

# HP Operations Orchestration

For Windows and Linux

HP OO Software Version 10.01

## Upgrade Guide

Document Release Date: August 2013

Software Release Date: August 2013



# Legal Notices

## Warranty

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

The information contained herein is subject to change without notice.

## Restricted Rights Legend

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

## Copyright Notice

© Copyright 2013 Hewlett-Packard Development Company, L.P.

## Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

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

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

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

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

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

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

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

UNIX® is a registered trademark of The Open Group.

# Documentation Updates

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

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

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

**<http://h20230.www2.hp.com/selfsolve/manuals>**

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

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

Or click the **New users - please register** link on the HP Passport login page.

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

# Support

Visit the HP Software Support Online web site at:

**<http://www.hp.com/go/hpsoftwaresupport>**

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

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

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

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

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

To find more information about access levels, go to:

**[http://h20230.www2.hp.com/new\\_access\\_levels.jsp](http://h20230.www2.hp.com/new_access_levels.jsp)**

# Contents

Upgrade Guide .....	1
Contents .....	5
Upgrade Overview .....	7
Phase 1: Installation of HP OO 10.00 .....	9
Prerequisites .....	9
Installing HP OO 10.00 .....	10
Data Upgraded During Installation .....	10
Upgraded LWSSO Settings .....	13
Phase 2: Upgrading to Later Versions of HP OO 10.x .....	13
Upgrading to 10.01 .....	13
Upgrading Central When it Uses User-Supplied JDBC Drivers .....	14
Upgrading When Your Company Does Not Allow Changing the Database Schema .....	15
Upgrading a Cluster .....	16
Rolling Back an Upgrade .....	16
Rolling Back When Your Company Does Not Allow Changing the Database Schema .....	17
Rolling Back a Cluster .....	18
Phase 3: Content Upgrade .....	18
Main Steps .....	18
Getting Started with Content Upgrade .....	19
Locating the Content Upgrade Utility .....	19
Preparing for Upgrade .....	19
Handling HP OO and HP OO Partner Content .....	21
Excluding Content from Upgrade .....	23
Upgrading Custom Content .....	25
Upgrading Operations Linked to Action Plugins .....	26
Hard Copy .....	26
Upgrading Content with the Content Upgrade Utility .....	28
Command Line Options .....	31
Viewing Information about the Content Upgrade .....	36

Console .....	36
Log File .....	36
Upgrade Report .....	37
Fixing Content After Upgrading .....	37
Invalid Content .....	38
Reviewing and Fixing Content After Upgrading .....	39
Deploying the Upgraded Content in Central .....	40
Known Issues and Troubleshooting .....	41
Differences in Flow Behavior After Upgrading .....	48
Phase 4: Synchronization of Content Data .....	57
Synchronizing Scheduling Data .....	57
Synchronizing Content Permissions Data .....	57
Synchronizing History .....	59
Synchronizing Studio Data .....	59

# Upgrade Overview

This document explains how to upgrade HP OO to version 10.01.

- If you have already installed HP OO 10.00, you only need to complete the steps in "[Phase 2: Upgrading to Later Versions of HP OO 10.x](#)" on page 13.
- If you are upgrading from HP OO 9.x, you will need to complete all four phases, as described in this document.

## Upgrading from HP OO 9.x

The upgrade procedure supports HP OO 9.03 and later.

Upgrading from HP OO 9.x to HP OO 10.01 consists of four main phases:

- Installation of HP OO 10.00
- Upgrading from HP OO 10.00 to 10.01
- Content upgrade
- Synchronization of content data

### Scenario: Upgrading Central, Studio, and One RAS

This section provides an overview of the major steps. For more detailed information, see the links to other topics.

#### Install HP OO 10.00

1. In Studio 9.x, check in all open flows to the shared repository. If there are multiple authors, make sure that all authors have checked in all files.
2. Install HP Operations Orchestration 10.00, as described in the *HP OO 10.00 Installation Guide*.
3. For each RAS host, install RAS 10.00.
4. For each additional Studio host, install Studio 10.00.

**Note:** Every author needs to install HP OO Studio 10.00.

For more information about installation, see "[Phase 1: Installation of HP OO 10.00](#)" on page 9.

#### Upgrade from 10.00 to 10.01

1. Expand and run the upgrade script.
2. Enter the required command line options.

For more information about upgrading from 10.00 to 10.01, see ["Phase 2: Upgrading to Later Versions of HP OO 10.x" on page 13](#).

### **Upgrade the 9.x content repository**

1. For customers who have developed iActions, copy all the iActions jars/dlls (for customer iActions) from the RASes to a local folder, which you will use in the upgrade.
2. Run the Content Upgrade Utility to create a content pack for HP OO 10.0x.
3. Open the resulting Studio project in Studio 10.01. and fix any flows with errors.
4. (Optional) To share the project with other authors, check the upgraded Studio project into the source code version control repository. Then, check out the project in all instances of Studio.
5. Load the upgraded content pack to Central 10.01.

For more information about upgrading content, see ["Phase 3: Content Upgrade" on page 18](#).

### **Synchronize content data**

1. Synchronize schedulers.
2. Synchronize flow permissions.
3. Synchronize the historical data of valid flows.

For more information about synchronizing content data, see ["Phase 4: Synchronization of Content Data" on page 57](#).

### **Enable the Central schedules**

1. Disable the Central 9.x scheduled triggers.
2. Enable the Central 10.01 scheduled triggers.

### **Uninstall the 9.x components**

1. For each RAS host, remove RAS 9.x.
2. For each Studio host, remove Studio 9.x.
3. Uninstall Central 9.x.

# Phase 1: Installation of HP OO 10.00

The first step in the upgrade process is to install HP OO 10.00.

- If you currently have HP OO 9.x, you must install HP OO 10.00 before upgrading to HP OO 10.01.
- If you have already installed HP OO 10.00, go to in ["Phase 2: Upgrading to Later Versions of HP OO 10.x" on page 13.](#)

Prerequisites .....	9
Installing HP OO 10.00 .....	10
Data Upgraded During Installation .....	10

## Prerequisites

The prerequisites for upgrading to HP OO 10.00 are the following:

- The HP OO 9.x server must be up during the upgrade.
- The HP OO 10.00 installation can be either on the HP OO 9.x machine or on a different machine.
- The HP OO 9.x database should be accessible from the machine where the installation is occurring. There should be remote access from the 10.00 machine.
- If the HP OO 10.00 installation is on the same machine as the HP OO 9.x installation, the administrator should define different ports for the HTTP/ HTTPS/ AJP connectors.


**Note:** The upgrade procedure does not modify the HP OO 9.x database and file system. HP OO version 10.00 or later requires a new schema during installation.

- All local changes should be published to the 9.x shared repository as this is the repository that is going to be upgraded.

Local changes that were not committed will not be upgraded to HP OO 10.00.

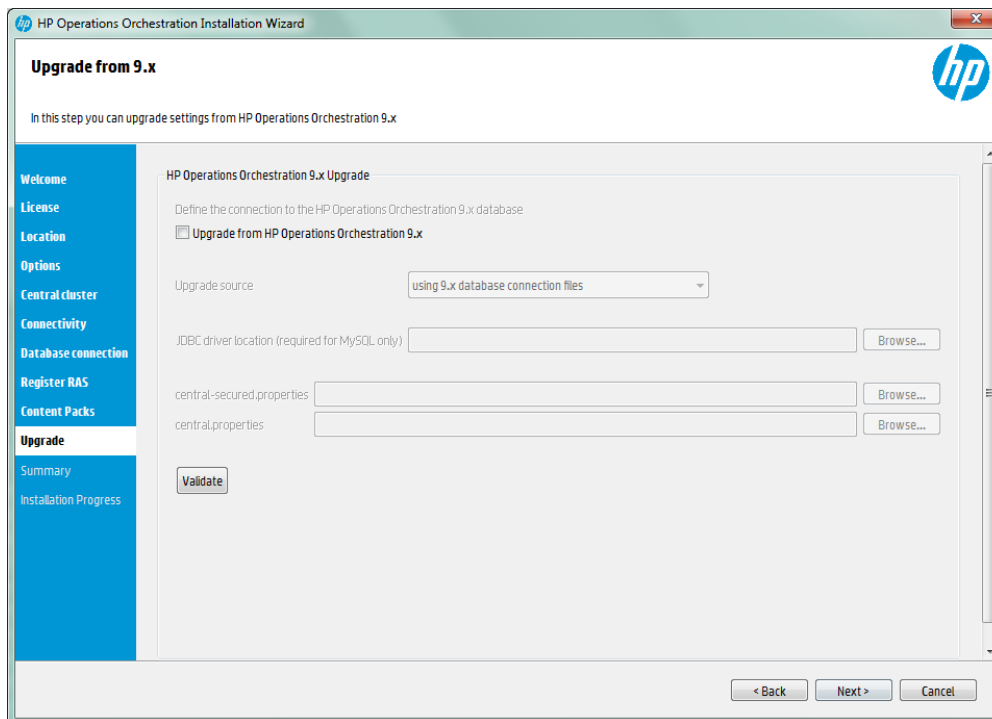
- If the database is MySQL, the administrator should provide the database driver for the upgrade procedure. HP cannot provide the MySQL driver as part of the upgrade procedure, because of legal restrictions.
- The upgrade can be done on the Windows and Linux platforms.
- Microsoft .NET Framework 4.5 or later, full installation. This is required for RAS installations and for debugging .NET operations in Studio. For more information on the supported environments see the System Requirements document.

