Hewlett Packard Enterprise

HPE Operations Orchestration

Software Version: 10.60
Windows and Linux Operating Systems

Upgrading to HPE OO 10.x from OO 9.x

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

Contents

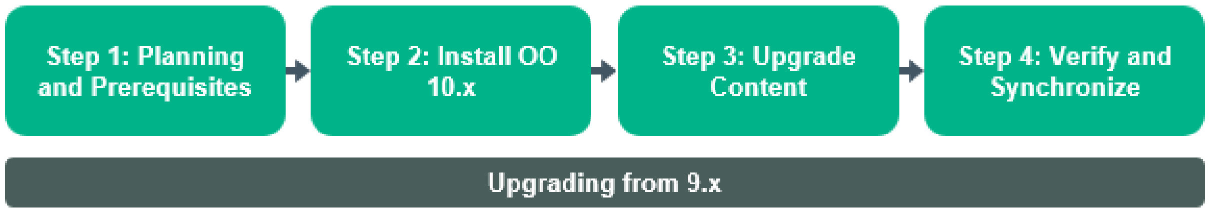
Upgrading from OO 9.x

This document is relevant for customers who are upgrading from OO 9.x to OO 10.x. The upgrade procedure supports OO 9.03 and later.

Note: If you are installing a clean installation of OO 10.6x, upgrading from a Community Edition, or upgrading from an earlier version of OO 10.x, see the *OO Installation, Upgrade and Configuration Guide*.

This procedure has multiple steps, so it has been broken down into several sections. Each section contains its own flowchart.

For information about what is and is not included in the upgrade process, see "Data Upgraded During Installation" on page 5.


Step 1: Planning and Prerequisites
Step 2: Install OO 10.x
Step 3: Upgrade Content
Step 4: Verify and Synchronize
Upgrading from 9.x

Step 1: Planning and Prerequisites

Step 2: Install OO 10.x

Step 3: Upgrade Content

Step 4: Verify and Synchronize

Notes:

- **LWSSO**: If you choose to upgrade the LWSSO settings from OO 9.x, these LWSSO settings will be migrated, but LWSSO will be disabled in OO 10.x (even if it was previously enabled in OO 9.x).
- The upgrade procedure does not modify the OO 9.x database and file system. OO 10.x requires a new schema during installation.
- The upgrade installs the Trial version of OO. You will need to install another license within 90 days. For more information, see "Setting Up Licensing" in the *Central User Guide*.

Data Upgraded During Installation

When you upgrade to HPE OO 10.x, the installation process automatically extracts most of the data that you need. Some data needs to be extracted manually, as described in "Step 4: Verify and Synchronize" on page 48.

Upgraded Security Data

The installation procedure extracts and loads the following LDAP parameters:

- LDAP URL
- List of LDAP contexts containing user groups
- LDAP search filter to match the user groups
- List of LDAP contexts containing users
- List of context attribute names that can be used as group
- LDAP search filter used in user search
- The default group that an LDAP authenticated user gets when there is no group matching
- If all enabled LDAPs have the same default role, this will be used as the default role. Otherwise, there will be a warning in the installer and no default role will be set.
- An internal OO account representing a user that has search capabilities under LDAP
- Active Directory domains (for AD only)

  Note: The installation automatically generates domains for non-AD configurations.

The LDAP credentials are encrypted in the upgrade files and in the HPE OO 10.x database.

If there are issues with upgrading an LDAP configuration, an error message appears in the installer.

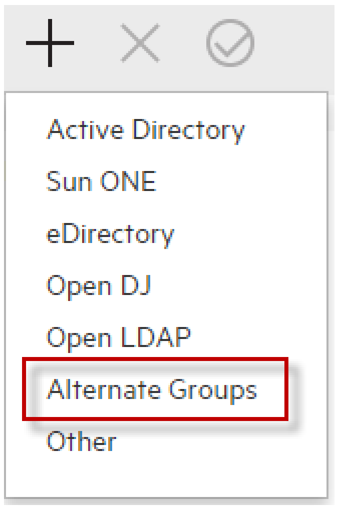
The installation also upgrades the LWSSO parameters.

Any 9.x LDAP configuration that is not Microsoft Active Directory will be migrated in 10.x as **Type**: **Other**.

Notes About Upgrading LDAP Information

- In OO 9.x, there was no option to enter a failover for an LDAP server. The workaround was to create another identical configuration with a different host and port. In OO 10.x, there is failover support, so there is no need for multiple configurations with the same domain name. Therefore, during upgrade, if there are multiple LDAP configurations where all parameters except host and port are identical, these are merged. A message appears, saying that the configurations are merged. In such a case, if you want to keep the multiple LDAP configurations with the same details, you will need to go to the **LDAP Configuration** tab in Central, check which configuration is missing, and add it manually.
- The installation procedure does not copy the 9.x **key.store**, and does not extract and load the Secure Reverse Proxy parameters. The **key.store** is ignored and Trust all is supported.
- The following are not upgraded:
  - Referrals (these are not supported)
  - Deprecated AD settings in an LDAP configuration

- If you use LDAP settings in 9.x and have configured something in the **List of LDAP contexts containing user groups** field, this causes the upgrade to import the LDAP type to **Alternate Groups** in HPE OO 10.x.



For more information about the **Alternate Groups** configuration, see the *Central User Guide* > "Setting Up Security - LDAP Authentication > Configure a list of attributes whose values will be used as groups".

Upgraded User Management Data

Upgraded Permissions for System Accounts

When you upgrade from 9.x to 10.x, the ADMINISTRATOR and PROMOTER roles are automatically entitled to view and use all system accounts. You can remove these default permissions in Central once the upgrade is complete.

Note: These default permissions are added on top of the ones you import from 9.x. See "Step 4: Verify and Synchronize" on page 48.

Upgraded Role Aliases

After an upgrade from 9.x, if a content pack contains the following role aliases (under **Configuration**/**Role Aliases**) ADMINISTRATOR, EVERYBODY, PROMOTER, SYSTEM_ADMIN, and END_USER, these are mapped to the corresponding OO 10.x roles in Central.

Note that some of the role aliases in the Base content pack (AUDITOR, LEVEL_ONE, LEVEL_TWO, and LEVEL_THREE) do not have a corresponding role in Central. After an upgrade to OO 10.x, these role aliases are considered deprecated.

Upgraded Permissions

After upgrade, the following permissions are mapped between 9.x and 10.x.

Note that in 9.x, roles were not case-sensitive and in 10.x, they are. After upgrading, you may see a message warning that the upgraded roles, which appear in upper case, are not consistent with the roles in 10.x. If this occurs, you may need to adjust the names of the roles.

9.x Permission	10.x Permission
MANAGE_ USERS	View Security Configuration, Manage Security Configuration
MANAGE_ GROUPS	View Security Configuration, Manage Security Configuration
AUTHOR	View Content Packs, Manage Content Packs, Manage Flow Permissions, Remote Debugging
SCHEDULE	Manage Schedules
MANAGE_ RUNS	Manage Others' Runs
MANAGE_ CONF	Manage Configuration Items, View Configuration Items, Manage System Settings, View System Settings, Manage Topology, View Topology, View Security Configuration, Manage Security Configuration, View Audit, Manage Data Cleanup
VIEW_ SCHEDULES	View Schedules

Note: The 9.x permissions HEADLESS_FLOWS and RUN_REPORTS are not mapped to any 10.x permissions.

Upgraded System Settings Data

The installation procedure extracts and loads the system settings, such as:

- Whether Return on Investment (ROI) reporting is visible.
- When using a URL to start flow runs in Central, the prefix, if any, that is required to start flow input names.

The following system settings are not upgraded, because they are not relevant in OO 10.x:

- Whether only administrators can author in the Central repository.
- The maximum number of versions of an object that are kept in the repository.
- Whether only members of the PROMOTER or ADMINISTRATOR group can publish.
- How frequently the Dashboard charts refresh.
- Whether to automatically resume headless runs that were interrupted by a Central server failure.
- How to enable load balancer management for a Central cluster.

Upgraded Configuration (Run Time Environment) Data

The installation procedure extracts and loads the system accounts and system properties that were configured in Central.

Note: Configuration items that were created in earlier versions of Central are upgraded and can be edited, even though it is no longer possible to create configuration items in Central.

Run time environment configuration data relating to Remote Action Services (RAS) is not upgraded, because it is not relevant in 10.x.

Upgraded RAS Data

In 9.x, an author could direct Central to a RAS by creating a RAS reference in Studio. The RAS reference had a name and a URL that accessed the RAS.

In 10.x, an author can create a RAS alias in Studio, and the administrator needs to map the RAS alias to a RAS group in Central.

During content upgrade, the RAS data is upgraded, and every RAS from the repository is upgraded to a new alias in the Studio project. For each RAS that existed for 9.x, you will need to manually install the RAS. Then, you will need to map the RAS alias to a RAS group in Central.

In 10.x, there is no need for two separate RASes for dot-net and another for Java. One RAS is enough for the same machine.

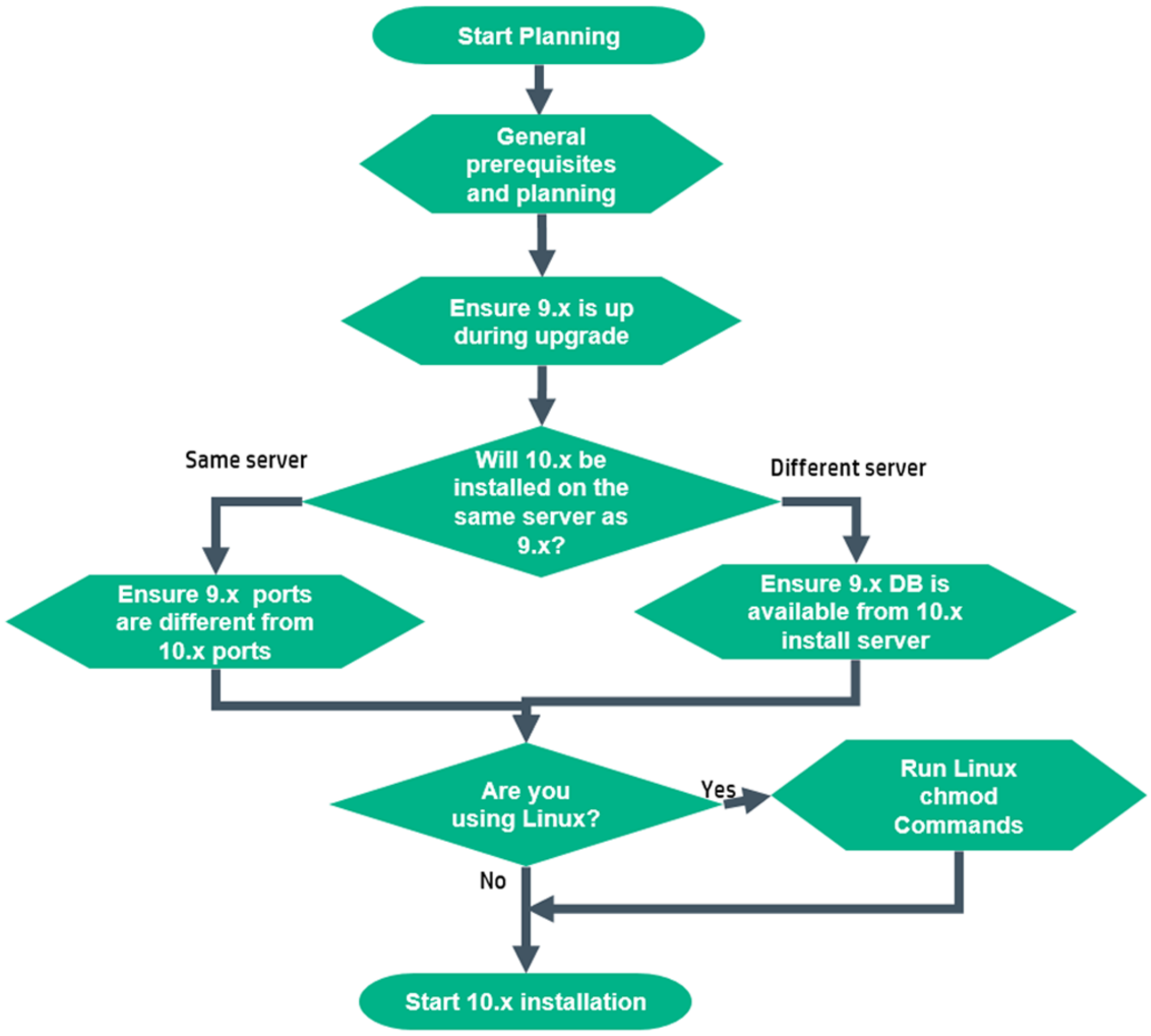
Upgraded LWSSO Settings

When you install 10.x, if you choose to upgrade the LWSSO settings from 9.x, these LWSSO settings will be upgraded, but LWSSO will be disabled in 10.x (even if it was previously enabled in 9.x).

Step 1: Planning and Prerequisites

The first step is to plan the upgrade and set things up so that the upgrade will run smoothly.

It is ***highly recommended*** to read this section and make sure that you have completed all the prerequisites before starting the upgrade.


Start Planning
General prerequisites and planning
Ensure 9.x is up during upgrade
Will 10.x be installed on the same server as 9.x?
Same server
Different server
Ensure 9.x ports are different from 10.x ports
Ensure 9.x DB is available from 10.x install server
Are you using Linux?
Yes
Run Linux chmod Commands
No
Start 10.x installation

Links to Steps

General Prerequisites and Planning
Ensure that 9.x is Up
Will 10.x be Installed on the Same Server as 9.x? • **Yes**: Ensure that the 10.x Ports are Different from the 9.x Ports • **No**: Ensure that the 9.x Database is Available from the 10.x Installation Server
Are you Using Linux? • **Yes**: Run the Linux chmod Commands • **No**: Go to the Next Step
Start the 10.x Installation

General Prerequisites and Planning

- Redesign your workflow for sharing files, because 10.x uses source control management instead of a public repository.
- Change the communication direction of firewall/reverse proxy on RASes, because on 9.x, Central is a client and RAS is a server, while on 10.x, RAS is a client and Central is a server.

  Note that some RASes may be firewall friendly (reverse RASes). In these cases, the communication direction is **Central > RAS**. For details on reverse RASes, see "Setting Up Topology – Workers and RASes" in the *Central User Guide*.

  For example, if Central exists inside the firewall/reverse proxy, and RAS exists outside, and you have a communication policy that only allow the communication direction from inside to outside, 10.x will violate your policy.

  You may find that you require new software for the firewall/reverse proxy .
- Check whether you need additional hardware for the upgrade. 10.x does not overwrite the existing installation of 9.x, but expects the previous version to remain as it is during the upgrade phase. This means if you want to continue to use the same hardware, versions 9.x and 10.x are installed on the same hardware. This could mean that the system requirements for memory cannot be achieved.
- Before starting the upgrade from 9.x to 10.x, you should take into account that there are many changes between the versions. Take a good look at these differences, as described in the *HPE OO 10.x Release Notes* and in this document, under "Differences in Behavior After Upgrading" on page 53.
- Before upgrading to 10.5x, you must download and install Microsoft Visual C++ 2010 Redistributable Package (x86). You need to install the version for the x86 platform, regardless of your Windows version.

  This package can be downloaded from: http://www.microsoft.com/en-us/download/confirmation.aspx?id=5555.
- Before you start the upgrade, it is recommended to estimate the effort that will need to be invested. You can do a test run of the Content Upgrade Utility on your repository. The **content-upgrade-report.html** file in the **Output \ Upgrade** folder will give you a detailed list of items that require action, and suggestions about how to resolve each type of problem. For more information about the report, see Viewing Information about the Content Upgrade.

- After content upgrade, all flows using operations from **/Library/Integrations/Hewlett-Packard/Operations Orchestration** will have their step references replaced with the corresponding operations from the **Library/Integrations/Hewlett-Packard/Operations Orchestration/10.x** folder.

  If you want to disable this behavior before running the Content Upgrade Utility, open the **/cmu/operation_references/operationReferenceReplacements.properties** file and comment (using #) the lines containing UUID correspondences, as shown below:

  ```
  #9.x/Dynamically Launch Flow > 10.x/Dynamically Launch Flow
  ```

  ```
  #0e227211-ffe5-4b24-8dd2-84071e3efa92=98102ebe-7a26-4398-afde-e01756578879
  ```

  For more information, see "Automatic Update of the OO to OO 10.x Integration" on page 74.

Back to the flowchart

Ensure that 9.x is Up

Ensure that 9.x is up during the upgrade process.

Back to the flowchart

Ensure that the 10.x Ports are Different from the 9.x Ports

When installing 10.x on the same server as 9.x, the 10.x ports must be different from the 9.x ports.

Make sure that you have two available ports for 10.x.

Back to the flowchart

Ensure that the 9.x Database is Available from the 10.x Installation Server

When installing 10.x on a different server from 9.x, ensure that the 9.x database is available from the 10.x installation server.

Back to the flowchart

Run Linux chmod Commands

In Linux, run the following commands from the installation root directory:

```
find . -name \*.sh -exec chmod 755 {} \;
find . -name \*java -exec chmod 755 {} \;
```

Back to the flowchart

Start the 10.x Installation

Once you have completed all the preparations for upgrade, proceed to "Step 2: Install HPE OO 10.x" on page 12.

Step 2: Install HPE OO 10.x

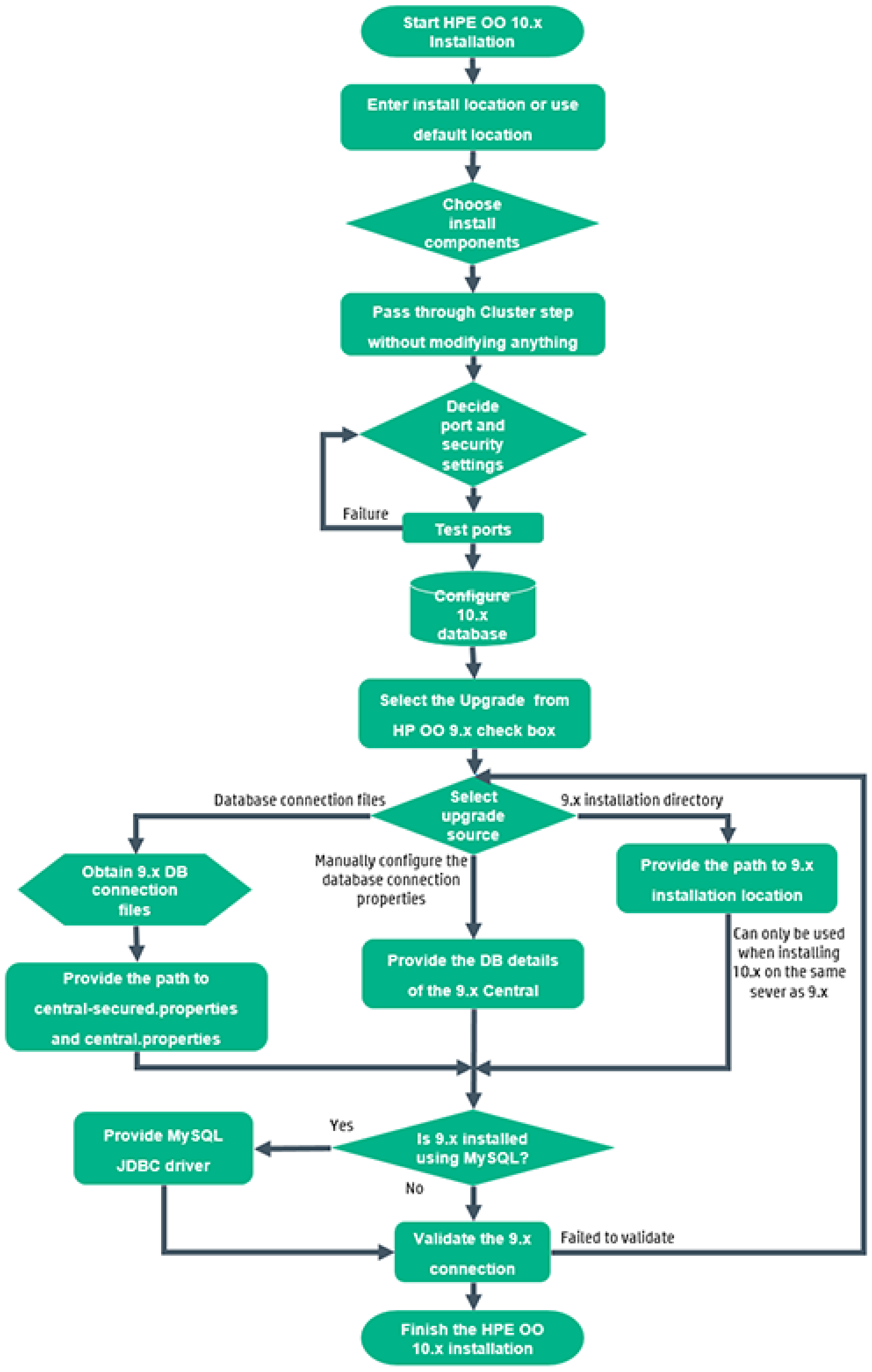
The next step is to install HPE OO 10.x, with the **Upgrade from 9.x** check box selected.

There are three options for selecting the upgrade source:

- Providing the path to the 9.x installation location - this option can be used if you are installing 10.x on the same server as 9.x
- Providing the 9.x connection files
- Manually configuring the database connection properties

Decide how you will be defining the upgrade source and then follow the relevant path on the flowchart.

If you are upgrading a cluster installation from 9.x, complete the upgrade of the first node, using the procedure described in this section. Then, install the additional nodes, using the procedure described in "Installing an HPE OO Central Cluster".


Start HPE OO 10.x Installation
Enter install location or use default location
Choose install components
Pass through Cluster step without modifying anything
Decide port and security settings
Failure
Test ports
Configure 10.x database
Select the Upgrade from HP OO 9.x check box
Database connection files
Select upgrade source
9.x installation directory
Obtain 9.x DB connection files
Manually configure the database connection properties
Provide the path to 9.x installation location
Can only be used when installing 10.x on the same sever as 9.x
Provide the path to central-secured.properties and central.properties
Provide the DB details of the 9.x Central
Yes
Provide MySQL JDBC driver
Is 9.x installed using MySQL?
No
Validate the 9.x connection
Failed to validate
Finish the HPE OO 10.x installation

Links to Steps

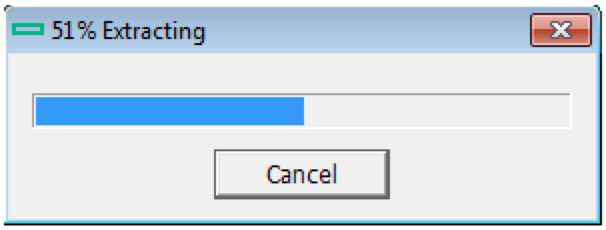
Start the HPE OO Installation
Enter the Installation Location or Use the Default Location
Choose the Installation Components
Pass Through the Cluster Step Without Modifying Anything
Decide Port and Security Settings
Test the Ports
Configure the Database
Select the Upgrade Source: • If you want to use the database connection files from 9.x, go to Provide the Path to central-secured.properties and central.properties • If you're installing 10.x on the same server as 9.x, and want to point to the 9.x installation location, go to Provide the Path to 9.x Installation Location • If you want to manually configure the database connection properties, go to Provide the DB Details of the 9.x Central
Was 9.x Installed Using MySQL? • **Yes**: Enter the path to the location of the JDBC driver. • **No**: Go to Validate the 9.x Connection
Validate the 9.x Connection
Finish the 10.x Installation

Start the HPE OO Installation

1. Download the ZIP file from the HPE SSO Portal and extract it into a local drive on your computer.
2. To start the installer:
   - On Windows: Double-click the **installer-win64.exe** installation file.
   - On Linux: Run this command from a Linux desktop/an X-Window terminal:

     ```
     bash installer-linux64.bin
     ```

     To start the installer, double-click the **installer-linux64.bin** file.
3. After you start the installer, the installation package is extracted, and the **HPE Operations Orchestration Installation and Configuration Wizard** automatically opens. Click **Next**.



4. In the **License** page, select **I Agree**, and then click **Next**.

Back to the flowchart

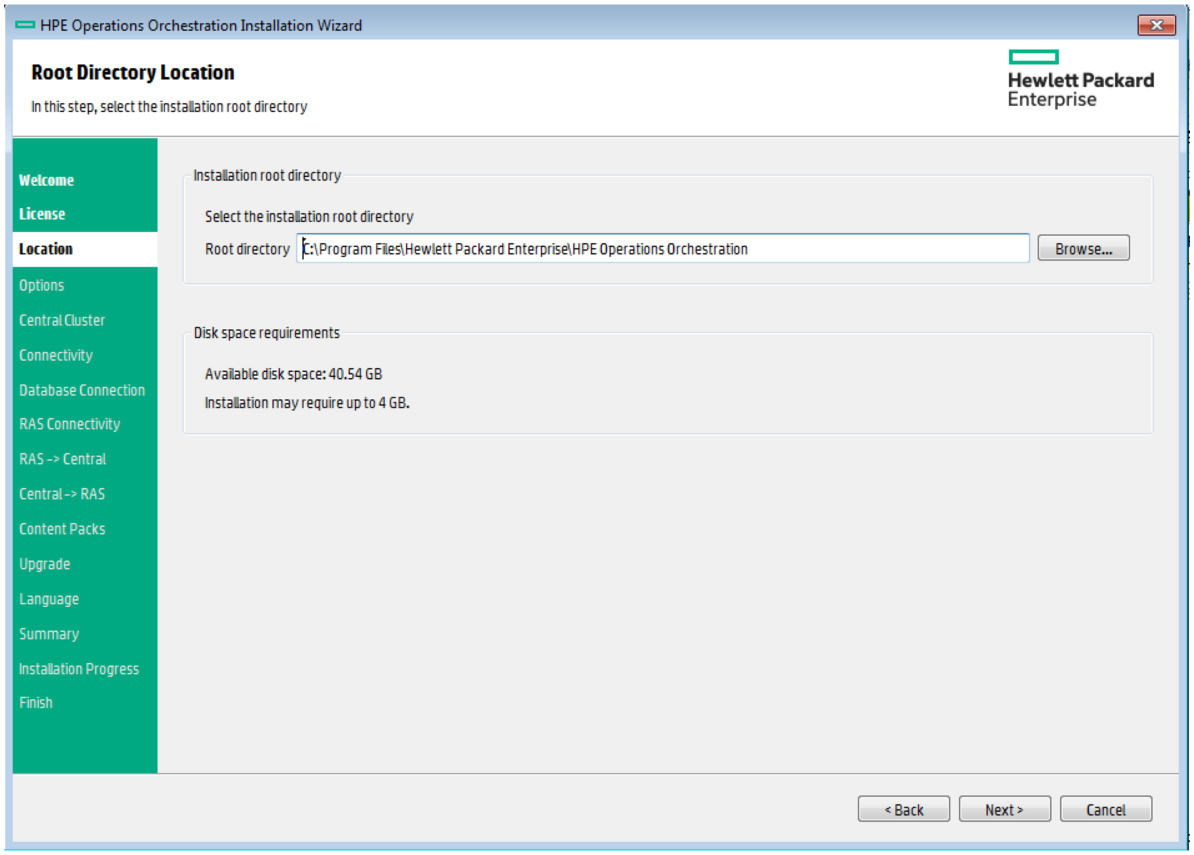
Enter the Installation Location or Use the Default Location

1. In the **Location** page, select the location for the installation root directory.

   If the directory does not exist, the directory is created automatically. You are prompted to confirm the creation of the new location.

