



Operations Orchestration

Software Version: 10.80

Windows and Linux

Get Started

Document Release Date: September 2017

Software Release Date: September 2017



Hewlett Packard
Enterprise

Legal Notices

Warranty

The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from Hewlett Packard Enterprise required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notice

© September 2017 Hewlett Packard Enterprise Development LP

Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark of The Open Group.

This product includes an interface of the 'zlib' general purpose compression library, which is Copyright © 1995-2002 Jean-loup Gailly and Mark Adler.

Documentation Updates

To check for recent updates or to verify that you are using the most recent edition of a document, go to: <https://softwaresupport.hpe.com/>.

This site requires that you register for an HP Passport and to sign in. To register for an HP Passport ID, click **Register** on the HPE Software Support site or click **Create an Account** on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HPE sales representative for details.

Contents

Get Started	4
Get Started with OO Installation	4
Key Concepts	5
Functional Architecture	6
OO Studio	7
OO Central	7
OO Remote Action Service (RAS)	8
OO Content	9
OO Personas	9
Operations Orchestration Processes	12
Authoring Flows	12
Promoting Content	14
Running and Monitoring Flows	16
OO Entities	17
Worker Groups and Group Aliases	24

Get Started

If this is the first time you have installed Operations Orchestration (OO), it is highly recommended to make yourself familiar with the basic OO *"Key Concepts" on the next page*. Then, you can get started with configuring your application.

OO is a system to create and use actions in structured sequences called flows. These structured sequences maintain, troubleshoot, repair, and provision your IT resources.

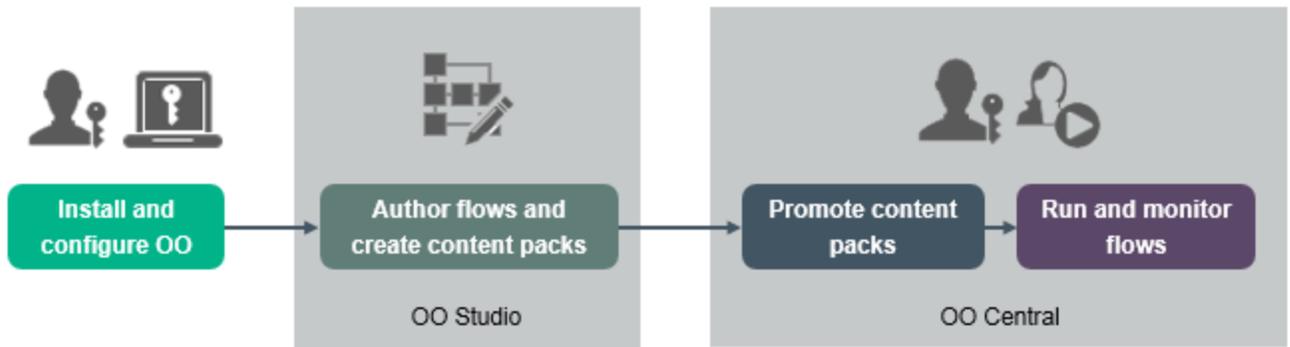
OO is composed of four main functional components:

- Studio
- Central
- Remote Action Service (RAS)
- Content

Together, the components of OO enable you to manage various services and devices across the organization and across their life cycle.

Get Started with OO Installation

The flowchart below displays the major high-level steps in the OO workflow.



- Configure your installation of OO
- Create flows in OO Studio and create a content pack

