HP Operations Orchestration

For Windows and Linux Operating Systems

Software Version: 10.00

Release Notes

Legal Notices

Warranty

The only warranties for HP 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. HP 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 HP 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 2013 Hewlett-Packard Development Company, L.P.

Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

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

AMD and the AMD Arrow symbol are trademarks of Advanced Micro Devices, Inc.

Google™ and Google Maps™ are trademarks of Google Inc.

Intel®, Itanium®, Pentium®, and Intel® Xeon® are trademarks of Intel Corporation in the U.S. and other countries.

Java is a registered trademark of Oracle and/or its affiliates.

Microsoft®, Windows®, Windows NT®, Windows® XP, and Windows Vista® are U.S. registered trademarks of Microsoft Corporation.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates.

UNIX® is a registered trademark of The Open Group.

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.

- Document Release Date, which changes each time the document is updated.

- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition of a document, go to:

http://h20230.www2.hp.com/selfsolve/manuals

This site requires that you register for an HP Passport and sign in. To register for an HP Passport ID, go to:

http://h20229.www2.hp.com/passport-registration.html

Or click the **New users - please register** link 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 HP sales representative for details.

Support

Visit the HP Software Support Online web site at:

http://www.hp.com/go/hpsoftwaresupport

This web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support web site to:

- Search for knowledge documents of interest

- Submit and track support cases and enhancement requests

- Download software patches

- Manage support contracts

- Look up HP support contacts

- Review information about available services

- Enter into discussions with other software customers

- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To register for an HP Passport ID, go to:

http://h20229.www2.hp.com/passport-registration.html

To find more information about access levels, go to:

http://h20230.www2.hp.com/new_access_levels.jsp

Contents

Introduction

This document provides an overview of the HP Operations Orchestration 10.00 release. It contains important information not included in the manuals or in the online help.

What's New in HP Operations Orchestration 10.00?

HP Operations Orchestration 10.00 brings the following new features, improvements, and changes to the HP Operations Orchestration platform and content.

New Architecture

The architecture of the product has been upgraded and modernized. The basis of the new architecture is the "task and worker" design pattern. See "Architecture" in the *HP OO 10.00 Concepts Guide* for more details.

Firewall Friendliness

All distributed components now initiate the communication channel. See "Architecture" in the *HP OO 10.00 Concepts Guide* for more details.

Central Look and Feel

The Central Web-based application has a new, more modern design, consistent with the HP Experience style.

The Central UI is organized to reflect the most common OO use cases. Functionalities for the same roles are grouped into the same UI areas:

- **Run Workspace** – used for running flows, monitoring runs, scheduling runs, and troubleshooting runs.

- **Content Workspace** – used for promotion tasks, such as deploying new content, setting permissions on flows, setting up configuration items, and rolling back to the previously deployed version of a content pack.

- **System Workspace** – where the system administrator can configure topology, and set up users, roles, LDAP authentication,and LWSSO.

For more information, see the *HP OO 10.00 Central User Guide*.

Content Packs

Content (flows, operations, configuration items, localization data, and action binaries) is now packaged into "content packs" and deployed to the Central server at run time with zero downtime, for immediate availability.

See "Content Packs" in the *HP OO 10.00 Concepts Guide* for more details.

Updated Support Matrix

PostgreSQL is now also supported, as well as newer versions of MySQL, Oracle, and MS SQLServer. In addition, more operating systems are now supported. See the *HP OO System Requirements*.

New Features For End Users

Simplified Execution Gateway

In previous versions, end users were sometimes exposed to Central. However, since Central is oriented to administrators, they were exposed to much more information than they really needed and in some cases, this could harm the system.

HP OO 10.00 has a refined permission model, which enables setting the end user permissions accurately. The administrator can set up permissions to allow end users to use Central and only be exposed to the information they need, limiting their ability to harm the system.

In addition, HP OO Central 10.00 provides interactive execution capabilities and UI embedding capabilities, which allow the end user to interact with the system in a much more usable fashion.

New Features for Administrators

Roles and Permissions

The way that permissions are managed has changed in HP OO 10.00.

In HP OO 10.00, user groups for Central can be configured in Central only, and not from within Studio, as in previous versions.

We have separated between permissions for Studio, permissions for Central, and permissions for content.

- **Studio** – Permissions to perform actions in Studio have been removed from the application and are now managed in the source control management system.

- **Central** – Permissions to perform actions and to view UI elements in Central are configured via roles.

  - A **Permission** is a predefined ability to perform a task. Central comes ready with a set of permissions that can be assigned to roles.

  - A **Role** is a collection of permissions. The administrator configures the roles (by assigning permissions to them) and then assigns the roles to users.

    The administrator can configure both **View** and **Manage** permissions for the different roles. For example, setting up permissions for a **Promoter** role to both view and manage schedules, and for an **End User** role who can run flows but does not even see the Scheduler module.

    The increased number of permissions enables the administrator to define exactly which parts of the Central web UI each role sees; thus, creating a dedicated UI experience for each group of users. For more information, see "Setting Up Security – Roles" in the *HP OO 10.00 Central User Guide*.

- **Content** – Permissions to access data (flows, operations, and folders) are also configured according to the roles, but this is done for individual flows or content folders, and not system-

wide. This is done as part of the promotion of a content pack.The administrator can assign permission to a role to have **View** permission or **Run** permission for a flow, operation, or folder. For example, you might want to give users with the role **Promoter** permission to view and run and run the contents of a selected folder, and give users with the role **End User** permission to view the content only.

For more information, see "Managing the Flow Library" in the *HP OO 10.00 Central User Guide*.

As a result of these changes, there are some features from previous versions that are no longer relevant:

- Setting permissions on system accounts is no longer supported.

- Setting permissions on operations is no longer supported.

- Execution permissions are configured in Central, and no longer in Studio.

- Separate permission for HEADLESS execution is not supported.

- Authoring permissions (read\write\linked to) are no longer managed and configured in Studio.

- The user group terminology has changed: users groups have been changed to roles.

Promotion to the Central Server

Content packs are promoted to the Central server. The aim of promotion is to deploy a new content pack to the Central Production server, to make the flows available to users.

In addition to deployment, the promotion process also includes:

- Setting up the configuration items in the content: aliases, system accounts, and so on

- Setting permissions for the flows in the content pack

- Testing and troubleshooting the flows in the content pack

See "Promoting Content Packs" in the *HP OO Central User Guide* for more details.

Note: System accounts are not deployed as part of content pack deployment. In order to add, update, or delete system accounts, you need to use the REST APIs. For more information, see the *HP OO Application Program Interface (API) Guide*.