## Installing HP OO 10.00

 During the installation process, the system administrator performs the following tasks:

- Installs HP OO 10.00
- Defines the database connection (SQL Server, Oracle, MySQL or Postgresql)
- Defines the Tomcat ports
- Starts the Tomcat server

In the Installation wizard, on the **Upgrade** page, the system administrator browses to and selects the **central-secure-properties** and the **central-properties** files. This extracts and loads the system configuration information, such as users, LDAP, LWSSO, security data, system properties, system accounts.



For more detailed information about installing HP OO 10.00, see the *HP OO Installation Guide*.

The installation process installs HP OO 10.00 in addition to the current installation of 9.x, and does not overwrite it.

## Data Upgraded During Installation

When you upgrade to HP OO 10.00, the installation process automatically extracts most of the data that you need. Some data needs to be extracted manually, as described in ["Phase 4: Synchronization of Content Data" on page 57](#).

## ***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
- Attribute of any group to use as group name
- List of LDAP contexts containing users
- List of context attribute names which 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
- An internal OO account representing a user that has search capabilities under LDAP
- Active-Directory domains (for AD only)

The LDAP credentials are encrypted in the upgrade files and in the HP OO 10.00 database.

The installation also upgrades the LWSSO parameters,

**Note:** The installation procedure does not copy the 9.x keystore, and does not extract and load the Secure Reverse Proxy parameters.

## ***Upgraded User Management Data***

If there are user groups in HP OO 9.x, the installation procedure upgrades them to HP OO 10.00 roles including their associated general permissions.

If there are enabled internal users in HP OO 9.x, the installation procedure upgrades them to HP OO 10.00 internal users.

**Note:** The installation procedure does not upgrade external users, such as LDAP users or reserved users. In HP OO 10.00, permissions are assigned to roles, so external users with permissions from HP OO 9.x cannot be upgraded.

All upgraded roles have the following general permissions by default:

- View Central
- View Portal

- Manage Tasks

## ***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.
- How LDAP referrals are handled from the primary LDAP server to other LDAP servers.

The following system settings are not upgraded, because they are not relevant in HP OO 10.00:

- 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 servers 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.

Run time environment configuration data relating to Remote Action Services (RAS) is not upgraded, because it is not relevant in HP OO 10.00.

## ***Upgraded RAS Data***

In HP OO 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 HP OO 10.00, 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 HP OO 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 HP OO 10.00, 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 HP OO 10.00, if you choose to upgrade the LWSSO settings from HP OO 9.x, these LWSSO settings will be migrated, but LWSSO will be disabled in HP OO 10.00 (even if it was previously enabled in HP OO 9.x).

## Phase 2: Upgrading to Later Versions of HP OO 10.x

After you have installed HP OO 10.00, you can upgrade to HP OO 10.01 or later.

Upgrading to 10.01 .....	13
Rolling Back an Upgrade .....	16

### Upgrading to 10.01

Upgrading to HP OO 10.01 is done using a command line script.

The relevant scripts are contained in a zip file. The scripts are:

- **apply-upgrade(.bat)** – for upgrading to HP OO 10.01
- **rollback(.bat)** – for rolling back to HP OO 10.00
- **generate-sql(.bat)** – if your company does not allow HP OO to change the database schema, this is used in addition to **apply-upgrade(.bat)** or **rollback(.bat)**

**Note:** There are two versions of each script file: with a **.bat** suffix for Windows and with no suffix for Linux.

### Prerequisite

The apply-upgrade script backs up the entire installation, so make sure that there is adequate disk space for this.

Once the upgrade is complete, you can move the backup directory (created at **<installation>/upgrade/<new-version>/backup**) elsewhere for archival. However, if you later wish to roll back the upgrade, you must move the backup directory back to the original location.

### Upgrading

To upgrade HP OO 10.00 to HP OO 10.01, you need to expand the zip file and run the **apply-upgrade(.bat)** script.

1. Expand the zip file in the root folder of the installation. This creates an **upgrade** folder containing a **<new-version>** folder (for example, **10.01**), which contains the scripts.

**Important!** Do not move the **upgrade** folder. The **apply-upgrade(.bat)** script will only work if the **upgrade** folder is directly under the main installation folder.

2. For Linux, use the following command inside the **<new-version>** folder, to change file permissions:

```
chmod 755 *
```

3. Open a command line and run the **apply-upgrade(.bat)** script.

(Optional) If required, use the following command line options:

-f, --force	Force-start the upgrade. This command upgrades the installation, without prompting.
-h, --help	Display help on the available parameters.
-n, --norestart	Do not restart Central/RAS after the upgrade.

4. Type **y** to apply the upgrade.

An **upgrade.log** file, describing the progress of the upgrade, is created in the main HP OO installation folder.

## ***Upgrading Central When it Uses User-Supplied JDBC Drivers***

During the installation of HP OO 10.00, it is possible to provide the installer with JDBC drivers (JAR files used to connect to the database) on two occasions:

- When configuring the database connection for HP OO, and using a database of the type MySQL.
- When setting up an upgrade from HP OO 9.x, which itself runs with MySQL.

The installer stores user-supplied drivers in two directories:

- **<installation>/central/lib**
- **<installation>/central/tomcat/lib**

When you run the **apply-upgrade** script, it attempts to locate these files in order to exclude them from deletion. Specifically, it looks for files whose names match either **\*mysql\*.jar** or **\*.userjdbc.jar**, and lists the search results.

If you supplied any drivers during installation, you must make sure that they are listed. Note that each driver must show up twice—once for each of the two directories mentioned above.

If any driver is missing, complete the following steps:

1. Cancel the upgrade.
2. Stop Central.
3. Locate the driver file in **<installation>/central/lib**, and change its file name extension from **.jar** to **.userjdbc.jar**.

**Note:** If your HP OO version 9.x uses MySQL, but your HP OO version 10.x does not, the MySQL driver will be missing from the installation. In this case, you must supply the file. You need this driver in order to import the data from the HP OO 9.x database.

Copy the missing file to **<installation>/central/lib** and make sure its name either contains **mysql** or has the extension **.userjdbc.jar**.

4. Repeat the previous step for **<installation>/central/tomcat/lib**.
5. Run **apply-upgrade** again, and make sure you see the driver file listed under both directories.

If you accidentally let **apply-upgrade** delete your driver, you can manually place the driver in the two directories, this time with a **.userjdbc.jar** extension, and then re-run **apply-upgrade**.

**Note:** If you are installing a cluster, the instructions above apply to all Central nodes.

## ***Upgrading When Your Company Does Not Allow Changing the Database Schema***

If your company does not allow the HP OO application to change the database schema, the upgrade procedure is different. You will need to use the **generate-sql(.bat)** script, which is also included in the upgrade zip file.

The **generate-sql(.bat)** script is used to generate the **upgrade.sql** file in the unzipped upgrade folder. This file contains the SQL for applying all of the upgrade's database changes.

1. Expand the zip file in the root folder of the installation. This creates an **upgrade** folder containing a **<new-version>** folder (for example, **10.01**), which contains the scripts.

**Important!** Do not move the **upgrade** folder. The **apply-upgrade(.bat)** script will only work if the **upgrade** folder is directly under the main installation folder.

2. For Linux, use the following command inside the **<new-version>** folder, to change file

permissions:

```
chmod 755 *
```

3. Open a command line and run the **generate-sql(.bat)** script.

The command line options for **generate-sql(.bat)** are:

-h, --help	Display help on the available parameters.
-r, --rollback	Generate the SQL for rollback. Use only after the database has been upgraded.

The **upgrade.sql** file is created under **upgrade/<version>**.

4. Before applying the database changes, upgrade HP OO using **apply-upgrade(.bat)** with the **-n** option.

**Note:** The order of the steps below is critical. If you don't complete the steps in the correct order (for example, if you do Step 5 before Step 4), then everything will seem to work, but rollback will not be possible.

The **-n** option prevents the upgrade script from restarting Central/RAS after the upgrade.

For example:

```
apply-upgrade -n
```

5. Apply the database changes by running **upgrade.sql** against the database with the required credentials.
6. Start Central/RAS.