   Note: Valid characters for the installation path are English letters, digits, spaces, hyphens (-) and underscores (_).

   The default path is `C:\Program Files\Hewlett-Packard Enterprise\HPE Operations Orchestration` for Windows and is `/opt/hpe/oo` for Linux.



Root Directory Location

In this step, select the installation root directory

Installation root directory

Select the installation root directory

Root directory: C:\Program Files\Hewlett Packard Enterprise\HPE Operations Orchestration

Disk space requirements

Available disk space: 40.54 GB

Installation may require up to 4 GB.

2. Click **Next**.

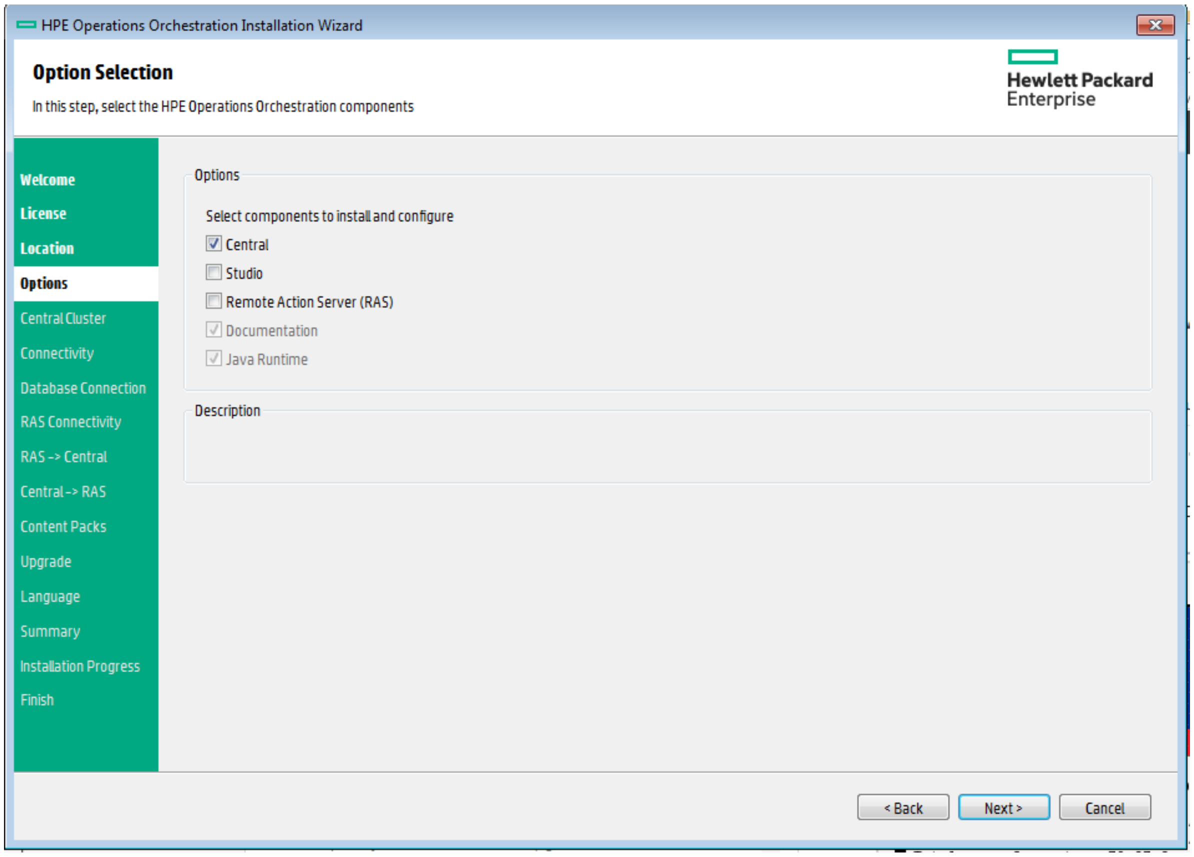
Back to the flowchart

Choose the Installation Components

1. In the **Options** page, select whether you want to install Central, Studio, RAS, or all three.

   Note: You can install Central without setting up a RAS server. If you install a RAS Server, it is recommended that you install this on a separate server to Central. See "Installing RAS Using the Installation Wizard" in the *Installation, Upgrade, and Configuration Guide*.

   See the *Architecture Guide* for more information about RASes.



Select components to install and configure

- [✓] Central
- [ ] Studio
- [ ] Remote Action Server (RAS)
- [✓] Documentation
- [✓] Java Runtime

2. Click **Next**.

Back to the flowchart

Pass Through the Cluster Step Without Modifying Anything

In the **Central Cluster** page, click **Next**.

For information about how to install a node in a cluster, see "Installing a Central Cluster" in the *Installation, Upgrade, and Configuration Guide*.

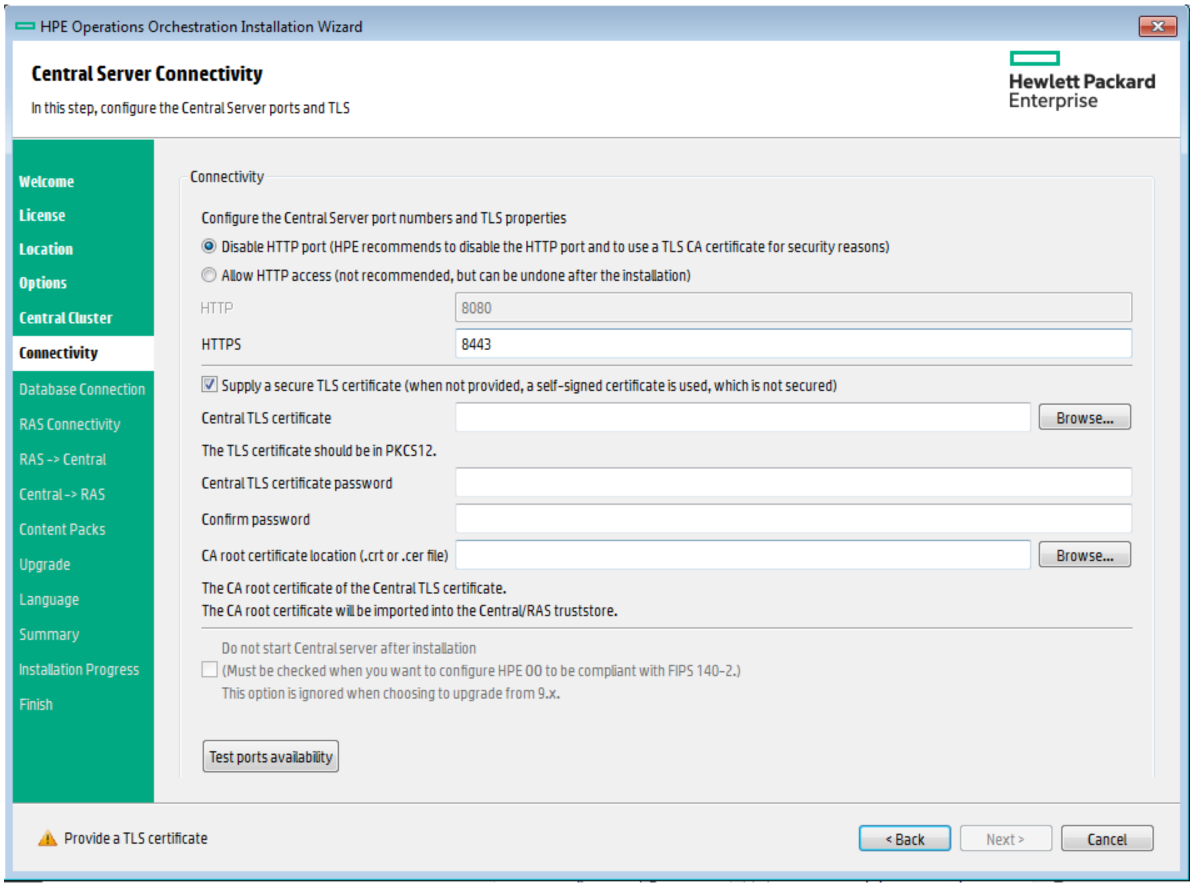
Back to the flowchart

Decide Port and Security Settings

Important! In an upgrade from 9.x, if you are installing 10.x on the same server as 9.x, ensure that the 10.x ports are different from the 9.x ports.

In the **Connectivity** page, configure the Central Server ports and SSL:

1. Configure available ports for the Central Server. Default values (8080 and 8443) appear for each port, but you can change these. Choose one of the following options:



Central Server Connectivity

In this step, configure the Central Server ports and TLS

Connectivity

Configure the Central Server port numbers and TLS properties

(●) Disable HTTP port (HPE recommends to disable the HTTP port and to use a TLS CA certificate for security reasons)

( ) Allow HTTP access (not recommended, but can be undone after the installation)

HTTP: 8080

HTTPS: 8443

[✓] Supply a secure TLS certificate (when not provided, a self-signed certificate is used, which is not secured)

Central TLS certificate

The TLS certificate should be in PKCS12.

Central TLS certificate password

Confirm password

CA root certificate location (.crt or .cer file)

The CA root certificate of the Central TLS certificate.
The CA root certificate will be imported into the Central/RAS truststore.

[ ] Do not start Central server after installation
(Must be checked when you want to configure HPE OO to be compliant with FIPS 140-2.)
This option is ignored when choosing to upgrade from 9.x.

Provide a TLS certificate

- (Recommended) Select **Disable HTTP Port** and configure a port in the **HTTPS** field.

  This option is recommended for security reasons, so that the communication channel is encrypted.

- (Not recommended) Select **Allow HTTP access** and configure two ports in the **HTTP** and **HTTPS** fields.

Note: Configuring at least one port is mandatory. If a port is not defined, or if the ports are occupied by other applications, you will not be able to complete the installation.

2. (Recommended) Select **Provide a secure TLS certificate**, and then click **Browse** to select the certificate.

   This step is recommended, for security reasons. If you do not select a Central TLS certificate, HPE OO uses the default self-signed certificate.

   Note: Do not use a network path for the location of the Central TLS certificate.

3. If you selected a Central TLS certificate, enter its password, and enter it again for confirmation.
4. Click **Browse** to specify the location of the CA root certificate, which will be imported into the TrustStore for Central/RAS.

   Note: Do not use a network path for the location of the CA root certificate.

For more information about installing HPE OO on a secured environment, see the *HPE OO Security and Hardening Guide*.

Note: The **Do not start Central server after installation** option is not available in an upgrade from 9.x.

Back to the flowchart

Test Ports

Click **Test ports availability**. If the ports are available, a **Success** check mark appears.

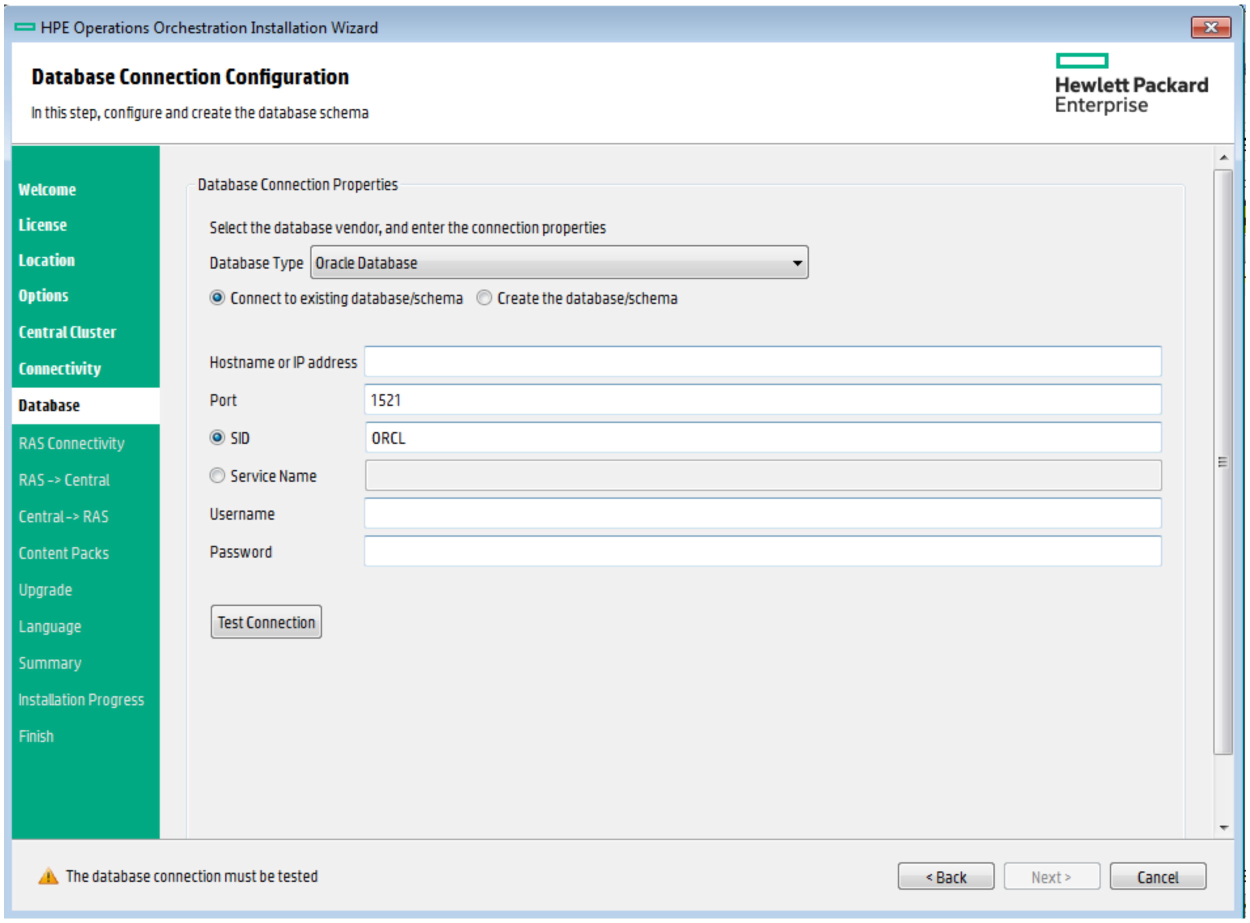
- If you encounter an error, adjust the ports accordingly and try again.
- If the **Success** check mark appears, click **Next**.

Back to the flowchart

Configure the Database

In the **Database Connection** page, you configure and create the database schema.

Note: If you have user input in two languages apart from English (for example, German and Chinese) then MS SQL should not be used. You should use an alternative database such as Oracle, MySQL, or Postgres with the recommended Unicode configuration for HPEOO.


HPE Operations Orchestration Installation Wizard

Database Connection Configuration

In this step, configure and create the database schema

Database Connection Properties

Select the database vendor, and enter the connection properties

Database Type: Oracle Database

[✓ Connect to existing database/schema] Create the database/schema

Hostname or IP address:

Port: 1521

[✓ SID]: ORCL

Service Name:

Username:

Password:

The database connection must be tested

1. From the **Database Type** list, select the database vendor, and then enter the connection properties.

   Note: When the **Connect to existing database/schema** option is selected, do not use administrative user accounts in the **Username** and **Password** fields, because this will install HPE OO under the administrative account.

   When the **Create the database/schema** option is used, provide a user with the relevant privileges in the **Admin username** and **Admin password** fields.

   You can select from the following database types:

   - **Oracle**: Do not use **SYS**, **SYSTEM**, or other administrative accounts credentials in the **Username** and **Password** fields.

     Note: If you are using Oracle 11g R2 or 11g R2 RAC, it is recommended to apply patch 20299013 before installing HPE OO.

   - **Microsoft SQL Server**: Do not use **sa** or other administrative account credentials in the **Username** and **Password** fields.

   - **Oracle MySQL**: Do not use the **root** credential in the **Username** and **Password** fields.

     If you are installing HPE OO with Oracle RAC (Real Application Cluster), you must choose **Other database** and provide the URL. For more information, see "Appendix B: Additional Guidelines for Oracle > Oracle Real Application Cluster (RAC)" in the *HPE OO Database Guide*.

   - **PostgreSQL**: Do not use the **postgres** credential in the **Username** and **Password** fields.

     Note: PostgreSQL database names are case-sensitive.

   - **Internal database**: This uses an H2 local database. This should not be used for production.

   - **Other database**: (use to enable advanced features in supported databases). If you select **Other database**, you can only use a database type that is supported for use with HPE OO. See "Appendix C: Installation Wizard 'Other Database' Option" in the *HPE OO Database Guide* for more information.

     Note: The **Other database** option also supports any valid JDBC URL.

2. After selecting the database type, select one of the following:

   - **Connect to existing database/schema**: Connect to an existing schema, user, or database. The installer verifies that the schema/database and user exist.

   - **Create the database/schema**: Enables you to create a new database or schema. Information in the **Database**, **Username** and **Password** fields will be used in order to create the new schema, user, or database for HPE OO.

     Confirm the password by typing it again in the **Confirm Password** field.

     Important! Make sure to use a strong password, in accordance with your organization's security policy. If the password is not strong enough, an error message will appear.

Provide existing database user credentials in the **Admin username** and **Admin password** fields. This elevated-privileges user must be able to connect to the database and create the new schema, user, or database for HPE OO.

DBA (Admin) credentials will only be used for creating the OO database and user/role. It is completely safe to provide these credentials, as they not saved and not used after the OO installation.

3. Enter the hostname or IP address and other connection details.

   Make sure to use the FQDN (Fully Qualified Domain Name).

   If you want to use IPv6, put the IPv6 address in brackets, for example, [3fff::20]. Otherwise, errors will occur.
4. (For Oracle) Select either **SID** or **Service Name**, and enter the SID or service name of the database.

   It is recommended to use Oracle database's service name rather than using the SID.

   Note: If you are upgrading from a 9.x version that is installed on Oracle, you must enter the SID of this database in the **SID** field, and not the database name.

For more information about setting up the database schema, see the *HPE OO Database Guide*.

If the database is MySQL, click **Browse** and select the location of the JDNC driver.

Click **Test Connection**. If you are unable to connect to the database, you will not be able to proceed to the next steps in the wizard.

If your password is not strong enough, a warning is displayed. You will still be able to proceed with the installation, but it is strongly recommended to replace it with a stronger password.

The installer checks for non-empty schemas/databases, and shows a warning message if the schema or database is not empty. If the installation fails during schema validation, the installation process is stopped.

Note: This test only verifies the connection between HPE OO and the selected database, and does not verify the conditions required by the database, like the user's read/write permissions on the provided schema.

Note: For all the database vendors, if you select to create a new database, the created database uses **case-sensitive** collation as follows:

- MySQL: **utf8_bin collation** is used for the new database.
- Postgres: Case-sensitive by design. No need for specific settings. **UTF-8** encoding is supported
- Oracle: Case-sensitive by default. No need for specific settings. **UTF-8** encoding is supported.
- MS SQL: Use only the following database collations per your required language:
  - English: SQL_Latin1_General_CP1_CS_AS
  - Japanese: Japanese_Unicode_CS_AS
  - Simplified Chinese: Chinese_Simplified_Stroke_Order_100_CS_AS
  - German: SQL_Latin1_General_CP1_CS_AS

- French: French_100_CS_AS
- Spanish: SQL_Latin1_General_CP1_CS_AS

However, if you already have a database installed, HPE OO creates the tables using the database specific collation. It is important to note that using other collations can cause characters to appear in gibberish in the user interface for localized installations. In addition, other collations are not officially supported in Microsoft SQL Server for localized installations.

If the installer is used in order to create a new SQL Server database, selecting your language in the language selection page sets the correct collation for the new database.

Using one of the above collations enables using the **varchar** datatype for textual columns instead of the **nvarchar** data type. Using the **varchar** data type is more efficient and reduces overall database size.

Selecting a specific language also means that an HPE OO system that uses SQL Server is limited to the set of languages supported by that specific collation. For example, if the **SQL_Latin1_General_CP1_CS_AS** collation is used, English, German, and Spanish characters may be used, but Japanese characters may not. If **Japanese_Unicode_CS_AS** is used, French accent characters will not be presented properly. For the complete specification of each collation, see the Microsoft SQL Server documentation.

Back to the flowchart

Select the Upgrade Source

1. In the **Upgrade** page, select **Upgrade from HPE Operations Orchestration 9.x**.
2. Continue to the next step, depending on how you have decided to define the upgrade source:
   - If you want to use the database connection files from 9.x, go to Provide the Path to central-secured.properties and central.properties
   - If you're installing 10.x on the same server as 9.x, and want to point to the 9.x installation location, go to Provide the Path to 9.x Installation Location
   - If you want to manually configure the database connection properties, go to Provide the DB Details of the 9.x Central

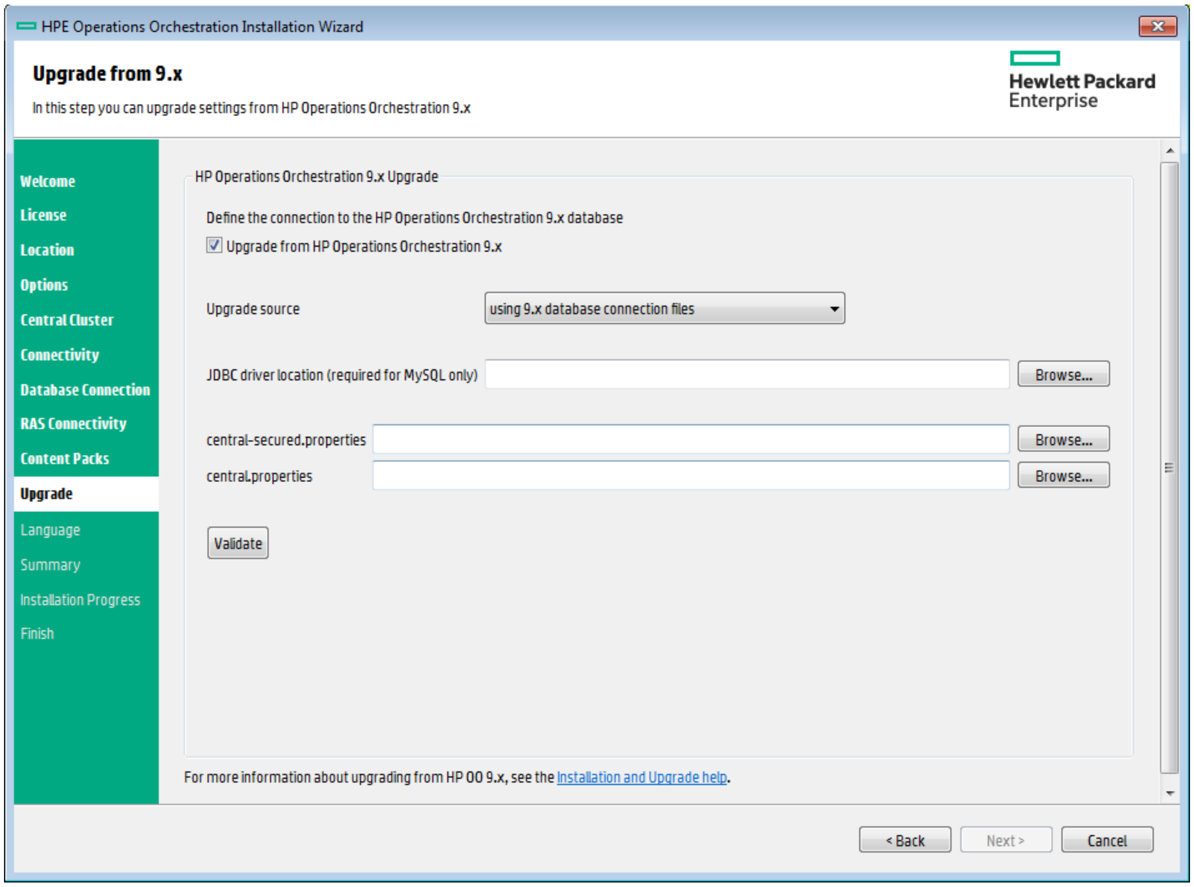
This step extracts and loads the system configuration information, such as users, LDAP, LWSSO, security data, system properties, and system accounts.

Back to the flowchart

Provide the Path to central-secured.properties and central.properties

Before selecting this option, make sure that you have access to the **central-secured.properties** file and the **central.properties** file from 9.x.

1. From the **Upgrade source** list, select **using 9.x database connection files**.



Upgrade from 9.x

In this step you can upgrade settings from HP Operations Orchestration 9.x

HP Operations Orchestration 9.x Upgrade

Define the connection to the HP Operations Orchestration 9.x database

[✓] Upgrade from HP Operations Orchestration 9.x

Upgrade source: using 9.x database connection files

JDBC driver location (required for MySQL only)

central-secured.properties

central.properties

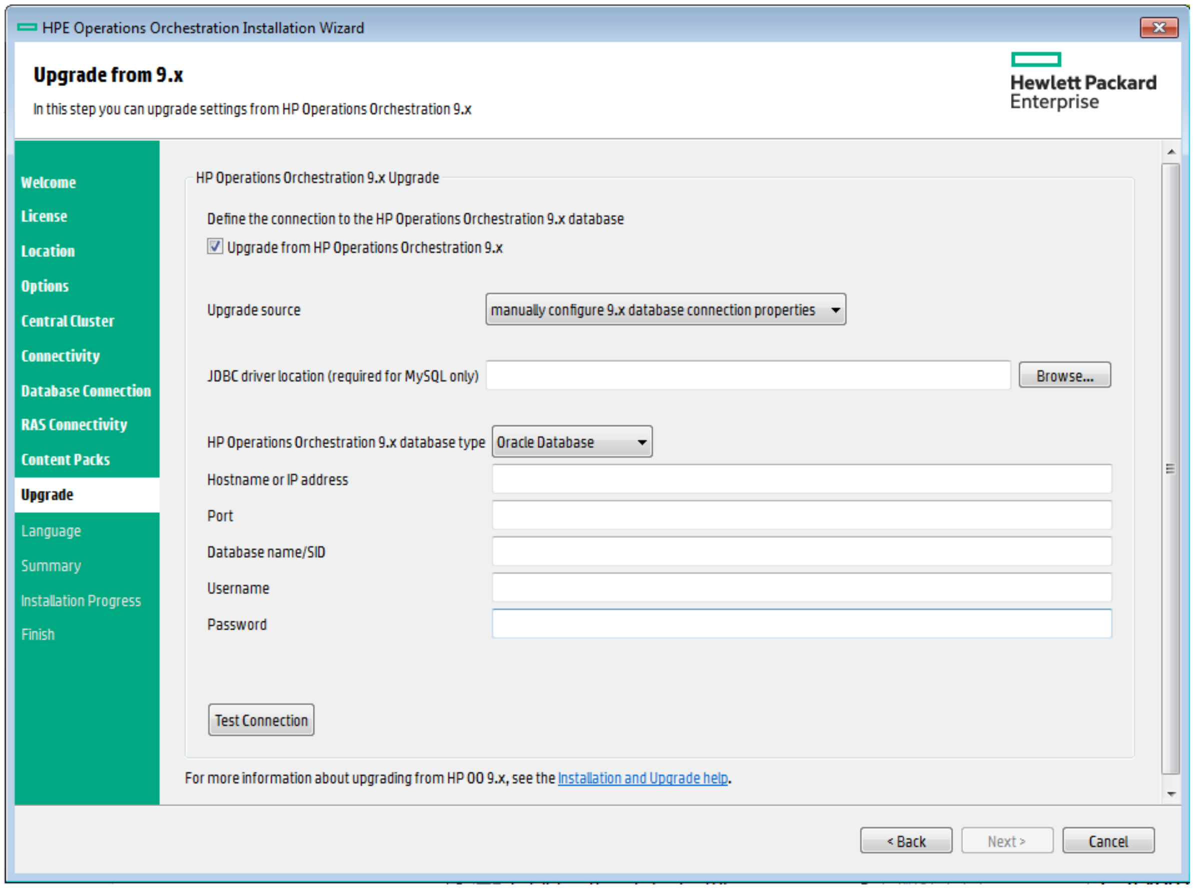
For more information about upgrading from HP OO 9.x, see the Installation and Upgrade help.