Aliases: Separation Between Run time and Authoring

In HP OO 10.00, we understand that the author working in Studio does not always know which environment the content will be deployed upon. Therefore, we have created the concept of an "alias", which is created in Studio, and then mapped to the relevant value after the content has been deployed to Central. If the content is deployed to different Central nodes, the mapping can be done differently on each node, without the need to modify the original alias in Studio.

An author can create an operation in Studio, and give this operation a "group alias". After the flow is deployed in Central, the administrator maps the group alias to a worker group in Central. For more information, see Configuration of Worker Groups and Group Aliases.

Workers

A worker is responsible for executing flows. The worker pulls tasks (executions) from Central and performs steps within these executions. This includes both invoking the actions and navigating between the steps of the flow.

Unlike a RAS in previous versions of HP OO, a worker in HP OO 10.00 processes all types of steps, and not only remote actions. Also, the communication direction of a worker in HP OO 10.00 is reversed from that of a RAS in previous versions. In previous versions, the Central server connected to the RAS server. In this version, the worker initiates the connection to the Central server. For more information, see "Components" in the *HP OO 10.00 Concepts Guide*.

Configuration of Worker Groups and Group Aliases

A worker group is a logical collection of workers. Having a group, rather than a single worker, enables workers to withstand a high action execution load, and increases availability of workers in a data center.

Group aliases let you separate between assigning an operation to a worker during authoring time and in the run time environment. At authoring time, the flow author defines an operation to run on a group alias rather than a group. When the content pack is imported to Central, the administrator maps the alias to a group in the run time environment. There is no need to edit the flows and modify the worker assignment manually. For more information, see "Groups and Group Aliases" in the *HP OO 10.00 Concepts Guide*.

Simpler RAS Management

HP OO 10.00 uses a group mechanism that separates between the logical notion of the target RAS (which is what the author is aware of) and between the physical notion of the target RAS (which is what the administrator is aware of). For every environment, the administrator can map between the logical and physical notions. Therefore, promotion of content between environments does not require adjustment, and the addition of a RAS does not require going back to the author.

In previous versions, the content had to be aware of the target RASes. This meant that when content was promoted between environments (for example Dev, Staging, and Production), it required adjustments. Therefore, the content was not identical between environments and tests results were not definite. An administrator who wanted to promote the content to the Production environment needed to have authoring knowledge or had to go back to the author to make this change. When a RAS needed to be scaled out to allow high availability or improved performance, it had to be added with a load balancer, and the content had to be adjusted again.

Live Scale Out

In HP OO 10.00, you can add a component without restarting the other components. You simply add it to the system in a live manner. Since the Central cluster is no longer based on Terracotta, you just add a new instance and point it to the database. You can scale out RASes via a new grouping mechanism, which means that there is no need for a load balancer for the RASes.

The run time repository was combined into the database, which means there is no need to manage a shared file system for clustering and improving the cluster's stability.

In previous versions, adding a component to the system required a restart of the system components. Terracotta needed to be updated and restarted as well as the Central nodes. If RASes were scaled out, it was required to use a load balancer and adjust the flow.

High Performance

HP OO 10.00 has an improved execution mechanism, which enables you to reach higher performance. HP OO 10.00 is more than three times faster than previous versions.

Highly Parallelized Runs

HP OO 10.00 has an asynchronous execution mechanism, which enables large bursts of executions. HP OO 10.00 supports launching 100 flows/second and there is no limit to the number of parallel executions.

In previous versions, the execution mechanism was based on synchronous execution, so the number of parallel executions was limited by the number of the system threads (with a default of 300).

Automatic Content Distribution

In HP OO 10.00, content binaries are automatically distributed to the various RASes. When there is a change in their version, the system automatically distributes them to the relevant RAS upon their first use.

In previous versions, content binaries needed to be distributed manually to the various RASes. You needed to make sure that before executing the content, the binaries of the correct version are in the relevant RASes.

Isolated Content Dependencies

HP OO 10.00 enables you to use a different third party version for every content pack. You can use your own third party versions and this is not affected by HP changing the out-of-the-box version. The only limitation is regarding dependencies that are shared between the plugin and the platform.

Previous versions used a single version of a third party library for all repository operations. This meant that you could not use your own version of the third party library, due to collisions with the out-of-the-box version. In addition, every new HP content pack that was deployed posed a risk, because it could change the out-of-the-box version and break the dependencies of your operations.

Authentication via LDAP Integration

If you have an LDAP service for authentication, you can add the LDAP configuration to Central, in the Central UI. This enables the Central security feature and users will need to be authenticated to use the system. This change is dynamic—there is no need to restart Central for the change to be applied.

When the next user logs in to Central, they will need to enter their user name and password, as required by the LDAP authentication.

The way that LDAP is configured in HP OO 10.00 is different from previous versions:

- In HP OO 10.00, you need to specify the port for the LDAP host.

  In previous versions, it was possible not to specify the port for the LDAP host, and Central would use the default port.

- In HP OO 10.00, you can configure multiple LDAP hosts by entering multiple values in the **Host** and **Ports** fields, with semicolons as delimiters. This saves time, because you don't need to enter the same information into multiple windows.

  In previous versions, the way to configure multiple LDAP hosts was by opening multiple LDAP windows with different IP addresses.

- In HP OO 10.00, if the LDAP connection is lost, there is no need to restart Central. If there are multiple LDAP hosts configured, Central will connect again. If the first LDAP host is down, after three seconds, Central will try to connect to the second one that was configured.

  In previous versions, Central was bound to the LDAP connection, so if the LDAP connection was lost, it was necessary to restart Central.

Internal Users

As an alternative to LDAP authentication, it is possible to configure internal users in Central. Internal users can log in using their user name and password, and do not require external authentication. When an internal user and user with LDAP authentication with the same role are logged in, there is no difference between them.

> **Note:** Using internal users is considered low security. For example, there is no password policy in place. If you require high security, you should use LDAP authentication.

Dynamic Login

If no authentication is configured, users will be able to enter Central without authentication. If authentication is configured, users will need to log in with authenticated user names and passwords—there is no need to restart Central for the change to be applied.

Modernized Extension Development

Java extension development is now Java annotation-driven and provides classpath isolation. See the *HP OO Extension Developers Guide* for more details.