## Upgrading a Cluster

In a cluster setup, stop all Central/RAS instances manually, and only then apply the upgrade to all of them.

If you have upgraded to version 10.01 and have Central (10.00) installed on a cluster, you must upgrade Central to 10.01 as the database version is now 10.01. If you do not upgrade Central it will fail to start.

## Rolling Back an Upgrade

You can roll back an upgrade using the rollback script. This restores the installation to the state it was before the upgrade, but without losing database data.

A component (Central, RAS, Studio) will only be rolled back if:

- The component is backed up under **<installation>/upgrade/<new-version>/backup/<component>**.
- The installed version is identical to the upgrade script's **<new-version>**.

If Central is being rolled back, the script rolls back the database schema changes and preserves data that was added after the upgrade. However, in some cases, data may still be lost due to the schema changes.

Note that any changes that were made in the file system since the upgrade will be lost.

1. Open a command line.
2. Run the **rollback(.bat)** script, which is included in the upgrade zip file.

(Optional) If required, use the following command line options:

-f, --force	Force-start the rollback. This command rolls back the installation, without prompting.
-h, --help	Display help on the available parameters.
-n, --norestart	Do not restart Central/RAS after the rollback.

The rollback script uses the same **upgrade.log** file as the upgrade script.

## ***Rolling Back When Your Company Does Not Allow Changing the Database Schema***

If your company does not allow the HP OO application to change the database schema, the rollback procedure is different. You will need to use the **generate-sql(.bat)** script with the **-r** option, which generates a **rollback.sql** file in the unzipped upgrade folder.

1. Open a command line and run the **generate-sql(.bat)** script with the **-r** option.

The command line options for **generate-sql(.bat)** are:

-h, --help	Display help on the available parameters.
-r, --rollback	Generate the SQL for rollback. Use only after the database has been upgraded.

For example:

```
generate-sql -r
```

The **rollback.sql** file is created in the unzipped upgrade<version> folder.

2. Stop Central/RAS.

3. Apply the database changes by running **rollback.sql** against the database with the required credentials.
4. Roll back HP OO 10.01 using **rollback(.bat)**.

Central/RAS restarts automatically after the rollback.

## ***Rolling Back a Cluster***

In a cluster setup, it is recommended to stop all Central/RAS instances manually before rolling back.

## **Phase 3: Content Upgrade**

The third major step in the upgrade procedure is to upgrade the HP OO 9.x content to content in HP OO 10.0x format.

The Content Upgrade Utility is a tool that extracts specified content from an existing HP OO 9.x repository, and upgrades the extracted content into a content pack that can be deployed in HP OO 10.00 and later.

**Note:** A **content pack** is a collection of operations, flows, actions (Java-based or .Net based), and configuration items (such as selection lists, domain terms, and so on). For more information about content packs, see the *HP OO Studio Authoring Guide*.

After you upgrade content from HP OO 9.x, the resulting Studio project will include repository content (operations, flows, configuration items, and so on), IAction jars, dlls, third party dependencies, and resource bundle (localization) files. This project can be opened in OO Studio and exported to a content pack that can be deployed in OO Central.

This step may be performed in a iterative manner. You can upgrade different parts of your content separately and create multiple HP OO 10.0x content packs.

## **Main Steps**

The main steps in content upgrade are the following:

1. Prepare for upgrade. See ["Getting Started with Content Upgrade" on the facing page](#).
2. Run the Content Upgrade Utility. See ["Upgrading Content with the Content Upgrade Utility" on page 28](#).
3. Open the report to check which content packs your content is dependent upon. See ["Fixing Content After Upgrading" on page 37](#).
4. Open HP OO Studio, and import the content packs that your content is dependent upon.

5. Import the two projects that were created by the Content Upgrade Utility, and go through them to make sure that all your important content is valid.
6. Fix whatever needs fixing in the invalid project, and move anything that was fixed to the valid project.
7. Create a content pack from the valid project.
8. Deploy to Central the content packs that your content is dependent upon.
9. Deploy the content pack with the upgraded, valid content.

**Important!** Make sure that you deploy the content pack that you created from Studio, from the project with the valid content. We do not recommend directly deploying the content pack that was created by the Content Upgrade Utility.

## Getting Started with Content Upgrade

### *Locating the Content Upgrade Utility*

The Content Upgrade Utility 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
- **<OO Installation Directory>/Studio/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
- **<OO Installation Directory>/studio/cmu/** for Linux

### *Preparing for Upgrade*

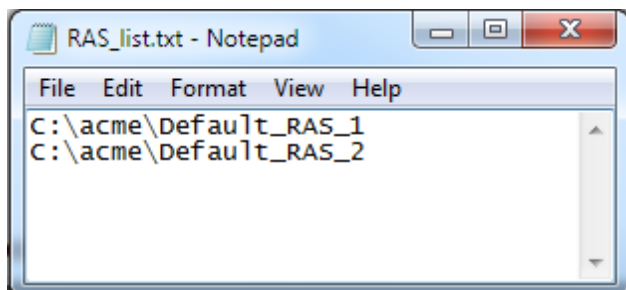


Before upgrading content, the flow author should prepare the content for upgrade.

- Make sure that the content does not include invalid data. Invalid data will not be upgraded.
- Sleep scripts are not supported as scriptlets in HP OO 10.0x. For example, under the **Scriptlet** tab or in the **Results** filter. It is recommended to rewrite these scripts in JavaScript before upgrading.
- Multi-instance steps work differently in HP OO 10.0x. For more information about how to prepare MI steps for upgrade, see ["Known Issues and Troubleshooting" on page 41](#).
- If you have multiple custom operations that are linked to the same iAction, consider changing these to soft copies before upgrading. For more information, see ["Upgrading Operations Linked to Action Plugins" on page 26](#).

 Before upgrading, the op admin should prepare for upgrade.

- It is recommended to consolidate all content into one repository, to be used for the upgrade. Make sure that all items are checked in.
- Export your HP OO 9.x repository from Studio 9.x:
  - a. Right-click **Repository**, and select **Export as New Repository**.
  - b. Select a name and location for the repository.
  - c. In the Select Export Options page, make sure that all the check boxes are cleared (not selected) apart from the **Give EVERYBODY group full access to exported items for which you have write access** and **Encrypt system properties** check boxes.
- Make sure you know the path to the exported repository, which is to be upgraded, and where you want to store the content pack result after the upgrade.
- If your repository has custom RAS iActions (Java or .Net), copy the folder **\$(ICONCLUDE\_HOME)\RAS\Java\Default** to an accessible location on the file system. If you have multiple RASes in your environment, copy this folder, for each RAS, to the accessible location. Create a text file with the full paths to the RAS folders, with each path on one line. For example:



For more information, see ["Upgrading Custom Content" on page 25](#).

- Before starting the upgrade process, make sure you have a copy of the HP OO 10.0x base

content pack, and any other HP out-of-the-box content packs that you need. These are provided in the product DVD or by downloading from HPLN.

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

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

**Note:** If you have modified out-of-the-box content, those changes will not be reflected in the resulting content packs.

## Handling HP OO and HP OO Partner Content

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

During the content upgrade process, the Content Upgrade Utility excludes HP OO out-of-the-box content from the upgrade. The Content Upgrade Utility also recognizes the HP partner content packs listed below and excludes their content from the upload, making sure that only your custom content is upgraded.

Content Pack Name	Availability/Location
HP out-of-the-box content packs	
Base content pack	Available on the HP OO 10.0x DVD and on HPLN under <a href="https://hpln.hp.com/group/operations-orchestration-content-packs">https://hpln.hp.com/group/operations-orchestration-content-packs</a> .  See the <i>HP OO 10.0x Release Notes</i> for more information on the these content packs.
Business Applications content pack	
Cloud content pack	
Databases content pack	
HP 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	

Content Pack Name	Availability/Location
Server Automation content pack	This content pack is not included in the HP OO 10.0x DVD. Please check availability in HPLN, under: <a href="https://hpln.hp.com/group/operations-orchestration-content-hp-server-automation">https://hpln.hp.com/group/operations-orchestration-content-hp-server-automation</a>
Storage Essentials content pack	This content pack is not included in the HP OO 10.0x DVD. Please check availability in HPLN, under: <a href="https://hpln.hp.com/group/operations-orchestration-content-hp-storage-essentials">https://hpln.hp.com/group/operations-orchestration-content-hp-storage-essentials</a>
NonStop content pack	This content pack is not included in the HP OO 10.0x DVD. Please check availability in HPLN, under: <a href="https://hpln.hp.com/group/operations-orchestration-content-hp-integrity-nonstop-computing">https://hpln.hp.com/group/operations-orchestration-content-hp-integrity-nonstop-computing</a>
Cloud Services Automation content pack	This content pack is not included in the HP OO 10.0x DVD. Please check availability in HPLN, under: <a href="https://hpln.hp.com/group/cloud-service-automation">https://hpln.hp.com/group/cloud-service-automation</a>
Service Manager content pack	This content pack is not included in the HP OO 10.0x DVD. Please check availability in HPLN, under: <a href="https://hpln.hp.com/group/operations-orchestration-content-hp-service-manager">https://hpln.hp.com/group/operations-orchestration-content-hp-service-manager</a>
<b>HP Community content packs</b>	

Content Pack Name	Availability/Location
BMC content packs (Atrium, Patrol, Remedy)	HP community content packs are available for download on HPLN, under: <a href="https://hpln.hp.com/group/community-contributed-content-operations-orchestration">https://hpln.hp.com/group/community-contributed-content-operations-orchestration</a>
Cisco content pack	
Citrix Presentation Server content pack	
Computer Associates content pack	
F5 content pack	
HP 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	

If you use content that is not listed above, refer to the HP OO HPLN community at <https://hpln.hp.com/node/21/contentfiles> in order to download the Content Upgrade Utility plugin along with the content pack itself.

