# Release Notes CWM 1.1

This section contains the following topics:

## Intro

This document provides information about Cisco Crosswork Workflow Manager 1.1, including new features, existing and updated product documentation, API changes, a list of bug fixes, limitations and known issues.

## Version history

| Release date | Version | Release notes |
|---|---|---|
| 9/Jun/2023 | Cisco Crosswork Workflow Manager 1.0 | Cisco Crosswork Workflow Manager 1.0 Release Notes |
| 25/Mar/2024 | Cisco Crosswork Workflow Manager 1.1 | Current ones |

## What's new in CWM 1.1

This section lists the new features and enhancements delivered in Cisco Workflow Manager 1.1.

| Feature | Description |
|---|---|
| Event handling | CWM 1.1 supports interaction with an external Kafka/AMQP/HTTP broker for handling external events. Workflows can now act as either consumers or producers of events. Events can be used to initiate a new workflow, or signal an existing workflow. By leveraging events, workflows can engage in external messaging systems to initiate dialogs with human operators or administrators. The CWM UI provides the ability to define Kafka/AMQP/HTTP events to be used in the system. |
| Compensation support | Workflow authors can now define compensation logic within their workflows. Compensation in workflows consists in undoing or reversing the work of one or more states which have already successfully completed, for example when a further step encounters an error. Note that compensation differs from actions like rolling back a transaction (undo), as it may encompass several logical steps and is integral to the overall business logic that needs to be explicitly defined within the workflow. |
| Logging messages within workflows | Workflow creators can now add custom log messages to their workflows. CWM prefixes each log entry with a workflow name, Run ID, and Worker ID. Messages are published to the system log and can be accessed in the Job Event logs in CWM UI. |
| Job execution scheduler | Support for creating schedules for workflows, both for one-time executions and reoccurring runs. Scheduler features available through CWM UI:- creation of scheduled jobs based on a specific date and time or based on cron expression - listed schedules in the Job Manager tab - ability to delete a scheduleAdvanced scheduler features available through CWM API: - edition of existing schedules, including changing overlap policy - creation of scheduled jobs based on frequency (calendar and intervals) - ability to pause/unpause a schedule |
| Adapter administrationin CWM UI | New features available through CWM UI: - adding, installing and deleting adapters - updating adapter detailsAdapter administration is described in the CWM Operator Guide, Install adapter section. |
| Secrets and resources management in CWM UI | New features available through CWM UI: - adding and deleting secrets and resources - updating secret and resource details - assigning secrets to resources |
| Support for child workflows(subFlowRef) | CWM execution engine extended to allow for execution of child workflows. Compliance enhancement to support subFlowRef from the Serverless Workflow Specification. Until now, only execution of action referring to a functionRef was supported. SubFlowRef definitions are used for invoking child workflows within a parent workflow. With child workflows, you can: - Separate the parent workflow code and workers from the child workflow code and workers. - Split the workload done by the workflow into smaller chunks for better separation of event history. This is especially helpful when your workflow is intended to spawn large numbers of activity executions. In the CWM UI, parent and child workflow runs are linked to make it easier to navigate between them. Job details view of the executed child workflow displays parent Run ID. Event logs of the parent run contain links to child executions. |

| Feature | Description |
|---------|-------------|
| Greater support for Serverless Specification | CWM execution engine now supports the following as per the Serverless Specification: - Action – Condition support. It is now possible to define a condition on an action to control whether that specific action should execute. - Better Error Handling support. You can now use Adapter-defined Error Types in workflows to set a specific error type as not retrievable and add additional logic if an error of a given type occurs. You can either trigger a compensation logic from your workflow or you can switch to another alternative path of your workflow or terminate an execution. - ForEach State – support for Parallel execution. - Support for inline JSON document as input to functions - 1.0 required JSON to be escaped and put into a string which was not user friendly for large documents. Now, input arguments to action can be JSON documents. - Complete support for Compensation. You now have the ability to add compensation logic to your workflows, by adding compensation states to your workflow definitions. Compensations can now be triggered at any state in your workflow and once the compensation is complete your workflow will continue from the state the compensation was triggered. - Actions can now be triggered to run asynchronously. |
| Role-based access control (RBAC) | Introduced a Role-based Access Control (RBAC) mechanism that grants or revokes user access to specific resources in the CWM application. As an administrator, by defining a policy in the API you can manage access to all the available CWM API endpoints for users assigned to a given permission set. |
| Job tags | Ability to define tags for new jobs, and ability filter and sort jobs based on tags. Through CWM UI, you can: - add tags to the new job runs - sort/filter jobs by tags in the Job Manager - display tags on the job details page |
| Bulk import/export of child workflows | The functionality of bulk import and export is available via CWM API. The imported child workflow definitions are visible in the CWM UI.You can bulk-import workflow definition with all its child workflows (defined via subFlowRef) into CWM. Imported parent workflow and child workflows are displayed as separate workflows under the All Workflows list.Bulk export allows you to export in bulk all child workflows of a parent workflow definition, by only providing the parent workflow UUID. The bulk export is possible via the CWM API. |
| Workflow definition tags | Added an option to define sets of tags to workflow definitions, that enables sorting and filtering workflows based on them. The workflow tags are managed via UI, where you can bulk-edit them for a number of workflow definitions. The workflow tags are separate from the job tags. |
| NxF platform v2.0 | CWM 1.1 uses NxF version 2.0 which offers a stronger SSH login encryption mechanism and advanced password complexity and password policy features. |