Changes in on-Fail Error Handling

If the **on-Fail** option is *not* selected in a step, when a run time exception occurs in an action or scriptlet, or relating to input or output assignment, the flow will fail to complete

If the **on-Fail** option *is* selected in a step, when a run time exception occurs in an action or scriptlet, or relating to input or output assignment, the flow finishes, and does not terminate in the middle of the flow.

In previous versions, when a run time exception occurred in an operation, the flow finished at the point where the run time exception occurred.

In HP OO 10.00, the **on-Fail** option for a response relates to all exceptions, for example, exceptions relating to input or output assignment, or exceptions that occur in the step scriptlet. In previous versions, it only related to exceptions in an operation.

Automatic Database Purging - Low Cost Partitions

HP OO 10.00 contains a new mechanism for handling a large stream of events and storing them to a standard relational database without using any advanced or enterprise database-specific features. Using this mechanism, a low-cost database on a low-cost machine can withstand tens of thousands of transactions, resulting in lower cost and higher efficiency.

Scheduling Personas

In HP OO 10.00, a single user creates the schedule, and is designated as the schedule owner. When the scheduled flow is run, this user is also the owner of this flow execution (who performs the scheduled run). This means that the same user is both the schedule owner and the executor.

In previous versions, there were two different scheduling personas:

- The **Schedule creator** was the user that created the schedule.

- The **Schedule executor** was configured from within the Administration settings (under **Administration** > **System Configuration** > **Scheduler** > **Scheduler Settings**) and this was the user that performed the scheduled run.

New Features for Authors

Standalone Studio

HP OO Studio functions as a standalone integrated development environment (IDE).

In previous versions, Studio required a connection to a live HP OO Central. The Studio login and many repository operations used this connection, which made it difficult to work in remote teams. In some cases, a cross-site connection was not allowed and in other cases, such operations were very slow.

In HP OO 10.00, Studio is a standalone tool that doesn't require a connection to Central. All of its repository operations are available offline. This results in improved performance and enables distributed work. If a source control interaction is required, you decide when the interaction occurs. Remote teams can use various standalone Studios, and it is possible to author flows outside of the office network.

Standard Source Control Integration

In previous versions, Studio included its own proprietary version control capabilities, which were limited by definition.It was not possible to connect to and use your organization's source control software.

HP OO Studio 10.00 is integrated with a standard source control software. The out-of-the-box solution is based on a commonly used source control application (SVN). This means that the capabilities of source control software are available for Studio, so you can connect to and use your organization's source control software. This also means that the automation code can reside together with other source codes (automation as code).

In HP OO Studio 10.00, the proprietary version control capabilities (check-in and check-out operations) have been removed. These have been replaced by the source control integration. When an author makes local changes, the local history is displayed as it was in previous versions.

For more information, see "Working With Source Control" in the *HP OO Studio Authoring Guide*.

Multiple Authors and Multiple Geographies

In HP OO 10.00, Studio is offline and leverages standard source control software to enable sharing work between multiple and distributed authors.

In previous versions, the capabilities of Studio were limited in regard to multi-authoring and multi-geographies. For example, Studio required a constant connection to Central, and collaboration capabilities were limited by the proprietary version control.

Fine-grained HP Content

The HP OO 10.00, content is broken down into a set of content packs, each for a different area of functionality. You have better control over which content packs to download and which to deploy. You can use only what you really need.

In previous versions, the HP content was provided as one big repository. This meant that you did not have control over what was downloaded and deployed. Content versions included much more content than you needed, and imposed risks on the deployment. Content versions could unintentionally affect your entire content.

Fine-grained Customer Content

In HP OO 10.00, the customer content can be separated into projects and managed separately for each author or author group. This gives you complete flexibility in defining the flows that are grouped together and defining the workspace of each author. In this manner, different authors get a focused development environment just with their relevant flows and don't affect other authors' flows.

In previous versions, the customer content was kept in one big repository. This meant that different groups needed to work simultaneously on the same large repository. This meant having to find ways to avoid collisions and manage permissions. Individual authors didn't get a focused development environment with just their own flows.

Projects Pane and Dependencies Pane

There are two new panes in Studio 10.00:

- **Projects** pane, which shows the project you're working in, and displays the editable flows, operations, and other objects.