## ***Excluding Content from Upgrade***

If you have partner content that does not appear in the list above, you will need to tell the Content Upgrade Utility to exclude this content. Follow this procedure to get the Content Upgrade Utility to recognize partner content and exclude it from the upgrade process.

**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.

After the content upgrade is done, deploy the partner content pack before deploying the upgraded content pack.

### **Working with the Content Upgrade Utility on a Central Installation**

1. Go to HPLN and locate the relevant content provider.

**Note:** Try starting at the content catalog, under <https://hpln.hp.com/node/21/contentfiles>.

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. Run the Content Upgrade Utility, now that it recognizes the partner content.

## Working with the Content Upgrade Utility on a Studio Installation

1. Go to HPLN and locate the relevant content provider.

**Note:** Try starting at the content catalog under <https://hpln.hp.com/node/21/contentfiles>.

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>\studio\cmu\exclusions**
  - Linux: **<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:
  - Windows: **<OO Installation Directory>\studio\cmu\plugin-mappings**
  - Linux: **<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:
  - Windows: **<OO Installation Directory>\studio\cmu\plugin2op\_mappings**
  - Linux: **<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. Run the Content Upgrade Utility, now that it recognizes the partner content.

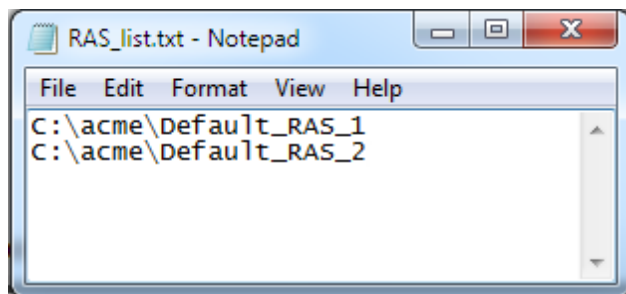
## Upgrading Custom Content

This section 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 *Extension 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 `$(CONCLUDE_HOME)\RAS\Java\Default` for each of your RASes.

For example:



3. Add this file as input to the Content Upgrade Utility, using the argument `-rd, --rases-dirs`.

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-dirs c:/acme/RAS_list.txt
```

## Upgrading Operations Linked to Action Plugins

### Hard Copy

In versions of HP OO prior to 10.00, 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 HP OO 10.0x, 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 HP OO 10.0x, 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.

## Upgrading Custom Operations as Soft Copies

If you have multiple custom operations that are linked to the same action plugin, you might prefer to upgrade these as soft copies. To do so, you can add 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.

Welcome x My SSH Command - Properties x

Name: My SSH Command

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

Assign Categories:

Inputs Outputs Responses Description Scriptlet

Inputs Summary

RAS Operation fields

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

Archive: JRASRepository.jar

RAS: /Configuration/Remote Action Services/RAS\_Operator\_Path

Override RAS: \${overrideJRAS}

## Upgrading Content with the Content Upgrade Utility

The administrator runs the Content Upgrade Utility to upgrade the content and supplies the path to the exported repository.

**Tip:** In order to avoid errors, we recommend consolidating all content into one repository, to be used for the upgrade.

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.

For example, in Windows:

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

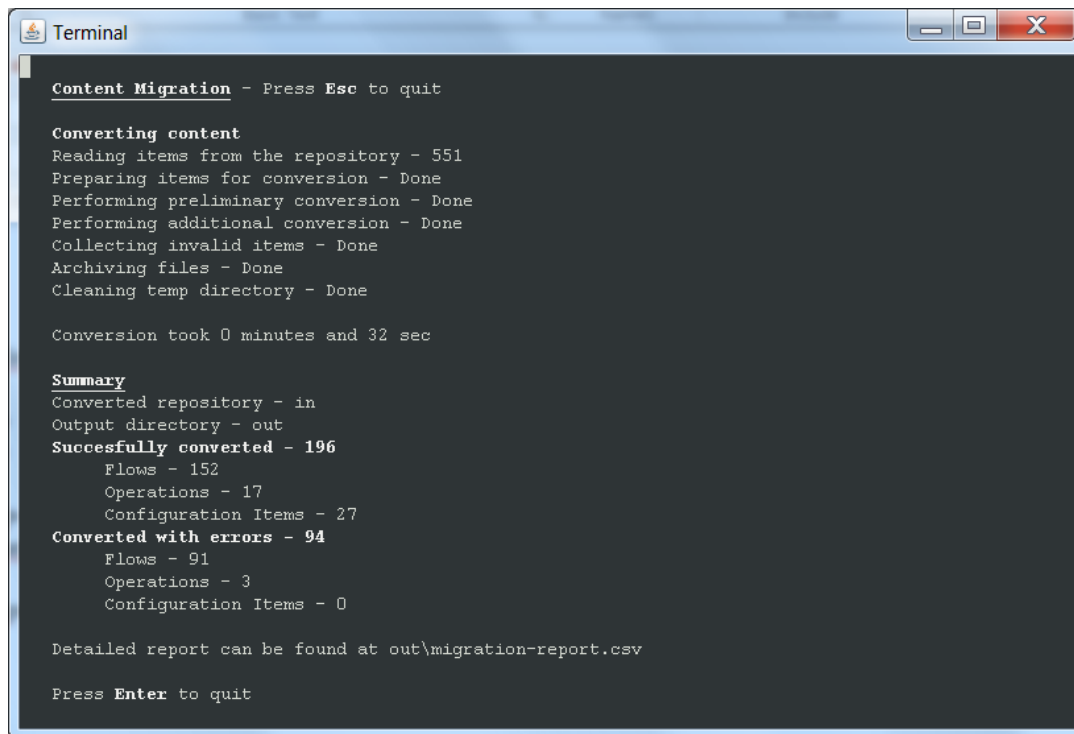
For example, in Linux:

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

In the example above:

- `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.



```

Terminal
Content Migration - Press Esc to quit

Converting content
Reading items from the repository - 551
Preparing items for conversion - Done
Performing preliminary conversion - Done
Performing additional conversion - Done
Collecting invalid items - Done
Archiving files - Done
Cleaning temp directory - Done

Conversion took 0 minutes and 32 sec

Summary
Converted repository - in
Output directory - out
Successfully converted - 196
  Flows - 152
  Operations - 17
  Configuration Items - 27
Converted with errors - 94
  Flows - 91
  Operations - 3
  Configuration Items - 0

Detailed report can be found at out\migration-report.csv

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.

For more detail about upgrade options and more examples of how to enter the upgrade options under different circumstances, see ["Command Line Options" on the facing page](#).

When the upgrade is complete, an HP OO 10.0x 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. To see more details about the upgrade, view the **content-upgrade-report.html** file.
4. 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.

After upgrading, we recommend opening the upgraded projects in Studio, checking if you need to fix any invalid content, and then exporting from Studio before even trying to deploy the content pack. Deploy the content pack that you created from the valid project in Studio and not the one created by the Content Upgrade Utility. See ["Fixing Content After Upgrading" on page 37](#).

## Command Line Options

While running the Content Upgrade Utility, you need to define the details of your content upgrade—which repository to upgrade, what to name it, where to store it, and other details.

This tool supports the following arguments:

### **--desc**

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

For example:

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

The default value is "Upgraded content pack".

### **--help, -h**

Displays help information about the different command line options.

### **--inclusions-dir**

Use the `--inclusions-dir` argument to point to the inclusions directory. This directory can contain one or more files where multiple paths are written, describing the locations of files that you want to include in the upgrade.

For example, you have a file called **MyInclusions.txt**, which includes the following text:

```
\Library\My Ops Flows
\Library\Utility Operations
\Configuration
```

You can include relative paths in the inclusion files. Paths within such files are relative to the repository path (`--repo`).

**Note:** Relative paths used with the `--inclusions-dir` parameter should use only the forward slash character if the Content Upgrade Utility is running on a Linux platform. For Windows platforms, both forward slash and backslash are accepted.

