



**Hewlett Packard**  
Enterprise

# HPE Operations Orchestration

Software Version: 10.60  
Windows and Linux Operating Systems

## Concepts Guide

Document Release Date: May 2016  
Software Release Date: May 2016

## 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. HPE 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 HPE 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

© Copyright 2016 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 download the most recent edition of a document, go to <https://softwaresupport.hp.com>.

# Contents

OO Concepts .....	4
What is HPE Operations Orchestration? .....	4
Why HPE OO? .....	4
Functional Architecture .....	8
HPE OO Studio .....	9
HPE OO Central .....	9
HPE OO Remote Action Service (RAS) .....	10
HPE OO Content .....	10
OO Personas .....	11
HPE OO Processes .....	13
Authoring Flows .....	13
Promoting Content .....	15
Running and Monitoring Flows .....	16
OO Entities .....	18
Worker Groups and Group Aliases .....	24

# OO Concepts

This document introduces the basic components and processes in HPE Operations Orchestration.

- [What is HPE Operations Orchestration?](#) ..... 4
- [Why HPE OO?](#) ..... 4
- [Functional Architecture](#) ..... 8
  - [HPE OO Studio](#) ..... 9
  - [HPE OO Central](#) ..... 9
  - [HPE OO Remote Action Service \(RAS\)](#) ..... 10
  - [HPE OO Content](#) ..... 10

## What is HPE Operations Orchestration?

HPE Operations Orchestration (**HPE OO**) is the industry-leading solution for IT process automation and runbook automation.

HPE OO is a system for creating and using actions in structured sequences (called flows) which maintain, troubleshoot, repair, and provision your Information Technology (IT) resources by:

- Checking the health of, diagnosing, and repairing, networks, servers, services, software applications, and individual workstations.
- Deploying applications, patching, and maintaining them by checking client, server, and virtual machines for required software and updates, and, if needed, performing the necessary installations, updates, and distributions.
- Performing repetitive tasks, such as checking status on internal or external web site pages.

## Why HPE OO?

### Why Orchestration?

In many companies, the following issues can result in poor service quality, delayed time-to-market, and high operating costs:

- Incidents – floods of alerts, unnecessary escalations
- Change and releases – too many manual errors, lack of audit trails
- Process management – need for processes for complex tasks, for example, disaster recovery
- Virtualization – inconsistent management of physical and virtual assets

Orchestration enables automation:

- Element automation – automation of either networks, servers, or storage, spanning tasks from provisioning and change management to compliance enforcement and reporting
- Runbook automation – automation of common and repeatable IT processes across all infrastructure tiers, IT groups, and systems
- Integrated automation of applications, servers, networks, storage, and common processes across the data center
- Automation of the business service, with continuous control of each phase of the service life cycle, across the data center and client end points, from automated operations to monitoring and ticketing

## Key Benefits of HPE OO

The key benefits of HPE OO include:

- Reduced operational cost with automation of common tasks and processes
- Improved service quality with accelerated incident resolution
- Improved audit compliance through documentation generation and reporting
- Integration with current IT environment to ensure minimal impact on procedures and tools

## Benefits for Flow Authors

### **Easy-to-use**

HPE OO Studio offers an intuitive drag-and-wire capability to design, create, share, and customize flows. The drag-and-wire visual interface enables rapid time-to-value. A visual flow debugger makes it easy to debug flows.

### **Out-of-the-box Content**

HPE OO offers out-of-the-box content to manage operating systems, databases, app/web servers and networking platforms. You can utilize out-of-the-box integrations with common HP and third-party systems management tools, such as ticketing, monitoring and event consoles, virtualization, CMDB, and data center automation.

### **Standalone Studio**

HPE OO Studio is a standalone tool that doesn't require a connection to Central. All of its repository operations are available offline. If a source control interaction is required, you decide when the interaction occurs. In this manner, remote teams can use various standalone Studios, and it is even possible to author outside of the office network.

### **Standard Source Control Integration**

HPE OO Studio integrates with standard source control software. Even the out-of-the box solution is based on common source control software (SVN or Git). This means that the common capabilities of source control software are available for Studio, so you can connect and use your organization's source control software. This also means that the automation code can reside with other source code and follow the same life cycle (automation as code).

## Multi-Authors and Multi-Geographies

HPE OO Studio works offline and leverages standard source control software to share work between multiple and distributed authors.

### Annotation-Based Content

HPE OO Studio includes '@Action' annotations that can be added directly to your own code. This means that your code can be leveraged to be OO content and still be tested in the context of your development framework.

