    "port": 8080
  },
  "resourceId": "NSOLocal",
  "resourceType": "cisco.nso.resource.v1.0.0",
  "secretId": "NSOSecret"
}
```

Step 3 Click **Execute**.

If the server response code is 201, the resource is successfully created and you're good to proceed.

Step 3: Set up NSO example service

The NSO example that we use for the purposes of our workflow is setting up a Layer3 VPN in a service provider MPLS network for two NSO-simulated devices. Here's how you set up the example:

Step 1 In a terminal, open your main NSO directory and go to `mpls-vpn-new-template`:

```
cd examples.ncs/service-provider/mpls-vpn-new-template
```

Step 2 Execute the Makefile by running:

Step 4: Run the workflow

```
make stop clean all start
```

This command will start your local NSO instance and the sample netsim devices.

Step 3 For the example workflow to execute successfully, execute a **Sync from** on all the netsim devices beforehand:

```
devices sync-from
```

Step 4: Run the workflow

Now that we have the NSO adapter, the worker, and the NSO example all up and running, we can create a workflow in the CWM UI and run the job.

Add new workflow

Step 1 In the CWM UI, select the **Workflows** tile from the navigation menu on the left.

Step 2 In the **Workflows** panel, click **Create new workflow**.

Step 3 In the **Create new workflow** modal, provide the required input:

- a) **Workflow definition name:** provide name for the example workflow definition: `CreateL3VPN`.
- b) **Version:** provide workflow definition version: `1.0`.

Step 4 Click **Create workflow**.

Figure 1: Create workflow

Create new workflow

Workflow definition name*

Version*

Run job

- Step 1** In the **Workflows** panel, enter the newly created workflow definition by clicking its name.
- Step 2** Click the **Code** tab and delete the sample content from the **Code** field.
- Step 3** Download the workflow file using the link below and unzip it. Open the JSON file and copy the contents. Open the file, copy the workflow definition and paste it inside the **Code** field, then click **Save changes**.

[Workflow_v1.json](#)

- Step 4** Click **Run**.
- Step 5** Download the data input file using the link below and unzip it. Open the JSON file and copy the contents. In the **Run job** modal, provide a name for the job and in the **Job variables** field, paste the copied data input.

[Data_input_v1.json](#)

- Step 6** Click **Run Job**.

Figure 2: Run job

Run job

Job name*

CreateL3VPN - 0

Workflow definition name Definition ID Version

CreateL3VPN 6c000761-ee49-4362-b1cb-2cf2a5c7024c 1.0

Start user

Job variables*

```
{
  "device0Name": "ce0",
  "device1Name": "ce1",
  "nsoResource": "NSOLocal"
}
```

Run Job **Cancel**

Step 5: Check results

Check the results of example workflow execution:

In CWM UI

- Step 1** In the CWM UI, select the **Job Manager** tile from the navigation menu on the left.

- Step 2** In the **All jobs** tab, find your job and check the status of the workflow execution in the **Status** table column.
- If the workflow is executed correctly, a green tick with **Completed** status will be visible.
 - If the workflow execution is still in progress or the engine is retrying an action, a blue label with the **Running** status will be displayed.
- Step 3** Click the job name to enter its details.
- Step 4** In the **Job Event Log** table, expand the bottommost **WorkflowExecution** entry by clicking its name.
- Step 5** In the JSON payload displayed, find the *data* key. It presents the final data output updated by the successful execution of the workflow actions for which *toStateData* inside the *actionDataFilter* was defined:

Figure 3: Event log

Job Event Log

Full event history in JSON format

Job Event Name	Job Event Type	Status	Attempts	Worker	Start Time	Close Time	Assignee
		All se					

```

"result": {
  "payloads": [
    {
      "metadata": {
        "encoding": "anNvbI9wbGFpbG=="
      },
      "data": {
        "checkSyncResult0": "in-sync",
        "checkSyncResult1": "in-sync",
        "createServiceResult": 201,
        "device0Name": "ce0",
        "device1Name": "ce1"
      }
    }
  ]
},
"workflowTaskCompletedEventId": "22"
}

```

In NSO

- Step 1** Log in to your NSO account and in the **Application hub** view, click the **Service manager** tile.
- Step 2** From the **Select service points** drop-down, select **/l3vpn:vpn/l3vpn**.
- Step 3** In the table, find **testnetwork** and click the **devices** arrow to see that your netsim devices **ce0** and **ce1** now belong to the **testnetwork** together with a **pe0** device.

Figure 4: L3VPN in NSO