If the directory containing the inclusions file is **c:\Inclusions**, the command is:

```
--inclusions-dir c:\Inclusions
```

## **--locale**

Use the `--locale` argument to set the locale of this content pack, for languages other than English. When the locale is specified, a properties option for this locale is generated, called **cp\_<locale>.properties**.

The default value is "en\_US".

## **--name**

Use the `--name` argument 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

The default value is `Customer`.

## **-nine,--nine-cp-version**

Use the `-nine, --nine-cp-version` argument to specify the 9.x content pack version that you have installed.

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, a duplicate is created of the original item, and any use of the original item is replaced with the version with your changes.

For example, the **Character Sets** 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 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 10
- Content Pack 11
- Content Pack 9
- Content Pack 8
- Content Pack 7

### ***--nocp***

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

### ***--no-user-interaction***

This option disables the console.

### ***--output***

Use `--output` 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.

The default value is the current directory.

### ***--publisher***

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

The default value is "Customer" .

### ***-rd,--rases-dirs***

Use `-rd, --rases-dirs` to upgrade custom IActions. Indicate the path to a file containing on each line the path to a local folder that contains the content of `${ICONCLUDE_HOME}\RAS\Java\Default` for each of your RASes.

## ***--repo***

Use `--repo` to specify the folder containing the HP OO 9.x repository. Point to the root directory of the HP OO 9.x project.

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

## ***--temp-dir***

Use `--temp-dir` to set the temp directory, overriding the default temp directory.

The default value is the value from `java.io.tmpdir`.

## ***--version***

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

The default value is "1.0.0".

## ***Example 1 (Windows)***

In this example:

- **9xContent** is the name given to the new content pack
- **MyCompany** is the name of the company
- **C:\temp\repo** is the location of the 9.x repository
- **D:\NewRepositories** is the location where the content pack will be stored.

For the options that have not been specified, the upgrade tool uses the default values.

```
>upgrade-content.bat --name 9xContent --publisher MyCompany --repo C:\temp\repo --output D:\NewRepositories
```

## ***Example 2 (Windows)***

In this example:

- **9xContent** is the name given to the new content pack
- **MyCompany** is the name of the company
- **C:\temp\repo** is the location of the 9.x repository

- **D:\NewRepositories** is the location where the content pack will be stored
- **C:\temp\inclusions** points to the C:\temp\inclusions directory, which contains a file that specifies the location of HP OO content that will be converted into a content pack.

For example, the directory C:\temp\inclusions contains a text file with the following lines:

```
\Library\My Ops Flows  
\Configuration
```

For the options that have not been specified, the upgrade tool uses the default values.

```
>upgrade-content.bat --name 9xContent --publisher MyCompany --repo C:\temp\r  
epo --output  
D:\NewRepositories --inclusions-dir C:\temp\inclusions
```

## Viewing Information about the Content Upgrade

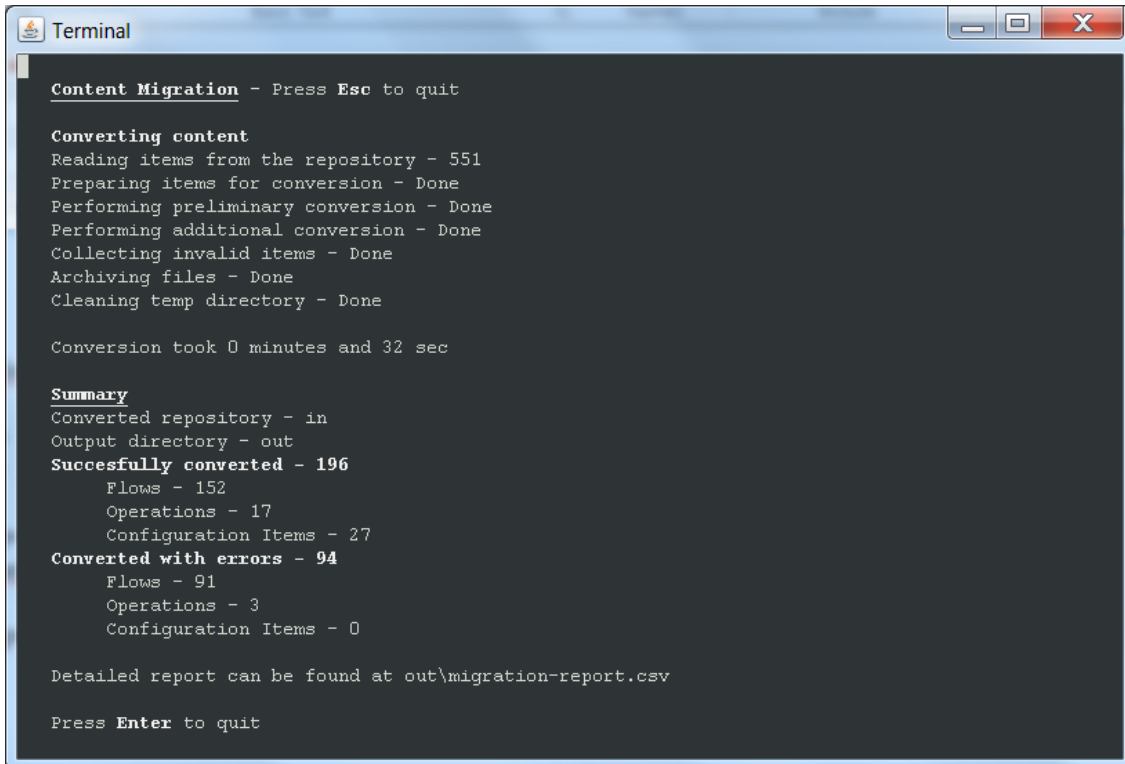
There are a number of ways to view the progress of the content upgrade.

### Console

While the content upgrade is running, the Content Upgrade Utility console displays messages, including:

- Warnings and errors
- General messages
- Progress indications

When the upgrade is complete, the console displays a summary of the content that was converted.



```
Terminal

Content Migration - Press Esc to quit

Converting content
Reading items from the repository - 551
Preparing items for conversion - Done
Performing preliminary conversion - Done
Performing additional conversion - Done
Collecting invalid items - Done
Archiving files - Done
Cleaning temp directory - Done

Conversion took 0 minutes and 32 sec

Summary
Converted repository - in
Output directory - out
Successfully converted - 196
  Flows - 152
  Operations - 17
  Configuration Items - 27
Converted with errors - 94
  Flows - 91
  Operations - 3
  Configuration Items - 0

Detailed report can be found at out\migration-report.csv

Press Enter to quit
```

### Log File

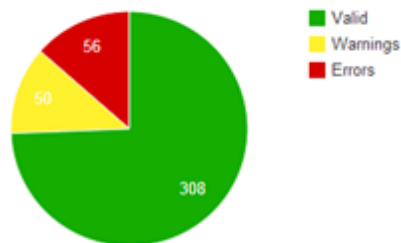
After the upgrade has run, you can view the **content-upgrade.log** file in the **output** folder. This file includes everything that was written to the console as well as debug messages.

## Upgrade Report

After the upgrade has run, you can view the **content-upgrade-report.html** file in the **Output \ Upgrade** folder.

### Operations Orchestration - Content Upgrade Report

#### Converted Objects - 364



#### 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](#)

The report includes:

- Number of converted objects.
- Number of objects that were not upgraded.
- Details of the objects that require action, and suggestions about how to resolve each type of problem.
- Details about objects that behave differently after upgrade.

For more information about fixing objects, see ["Fixing Content After Upgrading"](#) below.

For more information about objects that behave differently after upgrade, see ["Differences in Flow Behavior After Upgrading"](#) on page 48.

## Fixing Content After Upgrading

After the content upgrade has run, you can view the two HP OO 10.0x content packs that were generated and stored in the output location that you specified:

- **Valid content:** All the content that can be deployed and run is stored in a content pack for 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.

- If you have not defined a name using the `--name` argument, the project containing the valid content is called **Customer**.
- **Invalid content:** Content that cannot be deployed and run is stored in a content pack 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**.

## Invalid Content

Content that is not supported by HP OO 10.0x is excluded from the content pack, and stored in a separate HP OO 10.0x project. The project containing the invalid content can still be opened in HP OO 10.0x Studio.

The following cases are excluded from the content pack:

- Flows and operations that were also invalid in versions earlier than 10.00.
- Flows and steps that use Sleep scriptlets.
- Flows with a multi-instance step inside a parallel lane.
- Operations that point to missing plugins.
- 9.x integration plugins that are not backward compatible or not supported in 10.0x
- Flows and operations with missing dependencies. This includes unresolved domain terms, missing operations used in a step flow, missing subflows, missing system evaluators, missing system filters, missing system properties, missing system accounts, or missing selection lists.
- All flows dependent on content mentioned above.
- Flows containing steps with user and password credential inputs, where in the original operation, they are required but not assigned.
- Content with missing UUIDs, such as out-of-the-box content that is missing the `RAS_Operator_Path`, system account or system property.