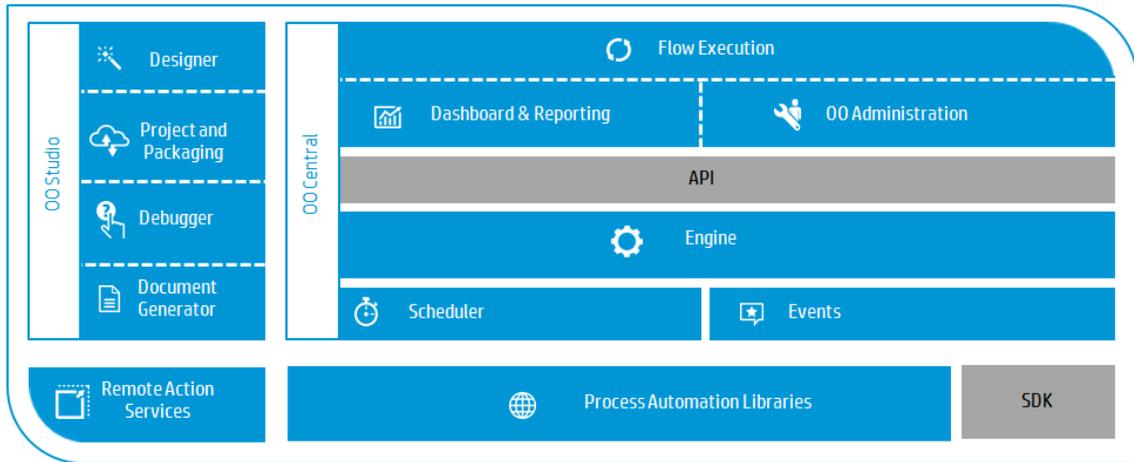
- Promote the content pack to OO Central
- Run and monitor flows in OO Central

Key Concepts

Following are the key features that make OO an industry-leading system:

- Helps you to maintain, troubleshoot, and repair your resources.
- Checks the health of, diagnose, and repair the networks, servers, software applications and workstations.
- Helps you to perform repetitive tasks, such as checking status on internal or external web site pages.
- Enables element automation, runbook automation, integrated automation, and automation of the business services.
- Offers an intuitive drag-and-wire capability to design, create, share, and customize flows.
- Offers out-of-the-box content to manage operating systems, databases, app/web servers and networking platforms.
- Powerful execution mechanism, which enables high performance. Central can be scaled to reach high numbers of steps per second.
- Provides REST APIs for the entire set of Central functionality. Every functionality that is exposed in Central is implemented over a public REST API.
- Provides interactive execution and UI-embedding capabilities, which enable you to interact with the system in a much more usable fashion.

Functional Architecture



OO Studio

OO Studio is a desktop-based application that is used by flow authors to create the OO flows. Studio enables the author to design flows, debug them, and package them. It provides automation via code capabilities, such as integration to Source Control Management software, project separation, and multi-authoring.

Designer

OO Studio provides a drag-and-wire graphical designer to formulate flows out of various operations and sub-flows.

Debugger

OO Studio provides a debugger to test the designed flows. The debugger reflects the behavior of the flow in the Central environment.

Project and Packaging

OO Studio lets you break down your set of authored content into granular projects. Each project groups together contents of a similar nature, for example, according to functionality, development owner, geographic location, and release time lines.

In addition, OO Studio lets you package each project to a content pack. The content pack is the read-only run time artifact for the project sources. The content pack is the artifact that is deployed to OO Central.

Document Generator

OO Studio lets you generate documents for every flow or group of workflows. These auto-generated documents include information about the flow, including its graphical presentation.

OO Central

OO Central is the run time environment of OO OO. It is used for running flows, monitoring the various runs, and generating reports. It has a web-based UI and a set of APIs, which are accessed by the administrators, end users, and integrators.

OO Central is available as a WAR file to be included within an application server or as a standalone installation.

Flow Execution

OO Central provides execution capabilities for the deployed flows. The execution is done from a web-based UI and from RESTful APIs. The execution capabilities include browsing the flow library, launching an execution, and tracking that execution until its end.

Dashboard & Reporting

OO Central provides reports about the various executions. These include running flows, finished flows, successful and failed flows, and so on.

In addition, the rich set of RESTful APIs let you to gather execution information and to compound a more advanced set of dashboard and reports.

OOAdministration

OO Central provides administrators with the means to administrate OO from system and operational perspectives. It covers areas such as setting permissions, defining system components, configuring LDAP, and more.

API

OO Central provides a complete set of RESTful APIs for every capability. In fact, every capability of the web UI is implemented on top of public RESTful API, enabling you to implement your own web UI and to combine OO capabilities within your web applications.

Engine

The engine works behind the scenes of Central. This is the back end component that processes and manages the entire flow execution. The engine manages the step execution, the persistency, and the manual interaction with the users.