2. Provide the paths to the **central-secured.properties** file and the **central.properties** file from 9.x.
3. Click **Validate** to verify your 9.x version.
4. Click **Next**.

Back to the flowchart

Manually Provide the DB Details of the 9.x Central

1. From the **Upgrade source** list, select **manually configure 9.x database connection properties**.



Upgrade from 9.x

In this step you can upgrade settings from HP Operations Orchestration 9.x

HP Operations Orchestration 9.x Upgrade

Define the connection to the HP Operations Orchestration 9.x database

[✓] Upgrade from HP Operations Orchestration 9.x

Upgrade source	manually configure 9.x database connection properties
JDBC driver location (required for MySQL only)	
HP Operations Orchestration 9.x database type	Oracle Database
Hostname or IP address	
Port	
Database name/SID	
Username	
Password	

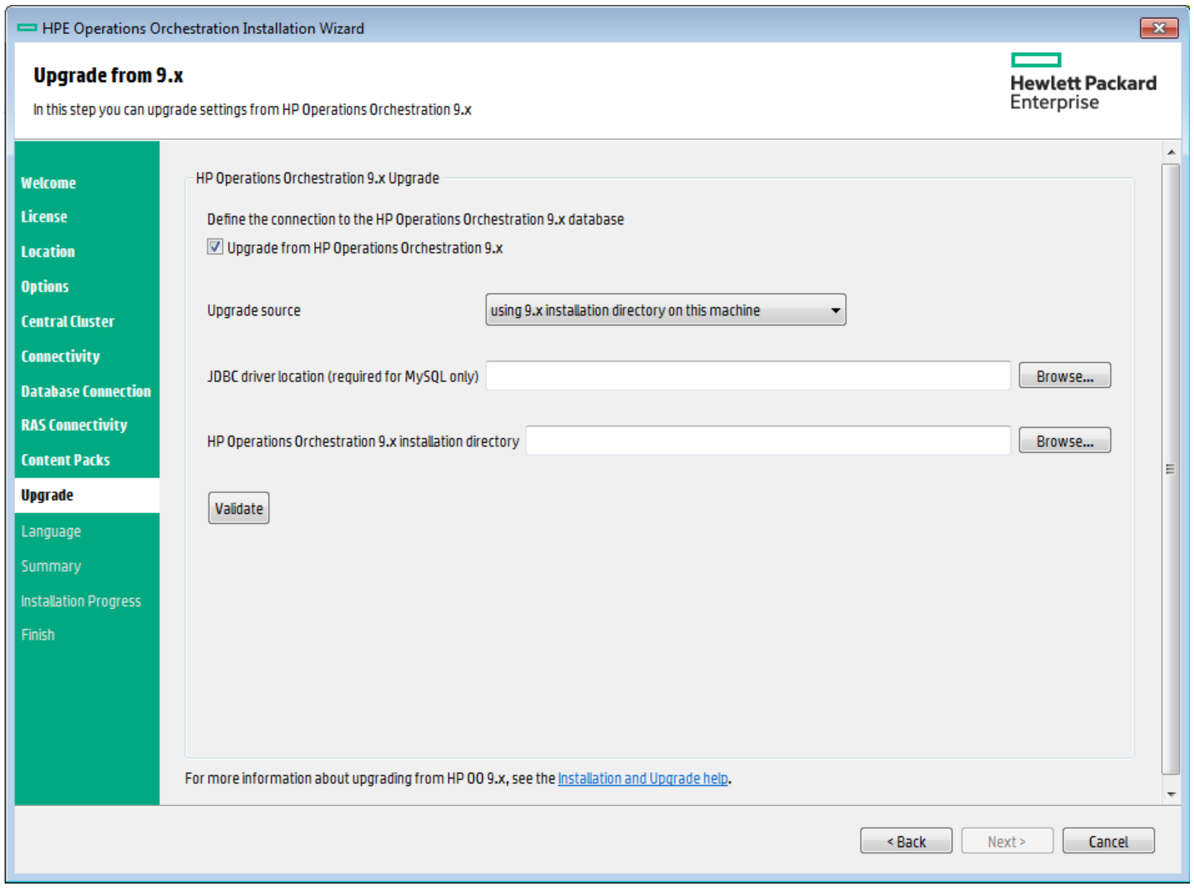
For more information about upgrading from HP OO 9.x, see the Installation and Upgrade help.

2. Select the database type and enter the hostname/IP address, port, user name, and password for the database.
3. Click **Test Connection** to verify the database properties.
4. Click **Next**.

Back to the flowchart

Provide the Path to 9.x Installation Location

1. If you are installing 10.x on the same server as 9.x, from the **Upgrade source** list, select **using 9.x installation directory on this machine**.


HPE Operations Orchestration Installation Wizard

Upgrade from 9.x

In this step you can upgrade settings from HP Operations Orchestration 9.x

HP Operations Orchestration 9.x Upgrade

Define the connection to the HP Operations Orchestration 9.x database

☑ Upgrade from HP Operations Orchestration 9.x

Upgrade source: using 9.x installation directory on this machine

JDBC driver location (required for MySQL only)

HP Operations Orchestration 9.x installation directory

For more information about upgrading from HP OO 9.x, see the Installation and Upgrade help.

2. Enter the installation directory for 9.x.

Back to the flowchart

Was 9.x Installed Using MySQL?

If yes, enter the path to the location of the JDBC driver (a JAR file used to connect to the MySQL database).

Back to the flowchart

Validate the 9.x Connection

The final step in the **Upgrade** page is to validate the connection to 9.x.

Depending on what you selected for Upgrade source, you will see either a **Validate** button or a **Test Connection** button. Click this button to verify that the installer can locate 9.x.

- If the validation fails, go back to Select the Upgrade Source and try again.
- If the validation is successful, click **Next** and go to Finish the OO10.x Installation.

Back to the flowchart

Finish the 10.x Installation

Complete the **Language** and **Summary** pages of the installer.

After upgrading to 10.5x, in order to use the Studio Git integration feature, you must install the Git client with version **git-1.9.5-preview20150319**.

1. Download the Git client from the following URL: https://github.com/msysgit/msysgit/releases/download/Git-1.9.5-preview20150319/Git-1.9.5-preview20150319.exe.
2. Save the Git client to **<oo_installation_folder>/studio/Git**, so that the **bin** folder is directly under **<oo_ installation_folder>/studio/Git**. In the Git installation wizard, use the default options.

Alternatively, if you already have a Git client installation with version **git-1.9.5-preview20150319** on your local disk, point Studio to use that Git installation by performing the following steps:

1. Close Studio.
2. Go to the user home folder **C:\Users\<user>\.oo** (the Studio workspace location) and locate the **Studio.properties** file.
3. Modify the **Studio.properties** file by adding the following property at the end of the file:

   ```
   studio.git.installation.location=<git-1.9.5-preview20150319_installation_folder>
   ```

   For example:

   ```
   studio.git.installation.location=C:/Program Files (x86)/Git
   ```

   The **bin** folder should be directly under **C:/Program Files (x86)/Git**. Note that **/** should be used as a path separator.
4. Save the **Studio.properties** file and start Studio.

Note: If you opted for this second alternative, you need to consider the following:

If you are using multiple workspaces and you want the Git location property to be added in each new workspace, you should edit the template properties file located in **Studio\conf\studio.properties.template**. Otherwise, each time you switch to a new workspace, you will have to set the Git location in the new workspace in the **.oo\Studio.properties** file.

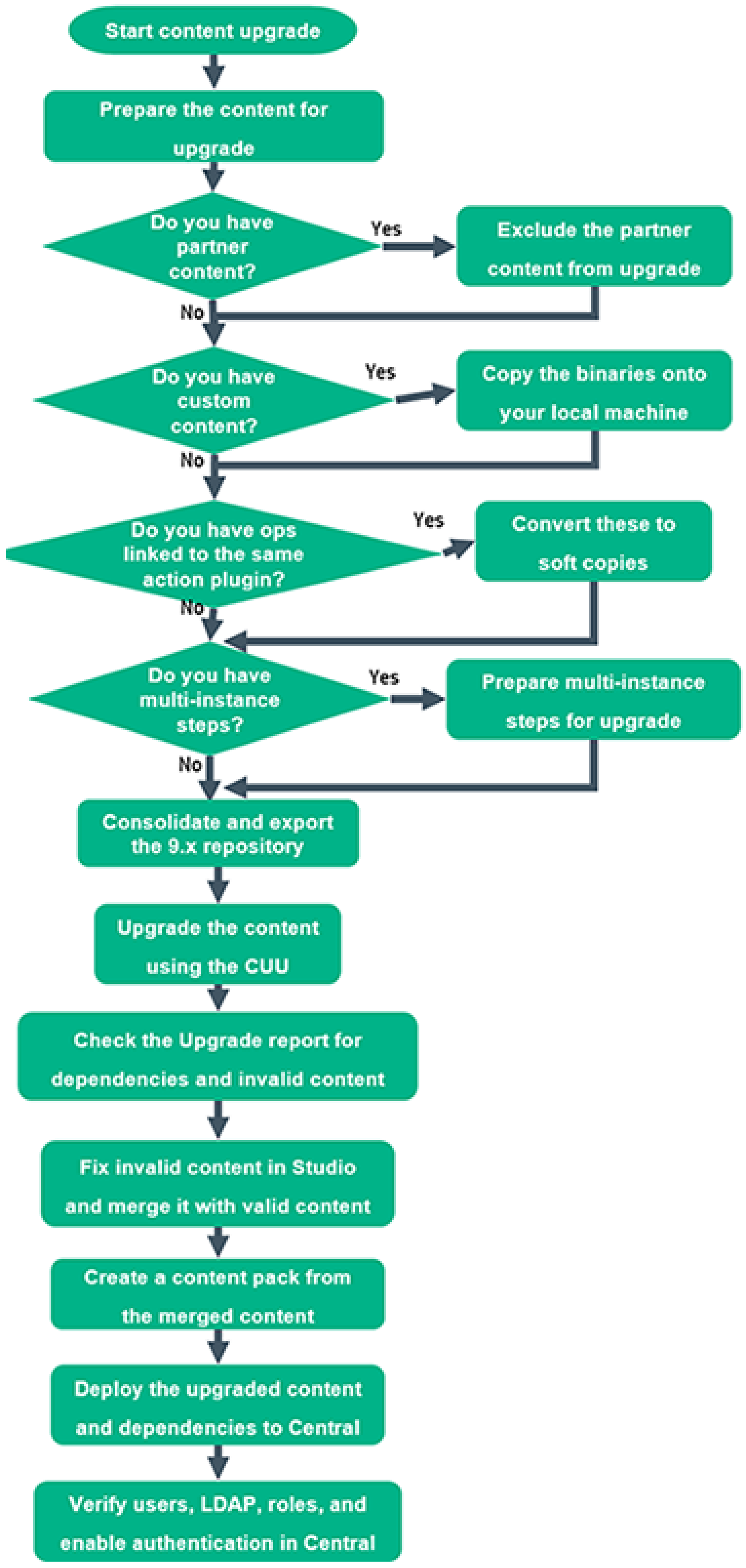
If you have another version of the Git client installed, note that you must use the **git-1.9.5-preview20150319** version with Studio. This is the version that was validated with Studio. While other versions might still work correctly, they are not officially supported.

Step 3: Upgrade Content

The next step is to upgrade the 9.x content to content in 10.x format.

This is done using the Content Upgrade Utility (CUU), which is included as part of the Central and Studio installations. The CUU extracts specified content from an existing 9.x repository and upgrades the extracted content into a content pack that can be deployed in 10.x.

All content packs that are supported by 9.03 and later are supported for upgrade.


Start content upgrade
Prepare the content for upgrade
Do you have partner content?
Yes
Exclude the partner content from upgrade
No
Do you have custom content?
Yes
Copy the binaries onto your local machine
No
Do you have ops linked to the same action plugin?
Yes
Convert these to soft copies
No
Do you have multi-instance steps?
Yes
Prepare multi-instance steps for upgrade
No
Consolidate and export the 9.x repository
Upgrade the content using the CUU
Check the Upgrade report for dependencies and invalid content
Fix invalid content in Studio and merge it with valid content
Create a content pack from the merged content
Deploy the upgraded content and dependencies to Central
Verify users, LDAP, roles, and enable authentication in Central

Links to Steps

Start Content Upgrade
Prepare Content for Upgrade
Do you have partner content? • **Yes**: Exclude Partner Content from Upgrade • **No**: Go to **Do you have custom content?**
Do you have custom content? • **Yes**: Prepare Custom Operations for Upgrade - Copy the Binaries to a Local Machine • **No**: Go to **Do you have operations linked to the same action plugin?**
Do you have operations linked to the same action plugin? • **Yes**: Prepare Custom Operations Linked to the same iAction - Convert Them to Soft Copies • **No**: Go to **Do you have multi-instance steps in your content?**
Do you have multi-instance steps in your content? • **Yes**: Prepare Multi-instance Steps for Upgrade • **No**: Go to **Consolidate and Export the 9.x Repository**
Consolidate and Export the 9.x Repository
Upgrade the Content Using the Content Upgrade Utility
Check the Upgrade Report for Dependencies and Invalid Content
Fix Invalid Content in Studio and Merge it with Valid Content
Create a Content Pack form the Merged Content
Deploy the Upgraded Content and Dependencies to Central

Start Content Upgrade

Content upgrade is done using the Content Upgrade Utility (CUU). But before you begin, you should prepare your content, in order to avoid problems.

Back to the flowchart

Prepare Content for Upgrade

General preparations on your 9.x content:

- Make sure that the content does not include invalid data. Invalid data will not be upgraded.
- Sleep scripts are not supported as scriptlets in 10.x. For example, under the **Scriptlet** tab or in the **Results** filter. You should rewrite these scripts in JavaScript before upgrading.
- Make sure that domain terms do not have more than 255 characters. Domain terms with too many characters can cause the upgraded content pack to fail deployment.

- Redesign/test client applications to call HPE OO flows, because of the discontinuation of **RSFlowInvoke**.
- Redesign/implement/test the monitoring system on 9.x, because the log messages in log files have changed.

Back to the flowchart

Exclude Partner Content from Upgrade

Partner content is content that is not owned and distributed by the HPE OO product team. It is developed and owned by other HPE product teams or by non-HPE companies.

During the content upgrade process, the Content Upgrade Utility (CUU) excludes OO out-of-the-box content from the upgrade. The CUU also recognizes the HPE partner content packs listed in the following table and excludes their content from the upload, making sure that only your custom content is upgraded.

If you have partner content that does not appear in the list, you will need to tell the CUU to exclude this content.

Note: The Content Upgrade Utility can read more than one properties file. This enables partners to supply a properties file, which you can place in the folder, as described below.

Partner Content that is Already Excluded from Upgrade

Content Pack Name	Availability / Location
HPE out-of-the-box content packs	
Base content pack	Available on the OO 10.x DVD and on HPE Live Network. See the *Release Notes* for more information on the these content packs.
Business Applications content pack	
Cloud content pack	
Databases content pack	
HPE Solutions content pack	
IT Operations content pack	
Middleware content pack	
Operating Systems content pack	
Virtualization content pack	
Cloud Orchestration content pack	
Infrastructure Orchestration content pack	
Partner content packs	
Server Automation content pack	This content pack is not included in the HPE OO 10.x DVD. Please check availability in HPE Live Network.
Storage Essentials content pack	This content pack is not included in the HPE OO 10.x DVD. Please check availability in HPE Live Network.

Content Pack Name	Availability / Location
NonStop content pack	This content pack is not included in the HPE OO 10.x DVD. Please check availability in HPE Live Network.
Cloud Services Automation content pack	This content pack is not included in the HPE OO 10.x DVD. Please check availability in HPE Live Network.
Service Manager content pack	This content pack is not included in the HPE OO 10.x DVD. Please check availability in HPE Live Network.
HPE Community content packs	
BMC content packs (Atrium, Patrol, Remedy)	HPE community content packs are available for download on HPE Live Network.
Cisco content pack	
Citrix Presentation Server content pack	
Computer Associates content pack	
F5 content pack	
HPE Peregrine Service Center content pack	
IBM content packs (Netcool, Tivoli)	
ITIL content pack	
Microsoft Operation Manager content pack	
Symantec content packs (Altiris, Backup Exec)	
VMware Server content pack	

Exclude the Upgrade of Partner Content on a Central Installation

1. Go to HPE Live Network and locate the relevant content provider.

   Note: Try starting at the content catalog.

2. Download the content pack, making sure to download the file **<CP Name>_oo10cuu-plugins.zip**. If this file does not exist, contact the partner to ask for it.
3. Unzip the **<PARTNER PLUGINS ZIP FILE NAME>** to a temporary folder on your file system (for example, **C:\Temp**).
4. From the temporary folder, copy the file **<PARTNER CP NAME>** to the following location:
   - Windows: **<OO Installation Directory>\central\cmu\exclusions**
   - Linux: **<OO Installation Directory>/central/cmu/exclusions**

Note: Before you copy the file, check if there is an older version of the file and delete it.

5. From the temporary folder, copy the file **<PARTNER CP NAME>_Conversions.properties** to the following location:
   - Windows: **<OO Installation Directory>\central\cmu\plugin-mappings**
   - Linux: **<OO Installation Directory>/central/cmu/plugin-mappings**

   Note: Before you copy the file, check if there is an older version of the file and delete it.

6. From the temporary folder, copy the file **<PARTNER CP NAME>_plugin2oo.properties** to the following location:
   - Windows: **<OO Installation Directory>\central\cmu\plugin2op_mappings**
   - Linux: **<OO Installation Directory>/central/cmu/ plugin2op_mappings**

   Note: Before you copy the file, check if there is an older version of the file and delete it.

7. When you run the Content Upgrade Utility, it will recognize the partner content.

After you have completed the content upgrade, you will need to deploy the partner content pack first, before deploying the upgraded content pack.

Exclude the Upgrade of Partner Content on a Studio Installation

1. Go to HPE Live Network and locate the relevant content provider.

   Note: Try starting at the content catalog.

2. Download the content pack, making sure to download the file **<CP Name>_oo10cuu-plugins.zip**. If this file does not exist, contact the partner to ask for it.
3. Unzip the **<PARTNER PLUGINS ZIP FILE NAME>** to a temporary folder on your file system (for example, **C:\Temp**).
4. From the temporary folder, copy the file **<PARTNER CP NAME>** to the following location:
   - **<OO Installation Directory>\studio\cmu\exclusions**

   Note: Before you copy the file, check if there is an older version of the file and delete it.

5. From the temporary folder, copy the file **<PARTNER CP NAME>_Conversions.properties** to the following location:
   - **<OO Installation Directory>\studio\cmu\plugin-mappings**

   Note: Before you copy the file, check if there is an older version of the file and delete it.

6. From the temporary folder, copy the file **<PARTNER CP NAME>_plugin2oo.properties** to the following location:
   - **<OO Installation Directory>\studio\cmu\plugin2op_mappings**

Note: Before you copy the file, check if there is an older version of the file and delete it.

7. When you run the Content Upgrade Utility, it will recognize the partner content.

Back to the flowchart

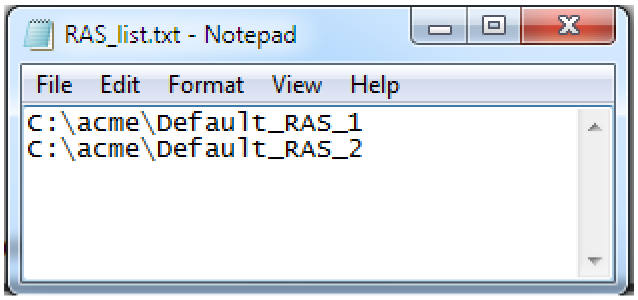
Prepare Custom Operations for Upgrade - Copy the Binaries to a Local Machine

This step is only relevant for the upgrade of a repository that also contains custom operations that point to iActions you have developed, whether in Java or .Net.

When you upgrade your repository, the upgraded content pack contains custom flows and operations, and a populated **Lib** folder containing custom iActions JARs/dlls, Maven plugins (wrappers for each custom JAR/dll) and third party dependencies in Maven repository format. For details on the Maven repository, see "Create a Local Maven Repository" in the *Action Developers Guide*.

1. If you have more RASes that are installed on different machines, copy the binaries onto your local machine, into different folders corresponding to the RASes.
2. Create a file containing the paths to these local folders. On each line of the file, write the path to a local folder that contains the content of **%ICONCLUDE_HOME%\RAS\Java\Default** for each of your RASes.

   For example:



```
C:\acme\Default_RAS_1
C:\acme\Default_RAS_2
```

3. When you run the Content Upgrade Utility, you will be able to add this file as input, using the argument `-rf,--rases-file`. This is described in Upgrade the Content Using the Content Upgrade Utility.

Note: If your repository includes custom .NET iActions, third party DLL dependencies for 32-bit platforms in the client RAS may cause problems. For example, **Microsoft.GroupPolicy.Management.dll** and **Microsoft.GroupPolicy.Management.Interop.dll**.

It is recommended to clean your RAS folder copy and leave only the DLLs that are required for your iActions before upgrading content.

Make sure to copy the RAS lib directories to a different location before deleting anything from them and upgrading them.

Back to the flowchart

Prepare Custom Operations Linked to the same iAction - Convert Them to Soft Copies

If you have multiple custom operations that are linked to the same iAction, consider changing these to soft copies before upgrading.

To convert a customer operation to a soft copy, you will need to create two property files:

1. Create a property file in **cmu/plugin_mappings**, which maps between the custom content jars to the plugin that was created by the CUU (you can see its GAV in Studio, on the action that uses it).

   For example, if the custom actions jar file is **customAction.jar**, you need to create a **oo10-customAction-cp-Conversions.properties** file and add the following line:

   ```
   customAction.jar=customer:oo-custom-action-legacy-plugin:1.0.0
   ```

2. Create a property file in **cmu/plugin2op_mapping**, which maps between the plugin and the operation.

   For example, if the operation that uses it is **ce8fce1a-c2d3-45c5-b87c-ec36c047a8c1**, create the **oo-10-cutomAction-cp-plugin2op.properties** file and add the following line:

   ```
   customer\:oo-custom-action-legacy-
   plugin\:1.0.0\:com.example.actions.CustomAction=ce8fce1a-c2d3-45c5-b87c-ec36c047a8c1
   ```

Back to the flowchart

Background about Hard Copies and Soft Copies

Hard Copy

In versions of OO prior to 10.x, when you copied an operation linked to an action plugin, this created a **hard copy**. The copied operation was directly linked to the action plugin in the same way that the original was. When the action was updated—for example if the name of the JAR or the class was changed—this had to be updated in all the hard copied operations.

In 10.x, hard copies are not created by default, as they were in previous versions. To create a hard copy, you need to create a new operation and select the relevant plugin. This method creates a new operation according to the IAction getTemplate or the @Action metadata.

Soft Copy

In 10.x, when you copy an operation linked to an action plugin, this creates a **soft copy**. This means that the copied operation continues to reference the original operation. If the action plugin is upgraded, when you update the original operation to call the new version, the copied operations are all updated automatically.

How Operations are Upgraded

When content is upgraded from 9.x, operations linked to action plugins are upgraded as follows, by default:

Type of Operation	Upgraded As
Copies of out-of-the-box operations	Soft copy
Custom operations	Hard copy
Operations linked to out-of-the-box JARs that were copied and renamed	Hard copy

For hard copied operations:

- When a plugin gets a new version (for example, after a bug fix), this plugin version must be updated in all hard copies.
- If the original operation resides in content pack 1 and the hard copy resides in content pack 2, the plugin shared by the two copies will be duplicated and placed in both content packs. This results in some redundancy.

If you have multiple custom operations that are linked to the same action plugin, you can upgrade these as soft copies, by adding your own mappings to the **plugin2op.properties** file.

1. Open the **\plugins_mapping\plugin2op.properties** in a text editor.
2. Add a line to the file, identifying the Jar name and version, the action class name, and the UUID of the parent operation.

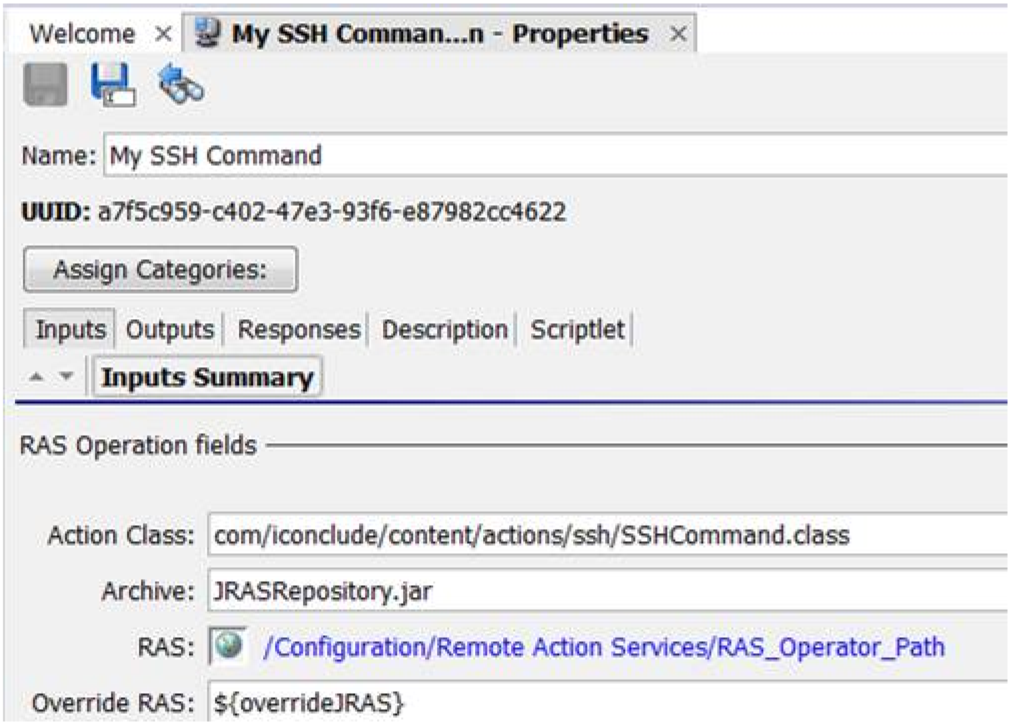
   ```
   com\:<jar file name>\:<jar file version>\:<action class name>=<UUID of the parent
   operation>
   ```

For example, if the operations you want to upgrade as soft copies are linked to an action plugin called **Custom.jar**, the line would be as follows:

```
com\:Custom\:1.0 \:com.hp.oo.content.actions.xen.tasks.FindTasks=09b0041d-e190-4fae-
8b4e-6f112cb99e3e
```

- `Custom` is the name of the Jar file
- `1.0` is the version of the Jar file
- `com.hp.oo.content.actions.xen.tasks.FindTasks` is the action class name
- `09b0041d-e190-4fae-8b4e-6f112cb99e3e` is the UUID of the parent operation

To locate the UUID, check the **Inputs Summary** of the parent operation.