### Fine-Grained HP Content

HPE OO content has been organized into a set of about 15 content packs. Each content pack provides flows and operations for a functional domain. You have control over which content packs to download and which to deploy. You can use only what you really need and ignore others.

### Fine-Grained Customer Content

In HPE OO Studio, your content can be separated into projects and managed separately for each author and group. This gives you complete flexibility in defining the flows that are grouped together and the workspace of each author. Different authors get a focused development environment with the flows that are relevant to them, and do not affect other authors' flows.

### Remote Debugging

HPE OO Studio allows the author to connect to a live Central environment and achieve full debugging capabilities for that environment. This enables multiple authors to test their flows on a real environment and control the testing from within the Studio debugging environment. Flow debugging does not affect the content that is deployed on Central and does not require pre-deployment; however, it provides full logging information in Central and is protected by entitlement.

# Benefits for Administrators

## UI Experience

The Central web UI has been designed to reflect common HPE OO use cases. Functionalities for the same roles are grouped into the same areas. The permissions assigned to the logged-on user define the parts of the UI that the user sees. By assigning these permissions to roles, the Administrator can formulate a dedicated UI experience for each group of users.

## Live Scale Out

In HPE OO, you can add a component while the system is live, without restarting the other components. You can scale out the Central nodes to achieve high availability and accelerate execution throughput. Simply install a new Central node and point it to the same database schema. You can add additional RAS instances via the worker group mechanism, without the need for a load balancer. You can add a RAS instance while the system is live, without needing to change flow.

For more information about how to scale out your HPE OO deployment, see the *HPE OO Architecture Guide*.

## High Performance

HPE OO has a powerful execution mechanism, which enables high performance. HPE OO Central can be scaled to reach high numbers of steps per second.

For more information about performance, see the *HPE OO 10 Benchmark* document, available on HPLN.

## Highly Parallelized Runs

The HPE OO execution mechanism is based on an asynchronous execution mechanism, which enables large bursts of executions. Central supports launching 100 flows/sec and there is no limit to the number of parallel executions.

## Automatic Content Distribution

In HPE OO, content binaries are automatically distributed to the appropriate RASes. When there is a change in their version, the system automatically distributes them to the relevant RAS upon their first use.

## Isolated Content Dependencies

HPE OO lets you use different versions of third-party libraries for every content pack. You can use your own required third party versions and these are not affected by HP changing the out-of-the-box version.

## Simple RAS Management

HPE OO uses a group mechanism that separates the logical notion of the target RAS (which is what the author is aware of) and the physical notion of the target RAS (which is what the op admin is aware of). For every environment, the op admin can map between the logical and physical notions. Therefore, promotion of content between environments does not require content changes, and the addition of a RAS doesn't require going back to the author.

# Benefits for Integrators

## Complete REST API

HPE OO provides a complete REST API for the entire set of Central functionalities. Every functionality that is exposed in Central is implemented over a public REST API, enabling you to utilize the same set of APIs to integrate your systems with HPE OO.