Scheduler

OO Central includes an out-of-the-box scheduler. This enables the OO administrator to define various recurrence patterns for flow executions, to track, and to manage them.

OO Remote Action Service (RAS)

The OO RAS enables execution in remote data centers and networks. The RAS interacts with Central and polls it for operations to execute.

In version 10.60 and later, you can configure RASes to initiate the connection to Central or to wait for Central to initiate the connection.

- If you set up the RAS to initiate the connection, you need to open ports for inbound communication only in Central. To achieve high availability of RASes, you simply add another RAS and point it to Central.
- If you set up the RAS so that Central initiates the connection, you must configure Central to register the RAS by providing all required information: host/IP, port, and so on. Until you do this, the RAS will be idle, waiting for Central to initiate the connection.

In addition, the RASes support a grouping mechanism, which enables you to correlate between a step in the flow and the type of the RASes that can execute this step. Therefore, the binding between the flow steps and the RASes is dynamic.

For more information, see the *OO Architecture Guide*.

OO Content

OO provides a rich set of out-of-the-box operations and flows that enable you to author complex flows, orchestrating various services. The OO content is delivered as a set of granular content packs that you can download, deploy, and manage individually. These are the **Process Automation Libraries**.

In addition, OO provides wizards for generating additional content over other services such as Web Service Wizard.

OO provides Java and .NET SDK, to enable developing custom content and operations.

By using the OO content, you can build a rich set of process automation libraries.

OO Personas

Operations Orchestration (OO) personas represent the people responsible for performing particular tasks in the OO workflow. Icons are used in the OO documentation to identify the persona who performs a task.

Please note that when we suggest a persona for a task, this is a recommendation only, as practices may vary across different workplaces. For example, some workplaces may have an integrator, who is responsible for promoting all content packs, while we have assigned this task to the Op Admin. In other workplaces, a flow author may also be an action developer.

Major Personas

The following major personas play a key role in the OO processes.

Flow Author

	<p>The Flow Author creates and debugs flows in OO Studio. The Flow Author leverages the OO out-of-the-box content, the content developed by Action Developers, and the utilities provided by SMEs.</p>
---	--

Operational Administrator (Op Admin)

	<p>The Op Admin is responsible for the day to day operation of OO. This includes configuration, maintenance, content pack promotion, and setting permissions for OO users.</p> <p>The Op Admin troubleshoots any failures in the production environment and either solves the problem or routes it to the System Administrator, to HP Support, or to the Flow Author (depending on the issue).</p>
---	--

System Administrator (Sys Admin)

	<p>The System Administrator is responsible for the OO hardware and software. The Sys Admin installs and patches OO (Central and RASes), and is responsible for the correct functioning of the application from the system perspective, dealing with things such as CPU, memory, and OS environment.</p>
---	---

End User

	<p>The End User triggers and monitors flows. The End User can access entitled OO flows directly through Central or indirectly through an embedded web UI in another application.</p>
---	--

Minor Personas

The following minor personas play a supportive role in the OO processes. It is possible to use OO successfully without requiring the services of an action developer or SME. However, these personas can enrich the content of your flows and operations.

Action Developer



The Action Developer is an expert in code development, who creates actions that can be used in flows. The Action Developer creates these actions mainly in Java and occasionally in .NET.

Subject Matter Expert (SME)



The SME may be an expert in DBA, Linux, or a specific application. The SME creates focused automations (through scripts, executable files, and so on) for day-to-day operations in their area of expertise. These focused automations can be leveraged into the OO flow authoring.

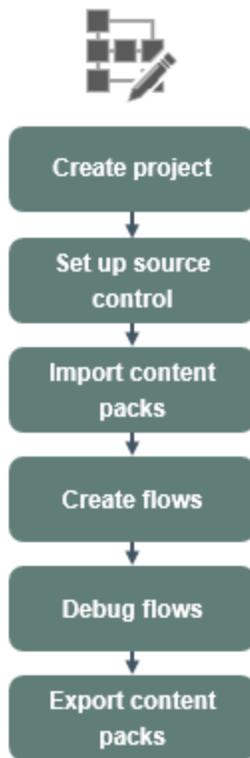
Operations Orchestration Processes

The major processes in Operations Orchestration OO are:

- Authoring flows
- Promoting content
- Running and monitoring flows

Authoring Flows

The authoring process is performed by the Flow Author .