Name: My SSH Command

UUID: a7f5c959-c402-47e3-93f6-e87982cc4622

Action Class: com/iconclude/content/actions/ssh/SSHCommand.class

Archive: JRASRepository.jar

RAS: /Configuration/Remote Action Services/RAS_Operator_Path

Override RAS: ${overrideJRAS}

Prepare Multi-instance Steps for Upgrade

The architecture of multi-instance steps has changed in 10.x. The changes include:

- Steps that run within multi-instance step branches (internal steps) have access only to branch context and not to the overall step and global contexts of the flow. This affects, among other things, the work of **List Appender** operations used to aggregate branch results.
- Usages that were considered bad practice in 9.x are not supported in 10.x, as follows:

- Direct transitions from the internal steps of a multi-instance step to return steps are not supported in 10.x. Only the **Group-done** transition of the multi-instance steps can be used for this purpose.
- Parallel steps cannot be used as internal steps in a multi-instance step.
- Other multi-instance steps cannot be used as internal steps in a multi-instance step.

If you have multi-instance steps that use any of the above practices, it is recommended to change these before upgrading the content.

For information about fixing multi-instance steps after upgrade, see "Step 4: Verify and Synchronize" on page 48.

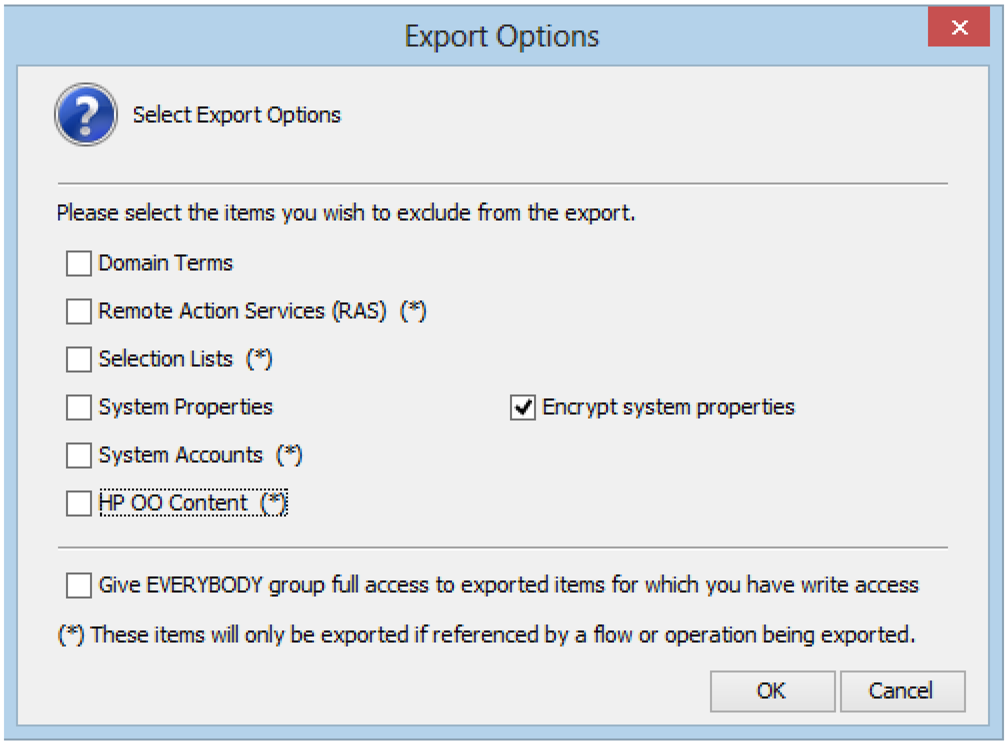
Back to the flowchart

Consolidate and Export the 9.x Repository

1. In Studio 9.x and consolidate all the content that you want to upgrade into one repository.
2. Make sure that all items are checked in.
3. To export your 9.x repository, right-click **Library**, and select **Export as New Repository**.

   Important Note: In order to upgrade the system accounts with a password, you must not perform an export of the repository, but rather create a file system copy of it and perform the content upgrade on that copy. For example, for a simple Central installation, use a copy of the **<OO_9_install_ path>\Central\rcrepo** folder as the repository for content upgrade.

4. In the file explorer dialog box, select the output path for the repository.
5. In the Export Options page, make sure that all the check boxes are cleared (not selected) apart from the **Encrypt system properties** check box. Checking the **Encrypt system properties** check box is optional. However, note that after content upgrade, the system properties will be obfuscated with the 10.x mechanism.



Please select the items you wish to exclude from the export.

- [ ] Domain Terms
- [ ] Remote Action Services (RAS) (*)
- [ ] Selection Lists (*)
- [ ] System Properties
- [✓] Encrypt system properties
- [ ] System Accounts (*)
- [ ] HP OO Content (*)

- [ ] Give EVERYBODY group full access to exported items for which you have write access

(*) These items will only be exported if referenced by a flow or operation being exported.

Make sure you know the path to the exported repository, and where you want to store the content pack result after the upgrade.

Back to the flowchart

Upgrade the Content Using the Content Upgrade Utility

The Content Upgrade Utility (CUU) is included as part of the Central and Studio installations. After installation, the utility can be found under:

- **<OO Installation Directory>\Central\bin\upgrade-content.bat** for Windows
- **<OO Installation Directory>\Studio\bin\upgrade-content.bat** for Windows
- **<OO Installation Directory>/Central/bin/upgrade-content.sh** for Linux

The Content Upgrade Utility configuration files are located under:

- **<OO Installation Directory>\central\cmu\** for Windows
- **<OO Installation Directory>\studio\cmu\** for Windows
- **<OO Installation Directory>/central/cmu/** for Linux

1. Run the bat/sh script that launches the Content Upgrade Utility.

   For Windows: `>upgrade-content.bat`

   For Linux: `>./upgrade-content`

2. Specify the upgrade options in the command line. The following commands are available:

   - `-d, --desc <arg>`

     Use to include brief description of the content pack. The description is used in the content pack meta data properties file.

     Example: `--desc Content pack containing Acme flows and operations`

     Default: Upgraded content pack

- `-h, --help`

  Displays help information about the different command line options.

- `-i, --repo <arg>`

  Use to specify the folder containing the 9.x repository. Point to the root directory of the 9.x project.

  If `--repo` is not defined correctly, there will be problems with the upgrade.

  If the repository is stored under a parent folder and –repo points to this parent folder rather than to the folder containing the repository itself, the repository will not be upgraded. The following error will appear:

  ```
  Invalid repository input argument. Pathrefs.xml file not found.
  ```

- `-ip, --include-passwords`

  Use to include system account passwords in the content upgrade. When the content is deployed on Central, the user names and passwords will be deployed. Note that the passwords will be obfuscated inside the project/content pack.

  Important note: In order to upgrade system accounts with a password, you must not perform an export on the repository, but rather create a file system copy of it and perform the content upgrade on that copy.

  For example, for a simple Central installation, use a copy of the **<OO_9_install_path>\Central\rcrepo** folder as the repository for content upgrade.

- `-n, --name <arg>`

  Use to enter the name that will be used for:

  - the directory containing all the content that can be deployed and run
  - the content pack archive (jar file)
  - the content pack metadata properties file

  Default: Customer

- `-ncp, --nocp`

  If this option is used, a content pack archive will not be created.

- `-nine,--nine-cp-version <arg>`

  Use to specify the 9.x content pack version that you have installed on the environment from where the 9.x repository was exported.

  If you made changes to any out-of-the-box configuration items (such as selection lists, domain terms, and so on) you must use this argument; otherwise, your changes will be lost. After upgrade, another configuration item is created with a different UUID and with the name changed to "custom_<previousName>".

  Note: This additional item is created only for configuration items that have changes, compared to the version specified in the `-nine` argument.

  Any use of the original item is replaced with the version with your changes.

Note: If you changed only the name of the configuration item and not the value, this will not be considered as a change and the item will not be duplicated.

For example, the **Character Sets** selection list contains items such as **UTF-8**, **UTF-16**, **UTF-32**, and so on. If you added **UTF-7**, you must use the `-nine` argument in order to avoid losing this item after upgrade. A new selection list with the same name will be created in your repository and any use of the out-of-the-box **Character Sets** list will be replaced with the new **Character Sets** list.

This argument can accept one of the following values:

```
"7.60.01", "7.60.02", "9.00.03", "9.00.04", "9.00.05", "9.00.06", "Content Pack 7"

"Content Pack 8", "Content Pack 9", "Content Pack 10", "Content Pack 11", "Content
Pack 12", "Content Pack 13"
```

- `-nui, --no-user-interaction`

  This option disables the console.

- `-o, --output <arg>`

  Use to enter the output location, in which the content pack and the **UUID/jar/dll** mapping XMLfile will be stored.

  Note: If there is already an output directory in this location, the Content Upgrade Utility will delete everything that was in it previously. If the directory does not exist, the Content Upgrade Utility will create it.

  Default: The current directory

- `-p, --publisher <arg>`

  Use to enter the name of the customer. The publisher name is used in the content pack meta data properties file.

  Default: Customer

- `-rf,--rases-file <arg>`

  Use to upgrade custom IActions. Indicate the path to a file containing on each line the path to a local or shared network folder that contains the content of `%ICONCLUDE_HOME%\RAS\Java\Default` for each of your RASes.

  If you had RASes that were installed on different machines, and you copied the binaries onto your local machine, and created a file containing the paths to these local folders, as described in Prepare Custom Operations for Upgrade - Copy the Binaries to a Local Machine, use the argument `-rf,--rases-file` to add the file as input to the CUU.

  For example, if the file with the path to the RAS folders is called **RAS_list.txt** and is stored in the **C:/acme** folder:

  ```
  --rases-file c:/acme/RAS_list.txt
  ```

- `-t, --temp-dir <arg>`

  Use to set the temp directory, overriding the default temp directory.

  Default: The value from `java.io.tmpdir`.

- `-v, --version <arg>`

  Use to enter the version for the content pack. The version is used in the content pack meta data properties file.

  Default: 1.0.0

An example of upgrade options in Windows:

```
>upgrade-content.bat --name AcmeContent --publisher Acme --repo
C:\old\OldRepositories --output D:\NewRepositories
```

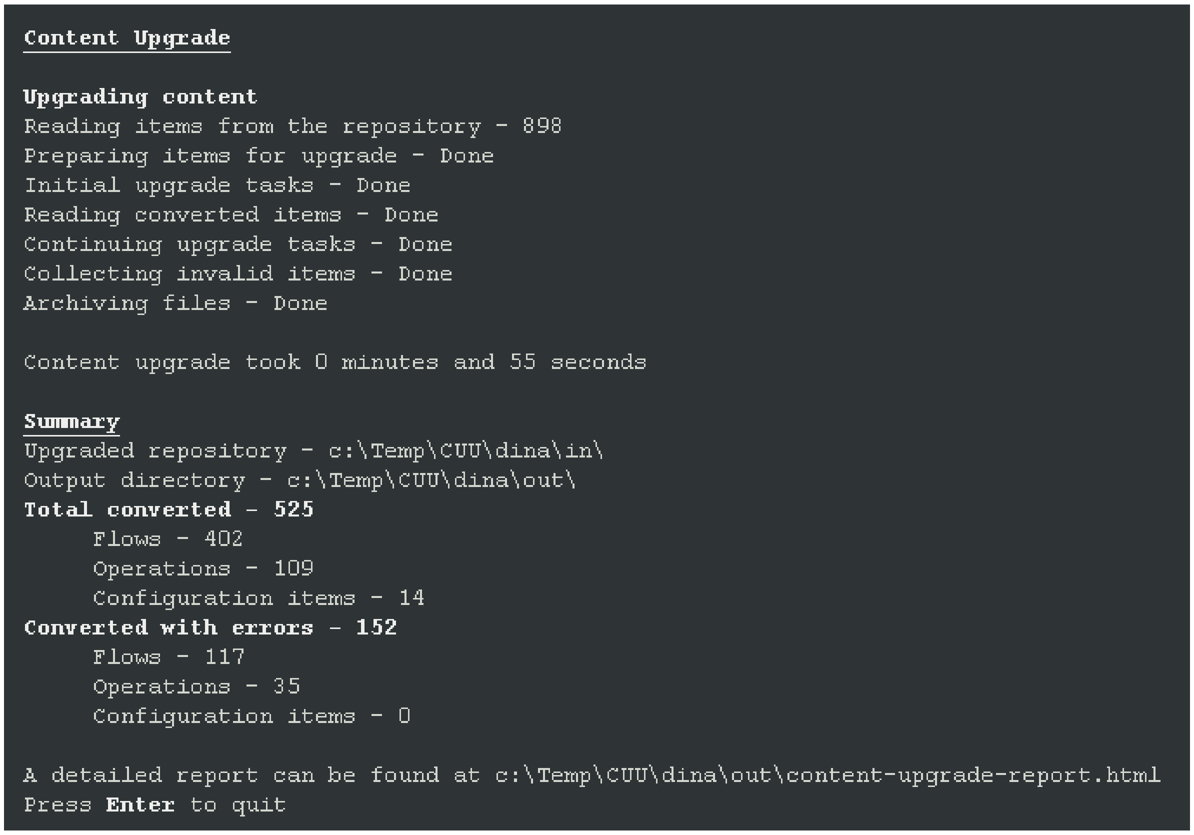
An example of upgrade options in Linux:

```
>./upgrade-content --name AcmeContent --publisher Acme --repo
/old/OldRepositories --output /NewRepositories
```

In the example:

- `AcmeContent` is the name given to the new content pack
- `Acme` is the name of your company
- `C:\old\OldRepositories` (Windows) or `/old/oldRepositories` (Linux) is the location of the 9.x repository
- `D:\NewRepositories` (Windows) or `/newRepositories` (Linux), is the location where the content pack will be stored.

The Content Upgrade Utility console displays the progress of the upgrade.



```
Content Upgrade

Upgrading content
Reading items from the repository - 898
Preparing items for upgrade - Done
Initial upgrade tasks - Done
Reading converted items - Done
Continuing upgrade tasks - Done
Collecting invalid items - Done
Archiving files - Done

Content upgrade took 0 minutes and 55 seconds

Summary
Upgraded repository - c:\Temp\CUU\dina\in\
Output directory - c:\Temp\CUU\dina\out\
Total converted - 525
     Flows - 402
     Operations - 109
     Configuration items - 14
Converted with errors - 152
     Flows - 117
     Operations - 35
     Configuration items - 0

A detailed report can be found at c:\Temp\CUU\dina\out\content-upgrade-report.html
Press Enter to quit
```

Note: If you provide a wrong (non-existent) folder in the `--repo` parameter, the Content Upgrade Utility displays an error message.

When the upgrade is complete, a 10.x user content pack and Studio project is generated and stored in the output location that you specified.

Note: Make sure not to delete the output files (user content pack and Studio project) during the upgrade procedure. This output is required by Studio.

In our example, all the content that can be deployed and run is stored in a Studio project called **AcmeContent**. The **AcmeContent-cp.jar** file is created as an archive in the **AcmeContent** directory. Any invalid content is stored in a Studio project called **AcmeContent-invalid**. Both the **AcmeContent** and **AcmeContent-invalid** projects can be opened and edited in Studio.

3. If there are multiple repositories, repeat the upgrade process for each repository (shared or local). However, we recommend consolidating all content into one repository, to be used for the upgrade.
4. After the content upgrade has run, your 9.x content is split into two projects: the valid content, which is ready to use, and the invalid content, which you need to fix manually in 10.x Studio.

   You can view the results of the upgrade in the following:

   - The **content-upgrade.log** file in the **output** folder. This file includes everything that was written to the console as well as debug messages.

   - The **content-upgrade-report.html** file in the **Output \ Upgrade** folder. See Check the Upgrade Report for Dependencies and Invalid Content.

- The two 10.x projects that were generated and stored in the output location that you specified:
    - **Valid**: All the content that can be deployed and run is stored in a project with the valid content. If you have defined a name for the upgraded repository output using the `--name` argument, the valid content is stored in a project of that name.

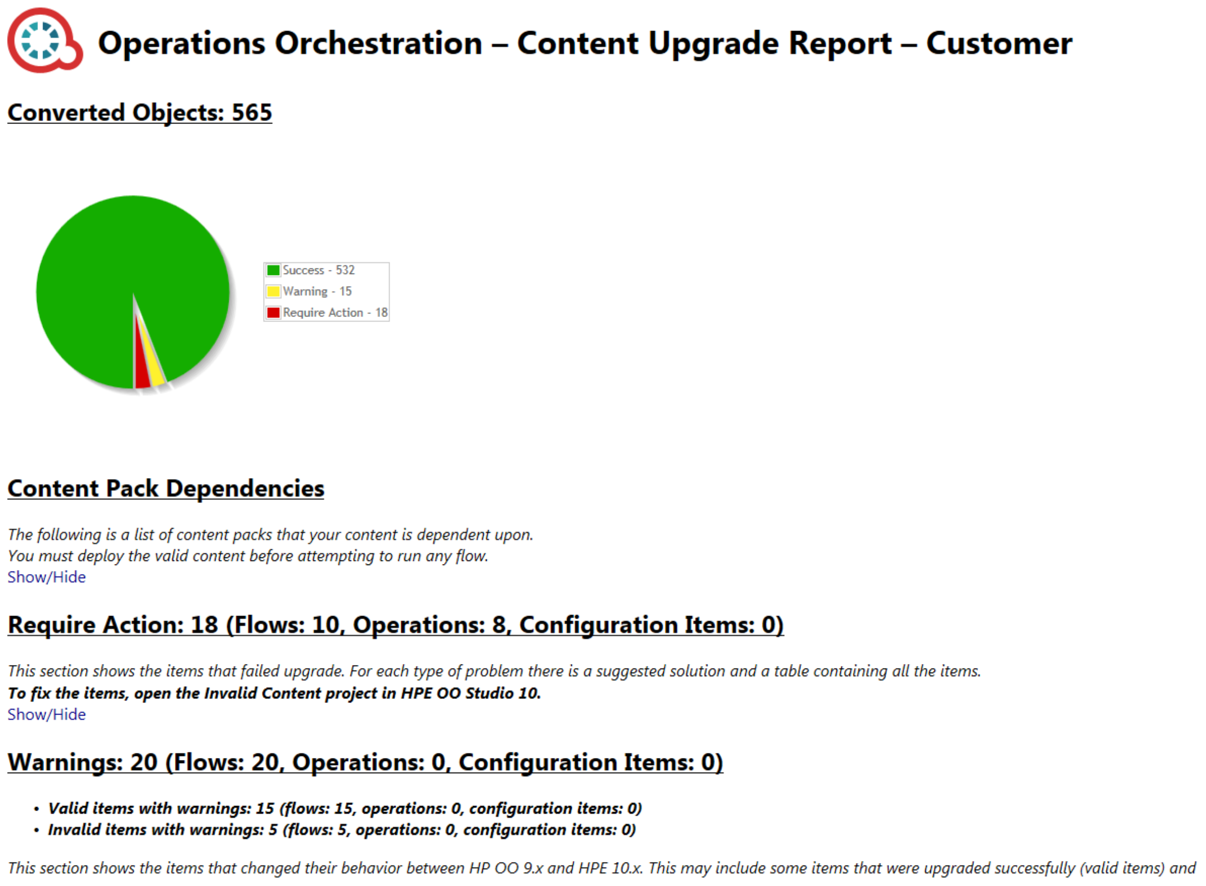
      For example, if the name is defined as AcmeContent, the valid content is stored in a directory called **AcmeContent**. The **AcmeContent-cp.jar** file is created as an archive of that directory.
    - **Invalid**: Content that cannot be deployed and run is stored in a project for invalid content. If you have defined a name for the upgraded repository output using the `--name` argument, the project containing the invalid content is called **<Name>-invalid**.

      If you have not defined a name using the `--name` argument, the project containing the invalid content is called **Customer-invalid**.

Back to the flowchart

Check the Upgrade Report for Dependencies and Invalid Content

After the project has been upgraded, you should check the Content Upgrade report to identify the content that was not upgraded successfully.



Operations Orchestration – Content Upgrade Report – Customer

Converted Objects: 565

🟩 Success - 532
🟨 Warning - 15
🟥 Require Action - 18

Content Pack Dependencies

The following is a list of content packs that your content is dependent upon.
You must deploy the valid content before attempting to run any flow.
Show/Hide

Require Action: 18 (Flows: 10, Operations: 8, Configuration Items: 0)

This section shows the items that failed upgrade. For each type of problem there is a suggested solution and a table containing all the items.
To fix the items, open the Invalid Content project in HPE OO Studio 10.
Show/Hide

Warnings: 20 (Flows: 20, Operations: 0, Configuration Items: 0)

- ***Valid items with warnings: 15 (flows: 15, operations: 0, configuration items: 0)***
- ***Invalid items with warnings: 5 (flows: 5, operations: 0, configuration items: 0)***

This section shows the items that changed their behavior between HP OO 9.x and HPE 10.x. This may include some items that were upgraded successfully (valid items) and

The report shows:

- The total number of converted objects.
- **Content Pack Dependencies** - the content packs that the content is dependent upon. These dependencies are listed separately for the valid project and the invalid project.

- **Success**: The number of objects that were upgraded with no warnings.
- **Require Action**: Objects that failed to be upgraded. These are stored in the **Invalid** project and you will need to fix them manually in Studio, before they can be used in 10.x.

  Note: Some objects that failed upgrade also have warnings. These will appear in both the **Requires Action** sections and the **Warning** section.

- **Warning**: Objects with one or more warnings. This may include some items that were upgraded successfully (valid items) and others that failed to be upgraded (and also appear in the **Require Action** section). Items with warnings may behave differently in 10.x, so it is recommended to check the new way that they work, and to see if there are corrections that need to be made.

  Note: Note that the **Warning** category in the pie chart only counts the valid items with warnings. This is because the invalid ones are already counted as **Require Action**.

- The **Changes in Out-of-the-box Configuration Item** section shows the configuration items that have updated names or values. These will appear if you modified the out-of-the-box configuration items and didn't use the **--nine-cp-version** option.

To review the upgraded content:

1. After the upgrade, locate the upgrade report **Output\content-upgrade-report.html**, and open it in a Web browser.
2. In the **Require Action** and **Warnings** sections, click **Show/Hide** to see tables containing details of the items and suggestions about how to resolve each type of problem.

   Items are grouped into separate tables according to the type of problem, and sorted according to path. For examples, see below.

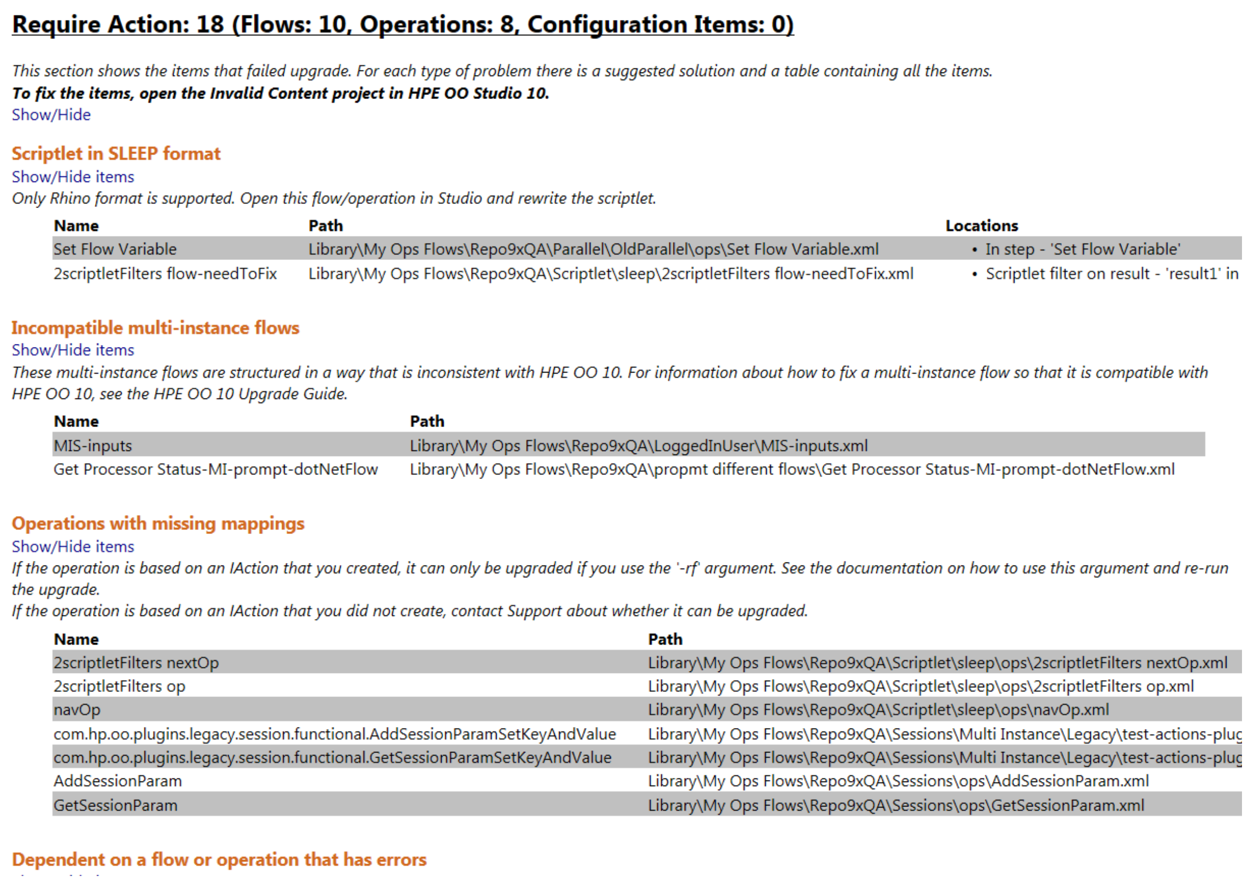
   Note: Flows that have more than one error (for example, both a Sleep scriptlet and dependency on an invalid item) will appear in more than one table. Flows with a general error or schema error will only appear once.