- **Dependencies** pane, which shows the available content packs, and displays the read-only flows, operations, and other objects.
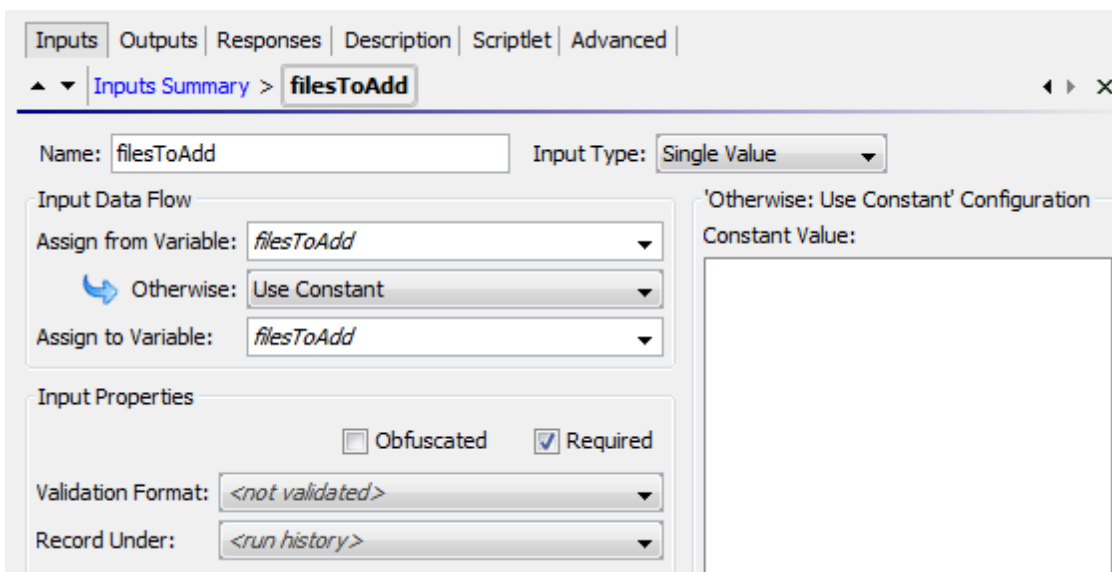
These two new panes replace the **Repository** pane and the **My Changes/Checkouts** pane from previous versions.
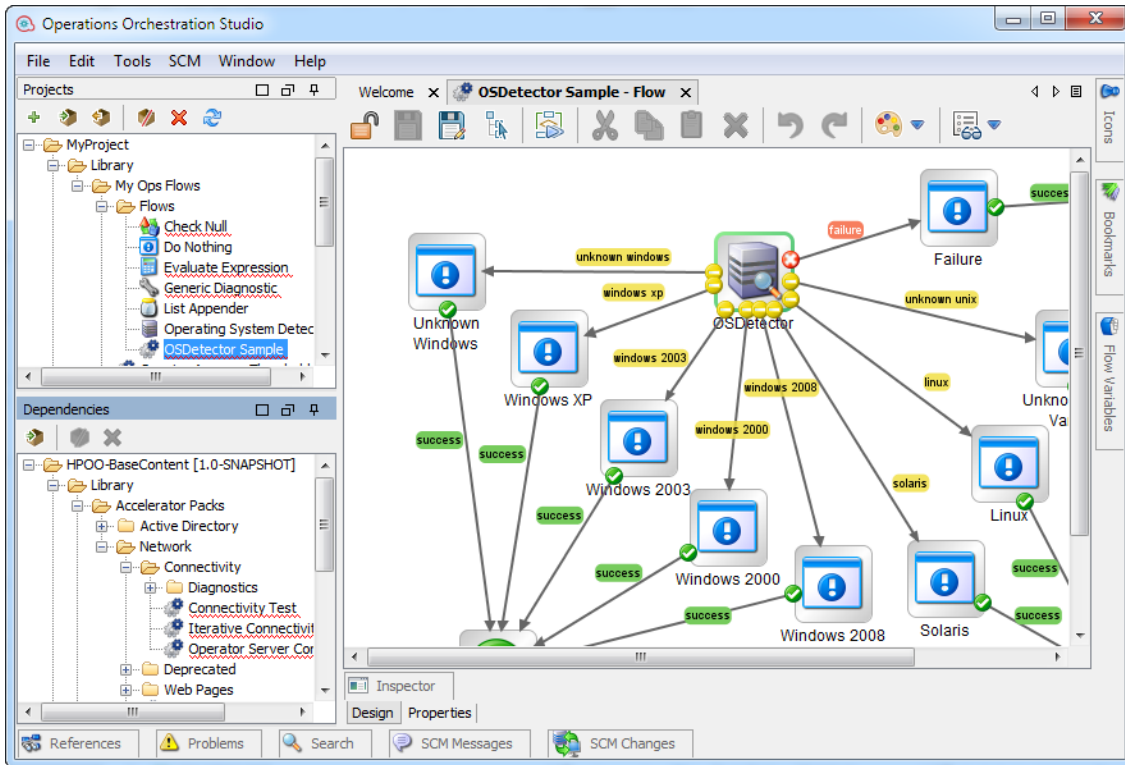
See the *HP OO Studio Authoring Guide* for more details.

Renamed check box in the Input Editor

In the Input Editor, the **Encrypted** check box has been renamed to **Obfuscated**. There is no change in the functionality, but the label is a more accurate description.

Credential Type

In HP OO 10.00, inputs of the type **Credentials**, which use a system account, must be defined as referring to either a user name or a password. There is a new field called **Credential Type**, with the options **Username** and **Password**.

New Way of Creating Operations: Soft Copy

In HP OO 10.00, there is a new method for creating new operations.

In previous versions, you could create operations from built-in operations in Studio. This is no longer possible, and there is no **New Operation** menu item. There are no longer operation types—WebOp, SSH, Telnet, and so on.

In HP OO 10.00, you can create operations by copying and modifying existing operations, or by creating an action plugin in Java and importing that action plugin.

When you create an operation by importing a Java action plugin, the copied operation continues to reference the original operation. So if the original operation is updated to link to a new version of the action plugin file, the copied operations are all updated automatically. This is known as a "soft copy".

> **Note:** If you want to have two operations that are based on two separate versions of the action plugin file, you need to create two operations. This cannot be done by copying an existing operation.

For more information about creating an operation, see "Creating Operations" in the *HP OO Studio Authoring Guide*.

No Separation Between Headless Runs and Those Started Within the Central UI

In HP OO 10.00, "headless" runs (which are started outside of Central) and flows that are run from within the Central UI are treated in the same way.

In previous versions, there was a separation between these types of run.

Multi-instance Steps

Multi-instance steps now reside within a multi-instance branch. You simply select the **Multi-instance** icon on the **Step** palette and drag it onto the authoring canvas. You then add one or more subflows or operations to the multi-instance branch, and set multiple targets for the step via an input list of values.

Each of the inner steps inside the multi-instance step will run multiple times, once for each target that is set.

The new approach is more consistent with the parallel execution visualization and capabilities. See the *HP OO Studio Authoring Guide*, and the *HP OO Content Upgrade Guide* for more details.

Changes in behavior with multi-instance steps:

- In HP OO 10.00, the multi-instance step inputs behave the same as a regular step. If you remove an input (that was declared in the operation) from the multi-instance step, then it won't

be calculated.

In previous versions, you could not remove operation inputs from the inputs of a multi-instance step (unlike a regular step).

- The whole structure of the multi-instance step has changed. There is now a multi-instance branch. So there are also changes in the multi-instance step properties.

- The multi-instance branch step scriptlet and results run N times (as a number of multi-instance instances).

- The results and scriptlet for all the inner steps (inside the multi-instance branch) will run once for each multi-instance branch run, meaning N times at total.

- Using a multi-instance step inside a parallel lane does not work. This cannot be created in HP OO 10.00 Studio, and will not pass migration. It is supported as a subflow step in parallel lane.

- A multi-instance step that has a transition to a parallel step (or another multi-instance step) that is not a "group done" transition, will also not pass migration, and will not work in HP OO 10.00.

- Multi-instance steps no longer support termination of the flow within the multi-instance step.

Two XML Files

The flow XML has been split into two XML files, in order to improve the readability of the files:

- Advanced Flow Language (AFL), which defines the structure of the flow.

- Graphical Hierarchical Language (GHL), which describes the position of the graphical flow components in the Studio layout.

Both languages provide an XML schema (XSD). For an example of the XSD, see the *HP OO Application Program Interface (API) Guide*.

Numeric Values Keep Their Original Form in Scriptlet Context

In HP OO 10.00, the string retains its original form. In previous versions, when a numeric value was put in a scriptlet context and retrieved back, it was changed to a string representation of float, which could lead to unpredictable behavior.