Step 1: Create a new project

 Create a project to contain the flows, operations, folders, and configuration items for a business purpose.

Step 2: Set up source control management



Set up either the SVN or Git source control management system.

See "Working with Source Control in Studio" section in *OO Use Guide*.

Step 3: Import a content pack



Import any content packs that you need, so you will be able to copy the relevant content into your project.

Note: The first two steps do not have to be performed in this order. It is possible to import a content pack before creating the project.

See "Importing Content Packs to a Project" section in *OO Use Guide*.

Step 4: Create a flow



Put together the operations, inputs, transitions, responses, and return steps that make up your flow.

See "Creating a Flow – Step-by-Step" and "Advanced Authoring" sections in *OO Use Guide*.

Step 5: Run and debug the flow



Validate your flow in the Debugger.

See "Testing and Debugging a Flow" section in *OO Use Guide*.

Step 6: Release the content, packaged into a content pack

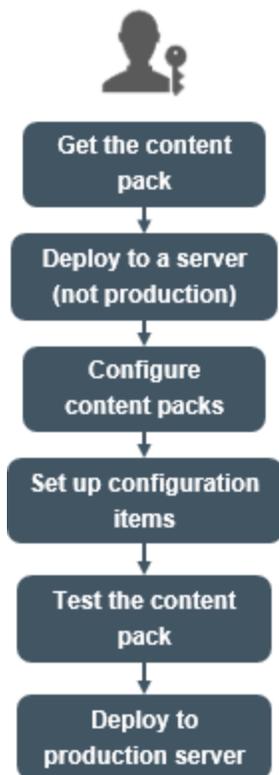


Package your project into a content pack, containing the flows, operations, actions, and configuration items, in order to promote it to Central.

See "Exporting a Content Pack" section in *OO Use Guide*.

Promoting Content

The promotion process is performed by the Op Admin .



Step 1: Get the Content Pack From the Flow Author

1.  The Flow Author creates a content pack in Studio.
2.  The Flow Author saves the content pack to the Artifact Repository.
3.  The Flow Author shares the content pack with the Op Admin/Promoter.

Step 2: Deploy to a non-Production Server

1.  The Op Admin receives the content pack from the Flow Author.
2.  The Op Admin deploys the content pack on the a non-Production server, for example, the

Staging server.

Step 3: Configure the Content in the Content Pack

In this step, the Op Admin adjusts the content pack to the environment by configuring the content in it.

1.  The Op Admin sets the content permissions on the flows.
2.  The Op Admin sets the persistence level and run timeout for the flows.

Step 4: Set up the Configuration Items in the Content Pack

In this step, the Op Admin adjusts the content pack to the environment by setting up the configuration items in it.

1.  If the content pack includes system and system properties, the Op Admin assigns values to these in the content pack.
2.  The Op Admin maps the aliases of the worker groups to actual worker groups.

Step 5: Test and Troubleshoot the Flows in the Content Pack

1.  The Op Admin runs each flow from **Content Management > Flow Library** and checks whether it runs properly.
2.  If a flow run fails, the Op Admin can drill down into a flow to identify the problems.

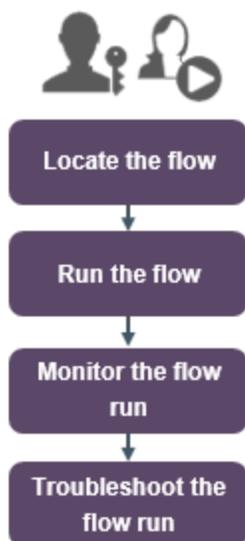
Step 6: Deploy to the Production Server

1.  If required, the Op Admin adjusts the configuration of the content pack, for the Production server. For example, it may be necessary to map the worker group aliases and system accounts differently for this server.
2.  Finally, the Op Admin deploys the content pack to the Production server.

For more information, see the *Central User Guide*.

Running and Monitoring Flows

The process of running and monitoring flows is performed by the Op Admin  or the end user 



Step 1: Find the flow that you want to run

 The Op Admin/End User locates the flow either from the Flow Library or the Flow Launcher. Go to the **Content Management > Flow Library** module or to the **Run Management > Flow Launcher** module.