3. Click **Show/Hide** to toggle between showing and hiding individual tables.

Back to the flowchart

Contents of the Report - Examples

Require Action - Examples



Require Action: 18 (Flows: 10, Operations: 8, Configuration Items: 0)

This section shows the items that failed upgrade. For each type of problem there is a suggested solution and a table containing all the items.
To fix the items, open the Invalid Content project in HPE OO Studio 10.
Show/Hide

Scriptlet in SLEEP format
Show/Hide items
Only Rhino format is supported. Open this flow/operation in Studio and rewrite the scriptlet.

Name	Path	Locations
Set Flow Variable	Library\My Ops Flows\Repo9xQA\Parallel\OldParallel\ops\Set Flow Variable.xml	• In step - 'Set Flow Variable'
2scriptletFilters flow-needToFix	Library\My Ops Flows\Repo9xQA\Scriptlet\sleep\2scriptletFilters flow-needToFix.xml	• Scriptlet filter on result - 'result1' in

Incompatible multi-instance flows
Show/Hide items
These multi-instance flows are structured in a way that is inconsistent with HPE OO 10. For information about how to fix a multi-instance flow so that it is compatible with HPE OO 10, see the HPE OO 10 Upgrade Guide.

Name	Path
MIS-inputs	Library\My Ops Flows\Repo9xQA\LoggedInUser\MIS-inputs.xml
Get Processor Status-MI-prompt-dotNetFlow	Library\My Ops Flows\Repo9xQA\propmt different flows\Get Processor Status-MI-prompt-dotNetFlow.xml

Operations with missing mappings
Show/Hide items
If the operation is based on an IAction that you created, it can only be upgraded if you use the '-rf' argument. See the documentation on how to use this argument and re-run the upgrade.
If the operation is based on an IAction that you did not create, contact Support about whether it can be upgraded.

Name	Path
2scriptletFilters nextOp	Library\My Ops Flows\Repo9xQA\Scriptlet\sleep\ops\2scriptletFilters nextOp.xml
2scriptletFilters op	Library\My Ops Flows\Repo9xQA\Scriptlet\sleep\ops\2scriptletFilters op.xml
navOp	Library\My Ops Flows\Repo9xQA\Scriptlet\sleep\ops\navOp.xml
com.hp.oo.plugins.legacy.session.functional.AddSessionParamSetKeyAndValue	Library\My Ops Flows\Repo9xQA\Sessions\Multi Instance\Legacy\test-actions-pluç
com.hp.oo.plugins.legacy.session.functional.GetSessionParamSetKeyAndValue	Library\My Ops Flows\Repo9xQA\Sessions\Multi Instance\Legacy\test-actions-pluç
AddSessionParam	Library\My Ops Flows\Repo9xQA\Sessions\ops\AddSessionParam.xml
GetSessionParam	Library\My Ops Flows\Repo9xQA\Sessions\ops\GetSessionParam.xml

Dependent on a flow or operation that has errors

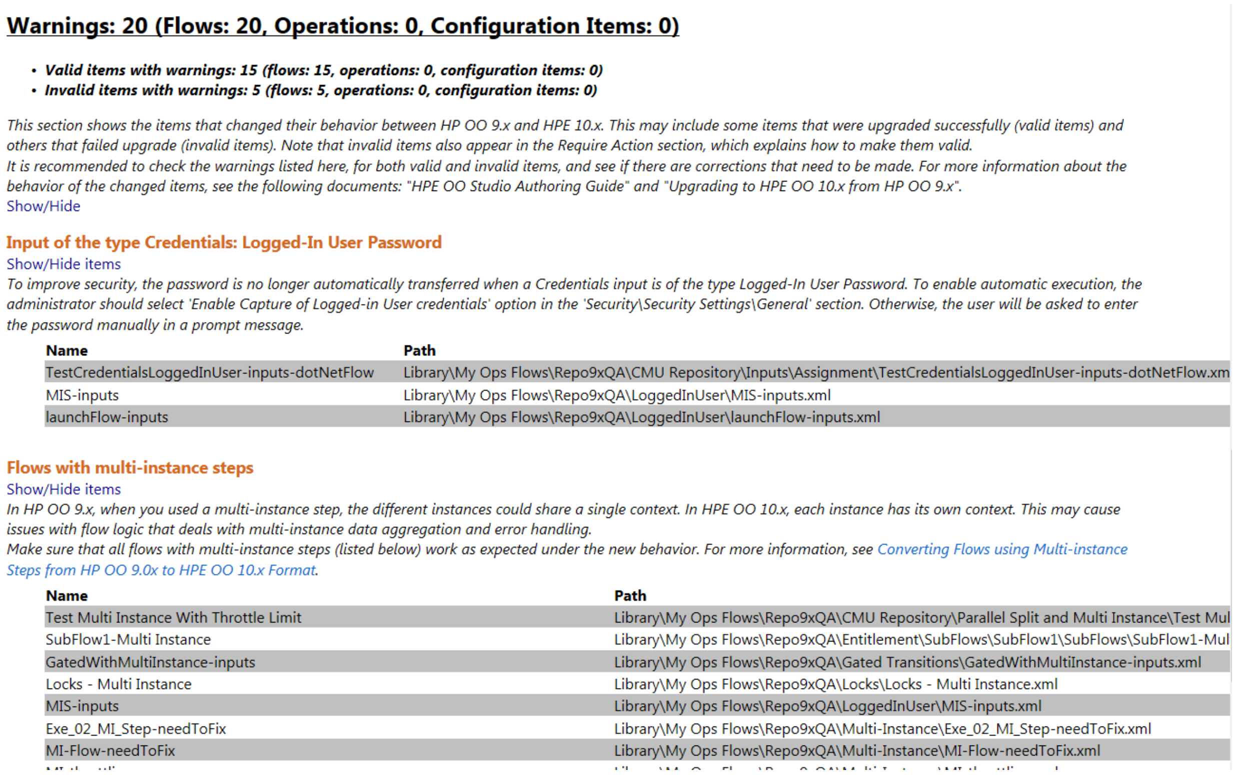
The **Require Action** section lists the objects that were not upgraded and suggests action items for fixing them. For example:

- Flows/operations that reference a missing item (flow, operation, selection list, domain term, system filter, scriptlet or RAS). You will need to fix the references in Studio so that they refer to existing items, and then upgrade the repository again.
- Flows/operations containing scriptlets in SLEEP format. You will need to open the flow/operation in Studio and rewrite the scriptlet in Rhino format.
- Flows with a multi-instance step inside a parallel or multi-instance lane. You will need to manually convert the step to a subflow with a multi-instance step.
- Multi-instance flows that are structured in a way that is inconsistent with 10.x. You will need to fix the multi-instance flow so that it is compatible with HP. For more information about multi-instance flows in 10.x, see "Differences in Behavior After Upgrading" on page 53.
- Operations with missing mappings for IActions. If an operation is missing a mapping, this means it's one of the following:
  - Custom IAction, in which case, you need to provide the jar and upgrade the repository again using the '-rf' argument. For more information, see Upgrade the Content Using the Content Upgrade Utility.
  - Copy of an IAction that is no longer supported, in which case it cannot be used anymore.
- Flows/operations that reference unsupported flows or operations. Some flows and operations no longer work in 10.x due to implementation changes. Among these are some scheduling and repository operations.

- Flows/operations that are dependent on a flow or operation that has errors. If you fix the flows and operations that these are dependent upon, these will also be fixed.
- For name changes - the item's UUID is replaced and all user content is modified to reference the new UUID.
    - For value changes - the item's UUID is replaced, and its name is added the 'custom_' prefix so as not to collide with the out-of-the-box item. All user content is modified to reference the new UUID.

  You may want to update these configuration items.

Warnings - Examples



Warnings: 20 (Flows: 20, Operations: 0, Configuration Items: 0)

- ***Valid items with warnings: 15 (flows: 15, operations: 0, configuration items: 0)***
- ***Invalid items with warnings: 5 (flows: 5, operations: 0, configuration items: 0)***

This section shows the items that changed their behavior between HP OO 9.x and HPE 10.x. This may include some items that were upgraded successfully (valid items) and others that failed upgrade (invalid items). Note that invalid items also appear in the Require Action section, which explains how to make them valid.
It is recommended to check the warnings listed here, for both valid and invalid items, and see if there are corrections that need to be made. For more information about the behavior of the changed items, see the following documents: "HPE OO Studio Authoring Guide" and "Upgrading to HPE OO 10.x from HP OO 9.x".
Show/Hide

Input of the type Credentials: Logged-In User Password
Show/Hide items
To improve security, the password is no longer automatically transferred when a Credentials input is of the type Logged-In User Password. To enable automatic execution, the administrator should select 'Enable Capture of Logged-in User credentials' option in the 'Security\Security Settings\General' section. Otherwise, the user will be asked to enter the password manually in a prompt message.

Name	Path
TestCredentialsLoggedInUser-inputs-dotNetFlow	Library\My Ops Flows\Repo9xQA\CMU Repository\Inputs\Assignment\TestCredentialsLoggedInUser-inputs-dotNetFlow.xm
MIS-inputs	Library\My Ops Flows\Repo9xQA\LoggedInUser\MIS-inputs.xml
launchFlow-inputs	Library\My Ops Flows\Repo9xQA\LoggedInUser\launchFlow-inputs.xml

Flows with multi-instance steps
Show/Hide items
In HP OO 9.x, when you used a multi-instance step, the different instances could share a single context. In HPE OO 10.x, each instance has its own context. This may cause issues with flow logic that deals with multi-instance data aggregation and error handling.
Make sure that all flows with multi-instance steps (listed below) work as expected under the new behavior. For more information, see Converting Flows using Multi-instance Steps from HP OO 9.0x to HPE OO 10.x Format.

Name	Path
Test Multi Instance With Throttle Limit	Library\My Ops Flows\Repo9xQA\CMU Repository\Parallel Split and Multi Instance\Test Mul
SubFlow1-Multi Instance	Library\My Ops Flows\Repo9xQA\Entitlement\SubFlows\SubFlow1\SubFlows\SubFlow1-Mul
GatedWithMultiInstance-inputs	Library\My Ops Flows\Repo9xQA\Gated Transitions\GatedWithMultiInstance-inputs.xml
Locks - Multi Instance	Library\My Ops Flows\Repo9xQA\Locks\Locks - Multi Instance.xml
MIS-inputs	Library\My Ops Flows\Repo9xQA\LoggedInUser\MIS-inputs.xml
Exe_02_MI_Step-needToFix	Library\My Ops Flows\Repo9xQA\Multi-Instance\Exe_02_MI_Step-needToFix.xml
MI-Flow-needToFix	Library\My Ops Flows\Repo9xQA\Multi-Instance\MI-Flow-needToFix.xml
[illegible]	[illegible]

The **Warnings** section lists both objects that were not upgraded and those that were upgraded, but may behave differently after the upgrade. For example:

- Flows/operations with multi-instance steps, which are still valid but which may work differently in 10.x.

  For example, in 9.x, when you used a multi-instance step, the different instances could share a single context. In 10.x, each instance has its own context. This may cause issues with flow logic that deals with multi-instance data aggregation and error handling.
- Operations that are copies of HPE OO integration operations, which have new inputs. The new inputs for these operations, and any flow using them, were added automatically. However, flows using them may now prompt for input values during execution.
- Flows/operations with input of the type **Credentials**: **Logged-In User Password**. To improve security, the password is no longer automatically transferred when a **Credentials** input is of the type **Logged-In User Password**. Instead, the user will be asked to enter the password manually in a prompt message.

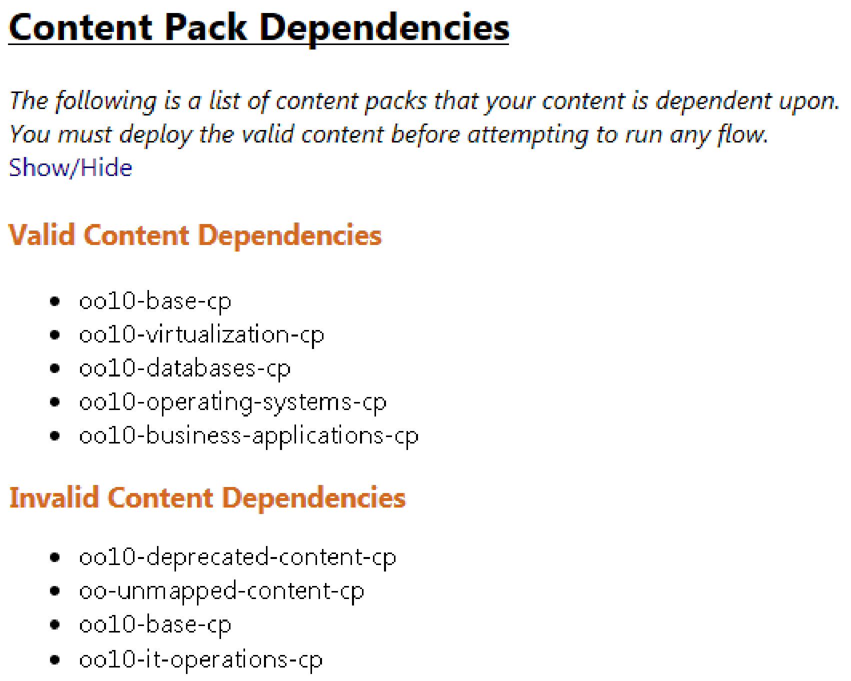
  > **Note:** You can change the prompt behavior.

- Flows/operations with unhandled usage of 'Store system account in flow variable'. The implementation of this operation has changed and not all cases can be handled automatically.

  If the **flowVariable** input was not assigned a constant value, this has to be fixed manually. You need to fix all the inputs that use this flow variable. For example, if the system account is assigned to a flow variable named **sysAcct**, then an input assigned from that variable that is supposed to accept the **Username** part of **sysAcct** should now be assigned from **sysAcctUsername**.
- Flows/operations with multi-instance steps, which are still valid but which may work differently in 10.x. For example, in 9.x, when you used a multi-instance step, the different instances could share a single context. In 10.x, each instance has its own context. This may cause issues with flow logic that deals with multi-instance data aggregation and error handling.

For more information about objects that behave differently after upgrade, see "Differences in Behavior After Upgrading" on page 53.

Dependencies - Example


Content Pack Dependencies

The following is a list of content packs that your content is dependent upon.
You must deploy the valid content before attempting to run any flow.
Show/Hide

Valid Content Dependencies

- oo10-base-cp
- oo10-virtualization-cp
- oo10-databases-cp
- oo10-operating-systems-cp
- oo10-business-applications-cp

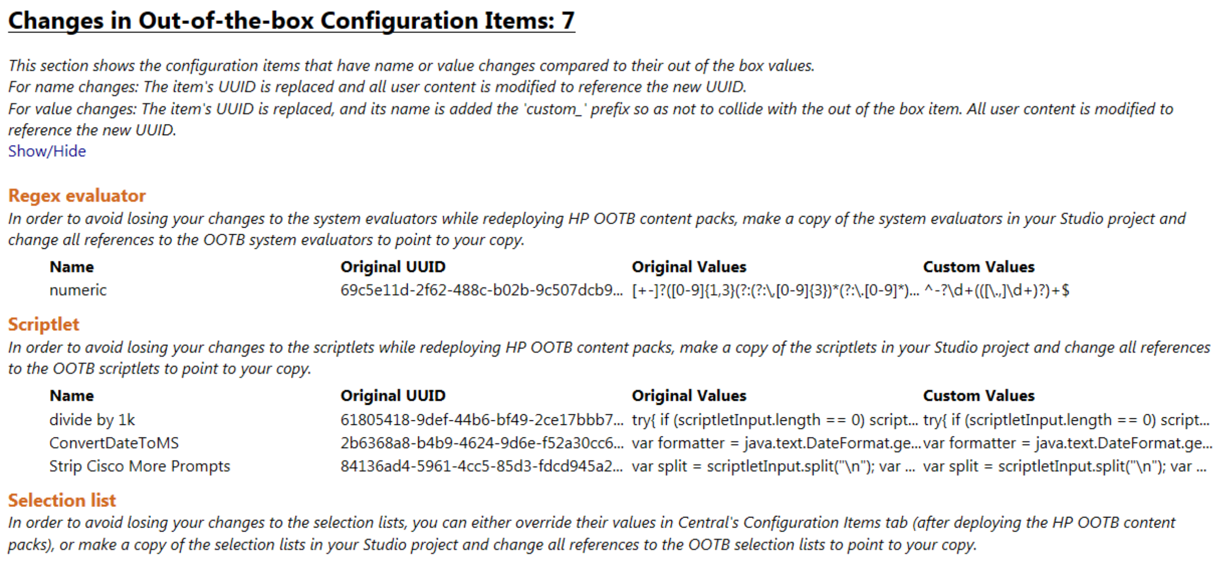
Invalid Content Dependencies

- oo10-deprecated-content-cp
- oo-unmapped-content-cp
- oo10-base-cp
- oo10-it-operations-cp

After the content upgrade has run, your 9.x content is split into two projects: the valid content, which is ready to use, and the invalid content, which you need to fix manually in 10.x Studio.

The **Content Pack Dependencies** part of the report displays the content pack dependencies for the valid project and the invalid project, in two separate lists.

Changes in out-of-the-box configuration items - Examples



Changes in Out-of-the-box Configuration Items: 7

This section shows the configuration items that have name or value changes compared to their out of the box values.
For name changes: The item's UUID is replaced and all user content is modified to reference the new UUID.
For value changes: The item's UUID is replaced, and its name is added the 'custom_' prefix so as not to collide with the out of the box item. All user content is modified to reference the new UUID.
Show/Hide

Regex evaluator
In order to avoid losing your changes to the system evaluators while redeploying HP OOTB content packs, make a copy of the system evaluators in your Studio project and change all references to the OOTB system evaluators to point to your copy.

Name	Original UUID	Original Values	Custom Values
numeric	69c5e11d-2f62-488c-b02b-9c507dcb9...	[+-]?([0-9]{1,3}(?:(?:\,[0-9]{3})*(?:\.[0-9]*)...	^-?\d+(([\.,]\d+)?)+$

Scriptlet
In order to avoid losing your changes to the scriptlets while redeploying HP OOTB content packs, make a copy of the scriptlets in your Studio project and change all references to the OOTB scriptlets to point to your copy.

Name	Original UUID	Original Values	Custom Values
divide by 1k	61805418-9def-44b6-bf49-2ce17bbb7...	try{ if (scriptletInput.length == 0) script...	try{ if (scriptletInput.length == 0) script...
ConvertDateToMS	2b6368a8-b4b9-4624-9d6e-f52a30cc6...	var formatter = java.text.DateFormat.ge...	var formatter = java.text.DateFormat.ge...
Strip Cisco More Prompts	84136ad4-5961-4cc5-85d3-fdcd945a2...	var split = scriptletInput.split("\n"); var ...	var split = scriptletInput.split("\n"); var ...

Selection list
In order to avoid losing your changes to the selection lists, you can either override their values in Central's Configuration Items tab (after deploying the HP OOTB content packs), or make a copy of the selection lists in your Studio project and change all references to the OOTB selection lists to point to your copy.

The **Changes in out-of-the-box configuration items** lists configuration items that have updated names or values. These will appear if you modified the out-of-the-box configuration items and didn't use the **--nine-cp-version** argument. For example:

- Scriptlet evaluators that were given new names
- Selection lists that were given new values

Fix Invalid Content in Studio and Merge it with Valid Content

1. In Studio 10.x, import the 10.x out-of-the-box Base Content Pack and all content packs that are mentioned in the report.
2. Open both the **Invalid** and **Valid** content packs in Studio 10.x.
3. In Studio, fix any content that needs fixing. Use the Upgrade report to see what needs to be fixed in the invalid project.

   In some cases, the content that was not upgraded is content that is not yet supported by 10.x, and you may prefer to keep it as it, while waiting for the features to be supported in later versions.

   In other cases, you may prefer to modify the content so that it can be used in 10.x. You may want to save a copy of the original content and create a duplicate that is modified.

   Note: For information about how to fix specific upgrade problems, see "Known Issues and Troubleshooting" on page 68.

4. We recommend dragging each fixed item from the **Invalid** project to the **Valid** project, after it is fixed. This way, you can more easily identify what is left to fix.
5. In addition to fixing the items in the **Invalid** project, you should test the valid flows, because some flows might behave differently. See "Differences in Behavior After Upgrading" on page 53.

Note: After content upgrade, the valid project and the resulting content pack will have duplicated configuration items with the same UUIDs. Therefore, you will not be able to import the project and the content pack at the same time into the Studio workspace.

Back to the flowchart

Create a Content Pack form the Merged Content

After making any required adjustments to the content in Studio 10.x, export the fixed content in the **Valid** project as a content pack, for deployment.

For more information about creating content packs, see the *HPE OO Studio Authoring Guide*.

Back to the flowchart

Deploy the Upgraded Content and Dependencies to Central

1. In Central, deploy the out-of-the-box Base Content Pack that is shipped with 10.x, and deploy all the content packs that your content is dependent upon to run.

   Note: Do not deploy the upgraded content pack until after you have deployed these content packs.

2. If you have partner content that was excluded from upgrade, deploy the partner content pack first, before deploying the upgraded content pack.
3. Deploy the content pack containing the upgraded content.

   Important! Do not directly deploy the content pack that was created by the Content Upgrade Utility. Make sure to use the content pack that you created in Studio, from the **Valid** project.

For more information about deploying content in Central, see the *HPE OO Central User Guide*.

Back to the flowchart

Verify Users, LDAP, Roles, and Enable Authentication in Central

1. In Central, check the users, roles, and LDAP configurations in the deployed, upgraded content.
2. Enable authentication in Central and verify that the users, roles, and LDAP configurations work as expected

For more information, see "Setting Up Security – Roles", "Setting Up Security – LDAP Authentication", the *HPE OO Central User Guide*.

Step 4: Verify and Synchronize

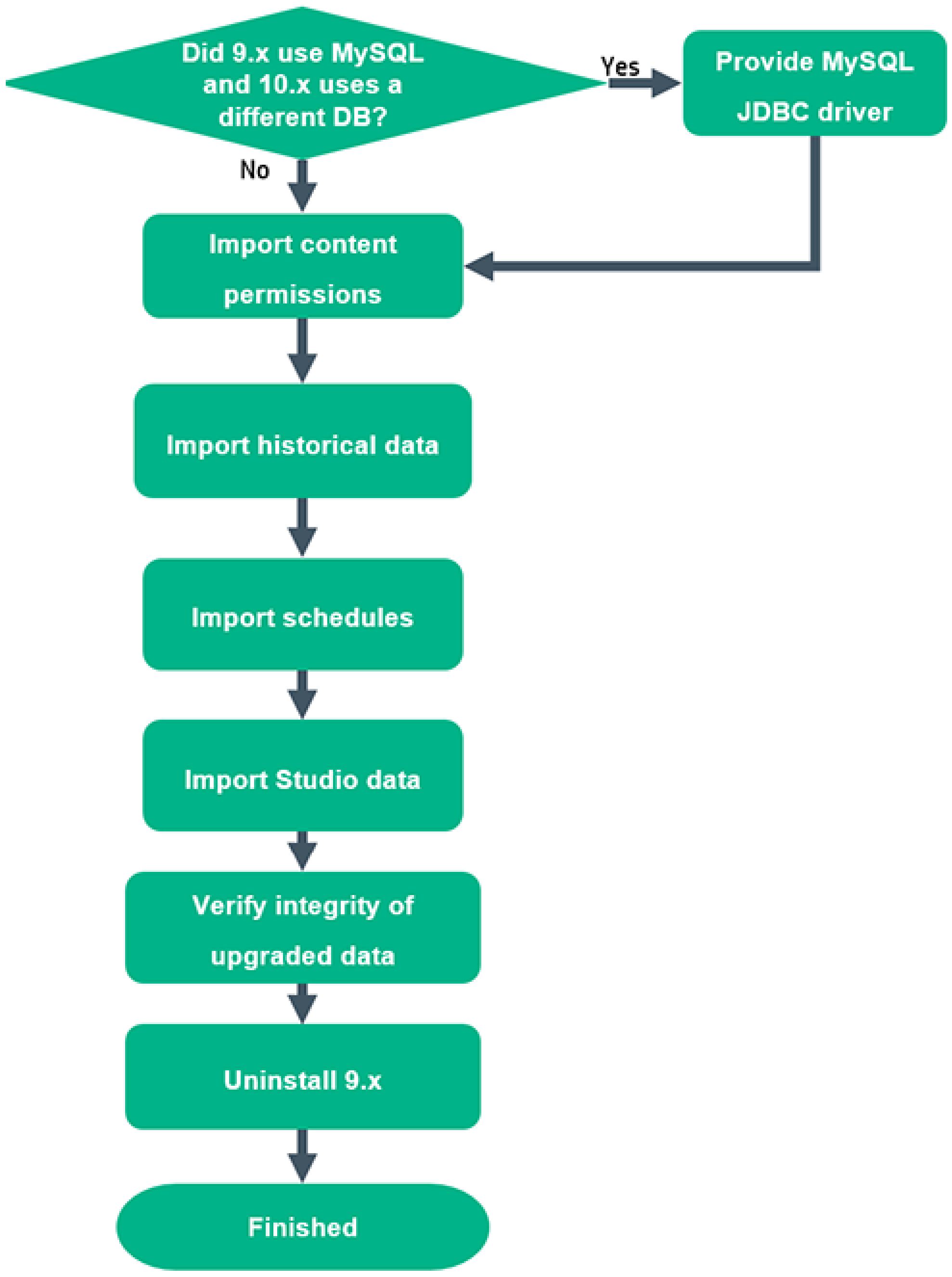
The final steps are to synchronize the content-related data from 9.x, which was not upgraded automatically, and to verify the integrity of the upgraded data before uninstalling 9.x.

Synchronization is performed only for deployed content. This means that if you have divided your content into several content packs, you will need to re-synchronize after each deployment. Alternatively, deploy all of them and only then synchronize.

You can perform the synchronizations either via REST API or using the Operations Orchestration Shell (OOSH) utility; however, we recommend using the OOSH utility.

The OOSH utility is located under **<installation path>\central\bin** and **<installation path>\ras\bin**. For more information, see the *HPE OO Shell (OOSH) User Guide*.

Note: Users who invoke upgrade-related commands from the OOSH utility must have the corresponding permissions.


Did 9.x use MySQL and 10.x uses a different DB?
Yes
Provide MySQL JDBC driver
No
Import content permissions
Import historical data
Import schedules
Import Studio data
Verify integrity of upgraded data
Uninstall 9.x
Finished

Links to Steps

Did 9.x use MySQL and 10.x uses a different database? • **Yes**: Provide the JDBC Driver • **No**: Go to **Import Content Permissions**
Import Content Permissions
Import Historical Data
Import Schedules
Import Studio Data
Verify the Integrity of Upgraded Data
Uninstall 9.x
Finished

Provide the JDBC Driver

If 9.x used a MySQL database, but 10.x uses a different database type, you must do the following before proceeding:

1. Copy the MySQL JDBC driver and paste it into the following two locations:

   <installation path>/central/lib

   <installation path>/central/tomcat/lib
2. Restart Central.
3. Repeat the steps above for each Central installation in your setup.