## Reviewing and Fixing Content After Upgrading

After the project has been upgraded, it is recommended to check the upgrade report to see if any content was not upgraded successfully. You can then open the project containing the invalid content in Studio 10.0x, fix the content, and create a new content pack.

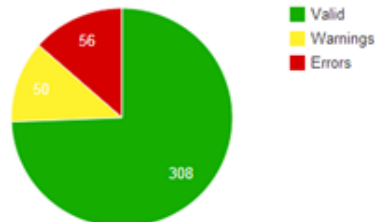
- In some cases, the content that was not upgraded is content that is not yet supported by HP OO 10.0x, and you may wish to keep it as it, while waiting for the features to be supported in later versions.
- In other cases, you may prefer to modify the content so that it can be used in HP OO 10.0x. You may prefer to save a copy of the original content and create a duplicate that is modified.

After the upgrade:

1. In your file browser, locate the upgrade report: **Output\content-upgrade-report.html**.
2. Open the report in a Web browser.

### Operations Orchestration - Content Upgrade Report

#### Converted Objects - 364



#### 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](#)

3. In the **Requires Action** section, click **Show/Hide** to see a table containing all the items that were not upgraded and suggestions about how to resolve each type of problem.

**Requires Action: 52 (Flows: 48, Operations: 4, 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 that need to be fixed. To fix the items, open the Invalid Content project in HP OO Studio 10.

[Show/Hide](#)

**Referencing a missing item**

These flows and operations have a reference to a missing item (flow, operation, selection list, domain term, system filter, scriptlet or RAS). To fix:

1. Open each original flow or operation in Studio 9.x.
2. Check whether there are missing references (these appear in red).
3. Fix the references so that they refer to existing items.
4. Export the repository again, and then upgrade the fixed repository to HP OO 10.00.

Name	Path
SFTP Get Attributes MissingRas Op	Library\My Ops Flows\CMU Repository\Missing Dependencies\SFTP Get Attributes MissingRas Op.xml
Missing SelectionList Two Levels	Library\My Ops Flows\CMU Repository\Missing Dependencies\Missing SelectionList Two Levels.xml
Missing SelectionList Three Levels 2	Library\My Ops Flows\CMU Repository\Missing Dependencies\Missing SelectionList Three Levels 2.xml
Missing SelectionListThree Levels	Library\My Ops Flows\CMU Repository\Missing Dependencies\Missing SelectionListThree Levels.xml
Missing SelectionList	Library\My Ops Flows\CMU Repository\Missing Dependencies\Missing SelectionList.xml
Missing Ras Four Levels	Library\My Ops Flows\CMU Repository\Missing Dependencies\Missing Ras Four Levels.xml
Missing Ras Three Levels	Library\My Ops Flows\CMU Repository\Missing Dependencies\Missing Ras Three Levels.xml
Missing Ras Two Levels	Library\My Ops Flows\CMU Repository\Missing Dependencies\Missing Ras Two Levels.xml
MissingSubflow	Library\My Ops Flows\CMU Repository\Missing Dependencies\MissingSubflow.xml
Missing Domain Term	Library\My Ops Flows\CMU Repository\Missing Dependencies\Missing Domain Term.xml
Missing Op	Library\My Ops Flows\CMU Repository\Missing Dependencies\Missing Op.xml

4. In Studio 10.0x, import the HP OO 10.0x out-of-the-box Base Content Pack and all content packs that are mentioned in the report.
5. Import the project files containing the upgraded content, both valid and invalid.
6. Open the project containing the invalid content in HP OO Studio 10.0x, and fix the problems.

**Note:** For information about how to fix specific upgrade problems, see ["Known Issues and Troubleshooting" on the facing page](#).

7. 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.
8. After making adjustments to the content in Studio 10.0x, repackage the fixed content as a content pack, for deployment. For more information, see the *HP OO 10.0x Studio Authoring Guide*.

## Deploying the Upgraded Content in Central

After you have exported the valid project into a content pack from Studio 10.0x, you can deploy this content in Central 10.0x.

We do not recommend directly deploying the content pack that was created by the Content Upgrade Utility. Make sure to use the content pack that you created in Studio, by exporting the valid project.

**Important!** We recommend opening the project in Studio and then exporting from Studio before every trying to deploy the content pack.

1. Deploy the out-of-the-box Base Content Pack that is shipped with HP OO 10.0x, and deploy all the content packs that your content is dependent upon to run.

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

2. Deploy the content pack containing the upgraded content.

For information about deploying content in Central, see the *HP OO 10.0x Central User Guide*.

## Known Issues and Troubleshooting

Because of the changes in HP OO 10.0x, 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 HP OO, and in the meantime, to create a new flow that works differently.

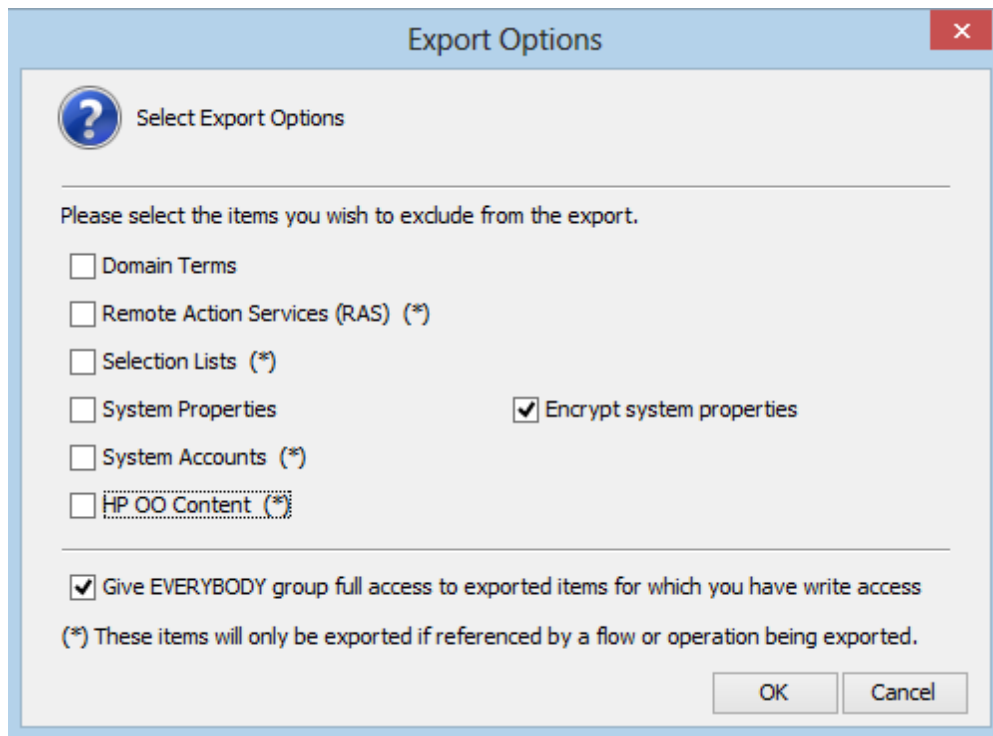
Content that is not supported by HP OO 10.0x is stored in a separate project after upgrade, and is not included in the content pack. For more information about what kind of content is not supported, see the *HP OO 10.00 Base Content Pack 1.0 Release Notes*. For more information about features that are not supported in HP OO 10.0x, see *Known Issues and Limitations* in the *HP OO 10.0x Release Notes*.

## 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 HP OO 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 dialog box, make sure that only the **Encrypt system properties** and **Give EVERYBODY group full access to exported items for which you have write access** check boxes are selected, and that the other check boxes are cleared.



4. Run the Content Upgrade Utility on that repository.

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

## ***Backslash Character in Linux***

**Issue:** If you use the backslash character in the `--inclusions-dir` parameter, files will not work in Linux.

Relative paths used with the `--inclusions-dir` parameter should use only the forward slash character if the Content Upgrade Utility runs on Linux platforms. For Windows platforms, both forward slash and backslash are accepted.

**Solution:** Do not use the backslash character in the `--inclusions-dir` parameter, in Linux.

## ***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. See ["Handling HP OO and HP OO Partner Content " on page 21](#) for more details about the content packs that the Content Upgrade Utility recognizes, and how to handle other content.

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 HP OO 10.0x. 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.

## Multi-instance Steps

In HP OO 10.0x, 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 HP OO 9.x, a multi-instance step was a single step, with a single set of properties. In HP OO 10.0x, 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

<b>Result - From key name</b>	Same as original	Same as the result name
<b>Result - Assignment</b>	Overwrite	Same as original
<b>Result - Filter</b>	Same as original	None
<b>Scriptlet</b>	Same as original	Upgrade scriptlet - will merge all the global variables of the inner steps.