Step 2: Run the flow

 The Op Admin/End User runs the flow.

 Alternatively, the Op Admin/End User schedules the flow to run later.

Step 3: Monitor the flow run

 The Op Admin/End User tracks the flow run in the Run Explorer.

 If required, the Op Admin/End User performs actions on flow runs, such as pausing, resuming, and canceling flow runs.

Step 4: Troubleshoot the flow run



If required, the Op Admin investigates any problems with the run.

For more information, see the *OO Use Guide*.

OO Entities

Operations Orchestration (OO) Studio

OO Studio is a standalone authoring program used for creating, modifying, and testing flows.

OO Central

OO Central is the runtime environment of OO. It is used for running flows, monitoring the various runs, and generating reports.

OO Content

OO contains an out-of-box content library with over 4,000 flows and operations, accelerator packs, and integrations.

RAS

A RAS is a remote action server, containing a [worker](#) and a remote protocol for connecting with Central.

Worker

Workers are the software components that are responsible for executing flows. A worker connects to Central to obtain tasks (flow execution messages) to process. For more information, see "[Worker Groups and Group Aliases](#)" on page 24.

Worker Group

A worker group is a logical collection of workers. A worker may belong to more than one group simultaneously. You can define a worker group in Central. For more information, see "[Worker Groups and Group Aliases](#)" on page 24.

Group Alias

Group aliases let you separate between assigning an operation to a worker during authoring time and in the run time environment. In Studio, the author can define an operation to execute on a group alias rather than a worker group. In Central, the Op Admin maps the group alias to the actual worker group. For more information, see ["Worker Groups and Group Aliases" on page 24](#).

Flow

A flow is the composition of [steps](#) and [transitions](#) together forming a set of actions that are linked by decision-making logic in order to automate tasks. For example, health checks, troubleshooting, or any other repetitive IT support tasks.

Flows are created in Studio and run in Central.

Subflow

A subflow is a flow that is used as a step within another flow. The flow that contains the subflow step is known as the parent flow. It is possible (and recommended) to debug the parent flow and subflow separately in OO Studio.

For more information, see "Creating a Subflow Within a Flow" in *OO Use Guide*.

Flow Run

A flow run is a single execution of a [flow](#) in Central. Flow runs collect data, enabling you to analyze the performance of your IT system.

Content Pack

A content pack is a file containing operations, flows, actions (Java-based or .Net based), localization data, and configuration items. Content packs are deployed to the Central server and stored in the database.

A content pack can be created in Studio by an author, or it can be provided by HP or a third party.

provides a wide range of content packs with prepackaged flows, out-of-the-box integrations, and orchestration of multiple systems. The content packs are available from LN.

Role Permission

A permission is a predefined ability to perform a task. Central includes a set of permissions that can be assigned to [roles](#).

For example, the **Schedule** permission grants the ability to view and create flow run schedules.

Role

A role is a collection of [permissions](#).

For example, the **Flow Administrator** role may be assigned the **View Schedules** permission and the **Manage Schedules** permission.

User

A user is an object associated with a person (or application identity) representing the person and defining their authorization.

[Roles](#) are assigned to users, to define the actions they have permission to perform in Central. For example, the user Joe Smith may be assigned the **Administrator** role.

It is possible to configure different kinds of users:

- **LDAP users** log on to Central using their LDAP user name and password.
- **Internal users** log on to Central using the user name and password that was set up in Central.
- **LWSSO** users can enter the Central application directly, bypassing the Central logon screen, if they have logged onto another product web client that has LWSSO enabled.

When an internal user and an LDAP user with the same role are logged in, there is no difference between their permissions..

Note: It is recommended to use LDAP users rather than internal users, because LDAP users are secured according to policies implemented by the LDAP provider.

Content Permission

Content permission is permission to view or run individual flows or the flows in a particular folder.

Users who have been assigned a specified role will be able to access the flows according to the content permissions assigned to their role.

For example, users with the **Administrator** role may be entitled to view and run all the flows in the system, while users with the **User** role may be entitled to run certain flows, and have view permission for others.

For more information about OO permissions, see the *OO Use Guide*.