Back to the flowchart

Import Content Permissions

The content upgrade procedure upgrades the following content permissions for flows and folders:

9.x	10.x:
Read	**View**
Execute	**Run**
Write	Not upgraded.
LinkTo	Not upgraded.

Note: The installation procedure does not extract and load content permissions for invalid flows.

Content permissions relating to operations are not upgraded. This remaining data needs to be synchronized.

During the upgrade, a file named **permissions.csv** is created under the **Output** folder. This file contains the read and run permissions of flows and folders from the 9.x repository.

You can synchronize the permissions using the Operations Orchestration Shell (OOSH) utility.

1. Double-click the **oosh.bat** file, to start the OOSH utility.

   The **oosh.bat** batch file is located under **<installation path>\central\bin** and **<installation path>\ras\bin**.

   > **Note:** The user who invokes upgrade-related commands from OOSH must have the corresponding permissions.

2. In the command line, use the following commands:
   - `delete-permissions-file` – this deletes the permissions file from Central, enabling you to upload a new file.
   - `upload-permissions-file` – this uploads a permissions file from which permissions can be imported. Using this command, you can supply the permission file that was created during upgrade.
   - `import-permissions` – this imports the permissions from the file. Only permissions of content that is already deployed are imported.
   - `permissions-status` – this displays the imported permissions either on screen or written to a file.

For example:

1. In OOSH, use the command `upload-permissions-file`.
2. Deploy a content pack to Central, with the flow f1.
3. In OOSH, use the command `import-permissions`.

   The permissions from f1 are imported.
4. Deploy a content pack to Central, with the flow f2.
5. In OOSH, use the command `import-permissions` again.

The synchronization is executed only on items that were not yet synchronized (for the current upload).

The synchronization overrides previous content permissions if any exist.

If you added new items to an upgraded content pack, after redeploying, if you synchronize again, it will synchronize only the new items that were added.

For more information about OOSH, see the *Operations Orchestration Shell (OOSH) User Guide*.

> **Note:** The **permissions.csv** file created by the Content Upgrade Utility takes into account the flows' paths, to match permissions to flows. If, after upgrade, you split the content into different content packs (thus, changing the flows' paths), the **permission.csv** file will not recognize those new paths, and will not be able to match the permissions.

Back to the flowchart

Import Historical Data

In OOSH, use the following commands to import historical data:

- `historical-data-start-date` – this sets the start date from which synchronization will begin in the format “dd/MM/yyyy”, for example, “13/12/2014”.
- `import-historical-data` – this imports historical data for deployed content.

> **Note:** This process runs in the background, so the fact that this command is finished does not mean that the import has finished.

- `historical-data-status` – this displays the import status.

For example:

1. In OOSH, use the command `historical-data-start-date`.
2. Deploy a content pack.
3. In OOSH, use the command `import-historical-data`.
4. In OOSH, use the command `historical-data-status`.

   This command may take several minutes. Check the status until you see that everything is done.
5. Repeat steps 2-4.

Back to the flowchart

Import Schedules

The installation procedure does not extract and load the scheduling configuration, and it does not extract and load flows with overdue schedule time.

The installation procedure disables all scheduled flows in 10.x.

It is your responsibility to disable the scheduled flows in 9.x and enable those scheduled flows in 10.x.

In OOSH, use the command `import-schedules` to import schedules for content that is deployed.

Back to the flowchart

Import Studio Data

Studio permission data is not upgraded during the installation of 10.x. This includes:

- Local settings (user preferences—colors, icons, and so on).
- Copy of the **rc_keystore** for remote repository connection or other authentication data.
- **Studio.properties** configuration. This configuration is not relevant in 10.x, because source control permissions are managed in the source control management application and not in Studio.

After the upgrade, you need to set up the Studio permissions in the source control management application.

For more information, see the *HPE OO Studio Authoring Guide*.

Back to the flowchart

Verify the Integrity of Upgraded Data

Before uninstalling 9.x, check the integrity of all your upgraded data. Check that you have all the runs, schedules, and so on that you need.

Back to the flowchart

Uninstall 9.x

When you are satisfied with the integrity of the upgraded data, you can uninstall 9.x.

For more information, see the relevant *9.x Installation Guide*.

Back to the flowchart

Finished

The upgrade is now complete!

Troubleshooting and Different Behavior

This section includes migration issues and solutions, and descriptions of how your flows may appear and behave differently after upgrading.

Differences in Behavior After Upgrading

The following are not issues, but highlight ways that your flows may appear and behave differently after upgrading.

Changes in Throttling

There is a difference in the way that throttling works in OO 10.x.

- 9.x would keep a consistent number of concurrent threads running. For example, if you had 100 items and you throttled at 10, the multi-instance step would keep 10 going at all times. If one finished, it would start one more.
- 10.x starts 10 of the 100, waits until all 10 are finished, and then starts another 10.

Please take this into account when you schedule your patching.

Changes in Behavior for Empty and Encrypted Values

In previous versions, when you use `assign-from`, if the variable is empty, the flow will behave as if the variable does not exist. However, if the variable is empty and encrypted, the input on which the `assign-from` is used will be overriden with an encrypted empty value.

The empty encrypted variable remains empty even if it is used in a sub-flow with: `assign-from: variable, otherwise: any non-empty value, assign-to: variable`.

In 10.x, the inputs are obfuscated, rather than encrypted. The flow will not initialize the empty obfuscated variable. If it uses the values described above, in the end, the variable will have the value that was initialized in the sub-flow.

Multi-instances and Parallel Split Steps Cannot be Non-blocking

In 10.x, it is not possible for multi-instance or parallel split steps to be non-blocking. If a flow contained non-blocking multi-instance or parallel split steps, after content upgrade, these steps will no longer be non-blocking.

Deprecated Step Results Converted into Proper Results

If a flow had deprecated step results (under the **Advanced** tab in the Step Inspector), after upgrade, they now appear under the **Results** tab.

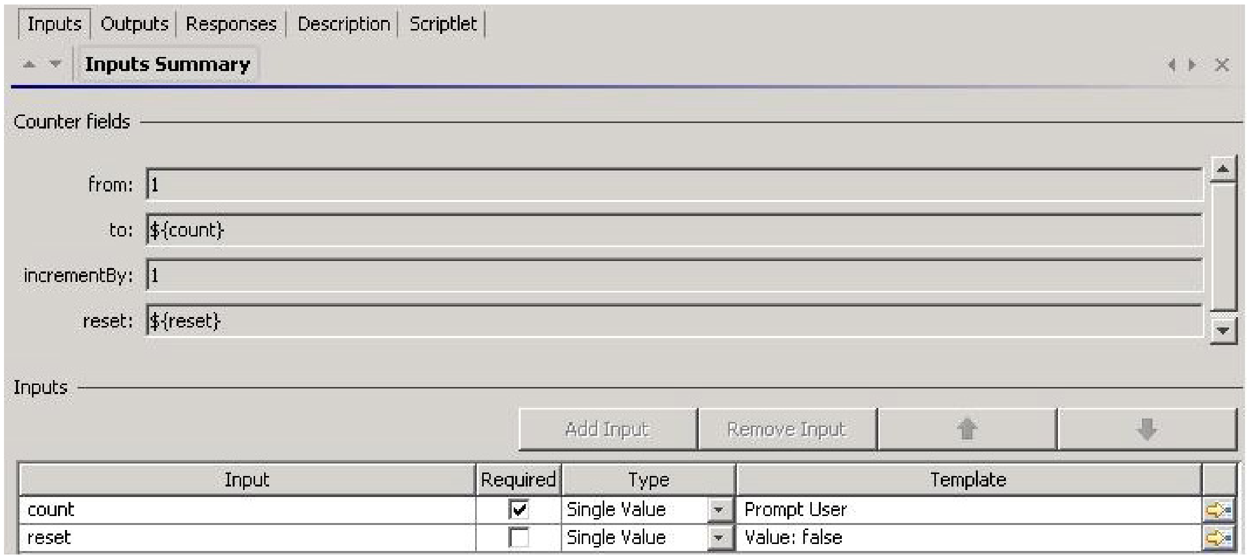
Be aware that these results are now located in a different place.

One Interface for all Operations

In previous versions, there were several types of operations with different interfaces. After upgrade, they all have the same interface. In 10.x, there is a only one type of operation.

Field Values Converted to Regular Inputs

In previous versions, some operations included field values. For example, the image below shows a loop operation in 9.x, where there are two inputs, **count** and **reset**, and fields to specify the starting count and increment.



Counter fields

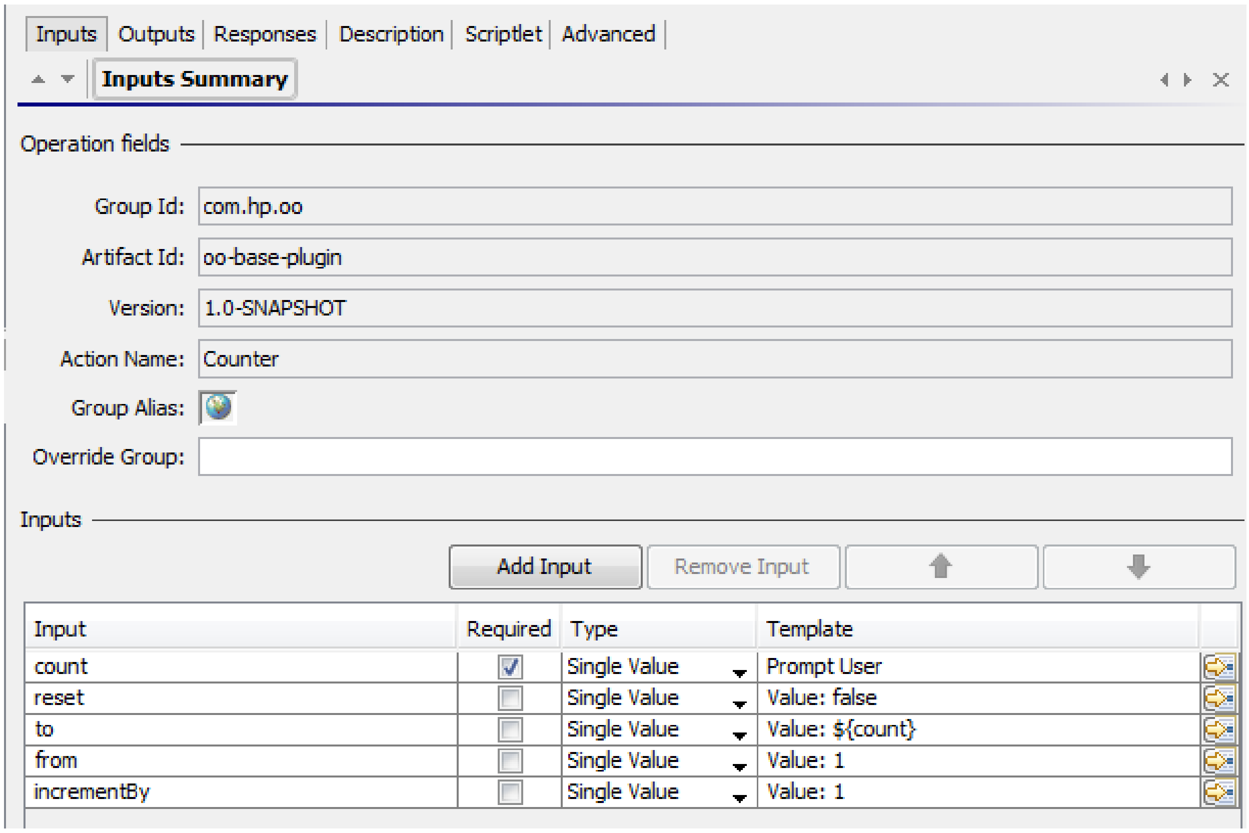
from:	1
to:	${count}
incrementBy:	1
reset:	${reset}

Inputs

Input	Required	Type	Template
count	✓	Single Value	Prompt User
reset		Single Value	Value: false

Field values were removed in 10.x. So, after upgrade, these field values are converted to regular inputs. In the example shown, the field values have been removed and replaced with inputs to control the starting count and increment.

Note: If the original field names included spaces, these spaces are converted to underscores in the input names after upgrade. For example, **Field 1** is changed to **Field_1**.


Inputs | Outputs | Responses | Description | Scriptlet | Advanced
Inputs Summary
Add Input
Remove Input

Operation fields

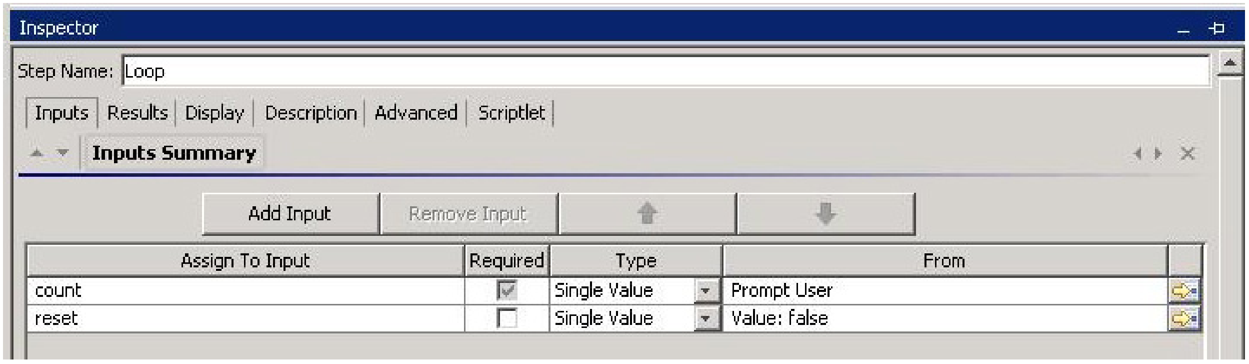
Group Id:	com.hp.oo
Artifact Id:	oo-base-plugin
Version:	1.0-SNAPSHOT
Action Name:	Counter
Group Alias:	
Override Group:	

Inputs

Input	Required	Type	Template
count	✓	Single Value	Prompt User
reset		Single Value	Value: false
to		Single Value	Value: ${count}
from		Single Value	Value: 1
incrementBy		Single Value	Value: 1

A similar replacement of field values with inputs occurs with steps that reference operations that previously had field values.

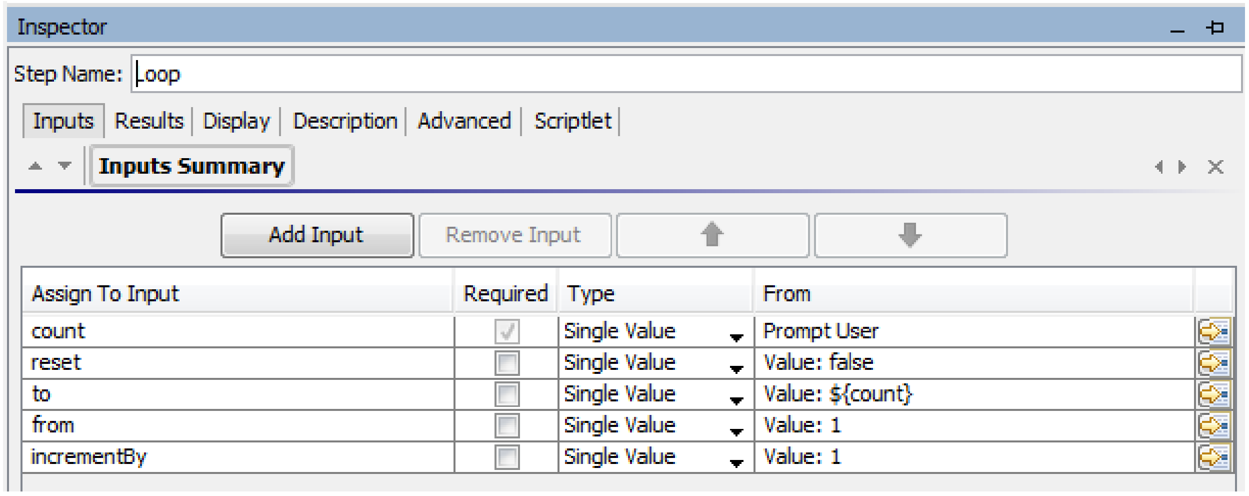
In this example of a step created from the loop operation in 9.x, there are only two inputs, **count** and **reset**.



Step Name: Loop

Assign To Input	Required	Type	From
count	✓	Single Value	Prompt User
reset		Single Value	Value: false

After upgrade to 10.x, there are now five inputs in the step, corresponding to the upgraded operation, which has five inputs.



Step Name: Loop

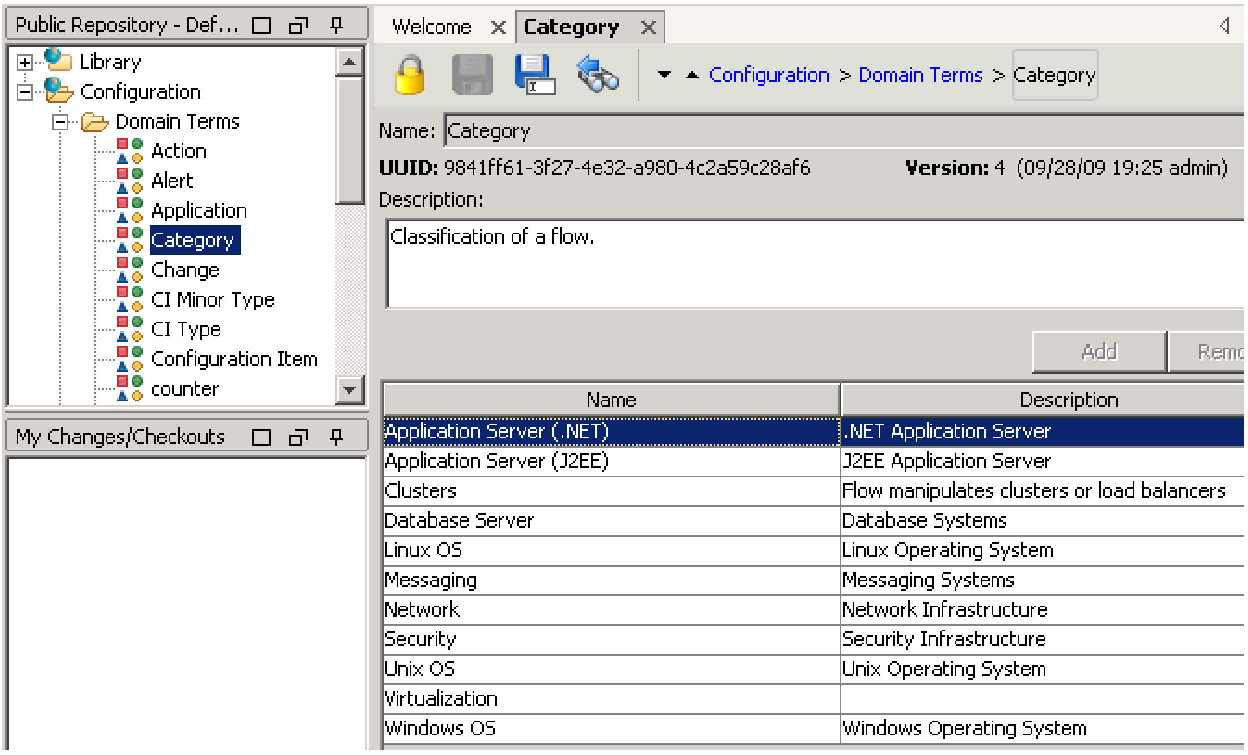
Assign To Input	Required	Type	From
count	✓	Single Value	Prompt User
reset		Single Value	Value: false
to		Single Value	Value: ${count}
from		Single Value	Value: 1
incrementBy		Single Value	Value: 1

If there are step results based on the original field values, these are converted into results based on the inputs, after upgrade.

Note: If the original field names included spaces, these spaces are converted to underscores in the input names after upgrade. For example, **Field 1** is changed to **Field_1**. Step results that use such fields are also changed accordingly.

Domain Term Category

In previous versions, there was a domain term called **Category**, which contained different classifications of a flow. In 10.x, there is a new **Categories** folder under **Configuration** folder.



Name: Category

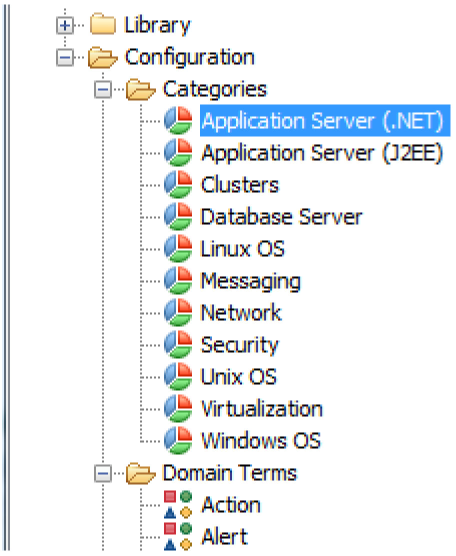
UUID: 9841ff61-3f27-4e32-a980-4c2a59c28af6 Version: 4 (09/28/09 19:25 admin)

Description:

Classification of a flow.

Name	Description
Application Server (.NET)	.NET Application Server
Application Server (J2EE)	J2EE Application Server
Clusters	Flow manipulates clusters or load balancers
Database Server	Database Systems
Linux OS	Linux Operating System
Messaging	Messaging Systems
Network	Network Infrastructure
Security	Security Infrastructure
Unix OS	Unix Operating System
Virtualization	
Windows OS	Windows Operating System

After upgrade, the items that previously appeared as rows in the **Category** domain term now appear as separate items in the **Categories** folder.



"Assign From Variable" Value

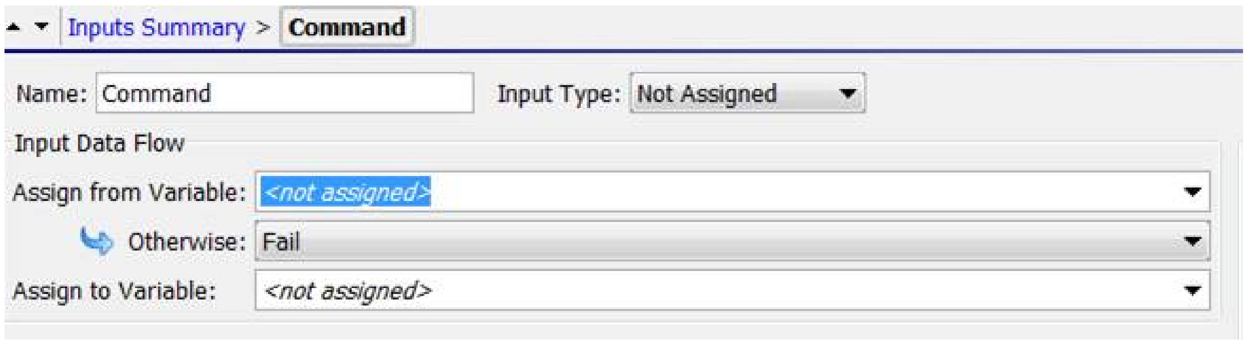
After upgrade, a change is made to flows that have inputs assigned in the following manner:

- **Assign from Variable**: <not assigned>
- **Otherwise**: Fail/Prompt User
- **Assign to Variable**: <not assigned>

Instead of **<Not Assigned>**, the **Assign from Variable** value will have the same name as the input.

In the example shown below, after the upgrade, the value of **Assign from Variable** is **Command**, like the input name.

Before upgrade:



Inputs Summary > Command

Name: Command Input Type: Not Assigned

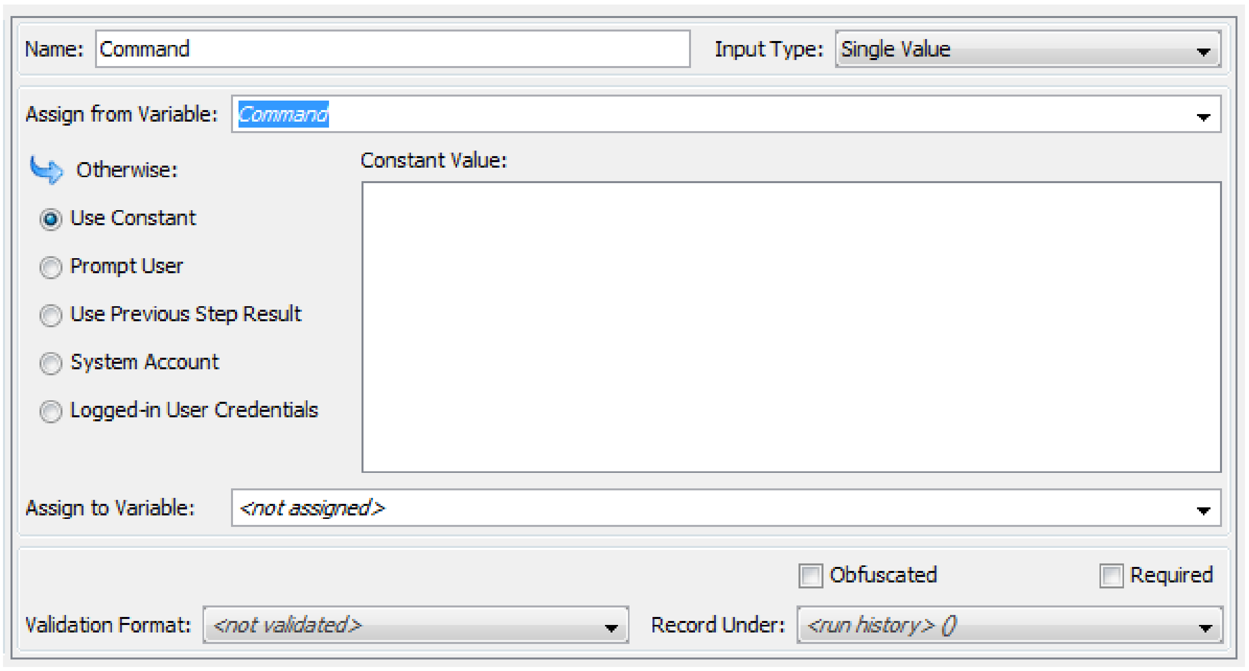
Input Data Flow

Assign from Variable: <not assigned>

Otherwise: Fail

Assign to Variable: <not assigned>

After upgrade:



Name: Command Input Type: Single Value

Assign from Variable: Command

Otherwise:

( ) Use Constant [✓]
( ) Prompt User
( ) Use Previous Step Result
( ) System Account
( ) Logged-in User Credentials

Constant Value:

Assign to Variable: <not assigned>

☐ Obfuscated ☐ Required

Validation Format: <not validated> Record Under: <run history> ()

Credentials Inputs Do Not Exist

In 10.x, the **Credentials** input type no longer exists. Instead the credentials are subtypes of the **Single Value** type, like **Prompt User** and **Use Constant**.