For example, in the following script:

```
var n = 1;

scriptletContext.put("numericValue", n);
```

- In HP OO 10.00, the context holds the original string "1"

- In previous versions, the context held the string "1.0"

> **Note:** Putting a float variable on the context is the same in HP OO 10.00 and in previous versions. In both cases, the float will keep its original form (for example,"`1.1`").

Scriptlet Context is Read Only

In HP OO 10.00, scriptlet filters have read-only access to the `scriptletContext`. Using `scriptletContext.put/putGlobal` has no effect outside of the script run. This change helps to enforce good practices, so that filters are only used for filtering values.

In previous versions, it was possible to change/add data to the `scriptletContext` in scriptlet filters.

ScriptletContext.get() Looks in Local Context First

HP OO 10.00 looks for the value in the local context first and will only look in the global context if the value is not available. This enables the flow to override variable values at run time and is more intuitive.

In previous versions, when a script called `scriptletContext.get(<paramName>)`, HP OO would first look for the `<paramName>` value in the global context, and would only look in the local context (in the flow or step) only if no value was found.

Automatic Initialization of Variables

In HP OO 10.00, non-existing variable references are initialized with an empty string `""`, instead of `"${var}"`.

In previous versions, if `${var}` syntax was used in a value field, and `${var}` was not initialized in the flow, HP OO used the literal string `"${var}"` as the value for variable substitution. This literal substitution in the parameter could result in confusion or errors in the flow.

> **Note:** In HP OO 10.00, the `$` symbol is used as an escape character. So that if you previously had a literal string `"${var}"`, you can add another `$` symbol to maintain the original string. If you use "`$${var}`", this will be "`${var}`" in execution.

Updated Script Engine

HP OO 10.00 uses the Rhino 1.7R3 script engine, while earlier HP OO versions used Rhino 1.7R1.

This may affect scriptlets, because Rhino 1.7R3 requires the symbol ";" .

For example, the following scriptlet would work in previous versions, but will not work in HP OO 10.00:

```
if ( (scriptletContext.get("glob1") == "1,2,3") || (scriptletContext.get("gl
ob1") == "1") || (scriptletContext.get("glob1") == "2") || (scriptletContext
.get("glob1") == "3") )
scriptletResponse = "success"
else scriptletResponse = "failure"
```

In HP OO 10.00, you would need to change the scriptlet to be as follows:

```
if ( (scriptletContext.get("glob1") == "1,2,3") || (scriptletContext.get("gl
ob1") == "1") || (scriptletContext.get("glob1") == "2") || (scriptletContext
.get("glob1") == "3") ){

scriptletResponse = "success";

}

else scriptletResponse = "failure";
```

In addition, error messages thrown by scriptlets may differ slightly from those in previous versions, because of the newer Rhino script engine.

Note that Sleep scripts are no longer supported.

Annotation-based Content

HP OO 10.00 adds '@Action' annotations that can be added directly onto your custom code. This means that your code can now be leveraged to be HP OO content and still be tested in the context of the customer development framework.

Previous versions provided an 'IAction' mechanism to create custom content. You had to implement that interface and test it within the HP OO application context.

Changes in Behavior in Studio

Because of the differences in behavior between HP OO 10.00, some flows may not work properly after content upgrade. For more information, and for suggested solutions, see the *HP OO 10.00 Content Upgrade Guide*.

- The Debugger displays only selection lists that are relevant for the flow, rather than all of them.

- The **on-Fail** rule in a response now takes into account exceptions that occur in the step scriptlet,and exceptions relating to input or output assignment.

- There is no longer a **Step Results (Deprecated)** section under the **Advanced** tab in the Step Inspector. After content upgrade, results in this section are moved to under the **Results** tab. See the *Content Upgrade Guide* for more details.

- In previous versions, it was possible to take the name of a selection list or system property and use it in an input title, in the format ${<name>}. For example, ${Boolean}. This is no longer possible for selection lists. After content upgrade, each selection list is copied as a system property, in order to maintain backwards compatibility. See the *Content Upgrade Guide* for more details.

- In previous versions, some types of operations included built-in field values, but these are not supported in HP OO 10.00. For example, in the loop operation in previous versions, there were two inputs, **count** and **reset**, and two fields, **from** and **increment by**, to specify the starting count and increment. After content upgrade to HP OO 10.00, the fields are converted into regular inputs. See the *Content Upgrade Guide* for more details.

- In 10.00, there is a new **Categories** folder in the **Configuration** folder. In previous versions, there was a domain term called **Categories**, which contained different classifications of a flow. After content upgrade, the items that previously appeared as rows in the **Categories** domain term now appear as separate items in the **Categories** folder. See the *Content Upgrade Guide* for more details.

- Non-existing variable references are initialized with an empty string "", instead of "${var}". See the *HP OO 10.00 Concepts Guide* for more details.

- In selection lists and domain terms, values cannot exceed 255 characters.

Changes in Behavior – Flow and Step Inputs

- The assignment chain has been changed. For example, in the following scenario of flow inputs:

Name	Type	Required	Encrypted	Assign from	Otherwise	Assign to	Value in execution
Input 1	Single Value	false	false	Input 1	Prompt user	Input 2	myhost
Input 2	Single Value	false	false	Input 2	Prompt user	targetHost	localhost

In previous versions, in the flow context, the result would be the variable "targetHost" with the value "localhost".

In HP OO 10.00, the result is the value "myhost".

- Inputs available when triggering a flow has changed. When you ask for the inputs of a flow (using the Rest API), it will show you only the inputs that have the same input name as the input "Assign From" parameter (as declared in Studio).

- Step inputs - assigning outputs with a filter.

  In HP OO 10.00, all the output values are processed with their filters, so even if an output is not mapped to a flow result, the flow will fail if the filter throws a run time exception.

  In previous versions, if a step output was not mapped to the flow result, its filter would not be processed. For example, if you have an output field filter that can cause a run time exception for some step result values, it won't fail the flow as long as that step output field is not mapped to a flow result.

- The transition of encrypted fields has been changed according to the following logic:

  - If an encrypted field is passed along to a non-encrypted field, it will be shown decrypted.

  - If an encrypted field is passed to the flow result, it will be shown decrypted.

- If a steps' encrypted field is assigned to another steps' encrypted field, it will remain encrypted.

- In general, when a field is in a step/operation/property that is marked as encrypted, it will not be shown. Otherwise it will be shown.

● In HP OO 10.00, the author needs to choose which attribute to extract from the logged in user. In addition, the password is never used for security issues, and will always be prompted .

In previous versions, the logged in user was an object consisting of the user name and password, and assignment took place according to the name of the input. For example, if the input name was "user name", the user name would be extracted from the object.