Studio Project

A project is a directory in Studio, containing the flows and operations that a flow author is working on. In order to create a flow, a flow author must be working within a project.

Every project contains two folders:

- **Library** - holds all the flows and operations in that project.
- **Configuration** - holds other OO objects (filters, scriptlets, system properties, and so on) that you can use to process operation results, create reports, and facilitate the running of flows

When flow authors complete a project, they package it into a content pack for promotion to the Central server.

For more information, see "Managing Projects" in *OO Use Guide*.

Action

An action is a software component invoked by the OO run time in order to achieve a certain task. An action can be developed using the Java or .NET programming language. Actions are packaged in the form of .JAR or .DLL files, and can be used as the basis for [operations](#).

In addition to the actions developed by OO, an action can be developed at your own site or provided by a third party vendor aiming to extend the capabilities of OO.

Operation

An operation holds the [inputs](#), [outputs](#), [responses](#), and other properties that are required to perform a certain action. Operations are available for the flow author to use as steps inside a flow. A rich set of out-of-the-box operations is available with Studio.

For example, one operation checks a web page to see whether it contains specific text, and another operation copies a file.

Step

Steps are the building blocks of a flow. A flow author creates a step by dragging an operation onto the authoring pane.

A step is an instance of the [operation](#), and it inherits the [inputs](#), [flow variables](#), and other properties of the operation. A step can be modified without affecting the original operation.

For more information, see "Creating Steps in a Flow" section in *OO Use Guide*.

Response

Responses are the possible outcomes of an operation. For example, a **Read Web Page** operation may have three possible responses:

- The web page can't be found (failure)
- The page is there and the desired text is present (success)
- The page is there but the text isn't present (partial success - needs another action)

For more information, see "Setting Responses" section in *OO Use Guide*.

Transition

A transition is the connection from an operation's **response** to one of the possible next steps.

For more information, see "Creating Transitions" section in *OO Use Guide*.

Input

Inputs give the operation the data that it needs to act upon. For example, an operation to check a web page needs to know which page to check and what text to look for.

Inputs can be:

- Set to a specific value
- Obtained from information gathered by another step
- Entered by the person running the flow, at the start of the flow

For more information, see "Creating Input section" in *OO Use Guide*.

Output

Outputs are data produced by operations—for example, success code, output string, error string, or failure message.

For more information, see "Setting Operation Outputs" section in *OO Use Guide*.

Primary Output

Primary output is the output used to populate the step's primary result. The primary output supplies a value to an input whose assignment is **Previous Step's Result**.

Result

Results are **output** produced by a step in a flow. For each step, you decide which of the operation's outputs you want to retain.

Outputs are not automatically retained in the flow. If they were, this could affect performance, by slowing down the flow with unnecessary data.

Raw Result

The raw result is *all* of the operation's return code, data output, and error strings.

Variable

Variables make data available in a flow run. There are two kinds of variable:

- **Global variables** are name/value pairs that come from system properties and selection lists, and are provided to a flow run at its start. You can also create a global variable by using the **Set Global Flow Variable** operations from the Base content pack.
- **Flow variables** are available only for the flow within which they are defined. You can assign a value to a flow variable from:
 - A step's result - for example, a step with an operation to count hits can be set to store the result in a flow variable
 - An input value - for example, a step where a user needs to enter an IP address can be set to store the address as a flow variable
 - A scriptlet - for example, a scriptlet that evaluates data that is returned from a step can be set to store the data in a flow variable

For more information, see "Working with Variables" section in *OO Use Guide*.

Context

The context is a container that holds various values that can be exchanged with a step at various points. There are two kinds of context:

- **Local context** exists for the duration of the step.
- **Global context** exists for the duration of the flow.

You can pass values to and from the local or global context.

Output Fields

In a **subflow** (a flow within a flow), output fields are used to store a step's results so that the result data is accessible to operations, transitions, and prompts in the parent flow.

Scriptlet

Scriptlets (written in JavaScript) are optional parts of an operation that you can use to manipulate data from either the operation's inputs or results for use in other parts of the operation or flow.

For more information, see "Using Scriptlets in a Flow" section in *OO Use Guide*.