If the upgraded repository contains **Credentials** inputs, these are converted into input types that are supported in 10.x.

For example, inputs of the type **System Account** and **Logged-In User Credentials** will be converted to **Single Value**. However **Credentials** inputs that would prompt the user in 9.x are converted to **Single Value – Prompt User** inputs.

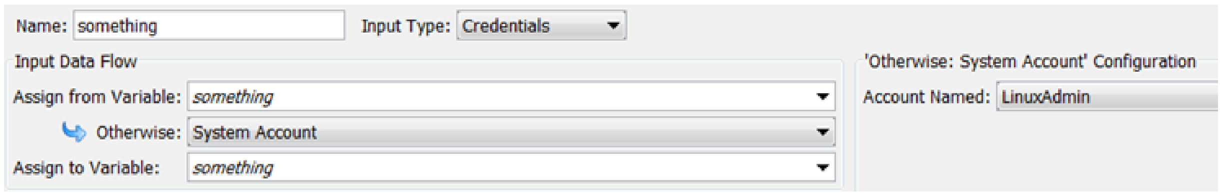
Note: You can change the prompt behavior.

For more details about what happens to **Credentials** inputs after content upgrade, see the use cases below.

When you upgrade content with **Credentials** inputs from earlier versions, these are updated according to the use cases described below.

Use Case 1 - Single Input, No Special Naming

In previous versions, it was possible to create a single input of the type **Credentials**, and assign a system account to that input. The name of the input did not indicate that it was either a user name or a password. This input would receive both the user name and the password of the system account.



Name: something

Input Type: Credentials

Input Data Flow

Assign from Variable: *something*

Otherwise: System Account

Assign to Variable: *something*

'Otherwise: System Account' Configuration

Account Named: LinuxAdmin

In 10.x, an input can no longer be simply attached to a system account. It also has to declare which of the values it is bound to (user name or password).

After upgrade, **Credentials** inputs like these are converted into two separate inputs for user name and password.

For example, if the original input was named **cred**, there will be two inputs named **credUsername** and **credPassword**. This change is also propagated to any steps with inputs that were assigned from **cred** and to subflows and dependent operations.

Use Case 2 - Single Input, No Reference to a System Account

This use case is similar to **Use Case 1**; however, in this case, the credential inputs have no reference to an actual system account.

In previous versions, these kinds of inputs caused the system to prompt the user for the credentials.

After upgrade, these inputs are converted into **Prompt User** inputs.

Use Case 3 - Two Inputs, Using Special Naming

In previous versions, it was possible to create a input of the type **Credentials**, and give it a name that indicated that it was either the user name or password of a system account. This was done by including the word "user" or "pass" in the input name. (For example, "user", "username", "UserName", "identityUser", and so on.)

Note: For more information about this naming convention, see the *HPEOO Studio Authoring Guide*.

This input would extract the user name or password from the system account.

When inputs such as these are upgraded, they are converted into either user name or password values.

Note: After content upgrade, an input is considered a user name input if it contains the word "user", regardless of case, and even if it doesn't conform to the exact format that was required by 9.x ("user", "username", "UserName", "identityUser", and so on.) The same applies for passwords with the word "pass".

Use Case 4 - User Prompts

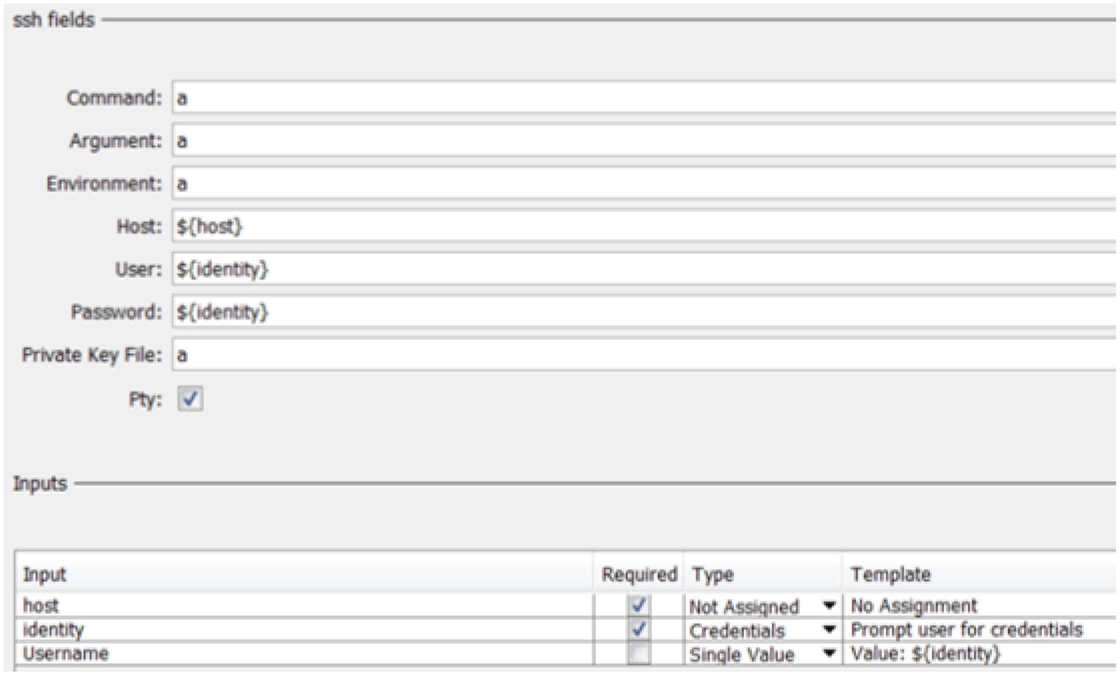
In previous versions, it was possible to create a single input of the type **Credentials**, where users would be prompted to provide both a user name and a password. This input was regarded as a single object containing both values.

After upgrade, **Credentials** inputs like these are converted into two separate user prompt inputs for user name and password. For example, if the original input was named **cred**, there are now two inputs named **credUsername** and **credPassword**. Both are the type **Single Value** and **credPassword** is marked as **Encrypted**.

Use Case 5 - SSH Operations

In previous versions, in SSH operations, it was possible to create **User** and **Password** inputs that took their values from a third, **Credentials** input, using the **${}** convention.

In the example shown below, the **User** and **Password** inputs take their value from **${identity}**, which is a **Credentials** input. In this example, it is a user prompt.



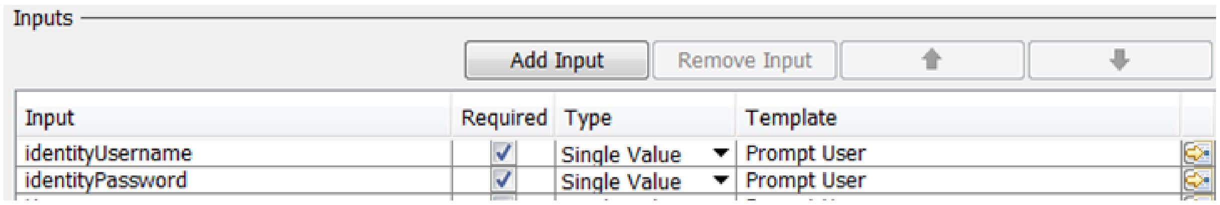
Command:	a
Argument:	a
Environment:	a
Host:	${host}
User:	${identity}
Password:	${identity}
Private Key File:	a
Pty:	✓

Input	Required	Type	Template
host	✓	Not Assigned	No Assignment
identity	✓	Credentials	Prompt user for credentials
Username		Single Value	Value: ${identity}

Field values were removed in 10.x, so it is no longer possible to use the **${}** convention.

After upgrade, the **Credentials** input in an SSH operation is separated into two inputs for user name and password.

In our example, the **${identity}** input is split into two parts: **IdentityUsername** and **identityPassword**. The **Username** and **Password** inputs that are created from the field values will reference those two new inputs.



Input	Required	Type	Template
identityUsername	✓	Single Value	Prompt User
identityPassword	✓	Single Value	Prompt User

Logged-in User Includes Prompt

In previous versions, it was possible to create an input for **Logged-in User Credentials**.

After upgrade, this input type includes a user prompt, to improve security.

The password is no longer automatically transferred when a **Credentials** input is of the type **Logged-In User Password**. To enable automatic execution, the administrator should select **Enable Capture of Logged-in User Credentials** in Central, under **Security**>**Security Settings**. Otherwise, the user will be asked to enter the password manually in a prompt message.

References to Integrations/Hewlett-Packard/Operations Orchestration/9.x Content are Changed to 10.x

After upgrade, the content in the **/Library/Integrations/Hewlett-Packard/Operations Orchestration/** folder is moved into the **/Library/Integrations/Hewlett-Packard/Operations Orchestration/9.x** folder. The operations and flows in this folder can be used to integrate with 9.x from 10.x.

As part of the 10.x upgrade process, any steps that reference legacy OO-to-OO operations from the **/Library/Integrations/Hewlett-Packard/Operations Orchestration/9.x** folder are replaced automatically with references to content from the **/Library/Integrations/Hewlett-Packard/Operations Orchestration/10.x** folder.

If you want to disable this behavior before running the Content Upgrade Utility, open the **/cmu/operation_references/operationReferenceReplacements.properties** file and comment out (using #) any lines containing UUID correspondences. For example:

```
#9.x/Dynamically Launch Flow -> 10.x/Dynamically Launch Flow

0e227211-ffe5-4b24-8dd2-84071e3efa92=98102ebe-7a26-4398-afde-e01756578879
```

FailureMessage and TimedOut Results

After upgrade, flows no longer have built-in FailureMessage and TimedOut results.

Source Control in Studio

In previous versions, Studio included its own proprietary version control capabilities. Studio 10.x is integrated with a standard source control application (SVN). Versioning of library items and configuration items is achieved via integration with the source control management system.

The following source control functions from Studio 9.x are not supported in Studio 10.x:

- Maintenance of users
- Access control to HPE OO content (permission to read/modify HPE OO content)
- Automatic backup of the public repository
- Updating the contents of Central with Studio (by direct update and check-in of the public repository or by publishing from the local repository)

Users of Studio 10.x must learn to achieve these functions using the SCM tools.

No OO Portal

OO Portal has been discontinued in 10.x.

Change in Behavior When a RAS is Unavailable

In 9.x, if a RAS is unavailable when the flow is invoked, the step using the target RAS fails. In 10.x, the flow gets the status **Paused - No Workers in Group** at the step.

This is a change in behavior, which might require you to change the design of some of your flows during upgrade from 9.x to 10.x.

For example, you may have a flow in your 9.x repository with a RAS override to access to the remote RAS, and a step using the RAS override expects "failure" at the step if the RAS is unavailable, and the flow sends an email when it fails. This scenario (flow logic) does not work on 10.x, because the step does not fail and will only be **Paused - No Workers in Group**.

Scheduled Runs Do Not Automatically Use an Empty Value for Inputs that Use Prompt User

In 9.x, if you authored a flow with **Prompt User** inputs and then you scheduled it without adding any value to the non-required inputs, the flow would run successfully, without stopping to prompt for values. It behaved as if the inputs were left empty.

In 10.x, this behavior was changed in order to achieve consistency between headless and interactive runs. The run is stopped and prompts for inputs, even if they are not required inputs.

In order to run the old flows in the same way that they ran in 9.x, you can use the **Use empty value for prompts** check box in the Scheduler. For any prompt that has no value assigned, it will use blank, and the scheduled flow will run exactly as it ran in 9.x.

Converting Flows using Multi-instance Steps from 9.x to 10.x Format

Background

The architecture of multi-instance steps has changed in 10.x.

The changes include:

- Steps that run within multi-instance step branches (internal steps) have access only to branch context and not to the overall step and global contexts of the flow. This affects, among other things, the work of **List Appender** operations used to aggregate branch results.
- Usages that were considered bad practice in 9.x are not supported in 10.x, as follows:
  - Direct transitions from the internal steps of a multi-instance step to return steps are not supported. Only the **Group-done** transition of the multi-instance steps can be used for this purpose.
  - Parallel steps cannot be used as internal steps in a multi-instance step.
  - Other multi-instance steps cannot be used as internal steps in a multi-instance step.

Accessing Step and Global Context from a Multi-Instance Branch

In 10.x, there is a separation between the flow context and the branch contexts of multi-instance and parallel steps. When variables are put into the context from internal steps of a multi-instance step, these variables do not reach the flow context unless a special step scriptlet is added to the multi-instance step, which merges these variables from branch to flow context.

How Steps with List Appender Operations are Converted

Using List Appender to Aggregate Branch Results

It is common practice in 9.x to use the **List Appender** content operation to aggregate the results of all branches of a multi-instance step, in order to handle them later on in the flow.

Since **List Appender** works by appending values to a variable located in the flow context, in 10.x by default this variable will not exist in the flow context after the **Group-done** transition of a multi-instance step. However, the step scriptlet of the multi-instance step can be used to merge the contents of each branch context into the flow (step or global) context.

For example:

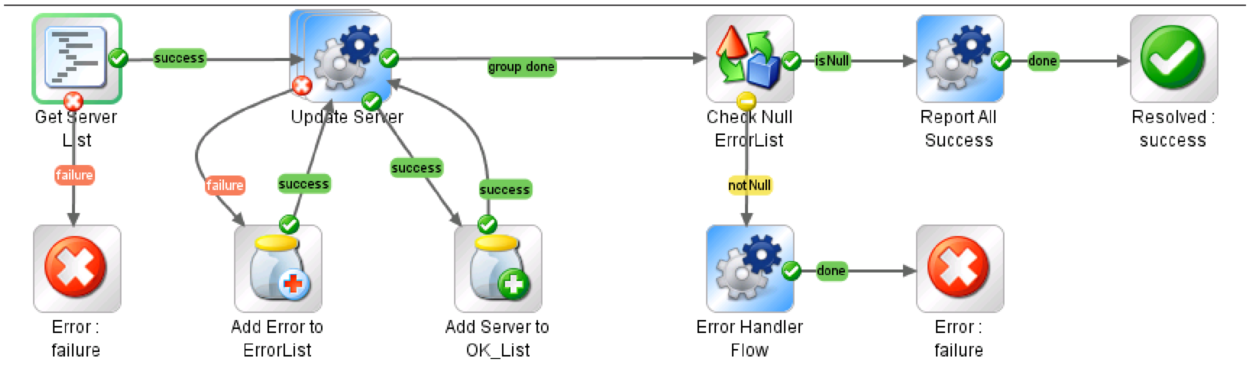
```
var appenderKeyName = scriptletBranchContext.get("appenderKeyName");
var appendedValue = scriptletBranchContext.get("appendedValue");
var delimiter = scriptletBranchContext.get("delimiter");
var currentVal = scriptletContext.get(appenderKeyName);
if(currentVal == null) {
scriptletContext.put(appenderKeyName, appendedValue);
} else {
scriptletContext.put(appenderKeyName, currentVal + delimiter + appendedValue);
}
```

Note: This scriptlet runs once at the end of each branch. The "appenderKeyName", "appendedValue" and "delimiter" variables should be created by the step scriptlet of the corresponding **List Appender** step from its respective inputs.

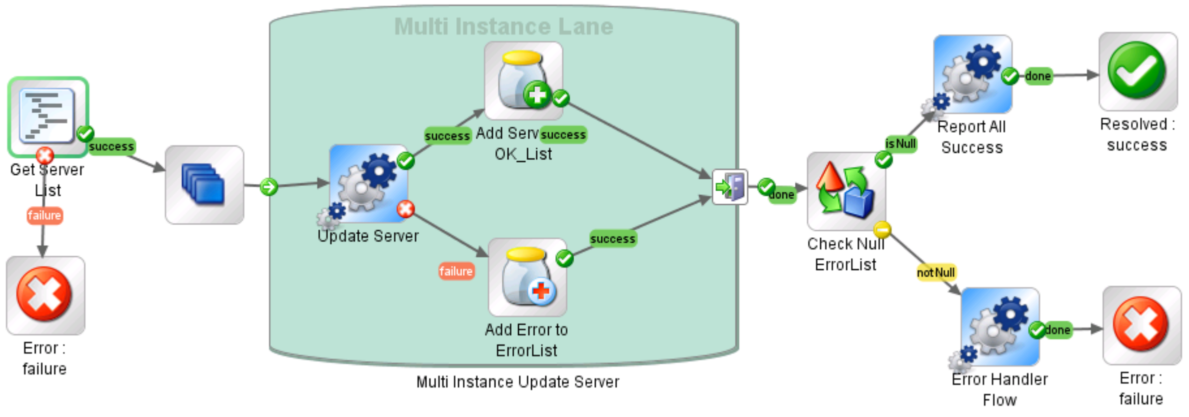
Important: In 10.21 and later, the Content Upgrade Utility creates similar scriptlets automatically. Therefore a 9.x flow using **List Appender** to aggregate the results of multi-instance branches will work well after conversion to 10.x format, without a need for further editing, provided that only one **List Appender** instance runs per branch. (See Special Cases).

Examples

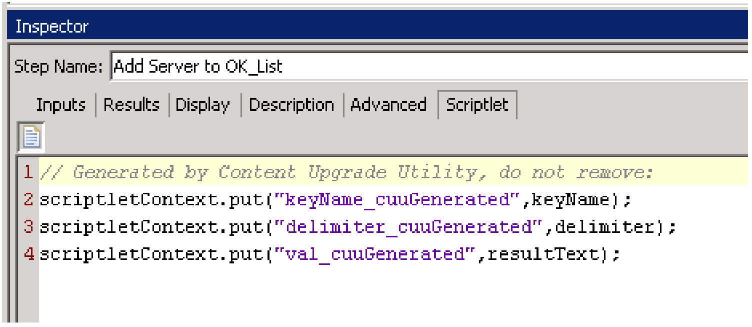
9.x flow before conversion:



10.x flow after conversion:



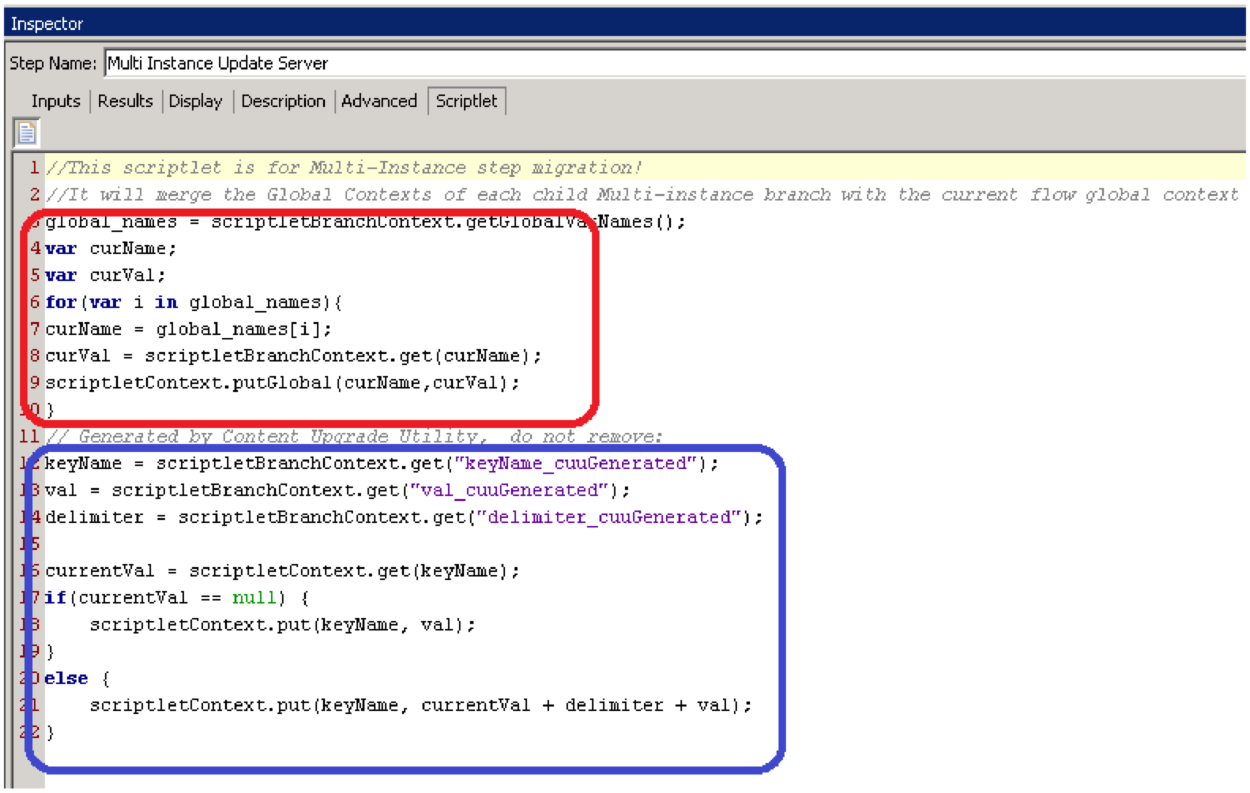
Scriptlet added to the **Add Server to OK_List** step:



```
// Generated by Content Upgrade Utility, do not remove:
scriptletContext.put("keyName_cuuGenerated",keyName);
scriptletContext.put("delimiter_cuuGenerated",delimiter);
scriptletContext.put("val_cuuGenerated",resultText);
```

Note: A similar script is added to the **Add Server to ErrorList** step. Since only one of these steps is executed for each branch of the multi-instance step, the corresponding variables put on the scriptletContext will not collide. Each **List Appender** step can have its own keyName, delimiter and appended value.

Scriptlet added to the multi-instance step:



```
//This scriptlet is for Multi-Instance step migration!
//It will merge the Global Contexts of each child Multi-instance branch with the current flow global context
global_names = scriptletBranchContext.getGlobalVarNames();
var curName;
var curVal;
for(var i in global_names){
curName = global_names[i];
curVal = scriptletBranchContext.get(curName);
scriptletContext.putGlobal(curName,curVal);
}
// Generated by Content Upgrade Utility, do not remove:
keyName = scriptletBranchContext.get("keyName_cuuGenerated");
val = scriptletBranchContext.get("val_cuuGenerated");
delimiter = scriptletBranchContext.get("delimiter_cuuGenerated");

currentVal = scriptletContext.get(keyName);
if(currentVal == null) {
    scriptletContext.put(keyName, val);
}
else {
    scriptletContext.put(keyName, currentVal + delimiter + val);
}
```

- The fragment highlighted in red is generic for all multi-instance steps. It merges all variables from the branch context to the global context.
- The fragment highlighted in blue is specific for multi-instance steps using **List Appenders**.

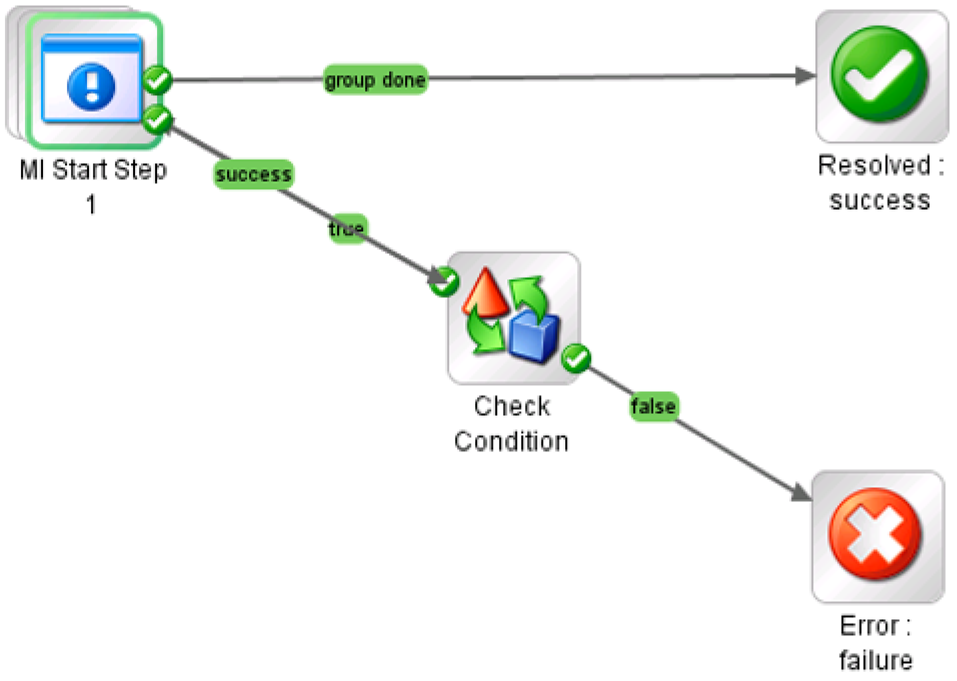
Bad Practice Usages Automatically Converted to Best Practice by CUU

10.x does not support direct transitions from internal steps in a multi-instance step to return steps (**Resolved**, **Error**, **Diagnosed** and **No Action Taken**). However, starting from 10.21, the Content Upgrade Utility can convert multi-instance steps with direct transitions to **Error** steps into a format compatible with 10.x.

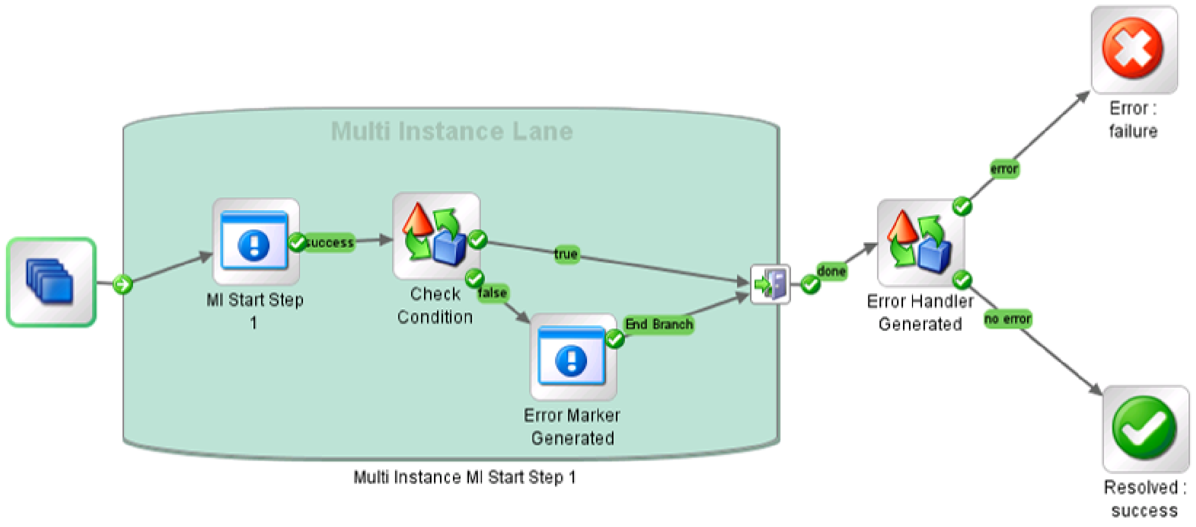
To accomplish this, the CUU creates a new internal **Error Marker** step for each multi-instance step with such a direct transition. This step is based on the **Do Nothing** operation and has a scriptlet that puts a variable on the branch context, marking that an error transition should be forced after the **Group-Done** transition. Additionally, a new **Error Handler** step is added after the **Group-done** transition, which checks this variable, and if it exists, chooses a transition to an **Error** step.