● In HP OO 10.00, the credentials input type does not exist and instead, the user will have to use the regular prompt input (and check it as encrypted).

In previous versions, the credentials input prompt was an object consisting of the user name and password and a message was prompted during run time to insert both.

● It is no longer possible to set the flow input as "not assigned" otherwise "fail". After content upgrade, these inputs will be modified to single value, otherwise "prompt".

Changes in Behavior – Exception Handling

● If an exception is thrown anywhere during flow execution, from an action\scriptlet\internal execution step, then the following logic applies:

If the author checked the **On Fail** flag on one of the operation responses, the flow will continue to run regularly (on all the execution steps) and chooses the transition mapped to this response. This will also be the case if the exception was thrown in the operation scriptlet or the step scriptlet.

If the author didn't check the **On Fail** flag on any response of the operation, the flow will stop its execution at the execution step that threw the exception in any case except for the following:

- If the author checked the **Toggle single response** option on the step, the flow will continue to run to the next step.

- If the exception was in a parallel \ multi-instance step, the branch with the error will stop execution. The other branches will finish (reach the end of the branch), and then the flow will stop execution.

- If the exception was in a sub-flow, the child and parent flows will stop the execution.

- If the **Default** check box is selected on one of the responses, the behavior is now different. In previous versions, the flow would continue to run to the transition mapped to the default response. In OO 10.0, we stop the execution of the flow in the execution step that the exception was thrown from, and end the flow.

In all the cases above for which the flow stops running, it will reach termination in an organized manner with a `FLOW_RESPONSE_TYPE=EXCEPTION`.

Changes in Behavior – Rules

- The following behavior of rules was redesigned in HP OO 10.0 to behave as follows:

Input 1	Rule Type	Input	Result in Previous Versions	10.0 Result
5abc	=	5iop	true	false
Two	>	Two	true	false

In general, if one of the inputs is not numeric, HP OO does lexicographic comparison.

Changes in Behavior – Scriptlets

- HP OO 10.00 uses the Rhino 1.7R3 script engine, while earlier HP OO versions used Rhino 1.7R1. This may affect scriptlets, because Rhino 1.7R3 requires the symbol ";" .

  For example, the following scriptlet worked in previous versions, but will not work in OO 10.0, because of the missing ";":

  ```
  if  (scriptletContext.get("glob1") == "1,2,3")

  scriptletResponse = "success"

  else scriptletResponse = "failure"
  ```

  You can fix the scriptlet like this:

  ```
  if (scriptletContext.get("glob1") == "1,2,3")Unknown macro: {scriptletResponse
  = "success";}

  else scriptletResponse = "failure";
  ```

- Error messages thrown by scriptlets may differ slightly from those in previous versions, because of the newer Rhino script engine.

- Scriptlet filters have read-only access to the `scriptletContext`. Using `scriptletContext.put/putGlobal` has no effect outside of the script run.

- `ScriptletContext.get()` looks in local context before looking in global context.

- In HP OO 10.0, when a script calls `scriptletContext.get(<paramName>)`, the system first looks for the value in the local context and only if the value is not available, it tries to fetch it from the global context. This is done in order to allow the flow to override variable values at run time. Also, it is more intuitive.

In previous versions, when a script called `scriptletContext.get(<paramName>)`, the system would first look for the `paramName` in the global context, and only if no value was found, it would look for the `paramName` in the local context (flow/step).

- In HP OO 10.0, when a numeric value is put in the scriptlet context and retrieved back, the string keeps its original form. In previous versions, it was changed to a string representation of float. See "Numeric Values Keep Their Original Form in Scriptlet Context " on page 17.

Changes in Behavior – Filters

In the **XML Get Attribute** filter, you can set the attribute name whose value you want to extract. You can also leave it empty, and then OO returns all the attributes in the given path.

In HP OO 10.0, the attributes are returned in the same order that they appear in the XML. In previous versions, they were returned in the opposite order.

For example:

Working on the following XML, if a user didn't set the attribute name, but only the path: "tickets/ticket/details/comment/".

```
<?xml version="1.0" encoding="utf-8"?>

  <tickets>

  <ticket id="1448" severity="3">

    <details>

    <description>A simple Test xml</description>

      <comment user="john" att2="att2val">Initially raising ticket</comment>

      <comment user="frank">Problem diagnosed, not a real issue</comment>

      <comment user="albert">ok, I'm going to close it.</comment>

      <state>Closed</state>

    </details>

  </ticket>

</tickets>
```

previous versions single match result:

```
att2val
```

HP OO 10.0 single match result:

```
john
```

previous versions table result:

```
Path,att2,user

/tickets/ticket/details/comment[1],att2val,john/tickets/ticket/details/comment[2
],,frank/tickets/ticket/details/comment[3],,albert
```

HP OO10.0 table result:

```
Path,user,att2

/tickets/ticket/details/comment[1],john,att2val/tickets/ticket/details/comment[2
],,frank/tickets/ticket/details/comment[3],,albert
```

Changes in Behavior – Non-blocking Steps

- In HP OO 10.00, the flow will wait for all non-blocking steps to terminate. The flow execution will remain in progress until then.

- In HP OO 10.00, a non-blocking step will run in parallel with the rest of the flow. This includes input binding, scriplets, operation, and filters. The entire step is parallel as opposed to the behavior in previous versions where just the operation was parallel.

- In HP OO 10.00, a subflow with a non-blocking step will not terminate until all its non-blocking steps end.

New Features for Integrators

Complete REST API

HP OO 10.00 provides a complete REST API to the entire set of Central functionalities. Every functionality that is exposed in Central is practically implemented over a public REST API, enabling you to utilize the same set of APIs to integrate your systems with HP OO.

Previous versions provided a sets of APIs (SOAP, REST,and so on) that covered mostly the execution functionalities. Integrators who needed to integrate their systems with HP OO were limited with the API capabilities.

Some of the RESTful APIs and SOAP APIs from previous versions are not supported in HP OO 10.00.

Live Event Stream

HP OO 10.00 exposes the execution events in a live stream with RSS and ATOM feed formats. You can listen to the feed and react to the execution events immediately.

Previous versions provided an XML summary of the executions steps at the end of the execution. This XML had a proprietary format.

Embeddable

HP OO 10.00 is provided not just as a standalone application but also as a WAR file. You can embed it in your application servers and do not need a separate service and management for HP OO.

Previous versions is installed as a standalone application. If you wanted to OEM the application, you needed to 'bundle' it. This meant that it included its own service and required its own management.