Soft Copied Operation

When you copy an operation that is linked to an action plugin jar file, the copied operation continues to reference the original operation. If the action plugin jar file is upgraded—for example if the name of the JAR or the class is changed—when you update the original operation to call the new version, the copied operations are all updated automatically. This is known as a **soft copy**.

For more information, see "Copying Flows and Operations" section in *Get Started* guide.

Hard Copied Operation

In a **hard copy** of an operation, the copy is directly linked to the action plugin in the same way that the original operation was. When the action plugin jar file is updated—for example if the name of the JAR or the class is changed—this needs to be updated in all the hard copied operations.

For more information, see "Copying Flows and Operations" section in *Get Started* guide.

Configuration Item

Configuration items are OO objects (filters, scriptlets, system properties, and so on) that you can use to process operation results, create reports, and facilitate the running of flows.

Worker Groups and Group Aliases

Worker Groups

Many deployments can benefit from having more than a single worker in a specific environment. For example, this could be helpful if you are managing a remote data center in which you need workers to be able to withstand the action execution load, or simply for high availability of the workers in that data center.

Therefore, Operations Orchestration (OO)10.x includes the notion of a “worker group”. A group is a logical collection of workers and a worker may belong to more than one group simultaneously. Having a worker group helps prevent bottlenecks, because a step can run on any of the workers in the group.

Groups can be created and workers assigned to groups in Central. For more information, see the *OO Central User Guide*.

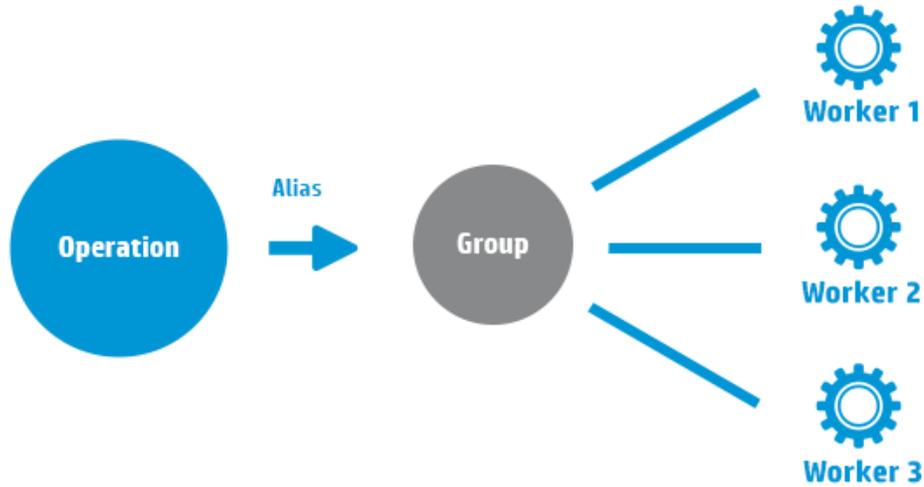
Note for users who have upgraded from version 9.x:

In previous versions, a load balancer was required to balance the load between multiple RASes, which Central would know as a single logical RAS. In version 10.x, there is no longer a need for a load balancer in front of the RASes. Each RAS communicates with Central.

Group Aliases

In OO 10.x, in addition to worker groups, there is another level of flexibility in the assignment of an operation to a worker (or workers). Group aliases let you separate between assigning an operation to a worker during authoring time and in the runtime environment. At authoring time, an operation can be defined to execute on a group alias rather than a group. The alias is then mapped to a group in the runtime environment at the Central level. As a fallback, if the group alias is identical to the group name, it is mapped automatically to that group. Optionally, at triggering time, it is possible to override the group alias and map the operation to a different worker group.

Using this functionality, the OO administrator can let the author use an alias, for example, **WorkerAlias1**. In Central, the administrator maps the alias to the actual worker group. When the content pack is imported to Central, there is no need for the administrator to dive into the flows and modify the worker assignment manually.



Note for users who have upgraded from version 9.x:

In previous versions, the author of a flow was exposed to the runtime topology when a RAS was defined for a specific step in the flow. In this situation, you could not change the host name of the runtime RAS without changing it in all the flows that used it, or it had to be “overridden” at runtime.