# Adapter changes

**Note**     For CWM 1.1, adapters need to be built using the latest SDK and XDK 1.1.0 versions.

# New adapters

CWM 1.1 ships with two new pre-built adapters:

| Adapter | Description |
|---------|-------------|
| Generic Email adapter | The Email adapter enables reporting from within workflows by providing basic functionality to send emails using an SMTP (Simple Mail Transport Protocol) server. For the 1.0.0 adapter version, you can use the Sendactivity to send an email with a message defined inside the workflow definition. |
| Generic CLI adapter | The CLI adapter enables workflows to send command line sequences to devices or other applications using the SSH (Secure Shell) interface and the Telnet interface. |

# New adapter XDK application

The XDK extends the capabilities of the Adapter SDK to enable developers to automatically build interfaces and message logic for custom adapters. It comes in two variants:

| XDK | Description |
|-----|-------------|
| XDK for NSO | The Adapter XDK for NSO (cwm-nsox) is an application that helps you generate interfaces and logic for custom adapters intended to interact with the Cisco Network Services Orchestrator(NSO).The primary purpose of cwm-nsox is to reduce the time-consuming and error-prone manual process of constructing paths and payloads required for CWM to communicate with NSO.The tool complements the Adapter SDK and is able to automatically define interfaces in .proto files and implementation of logic in the adapter go module. This is achieved by parsing yang files and points of interestprovided by the Adapter Developer. |
| XDK for OpenAPI | The Adapter XDK for OpenAPI (cwm-oasx) is used to automatically build interfaces and message logic for custom adapters that require communicating with OpenAPI-based systems. With the cwm-oasx tool, you point to an OpenAPI operation defined in JSON, which cwm-oasx will then use to generate a new adapter activity with a predefined rpc and I/O messages in the .proto files, as well as a ready-to-execute implementation in the adapter .go files. |

# Pre-built adapters enhancements

| Feature | Description |
|---------|-------------|
| Support for connection via proxy | Updated the REST client in the existing pre-built adapter modules to support sending requests via proxy. |
| Improved error handling | Improved handling of errors by adding additional layers of wrapped errors using different status codesthat can be specified in a workflow and used for error compensation. |
| More data types handled by adapters | It's possible to define activity payloads for adapters not only in string format but in any data type, for example, JSON object. This is especially useful with multiple-line payloads as it makes handling them more streamlined. |

# Adapter SDK enhancements

| Feature | Description |
|---|---|
| The export-lib command | Introduction of the export-lib command for exporting the SDK go module to your adapter directory or to any specified location so that the SDK lib can be shared among multiple custom adapters. The new command replaces the export-proto command. |
| The update-adapter command improvements | Removed the generate-documentation recipe from the makefile and added this feature to cwm-sdk update-adapter instead. Also, update-adapter is now used to compile the .proto files to .pb.go. This way the generated output will always be in sync with adapter .proto files. |
| Go version upgrade | Updated the SDK go language version to 1.21. Existing custom adapters developed for CWM 1.0 need to be updated to work correctly. The SDK can be used to upgrade existing adapter to be CWM 1.1 compatible |

# Adapter library

| Library | Description |
|---|---|
| SDK | The library for the SDK contains:predefined .proto files for Resource and Secret handling,go module with packages supporting logging, error handling and others. |
| XDK | The library for the XDK contains:predefined .proto files for CLI and Cisco NSO,go module with packages supporting REST, CLI and Cisco NSO. |
| Go version upgrade | Updated the SDK go language version to 1.21. Existing custom adapters developed for CWM 1.0 need to be updated to work correctly. The SDK can be used to upgrade existing adapter to be CWM 1.1 compatible |

# Documentation changes

The following table lists all documents that are provided for Cisco Workflow Manager, along with recent updates for 1.1.