Examples:

9.x flow before conversion:



10.x flow after conversion:



Bad Practice Usages That Require Manual Editing After Conversion

Direct transitions from internal steps of a multi-instance step to return steps (other than Error)

Such flows cannot be converted automatically to 10.x. The corresponding transition will be removed by the CUU and the resulting flow will be placed under the **Invalid Content** sub-folder in the CUU output folder.

Solution: Consider adding a new transition to the **Group-done** step of the multi-instance step, instead of the removed one.

Parallel steps used as internal steps of a multi-instance step

10.x, does not support multi-instance steps using a parallel step as an internal step. Such 9.x flows cannot be converted automatically to 10.x. The corresponding transition will be removed by the CUU and the resulting flow will be placed under the **Invalid Content** sub-folder in the CUU output folder.

Solution: To make the flow valid in 10.x, you have options:

- Wrap each internal parallel step in a sub-flow
- Refactor the flow, so that the parallel step is executed after the **Group-done** transition of the multi-instance step

Other multi-instance steps used as internal steps of a multi-instance step

10.x does not support multi-instance steps having another multi-instance step as an internal step. The corresponding transition will be removed by the CUU and the resulting flow will be placed under the **Invalid Content** sub-folder in the CUU output folder.

Solution: To make the flow valid in 10.x, you have two options:

- Wrap each internal multi-instance step in a sub-flow
- Refactor the flow, so that the second multi-instance step is executed after the **Group-done** transition of the first one

Special Cases

Direct transitions to distinct Error steps from a single multi-instance step

Direct transitions to a single **Error** step from within a multi-instance branch can now be converted automatically to 10.x format by the CUU. However, in the case of direct transitions to multiple distinct **Error** steps from a single multi-instance branch, the flow will still be placed under the **Invalid Content** sub-folder and might require manual editing. The reason is that some **Error** steps may become orphan (without any incoming transitions) after the conversion.

Multiple List Appender steps used in the same branch of a multi-instance step

After conversion, such flows may behave differently from the way they behaved in 9.x. The reason is that the scripts added by the CUU take into account the common case when a single instance of a **List Appender** is used in every branch (in the example above, either **Error Appender** or **Success Appender**, but not both).

Known Issues and Troubleshooting

Because of the changes in 10.x, there are some types of flows that may not upgrade properly. In some cases, you will be able to modify the flows, to get them to work properly. In other cases, it is recommended to save the original flow, to be used in later versions of HPE OO, and in the meantime, to create a new flow that works differently.

Content that is not supported by 10.x is stored in a separate project after upgrade, and is not included in the content pack. For more information about what kind of content is not supported, see the *OO 10.00 Base Content Pack 1.0 Release Notes*. For more information about features that are not supported in 10.x, see *Known Issues and Limitations* in the *OO 10.x Release Notes*.

Missing System Accounts

Issue: After content upgrade, there may be flows or steps in which inputs reference system accounts that do not exist.

Solution: Open the flow in Studio 10.x and change the input so that it references an existing system account.

In cases such as this, it does not help to go to the original repository in 9.x to look for the missing reference. It must be fixed in the 10.x project.

Duplicated UUIDs in an Upgraded Project

Issue: If you upgraded content with versions prior to 10.02, you may get errors in Studio 10.x and in Central 10.x deployment, and you may have problems with installing the out-of-the-box content packs, such as the Base Content Pack 13.

The problems are caused by duplication of configuration items (system properties, system accounts, domain terms, selection lists, and so on). These items were part of the HPE out-of-the-box content and there may be duplications of these in your upgraded content. Out-of-the-box content should be treated as read-only. This principle was violated by former versions of the Content Upgrade Utility and version 10.x fixes this.

Solution:

- Upgrade your repository again, using the Content Upgrade Utility for 10.x (recommended).
- Fix your repository manually:
  a. Open your project in Studio, and go to the configuration folders.
  b. Locate the duplicated items (they have an error indication) and delete them.

     Note: The **Category** domain term cannot be deleted from within Studio. If this item is duplicated, you will need to delete it from your file browser, outside of Studio.

     For example, right-click on the item, select **Show in Explorer**, delete it from the file system, and then refresh the project in Studio.

  c. Deploy your exported content pack with the HPE Base Content Pack (version 12\13).

  Note: If you edited selection lists from the out-of-the-box content, you need to create them again, rename, and replace all usages.

Modified OOTB Scriptlets are Overwritten After Upgrade

Issue: If the 9.x repository contains a copy of an out-of-the-box operation with a scriptlet, and that scriptlet was modified, after content upgrade, the modified scriptlet will be overwritten by the out-of-the-box version.

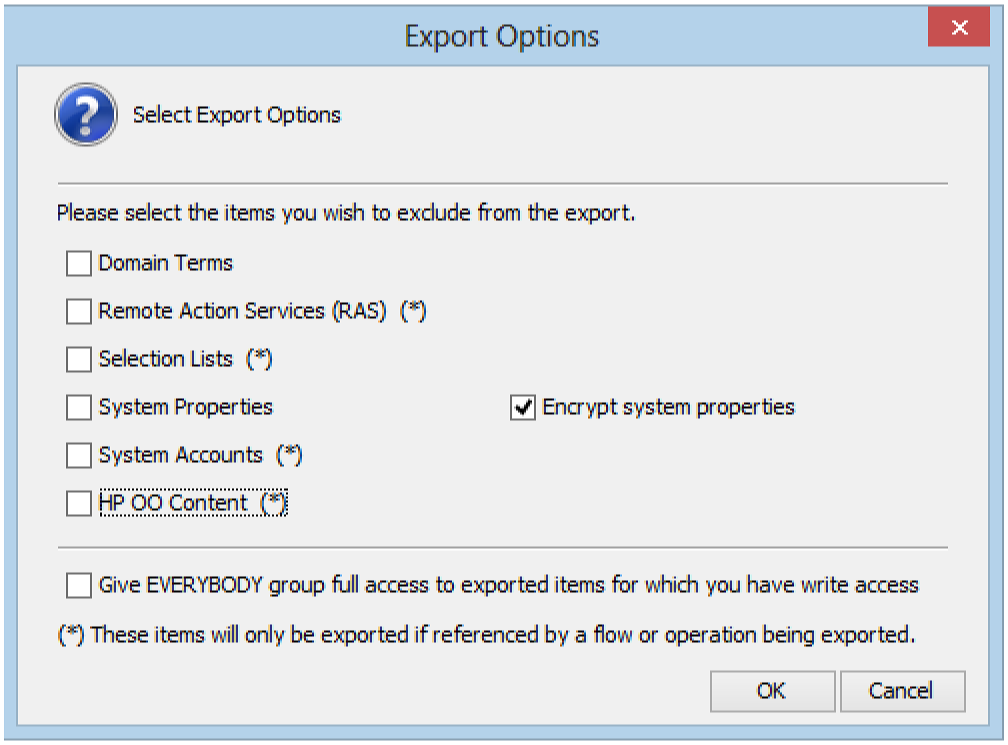
Solution: If the change in the scriptlet is important, you will need to manually edit the scriptlet in the copied operation.

Problems Upgrading Repositories

Issue:When you upgrade a repository that is not fully exported from Studio you may receive a warning, because some data is invalid.

Solution:

1. In 9.x, export the repository from Studio: Right-click **Repository**, and select **Export as New Repository**.
2. Select a name and location for the repository.
3. In the Export Options page, make sure that all the check boxes are cleared (not selected) apart from the **Encrypt system properties** check box. Checking the **Encrypt system properties** check box is optional. However, note that after content upgrade, the system properties will be obfuscated with the 10.x mechanism.



Export Options

Select Export Options

Please select the items you wish to exclude from the export.

- [ ] Domain Terms
- [ ] Remote Action Services (RAS) (*)
- [ ] Selection Lists (*)
- [ ] System Properties
- [x] Encrypt system properties
- [ ] System Accounts (*)
- [ ] HP OO Content (*)

- [ ] Give EVERYBODY group full access to exported items for which you have write access

(*) These items will only be exported if referenced by a flow or operation being exported.

4. Run the Content Upgrade Utility on that repository.

Note: Exporting a repository fixes the cases of invalid data.

Studio Errors After Upgrading a DotNet iAction

Issue: After you upgrade content that includes .NET iActions, there may be Studio errors because of a third party DLL dependency that is only for 32-bit platforms. For example, **Microsoft.GroupPolicy.Management.dll** and **Microsoft.GroupPolicy.Management.Interop.dll**.

Solution: If the repository includes custom .NET iActions, it is recommended to clean your RAS folder copy and leave only the DLLs that are required for your iActions before upgrading content.

Note: Make sure to copy the RAS lib directories to a different location before deleting anything from them and upgrading them.

Backup Corruption

When rolling back the 10.x patch, if one of the folders was renamed or if some files were deleted, the rollback will fail and the application (Central\ Studio\ RAS) will not start.

Solution: Save the **backup** folder that is created after the upgrade process as a ZIP file, and extract it to the same location before rolling back the upgrade.

Permission Errors in Linux

Issue: The content upgrade package for Linux does not have execution rights by default for the files **upgrade-content.sh** and **/java/bin/java**. This may cause you to receive error messages like the following:

```
-bash: ./upgrade-content.sh: Permission denied

./upgrade-content.sh: line 5: ../java/bin/java: Permission denied
```

Solution: To avoid this, grant execution rights for the files.

Content Pack Dependencies

Issue: Deployment of upgraded content fails due to a dependency error.

Solution: Review the upgrade report and make sure you have deployed all the content packs that are mentioned in it. For more details about the content packs that the Content Upgrade Utility recognizes, and how to handle other content, see "Exclude Partner Content from Upgrade" in "Step 3: Upgrade Content" on page 26 .

Review the deployment error and look for the missing UUIDs in your 9.x Studio content. This will enable you to better identify the missing content.

Sleep Scripts Not Supported

Issue:Sleep scripts are not supported in 10.x. If you upgrade flows with scriptlets that are written in Sleep, they will be separated to the project containing the invalid content.

Solution: After upgrade, rewrite the Sleep script as JavaScript.

Upgrade Fails for Flows With Input Binding of Type Credential

Issue: Upgrade fails for flows that have input binding of the type **Credential**, which is not assigned to any system account.

Solution: Change these inputs to user prompt inputs.

Long Group Names Not Upgraded

Issue: Groups with names longer than 255 characters are not upgraded to roles, during content upgrade.

Solution: Change the group names to shorter names, before upgrading.

Long Inputs Not Upgraded

Issue: Schedules containing input values longer than 4000 characters are not upgraded, during content upgrade.

Solution: Change the input values so that they are less than 4000 characters, before upgrading.

LDAP Configuration is not Upgraded

Issue: LDAP is not upgraded and a 400 error appears on `POST http://localhost:<port>/oo/rest/authns/ldap-config`.

Solution: This may occur because there are LDAP configurations that are supported in 9.x but are not supported in 10.x. In this case, the **User** filter was `(objectClass=*)` and did not contain a user name.

The LDAP upgrade calculates the **User ID** attribute according to the **User** filter.

- If the filter contains one user name, this user name is taken.

  For example, if the **User** filter is `(&(objectClass=person)(sAMAccountName={0}))`, the **User ID** attribute is **sAMAccountName**.
- If the filter contains 0 or more than one user name, the **User ID** attribute will be one of the following:
  - **sAMAccountName**, if the LDAP is Active Directory
  - **uid**, otherwise

  For example, if the filter is `(&(objectClass=person)(|(sAMAccountName={0})(uid={0})))`, then the **User ID** attribute will be **sAMAccountName** if the LDAP is Active Directory and **uid**, otherwise.

Operations with Missing Responses Not Identified as Invalid

Issue: In some situations, operations with missing responses end up in the valid project instead of the invalid project.

Solution: After upgrading, you will need to manually check the operations to see if they have missing responses. These are not identified as invalid during the upgrade process.

Different Size Limit for Prompts

Issue: The size limit for prompts in a flow is now 255 characters, while in previous versions, the size limit was 1024 characters. After an upgrade, prompts with more than 255 characters will be cut down to the correct size.

Solution: After upgrading, check your prompts and see if there are any that are missing information and need to be rephrased.

Uncompleted Flows Not Upgraded

Issue: If there are flows that were not completed before an upgrade (for example, those that are running or paused), they will be canceled during the upgrade.

Solution: Make sure that all flows have finished running before starting the upgrade.

Multi-instance Steps

In 10.x, multi-instance steps are created in a multi-instance branch. In previous versions, flow authors would use the **Toggle Multi-instance** option to turn a regular step into a multi-instance step, and would create multiple loops for the different targets of the step.

Issue: In 9.x, a multi-instance step was a single step, with a single set of properties. In 10.x, there are properties for the MI branch step and also properties for each inner step within the branch. The flow will not work properly after upgrade, unless the properties of the MI branch step and the inner steps are set up correctly.

Solution: Adjust the properties for the MI branch step and the inner steps, so that they correspond with the guidelines in the table below.

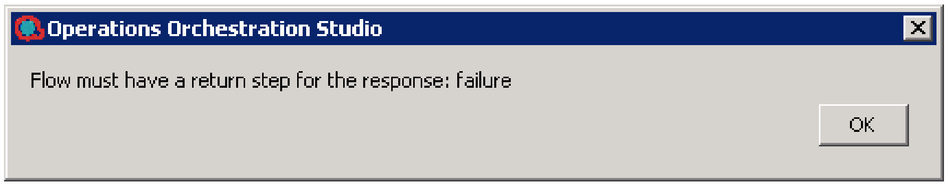
Open the Step Inspector for the MI branch step by double-clicking the **Multi-step** icon at the start of the step.

Open the Step Inspector for each inner step by double-clicking the relevant step icon, inside the branch.

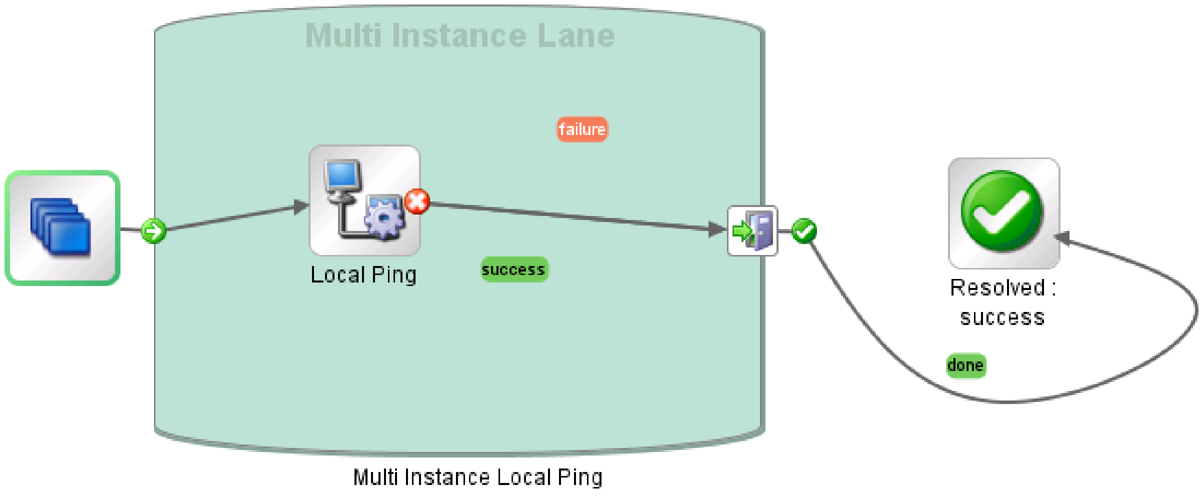
Property	Inner steps	MI branch step
Step UUID	Same as original (before upgrade)	Same as original
Assign Input from	Input name	Same as original
Assign Input to	Not assigned	Same as original
Result - From key name	Same as original	Same as the result name
Result - Assignment	Overwrite	Same as original
Result - Filter	Same as original	None
Scriptlet	Same as original	Upgrade scriptlet - will merge all the global variables of the inner steps.

Can't Save a Flow with a Multi-instance Step

Issue: After content upgrade to 10.x, a flow with a multi-instance step goes to the invalid project. After fixing the flow in Studio 10.x, it is not possible to save the flow.


Operations Orchestration Studio

Flow must have a return step for the response: failure



Solution: Delete all return steps and replace them with new return steps in Studio 10.x.

For more information about upgrading content with multi-instance steps, see "Converting Flows using Multi-instance Steps from 9.x to 10.x Format" on page 62.

Automatic Update of the OO to OO 10.x Integration

After upgrade, all flows using operations from **/Library/Integrations/Hewlett-Packard Enterprise/Operations Orchestration** will have their step references replaced with the corresponding operations from the **Library/Integrations/Hewlett-Packard Enterprise/Operations Orchestration/10.x** folder.

For example, if you upgrade a flow that uses **Launch Flow** (611ee86d-249e-41e2-b464-fef310ecb492) from Content Pack 11, you will get a flow that uses **Launch Flow** (4556e22b-6acf-47b9-801f-9a1d3615545e) from the **10.x** folder.

If you want to disable this behavior before running the Content Upgrade Utility, open the **/cmu/operation_references/operationReferenceReplacements.properties** file and comment (using #) the lines containing UUID correspondences, as shown below:

```
#9.x/Dynamically Launch Flow > 10.x/Dynamically Launch Flow
```

```
#0e227211-ffe5-4b24-8dd2-84071e3efa92=98102ebe-7a26-4398-afde-e01756578879
```

New Inputs: Host, Port, Protocol, Username, Password

Issue: Some operations worked as plugins in 9.x and were missing connection and credential inputs. In 10.x, these operations connect to Central through REST API and therefore need these new inputs.

After upgrade, your copies of these operations and the steps in your flows referencing these operations have been added the new host, port, protocol, username, password inputs automatically, but the **Assign From** is set to **Prompt User**.

These are the operations and the added inputs:

- **Get All Children Named** (08571e7b-24ef-4027-858f-619ad36ab36e): host, username, password, port, protocol
- **Get Children of Path** (5e0e3894-8b18-4a61-9c23-784204f829ad): host, username, password, port, protocol
- **Delete Flow Schedule** (49f89d9d-99e2-46f5-8b10-ba878a203303): username, password
- **Get Flow Schedules** (590cde98-60d5-43e1-b3ed-35e16e8a3075): username, password
- **Get Schedule Details** (2316a0c2-5013-4a37-9a93-dae4623c3c17): username, password
- **Flow Run Summary Report** (2e871c48-8073-4c08-ae83-8fa9ccfafeb6): host, username, password, port, protocol
- **Flow Run Counter** (6431833f-2f65-4b55-971c-a9b7d8be36ce): host, username, password, runId, port, protocol
- **Resume Flow Run** (55188216-67cb-49b2-a522-82667cc4b07d): host,username,password,port,protocol
- **Set System Account** (046a86df-3bf4-4e66-aa48-88826e741215): host, username, password, port, protocol, persistInDatabase
- **Set System Property** (88991b3a-756d-44b9-9fed-107caf581624): host, username, password, port, protocol, persistInDatabase

Solution: Change the default **Assign From** for these inputs as required.

Operations With No Equivalent in 10.x

Issue: Some operations from previous versions do not have an equivalent in 10.x. The following legacy out-of-the-box content from the **/Library/Integrations/Hewlett-Packard/Operations Orchestration** folder are not supported in this version:

- **Get Cluster Servers** (7b925f4f-ca85-492f-895c-f0d5477ab0d3)
- **HPE Load Balancer admin** (a2888f6b-e860-4370-803c-3a18b0e809e8)
- **Get Cluster Servers** (4b3eed95-c688-4e79-9c52-0ff4d31c3822)
- **Get Server Status** (6e48fd8a-b321-4a61-be78-6e1ccb9eb1f2)
- **Get Stored Flow Variable** (fe7b058d-3925-4447-af51-6f4d178a2a8b)
- **Store Flow Variable** (cb5e7fa1-83f2-4e1d-9ce9-e6f13848fae9)
- **Flow Run Counter** (c2ccf035-fead-4228-a27f-d1f48afcd83a)
- **Publish Master to Slaves** (3fcba2af-eda9-42b8-b8b2-2391c46f31e5)
- **Unlock Repository** (b23f42fa-028a-4e06-ac4e-b6a9b32333c3)
- **Clean Deprecated Library Content** (11fd655a-c254-49e9-9b91-2c22daebc296)
- **Publish Staging to Production Cluster** (595363bb-4334-44e9-ab7b-6a7fb16af608)
- **Publish To Central** (aa682862-187a-4d4a-867c-3da35a875096)
- **Rollback From Snapshot** (c8ec64a3-38a8-460f-b361-c802b2f96145)
- **Set Selection List from Database** (cc11ad02-54ed-48eb-acd8-123b32c0aa8d)
- **Single Pass Delete Unused Content** (7a285270-257e-48b2-9267-0947b0a01415)
- **Snapshot Then Publish** (031449ec-6f99-4fbd-b79f-8a9d74d912fd)
- **Check In** (c2de85be-5b42-44e1-8f01-78fa51de6d54)
- **Create Snapshot** (5364cba6-77f9-4243-ae0f-472ce7895c8e)
- **Delete Path** (0ed348f1-d307-4326-bced-1d9bffceafa1)
- **Delete Snapshot** (10ef3842-5e07-4751-ac0c-582d59c0f0b2)
- **Get Last Modified By** (6172353d-f160-42a6-bb31-5328992ba154)
- **Get References to Path** (4f3970a8-84dd-4934-b21b-35e20a75c4b8)
- **Repository Sync** (592493c1-9eb5-4977-be23-5eda23b72e68)
- **Basic Schedule Sample** (56d68a64-9dfb-4c3c-8073-f48bcc40e70c)
- **Data Persistence Sample** (45a14f1d-1a26-4f15-b60f-a433b896df79)
- **Data Persistence Table Sample** (885ae2d9-20a1-4e59-940d-4a24ffed1bb1)
- **Generate and Send Documentation** (9fb281c9-8732-4fe3-8412-f42ab44c087c)
- **Get New Email Data Persistance Sample** (351d73de-9ca0-48b4-bdb5-419f0e6413ea)
- **Schedule Flow on OO up to 9.02** (8c4f42ca-d3aa-4d3a-8264-388065c50f6c)
- **Generate Documentation** (591cc7f1-1f6d-4999-8e48-46cf227fa6f3)
- **Generate Documentation with Hidden Folders** (7147b4a8-15e6-4f9e-bdd3-ce9f122feb2d)

Other Integration Issues

Get All Children Named

- The 10.x operation will only return a list of flows unlike the 9.x operation that also returns operations.

Get Children of Path

- The 10.x operation will only return a list of flows unlike the 9.x operation that also returns operations.

Get Schedule Details

- Missing outputs (in 10.x compared to 9.x): **executing**, **runsLeft**
- The 10.x **repeatInterval** output is in cron format (for example, */3600000) while in 9.x, it is in milliseconds (for example, 300000).
- The 10.x **endTime** output value is 0 in 10.x if no **endTime** was set for the schedule and is empty in 9.x for the same situation.

Deprecated\Flow Run Counter

- In addition to the non-deprecated operation, there is a **Deprecated/Flow Run Counter** operation. This operation will not upgrade correctly. You will need to replace the steps with the valid out-of-the-box operation.

Resume Flow Run

- The outputs **runHistoryId**, **runReturnCode** and **runReportXML** are no longer used (are empty in 10.x).

List Flow Run History

- The input **startIndex** is missing in 10.x.
- The 9.x **Result** and **returnResult** contain a json field named **status** that is a capitalized string (for example, **Resolved**) while the 10.x equivalent field output is all caps (for example, **RESOLVED**).
- The 9.x **Result** and **returnResult** contain a json field named **flowRevision**. This field is missing in 10.x.

Set Selection List

- The input description is missing in 10.x.
- The 10.x **returnResult** output contains the selection's list UUID while the 9.x **returnResult** has a descriptive text indicating success or failure.

Get Run Status

- The 9.x **runResponse** output is a capitalized string (for example, 'Resolved') while the 10.x **runResponse** output is all caps (for example, 'RESOLVED').
- The 10.x **status** output can take values from **RUNNING**, **COMPLETED**, **SYSTEM_FAILURE**, **PAUSED**, **PENDING_PAUSE**, **CANCELED**, and **PENDING_CANCEL** , while the 9.x status output takes values from **Unknown**, **Running**, **Paused**, **EndedOrFinished**, and **Canceled**.

Get Status For Runs

- The 10.x **Result** and **returnResult** contain a json field named **status** that takes values from **RUNNING**, **COMPLETED**, **SYSTEM_FAILURE**, **PAUSED**, **PENDING_PAUSE**, **CANCELED**, and **PENDING_CANCEL** while the 9.x status takes values from **Unknown**, **Running**, **Paused**, **EndedOrFinished**, and **Canceled**.
- The 10.x **Result** and **returnResult** contain a json field named **runResumeUrl**. This field is missing in the 10.x equivalent outputs.

Store System Account in Flow Variable

- The credential input/flow variable is no longer supported in 10.x; therefore, this operation will assign the account to two separate variables. So if the input **flowVar** was **sysAcct**, two flow vars will be created: **sysAcctUsername** and **sysAcctPassword**. Your flow steps have been upgraded to use these two flow variables.
- The **returnResult** and **response** output fields are missing in 10.x. Use the operation response instead of the output field response and the operation's primary result instead of the **returnResult** output field.

Run Flow(s) From E-Mail(s)

- These flows have been moved to the Utility Orchestration content pack.

Dynamically Launch Flow

- The output fields **Result** and **returnResult** contain an xml that has the tags **flowresponse** and **flow-return-code** among others. The **flow-response** from the 10.x operation takes values from **RESOLVED**, **DIAGNOSED**, **ERROR**, and **NO_ACTION_TAKEN**, while the OO 9.x operation takes values from **success**, **failure**, **diagnosed**, and **no action**. The **flow-return-code** also changes from 9.x (for example, **Finished**) to 10.x (for example, **RESOLVED**).

Set System Account

- If **persistInDatabase** and the **host**, **port**, and **protocol** inputs are left empty, the operation will try to connect by default to this central url: http://localhost:8080/oo.

Set System Property

- This operation also has a **persistInDatabase** input field (like **Set System Account**). If this input is set to **false**, the server and authentication inputs are ignored (**host**, **port**, **protocol**, **username**, **password**).
- If **persistInDatabase**=true and the **host**, **port**, and **protocol** inputs are left empty, the operation will try to connect by default to this central url: http://localhost:8080/oo.

  As a workaround, make sure you set **persistInDatabase** to **false** if you need to run the operation in Studio only (and not connect to Central).

Generate Run URL

- When run in Central, this operation uses the context variables **CENTRAL_URL** and **run_id** to create the drilldown URL for the current run. If run from Studio, it returns "http://localhost:8080/oo".

10.x/ (Dynamically) Launch Flow

- The operation will not work if authentication is disabled in Central. As a workaround, enable authentication in Central.

Hewlett Packard
Enterprise

Java™
COMPATIBLE