## ***Multi-instance Steps in a Parallel Lane***

**Issue:** A multi-instance step in a parallel lane will not work after upgrade.

**Solution:** Replace the multi-instance step in the parallel lane with a subflow step in the parallel lane.

## ***Multi-instance Steps with Transition to a Parallel Step or Another Multi-instance Step***

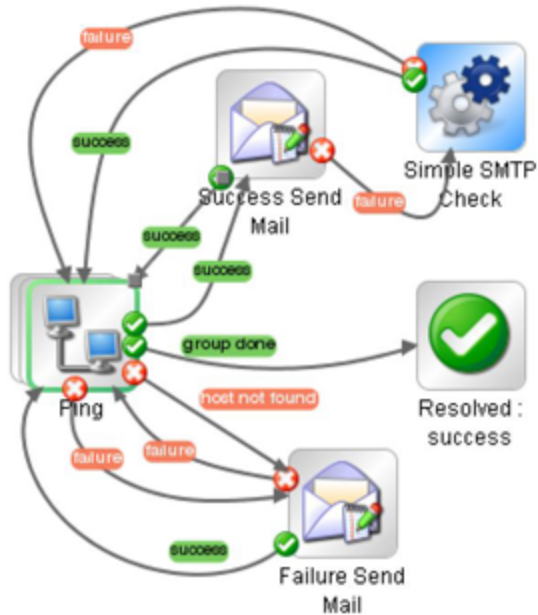
**Issue:** A multi-instance step that has a transition to a parallel lane step or to another multi-instance step (rather than via a **Group Done** transition) will not work after upgrade.

**Solution:** Replace the transition with a **Group Done** transition at the end of the multi-instance step.

## ***Multi-instance Steps with Results Inside the Loop***

**Issue:** When you upgrade a flow with a multi-instance loop, if one of the steps in the loop (not the multi-instance step) assigns results to flow variables and/or flow output fields, then these results will not be available outside the multi-instance lane.

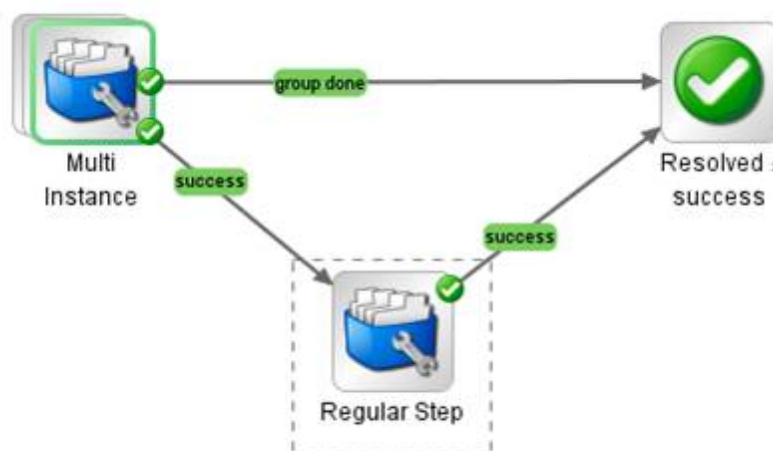
For example, see the flow below. If you added results to **Success Send Mail**, **Simple SMTP Check** or **Failure Send Mail**, the flow variables created in those results will not be available after the **Group done** transition.



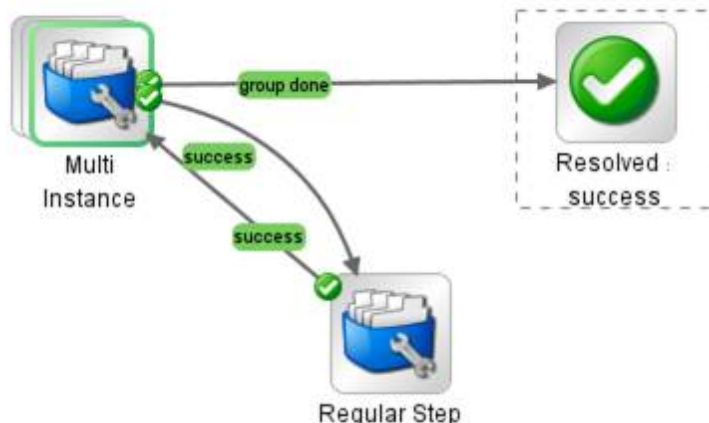
**Solution:** To make the results available outside the multi-instance lane, reassign them in the multi-instance step properties after upgrade.

## ***Flows where Response Destinations are Not Connected Back to Multi-instance Steps***

**Issue:** There are problems with upgrading a flow with a multi-instance step, where a regular step goes directly to the end step, and not back to the multi-instance step. This is considered bad practice in flow design.

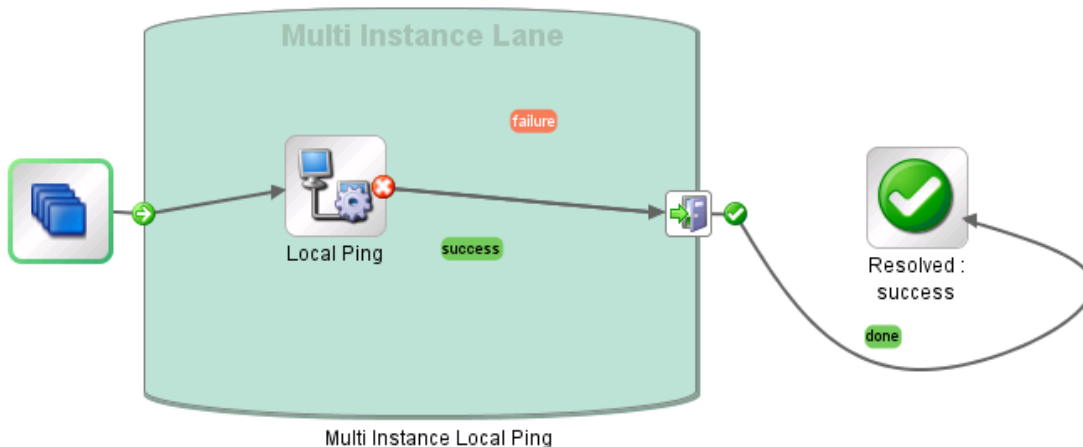
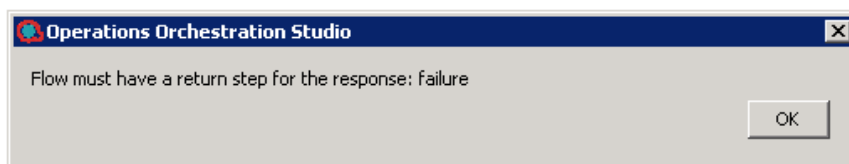


**Solution:** Fix the flow so that the regular step loops back to the multi-instance step, and only from there goes on to the end step.



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

**Issue:** After content upgrade to HP OO 10.0x, a flow with a multi-instance step goes to the invalid project. After fixing the flow in Studio 10.0x, it is not possible to save the flow.



**Solution:** Delete all return steps and replace them with new return steps in Studio 10.0x.

## Only One AD Configuration is Upgraded

**Issue:** During the content upgrade process, if there are multiple AD configurations, the Content Upgrade Utility only upgrades one.

**Solution:** The other AD configurations need to be added to HP OO 10.0x manually.

## ***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 HP OO 9.x but are not supported in HP OO 10.0x. 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.

## Resume Asynchronously Has Inverted Meaning

**Issue:** When a flow with asynchronous resume is upgraded, the asynchronous resume option doesn't work as expected. When this value is set to be true, the flow runs synchronously, instead of asynchronously. In other words, the flow run is resumed and the operation waits for the resumed run to finish and collect its results, rather than the expected behavior that the flow run is resumed but the operation does not wait for the resumed run to finish.

## Differences in Flow Behavior After Upgrading

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

## 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 HP OO 10.0x, 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 HP OO 9.x, where there are two inputs, **count** and **reset**, and fields to specify the starting count and increment.

The screenshot shows the 'Inputs Summary' tab in the HP OO 9.x interface. It displays the configuration for a loop operation. The 'Counter fields' section includes input fields for 'from' (set to 1), 'to' (set to \${count}), 'incrementBy' (set to 1), and 'reset' (set to \${reset}). Below this, the 'Inputs' section contains a table with two inputs: 'count' and 'reset'.

Input	Required	Type	Template
count	<input checked="" type="checkbox"/>	Single Value	Prompt User
reset	<input type="checkbox"/>	Single Value	Value: false

Field values were removed in HP OO 10.0x. 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**.