| Document | What is included | Updates |
|---|---|---|
| Get Started Guide | - Overview of CWM - Core Concept glossary with definition - Tutorial for running example workflow using NSO adapter | Added section about CWM upgrade Updated Core Concept part. Updated Tutorial chapter with 1.1 UI changes. |
| Workflow Creator Guide | Instructions for creators of workflow definitions based on Serverless Workflow spec. The guide contains: - Overview of supported features of Serverless Workflow spec - Workflow creation tutorial | Updated Overview chapter – extended the supported features of the Serverless Workflow specification: - added subsection about **SubFlowRef** definitions |

| Document | What is included | Updates |
|----------|------------------|---------|
| Adapter Developer Guide | This guide is dedicated to an audience interested in developing their own custom adapters. The guide contains: - Overview of how the adapter is built - SDK description and usage - A tutorial for creating a custom adapter - XDK for NSO and OpenAPI - description and usage | - Updated contents of 'Use Adapter SDK' and adapter creation tutorial - Added chapter on XDK for NSO and OpenAPI |
| Operator Guide | This guide is dedicated mostly to users of CWM UI and contains: - Overview of CWM UI - How to set up CWM to start workflow execution - How to run/schedule jobs | Added sections about scheduler, managing adapters, secrets and resources in the UI. General update of the UI Overview chapter based on changes in the UI. Minor improvements regarding navigation, consistency and style. - General update of the UI Overview chapter based on changes in the UI, including screenshots and textual content. - Additional step-by-step instructions about bulk workflow tags editing in the Add workflow section. - Refined the Create secrets and resources section: Add secret part. - Added notes about parent-child relations for Event Logs and Job Details. |
| Admin Guide | This guide is dedicated to the administrators of the platform and contains the following: - Installation and upgrade guide - System description, including architecture - Basic API documentation - User management - Platform health and log collection | Rewritten API chapter with new or changed endpoints: - Added the `/schedule` endpoint description - Updated endpoints paths and parameters as described in API changes below<br><br>Enhanced log collection chapter<br><br>Documentation for Role-Based Access Control: creating and updating new access policies.<br><br>Description of policy API endpoint in the Manage via API chapter.<br><br>Updated installation guide. |

# API changes

This section summarizes the changes introduced to the CWM API across multiple versions of the application.

# Table keys

- ✓ - indicates that an endpoint exists for given version.

- **x** - indicates there is no such endpoint for given version.

- **/example/api/path** - shows how existing endpoint path changed from version to version.

# Adapters API

| Endpoint | v10 | v11 |
|---|---|---|
| PATCH/adapter/{adapterId} | x | ✓ |

# Policy API

| Method | v1.0 path | v1.1 path |
|---|---|---|
| GET | x | /policy/{roleId} |
| PUT | x | /policy/{roleId} |
| POST | x | /policy/{roleId} |

# Resources API

| Endpoint | v10 | v11 |
|---|---|---|
| PUT/resource/{resourceId} | ✓ | x |
| PATCH/resource/{resourceId} | x | ✓ |

# Scheduler API

| Endpoint | v10 | v11 |
|---|---|---|
| GET/schedule | x | ✓ |
| POST/schedule | x | ✓ |
| GET/schedule/{scheduleId} | x | ✓ |
| DELETE/schedule/{scheduleId} | x | ✓ |
| PATCH/schedule/{scheduleId} | x | ✓ |

# Secrets API

| Method | v1.0 path | v1.1 path |
|---|---|---|
| GET | /secret/{id} | /secret/{secretId} |
| DELETE | /secret/{id} | /secret/{secretId} |
| PATCH | /secret/{id} | /secret/{secretId} |
| GET | /secret/types | /secretType |
| GET | /secret/type/{type} | /secretType/{secretTypeId} |

# Workflow API

| Method | v1.0 path | v1.1 path |
|---|---|---|
| GET | /workflow/{id} | /workflow/{workflowId} |
| PUT | /workflow/{id} | /workflow/{workflowId} |
| DELETE | /workflow/{id} | /workflow/{workflowId} |

| Endpoint | v10 | v11 |
|---|---|---|
| GET/workflowExport | x | ✓ |
| POST/workflowImport | x | ✓ |

# Bug fixes

## Backend bug fixes

| CDETS ID | Title | Description |
|---|---|---|
| CSCwi23923 | Jobs failing after using PATCH/schedule API | When updating existing schedules using PATCH/schedule, jobs are now processed correctly when the schedule is triggered. |
| CSCwi33805 | Creating a secret without input data via API is possible | Added a validation for the POST/secret API so that if any input data is missing or additional data is added to the body, a validation error is displayed. |
| CSCwi39979 | API user authorization doesn't work when updating a policy using PUT/policyAPI | Fixed the issue of user authorization in the API in case of the PUT/policy endpoint. |
| CSCwf38815 | Pod memory usage always increasing | Introduced a fix for Kubernetes pod memory usage persistent increase. When running tests over a long period of time, Workflow History, Postgres and Worker Manager pods are now under control regarding memory usage. |
| CSCwf43739 | Duplicated output when using arrays as output from Parallel State branches | When using branches with a Parallel State in a workflow definition, each branch saved a result to a different variable, causing a duplicate. Fix delivers an update so that each entry is saved once instead of several times (i.e. if you use three branches, there is just one entry produced for each with the same content). |
| CSCwf43745 | Workflow blocks if some of the action in Parallel State fails | For workflows with a Parallel State, fixed an error which caused the whole workflow execution to block in case some of the actions in the Parallel State are failing. |