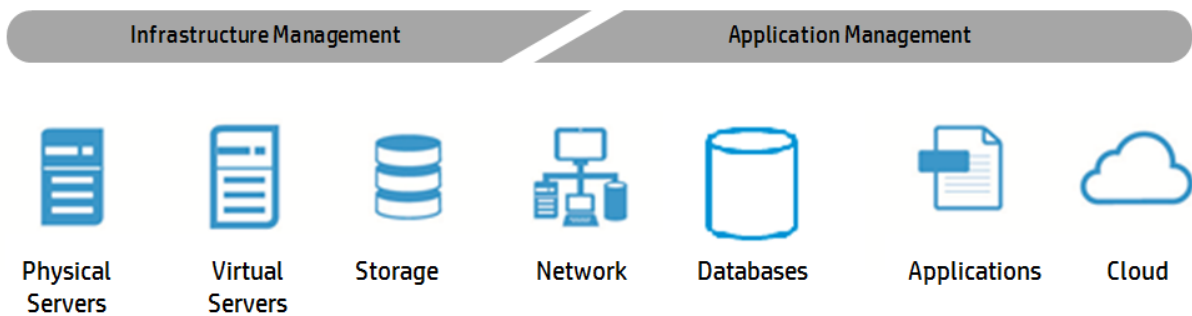
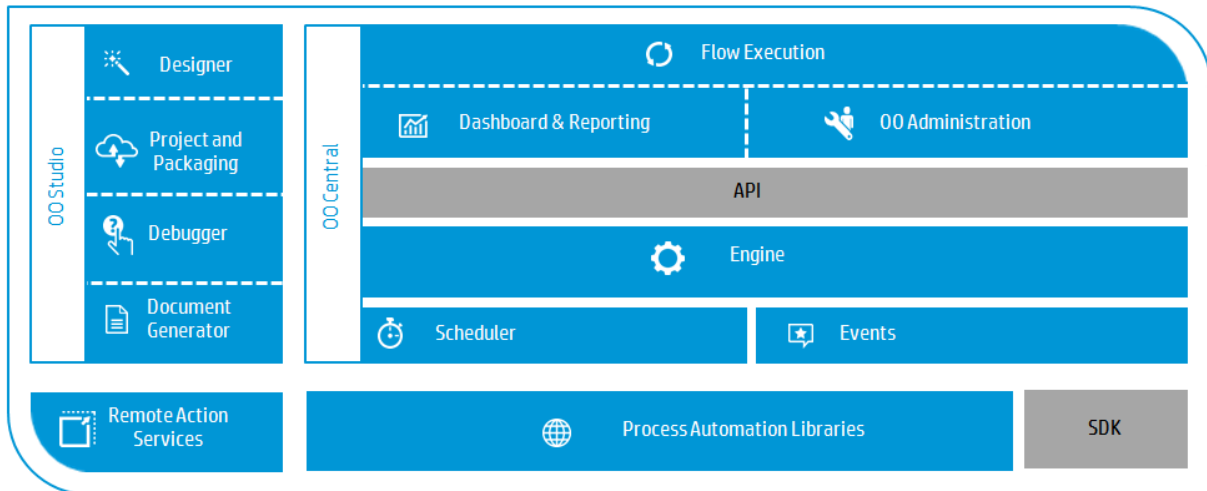
# Benefits for End Users

## Simplified Execution Gateway

HPE OO Central has a refined permission model, which lets you set end user permissions accurately, to allow end users to use Central and be exposed just to the information they need, while limiting their ability to harm the system.

In addition, HPE OO Central provides interactive execution capabilities and UI-embedding capabilities, which enable end users to interact with the system in a much more usable fashion.

# Functional Architecture



HPE Operations Orchestration 10.x is composed of four main functional components:

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

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



## HPE OO Studio

HPE OO Studio is a desktop-based application that is used by flow authors to create the HPE 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**

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

### **Debugger**

HPE 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**

HPE 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, HPE 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 HPE OO Central.

### **Document Generator**

HPE 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.

## HPE OO Central

HPE OO Central is the run time environment of HPE 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.

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

### **Flow Execution**

HPE 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**

HPE 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.

### **HPE OO Administration**

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

### API

HPE 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 HPE 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

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

## HPE OO Remote Action Service (RAS)

The HPE 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 *HPE OO Architecture Guide*.

## HPE OO Content

HPE OO provides a rich set of out-of-the-box operations and flows that enable you to author complex flows, orchestrating various services. The HPE 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, HPE OO provides wizards for generating additional content over other services such as Web Service Wizard.

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

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

# OO Personas

HPE OO personas represent the people responsible for performing particular tasks in the HPE OO workflow. Icons are used in the HPE 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 HPE OO processes.

### Flow Author



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

### Operational Administrator (Op Admin)



The Op Admin is responsible for the day to day operation of HPE OO. This includes configuration, maintenance, content pack promotion, and setting permissions for HPE OO users.

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).

### System Administrator (Sys Admin)



The System Administrator is responsible for the HPE OO hardware and software. The Sys Admin installs and patches HPE 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.

### End User



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

## Minor Personas

The following minor personas play a supportive role in the HPE OO processes. It is possible to use HPE 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 HPE OO flow authoring.

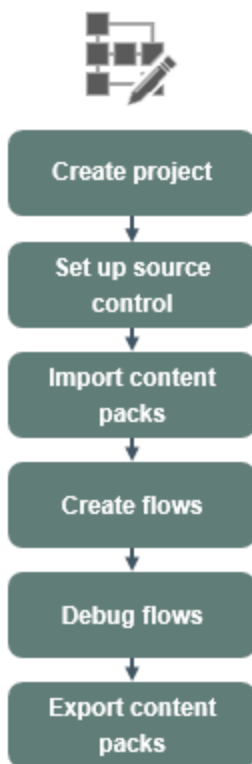
# HPE OO Processes

The major processes in HPE 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.

### 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.


### Step 4: Create a flow

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

### Step 5: Run and debug the flow

 Validate your flow in the Debugger.