HP OO 10.00 Documentation

The documentation for HP OO 10.00 has been rewritten, so that it is more concise and easier to navigate.

Persona icons have been added to help identify task owners, and flow diagrams provide an overview of processes. In the HP OO help, these flow diagrams link directly to the relevant help topic.

Note: In order for the links in the flow diagrams to work properly, make sure that the browser window containing the HP OO help is expanded. It the browser window is contracted, the links in the flow diagrams may not work correctly.

Context-sensitive Documentation

The help in HP OO 10.00 is context-sensitive.

This means that by clicking the **Help** button on any HP OO window or dialog box, you display the relevant help topic. Click the **Open topic with navigation** link in the top left corner to display the navigation panel with links to all the help topics.

Open topic with navigation

Tracking Flow Runs

👥 👤 While a flow is running, you can track the run and monitor its progress in the **Run Explorer** tab. The table displays information for basic monitoring of your runs and those of other users:

- Run Name

Documentation Updates

Check the HP OO 10.00 Resources page on HPLN at https://hpln.hp.com/node/21/otherfiles for updates to the documentation.

Content Documentation

Download the following HP OO 10.00 Content documents from the HPLN **Operations Orchestration Content 10.00 Content** page:

- Release Notes

- Wizard Guides

- Integration HTML

- Localized documentation

Known Issues and Limitations

Temporarily Removed Features

- It is not possible to use the **Reference** option for rules in a response. References to external rules are not supported at run time, and operations that use them will fail on deployment.

- Reports are not currently available.

- Debugging scriptlets in the Javascript Debugger is not available in this version.

- Input evaluators are not available during flow execution, in this version.

- In the **Advanced** tab of the Flow Inspector, the **Make Flow Variables Global When Not a SubFlow** check box is not supported.

- In the **Advanced** tab of the Step Inspector, the following check boxes are not supported:

  - **This step saves the whole run state**

  - **This step saves the run history**

  > **Note:** The check boxes **This step saves the raw output** and **This step saves the primary output** have been removed pernamently.

- Flow inputs that were added in the Central Scheduler window in previous versions are not supported in HP OO 10.00 and are not upgraded during content upgrade.

- The Dashboard is not available in HP OO 10.00.

- Guided runs are not available in HP OO 10.00.

- Folder descriptions do not appear in Studio and Central.

Troubleshooting

- On Windows 7, Vista, and Windows 2008 Server, there may be problems running SVN with Studio, if you are missing the correct version of the MS Visual C++ DLLs. The following error message appears in the **SCM Messages** pane:

```
Cannot run program "E:\678\studio-win64-1.25-SNAPSHOT(1)\studio\SlikSvn\bin\svn": CreateProcess error=14001...
```

Workaround: Download and install the Microsoft Visual C++ 2008 SP1 redistributable package according to your operation system. For example, http://www.microsoft.com/en-us/download/details.aspx?id=5582.

- If flow authors create two system accounts with the same name in different Studio projects, these will appear only once in Central.

  Workaround: If you want these system accounts to be separate, change the system account name in one of the projects.

- In some cases, the content deployment does not end or does not appear to end. This may be caused by the anti-virus blocking the browser read actions.

  Workaround: Check the CPU consumption of the anti-virus process at the time of deployment, or check the anti-virus log file. If required, contact your server system administrator to discuss possible solutions.

- In some cases, Central is not started as part of the installation, or the Central URL cannot be accessed from a remote location. This may be caused by the Windows firewall.

  Workaround: Check the Windows firewall logs and policies. If required, contact your server system administrator to discuss possible solutions.

- Stopping OO mgmt service from Windows services may result in an error.

  Workaround: If this occurs, stop the service from the Task Manager.

- Scriptlets are case-sensitive. For example, flows with scriptlets may result in a navigation error if the response of the operation is **Success** rather than **success**.

  Workaround: In the scriplet code, change the response of the operation from **Success** to **success**, because HP OO 10.00 is case-sensitive when calculating transitions.

- After an installation of HP OO 10 with a MySQL database, the following error may appear in the logs:

  "Packet for query is too large (2515019 > 1048576). You can change this value on the server by setting the max_allowed_packet variable."

  Workaround: Before installing HP OO 10.00 with a MySQL database, make sure that the **max_allowed_packet** variable has a minimum value of 200 M. In order to do this, you can set the value for a session or as a permanent option of the MySQL server:

  - To set it for within the session, run this command:

    ```
    mysql --max_allowed_packet=200M
    ```

  - To add it as a permanent option of the MySql server, add the following line in **<MYSQL_HOME_DIR>\my.ini** and then restart the MySQL server:

```
max_allowed_packet=200M
```

> **Note:** The minimum is 200 M, but some environments may require more, depending on the size of the repository.

- When Central, the RAS, and the Central client (browser) are in different time zones, this can cause problems when a flow is run via the Scheduler. For example, the start time in the flow will be different from the start time in the step.

  Workaround: Change the time zone on the RAS to be the same as the time zone on Central.

- When you run content upgrade including schedules, the schedules get upgraded only if they were not yet scheduled in HP OO 10.00. If the flows were already scheduled in HP OO 10.00, these schedules are not upgraded.

  Workaround: Make sure to complete the upgrade before working with the HP OO 10.00 environment for scheduling and running flows.

- If a content pack containing a flow is deployed and an operation is deleted from the flow and from the content pack, this can cause a second deployment of the content pack to fail.

  Workaround: Do not delete the operation from the content pack.

Known Issues

- In a clustered environment, the clocks on the different machines must be synchronized using some form of time-sync service (daemon) that runs very regularly. The clocks must be within a second of each other. For instructions on how to do this, see http://www.nist.gov/pml/div688/grp40/its.cfm.

- SSO Kerberos authentication is not supported.

- Studio SCM does not support conflicts and merging. You need to lock an item before starting to edit it. If a flow is already locked, you must wait for the lock to be released, before editing the flow.

- Descriptions (for example, of flows or inputs) that are longer than 4,000 bytes are truncated, so that only part of the description is visible.

- The Operations Orchestration Shell (OOSH) utility does not support HTTP Proxies.

- For flows with inputs at flow level that have **Use Constant**, you must re-enter the value of the **Use Constant** inputs at run time. The **Run** button is not available until all inputs are re-entered.

- In Central, it is not possible to reassign a run to a LDAP user.

- LDAP group membership is only supported when the group membership field value is a DN.

- Special characters in an LDAP CN are not supported.

- LDAP group inheritance is not supported.

- The operating system time and time zone of the Central and RAS machines must not be changed after installation.