The screenshot shows the 'Inputs Summary' window for an operation named 'Counter'. The operation fields are:

- Group Id: com.hp.oo
- Artifact Id: oo-base-plugin
- Version: 1.0-SNAPSHOT
- Action Name: Counter
- Group Alias: (icon)
- Override Group: (empty field)

Below the fields is a table of inputs:

Input	Required	Type	Template
count	<input checked="" type="checkbox"/>	Single Value	Prompt User
reset	<input type="checkbox"/>	Single Value	Value: false
to	<input type="checkbox"/>	Single Value	Value: \${count}
from	<input type="checkbox"/>	Single Value	Value: 1
incrementBy	<input type="checkbox"/>	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 HP OO 9.x, there are only two inputs, **count** and **reset**.

The screenshot shows the 'Inspector' window for a step named 'Loop'. The 'Inputs Summary' section shows two inputs:

Assign To Input	Required	Type	From
count	<input checked="" type="checkbox"/>	Single Value	Prompt User
reset	<input type="checkbox"/>	Single Value	Value: false

After upgrade to HP OO 10.0x, there are now five inputs in the step, corresponding to the upgraded operation, which has five inputs.

Inspector

Step Name:

Inputs Results Display Description Advanced Scriptlet

**Inputs Summary**

Add Input Remove Input ↑ ↓

Assign To Input	Required	Type	From
count	<input checked="" type="checkbox"/>	Single Value	Prompt User
reset	<input type="checkbox"/>	Single Value	Value: false
to	<input type="checkbox"/>	Single Value	Value: \${count}
from	<input type="checkbox"/>	Single Value	Value: 1
incrementBy	<input type="checkbox"/>	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 10.0x, there is a new **Categories** folder under **Configuration** folder.

Public Repository - Def...

Library

- Configuration
  - Domain Terms
    - Action
    - Alert
    - Application
    - Category**
    - Change
    - CI Minor Type
    - CI Type
    - Configuration Item
    - counter

My Changes/Checkouts

Welcome x **Category** x

Configuration > Domain Terms > Category

Name:

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

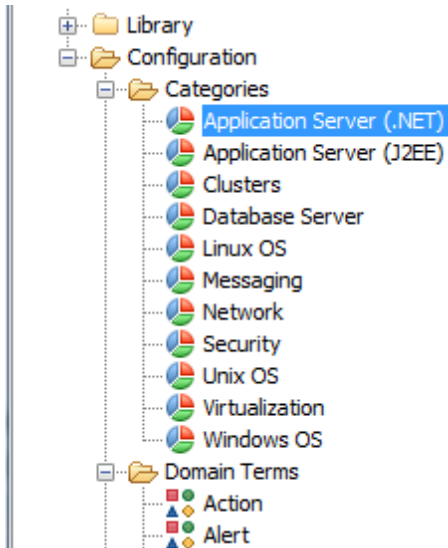
Description:

Classification of a flow.

Add Remove

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

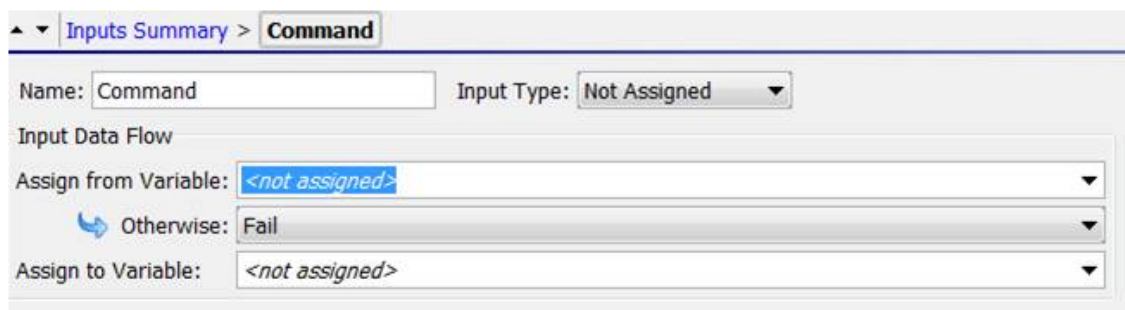
After upgrade, the items that previously appeared as rows in the **Category** domain term now appear as separate items in the **Categories** folder.



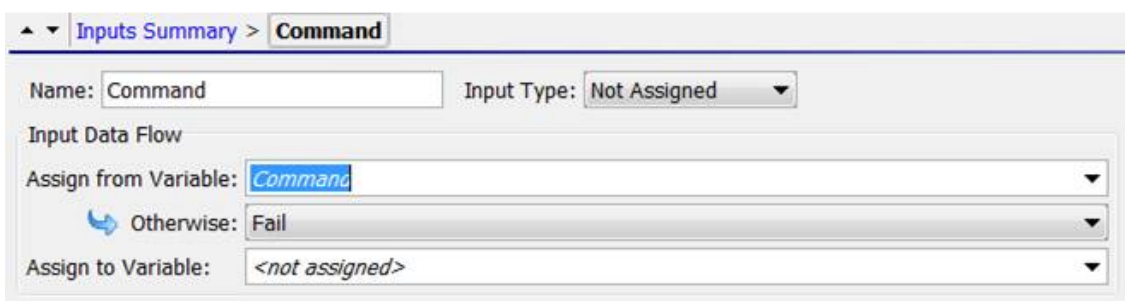
## "Assign From Variable" Value

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, the value of **Assign from Variable** is **Command**.



## Credentials Type is not Supported

In HP OO 10.0x, inputs of the type **Credentials** are no longer supported. All the options that previously existed under a **Credentials** input (**System Account**, **Logged-In User Credentials** and **Prompt User**) are now available under the **Single Value** input type.

The first screenshot shows the 'Inputs Summary' tab for an input named 'GG'. The 'Input Type' is set to 'Single Value'. The 'Input Data Flow' section shows 'Assign from Variable' as 'GG' and 'Assign to Variable' as 'GG'. The 'Otherwise' dropdown is set to 'Prompt User'.

The second screenshot shows the same input 'GG' with the 'Input Type' set to 'Single Value'. The 'Input Data Flow' section shows 'Assign from Variable' as 'GG' and 'Assign to Variable' as 'Prompt User'. The 'Input Properties' section shows 'Use Previous Step Result' and 'System Account' selected.

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.

The screenshot shows the 'Inputs Summary' tab for an input named 'something'. The 'Input Type' is set to 'Credentials'. The 'Input Data Flow' section shows 'Assign from Variable' as 'something' and 'Assign to Variable' as 'something'. The 'Otherwise' dropdown is set to 'System Account'. The 'Input Properties' section shows 'Use Previous Step Result' and 'System Account' selected. The 'Account Named' field is set to 'LinuxAdmin'.

In HP OO 10.0x, 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 **credUser** 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 an 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 *HP OO 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 HP OO 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 **credUser** 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.

ssh fields

Command: a

Argument: a

Environment: a

Host: \${host}

User: \${identity}

Password: \${identity}

Private Key File: a

Pty: ☒

Inputs



Input	Required	Type	Template
host	<input checked="" type="checkbox"/>	Not Assigned	No Assignment
identity	<input checked="" type="checkbox"/>	Credentials	Prompt user for credentials
Username	<input type="checkbox"/>	Single Value	Value: \${identity}

Field values were removed in HP OO 10.0x, 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.

Inputs

Add Input Remove Input  

Input	Required	Type	Template
identityUsername	<input checked="" type="checkbox"/>	Single Value	Prompt User
identityPassword	<input checked="" type="checkbox"/>	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.

## 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 HP OO 9.x from HP OO 10.0x.

As part of the HP OO 10.0x 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



## Phase 4: Synchronization of Content Data

After you have done the installation, performed the content upgrade, and deployed the content pack to Central, you need to synchronize the content-related data from HP OO 9.x, which was not upgraded automatically.

This includes:

- Schedules
- Studio permissions
- Content permissions
- History

This step may be performed in an iterative manner. You can synchronize different parts of your data separately.

Note that 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. For more information about the OOSH utility, see the *HP Operations Orchestration Shell User Guide*.

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

## Synchronizing Scheduling Data

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 HP OO 10.0x.

It is your responsibility to disable the scheduled flows in HP OO 9.x and enable those scheduled flows in HP OO 10.0x.

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

## Synchronizing Content Permissions Data

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

<b>HP OO 9.x</b>	<b>HP OO 10.0x:</b>
<b>Read</b>	<b>View</b>

HP OO 9.x	HP OO 10.0x:
<b>Execute</b>	<b>Run</b>
<b>Write</b>	Not upgraded.
<b>LinkTo</b>	Not upgraded.

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

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

During the installation, 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` batch file, to start the OOSH utility.

**Note:** The user who invokes upgrade-related commands from OOSH should 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.

## Synchronizing History

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, “13/12/2009”.
- `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

## Synchronizing Studio Data

Studio permission data is not upgraded during the installation of HP OO 10.0x. 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 HP OO 10.0x, 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.