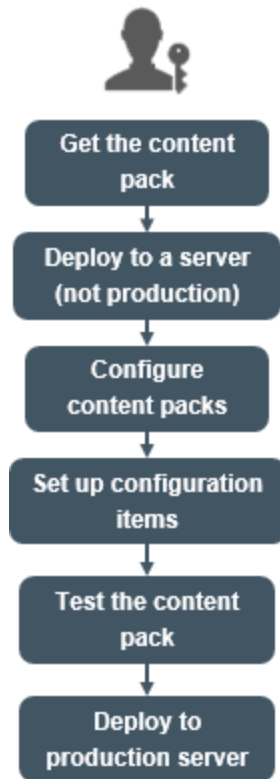
### Step 6: Release the content, packaged into a content pack

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




For more information, see the *Studio Authoring 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 Workspace > 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

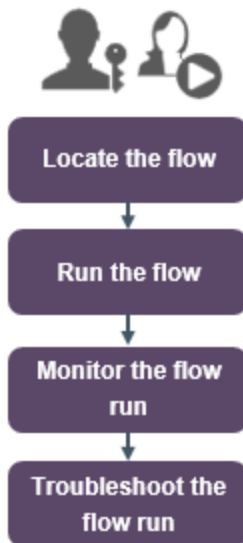
1.  If required, the Op Admin adjusts the configuration of the content pack, for the Production server. For example, it may be necessary to map the worker group aliases and system accounts differently for this server.
2.  Finally, the Op Admin deploys the content pack to the Production server.

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 Workspace > 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 *Central User Guide*.

# OO Entities

## HPE OO Studio

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

## HPE OO Central

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

## HPE OO Content

HPE 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 HPE OO Studio.

## 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.

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

## 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 HPE OO Central application directly, bypassing the HPE OO Central logon screen, if they have logged onto another HPE 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 HPE OO concepts, see the *HPE OO Concept Guide*.

## Studio Project

A project is a directory in HPE OO 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 HPE 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.

## Action

An action is a software component invoked by the HPE 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 HPE, an action can be developed at your own site or provided by a third party vendor aiming to extend the capabilities of HPE 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 HPE OO 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.

## 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)

## Transition

A transition is the connection from an operation's [response](#) to one of the possible next steps.

## 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

## Output

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

## 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.
- **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

## 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.

## 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**.

## 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.

## Configuration Item

Configuration items are HPE 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, HPE 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 *HPE OO Central User Guide*.

### **Note for users who have upgraded from HPE OO 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 HPE OO 10.x, there is no longer a need for a load balancer in front of the RASes. Each RAS communicates with Central.

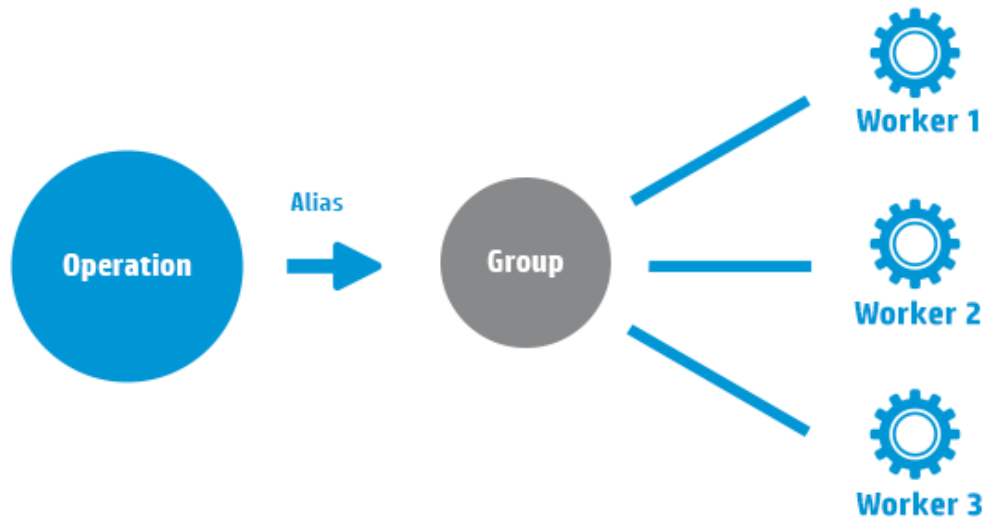
For more information, see the *HPE OO Architecture Guide*.

## Group Aliases

In HPE Operations Orchestration 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 HPE 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 HPE OO 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.