- When authentication has been enabled on Central and Studio has been configured to work with the Remote Debugger, if a user enters the wrong user name or password, there is a need to restart Studio.

- When upgrading a schedule with a start date in the past, the start date will be overridden with "now". This may also affect the number of occurrences.

- The end of a schedule is not computed properly. The Scheduler calculates DATE + 12:00 PM when it should be DATE + 12:00 AM.

- When a schedule has a recurrence range with the start date in the past, if you edit the schedule, this will cause the start date to be automatically updated, thus affecting the recurrence pattern.

- When you run content upgrade on a schedule with a simple trigger, the upgraded schedule may have a different time zone.

- Flow inputs that were added in the Scheduler window in previous versions are not upgraded.

- Some RSS readers may display the flow run duration inaccurately.

- During content upgrade to HP OO 10.00, flow inputs and outputs are not upgraded from previous versions for historical data. Only the flow result is migrated.

- After content upgrade to HP OO 10.00, the folder descriptions do not appear in Studio.

- During content upgrade to HP OO 10.00, if a flow has a required credential input identity, this flow is not upgraded.

- During an upgrade to HP OO 10.00, if there are multiple AD configurations, the installation process only upgrades one. The other AD configurations need to be added to HP OO 10.00 manually.

- In Studio, in some cases, the Debugger may use the wrong system account value when more than one project is loaded.

- When running a flow that is not deployed on Central in the Remote Debugger, if the run gets to a pause, the Debugger fails to resume after the pause.

- It is not possible to connect the Studio Remote Debugger to Central via HTTP proxy.

- Hand off has no indication in the Studio Debugger.

- The Remote Debugger does not support features that require user information on the context. This includes logged in user credentials,user data, roles, and gated transitions.

- The APIs **Delete flow schedule** and **Enable flow schedule** return 200 for non-existent schedules, instead of returning 404 Not Found (like GET /schedules/123).

- The GET/group-aliases REST API returns group aliases that are mapped into groups. The only way to map a group alias to a group is by using the POST/group-aliases REST API. If you deploy a content pack that contains group aliases that haven't been mapped into groups yet, they will not appear in the GET/group-aliases results.

- Flow execution fails on Oracle if you use a system account with an empty user.

- It is not possible to execute some WsWizard operations in Studio and Central.

- The following SOAP APIs are not supported in this version. Some of these methods can be used using REST API instead.

Configurations	getLWSSOConfig	Use REST API
	updateLWSSOConfig	Use REST API
Clusters	getClusterNodes	Not supported
Flows	getFlowInputDescriptions	Use REST API
	getFlowGraph	Not supported
Groups and Users Management	createGroup	Use REST API
	updateGroup	Use REST API
	deleteGroup	Use REST API
	getUserGroups	Use REST API
	createUser	Use REST API
	updateUser	Use REST API
	deleteUser	Use REST API

Repositories	getPermissions	Use REST API
	setPermissions	Use REST API
	getAttributes	Use REST API
	renameRepoEntity	Not supported
	deleteRepoEntity	Not supported
	moveFlow	Not supported
	updateDescription	Not supported
	createFolder	Not supported
	moveFolder	Not supported
Runs	getFlowsRunHistory	Use REST API
Scheduler	isScheduledFlowPaused	Use REST API
	isSchedulerPaused	Use REST API
	isSchedulerEnabled	Use REST API
	getSchedulesForFlowCategory	Use REST API
	pauseScheduledFlow	Use REST API
	pauseScheduledFlow	Use REST API
	pauseSchedule	Use REST API
	resumeSchedule	Use REST API
	scheduleFlow	Use REST API
	getSchedule	Use REST API
	deleteSchedule	Use REST API
	getScheduledFlows	Use REST API
	getSchedulesOfFlow	Use REST API
	resumeScheduledFlow	Use REST API
	deleteScheduledFlow	Use REST API
Selection Lists	getSelectionList	Use REST API

- There are a number of operations related to the getScheduledFlowsHP OO plugins in previous versions. These operations require engine support. Due to the limitations on the platform side, there are out-of-the-box operations and flows that are not supported at this time.

Cluster	Get Cluster Servers
	Get Server Status
Cross Run Data Persistence	Get Stored Flow Variable
	Store Flow Variable
Flow Execution	Get Run Status
	Get Status For Runs
	List Flow Run History
Repository	Check In
	Create Snapshot
	Delete Path
	Delete Snapshot
	Get All Children Named
	Get Children of Path
	Get Folder Flows
	Get Last Modified By
	Get References to Path
	Repository Sync
	Set Selection List
	Set System Account
	Set System Property
	Store System Account in Flow Variable
Scheduling	Delete Flow Schedule
	Get Flow Schedules
	Get Schedule Details
	Schedule Flow

Other	Acquire Lock
	Dynamically Launch Flow
	Flow Run Counter
	Flow Run Summary Report
	Generate Documentation
	Generate Documentation with Hidden Folders
	Generate Run URL
	Launch Flow
	Release Lock
	Resume Flow Run

Deprecation Notes

- The SOAP API has been replaced by a new REST API. For more information, see the *HP OO 10.00 API Guide*.

- The REST API that starts with /PAS has been replaced by a new REST API. For more information, see the *HP OO 10.00 API Guide*.

- The URL flow invocation (launch API) that starts with /PAS has been replaced by a new API that starts with /oo. For more information, see the *HP OO 10.00 API Guide*.

- The IAction interface for Java has been replaced by @Action. For more information, see the *HP OO 10.00 Extension Developers Guide*.

- The RSFlowInvoke and JRSFlowInvoke utilities have been replaced by the HP OO Shell Utility. For more information, see the *HP OO Shell Guide*.

Installing HP OO 10.00

Refer to the *HP OO 10.00 Installation Guide* for instructions on how to install HP OO.

Installation Notes

- If the user used to connect to the database does not have permission to create tables, you can run the scripts manually.You can use SQL scripts from the ISO image to manually create the tables and schema that are required for a remote installation.

  The SQL scripts located at **\docs\sql** on the ISO image. They include:

  - mssql.sql

  - mysql.sql

  - oracle.sql

  - postgres.sql

- The installation path does not support special characters.

- In a cluster with MySQL database, after the first node was installed (using the MySQL connector file), the installation succeeded and Central started. The second node was installed, and all required files were copied from the first Central and the MySQL connector file. After the installation was finished, Central couldn't start because of a missing JDBC driver.

  When the JAR file was copied to the **tomcat\lib** folder, Central started without any problem.