| CDETS ID | Title | Description |
|----------|-------|-------------|
| CSCwf51419 | GET adapter API returns 204 success response with no content | When using GET method on the /rest/api/v1/adapter path, the CWM API returned 204 success code, but no content. Now, it case there are no activities, the CWM API either returns an empty array with 200 code response or, in case of any error, a failure response with a correct error code. |
| CSCwf80391 | Possible to start job referencing non-existent workflow | A bug in the application made it possible to start a job referencing a workflow that didn't exist in CWM database. Now, the response for such requests is an error code followed by a message explaining there is no matching workflow in the database. |
| CSCwh01281 | Always created a Default Job for new schedule | When creating a new cron schedule, the default job was created automatically and kept running until deleting the schedule. Issue is fixed by making the CWM API only return workflow types "ExecuteWorkflow" and "ExecuteWorklfowNameAndVersion" from the Execution Engine. |
| CSCwi28014 | Duplicated content returns while using JQ expression | Fixed the incorrect behaviour that occurred while using JQ expressions for querying workflow definition. Previously, when using fromJSON, it returned duplicated items in the result. The merging strategy has changed for lists from 'append' to 'skip' identical so that only unique objects are left in the lists. |
| CSCwi23922 | Possibility to update a workflow with an empty name or empty version | Previously, a workflow definition could be updated with an empty name or an empty version. Now it's not possible to update a workflow definition without a defined version or name and an appropriate error is thrown upon such attempts. |
| CSCwi35956 | CWM upgrade does not work correctly | Fixed the CWM upgrade feature, which worked incorrectly due to a service name change issue. |
| CSCwi28004 | CWM engine sends incorrect resourceId to the adapter | Fixed the issue with the incorrect resourceId sent to the adapter. CWM engine was sending the wrong resourceId to states other than the current resourceId value – the problem appeared when a workflow definiton used a different resourceId of the same adapter in multiple states. |
| CSCwi21785 | Adapter SDK: Updating an adapter removes line breaks | Fixed an issue with the cwm-sdk update-adapter command so that it doesn't wipe out all '\n' in pre-existing activities. |

# UI bug fixes

| CDETS ID | Title | Description |
|----------|-------|-------------|
| CSCwi90675 | Resource creation fails when configuration is empty | Fixed the behavior of the OtherSettings field used when creating a Kafka resource. It is now possible to leave this field empty and successfully create a resource. |

| CDETS ID | Title | Description |
|---|---|---|
| CSCwf31235 | Incorrect encoding for worker names | Fixed the problems with encoding in the worker names caused by the special character "/". |
| CSCwh63208 | Space in schedule ID blocked deletion | The space in the schedule ID caused incorrect behaviour during the deletion attempt and made the schedule impossible to remove from CWM. |
| CSCwh60544 | Schedule type mismatch in Scheduled jobs tab | In the Scheduled job tab, the schedule type under the Type column had not been assigned correctly when the schedule was created via API as a calendar or interval type. After the fix, the schedule type displays consistently for all schedules, no matter if they were created via UI or via API. |
| CSCwf49549 | Deselect all table rows not working in worker table | Fixed the worker table Deselect all checkbox to work as intended. |
| CSCwf99753 | Incorrect styling for delete button in Delete schedule modal popup | Corrected the padding for the Delete button in the Delete schedule modal to have more padding on the left and the right. |
| CSCwh01333 | Cancel button is displayed for already cancelled/terminated jobs | In the Jobs tab, for jobs already cancelled or terminated, the Cancel button is now not available. |
| CSCwf15317 | Icons in header when active has wrong color | When active, header icons now don't change color, but remain white instead. |
| CSCwf40886 | Dummy adapter activities are incorrectly partially checked | When showing activities for multiple products from the same vendor, empty/dummy activities now remain unchecked. |

# Known issues

| Issue | Description |
|---|---|
| Adapter Installation issue if server crashes mid adapter install | If an adapter is being installed, and whilst the installation is in progress, the API server crashes - it has the potential to leave the system in an unstable state. In this instance, re-installing the adapter results in an error stating the adapter is already installed, but the installation is not completed, and the same adapter cannot be deleted. |
| Concurrent map writes error | This issue has been observed intermittently, in a very limited number of cases where multiple activities (where parallel state or async activity execution may be used) are running in parallel and each activity is trying to update the workflow data. The issue causes the workflow worker pod to crash and restart. Upon restart, the workflow continues execution. However, a small chance remains of encountering the same issue again. |